Aurelio Ghersi Raffaele Coraggio

ANALISI MATRICIALE DI STRUTTURE INTELAIATE

Il personal computer nel calcolo di edifici

CUEN

© CUEN 1990
© CUEN s.r.l. 1995
in Area Industrie della Cultura
80124 Napoli - Via Coroglio, 156
Tel. 081/2301019 pbx Fax 081/2301044

INDICE

PREMESSA ..... pag. 9

PARTE PRIMA – TEORIA

CAPITOLO 1
INTRODUZIONE

1. L'analisi strutturale ..... pag. 15
2. Metodi di analisi ..... pag. 17
3. Schematizzazione di un edificio ..... pag. 20
4. Procedimento risolutivo ..... pag. 23

CAPITOLO 2
LA SINGOLA ASTA

1. Generalità ..... pag. 25
2. Azioni agli estremi dell'asta ..... pag. 27
3. Componenti di movimento degli estremi dell'asta ..... pag. 31
4. Caratteristiche di sollecitazione e spostamento dei punti interni dell'asta ..... pag. 34
5. Relazione tra azioni e componenti di movimento degli estremi dell'asta ..... pag. 36
   - 5.1. Matrici fondamentali di rigidezza e di deformabilità ..... pag. 36
   - 5.2. Matrice di rigidezza nel sistema di riferimento locale ..... pag. 39
   - 5.3. Matrice di rigidezza nel sistema di riferimento globale ..... pag. 40

6. Matrice di rigidezza dell'asta: altri casi ..... pag. 41
6.1. Effetto della deformazione dovuta al taglio ..... pag. 41
6.2. Asta con una cerniera all'estremità ..... pag. 43
6.3. Asta incernierata ai due estremi (pendolo) ..... pag. 46
6.4. Trave su suolo elastico (alla Winkler) ..... pag. 49
6.5. Asta con tratti rigidi all'estremità ..... pag. 56
6.6. Nodo rigido non puntiforme (asta con tratti rigidi non coassiali) ..... pag. 61
7. Azioni di incastro perfetto ..... pag. 66
7.1. Carichi uniformemente distribuiti su tutta l'asta ..... pag. 68
7.2. Carichi concentrati in un punto dell'asta ..... pag. 68
7.3. Carichi distribuiti con andamento lineare su parte dell'asta .... pag. 69
7.4. Variazioni termiche ..... pag. 71
7.5. Effetto della deformazione dovuta al taglio ..... pag. 72
7.6. Asta con cerniere all'estremità ..... pag. 72
7.7. Trave su suolo elastico alla Winkler ..... pag. 74
7.8. Asta con tratti rigidi ..... pag. 74
8. Relazione tra carichi, azioni e componenti di movimento di estremità ..... pag. 75
9. Determinazione delle caratteristiche di sollecitazione nei punti interni dell'asta ..... pag. 75

CAPITOLO 3

L'INSIEME PIANO DI ASTE

1. Generalità ..... pag. 79
2. Azioni e componenti di movimento degli estremi delle aste ..... pag. 83
3. Le incognite del problema ..... pag. 85
4. Relazione tra incognite, componenti di movimento dei nodi e componenti di movimento degli estremi delle aste ..... pag. 89
5. Le azioni nodali ..... pag. 91
6. Relazione tra azioni nodali ed azioni agli estremi delle aste ..... pag. 93
7. Relazione tra carichi, azioni nodali e spostamenti nodali ..... pag. 94
8. Equilibrio nodale e risoluzione del problema ..... pag. 98
9. Determinazione diretta della matrice di rigidezza ..... pag. 99

CAPITOLO 4

L'INSIEME SPAZIALE DI TELAI PIANI

1. Macroelementi e sottostrutture ..... pag. 105

2. Condensazione della matrice di rigidezza
e del vettore azioni di incastro ..... pag. 108

3. Matrice di rigidezza traslante di un telaio piano ..... pag. 111

4. Matrice di rigidezza traslante di un telaio piano
nel riferimento spaziale ..... pag. 114

5. Matrice di rigidezza dell'insieme spaziale di telai ..... pag. 119

6. Equilibrio degli impalcati e risoluzione del problema ..... pag. 120

PARTE SECONDA – PROGRAMMI

CAPITOLO 1

INTRODUZIONE

1. Analisi strutturale mediante il personal computer ..... pag. 125

2. Caratteristiche del Quick Basic ..... pag. 126

2.1. Sequenza delle istruzioni ..... pag. 126

2.2. Codifica delle strutture logiche ..... pag. 128

2.3. I sottoprogrammi: routine e procedure ..... pag. 130

2.4. Altre istruzioni ..... pag. 133

3. Contenuto del dischetto ..... pag. 134

4. Utilizzazione dei programmi ..... pag. 136

5. Utilizzazione delle procedure e dei programmi in BASIC ..... pag. 137

CAPITOLO 2

OPERAZIONI MATRICIALI

1. Generalità ..... pag. 139

2. Procedimenti per l'esecuzione delle operazioni ..... pag. 141

3. Codifica ..... pag. 146
4. Esempio ..... pag. 165

CAPITOLO 3
SOLUZIONE DEI SISTEMI DI EQUAZIONI LINEARI

1. Generalità ..... pag. 169
2. Il metodo di riduzione di Gauss ..... pag. 171
   2.1. Le fasi del procedimento ..... pag. 171
   2.2. Riduzione e fattorizzazione ..... pag. 173
   2.3. Pivotaggio parziale ..... pag. 175
   2.4. Risoluzioni multiple ..... pag. 176
3. Applicazione all'analisi strutturale ..... pag. 178
   3.1. Ordine di esecuzione del procedimento ..... pag. 178
   3.2. Riduzione di una matrice simmetrica ..... pag. 179
   3.3. Riduzione di una matrice a banda ..... pag. 181
4. Codifica ..... pag. 185
5. Esempio ..... pag. 191

CAPITOLO 4
LA SINGOLA ASTA

1. Generalità ..... pag. 201
2. Codifica ..... pag. 203
3. Esempio: diagramma delle caratteristiche di sollecitazione ..... pag. 226
   3.1. Generalità ..... pag. 226
   3.2. Applicazione ..... pag. 230

CAPITOLO 5
L'INSIEME PIANO DI ASTE

1. Geometria dello schema e tipologia dei carichi ..... pag. 237
2. Descrizione del programma ..... pag. 240
3. Elenco delle variabili ..... pag. 247

4. Codifica ..... pag. 253
5. Modalità di utilizzazione del programma ..... pag. 277
6. Risultati forniti ..... pag. 280
7. Esempio ..... pag. 281

CAPITOLO 6
SCHEMI PARTICOLARI

1. Preprocessor e postprocessor ..... pag. 287
2. Il telaio a maglie rettangolari ..... pag. 289
3. Il telaio a maglie trapezie con trave di fondazione ..... pag. 292
4. Descrizione del programma ..... pag. 297
5. Elenco delle variabili ..... pag. 300
6. Codifica ..... pag. 307
7. Modalità di utilizzazione del programma ..... pag. 336
8. Risultati forniti ..... pag. 340
9. Esempio ..... pag. 341

CAPITOLO 7
L'INSIEME SPAZIALE DI TELAI PIANI

1. Generalità ..... pag. 349
2. Descrizione del programma ..... pag. 351
3. Elenco delle variabili ..... pag. 355
4. Codifica ..... pag. 358
5. Modalità di utilizzazione del programma ..... pag. 380
6. Risultati forniti ..... pag. 387

PARTE TERZA – APPLICAZIONI

EDIFICIO PER 54 APPARTAMENTI NEL COMUNE DI GIUGLIANO

Il progetto (F.Ghersi) ..... pag. 391

Relazione generale di calcolo ..... pag. 399

Elaborati numerici ..... pag. 405

BIBLIOGRAFIA ..... pag. 441

PREMESSA

Appena una decina di anni fa il computer era uno strumento riservato a pochi, disponibile solo in ambienti universitari, grosse industrie e qualche raro studio professionale. L'evoluzione tecnica lo ha in breve trasformato in uno strumento di massa, rendendolo un componente essenziale ed ineliminabile della progettazione strutturale. I programmi disponibili in tale campo, inizialmente limitati alla risoluzione di schemi relativamente semplici, con dati geometrici e di carico totalmente definiti dall'utente, sono notevolmente cresciuti in complessità e completezza. I compiti e l'atteggiamento dell'ingegnere civile vanno quindi man mano cambiando. Scomparsa la fatica bruta del calcolo manuale, è aumentata l'importanza della sua preparazione teorica, indispensabile per valutare consapevolmente le ipotesi e le scelte del programmatore. Sempre fondamentale rimane l'esperienza pratica, necessaria per prevedere l'ordine di grandezza dei risultati ed effettuare un riscontro con i valori forniti dall'elaboratore. Forse meno essenziale, ma ugualmente importante, è anche la padronanza delle tecniche di programmazione, che consente di giudicare con più maturità i programmi fatti dagli altri, adattarli alle proprie esigenze o crearne di specifici per le necessità individuali.

A questa rapida evoluzione tecnica corrisponde una lenta trasformazione dell'insegnamento universitario. Il seminario "Il personal computer nel calcolo di edifici", da me tenuto a partire dal 1983/84, organizzato nell'ambito dei corsi di Complementi di Tecnica delle Costruzioni della Facoltà di Ingegneria di Napoli, vuole essere un piccolo contributo in questa direzione. I suoi contenuti sono andati via via crescendo e precisandosi, anche se l'assetto raggiunto non può ancora essere considerato definitivo. Attualmente esso è diviso in tre parti. La prima, "introduzione all'uso del personal computer", vuol fornire agli studenti informazioni generali sugli elaboratori elettronici e su alcuni programmi di base (in particolare programmi per elaborazione di testi e di disegni), con l'obiettivo di metterli in grado di utilizzare il computer con maturità e consapevolezza dei suoi limiti. La seconda parte, "introduzione alla programmazione", ha lo scopo di fornire

i concetti necessari per lo sviluppo di programmi nel campo dell'ingegneria civile. Nella terza, infine, sono stati trattati nel corso degli anni svariati argomenti, come il progetto e la verifica di sezioni in c.a. presso e tensoinflesse, il calcolo di travi continue e travi di fondazione, la risoluzione matriciale di insiemi di aste, la risoluzione iterativa di schemi intelaiati piani e spaziali a maglie rettangolari.

Ho ritenuto che il trasporre gli argomenti delle lezioni in un insieme di testi dedicati alle applicazione del personal computer nel calcolo strutturale potesse risultare utile agli allievi che seguono il seminario ed anche, più in generale, a studenti ed ingegneri che indipendentemente dall'università sentono la necessità di affrontare queste problematiche. Al primo volume, "Introduzione alla programmazione", segue ora questo testo, realizzato con la valida collaborazione di Raffaele Coraggio, ex allievo del corso ed ora brillante ingegnere.

Il testo si pone sostanzialmente tre obiettivi. Innanzitutto, esso vuole costituire una introduzione generale alla metodologia matriciale di analisi delle strutture ed alla applicazione di essa ai casi particolari dei telai piani di forma generica e degli insiemi spaziali di detti telai, mutuamente collegati da impalcati rigidi. In secondo luogo, vuole mostrare in che modo la teoria viene tradotta in programmi per personal computer, per abituare il lettore ad una metodologia, di analisi e di programmazione, che potrà essergli utile per far fronte autonomamente a situazioni diverse da quelle trattate. Infine, vuole fornire una biblioteca di procedure generali e di programmi applicativi in linguaggio BASIC, che consente di analizzare, oltre che edifici regolari, anche quelli che presentino particolarità geometriche quali travi a ginocchio, piani sfalsati, impalcati non continui, piano di posa non orizzontale.

Il volume è stato suddiviso in tre parti. La prima contiene tutto l'apparato teorico dell'argomento. Dopo aver presentato brevemente i problemi connessi alla modellazione geometrica di una struttura, alle metodologie di analisi ed alla notazione matriciale, si prende innanzitutto in esame il comportamento di una singola asta nel piano, definendo le componenti di movimento e le azioni di estremità ed esprimendone il legame reciproco mediante la matrice di rigidezza. Per fornire al lettore un'ampia panoramica di situazioni diverse vengono discussi numerosi casi particolari (aste con tratti rigidi di estremità, aste su suolo alla Winkler, ecc.) che possono essere anche tralasciati in una prima lettura del testo. Si prende successivamente in esame l'insieme piano di aste, evidenziando il contributo fornito dalla singola asta alla matrice di rigidezza globale della struttura e pervenendo così all'espressione matriciale della condizione di equilibrio e quindi alla soluzione del problema statico. Si introduce infine la definizione di

sottostruttura, mostrando come condensare la matrice di rigidezza globale per ottenere le relazioni tra un insieme ridotto di azioni e componenti di movimento. Si ricava in tal modo la matrice di rigidezza traslante di un telaio piano e la si utilizza per l'analisi di un insieme spaziale di telai.

La seconda parte mostra come tradurre la trattazione teorica in un insieme di programmi. Vengono fatti ampi richiami all'impostazione generale di programmazione strutturata ed al linguaggio utilizzato (Quick Basic), nonché ai problemi numerici connessi alle operazioni matriciali ed alla risoluzione di sistemi di equazioni lineari. Le procedure presentate costituiscono un'ampia biblioteca matematica che può essere utilizzata per tutti i problemi che coinvolgono operazioni matriciali o sistemi di equazioni, anche indipendentemente dall'ambito strutturale. Si passa quindi all'applicazione diretta della teoria, fornendo procedure relative a tutti i tipi di asta considerati ed utilizzandone una parte nella realizzazione di un programma per la risoluzione di un insieme piano di aste (telaio generico o travatura reticolare). Le fasi di ingresso dati e stampa risultati vengono poi particolarizzate per una tipologia descrivibile in maniera più sintetica, il telaio a maglie rettangolari o trapezie con trave di fondazione. Si presenta infine un programma per la risoluzione di un insieme spaziale di telai piani, che costituisce uno strumento molto potente per l'analisi di schemi anche complessi, come edifici con piani sfalsati, piano di posa non orizzontale, travi a ginocchio.

La terza parte mostra in dettaglio l'utilizzazione del programma per il calcolo di un edificio reale, con l'obiettivo di chiarirne l'uso anche al lettore non interessato alla programmazione. Si è preso in esame un complesso di edifici per 54 appartamenti da realizzare nel comune di Giugliano, il cui progetto è stato commissionato dall'impresa Bengoa agli architetti Fabio Ghersi e Rita De Simone. La caratteristica più interessante dal punto di vista strutturale è costituita dalla presenza di impalcati sfalsati, situazione in genere non molto ben vista dagli strutturisti, un po' per le incertezze sul comportamento dinamico ma soprattutto per la difficoltà di trovare uno strumento di calcolo adeguato. Il progetto comprendeva solo un dimensionamento di massima della struttura; il calcolo dettagliato è stato quindi effettuato da noi, espressamente per questo libro. Nell'esposizione si è voluto evidenziare anche l'impostazione della relazione di calcolo, che segue i criteri utilizzati da me e da numerosi altri progettisti. Essa è costituita da una relazione generale che illustra i principi seguiti e da una serie di elaborati numerici che contengono il dettaglio dei calcoli svolti. Di questi ultimi è stata riportata solo una brevissima sintesi, che mette in luce esclusivamente i problemi connessi all'utilizzazione del programma ed all'esame dei risul-

tati da esso forniti; una trattazione più approfondita delle problematiche connesse alla progettazione di fabbricati in zona sismica è contenuta nel mio libro "Edifici antisismici con struttura intelaiata in cemento armato".

Nel concludere questo lavoro, desidero ringraziare le persone che maggiormente hanno contribuito a stimolarmi ed aiutarmi. Innanzitutto, il prof. Michele Pagano, che mi ha proposto di tenere ai suoi studenti questo seminario lasciandomi piena libertà nella scelta degli argomenti da trattare e nell'impostazione da dare alle lezioni. In secondo luogo tutti i colleghi e gli studenti con i quali ho avuto modo di discutere gli argomenti qui esposti, ricevendo spesso utili suggerimenti. Sono grato in particolare a Carmelo Roccasalva, che ha svolto una tesi di laurea sull'argomento, e ad Alfonso Impellicceri che ha riletto con cura le bozze della prima parte. Quanto sia stato grande il contributo di Raffaele Coraggio è mostrato dal fatto che, dopo aver affrontato l'argomento nella sua tesi di laurea, è diventato a pieno titolo coautore di questo libro. Un ringraziamento speciale va a mio fratello Fabio, che ha messo a disposizione il progetto illustrato nella terza parte, e al quale devo dare atto di essere uno dei pochi architetti con cui uno strutturista può andare d'accordo. L'ultimo pensiero infine per mia moglie Lia che ha sopportato con pazienza l'assorbimento totale con cui ho lavorato al libro e mi ha costantemente incoraggiato, aiutandomi a superare l'inevitabile stanchezza.

Aurelio Ghersi

PARTE PRIMA

TEORIA

CAPITOLO PRIMO

INTRODUZIONE

1. L'analisi strutturale

Gli edifici, come in generale ogni altro oggetto dell'attenzione dell'ingegnere, sono organismi complessi che possono essere esaminati solo mediante una opportuna schematizzazione degli elementi resistenti, cioè mediante la definizione di un modello fisico–matematico, lo *schema strutturale*. Lo studio di questo modello è detto *analisi strutturale*.

Nell'ambito dell'analisi strutturale si possono individuare due tipi di problemi. Il primo è quello dello studio di un singolo elemento, ad esempio una trave, una lastra, una volta. La meccanica del continuo analizza tali tipologie mediante algoritmi matematici appartenenti all'analisi infinitesimale, come le equazioni differenziali. Il secondo è quello dell'esame di insiemi di elementi mutuamente collegati, per il quale si utilizzano invece algoritmi matematici appartenenti all'algebra lineare, come i sistemi di equazioni lineari.

Nel passato ha avuto grande sviluppo soprattutto lo studio del continuo. Il predominio di tale aspetto è testimoniato dal fatto che in alcuni casi si preferiva analizzare strutture complesse, ad esempio le volte reticolari, facendo ricorso ad un continuo equivalente. Le difficoltà numeriche connesse alla risoluzione di sistemi di equazioni con molte incognite rendevano invece praticamente impossibile esaminare in maniera generale gli insiemi di elementi. Fino a non molti anni fa vi è stato quindi un fiorire

di procedimenti specifici per la risoluzione di particolari schemi strutturali come la trave continua, il telaio piano a maglie rettangolari a nodi fissi oppure a nodi spostabili, le travature reticolari.

La rapida diffusione dei calcolatori elettronici, che consentono di risolvere con facilità ed immediatezza sistemi lineari con un numero anche elevato di incognite, ha completamente ribaltato la situazione. Oggi è possibile affrontare in maniera molto generale l'analisi di insiemi di elementi, ovviamente partendo sempre dalla conoscenza del comportamento del singolo componente. Tale capacità è sfruttata spesso anche nell'esame di strutture continue che presentino una complessità tale da rendere difficile uno studio rigoroso mediante l'analisi infinitesimale. Il *metodo degli elementi finiti* consiste infatti nel discretizzare tali strutture, cioè nello scomporle in un insieme di elementi di dimensioni finite.

In tale ottica, si indicherà nel seguito col termine struttura un insieme di elementi deformabili collegati mutuamente in punti dello spazio detti nodi. L'impostazione seguita è anzi meglio evidenziata ribaltando i termini di tale enunciato e definendo *struttura* un insieme di nodi rigidi, liberi o vincolati, collegati tra loro mediante elementi deformabili. I nodi sono usualmente considerati puntiformi, cioè di dimensioni nulle. È però possibile, e in alcuni casi opportuno, considerare nodi di dimensioni finite; un tipico esempio si ritrova negli edifici, quando gli assi di travi e pilastri non convergono in un unico punto (fig.1.1).

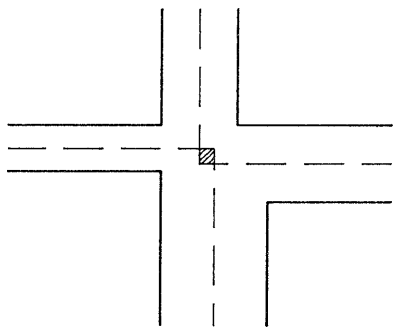


Fig. 1.1 — Nodo di un telaio, con trave a spessore e trave emergente e pilastri di differente sezione

Gli elementi deformabili possono essere in generale aste monodimensionali, schematizzazione usata quando una dimensione è nettamente prevalente sulle altre due, come nel caso delle travi di un edificio; oppure elementi bidi-

mensionali, come membrane o piastre; o, infine, elementi tridimensionali. L'importante è che per ciascuno di essi si conosca, eventualmente con ipotesi semplificative, la relazione tra stato tensionale e stato di deformazione, e quindi tra azioni applicate nei punti di estremità e componenti di movimento degli stessi. Nel caso di elementi monodimensionali, gli unici che verranno utilizzati nel presente testo, la relazione è fornita dalla teoria di De Saint–Venant. Nel caso di elementi bi o tri–dimensionali è necessario formulare delle ipotesi sul comportamento, ad esempio che lo stato tensionale vari linearmente, oppure che vari in tale maniera lo stato deformativo, e si avranno per essi in conseguenza modelli differenti.

2. Metodi di analisi

L'approccio più generale all'analisi di una struttura consiste nel considerarla scomposta nei singoli elementi, assumere come incognite le azioni (forze e momenti) esercitate agli estremi di ciascuno di essi e le componenti di movimento (spostamenti e rotazioni) di tali estremi, e scrivere quindi il necessario numero di equazioni di equilibrio e congruenza. Questa via presenta una complessità operativa elevata e non indispensabile. Si utilizzano pertanto due procedimenti, analoghi e duali rispetto alle grandezze in considerazione: il metodo delle forze e quello degli spostamenti.

Nel *metodo delle forze* (detto anche delle reazioni) si prende in esame uno schema principale (in genere, ma non necessariamente, isostatico) ottenuto effettuando sulla struttura delle sconnessioni e si assumono come incognite le azioni trasmesse attraverso esse. Si opera quindi su un sistema equilibrato, e si determina il valore delle incognite mediante la risoluzione del sistema di equazioni di congruenza che impone il rispetto delle connessioni eliminate, individuando in tal modo tra le infinite configurazioni equilibrate l'unica che sia anche congruente. Lo stato tensionale e deformativo dei singoli elementi è ottenuto come somma di quanto provocato sullo schema principale dai carichi esterni e dalle azioni determinate in corrispondenza delle sconnessioni.

Nel *metodo degli spostamenti* (detto anche dei cedimenti o delle deformazioni) si prende in esame uno schema principale ottenuto aggiungendo vincoli che impediscono spostamenti e rotazioni dei nodi della struttura (in genere, ma non necessariamente, tutti) e si assumono come incognite le componenti di movimento impedite. Si opera quindi su un sistema congruente, e si determina il valore delle incognite mediante la risoluzione del sistema di equazioni che impone l'equilibrio in assenza dei vincoli aggiunti,

cioè l'equilibrio tra i carichi nodali, le azioni sui nodi che insorgono nello schema principale per effetto dei carichi applicati ai singoli elementi (azioni di incastro perfetto) e le azioni sui nodi conseguenti alle componenti di movimento incognite, individuando in tal modo tra le infinite configurazioni congruenti l'unica che sia anche equilibrata. Lo stato tensionale e deformativo dei singoli elementi è ottenuto come somma di quanto provocato nello schema principale dai carichi esterni e dalle componenti di movimento determinate.

Il metodo delle forze ha avuto un grosso sviluppo nel passato, anche perché le sue incognite, forze e momenti, sono entità più familiari all'ingegnere, che è in genere in grado di valutarne l'ordine di grandezza e tenere quindi sotto controllo il procedimento numerico di soluzione. Il ricorso a tale metodo, accoppiato a tecniche iterative per la risoluzione del sistema di equazioni, è stato quindi il primo approccio per l'organizzazione di programmi di calcolo strutturale automatico. La scelta dello schema principale tra le numerose alternative possibili (fig.1.2, 1.3) è però un problema molto delicato e può condizionare sia la semplicità di scrittura delle equazioni di congruenza che l'accuratezza della soluzione. Questa impostazione è pertanto oggi limitata al caso di particolari tipologie strutturali, per le quali esistono dei criteri automatici per tale scelta.

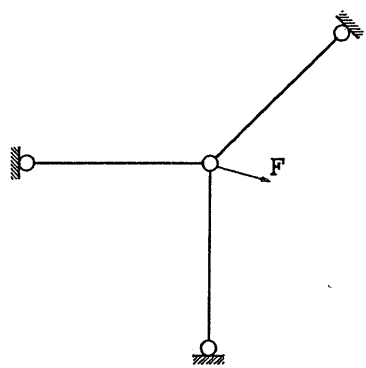


Fig. 1.2 — Schema strutturale

Il metodo degli spostamenti ha invece avuto, fino all'avvento dei computer, poche applicazioni, a volte anche parzialmente mascherate (come nel caso del metodo delle rigidità globali per la risoluzione di telai piani, che utilizza come incognite momenti di nodo e piano proporzionali alle rotazioni dei nodi e delle aste). Esso però consente una individuazione automatica delle incognite, che sono sempre costituite dalle componenti di movimento dei nodi (fig.1.4).

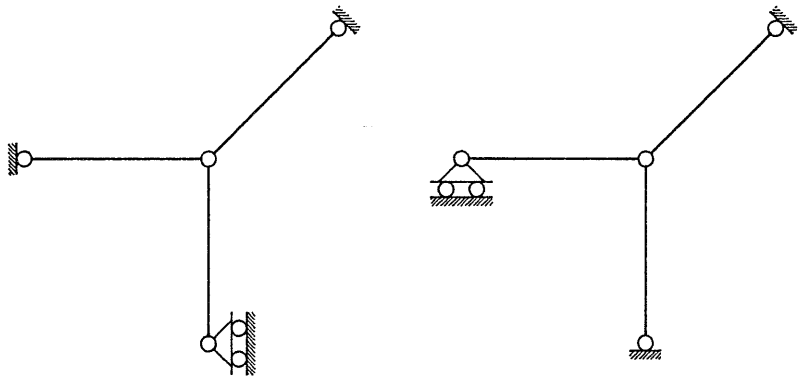


Fig. 1.3 — Due possibili schemi principali per il metodo delle forze

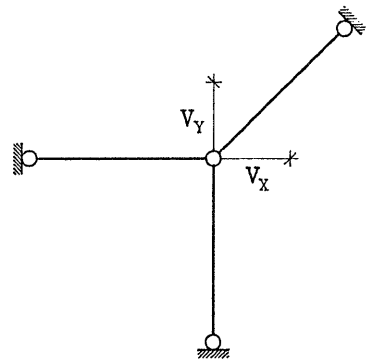


Fig. 1.4 — Incognite per il metodo degli spostamenti

Si presta quindi notevolmente all'impostazione schematica necessaria nella realizzazione di programmi per calcolatori elettronici e per tale motivo ha oggi soppiantato quasi totalmente l'altro procedimento.

L'impostazione classica del metodo delle forze o degli spostamenti prevede la scrittura delle equazioni di congruenza o di equilibrio in maniera esplicita, in genere basandosi su un'analisi teorica preliminare valida solo per il particolare schema geometrico in esame. Negli ultimi decenni, parallelamente allo sviluppo dell'impiego dell'elaborazione elettronica si è avuto una rivoluzione formale dell'approccio al problema, consistente nell'uso della notazione matriciale; oggi si parla quindi di *analisi matriciale delle strutture*. Ovviamente, tale notazione non introduce nulla di nuovo, poiché i principi di base e le equazioni cui si giunge sono sempre gli stessi. Il primo impatto per chi non abbia con essa sufficiente familiarità è indubbiamente duro, e può a volte sembrare difficile mantenere il legame tra nuovi simboli e realtà

fisica da essi rappresentata. Una volta acquisito un minimo di padronanza ci si rende però conto di come essa consenta una migliore visione globale ed una notevole sintesi espressiva, sia nell'analisi dei problemi che nella loro traduzione in programmi per computer. Diventa quindi agevole affrontare l'analisi strutturale di schemi molto generali.

Infine, un'ultima innovazione riguarda il modo di risolvere il sistema di equazioni che caratterizza il problema strutturale. L'approccio manuale era sempre effettuato in maniera iterativa, ad esempio col metodo di Gauss–Siedel, che presenta il vantaggio di possedere un preciso significato fisico. Nel metodo degli spostamenti esso equivale infatti ad eliminare di volta in volta uno dei vincoli aggiuntivi e calcolare le deformazioni che insorgono per assicurare l'equilibrio. Il ricorso a calcolatori ha reso possibile l'uso di altri algoritmi dell'analisi numerica, come ad esempio il metodo di riduzione di Gauss, che consentono di risolvere il sistema con un numero predefinito di operazioni. È però opportuno sottolineare che nessuna delle due impostazioni è esente da problemi applicativi. Nella prima, la convergenza del procedimento iterativo è fortemente condizionata dai rapporti di rigidezza degli elementi strutturali. Nella seconda, gli errori di arrotondamento possono esaltarsi in casi particolari (matrici mal condizionate) inficiando la correttezza dei risultati.

3. Schematizzazione di un edificio

La scelta del modello strutturale da utilizzare per valutare il comportamento di un edificio reale è sempre complessa e dipende in generale dalla sua geometria, dagli elementi non strutturali in esso presenti (tramezzi, tompagni, ecc.), dai carichi cui esso è soggetto, dal terreno e dalla tipologia della struttura di fondazione.

In assenza di carichi orizzontali e con forti ipotesi limitative (presenza di robuste tompagnature o altri elementi irrigidenti fuori calcolo che rendono gli impalcati fissi, luci contenute, numero limitato di piani) è possibile considerare indipendenti i diversi componenti strutturali e adottare per essi schemi monodimensionali, come quello di trave continua ad appoggi fissi per le travi e i solai o di singolo elemento soggetto ad azioni assiali per i pilastri.

In presenza di carichi orizzontali alcune limitazioni geometriche (struttura intelaiata con doppio asse di simmetria in pianta, impalcati orizzontali continui e molto rigidi nel loro piano) rendono possibile esaminare il comportamento della struttura separatamente in due direzioni ortogonali adottando schemi bidimensionali, come quello che ipotizza che la struttura

sia costituita da un insieme di telai piani mutuamente collegati da pendoli inestensibili (telaio traslante o treno di telai).

Nei casi più frequenti occorre però un modello che consenta di valutare l'effettivo comportamento tridimensionale dell'edificio. Lo schema più generale di telaio spaziale è quello che vede la struttura come un insieme di aste nello spazio, con eventuali vincoli mutui tra i nodi che rappresentano la presenza di impalcati indeformabili nel loro piano. Un tale modello, eventualmente ancor più generalizzato con l'aggiunta di elementi bidimensionali, lastre o piastre, può essere analizzato mediante programmi come il SAP o il NASTRAN, il primo dei quali è ormai disponibile anche per i più diffusi personal computer. La maggiore generalità comporta ovviamente problemi operativi, come l'elevato ingombro di memoria, la lunghezza dei tempi di soluzione, la complessità della fase di input e dell'esame dei risultati. I primi due problemi perdono man mano rilevanza, grazie alla introduzione di calcolatori sempre più potenti e veloci. Contemporaneamente, la diffusione di pre e post–processor grafici rende più agevole il controllo dei dati e la valutazione dell'attendibilità dei risultati. Nonostante ciò, questa impostazione è ancora molto onerosa e quindi inadatta per un uso quotidiano da parte del progettista strutturale.

Una possibile semplificazione consiste nel considerare strutture spaziali con impalcati orizzontali planimetricamente indeformabili, pilastri verticali e travi poste solo a livello dell'impalcato. Una tale impostazione, adottata da programmi come l'ETABS, è di facile e rapido uso e consente di analizzare anche edifici con piante molto irregolari. Non può però essere utilizzata in presenza di travi fuori piano, come quelle delle scale, di tetti a falde inclinate, impalcati sfalsati, o piano di posa a quota non costante.

In alternativa si può considerare la struttura spaziale come costituita da un insieme di telai piani, ed eventualmente anche nuclei irrigidenti tridimensionali, collegati da impalcati indeformabili. In questo caso non si tiene conto innanzitutto della congruenza verticale tra i nodi comuni a due o più telai incidenti, cioè si accettano spostamenti verticali differenti per uno stesso nodo a seconda che esso sia considerato appartenente ad un telaio oppure ad un altro. Se però il nodo appartiene in entrambi i telai a pilastrate verticali, l'elevata rigidezza estensionale dei pilastri rende in assoluto modesti gli spostamenti verticali e quindi poco rilevante l'effetto di tale incongruenza. In secondo luogo, viene trascurato l'accoppiamento flessionale e torsionale delle travi di due telai incidenti, cioè il momento flettente e torcente che nasce nelle aste di un telaio per effetto di una rotazione nel piano dell'altro telaio. Si osservi però che la flessione anzidetta è rigorosamente nulla se i due telai sono tra loro perpendicolari in pianta e, se non

lo sono, è tanto più modesta quanto più ci si avvicina a tale situazione. L'entità della torsione dipende invece dalla rigidezza torsionale delle travi, che è usualmente molto minore rispetto a quella flessionale. Questa impostazione cade quindi in difetto solo nel caso di piante molto irregolari, nelle quali non sia possibile individuare due famiglie di telai sostanzialmente (ma non rigorosamente) ortogonali.

Chiedersi quale tra i modelli qui presentati sia in assoluto il migliore per la pratica professionale è una domanda sostanzialmente scorretta perché sottointende la convinzione di poter utilizzare per qualsiasi applicazione sempre la stessa schematizzazione, ovvero lo stesso programma per il calcolatore. Bisogna invece rendersi conto che edifici diversi richiedono differenti modelli, e quindi che l'organizzazione di uno studio professionale richiede la disponibilità di una gamma di programmi sufficientemente ampia per soddisfare tutte le esigenze.

Analizzare in uno stesso testo tutti gli schemi innanzi descritti è un compito molto oneroso. In questo volume ci si limita pertanto all'esame di due modelli. Il primo è quello di telaio piano, o più propriamente di *insieme piano di aste*, perché in esso sono compresi anche quegli schemi in cui la resistenza della struttura è affidata prevalentemente agli sforzi estensionali (travature reticolari) anziché a quelli flesso–taglianti. L'argomento è trattato in numerosi testi, sia in lingua inglese che in italiano, nei quali sono però poco evidenziati alcuni problemi particolari ma utili nell'applicare il modello agli edifici, come quello della indeformabilità estensionale dei traversi. La sua semplicità lo rende inoltre particolarmente adatto per un primo approccio all'analisi matriciale da parte dei molti studenti o ingegneri che non hanno ancora familiarità con questa metodologia. Si è preferito affrontare il problema direttamente in maniera generale. Contemporaneamente si è però ritenuto opportuno dedicare ampio spazio alla descrizione di pre–processor (o interfaccia di ingresso dati) che consentono di particolarizzare la preparazione dei dati al singolo caso in esame; è così possibile, ad esempio, descrivere la geometria di telai a maglie rettangolari fornendo direttamente luci delle campate e altezze degli interpiani, anziché le coordinate di tutti i nodi. A ciascun pre–processor corrisponde un post–processor (o interfaccia di uscita), che ad esempio presenta i risultati individuando travi e pilastri mediante la numerazione in carpenteria anziché con un numero progressivo.

Il secondo modello è quello di *insieme spaziale di telai piani.* Il rilievo ad esso dato dalla letteratura tecnica corrente, in particolare da quella italiana, è molto limitato. Il suo studio presenta invece interessanti risvolti teorici, come quello della scomposizione di una struttura in sottostrutture

più semplici, ma soprattutto grosse potenzialità applicative. Nel testo si mostra quindi esplicitamente come esso possa essere utilizzato per l'analisi di edifici con travi a ginocchio, impalcati sfalsati o piano di posa a quota non costante.

4. Procedimento risolutivo

Entrambi i modelli strutturali qui descritti sono costituiti da nodi rigidi collegati da elementi monodimensionali, cioè da solidi in cui una dimensione sia predominante rispetto alle altre due. Questi elementi vengono indicati col termine *aste*.

Per adottare nella risoluzione dello schema il metodo degli spostamenti, occorre innanzitutto analizzare il comportamento della singola asta, cioè definire le relazioni che intercorrono tra le azioni applicate ai suoi estremi e le componenti di movimento degli stessi (matrice di rigidezza dell'asta), nonché le azioni di estremità che insorgono in conseguenza ai carichi applicati all'interno dell'asta quando il movimento dei suoi estremi è impedito (azioni di incastro perfetto).

Il passo successivo consiste nel determinare il comportamento complessivo di un insieme piano di aste, cioè le relazioni esistenti tra le azioni applicate ai nodi e le componenti di movimento di essi (matrice di rigidezza dell'insieme).

Nel caso di un telaio piano, una volta calcolate le azioni nodali (sia esterne che derivate da carichi agenti sulle aste) è possibile utilizzare le relazioni anzidette per determinare le componenti di movimento nodale e da queste risalire alle caratteristiche di sollecitazione nelle aste.

Volendo analizzare un insieme spaziale di telai, occorre invece modificare tali relazioni in modo da ottenere per il singolo telaio piano un collegamento diretto tra forze orizzontali e spostamento dei traversi (matrice di rigidezza laterale o traslante dello schema piano); occorre inoltre valutare le azioni orizzontali che insorgono per effetto dei carichi (nodali e sulle aste) quando gli spostamenti dei traversi sono impediti. Si può quindi passare all'esame dell'insieme spaziale di telai, determinando l'analogo collegamento esistente tra azioni orizzontali e componenti di movimento degli impalcati (matrice di rigidezza laterale dello schema spaziale). Utilizzando queste relazioni, si può risalire dai carichi orizzontali alle componenti di movimento degli impalcati ed agli spostamenti dei singoli traversi, e quindi alle componenti di movimento nodale ed alle caratteristiche di sollecitazione delle aste di ciascun telaio piano.

CAPITOLO SECONDO

LA SINGOLA ASTA

1. Generalità

Caratteristica comune alle due tipologie strutturali prese in esame in questo testo è che le aste da cui esse sono costituite rientrino nell'ambito dei sistemi monodimensionali a comportamento piano. Dal punto di vista geometrico ciò richiede che l'asse di ciascuna asta sia contenuto in un piano e che la sua sezione retta sia simmetrica rispetto ad esso; in tutte le applicazioni verranno anzi considerate, in maniera più restrittiva, solo aste ad asse rettilineo. In secondo luogo, è necessario che l'asta sia caricata simmetricamente rispetto al piano, in modo che gli spostamenti dell'asse siano contenuti in esso. Questa condizione è implicitamente rispettata dal modello di telaio piano. Nel caso dell'insieme spaziale di telai essa equivale a considerare irrilevante, e quindi trascurare, la capacità di ciascun telaio di assorbire carichi al di fuori del piano cui esso appartiene.

Per la validità della trattazione occorre effettuare alcune ulteriori ipotesi. La prima è quella cosiddetta di piccoli spostamenti. Questi devono essere tali da consentire di determinare le caratteristiche di sollecitazione facendo riferimento alla struttura indeformata. In secondo luogo, il materiale deve essere linearmente elastico. Queste due ipotesi rendono valido il principio di sovrapposizione degli effetti, che consente di valutare separatamente lo stato tensionale e deformativo provocato da un singolo carico o da uno spostamento imposto. Si ipotizza infine la conservazione delle sezioni

piane; ciò è vicino al vero quando gli spostamenti dovuti alla sollecitazione locale di taglio sono piccoli rispetto a quelli dovuti alla deformazione complessiva della struttura, cioè quando la lunghezza dell'asta predomina sulle dimensioni della sezione trasversale.

Avendo a che fare con sistemi piani, si userà come riferimento una terna ortogonale destrorsa xyz tale da contenere le aste nel piano xy. Essa viene denominata *sistema di riferimento globale* perché è valida per tutte le aste dell'insieme. Nelle figure si rappresenterà l'asse x orizzontale ed orientato da sinistra verso destra, quello y verticale ed orientato dal basso verso l'alto (fig.2.1); il verso positivo dell'asse z è di conseguenza quello uscente dal foglio.

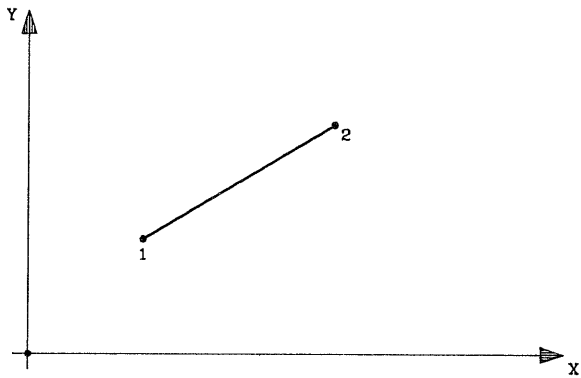


Fig. 2.1 — Sistema di riferimento globale

Nell'esaminare una singola asta, è in genere più comodo utilizzare un diverso sistema di riferimento, denominato *sistema di riferimento locale* perché specifico dell'asta in esame e strettamente legato alla sua geometria. L'origine è infatti posta in corrispondenza del suo primo estremo; l'asse x coincide con l'asse dell'asta ed è orientato nel verso che va dal primo al secondo estremo di questa; l'asse y è ad esso perpendicolare ed orientato in verso tale che il semiasse positivo x possa sovrapporsi a quello y con una rotazione antioraria di 90 gradi (fig.2.2). In tutti i casi in cui è necessario distinguere i due sistemi di riferimento, gli assi di quello locale verranno contraddistinti con un apice ($x'y'$).

Nei paragrafi che seguono viene analizzato il comportamento di una generica asta. Ciò richiede innanzitutto la definizione delle azioni (forze e momenti) esercitate ai suoi estremi e delle componenti di movimento (spostamenti e rotazioni) di tali punti, che in termini matriciali vengono rispettivamente indicate mediante i vettori **S** ed **s**. Poiché queste grandezze

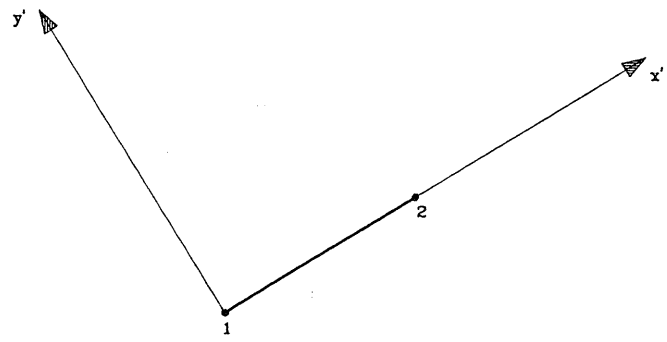


Fig. 2.2 — Sistema di riferimento locale

possono essere riferite ad entrambi i sistemi di assi innanzi descritti, vengono individuate le relazioni che consentono il passaggio dall'uno all'altro.

Sono poi richiamate le equazioni differenziali che legano carichi, caratteristiche di sollecitazione e componenti di movimento dei punti dell'asse dell'asta, note dalla Scienza delle Costruzioni. Dalla loro risoluzione si determina la relazione lineare che insorge, in assenza di carichi, tra azioni e componenti di movimento di estremità, sintetizzata dalla matrice di rigidezza $\mathbf{k}$ dell'asta. Questa viene esplicitata per i casi che più frequentemente possono ricorrere nella pratica professionale.

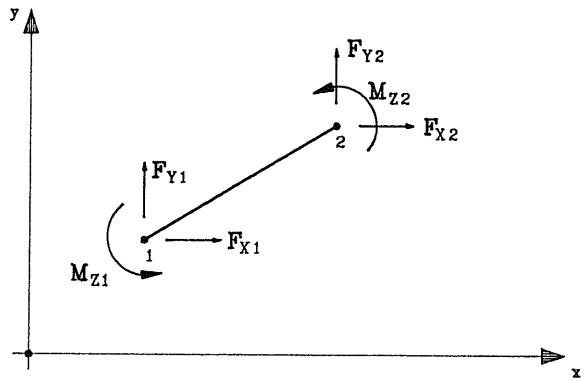
Vengono quindi definite le azioni di incastro perfetto $\bar{\mathbf{S}}$, che insorgono per effetto dei carichi agenti sull'asta quando le componenti di movimento dei suoi estremi sono impedite. Per comodità del lettore sono richiamate le espressioni analitiche corrispondenti ad una vasta gamma di tipologie di carico. Il principio di sovrapposizione degli effetti rende così possibile esprimere il legame tra carichi, azioni e componenti di movimento di estremità mediante la relazione generale $\mathbf{S} = \bar{\mathbf{S}} + \mathbf{k}\,\mathbf{s}$.

Sono infine riportate le formulazioni che consentono di determinare le caratteristiche di sollecitazione in un qualsiasi punto dell'asta, in funzione dei carichi e delle azioni di estremità.

2. Azioni agli estremi dell'asta

A ciascun estremo di un'asta si può considerare agente, in generale, una forza comunque orientata ed una coppia. Quando si è interessati ad evidenziare le relazioni tra l'asta in esame e le altre dell'insieme, risulta conveniente descrivere queste azioni mediante le sei componenti rispetto al sistema di riferimento globale, mostrate col loro verso positivo nella figu-

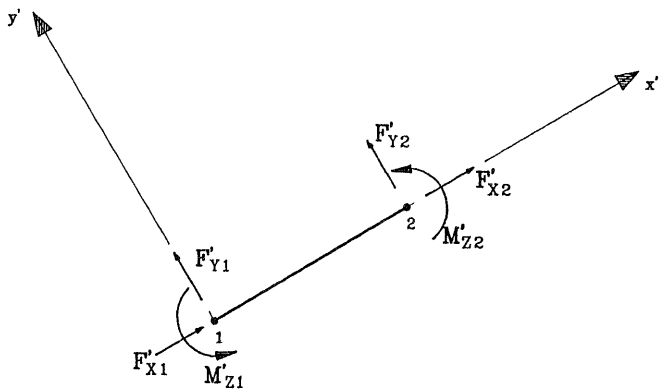
ra 2.3 e racchiuse nel vettore $\mathbf{S}_g$.



$$\mathbf{S}_g = \begin{bmatrix} F_{x_1} \\ F_{y_1} \\ M_{z_1} \\ F_{x_2} \\ F_{y_2} \\ M_{z_2} \end{bmatrix}$$

Fig. 2.3 — Azioni di estremità nel riferimento globale

Quando invece si analizza un'asta indipendentemente dalle altre è preferibile descrivere le azioni mediante le componenti rispetto al sistema di riferimento locale, mostrate col loro verso positivo nella figura 2.4 e racchiuse nel vettore $\mathbf{S}_\ell$.



$$\mathbf{S}_\ell = \begin{bmatrix} F'_{x_1} \\ F'_{y_1} \\ M'_{z_1} \\ F'_{x_2} \\ F'_{y_2} \\ M'_{z_2} \end{bmatrix}$$

Fig. 2.4 — Azioni di estremità nel riferimento locale

Il passaggio dal secondo al primo insieme di grandezze può essere espresso in funzione dall'angolo α, che rappresenta l'inclinazione dell'asse dell'asta rispetto all'asse x del riferimento globale (positivo se il semiasse

positivo x deve ruotare in verso antiorario per sovrapporsi ad x'):

$$\begin{aligned} F_{x1} &= \cos\alpha\ F'_{x1} - \sin\alpha\ F'_{y1} & F_{x2} &= \cos\alpha\ F'_{x2} - \sin\alpha\ F'_{y2} \\ F_{y1} &= \sin\alpha\ F'_{x1} + \cos\alpha\ F'_{y1} & F_{y2} &= \sin\alpha\ F'_{x2} + \cos\alpha\ F'_{y2} \\ M_{z1} &= M'_{z1} & M_{z2} &= M'_{z2} \end{aligned}$$

In termini matriciali si può scrivere

$$\mathbf{S}_g = \mathbf{L}_d \mathbf{S}_\ell \tag{2.1}$$

indicando con $\mathbf{L}_d$ la *matrice di rotazione*, che consente di tener conto della rotazione del sistema di riferimento,

$$\mathbf{L}_d = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & 0 & 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

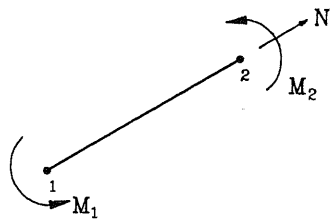
Ad ogni vettore $\mathbf{S}_\ell$ corrisponde mediante la relazione innanzi scritta un unico vettore $\mathbf{S}_g$. Si può facilmente constatare che la corrispondenza tra i due insiemi è biunivoca e si possono quindi scrivere, in maniera analoga, le espressioni che consentono il passaggio inverso, da $\mathbf{S}_g$ a $\mathbf{S}_\ell$. In termini matriciali si ottiene

$$\mathbf{S}_\ell = \mathbf{L}_d^T \mathbf{S}_g$$

Si noti come la trasformazione inversa avvenga in questo caso mediante la trasposta della matrice di rotazione, che consentiva la trasformazione diretta, e che si ha quindi

$$\mathbf{L}_d^{-1} = \mathbf{L}_d^T$$

Quando si descrivono le azioni agli estremi dell'asta mediante il vettore $\mathbf{S}_g$ oppure $\mathbf{S}_\ell$, non è possibile assegnare alle sei componenti valori arbitrari. Occorre infatti che l'insieme costituito da azioni di estremità e carichi sull'asta rispetti tre condizioni di equilibrio (tre perché si ha a che fare con uno schema piano). Per definire in maniera completa ed univoca le azioni sono quindi sufficienti tre sole componenti, che costituiscono l'insieme minimo, o insieme fondamentale, di azioni di estremità. La figura 2.5 mostra le grandezze scelte (tra le possibili alternative), racchiuse nel vettore $\mathbf{S}_f$.



$$\mathbf{S}_f = \begin{bmatrix} M_1 \\ M_2 \\ N \end{bmatrix}$$

Fig. 2.5 — Insieme fondamentale di azioni

Il passaggio dal vettore $\mathbf{S}_f$ a $\mathbf{S}_\ell$ richiede la conoscenza dei carichi agenti sull'asta. Per un'asta scarica si ha

$$\begin{aligned} F'_{x1} &= -N & F'_{x2} &= N \\ F'_{y1} &= \frac{M_1}{\ell} + \frac{M_2}{\ell} & F'_{y2} &= -\frac{M_1}{\ell} - \frac{M_2}{\ell} \\ M'_{z1} &= M_1 & M'_{z2} &= M_2 \end{aligned}$$

In termini matriciali si può scrivere

$$\mathbf{S}_\ell = \mathbf{T}\,\mathbf{S}_f \tag{2.3}$$

indicando con $\mathbf{T}$ la *matrice di trasformazione* che consente tale passaggio, funzione esclusivamente della lunghezza ℓ dell'asta

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & -1 \\ \frac{1}{\ell} & \frac{1}{\ell} & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{\ell} & -\frac{1}{\ell} & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.4}$$

Anche in questo caso ad ogni vettore $\mathbf{S}_f$ corrisponde un unico vettore $\mathbf{S}_\ell$. Esistono però infiniti vettori $\mathbf{S}_\ell$ che non corrispondono ad alcun $\mathbf{S}_f$ (perché non rispettano le condizioni di equilibrio). La corrispondenza tra i due insiemi non è quindi biunivoca, e non è possibile effettuare la trasformazione inversa.

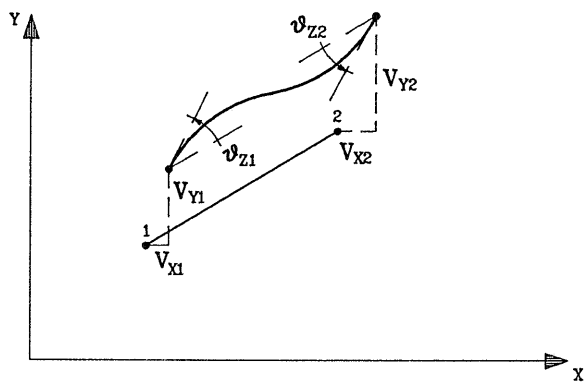
Utilizzando le espressioni innanzi riportate, si ottiene la relazione che lega l'insieme fondamentale di azioni alle loro componenti nel sistema di riferimento globale

$$\mathbf{S}_g = \mathbf{L}_d \; \mathbf{T} \; \mathbf{S}_f \tag{2.5}$$

Ovviamente anche questa relazione non è invertibile.

3. Componenti di movimento degli estremi dell'asta

Il movimento di ciascun estremo di un'asta può essere definito, in generale, mediante una traslazione comunque diretta ed una rotazione. Quando si è interessati ad evidenziare le relazioni tra l'asta in esame e le altre dell'insieme, risulta conveniente descrivere questo moto mediante le sei componenti rispetto al sistema di riferimento globale, mostrate in scala amplificata col loro verso positivo nella figura 2.6 e racchiuse nel vettore $\mathbf{s}_g$.



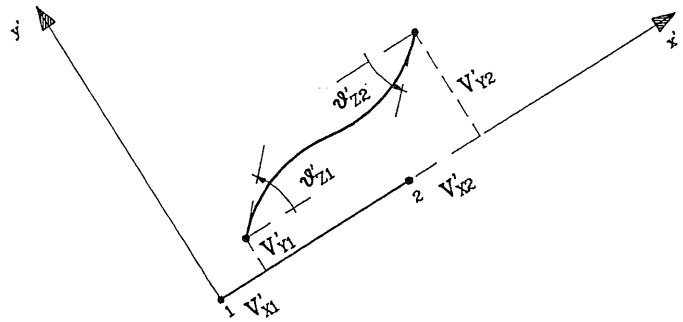
$$\mathbf{s}_g = \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ \theta_{z_1} \\ v_{x_2} \\ v_{y_2} \\ \theta_{z_2} \end{bmatrix}$$

Fig. 2.6 — Componenti di movimento nel riferimento globale

Quando invece si analizza un'asta indipendentemente dalle altre è preferibile descriverne il movimento mediante le componenti rispetto al sistema di riferimento locale, mostrate col loro verso positivo nella figura 2.7 e racchiuse nel vettore $\mathbf{s}_\ell$.

La corrispondenza tra $\mathbf{s}_g$ ed $\mathbf{s}_\ell$ è biunivoca ed è fornita, in maniera identica a quanto mostrato nel paragrafo precedente, da

$$\mathbf{s}_g = \mathbf{L}_d \; \mathbf{s}_\ell \qquad \mathbf{s}_\ell = \mathbf{L}_d^T \; \mathbf{s}_g \tag{2.6}$$



$$\mathbf{s}_\ell = \begin{bmatrix} v'_{x_1} \\ v'_{y_1} \\ \theta'_{z_1} \\ v'_{x_2} \\ v'_{y_2} \\ \theta'_{z_2} \end{bmatrix}$$

Fig. 2.7 — Componenti di movimento nel riferimento locale

A differenza di quanto avviene per le azioni di estremità, è possibile assegnare qualsiasi valore alle sei componenti di movimento, siano esse riferite al sistema globale o a quello locale. Occorre però rilevare che lo stato deformativo dell'asta non è influenzato da un suo moto rigido, ed è quindi identico per tutti quegli insiemi di componenti di movimento che differiscono tra loro solo per un tale moto. Per definire in maniera completa ed univoca la deformazione di un'asta sono pertanto sufficienti tre sole grandezze, che costituiscono l'insieme minimo, o insieme fondamentale, di componenti di movimento. La figura 2.8 mostra le grandezze scelte, racchiuse nel vettore $\mathbf{s}_f$.

Ad ogni vettore $\mathbf{s}_\ell$ corrisponde un unico vettore $\mathbf{s}_f$, mediante le relazioni

$$\varphi_1 = \theta'_{z1} - \frac{v'_{y2} - v'_{y1}}{\ell} = \frac{v'_{y1}}{\ell} + \theta'_{z1} - \frac{v'_{y2}}{\ell}$$

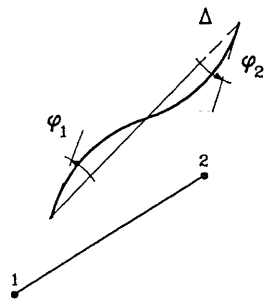
$$\varphi_2 = \theta'_{z2} - \frac{v'_{y2} - v'_{y1}}{\ell} = \frac{v'_{y1}}{\ell} - \frac{v'_{y2}}{\ell} + \theta'_{z2}$$

$$\Delta = v'_{x2} - v'_{x1}$$

In termini matriciali si può scrivere

$$\mathbf{s}_f = \mathbf{T}^T \, \mathbf{s}_\ell \tag{2.7}$$

essendo $\mathbf{T}^T$ la trasposta della matrice di trasformazione già definita nel paragrafo precedente.



$$\mathbf{s}_f = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \Delta \end{bmatrix}$$

Fig. 2.8 — Insieme fondamentale di componenti di movimento

Ad ogni vettore $\mathbf{s}_f$ corrispondono invece infiniti vettori $\mathbf{s}_\ell$, e non è quindi possibile la trasformazione inversa.

Utilizzando le espressioni innanzi riportate si ottiene la relazione, non invertibile, che lega le componenti di movimento nel sistema di riferimento globale al corrispondente insieme fondamentale

$$\mathbf{s}_f = (\mathbf{L}_d\ \mathbf{T})^T\ \mathbf{s}_g \tag{2.8}$$

I tre diversi insiemi di componenti di movimento definiti in questo paragrafo sono duali con gli insiemi di azioni del paragrafo precedente, perché ogni componente di movimento è associata alla corrispondente azione. Anche tra le relazioni che legano mutuamente i primi insiemi o i secondi vi è una stretta analogia. Non si tratta di un fatto casuale, bensì della conseguenza di una legge generale, che può essere denominata *principio di dualità tra forze e spostamenti.*

Si considerino infatti due insiemi $\mathbf{S}_1$ ed $\mathbf{S}_2$ di forze (e coppie) agenti su un corpo, staticamente equivalenti e legati tra loro da una trasformazione lineare, non necessariamente invertibile, espressa matricialmente da

$$\mathbf{S}_2 = \mathbf{B}\ \mathbf{S}_1$$

Ai due sistemi di forze sono associati i due insiemi di spostamenti (e rotazioni) $\mathbf{s}_1$ ed $\mathbf{s}_2$, costituiti dalle componenti di movimento corrispondenti alle singole azioni. Applicando al corpo gli spostamenti $\mathbf{s}_1$, il lavoro W_1 compiuto dalle forze $\mathbf{S}_1$ è pari alla somma dei prodotti di forze per spostamenti corrispondenti, cioè

$$W_1 = \mathbf{S}_1^T\ \mathbf{s}_1$$

Per il principio dei lavori virtuali, essendo i due sistemi di forze e di spostamenti tra loro equivalenti, il lavoro compiuto da ciascun sistema deve essere lo stesso. Deve quindi essere

$$\mathbf{S}_1^T \ \mathbf{s}_1 = \mathbf{S}_2^T \ \mathbf{s}_2$$

La relazione tra i due insiemi di azioni può indicarsi anche con

$$\mathbf{S}_2^T = \mathbf{S}_1^T \ \mathbf{B}^T$$

che sostituita nella precedente fornisce

$$\mathbf{S}_1^T \ \mathbf{s}_1 = \mathbf{S}_1^T \ \mathbf{B}^T \ \mathbf{s}_2$$

e quindi

$$\mathbf{s}_1 = \mathbf{B}^T \ \mathbf{s}_2$$

Si può pertanto affermare in generale che, se esiste una trasformazione lineare da un sistema di forze $\mathbf{S}_1$ ad uno equivalente $\mathbf{S}_2$, espressa mediante una matrice $\mathbf{B}$, è sempre possibile trasformare il sistema di spostamenti $\mathbf{s}_2$ (associato ad $\mathbf{S}_2$) nell'equivalente $\mathbf{s}_1$ (associato ad $\mathbf{S}_1$) mediante la trasposta della stessa matrice $\mathbf{B}$.

4. Caratteristiche di sollecitazione e spostamento dei punti interni dell'asta

Le ipotesi formulate nel primo paragrafo consentono di applicare all'asta la teoria della trave di De Saint-Venant, che fornisce la soluzione delle equazioni dell'equilibrio elastico per le sollecitazioni semplici di trazione, flessione e flessione+taglio, e di utilizzare tutte le relazioni che da essa derivano per le travi piane ad asse rettilineo.

Si consideri un'asta nel sistema di riferimento locale (fig.2.9), la cui sezione retta abbia area A e momento d'inerzia I, eventualmente variabili in funzione della distanza x dal primo estremo. Siano inoltre χ il fattore di taglio, parametro adimensionale che tiene conto del discostarsi delle tensioni tangenziali dal loro valore medio, ed E e G i moduli di elasticità normale e tangenziale del materiale. Si indicano con φ la rotazione della sezione e con u e v le componenti in direzione x ed y dello spostamento del suo baricentro. Si indicano inoltre con N, M e T le caratteristiche di sollecitazione e con

n, m e q i carichi distribuiti (assiali, flettenti e taglianti) agenti sull'asta. Il verso positivo di tutte queste grandezze è indicato nella figura.

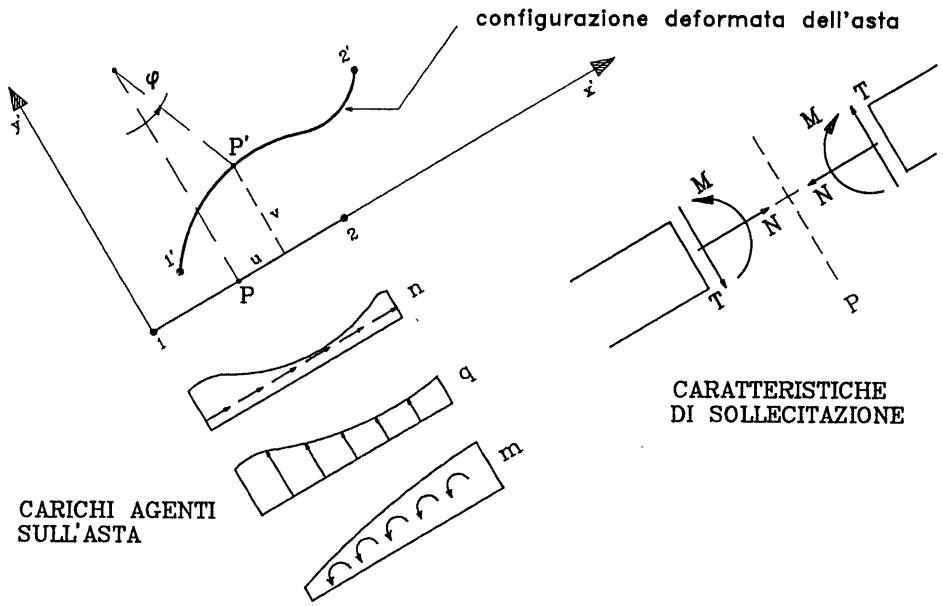


Fig. 2.9 — Carichi, caratteristiche di sollecitazione e spostamenti dei punti interni di un'asta

Per gli schemi più comuni di carico e di vincolo, le relazioni tra componenti di movimento, caratteristiche di sollecitazione e carichi sono fornite da espressioni analitiche ben note che verranno di volta in volta richiamate. Per le situazioni meno usuali, come quella di aste deformabili a taglio o di travi su suolo elastico, esse saranno invece esplicitamente ricavate a partire dalle equazioni differenziali che legano tali enti[1]

[1] In numerosi testi di Scienza delle Costruzioni l'asta è riferita ad una terna xyz nella quale l'asse z coincide con l'asse dell'asta e l'asse y è orientato verso il basso (nelle rappresentazioni piane); stesso verso positivo ha il carico tagliante q. La formulazione ivi riportata delle suddette equazioni differenziali differisce quindi per lo scambio delle variabili x e z e per il segno di alcuni termini legati alla y.

$$\frac{dy}{dx} = \frac{N}{EA} \qquad \frac{dN}{dx} = -n \tag{2.9}$$

$$\begin{aligned} \frac{dv}{dx} &= \varphi - \frac{\chi T}{GA} & \frac{d\varphi}{dx} &= \frac{M}{EI} \\ \frac{dM}{dx} &= T - m & \frac{dT}{dx} &= q \end{aligned} \tag{2.10}$$

Le equazioni (2.10) possono essere accorpate in un'unica relazione

$$\frac{d^4 v}{dx^4} = \frac{q}{EI} - \frac{1}{EI}\frac{dm}{dx} - \frac{\chi}{GA}\frac{d^2 q}{dx^2} \tag{2.11}$$

Il diagramma degli spostamenti v viene usualmente denominato linea elastica. Di conseguenza quest'ultima espressione, che mette direttamente in relazione i valori di v con i carichi, viene detta *equazione della linea elastica.*

5. Relazione tra azioni e componenti di movimento degli estremi dell'asta

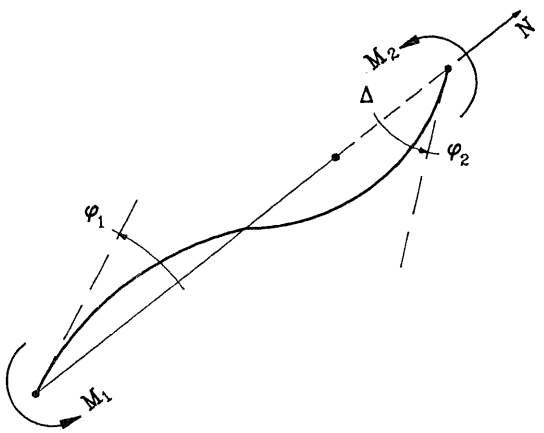
Applicando la teoria della trave ad un'asta soggetta solo a forze e coppie applicate ai suoi estremi (tali da rendere equilibrata la configurazione), è possibile determinare le caratteristiche di sollecitazione sforzo normale, momento flettente e taglio, risalire da esse alle tensioni e deformazioni in un punto qualsiasi dell'asta ed ottenere infine le componenti di movimento dei suoi estremi (a meno di un moto rigido). Viceversa, conoscendo spostamento e rotazione degli estremi dell'asta, si possono determinare successivamente le deformazioni, le tensioni, le caratteristiche di sollecitazione ed infine le azioni di estremità.

5.1. Matrici fondamentali di rigidezza e di deformabilità

Si prenda in esame un'asta, la cui sezione retta sia costante per tutta la lunghezza (fig.2.10). Si ipotizza di poter trascurare lo scorrimento della sezione dovuto alla deformazione a taglio $\left(\gamma = \frac{\chi T}{GA} = 0\right)$ e di poter applicare le azioni di estremità indifferentemente sulla configurazione indeformata o su quella deformata. La relazione tra l'insieme fondamentale

di azioni e di componenti di movimento di estremità potrebbe essere ricavata dalla risoluzione delle equazioni differenziali innanzi riportate, con le condizioni ai limiti

$$\varphi(0) = \varphi_1 \qquad \varphi(\ell) = \varphi_2$$
$$v(0) = 0 \qquad v(\ell) = 0$$
$$u(0) = 0 \qquad u(\ell) = \Delta$$



$$\mathbf{S}_f = \begin{bmatrix} M_1 \\ M_2 \\ N \end{bmatrix} \qquad \mathbf{s}_f = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \Delta \end{bmatrix}$$

Fig. 2.10 — Asta a sezione costante: insieme fondamentale di azioni e componenti di movimento

Nel caso particolare essa è però già nota (fig.2.11) e fornita dalle espressioni

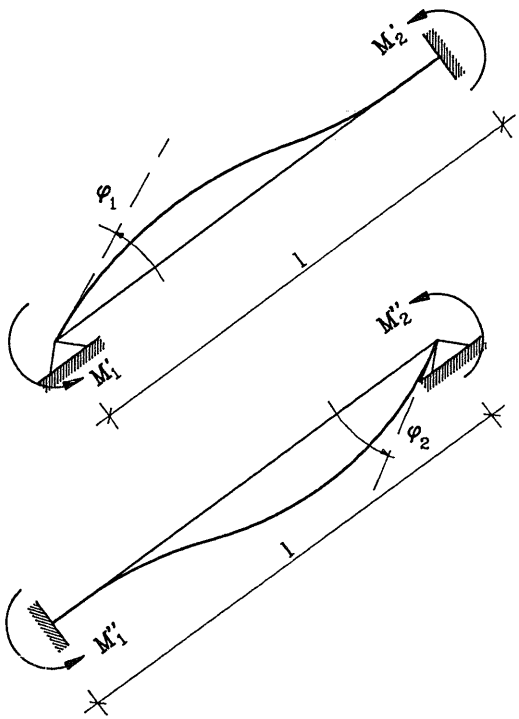
$$M_1 = 4\frac{EI}{\ell}\varphi_1 + 2\frac{EI}{\ell}\varphi_2$$

$$M_2 = 2\frac{EI}{\ell}\varphi_1 + 4\frac{EI}{\ell}\varphi_2$$

$$N = \frac{EA}{\ell}\Delta$$

che in termini matriciali possono essere sintetizzate in

$$\mathbf{S}_f = \mathbf{k}_f \, \mathbf{s}_f \tag{2.12}$$



$$\varphi_1 = \frac{M_1' \ell}{4EI} \qquad M_2' = \frac{1}{2} M_1'$$

e quindi

$$M_1' = 4\frac{EI}{\ell}\varphi_1 \qquad M_2' = 2\frac{EI}{\ell}\varphi_1$$

$$\varphi_2 = \frac{M_2'' \ell}{4EI} \qquad M_1'' = \frac{1}{2} M_2''$$

e quindi

$$M_1'' = 2\frac{EI}{\ell}\varphi_2 \qquad M_2'' = 4\frac{EI}{\ell}\varphi_2$$

Fig. 2.11 — Relazioni tra coppie flettenti e rotazioni

essendo

$$\mathbf{k}_f = \begin{bmatrix} 4\dfrac{EI}{\ell} & 2\dfrac{EI}{\ell} & 0 \\ 2\dfrac{EI}{\ell} & 4\dfrac{EI}{\ell} & 0 \\ 0 & 0 & \dfrac{EA}{\ell} \end{bmatrix} \tag{2.13}$$

La matrice $\mathbf{k}_f$ prende il nome di *matrice di rigidezza* (fondamentale) dell'asta. Il suo generico termine k_{ij} rappresenta l'azione i–esima che insorge nell'asta per effetto della componente di movimento j–esima unitaria, quando tutte le altre componenti di movimento sono impedite. La matrice è invertibile, e la sua inversa

$$\mathbf{f}_f = \mathbf{k}_f^{-1}$$

è denominata *matrice di deformabilità* (o di flessibilità) dell'asta e consente di determinare le componenti di movimento in funzione delle azioni di estremità

$$\mathbf{s}_f = \mathbf{f}_f \mathbf{S}_f$$

In particolare, nel caso in esame si ha

$$\mathbf{f}_f = \begin{bmatrix} \frac{\ell}{3EI} & -\frac{\ell}{6EI} & 0 \\ -\frac{\ell}{6EI} & \frac{\ell}{3EI} & 0 \\ 0 & 0 & \frac{\ell}{EA} \end{bmatrix}$$

5.2. Matrice di rigidezza nel sistema di riferimento locale

La relazione tra azioni e componenti di movimento nel sistema di riferimento locale può essere ottenuta da quella relativa all'insieme fondamentale utilizzando le espressioni che consentono il passaggio tra un insieme e l'altro. Si ha infatti

$$\mathbf{S}_\ell = \mathbf{T}\,\mathbf{S}_f \qquad \mathbf{S}_f = \mathbf{k}_f \mathbf{s}_f \qquad \mathbf{s}_f = \mathbf{T}^T\,\mathbf{s}_\ell$$

e quindi

$$\mathbf{S}_\ell = \mathbf{k}_\ell\,\mathbf{s}_\ell \tag{2.14}$$

con

$$\mathbf{k}_\ell = \mathbf{T}\,\mathbf{k}_f\,\mathbf{T}^T \tag{2.15}$$

Per l'asta in esame, utilizzando la definizione (2.4) della matrice di trasformazione $\mathbf{T}$ e l'espressione (2.13) di $\mathbf{k}_f$ si ottiene come matrice di rigidezza nel sistema di riferimento locale

$$\mathbf{k}_\ell = \begin{bmatrix} \frac{EA}{\ell} & 0 & 0 & -\frac{EA}{\ell} & 0 & 0 \\ 0 & 12\frac{EI}{\ell^3} & 6\frac{EI}{\ell^2} & 0 & -12\frac{EI}{\ell^3} & 6\frac{EI}{\ell^2} \\ 0 & 6\frac{EI}{\ell^2} & 4\frac{EI}{\ell} & 0 & -6\frac{EI}{\ell^2} & 2\frac{EI}{\ell} \\ -\frac{EA}{\ell} & 0 & 0 & \frac{EA}{\ell} & 0 & 0 \\ 0 & -12\frac{EI}{\ell^3} & -6\frac{EI}{\ell^2} & 0 & 12\frac{EI}{\ell^3} & -6\frac{EI}{\ell^2} \\ 0 & 6\frac{EI}{\ell^2} & 2\frac{EI}{\ell} & 0 & -6\frac{EI}{\ell^2} & 4\frac{EI}{\ell} \end{bmatrix} \tag{2.16}$$

Si noti che, a differenza di $\mathbf{k}_f$, la matrice $\mathbf{k}_\ell$ non è invertibile. Dal punto di vista numerico, ciò è dovuto al fatto che le sue righe non sono linearmente indipendenti: la prima è ottenuta cambiando segno alla quarta, la seconda sommando terza e sesta e dividendo per ℓ il risultato, e così via. Come significato fisico, non può esistere una formulazione che fornisca le componenti di spostamento in funzione di un arbitrario insieme di azioni (nel riferimento locale) perché, come già detto nel secondo paragrafo, un tale insieme arbitrario non è ammissibile ed è possibile assegnare soltanto azioni che rispettino le condizioni di equilibrio del corpo.

5.3. Matrice di rigidezza nel sistema di riferimento globale

Anche la relazione tra azioni e componenti di movimento nel sistema globale può essere ottenuta mediante le relazioni tra questi insiemi e quelli locale e fondamentale. Si ha

$$\mathbf{S}_g = \mathbf{L}_d\ \mathbf{S}_\ell \qquad \mathbf{S}_\ell = \mathbf{k}_\ell\ \mathbf{s}_\ell \qquad \mathbf{s}_\ell = \mathbf{L}_d^T\ \mathbf{s}_g$$

e quindi

$$\mathbf{S}_g = \mathbf{k}_g\ \mathbf{s}_g \tag{2.17}$$

con

$$\mathbf{k}_g = \mathbf{L}_d\ \mathbf{k}_\ell\ \mathbf{L}_d^T \tag{2.18}$$

o anche

$$\mathbf{k}_g = \mathbf{L}_d\ \mathbf{T}\ \mathbf{k}_f\ \mathbf{T}^T\ \mathbf{L}_d^T = (\mathbf{L}_d\ \mathbf{T})\ \mathbf{k}_f\ (\mathbf{L}_d\ \mathbf{T})^T \tag{2.19}$$

Anche $\mathbf{k}_g$, come $\mathbf{k}_\ell$, non è invertibile per le stesse ragioni innanzi riportate. Per l'asta in esame, l'espressione completa della matrice di rigidezza nel sistema di riferimento globale è fornita da

$$\mathbf{k}_g = \begin{bmatrix} k_1 & k_7 & -k_3 & -k_1 & -k_7 & -k_3 \\ k_7 & k_2 & k_4 & -k_7 & -k_2 & k_4 \\ -k_3 & k_4 & k_5 & k_3 & -k_4 & k_6 \\ -k_1 & -k_7 & k_3 & k_1 & k_7 & k_3 \\ -k_7 & -k_2 & -k_4 & k_7 & k_2 & -k_4 \\ -k_3 & k_4 & k_6 & k_3 & -k_4 & k_5 \end{bmatrix} \tag{2.20}$$

avendo posto

$$
\begin{aligned}
k_1 &= \frac{EA}{\ell}\cos^2\alpha + 12\frac{EI}{\ell^3}\sin^2\alpha \\
k_2 &= \frac{EA}{\ell}\sin^2\alpha + 12\frac{EI}{\ell^3}\cos^2\alpha \\
k_3 &= 6\frac{EI}{\ell^2}\sin\alpha \\
k_4 &= 6\frac{EI}{\ell^2}\cos\alpha \\
k_5 &= 4\frac{EI}{\ell} \\
k_6 &= 2\frac{EI}{\ell} \\
k_7 &= \frac{EA}{\ell}\sin\alpha\cos\alpha - 12\frac{EI}{\ell^3}\sin\alpha\cos\alpha
\end{aligned}
\tag{2.21}
$$

6. Matrice di rigidezza dell'asta: altri casi

6.1. Effetto della deformazione dovuta al taglio

L'espressione della matrice di rigidezza fondamentale della trave a sezione costante riportata nel paragrafo precedente è ricavata trascurando la deformazione a taglio dell'asta, cioè il termine $\chi\, T/GA$. Si tratta di una semplificazione molto diffusa, che comporta in genere un'approssimazione pienamente accettabile. Diventa opportuno tenerne conto solo per quegli elementi la cui sezione trasversale abbia dimensioni comparabili con la lunghezza; ciò accade in particolare, nel caso di un edificio, per pilastri con sezione molto allungata o pareti di taglio.

In questo caso, è preferibile ricavare la relazione tra l'insieme fondamentale di azioni e di componenti di movimento risolvendo le equazioni differenziali (2.9) e (2.10). Le prime, che riguardano azioni e deformazioni estensionali, non sono influenzate dalla deformazione a taglio e forniscono ancora

$$N = \frac{EA}{\ell}\Delta$$

Dalle seconde, essendo nulli q ed m, si ricava per successive integrazioni

$$T = c_1$$

$$M = c_1 x + c_2$$

$$\varphi = \frac{1}{EI}\left(c_1 \frac{x^2}{2} + c_2 x\right) + c_3$$

$$v = \frac{1}{EI}\left(c_1 \frac{x^3}{6} + c_2 \frac{x^2}{2}\right) + \left(c_3 - \frac{\chi}{GA} c_1\right) x + c_4$$

Il valore delle quattro costanti c_1, c_2, c_3, c_4 può essere determinato imponendo le condizioni al contorno

$$\varphi(0) = \varphi_1 \qquad \varphi(\ell) = \varphi_2$$

$$v(0) = 0 \qquad v(\ell) = 0$$

Da esse si ottiene infatti il sistema di quattro equazioni

$$c_3 = \varphi_1$$

$$\frac{\ell^2}{2EI} c_1 + \frac{\ell}{EI} c_2 + c_3 = \varphi_2$$

$$c_4 = 0$$

$$\left(\frac{\ell^3}{6EI} - \frac{\chi\,\ell}{GA}\right) c_1 + \frac{\ell^2}{2EI} c_2 + \ell\, c_3 + c_4 = 0$$

che fornisce il risultato

$$c_1 = 6 \frac{EI}{\ell^2} \frac{1}{1 + 4\varepsilon} (\varphi_1 + \varphi_2)$$

$$c_2 = -4 \frac{EI}{\ell} \frac{1 + \varepsilon}{1 + 4\varepsilon} \varphi_1 - 2 \frac{EI}{\ell} \frac{1 - 2\varepsilon}{1 + 4\varepsilon} \varphi_2$$

$$c_3 = \varphi_1$$

$$c_4 = 0$$

avendo posto

$$\varepsilon = \frac{3\ EI\ \chi}{GA\ \ell^2} \tag{2.22}$$

Si può così ricavare il valore delle coppie M_1 e M_2

$$M_1 = -M(0) = 4\frac{EI}{\ell}\ \frac{1+\varepsilon}{1+4\varepsilon}\ \varphi_1 + 2\frac{EI}{\ell}\ \frac{1-2\varepsilon}{1+4\varepsilon}\ \varphi_2$$

$$M_2 = \quad M(\ell) = 2\frac{EI}{\ell}\ \frac{1-2\varepsilon}{1+4\varepsilon}\ \varphi_1 + 4\frac{EI}{\ell}\ \frac{1+\varepsilon}{1+4\varepsilon}\ \varphi_2$$

La matrice di rigidezza fondamentale è quindi

$$\mathbf{k}_f = \begin{bmatrix} 4\dfrac{EI}{\ell}\dfrac{1+\varepsilon}{1+4\varepsilon} & 2\dfrac{EI}{\ell}\dfrac{1-2\varepsilon}{1+4\varepsilon} & 0 \\ 2\dfrac{EI}{\ell}\dfrac{1-2\varepsilon}{1+4\varepsilon} & 4\dfrac{EI}{\ell}\dfrac{1+\varepsilon}{1+4\varepsilon} & 0 \\ 0 & 0 & \dfrac{EA}{\ell} \end{bmatrix} \tag{2.23}$$

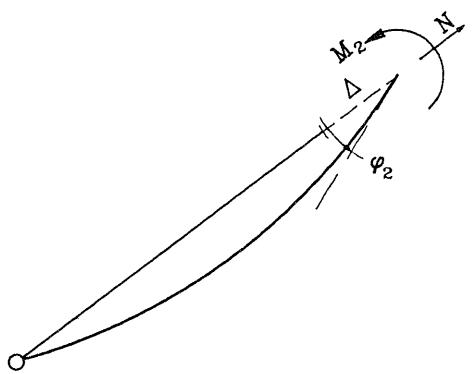
Utilizzando le (2.15) e le (2.19) si possono infine determinare anche le matrici di rigidezza rispettivamente nel sistema di riferimento locale e in quello globale.

6.2. Asta con una cerniera all'estremità

Nella quasi totalità delle strutture intelaiate in cemento armato, ed anche nella maggioranza di quelle in acciaio, le aste che convergono in un nodo sono mutuamente incastrate e presentano quindi una stessa rotazione, che coincide con la rotazione del nodo. In alcuni casi particolari di analisi elastica, o più spesso quando si vuol fare un calcolo elastoplastico step–by–step che preveda successive plasticizzazioni delle sezioni, il modello geometrico può richiedere che nel nodo una o più aste siano collegate alle altre mediante una cerniera. La risoluzione di questi schemi potrebbe essere effettuata aumentando il numero complessivo delle incognite, cioè inserendo tra queste anche le rotazioni degli estremi incernierati delle aste. È però più comodo mantenere inalterato il numero delle incognite ed esaminare

a parte il modello di asta con cerniera all'estremità, per determinarne la matrice di rigidezza.

La presenza di una cerniera ad un estremo dell'asta comporta che in tale punto il momento flettente debba essere nullo. È quindi impossibile applicarvi una coppia esterna, e le azioni fondamentali di estremità si riducono alle due mostrate nella figura 2.12. Inoltre, non è più possibile assegnare in maniera arbitraria la rotazione dell'estremo dell'asta in cui è posta la cerniera, ed anche le componenti di movimento fondamentali si riducono a due.



$$\mathbf{S}_f = \begin{bmatrix} M_2 \\ N \end{bmatrix} \qquad \mathbf{s}_f = \begin{bmatrix} \varphi_2 \\ \Delta \end{bmatrix}$$

Fig. 2.12 — Asta con una cerniera al primo estremo: insieme fondamentale di azioni e componenti di movimento

La relazione tra questi due insiemi è nota (fig.2.13) e fornita (se la cerniera è posta al primo estremo) dalle espressioni

$$M_2 = 3\frac{EI}{\ell}\varphi_2$$

$$N = \frac{EA}{\ell}\Delta$$

La matrice fondamentale di rigidezza è pertanto

$$\mathbf{k}_f = \begin{bmatrix} 3\dfrac{EI}{\ell} & 0 \\ 0 & \dfrac{EA}{\ell} \end{bmatrix} \tag{2.24}$$

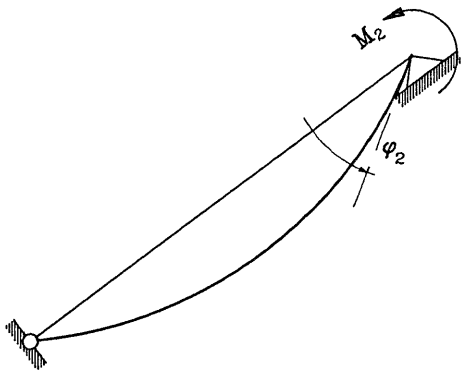
Se si vuol tener conto della deformabilità a taglio dell'asta, occorre risol-

vere le equazioni differenziali (2.10) ponendo come condizioni al contorno

$$M(0) = 0 \qquad \varphi(\ell) = \varphi_2$$
$$v(0) = 0 \qquad v(\ell) = 0$$

Si ottiene in tal modo

$$\mathbf{k}_f = \begin{bmatrix} 3\dfrac{EI}{\ell}\dfrac{1}{1+\varepsilon} & 0 \\ 0 & \dfrac{EA}{\ell} \end{bmatrix} \tag{2.25}$$



$$\varphi_2 = \frac{M_2 \ell}{3EI}$$

Fig. 2.13 — Relazione tra coppia e rotazione

Le azioni di estremità possono essere ancora descritte mediante le loro componenti nel sistema di riferimento locale, racchiuse nel vettore $\mathbf{S}_\ell$ già definito. Il passaggio da $\mathbf{S}_f$ a $\mathbf{S}_\ell$ nel caso di un'asta con cerniera al primo estremo è fornito da

$$F'_{x1} = -N \qquad F'_{x2} = N$$

$$F'_{y1} = \frac{M_2}{\ell} \qquad F'_{y2} = -\frac{M_2}{\ell}$$

$$M'_{z1} = 0 \qquad M'_{z2} = M_2$$

In termini matriciali vale quindi sempre la relazione (2.3)

$$\mathbf{S}_\ell = \mathbf{T}\,\mathbf{S}_f$$

e la matrice di trasformazione è in questo caso

$$\mathbf{T} = \begin{bmatrix} 0 & -1 \\ \frac{1}{\ell} & 0 \\ 0 & 0 \\ 0 & 1 \\ -\frac{1}{\ell} & 0 \\ 1 & 0 \end{bmatrix} \tag{2.26}$$

Si noti che questa matrice è ottenuta da quella relativa all'asta priva di cerniere (2.4) eliminando la prima colonna, cioè quella corrispondente alla coppia M_1 non più presente nell'insieme fondamentale. Analogamente, quando la cerniera è situata nel secondo estremo dell'asta la matrice $\mathbf{T}$ è ricavata eliminando la seconda colonna.

La matrice di trasformazione così definita consente il passaggio duale dall'insieme di componenti di movimento nel riferimento locale a quello fondamentale con la relazione (2.7)

$$\mathbf{s}_f = \mathbf{T}^T \, \mathbf{s}_\ell$$

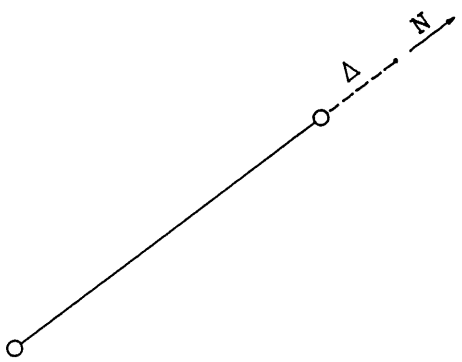
e la determinazione delle matrici di rigidezza nel riferimento locale e globale mediante le (2.15) e (2.19).

6.3. Asta incernierata ai due estremi (pendolo)

Il modello di asta pendolare, cioè con cerniere all'estremità, non è quasi mai utilizzato nel caso di strutture in cemento armato. L'unica sua possibile applicazione si ritrova nello schematizzare eventuali pannelli murari collaboranti con la struttura vera e propria. Esso è invece molto utilizzato per le travature reticolari in acciaio, nelle quali anche in presenza di nodi incastro le sollecitazioni estensionali sono nettamente prevalenti su quelle flessionali.

Analogamente a quanto indicato per il caso precedente, se in una struttura sono presenti alcune aste incernierate all'estremità è preferibile non aumentare il numero delle incognite ed esaminare esplicitamente il modello

di asta pendolo. Essendo impossibile applicare ad essa coppie di estremità, le azioni fondamentali si riducono alla sola forza assiale N e le componenti di movimento alla variazione di lunghezza Δ (fig.2.14).



$$\mathbf{S}_f = \left[N \right]$$

$$\mathbf{s}_f = \left[\Delta \right]$$

Fig. 2.14 — Asta incernierata agli estremi (pendolo): insieme fondamentale di azioni e componenti di movimento

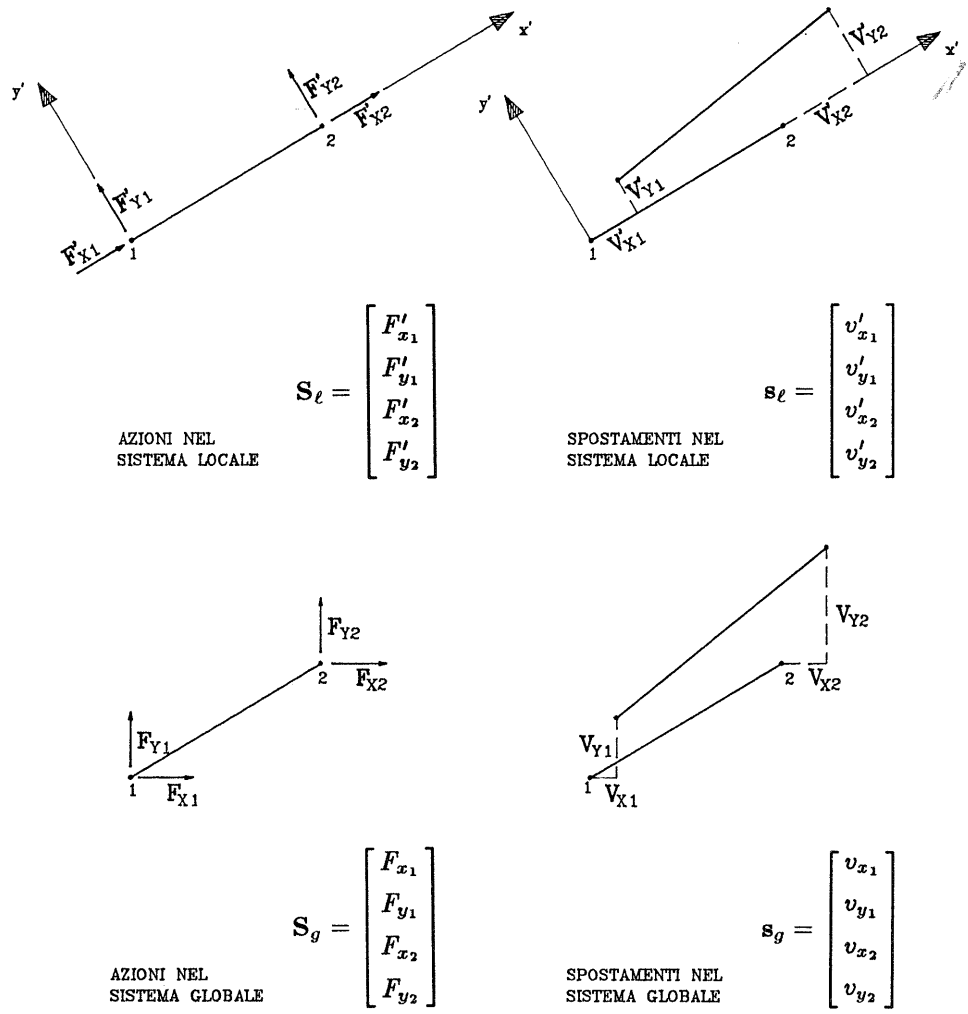
La matrice di rigidezza fondamentale è quindi

$$\mathbf{k}_f = \left[\frac{EA}{\ell} \right] \tag{2.27}$$

Le azioni di estremità possono essere anche in questo caso descritte mediante le componenti nel riferimento locale $\mathbf{S}_\ell$, che sono ancora legate ad $\mathbf{S}_f$ dalla relazione (2.3), nella quale è ora

$$\mathbf{T} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.28}$$

cioè coincide con la terza colonna della matrice di trasformazione dell'asta priva di cerniere.



$$\mathbf{S}_\ell = \begin{bmatrix} F'_{x_1} \\ F'_{y_1} \\ F'_{x_2} \\ F'_{y_2} \end{bmatrix} \qquad \mathbf{s}_\ell = \begin{bmatrix} v'_{x_1} \\ v'_{y_1} \\ v'_{x_2} \\ v'_{y_2} \end{bmatrix}$$

$$\mathbf{S}_g = \begin{bmatrix} F_{x_1} \\ F_{y_1} \\ F_{x_2} \\ F_{y_2} \end{bmatrix} \qquad \mathbf{s}_g = \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ v_{x_2} \\ v_{y_2} \end{bmatrix}$$

Fig. 2.15 — Azioni e componenti di movimento nei sistemi di riferimento locale e globale per insiemi di aste pendolari

Le matrici di rigidezza nel riferimento locale e globale vengono ancora ottenute mediante le (2.15) e (2.19). È facile constatare che esse coincidono con quelle che si avrebbero per la stessa asta, priva di cerniere, se si considera nullo il momento d'inerzia della sua sezione.

Una considerazione a parte richiede il caso di strutture costituite da un insieme di aste tutte pendolari, come ad esempio le travature reticolari.

L'impostazione generale, che considera tre azioni e tre componenti di movimento per ciascun estremo delle aste è ancora valida. Occorre però notare che in ogni nodo convergono solo aste incernierate, che non ne impediscono la rotazione. Questa deve pertanto essere impedita imponendo a ciascun nodo un vincolo esterno.

Un'alternativa molto diffusa consiste nell'eliminare direttamente dall'insieme di azioni locali e globali le coppie di estremità e dall'insieme di componenti di movimento le rotazioni (fig.2.15). In tal caso la matrice di trasformazione è

$$\mathbf{T} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{2.29}$$

e quella di rotazione

$$\mathbf{L}_d = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & -\sin\alpha \\ 0 & 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{2.30}$$

6.4. Trave su suolo elastico (alla Winkler)

Col termine trave su suolo elastico si indica, come noto, un'asta collegata in maniera continua ad un mezzo elastico che esplica su essa, nel caso di contatto liscio, azioni ortogonali all'asse, dipendenti dalle componenti di spostamento in tale direzione. In particolare è molto utilizzato il modello di Winkler, che schematizza il suolo come un insieme di molle indipendenti che esercitano in ciascun punto dell'asta un'azione legata esclusivamente allo spostamento del punto e proporzionale ad esso. Il suolo elastico è in tal caso definito in maniera completa mediante un unico parametro c, detto costante di sottofondo, che rappresenta il rapporto tra la pressione mutuamente esercitata tra trave e letto di molle ed il corrispondente spostamento.

Il modello di trave su suolo alla Winkler è stato il primo, e per lungo tempo l'unico, utilizzato per il calcolo di travi di fondazione. I notevoli

sviluppi teorici avutisi nel campo della geotecnica, uniti alla possibilità di ricorrere agli elaboratori elettronici per gli sviluppi numerici, hanno reso disponibili altri modelli, da quello di semispazio elastico o di strato elastico di dimensioni finite ai più complessi modelli elastoplastici. A differenza di quello di Winkler, tutti questi modelli prevedono un effetto mutuo tra punti distinti della trave, perché allo spostamento di un punto della superficie di contatto tra trave e suolo concorrono non solo i carichi ad esso direttamente applicati, ma anche quelli agenti su altre aree del piano.

La disponibilità di tanti nuovi modelli non ha comunque annullato la possibilità d'uso di quello di Winkler. Esso fornisce infatti risultati attendibili nel caso di travi di fondazione relativamente flessibili, ma anche per travi relativamente rigide poste su banchi di terreno di spessore ridotto. Inoltre, esso consente di schematizzare adeguatamente anche altri elementi strutturali diversi dalle fondazioni. Si pensi, ad esempio, al caso del travetto di ripartizione di un solaio, per il quale il letto di molle è costituito dai travetti stessi del solaio che esercitano effettivamente un'azione proporzionale al loro abbassamento; oppure al caso di serbatoi o silos cilindrici, nei quali il vincolo elastico è fornito dalle fibre anulari che si oppongono agli spostamenti delle fibre longitudinali mediante la loro rigidezza estensionale.

Ai fini dell'approccio matriciale, nell'analisi di strutture discretizzate mediante elementi finiti, il modello di Winkler presenta infine un grande vantaggio. Un punto fondamentale di tale approccio è il fatto che sollecitazioni e deformazioni di un'asta dipendono esclusivamente dalle componenti di movimento e dalle azioni dei punti di estremità e dai carichi su essa applicati. Considerare il suolo elastico come costituito da molle tra loro indipendenti rende possibile correlare direttamente l'azione del suolo alla deformazione della trave, ottenendo quindi, come verrà tra poco mostrato, una matrice di rigidezza della trave su suolo alla Winkler, cioè dell'intero complesso trave–suolo. Negli altri modelli citati, l'azione del suolo dipende invece anche dalla deformazione delle altre aste poste a contatto con esso. Occorre quindi considerare le aste separatamente dal suolo ed anzi suddividerle in un numero elevato di elementi per imporre la congruenza asta–suolo in un numero adeguato di punti, aumentando quindi notevolmente il numero di incognite e la complessità del problema.

Nell'esaminare il complesso costituito da una trave e dal suolo elastico alla Winkler, occorre riesaminare brevemente quanto scritto a proposito di azioni e componenti di movimento di estremità. A differenza di quanto accade per un'asta libera nel piano, nel caso in esame è possibile assegnare

valori completamente arbitrari alle quattro azioni di estremità F'_{y1}, M'_{z1}, F'_{y2}, M'_{z2}. L'equilibrio dell'asta è infatti garantito dalle azioni esercitate sulla trave deformata dal terreno ad essa solidale. È invece possibile assegnare solo una azione assiale, perché le molle elastiche non esplicano alcuna forza in tale direzione. L'insieme fondamentale di azioni è quindi costituito dalle cinque componenti mostrate col loro segno positivo nella figura 2.16. In maniera duale, lo stato deformativo dell'asta è influenzato da rotazioni e spostamenti degli estremi ortogonali all'asse, ma non da un moto rigido in direzione dell'asse. Le cinque componenti di movimento fondamentali sono mostrate nella figura 2.17.

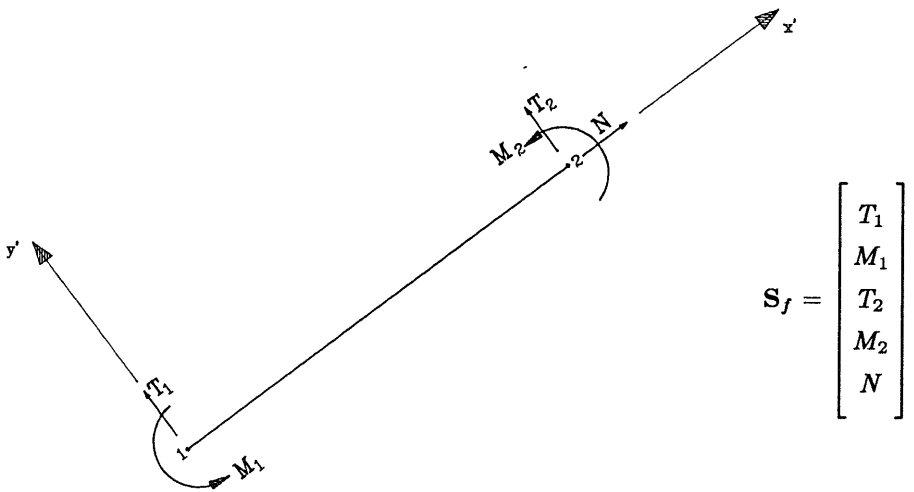


Fig. 2.16 — Trave su suolo elastico alla Winkler: insieme fondamentale di azioni

La relazione tra l'insieme fondamentale di azioni e di componenti di movimento deve essere ricavata dalle equazioni differenziali (2.9) e (2.10). Dalle prime si ottiene come sempre

$$N = \frac{EA}{\ell}\Delta$$

Nelle seconde, in assenza di carichi esterni il carico per unità di lunghezza q è dovuto esclusivamente al suolo; esso è proporzionale allo spostamento v ed ha verso opposto ad esso, e vale quindi

$$q = -c\ b\ v$$

avendo indicato con b la larghezza della trave a contatto col suolo.

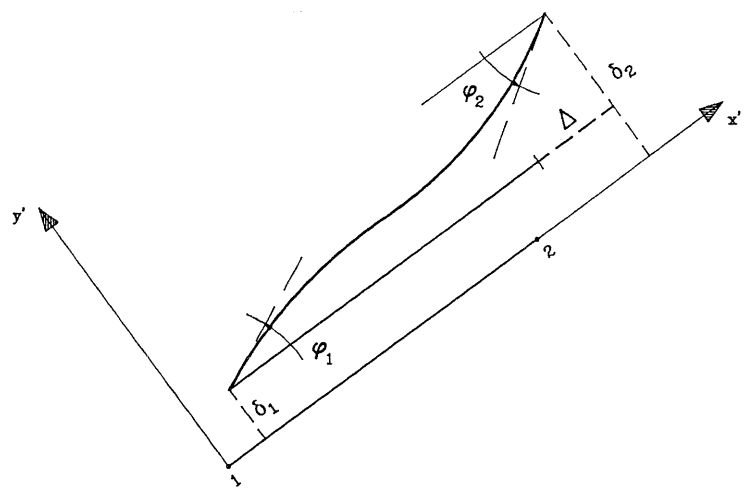


Fig. 2.17 — Trave su suolo elastico alla Winkler:
insieme fondamentale di componenti di movimento

L'equazione (2.11) della linea elastica, che sintetizza le (2.10), diventa pertanto, trascurando la deformazione da taglio

$$\frac{d^4 v}{dx^4} + \frac{cb}{EI} v = 0$$

ovvero

$$\frac{d^4 v}{dx^4} + 4\lambda^4 v = 0$$

avendo posto

$$\lambda = \sqrt[4]{\frac{cb}{4EI}} \tag{2.31}$$

Una soluzione generale di questa equazione differenziale può scriversi nella forma

$$v(x) = c_1 e^{\lambda x} \sin \lambda x + c_2 e^{\lambda x} \cos \lambda x + c_3 e^{-\lambda x} \sin \lambda x + c_4 e^{-\lambda x} \cos \lambda x$$

Da essa si ricavano le espressioni della rotazione e delle caratteristiche

di sollecitazione

$$\varphi(x) = \frac{dv}{dx} = c_1\lambda e^{\lambda x}(\sin\lambda x + \cos\lambda x) +$$
$$+ c_2\lambda e^{\lambda x}(-\sin\lambda x + \cos\lambda x) +$$
$$+ c_3\lambda e^{-\lambda x}(-\sin\lambda x + \cos\lambda x) +$$
$$+ c_4\lambda e^{-\lambda x}(-\sin\lambda x - \cos\lambda x)$$

$$M(x) = EI\frac{d\varphi}{dx} = 2EI\lambda^2\Big[c_1 e^{\lambda x}\cos\lambda x - c_2 e^{\lambda x}\sin\lambda x + \tag{2.32}$$
$$- c_3 e^{-\lambda x}\cos\lambda x + c_4 e^{-\lambda x}\sin\lambda x\Big]$$

$$T(x) = \frac{dM}{dx} = 2EI\lambda^2\Big[c_1\lambda e^{\lambda x}(-\sin\lambda x + \cos\lambda x) +$$
$$+ c_2\lambda e^{\lambda x}(-\sin\lambda x - \cos\lambda x) +$$
$$+ c_3\lambda e^{-\lambda x}(\sin\lambda x + \cos\lambda x) +$$
$$+ c_4\lambda e^{-\lambda x}(-\sin\lambda x + \cos\lambda x)\Big]$$

Il valore delle quattro costanti c_1, c_2, c_3, c_4 può essere determinato imponendo le condizioni al contorno

$$v(0) = \delta_1 \qquad v(\ell) = \delta_2$$
$$\varphi(0) = \varphi_1 \qquad \varphi(\ell) = \varphi_2$$

Da esse, ponendo per chiarezza di scrittura

$$s' = e^{\lambda\ell}\sin\lambda\ell \qquad s'' = e^{-\lambda\ell}\sin\lambda\ell$$
$$c' = e^{\lambda\ell}\cos\lambda\ell \qquad c'' = e^{-\lambda\ell}\cos\lambda\ell$$

si ottiene infatti il sistema di quattro equazioni

$$c_2 + c_4 = \delta_1$$
$$\lambda c_1 + \lambda c_2 + \lambda c_3 - \lambda c_4 = \varphi_1$$
$$s'c_1 + c'c_2 + s''c_3 + c''c_4 = \delta_2$$
$$\lambda(s' + c')c_1 + \lambda(-s' + c')c_2 + \lambda(-s'' + c'')c_3 + \lambda(-s'' - c'')c_4 = \varphi_2$$

che in termini matriciali può essere scritto

$$\mathbf{A}\ \mathbf{C} = \mathbf{s}'_f$$

essendo $\mathbf{A}$ la matrice dei coefficienti, $\mathbf{C}$ l'insieme di costanti incognite ed $\mathbf{s}'_f$ il sottoinsieme di $\mathbf{s}_f$ racchiudente spostamenti ortogonali all'asse e rotazioni

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ \lambda & \lambda & \lambda & -\lambda \\ s' & c' & s'' & c'' \\ \lambda(s'+c') & \lambda(-s'+c') & \lambda(-s''+c'') & \lambda(-s''-c'') \end{bmatrix} \tag{2.33}$$

$$\mathbf{C} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \qquad \mathbf{s}'_f = \begin{bmatrix} \delta_1 \\ \varphi_1 \\ \delta_2 \\ \varphi_2 \end{bmatrix}$$

La risoluzione del sistema per via analitica presenta una certa complessità e non è idonea a fornire formulazioni di immediata applicabilità. La soluzione numerica può però essere facilmente ottenuta mediante l'inversione della matrice $\mathbf{A}$

$$\mathbf{C} = \mathbf{A}^{-1}\ \mathbf{s}'_f \tag{2.34}$$

Si può così ricavare il valore delle azioni T_1, M_1, T_2, M_2

$$\begin{aligned} T_1 = \ \ T(0) \ &= 2EI\lambda^2\left[\lambda c_1 - \lambda c_2 + \lambda c_3 + \lambda c_4\right] \\ M_1 = -M(0) \ &= 2EI\lambda^2\left[-c_1 + c_3\right] \\ T_2 = -T(\ell) \ &= 2EI\lambda^2\left[\lambda(s'-c')c_1 + \lambda(s'+c')c_2 + \right. \\ &\qquad \left. + \lambda(-s''-c'')c_3 + \lambda(s''-c'')c_4\right] \\ M_2 = \ \ M(\ell) \ &= 2EI\lambda^2\left[c'c_1 - s'c_2 - c''c_3 + s''c_4\right] \end{aligned}$$

Indicando con $\mathbf{S}'_f$ il sottoinsieme di $\mathbf{S}_f$ contenente le azioni innanzi elencate, si può scrivere in termini matriciali

$$\mathbf{S}'_f = \mathbf{B}\ \mathbf{C} \tag{2.35}$$

con

$$\mathbf{B} = 2EI\lambda^2 \begin{bmatrix} \lambda & -\lambda & \lambda & \lambda \\ -1 & 0 & 1 & 0 \\ \lambda(s'-c') & \lambda(s'+c') & \lambda(-s''-c'') & \lambda(s''-c'') \\ c' & -s' & -c'' & s'' \end{bmatrix} \tag{2.36}$$

Dalle (2.34) e (2.35) si ricava la relazione tra i due sottoinsiemi di azioni e componenti di movimento

$$\mathbf{S}'_f = \mathbf{k}'_f \, \mathbf{s}'_f$$

avendo posto

$$\mathbf{k}'_f = \mathbf{B} \, \mathbf{A}^{-1} \tag{2.37}$$

La matrice $\mathbf{k}'_f$ così ottenuta è costituita da 4 righe e 4 colonne ed è una sottomatrice della matrice di rigidezza fondamentale $\mathbf{k}_f$ della trave su suolo elastico

$$\mathbf{k}_f = \begin{bmatrix} & & & & 0 \\ & \mathbf{k}'_f & & & 0 \\ & & & & 0 \\ & & & & 0 \\ 0 & 0 & 0 & 0 & \dfrac{EA}{\ell} \end{bmatrix} \tag{2.38}$$

Azioni e componenti di movimento nel riferimento locale possono essere ottenuti dagli insiemi fondamentali mediante le relazioni (2.3) e (2.7). Si può con facilità controllare che nel caso in esame la matrice di trasformazione $\mathbf{T}$ è espressa mediante

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.39}$$

La matrice di rigidezza nel sistema di riferimento locale può ancora essere valutata mediante la (2.15). Se si immagina di dividere la matrice $\mathbf{k}'_f$ in 4 sottomatrici di 2 righe e 2 colonne,

$$\mathbf{k}'_f = \begin{bmatrix} \mathbf{k}'_{f_{11}} & \mathbf{k}'_{f_{12}} \\ \mathbf{k}'_{f_{21}} & \mathbf{k}'_{f_{22}} \end{bmatrix} \tag{2.40}$$

si può constatare che la (2.15) in questo caso provoca come unico effetto il duplicamento dei termini EA/ℓ ed un posizionamento diverso delle 4 sottomatrici ora definite. Si ha cioè

$$\mathbf{k}_\ell = \begin{bmatrix} \frac{EA}{\ell} & 0 & 0 & -\frac{EA}{\ell} & 0 & 0 \\ 0 & & & 0 & & \\ & \mathbf{k}'_{f_{11}} & & & \mathbf{k}'_{f_{12}} & \\ 0 & & & 0 & & \\ -\frac{EA}{\ell} & 0 & 0 & \frac{EA}{\ell} & 0 & 0 \\ 0 & & & 0 & & \\ & \mathbf{k}'_{f_{21}} & & & \mathbf{k}'_{f_{22}} & \\ 0 & & & 0 & & \end{bmatrix} \tag{2.41}$$

La matrice di rigidezza nel sistema globale si può infine ottenere, a partire da quella nel riferimento locale, mediante la (2.18).

6.5. Asta con tratti rigidi all'estremità

Il modello di telaio costituito da aste monodimensionali convergenti in punti nodali approssima correttamente la realtà quando le aste presentano effettivamente una sezione trasversale molto piccola rispetto alla loro lunghezza. Ciò non si verifica nel caso di telai–pareti (walled frames) o di pareti di taglio (shear walls), molto comuni per edifici alti. In tali situazioni la zona nodale non è più puntiforme ed in essa non è valido il modello di trave alla De Saint Venant.

Uno schema limite frequentemente utilizzato, valido quando gli assi delle aste convergono in un unico punto, considera presente a ciascun estremo di esse un tratto infinitamente rigido, corrispondente alla parte di asta contenuta nel nodo. Quando invece gli assi delle aste non convergono in un unico punto è più corretto, anche se meno utilizzato, il modello di nodo rigido non puntiforme, esaminato nel paragrafo successivo.

Si consideri un'asta di lunghezza ℓ, con tratti rigidi di estremità aventi lunghezza $s\,\ell$ e $t\,\ell$ (fig.2.18). Per differenziare le grandezze relative al tratto centrale, deformabile, da analoghe entità relative all'intera asta si indicano nel seguito le prime con un doppio apice. In particolare, si indica con ℓ'' la lunghezza di tale tratto. Essa vale

$$\ell'' = (1 - s - t)\ell = \frac{\ell}{c} \tag{2.42}$$

avendo posto

$$c = \frac{1}{1 - s - t} \tag{2.43}$$

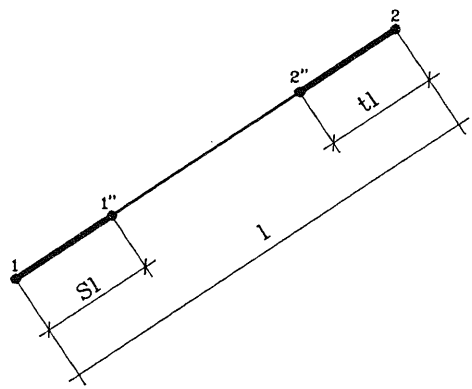


Fig. 2.18 — Asta con tratti rigidi di estremità

In figura 2.19 è mostrato l'insieme fondamentale di azioni $\mathbf{S}_f$ e l'analogo insieme di azioni $\mathbf{S}''_f$ mutuamente trasmesse agli estremi del tratto deformabile. Ricordando la relazione tra taglio e momento nel tratto deformabile

$$T''_1 = T''_2 = \frac{M''_1 + M''_2}{\ell''}$$

per l'equilibrio del tratto rigido si ha

$$M_1 = M_1'' + T_1''\ s\ \ell = (1 + c\ s)M_1'' + c\ s\ M_2''$$
$$M_2 = M_2'' + T_2''\ t\ \ell = c\ t\ M_1'' + (1 + c\ t)\ M_2''$$
$$N\ = N''$$

In termini matriciali queste relazioni sono espresse da

$$\mathbf{S}_f = \mathbf{C}\ \mathbf{S}_f'' \tag{2.44}$$

indicando con $\mathbf{C}$ la matrice che consente il passaggio tra i due insiemi

$$\mathbf{C} = \begin{bmatrix} 1 + c\ s & c\ s & 0 \\ c\ t & 1 + c\ t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.45}$$

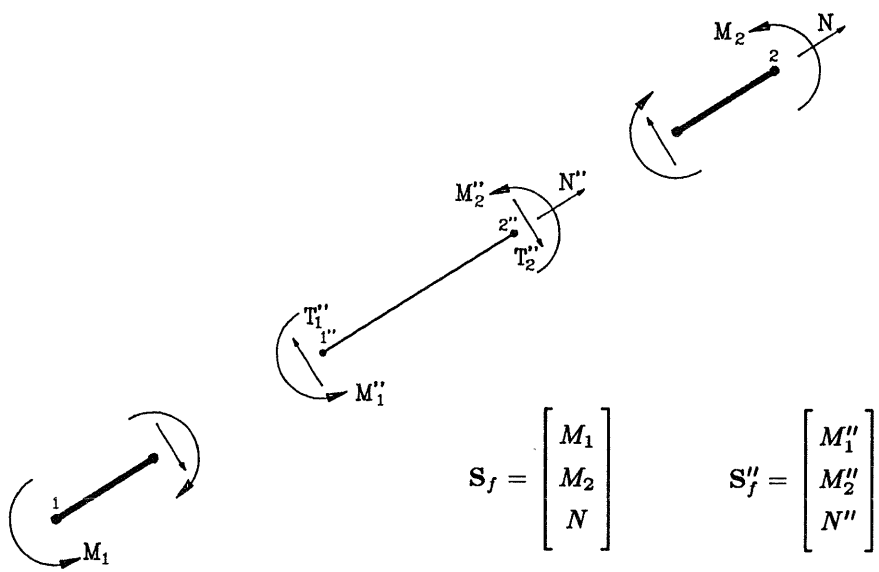
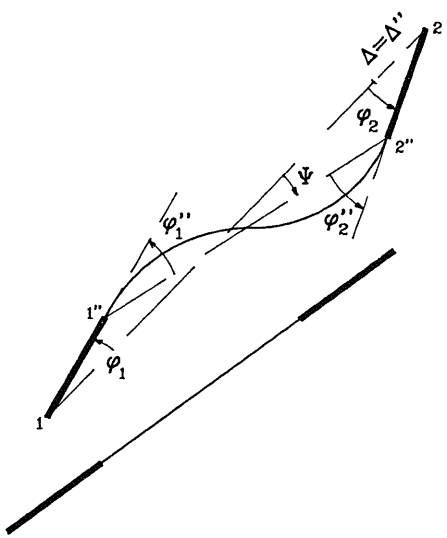


Fig. 2.19 — Azioni fondamentali all'estremità dell'asta e del tratto deformabile

Nella figura 2.20 sono mostrati l'insieme fondamentale $\mathbf{s}_f$ delle componenti di movimento degli estremi dell'asta e l'insieme $\mathbf{s}''_f$ che racchiude i corrispondenti movimenti degli estremi del tratto deformabile.



$$\mathbf{s}_f = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \Delta \end{bmatrix}$$

$$\mathbf{s}''_f = \begin{bmatrix} \varphi''_1 \\ \varphi''_2 \\ \Delta'' \end{bmatrix}$$

Fig. 2.20 — Componenti di movimento fondamentale degli estremi dell'asta e del tratto deformabile

Essendo i tratti di estremità infinitamente rigidi, i punti 1'' e 2'' subiscono uno spostamento in direzione ortogonale all'asse rispettivamente pari a $\varphi_1 s\ \ell$ e $\varphi_2 t\ \ell$. L'asse del tratto deformabile è quindi ruotato, rispetto all'asse dell'asta, dell'angolo ψ

$$\psi = c\ s\ \varphi_1 + c\ t\ \varphi_2$$

Le componenti di movimento degli estremi del tratto deformabile valgono pertanto

$$\varphi''_1 = \varphi_1 + \psi = (1 + c\ s)\ \varphi_1 + c\ t\ \varphi_2$$
$$\varphi''_2 = \varphi_2 + \psi = c\ s\ \varphi_1 + (1 + c\ t)\ \varphi_2$$
$$\Delta'' = \Delta$$

In termini matriciali, queste relazioni sono espresse da

$$\mathbf{s}''_f = \mathbf{C}^T\ \mathbf{s}_f \qquad (2.46)$$

mediante la trasposta della matrice C già definita.

Il tratto centrale, deformabile, è a tutti gli effetti una trave alla De Saint Venant. Per esso quindi la relazione tra azioni e componenti di movimento fondamentali di estremità è espressa mediante la (2.12), che in questo caso si può scrivere

$$\mathbf{S}''_f = \mathbf{k}''_f \, \mathbf{s}''_f \tag{2.47}$$

Utilizzando le relazioni (2.44), (2.46) e (2.47) si ottiene

$$\mathbf{S}_f = \mathbf{C} \, \mathbf{k}''_f \, \mathbf{C}^T \, \mathbf{s}_f$$

e quindi l'espressione della matrice di rigidezza per l'asta con tratti rigidi di estremità

$$\mathbf{k}_f = \mathbf{C} \, \mathbf{k}''_f \, \mathbf{C}^T \tag{2.48}$$

Se per il tratto centrale si trascura l'effetto della deformazione dovuta al taglio, la formulazione generale qui riportata può essere esplicitata; si ha infatti

$$\mathbf{k}''_f = \begin{bmatrix} 4\dfrac{EI}{\ell''} & 2\dfrac{EI}{\ell''} & 0 \\ 2\dfrac{EI}{\ell''} & 4\dfrac{EI}{\ell''} & 0 \\ 0 & 0 & \dfrac{EA}{\ell''} \end{bmatrix}$$

e si ottiene quindi

$$\mathbf{k}_f = \begin{bmatrix} 4\dfrac{EI}{\ell}a_1 & 2\dfrac{EI}{\ell}b & 0 \\ 2\dfrac{EI}{\ell}b & 4\dfrac{EI}{\ell}a_2 & 0 \\ 0 & 0 & \dfrac{EA}{\ell}c \end{bmatrix} \tag{2.49}$$

con

$$\begin{aligned} a_1 &= c\left[(1-t)^2 + s(1-t) + s^2\right] \\ a_2 &= c\left[(1-s)^2 + t(1-s) + t^2\right] \\ b &= c\left[1 + s + t - 2(s^2 - st + t^2)\right] \end{aligned} \tag{2.50}$$

Nelle applicazioni concrete, la presenza di tratti rigidi è quasi sempre accoppiata ad aste con sezione trasversale non trascurabile rispetto alla lunghezza. In tal caso è più corretto tener conto della deformabilità

tagliante ed utilizzare per la matrice di rigidezza del tratto centrale l'espressione (2.23). La sostituzione diretta di essa nell'espressione (2.48) non porta a formulazioni di agevole scrittura ed è quindi preferibile operare effettuando per via numerica il prodotto matriciale.

Come per tutti gli altri casi esposti, le matrici di rigidezza nel sistema di riferimento locale e globale possono essere ricavate da quella fondamentale utilizzando le (2.15) e (2.19).

6.6. Nodo rigido non puntiforme (asta con tratti rigidi non coassiali)

Nelle applicazioni professionali, in particolare quando si progettano edifici in cemento armato, accade frequentemente che gli assi delle aste convergenti in un nodo non si incontrino in un unico punto. Si pensi ad esempio alla presenza contemporanea di travi emergenti e a spessore, oppure a pilastri che risegano in maniera non simmetrica. Quando l'eccentricità degli assi assume entità rilevante, non più trascurabile, occorre superare il concetto di nodo puntiforme e riguardare la zona nodale come una zona indeformabile di dimensioni finite. Non esistono criteri assoluti per determinare l'ampiezza di essa; un minimo, costituito dall'area delimitata dagli assi eccentrici (fig.2.21a), può essere adottato quando si vuole tenere conto esclusivamente dell'effetto del disassamento; un valore maggiore (fig.2.21b) può essere giustificato dal desiderio di valutare anche l'irrigidimento costituito dalla sovrapposizione delle aste.

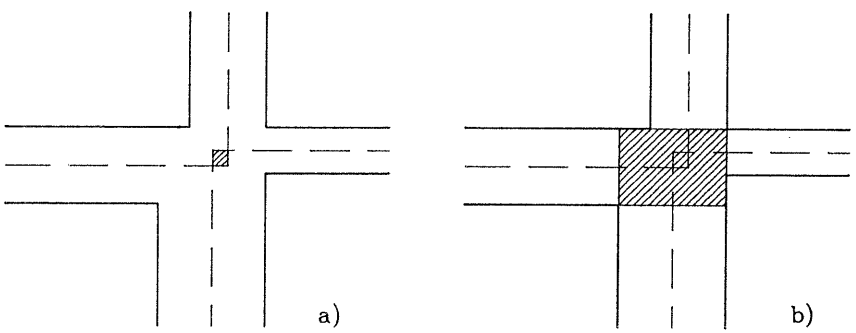


Fig. 2.21 — Nodo rigido non puntiforme
a) dimensioni minime
b) dimensioni massime

Il modo più semplice per tener conto di nodi rigidi non puntiformi con-

siste nell'assumere come nodo un punto (arbitrario) dell'area rigida e considerare che ciascuna asta in esso confluente sia dotata all'estremità di un tratto rigido che va dal punto nodale all'intersezione tra area rigida e asse dell'asta (fig.2.22). Si effettua in tal modo una generalizzazione del caso, già descritto, di asta con tratti rigidi all'estremità. La differenza sostanziale consiste nel fatto che ora l'asse del tratto deformabile non è più coincidente con la congiungente gli estremi dell'asta.

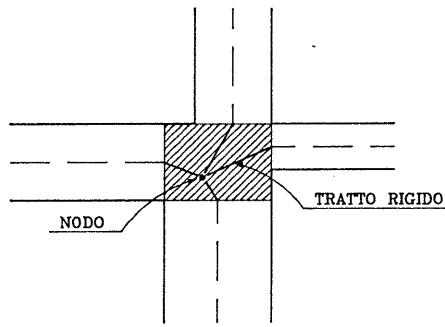


Fig. 2.22 — Schematizzazione del nodo e delle aste

Si consideri un'asta dotata ad entrambe le estremità di tratti rigidi non coassiali (fig.2.23).

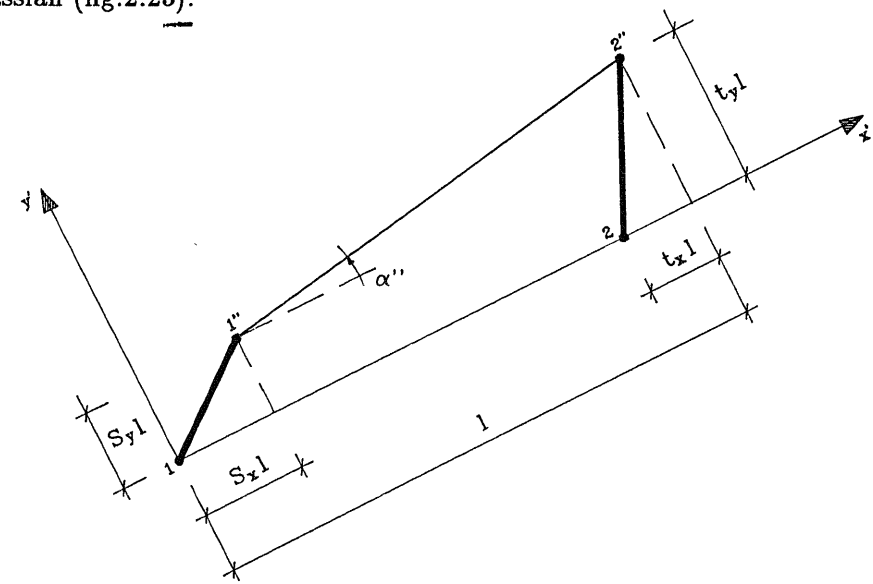


Fig. 2.23 — Asta con tratti rigidi non coassiali

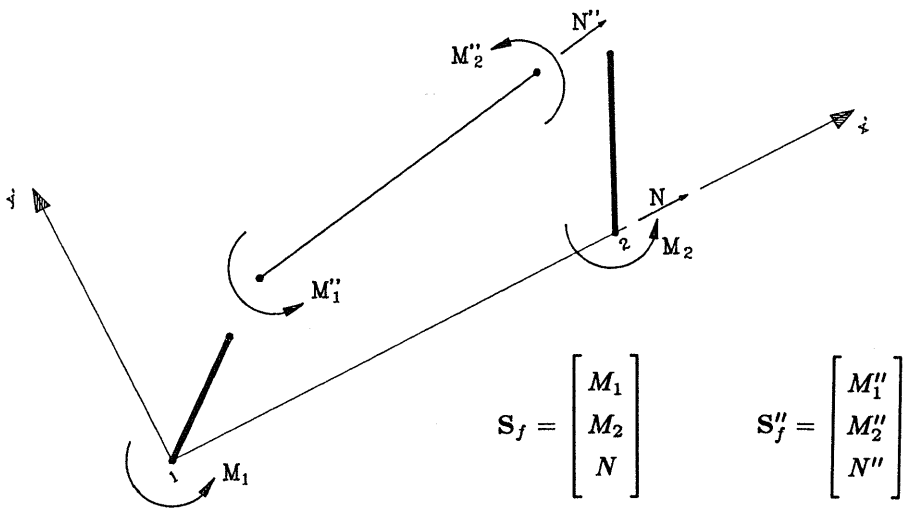


Fig. 2.24 — Azioni fondamentali all'estremità dell'asta e del tratto deformabile

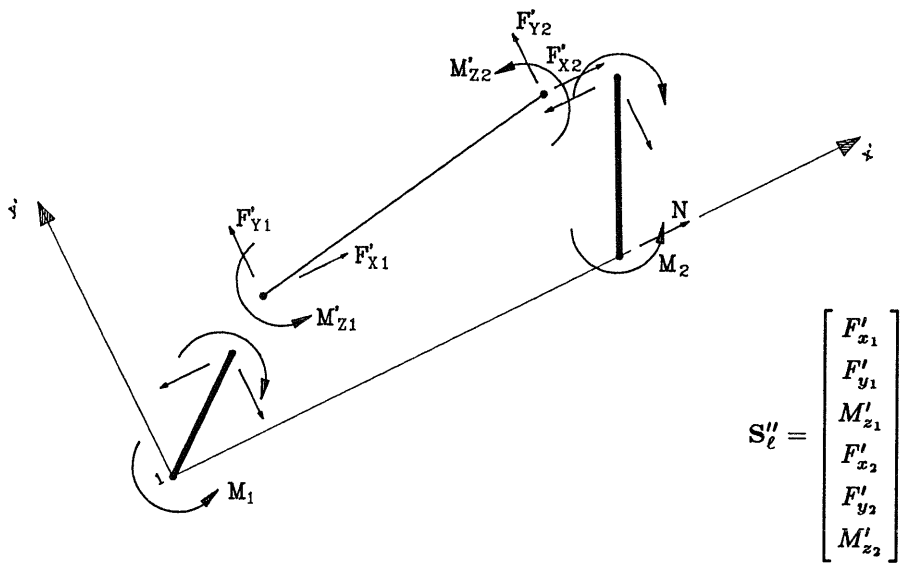


Fig. 2.25 — Componenti rispetto al sistema locale delle azioni trasmesse agli estremi del tratto deformabile

Si indica con ℓ la distanza tra i nodi (1-2) ed ℓ'' la lunghezza del tratto deformabile (1''-2''). Si indica inoltre con $s_x\ \ell$ ed $s_y\ \ell$ le componenti del tratto rigido del primo estremo rispetto agli assi locali $x'y'$, $t_x\ \ell$ e $t_y\ \ell$ le analoghe componenti del tratto rigido del secondo estremo, considerate positive se di verso concorde con tali assi. Con queste posizioni, la lunghezza del tratto deformabile vale

$$\ell'' = \ell\sqrt{(1 + t_x - s_x)^2 + (t_y - s_y)^2} = \frac{\ell}{c} \tag{2.51}$$

avendo posto

$$c = \frac{1}{\sqrt{(1 + t_x - s_x)^2 + (t_y - s_y)^2}} \tag{2.52}$$

Il tratto deformabile è inclinato, rispetto alla congiungente i nodi di estremità, di un angolo α'', fornito da

$$\tan\alpha'' = \frac{t_y - s_y}{1 + t_x - s_x} \tag{2.53}$$

Nella figura 2.24 è mostrato l'insieme fondamentale di azioni $\mathbf{S}_f$ e l'analogo insieme di azioni $\mathbf{S}''_f$ mutuamente trasmesso agli estremi del tratto deformabile.

Nella figura 2.25 è mostrato l'insieme $\mathbf{S}''_\ell$ equivalente ad $\mathbf{S}''_f$, che contiene le componenti di tali azioni rispetto al riferimento locale $x'y'$. $\mathbf{S}''_\ell$ può essere ottenuta da $\mathbf{S}''_f$ mediante una matrice di trasformazione $\mathbf{T}''$ ed una di rotazione $\mathbf{L}''_d$, in maniera analoga a quanto mostrato nel paragrafo 2

$$\mathbf{S}''_\ell = \mathbf{L}''_d\mathbf{T}''\mathbf{S}''_f \tag{2.54}$$

Le matrici $\mathbf{L}''_d$ e $\mathbf{T}''$ sono ottenute dalle (2.2) e (2.4), inserendo in esse l'angolo α'' e la lunghezza ℓ''.

Imponendo la condizione di equilibrio dei tratti rigidi di estremità, si ottiene una relazione tra gli insiemi $\mathbf{S}_f$ ed $\mathbf{S}''_\ell$

$$M_1 = -F'_{x1}s_y\ \ell + F'_{y1}s_x\ \ell + M'_{z1}$$

$$M_2 = -F'_{x2}t_y\ \ell + F'_{y2}t_x\ \ell + M'_{z2}$$

$$N = F'_{x2}$$

e quindi

$$\mathbf{S}_f = \mathbf{C}''\mathbf{S}''_\ell \tag{2.55}$$

con

$$\mathbf{C}'' = \begin{bmatrix} -s_y\ell & s_x\ell & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -t_y\ell & t_x\ell & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.56}$$

Dalle (2.54) e (2.55) si ottiene infine

$$\mathbf{S}_f = \mathbf{C}\ \mathbf{S}''_f \tag{2.57}$$

avendo posto

$$\mathbf{C} = \mathbf{C}''\mathbf{L}''_d\mathbf{T}'' \tag{2.58}$$

Il suddetto prodotto matriciale può essere effettuato anche analiticamente ottenendo in tal modo

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{22} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{2.59}$$

con

$$c_{11} = 1 + c(s_y \sin\alpha'' + s_x \cos\alpha'')$$
$$c_{12} = c(s_y \sin\alpha'' + s_x \cos\alpha'')$$
$$c_{13} = (s_y \cos\alpha'' - s_x \sin\alpha'')\ell$$
$$c_{21} = -c(t_y \sin\alpha'' + t_x \cos\alpha'')$$
$$c_{22} = 1 - c(t_y \sin\alpha'' + t_x \cos\alpha'')$$
$$c_{23} = -(t_y \cos\alpha'' - t_x \sin\alpha'')\ell$$
$$c_{31} = \frac{c \sin\alpha''}{\ell}$$
$$c_{32} = \frac{c \sin\alpha''}{\ell}$$
$$c_{33} = \cos\alpha''$$

La relazione (2.57) coincide formalmente con la (2.44) ottenuta per tratti rigidi coassiali. L'espressione (2.45) della matrice $\mathbf{C}$ si può ottenere come caso particolare della (2.59) ponendo

$$s_y = t_y = 0 \qquad s_x = s \qquad t_x = -t$$

e quindi

$$\sin\alpha'' = 0 \qquad \cos\alpha'' = 1$$

In maniera perfettamente analoga si possono definire le componenti di movimento fondamentali agli estremi dell'asta, $\mathbf{s}_f$, e del tratto deformabile, $\mathbf{s}''_f$. Si può facilmente dimostrare, anche in virtù del principio di dualità dimostrato nel paragrafo 3, che esse sono legate dalla relazione

$$\mathbf{s}''_f = \mathbf{C}^T \mathbf{s}_f \tag{2.60}$$

Azioni e componenti di movimento agli estremi del tratto deformabile sono sempre legati tra loro mediante la matrice di rigidezza di tale tratto $\mathbf{k}''_f$. La matrice di rigidezza dell'asta $\mathbf{k}_f$ è pertanto fornita ancora formalmente dalla (2.48), ricavata per tratti rigidi coassiali, inserendo in essa l'espressione qui ricavata per la matrice $\mathbf{C}$.

7. Azioni di incastro perfetto

Nei paragrafi precedenti si sono esaminate le relazioni esistenti tra azioni e componenti di movimento di estremità in un'asta non soggetta a carichi nei punti interni (cioè diversi da quelli di estremità). Per completare lo studio dell'asta occorre però valutare anche le azioni che insorgono agli estremi per effetto di carichi o distorsioni termiche applicate all'interno dell'asta, quando il movimento dei suoi estremi è impedito. Per evidenziare questo vincolo, tali azioni sono dette *azioni di incastro perfetto.*

Così come indicato in generale per le azioni di estremità, anche quelle di incastro perfetto possono essere descritte mediante le sei componenti rispetto al sistema di riferimento globale, mostrate col loro verso positivo nella figura 2.26 e racchiuse nel vettore $\mathbf{S}_g$.

Per determinarne il valore è però sempre più comodo operare nel sistema di riferimento locale e descriverle quindi mediante le componenti mostrate in figura 2.27 e racchiuse nel vettore $\mathbf{S}_\ell$. La trasformazione dall'uno all'altro insieme è possibile mediante la relazione (2.1) e la sua inversa.

Le azioni di incastro perfetto possono in generale essere calcolate mediante le equazioni differenziali (2.9) e (2.10), imponendo come condizione ai limiti l'annullarsi di spostamento e rotazione nei punti di estremità. La loro determinazione è ancora più immediata nei casi in cui sono disponibili relazioni algebriche tra carichi e deformazioni per schemi di trave semplicemente appoggiata. Poiché questo modo di procedere è fin troppo noto, si ritiene inutile approfondire in questo testo i passaggi necessari, concettualmente semplici ma in alcuni casi analiticamente complessi. Nei paragrafi

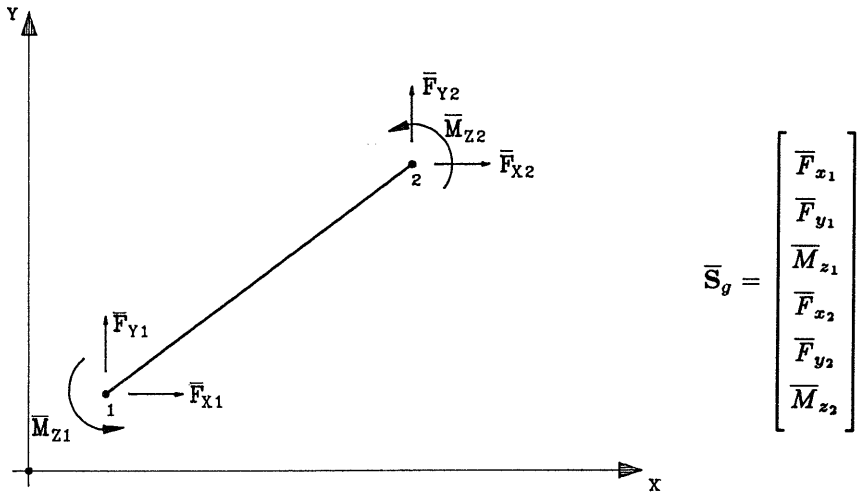


Fig. 2.26 — Azioni di incastro perfetto nel riferimento globale

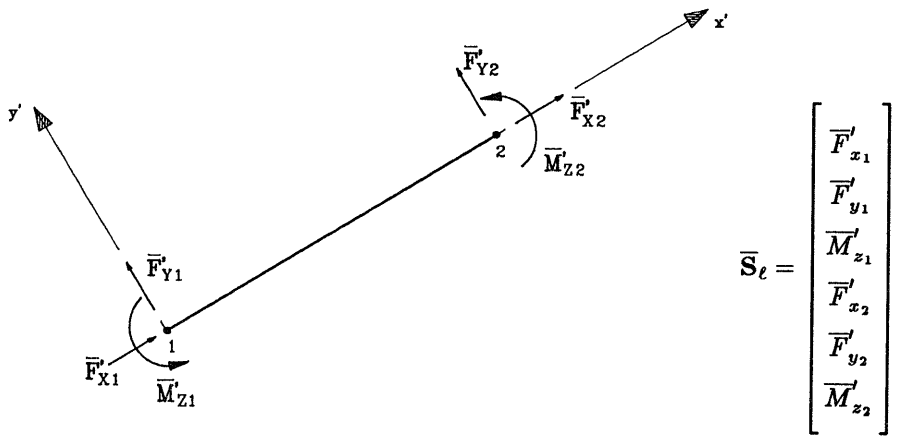


Fig. 2.27 — Azioni di incastro perfetto nel riferimento locale

successivi sono quindi riportate direttamente le formulazioni finali per numerosi schemi di carico, relative al caso di asta con sezione costante e per la quale si trascura l'aliquota di deformazione dovuta al taglio. Per i casi particolari di aste mostrati in precedenza sono riportate solo le espressioni relative ad un carico uniforme ortogonale all'asse; per carichi agenti nella direzione dell'asse valgono ancora le formulazioni dell'asta a sezione costante.

7.1. Carichi uniformemente distribuiti su tutta l'asta

Si prende in considerazione un carico uniformemente distribuito q agente su tutta l'asta, in direzione ortogonale all'asse, ed un carico n agente in direzione dell'asse. I versi positivi di tali carichi, concordi con quelli del sistema locale di riferimento, sono mostrati nella figura 2.28.

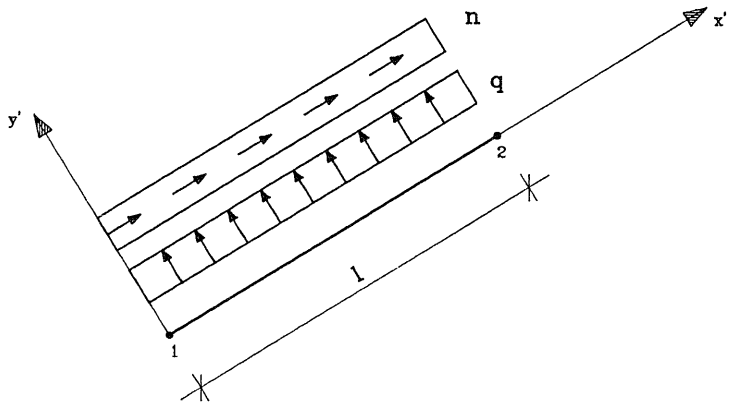


Fig. 2.28 — Carichi uniformemente distribuiti su tutta l'asta

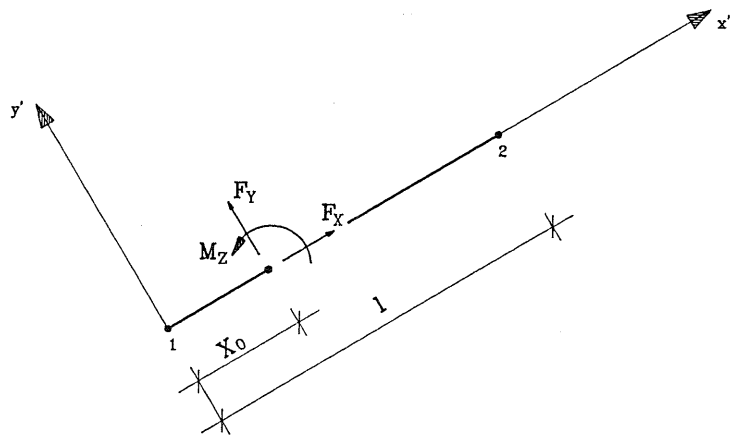
Il vettore che racchiude le componenti delle azioni di incastro perfetto nel riferimento locale è fornito dalle ben note espressioni

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -\frac{\ell}{2} \\ -\frac{\ell^2}{12} \\ 0 \\ -\frac{\ell}{2} \\ \frac{\ell^2}{12} \end{bmatrix} q + \begin{bmatrix} -\frac{\ell}{2} \\ 0 \\ 0 \\ -\frac{\ell}{2} \\ 0 \\ 0 \end{bmatrix} n \qquad (2.61)$$

7.2. Carichi concentrati in un punto dell'asta

Si prendono in considerazione una forza di componenti F_x ed F_y rispetto

al sistema locale ed una coppia M_z, applicate in un punto dell'asta posto a distanza x_0 dal primo estremo (fig.2.29).



$$a = \frac{x_0}{\ell}$$
$$b = 1 - a$$

Fig. 2.29 — Carichi concentrati in un punto dell'asta

Le azioni di incastro perfetto sono funzione dei carichi, della lunghezza dell'asta e delle grandezze adimensionali a e b che individuano il punto di applicazione dei carichi e sono fornite dalle espressioni

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -b^2(1+2a) \\ -ab^2\ell \\ 0 \\ -a^2(1+2b) \\ a^2b\ell \end{bmatrix} F_x + \begin{bmatrix} -b \\ 0 \\ 0 \\ -a \\ 0 \\ 0 \end{bmatrix} F_y + \begin{bmatrix} 0 \\ 6\frac{ab}{\ell} \\ b(2a-b) \\ 0 \\ -6\frac{ab}{\ell} \\ a(2b-a) \end{bmatrix} M_z \qquad (2.62)$$

7.3. Carichi distribuiti con andamento lineare su parte dell'asta

Si considera un carico distribuito agente in direzione ortogonale all'asse dell'asta, variabile linearmente dal valore q_1 al valore q_2, ed applicato sulla parte di essa compresa tra le ascisse x_1 ed x_2 (fig.2.30). Per valutarne

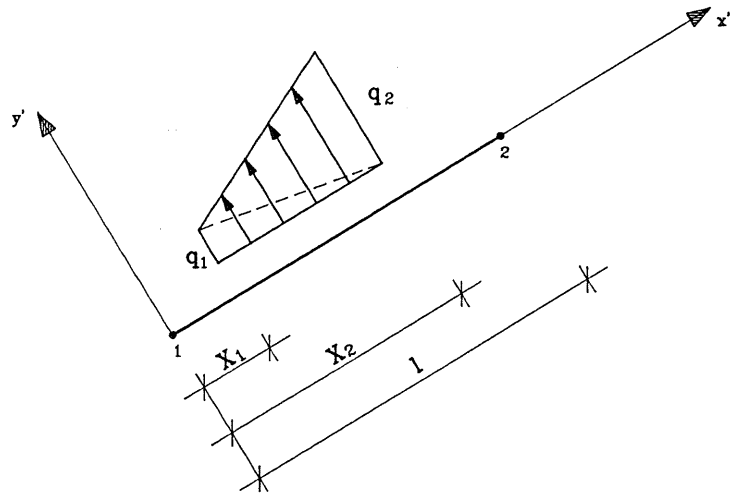


Fig. 2.30 — Carico linearmente distribuito su parte dell'asta

l'effetto, esso può essere considerato come somma di due carichi triangolari, il primo variabile da q_1 a 0, il secondo da 0 a q_2.

Si definiscono le seguenti grandezze adimensionali, dipendenti dalla posizione e dall'ampiezza del tratto in cui è applicato il carico

$$r_1 = \frac{3}{2} - \frac{x_1 + x_2/2}{\ell} \qquad s_1 = 3 - 2r_1$$

$$r_2 = \frac{x_2 + x_1/2}{\ell} \qquad s_2 = 3 - 2r_2 \tag{2.63}$$

$$t = \frac{x_2 - x_1}{\ell}$$

ed inoltre f_1, g_1, h_1 ed f_2, g_2, h_2 valutate rispettivamente in funzione di r_1, s_1 ed r_2, s_2 mediante le espressioni

$$\begin{aligned} f(r,s,t) &= \frac{t}{27}\left[40r^2 s + t^2(2t - 30r + 15)\right] \\ g(r,s,t) &= \frac{t}{18}\left[20rs^2 - t^2(2t - 30r + 30)\right] \\ h(r,s,t) &= 3f - 2g \end{aligned} \tag{2.64}$$

Tutte queste grandezze assumono valore unitario quando il carico è applicato all'intera asta, cioè quando $x_1 = 0$ ed $x_2 = \ell$.

Le azioni di incastro perfetto nel riferimento locale sono quindi fornite da

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -\dfrac{20r_1t + h_1}{60}\,\ell \\ -\dfrac{f_1\ell^2}{20} \\ 0 \\ -\dfrac{10s_1t - h_1}{60}\,\ell \\ \dfrac{g_1\ell^2}{30} \end{bmatrix} q_1 + \begin{bmatrix} 0 \\ -\dfrac{10s_2t - h_2}{60}\,\ell \\ -\dfrac{g_2\ell^2}{30} \\ 0 \\ -\dfrac{20r_2t - h_2}{60}\,\ell \\ \dfrac{f_2\ell^2}{20} \end{bmatrix} q_2 \qquad (2.65)$$

7.4. Variazioni termiche

Si considera una variazione termica, applicata in maniera uguale lungo tutta l'asta, costituita da un'aliquota ΔT_u costante per tutti i punti della sezione ed un'aliquota ΔT_f variabile linearmente, a farfalla (fig.2.31).

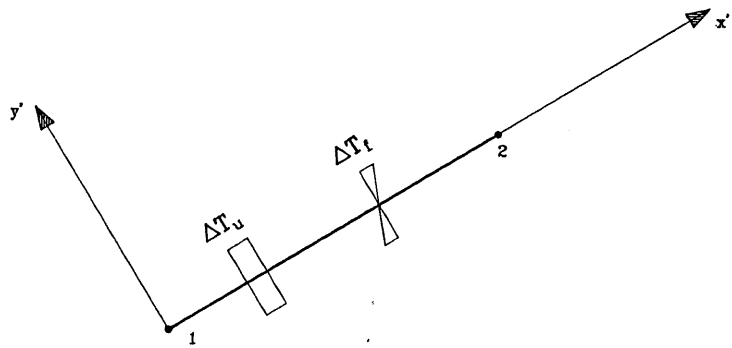


Fig. 2.31 — Variazioni termiche

La variazione termica uniforme è considerata positiva quando corrisponde ad un incremento di temperatura. La variazione a farfalla è positiva quando si ha un incremento di temperatura sull'estremo della sezione posto nel verso positivo dell'asse y' (cioè superiormente nella figura). L'effetto della variazione termica sull'asta incastrata dipende dal coefficiente di deformabilità termica α e dalle caratteristiche geometriche della sezione tra-

sversale dell'asta, espresse mediante l'altezza H, l'area A ed il momento d'inerzia I. Le azioni di incastro perfetto sono fornite da

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} EA\alpha \\ 0 \\ 0 \\ -EA\alpha \\ 0 \\ 0 \end{bmatrix} \Delta T_u + \begin{bmatrix} 0 \\ 0 \\ -\dfrac{EI\alpha}{H} \\ 0 \\ 0 \\ -\dfrac{EI\alpha}{H} \end{bmatrix} \Delta T_f \qquad (2.66)$$

7.5. Effetto della deformazione dovuta al taglio

Si prenda nuovamente in esame l'asta soggetta ad un carico q uniformemente distribuito, ortogonale all'asse. Per valutare le azioni di incastro perfetto tenendo conto anche della deformazione dovuta al taglio si può partire dallo schema di trave semplicemente appoggiata ed applicare ad esso due coppie tali da annullare la rotazione degli estremi. Per la simmetria geometrica e di carico, il taglio provocato da q ha un andamento lineare emisimmetrico. Le deformazioni da esso provocate nella parte destra dell'asta sono quindi uguali ed opposte a quelle provocate nella parte sinistra e non provocano rotazione degli estremi. Sempre per la simmetria, le coppie necessarie per la congruenza sono uguali ed opposte e non provocano taglio. Di conseguenza, le azioni di incastro perfetto sono, per questo caso come per qualunque altro schema di carico che rispetti la simmetria, identiche a quelle determinate trascurando la deformazione dovuta al taglio.

7.6. Asta con cerniere all'estremità

Per lo schema di asta a sezione costante con una cerniera al primo estremo, soggetta ad un carico uniformemente distribuito q ortogonale

all'asse, trascurando l'effetto della deformazione dovuta al taglio si ha

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -\frac{3}{8}\ell \\ 0 \\ 0 \\ -\frac{5}{8}\ell \\ \frac{\ell^2}{8} \end{bmatrix} q \tag{2.67}$$

Se la cerniera è al secondo estremo si ha invece

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -\frac{5}{8}\ell \\ -\frac{\ell^2}{8} \\ 0 \\ -\frac{3}{8}\ell \\ 0 \end{bmatrix} q \tag{2.68}$$

Se entrambi gli estremi sono incernierati si ha infine

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -\frac{\ell}{2} \\ 0 \\ 0 \\ -\frac{\ell}{2} \\ 0 \end{bmatrix} q \tag{2.69}$$

7.7. Trave su suolo elastico alla Winkler

Si prende in esame un'asta poggiata su un suolo alla Winkler, di costante elastica c. Si indica con b la larghezza della superficie di contatto col suolo. Se la trave è libera agli estremi, un carico q uniformemente distribuito ortogonale al suo asse provoca uno spostamento nella direzione di applicazione uguale in tutti i punti e pari a $\frac{q}{cb}$. Le azioni di incastro perfetto devono essere tali da annullare questo spostamento, cioè da provocare uno spostamento $-\frac{q}{cb}$. Possono essere quindi determinate mediante la matrice di rigidezza nel sistema locale, con

$$\overline{\mathbf{S}}_\ell = \mathbf{k}_\ell \begin{bmatrix} 0 \\ -\frac{1}{cb} \\ 0 \\ 0 \\ -\frac{1}{cb} \\ 0 \end{bmatrix} q \tag{2.70}$$

7.8. Asta con tratti rigidi

Si prende in esame l'asta con tratti rigidi coassiali di estremità, già mostrata nella figura 2.18. Con i simboli ivi indicati, e considerando agente un carico q uniformemente distribuito ortogonale all'asse, si ha

$$\overline{\mathbf{S}}_\ell = \begin{bmatrix} 0 \\ -\frac{\ell}{2}(1+s-t) \\ -\frac{\ell^2}{12}\, z_1 \\ 0 \\ -\frac{\ell}{2}(1-s+t) \\ \frac{\ell^2}{12}\, z_2 \end{bmatrix} q \tag{2.71}$$

con

$$z_1 = (1-s-t)^2 + 6s(1-s-t) + 6s^2$$
$$z_2 = (1-s-t)^2 + 6t(1-s-t) + 6t^2 \qquad (2.72)$$

8. Relazione tra carichi, azioni e componenti di movimento di estremità

Nei paragrafi precedenti si è mostrata la relazione lineare che esiste tra azioni e componenti di movimento di estremità in assenza di carichi interni all'asta, sintetizzata mediante la matrice di rigidezza. Si sono poi fornite le espressioni per valutare le azioni che nascono, per effetto dei carichi, negli estremi dell'asta impediti al movimento. Poiché con le ipotesi inizialmente formulate è valido il principio di sovrapposizione degli effetti, si possono combinare i risultati ottenuti, scrivendo le relazioni generali che legano carichi, azioni e componenti di movimento di estremità

$$\mathbf{S}_\ell = \overline{\mathbf{S}}_\ell + \mathbf{k}_\ell \, \mathbf{s}_\ell$$
$$\mathbf{S}_g = \overline{\mathbf{S}}_g + \mathbf{k}_g \, \mathbf{s}_g \qquad (2.73)$$

valide rispettivamente nel riferimento locale e in quello globale. Queste relazioni costituiscono la base di partenza per la scrittura del sistema di equazioni di equilibrio, come mostrato nel successivo capitolo. Mediante esse, inoltre, una volta determinati i valori delle componenti di movimento che soddisfano tali equazioni è possibile determinare l'entità delle azioni che insorgono agli estremi di ciascun'asta e quindi anche le caratteristiche di sollecitazione nei punti interni ad essa.

9. Determinazione delle caratteristiche di sollecitazione nei punti interni dell'asta

Nel paragrafo 4 si sono scritte le relazioni generali che legano carichi, componenti di movimento e caratteristiche di sollecitazione nei punti interni dell'asta. La conoscenza delle componenti di movimento e delle azioni nodali consente di scrivere le condizioni ai limiti necessarie per la determinazione di quelle grandezze. L'entità delle caratteristiche di sollecitazione può però essere determinata in maniera immediata, perché nell'asta le azioni nodali ed i carichi interni costituiscono un insieme equilibrato di forze. Nella generica sezione a distanza x dal primo estremo, lo sforzo normale, il momento flettente ed il taglio possono quindi essere calcolati come

risultante delle forze (nodali ed interne) agenti alla sinistra della sezione. Essendo valido il principio di sovrapposizione degli effetti, si possono valutare separatamente le aliquote dovute alle azioni nodali ed ai singoli carichi interni. Per comodità del lettore si riportano nel seguito le relative formulazioni, valide con la simbologia e la convenzione dei segni definite in precedenza.

Azioni di estremità.

$$N(x) = -F'_{x1}$$
$$M(x) = F'_{y1}x - M'_{z1}$$
$$T(x) = F'_{y1}$$

Carichi uniformemente distribuiti su tutta l'asta.

$$N(x) = -nx$$
$$M(x) = \frac{qx^2}{2}$$
$$T(x) = qx$$

Carichi concentrati in un punto dell'asta.

per $x > x_0$

$$N(x) = -F_x$$
$$M(x) = F_y(x - x_0) - M_z$$
$$T(x) = F_y$$

Carichi distribuiti con andamento lineare su parte dell'asta.

ponendo

$$\Delta x_1 = x - x_1 \qquad \Delta x_2 = x - x_2$$
$$\Delta \quad = x_2 - x_1 \qquad u = \frac{x - x_1}{x_2 - x_1}$$

si ha per $x_1 < x < x_2$

$$M(x) = \left[\left(1 - \frac{u}{3}\right) q_1 + \frac{u}{3} q_2\right] \frac{\Delta x_1^2}{2}$$

$$T(x) = \left[\left(1 - \frac{u}{2}\right) q_1 + \frac{u}{2} q_2\right] \Delta x_1$$

e per $x \geq x_2$

$$T_0 = \frac{q_1 + q_2}{2} \Delta$$

$$M_0 = \left(\frac{2}{3} q_1 + \frac{1}{3} q_2\right) \frac{\Delta^2}{2}$$

$$M(x) = T_0 \ \Delta x_2 + M_0$$

$$T(x) = T_0$$

Le espressioni innanzi riportate valgono per tutti i tipi di asta, sia che gli estremi siano solidali al nodo oppure incernierati e che si tenga conto, o no, della deformazione a taglio. Solo il caso di trave su suolo elastico alla Winkler si presenta differente, perché in esso bisogna tener conto anche dei carichi trasmessi dal terreno, proporzionali agli spostamenti e quindi non noti a priori. In tale situazione è quindi necessario rifarsi alle relazioni generali, con criterio analogo a quello seguito per la determinazione della matrice di rigidezza. In particolare, spostamenti e caratteristiche di sollecitazione sono fornite dalle (2.32), nelle quali vanno inseriti i valori delle costanti di integrazione determinati mediante la (2.34) in base ai valori degli spostamenti di estremità.

CAPITOLO TERZO

L'INSIEME PIANO DI ASTE

1. Generalità

La prima tipologia strutturale che si prende in esame è denominata *insieme piano di aste* perché costituita da nodi tutti appartenenti ad un piano, collegati da aste la cui sezione retta è simmetrica rispetto a tale piano. I carichi agenti sui nodi o sulle aste sono anch'essi simmetrici rispetto al piano e provocano quindi spostamenti contenuti in esso.

Nell'esaminare questa tipologia, si userà sempre come riferimento la terna ortogonale destrorsa xyz, che contiene nodi ed aste nel piano xy, già presentata nel capitolo precedente e denominata *sistema di riferimento globale* (fig.3.1). Poiché, salvo specifica indicazione contraria, tutte le grandezze utilizzate in questo capitolo sono riferite a questa terna, si ometterà in seguito il pedice g che nel capitolo precedente era stato utilizzato come segno distintivo rispetto a grandezze riferite al sistema locale o all'insieme fondamentale.

Le aste della struttura sono contraddistinte da numeri consecutivi. Si indicherà con j l'asta generica, con m il numero totale di aste. Per ciascuna di esse valgono le ipotesi formulate nel capitolo precedente. Ogni asta è collegata all'estremità a due nodi; l'insieme delle informazioni relative ai collegamenti aste–nodi della struttura è denominato *tabella di incidenza delle aste* (un esempio è mostrato più avanti nella figura 3.7).

Anche i nodi sono contrassegnati con numeri consecutivi. Si indicherà

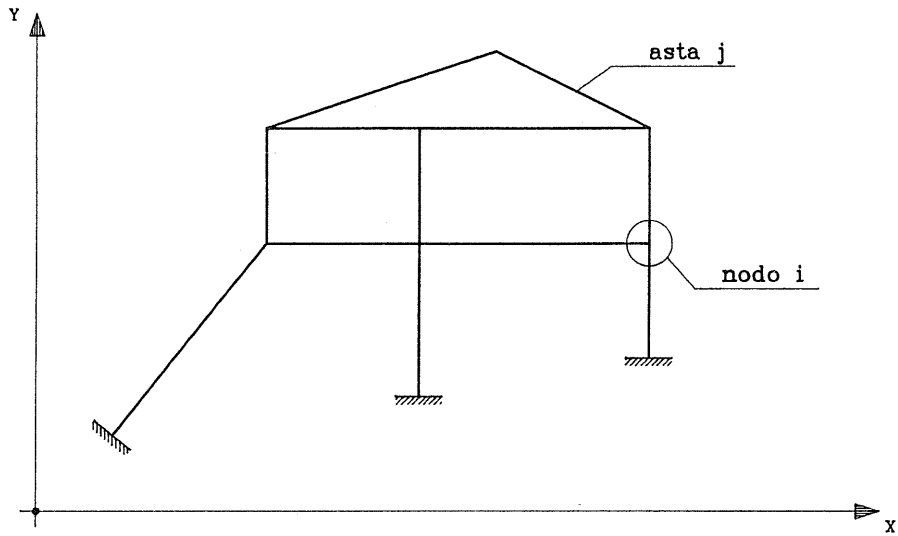


Fig. 3.1 — Insieme piano di aste – sistema di riferimento globale

con i il nodo generico, con n il numero totale di nodi. Ciascun nodo è supposto puntiforme, cioè privo di dimensioni fisiche ovvero coincidente con un punto geometrico; la sua posizione è individuata mediante le coordinate x ed y. L'eventuale presenza di zone rigide di dimensioni finite all'incrocio delle aste può essere schematizzata considerando come nodo un punto qualsiasi della zona di incrocio, ed aggiungendo agli estremi delle aste tratti di connessione infinitamente rigidi (fig.3.2).

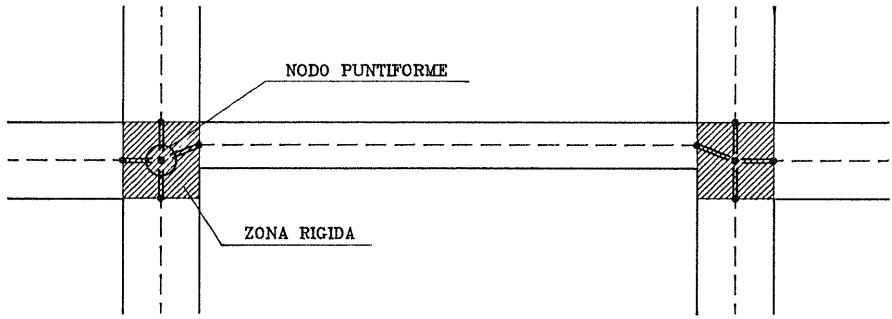


Fig. 3.2 — Schematizzazione nel caso di zone nodali rigide non puntiformi

In generale, una struttura può essere costituita da nodi–cerniera o da nodi rigidi. Il primo termine indica che le aste che confluiscono nel nodo sono libere di ruotare, ciascuna indipendentemente dalle altre, e non trasmettono mutuamente azioni flettenti (fig.3.3a). Questo modello è utiliz-

zato per le travature reticolari nelle quali le aste, pur non essendo rigorosamente incernierate, lavorano sostanzialmente a sforzo assiale consentendo quindi di trascurare per semplicità di calcolo le sollecitazioni flesso–taglianti. Il movimento dei nodi–cerniera consiste solo in uno spostamento, individuato in uno schema piano mediante le due componenti v_x e v_y.

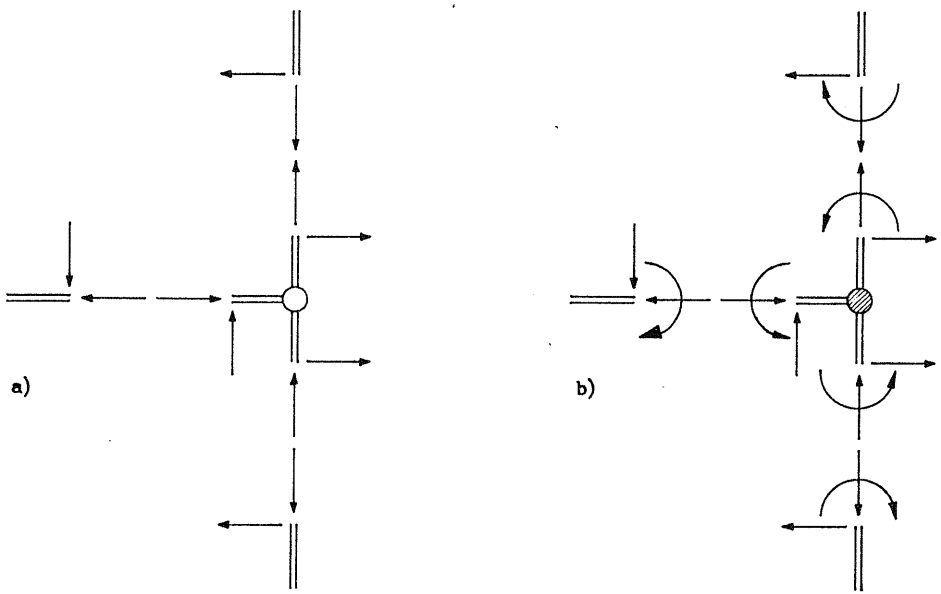


Fig. 3.3 — a) nodo–cerniera
b) nodo rigido

Col termine nodo rigido, o meglio "connesso rigidamente alle aste", si indica che spostamento e rotazione del nodo coincidono con quelli delle aste in esso confluenti (fig.3.3b). Questo modello è utilizzato nelle strutture intelaiate, nelle quali la trasmissione di azioni flettenti tra le aste è essenziale per l'assorbimento dei carichi esterni. Poiché il presente testo è finalizzato all'analisi di questa tipologia, si suppone nel seguito che tutti i nodi siano rigidi. Questa ipotesi non inficia comunque la generalità della trattazione. L'eventuale presenza di un nodo cerniera può infatti essere efficacemente descritta con l'ausilio di elementi che dispongano di tale grado di vincolo all'estremità convergente nel nodo in questione (fig.3.4a).

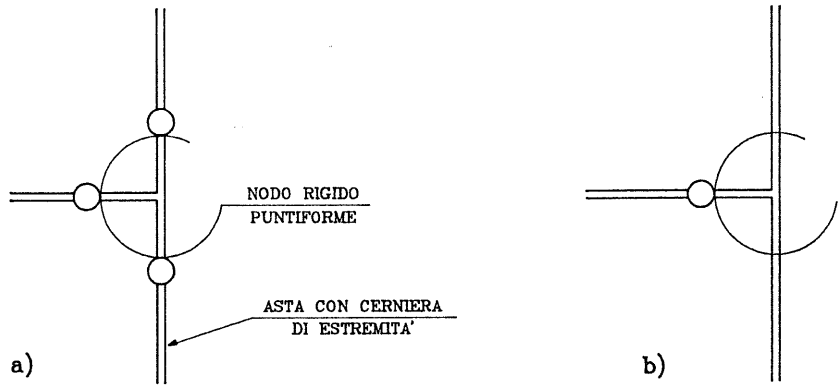
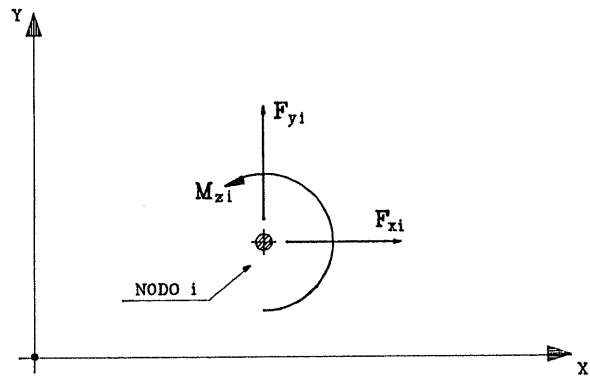
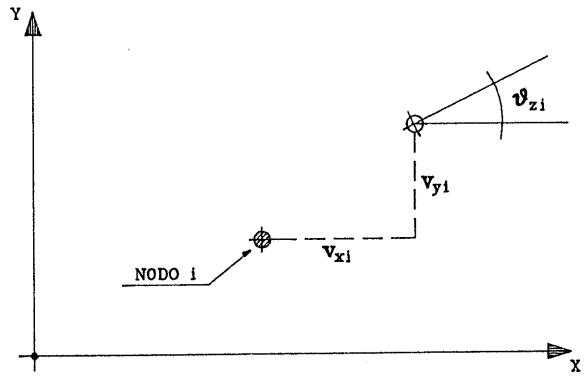


Fig. 3.4 — Utilizzazione di aste con cerniere di estremità
a) per simulare un nodo cerniera
b) in presenza contemporanea di aste vincolate rigidamente



$$\mathbf{R}_i = \begin{bmatrix} F_{x_i} \\ F_{y_i} \\ M_{z_i} \end{bmatrix}$$

Fig. 3.5 — Azioni sul nodo rigido



$$\mathbf{r}_i = \begin{bmatrix} v_{x_i} \\ v_{y_i} \\ \theta_{z_i} \end{bmatrix}$$

Fig. 3.6 — Componenti di movimento del nodo rigido

Va anzi osservato come una siffatta impostazione consenta, con estrema facilità e senza variazione alcuna dell'impostazione generale del problema, di contemplare anche il caso più generale della confluenza in uno stesso nodo di aste vincolate rigidamente ed aste incernierate (fig.3.4b).

Le azioni applicate dall'esterno o trasmesse dalle aste al generico nodo i di uno schema piano possono essere definite mediante una forza comunque diretta ed una coppia, cioè con le componenti rispetto al sistema di riferimento globale mostrate col loro verso positivo nella figura 3.5 e racchiuse nel vettore $\mathbf{R}_i$. In maniera duale, il suo movimento è definito dalle componenti mostrate nella figura 3.6 e racchiuse nel vettore $\mathbf{r}_i$.

Per analizzare la struttura, occorre conoscere se ciascun nodo è libero o se esistono vincoli esterni che ne limitano il movimento. In generale può essere presente un vincolo qualsiasi, ad esempio un carrello che impedisca lo spostamento in una direzione inclinata rispetto agli assi. Tuttavia nelle applicazioni svolte nel testo verranno considerati solo vincoli, rigidi od elastici, immediatamente corrispondenti alle componenti di movimento definite.

Nelle strutture sono possibili anche vincoli mutui tra i nodi. Nel caso di un edificio, una situazione del genere è provocata dagli impalcati, che presentano quasi sempre una elevata rigidezza nel loro piano (usualmente orizzontale). Nello schema piano che qui si esamina si considerano quindi possibili vincoli mutui che impongano l'uguaglianza degli spostamenti orizzontali di gruppi di nodi.

2. Azioni e componenti di movimento degli estremi delle aste

Come già illustrato nel capitolo precedente, le azioni applicate agli estremi della generica asta j sono costituite da una forza, individuata mediante le due componenti rispetto al sistema di riferimento globale, ed una coppia per ciascuno dei due estremi. Queste sei componenti sono raggruppate nel vettore $\mathbf{S}_j$.

In analogia a quanto fatto per la singola asta, si puo definire per un insieme piano di m aste il vettore di azioni di estremità $\mathbf{S}$ che ha come componenti l'insieme ordinato dei vettori azioni agli estremi di ogni asta; l'ordine assunto per tali vettori coincide con quello definito mediante la numerazione delle aste

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \\ \cdots \\ \mathbf{S}_j \\ \cdots \\ \mathbf{S}_m \end{bmatrix} \tag{3.1}$$

In maniera duale, le componenti di movimento degli estremi della generica asta j sono costituite dalle due componenti di traslazione e dalla rotazione dei due estremi, raggruppate nel vettore $\mathbf{s}_j$. Per l'insieme piano di aste, si può definire il vettore di componenti di movimento degli estremi $\mathbf{s}$ che è costituito dall'insieme ordinato dei vettori spostamento delle estremità di ogni asta

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \cdots \\ \mathbf{s}_j \\ \cdots \\ \mathbf{s}_m \end{bmatrix} \tag{3.2}$$

L'introduzione di queste entità vettoriali non fornisce alcuna informazione aggiuntiva rispetto a quelle già disponibili per il singolo elemento asta. In particolare, con essa non si tiene ancora conto della presenza di connessioni mutue né della presenza di vincoli esterni. Consente però notevoli semplificazioni formali nell'analisi del problema.

Si osservi come la simbologia adottata coincida, a meno del pedice j, con quella delle singole aste. Pertanto, nel seguito, la mancanza di tale pedice indicherà che le relazioni sono riferite, invece che ad una asta, ad un loro insieme piano.

Nel capitolo precedente si è mostrato come la relazione fra carichi interni, azioni e componenti di movimento degli estremi della generica asta j si concretizza, mediante la definizione del vettore azioni di incastro perfetto e della matrice di rigidezza dell'asta stessa, nell'espressione matriciale

$$\mathbf{S}_j = \overline{\mathbf{S}}_j + \mathbf{k}_j \mathbf{s}_j$$

Tale relazione può essere formalmente estesa all'insieme piano di aste, eliminando il pedice j e scrivendo

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{k}\,\mathbf{s} \tag{3.3}$$

Il vettore $\overline{\mathbf{S}}$, analogamente a quanto visto per $\mathbf{S}$ ed $\mathbf{s}$, è semplicemente un raggruppamento delle azioni di incastro perfetto delle singole aste. Allo stesso modo, anche la matrice di correlazione $\mathbf{k}$ è ottenuta assemblando le matrici di rigidezza corrispondenti, nell'ordine definito dalla numerazione

delle aste

$$\mathbf{k} = \begin{bmatrix} \mathbf{k}_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \mathbf{k}_2 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{k}_j & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & \mathbf{k}_m \end{bmatrix} \tag{3.4}$$

3. Le incognite del problema

Nel metodo degli spostamenti le incognite del problema sono costituite dalle componenti di movimento dei nodi della struttura. Se non esistessero vincoli esterni o vincoli mutui tra i nodi, tali componenti sarebbero tutte ammesse. Le incognite del problema sarebbero quindi individuate dal vettore spostamenti nodali

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \dots \\ \mathbf{r}_i \\ \dots \\ \mathbf{r}_n \end{bmatrix}$$

che racchiude le componenti di movimento di tutti i nodi della struttura, nell'ordine definito dalla numerazione dei nodi stessi.

Non è però possibile prendere in esame una struttura i cui nodi siano tutti liberi. Essa sarebbe infatti labile e, anche se soggetta ad un insieme di carichi in equilibrio, i suoi spostamenti sarebbero indeterminati in quanto definiti a meno di un moto rigido. È pertanto necessaria la presenza di vincoli esterni che annullino alcune componenti di movimento, che non vanno computate nel numero delle incognite. Il vettore spostamenti nodali racchiuderà quindi solo le componenti di movimento compatibili coi vincoli.

Si consideri come esempio la struttura mostrata in figura 3.7, costituita da quattro aste e cinque nodi.

Se essa fosse, per assurdo, priva di qualsiasi vincolo presenterebbe $3 \times 5 = 15$ gradi di libertà. La presenza di un carrello nel nodo 1, consentendo solamente rotazione e traslazione orizzontale, rende nulla, e quindi nota a priori, la componente verticale dello spostamento nodale. La presenza di un incastro nei nodi 4 e 5 rende nulle, e quindi ancora una volta note, le tre componenti del moto di tali nodi. Il numero delle incognite del problema risulta pertanto ridotto a $15 - 1 - 2 \times 3 = 8$.

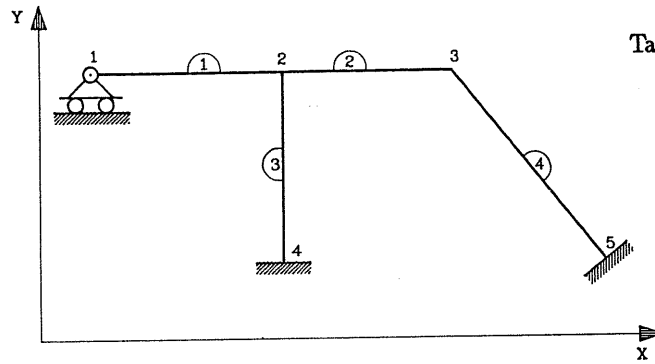


Tabella di incidenza delle aste

	estremo	
asta	1	2
1	1	2
2	2	3
3	2	4
4	3	5

Fig. 3.7 — Schema di struttura con aste deformabili estensionalmente

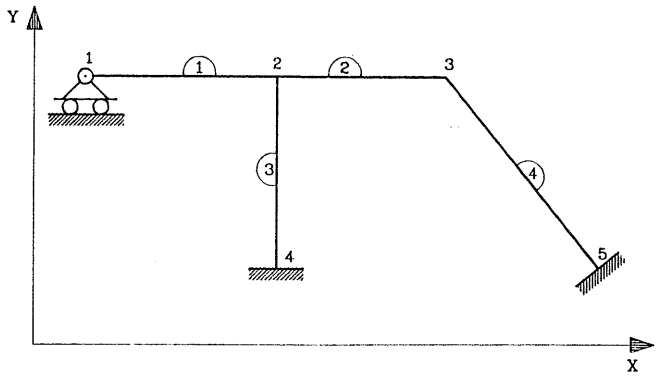
Nel caso in esame, il vettore spostamenti nodali è pertanto costituito da

$$\mathbf{r} = \begin{bmatrix} v_{x1} \\ \theta_{z1} \\ v_{x2} \\ v_{y2} \\ \theta_{z2} \\ v_{x3} \\ v_{y3} \\ \theta_{z3} \end{bmatrix}$$

Una ulteriore riduzione delle incognite è determinata dalla introduzione di vincoli mutui fra i nodi. È il caso, già citato, della presenza di impalcati infinitamente rigidi in uno schema intelaiato spaziale, ovvero di traversi orizzontali indeformabili estensionalmente in uno schema piano. Tale circostanza determina l'uguaglianza delle componenti in direzione x degli spostamenti dei nodi appartenenti al traverso e quindi la sostituzione di più incognite (gli spostamenti orizzontali dei nodi) con una sola (lo spostamento orizzontale del traverso). Il vettore spostamenti nodali, che individua le incognite, sarà costituito da

$$\mathbf{r} = \begin{bmatrix} \mathbf{t} \\ \mathbf{r}'_1 \\ \cdots \\ \cdots \\ \mathbf{r}'_n \end{bmatrix} \tag{3.5}$$

avendo indicato con $\mathbf{t}$ il vettore che contiene gli spostamenti orizzontali dei traversi, nell'ordine a questi attribuito, ed $\mathbf{r}'_1 \ldots \mathbf{r}'_n$ le componenti di movimento dei nodi non soggette a vincoli esterni o mutui.



I nodi 1, 2, 3 sono collegati da un traverso inestensibile

Fig. 3.8 — Schema con traverso orizzontale indeformabile estensionalmente

Nella figura 3.8 è mostrato uno schema che differisce da quello dell'esempio precedente solo per il fatto che i nodi 1, 2 e 3 fanno parte di un traverso inestensibile. In tal caso gli spostamenti v_{x1}, v_{x2} e v_{x3} devono essere uguali e possono denominarsi t_1. Il numero di incognite si riduce quindi a $8-2=6$ e si ha

$$\mathbf{r} = \begin{bmatrix} t_1 \\ \theta_{z1} \\ v_{y2} \\ \theta_{z2} \\ v_{y3} \\ \theta_{z3} \end{bmatrix}$$

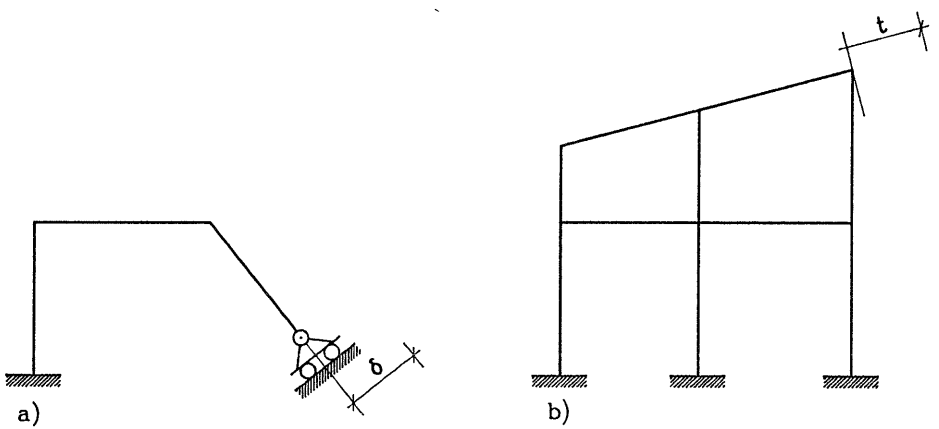


Fig. 3.9 — Vincoli esterni o mutui in direzione inclinata
a) carrello su un piano inclinato
b) traverso inestensibile non orizzontale

L'impostazione qui seguita nella individuazione delle incognite è molto generale ma non si adatta ad alcune situazioni che possono verificarsi, sia pur raramente, nella pratica professionale. Un primo caso è quello di vincoli esterni che impongono spostamenti in direzioni prefissate (fig.3.9a); un altro è quello di traversi inestensibili inclinati che vincolano mutuamente i nodi in una direzione diversa dall'orizzontale (fig.3.9b).

Per entrambi si dovrebbe assumere come componente di movimento incognita la traslazione nella direzione in questione, che non coincide con quella degli assi coordinati. Ad esempio, nello schema descritto dalla figura 3.10, nel quale il nodo 6 è vincolato da un carrello non orizzontale ed i nodi 1, 2 e 3 sono collegati da un traverso inestensibile inclinato, il vettore spostamenti nodali incogniti è costituito da

$$\mathbf{r} = \begin{bmatrix} t_1 \\ v_{y1} \\ \theta_{z1} \\ v_{y2} \\ \theta_{z2} \\ v_{y3} \\ \theta_{z3} \\ \delta_6 \\ \theta_{z6} \end{bmatrix}$$

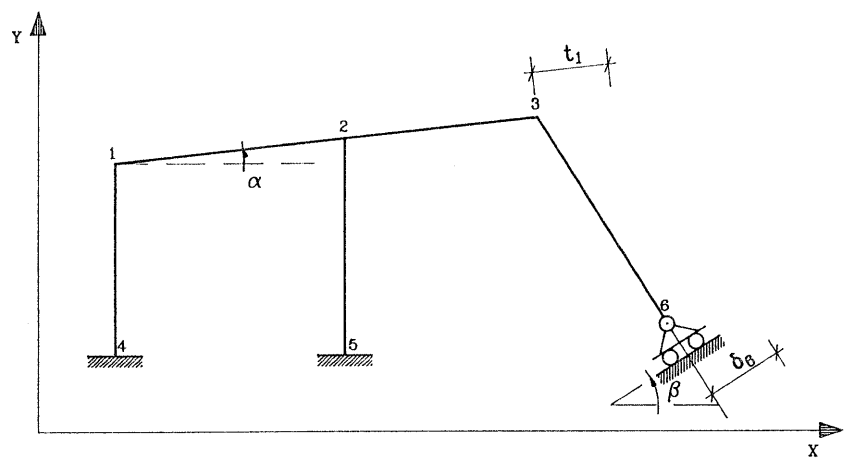


Fig. 3.10 — Schema con carrello inclinato e traverso inestensibile non orizzontale

In tutto il testo verrà mantenuta la prima impostazione, assumendo sempre come incognite le componenti di traslazione in direzione degli assi coordinati. La teoria sviluppata non viene però inficiata da questa limitazione; nel corso dell'esposizione si mostreranno quindi le variazioni applicative che si dovrebbero effettuare per rimuoverla.

4. Relazione tra incognite, componenti di movimento dei nodi e componenti di movimento degli estremi delle aste

Per quanto detto nel paragrafo precedente, le componenti di movimento del generico nodo i sono nulle (se impedite da un vincolo) oppure coincidenti con una delle incognite. Esiste quindi una relazione lineare tra il vettore componenti di movimento del nodo $\mathbf{r}_i$ ed il vettore spostamenti nodali $\mathbf{r}$, che può esprimersi mediante una matrice di corrispondenza $\mathbf{a}_i$ con

$$\mathbf{r}_i = \mathbf{a}_i \, \mathbf{r} \tag{3.6}$$

Ad esempio, per lo schema di figura 3.8 si ha

$$\mathbf{a}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{a}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{a}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{a}_4 = \mathbf{a}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Per l'impostazione adottata, tutti i termini delle matrici $\mathbf{a}_i$ sono nulli oppure unitari. Inoltre, può essere diverso da zero solo un termine per ogni riga della matrice. Per questi motivi, nelle applicazioni concrete mediante elaboratori elettronici le informazioni fornite dalla matrice $\mathbf{a}_i$ possono essere contenute, con un minor ingombro di memoria, in un vettore $\mathbf{n}_i$, avente lo stesso numero di righe (*vettore di corrispondenza*). L'unico termine di

ciascuna riga assume il valore 0 se tutti i termini della riga di $\mathbf{a}_i$ sono nulli, ovvero il valore k, se k è il numero d'ordine della colonna che contiene il valore 1, cioè se l'incognita corrispondente alla componente di movimento in questione è in posizione k nel vettore $\mathbf{r}$. Nel caso in esame i vettori di corrispondenza sono

$$\mathbf{n}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \qquad \mathbf{n}_2 = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} \qquad \mathbf{n}_3 = \begin{bmatrix} 1 \\ 5 \\ 6 \end{bmatrix} \qquad \mathbf{n}_4 = \mathbf{n}_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Questa semplificazione non è invece attuabile quando si ammettono come incognite le componenti di traslazione in direzione inclinata rispetto agli assi. Ad esempio, per lo schema di figura 3.10 la matrice $\mathbf{a}_1$ vale

$$\mathbf{a}_1 = \begin{bmatrix} \cos\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sin\alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

e contiene i termini $\sin\alpha$ e $\cos\alpha$, perché la traslazione t_1 contribuisce in tale proporzione agli spostamenti in direzione x ed y; inoltre la seconda riga contiene due termini non nulli, perché sia t_1 che v_{y1} provocano spostamento verticale.

Poiché la connessione nodo–asta è rigida, le componenti di movimento (nel riferimento globale) degli estremi di un'asta coincidono con le componenti di movimento dei nodi di estremità. Per la generica asta j, che collega i nodi i e k, si può scrivere quindi la relazione

$$\mathbf{s}_j = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{r}_k \end{bmatrix}$$

che impone la congruenza tra gli estremi dell'asta j ed i nodi i e k. Da questa, utilizzando la (3.6) si ricava la relazione tra le componenti di movimento degli estremi dell'asta e le incognite

$$\mathbf{s}_j = \begin{bmatrix} \mathbf{a}_i \\ \mathbf{a}_k \end{bmatrix} \mathbf{r}$$

che può essere estesa all'intero insieme di aste, ottenendo

$$\mathbf{s} = \mathbf{a}\,\mathbf{r} \tag{3.7}$$

La matrice $\mathbf{a}$ è costituita dalle coppie a_i a_k relative a tutte le aste che compongono la struttura, nell'ordine definito dalla loro numerazione. Essa indica quale incognita corrisponde a ciascuna componente di movimento di estremità di un'asta ed impone quindi globalmente la congruenza tra estremi delle aste e nodi; per tale motivo è detta *matrice di congruenza.*

Nell'esempio già citato, in base alla tabella di incidenza mostrata nella figura 3.7 e valida anche per lo schema a traversi inestensibili della figura 3.8, si ottiene

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_2 \\ a_3 \\ a_2 \\ a_4 \\ a_3 \\ a_5 \end{bmatrix}$$

5. Le azioni nodali

Per il generico nodo i dello schema strutturale si è definito il vettore $\mathbf{R}_i$, che contiene le componenti nel riferimento globale delle azioni ad esso applicate dall'esterno, o trasmesse dalle aste che vi confluiscono. Prescindendo per un attimo dalla presenza di vincoli esterni o mutui, questa definizione può essere estesa all'intero insieme di nodi, introducendo un vettore azioni nodali $\mathbf{R}$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \cdots \\ \mathbf{R}_i \\ \cdots \\ \mathbf{R}_n \end{bmatrix}$$

che racchiude le azioni applicate a tutti i nodi della struttura, nell'ordine definito dalla numerazione dei nodi stessi.

Non tutte le azioni nodali sono però rilevanti ai fini della scrittura del sistema di equazioni che regola l'equilibrio dello schema. In presenza di vincoli esterni, non ha senso prendere in esame quelle azioni che vengono assorbite direttamente dai vincoli. Il vettore azioni nodali conterrà quindi solo le azioni non corrispondenti ai vincoli, cioè le azioni duali delle componenti di movimento incognite.

Ad esempio, per lo schema di figura 3.7 il vettore $\mathbf{R}$ contiene 8 elementi, corrispondenti agli 8 gradi di libertà ammessi

$$\mathbf{R} = \begin{bmatrix} F_{x1} \\ M_{z1} \\ F_{x2} \\ F_{y2} \\ M_{z2} \\ F_{x3} \\ F_{y3} \\ M_{z3} \end{bmatrix}$$

La presenza di vincoli mutui tra i nodi, come quelli provocati da impalcati orizzontali planimetricamente indeformabili e da traversi inestensibili, riduce ulteriormente il numero di elementi del vettore azioni nodali. Si osserva infatti che all'equilibrio di un traverso partecipano contemporaneamente tutte le forze orizzontali applicate ai nodi che ad esso appartengono. Il vettore azioni nodali conterrà quindi direttamente la loro somma e sarà costituito da

$$\mathbf{R} = \begin{bmatrix} \mathbf{T} \\ \mathbf{R}'_1 \\ \cdots \\ \cdots \\ \mathbf{R}'_n \end{bmatrix} \tag{3.8}$$

avendo indicato con $\mathbf{T}$ il vettore che contiene le somme delle forze orizzontali applicate ai nodi dei traversi, nell'ordine a questi attribuito, e con $\mathbf{R}'_1 \ldots \mathbf{R}'_n$ le componenti di azioni nodali non corrispondenti a vincoli esterni o mutui.

Così, per lo schema di figura 3.8 si indica con T_1 la somma delle forze orizzontali F_{x1}, F_{x2}, F_{x3} e si ha

$$\mathbf{R} = \begin{bmatrix} T_1 \\ M_{z1} \\ F_{y2} \\ M_{z2} \\ F_{y3} \\ M_{z3} \end{bmatrix}$$

Le azioni racchiuse nel vettore definito dalla (3.8) possono essere sia applicate ai nodi direttamente dall'esterno che trasmesse ad essi dalle aste. Nel seguito si utilizzerà il simbolo $\mathbf{R}$ per indicare l'azione totale esercitata,

ed i simboli $\mathbf{R}_e$ ed $\mathbf{R}_a$ per indicare rispettivamente le azioni esterne e quelle derivanti dalle aste. Si ha pertanto

$$\mathbf{R} = \mathbf{R}_e + \mathbf{R}_a \tag{3.9}$$

6. Relazione tra azioni nodali ed azioni agli estremi delle aste

Le componenti delle azioni applicate al generico nodo i, contenute nel vettore $\mathbf{R}_i$, possono essere assorbite direttamente da un vincolo oppure contribuire ad una delle componenti delle azioni nodali $\mathbf{R}$. Il contributo di $\mathbf{R}_i$ ad $\mathbf{R}$ è fornito da una relazione lineare che può essere espressa mediante una matrice. Si può facilmente constatare che tale matrice è la trasposta della matrice di corrispondenza $\mathbf{a}_i$ definita nel paragrafo 4. Si può scrivere quindi

$$\mathbf{R}' = \mathbf{a}_i^T \, \mathbf{R}_i \tag{3.10}$$

Al primo membro di questa relazione si è utilizzato il simbolo $\mathbf{R}'$ (anziché $\mathbf{R}$), per evidenziare che le azioni fornite dal prodotto $\mathbf{a}_i^T \, \mathbf{R}$ sono solo un'aliquota delle azioni nodali $\mathbf{R}$.

Se si applica la (3.10) al nodo 1 dello schema di figura 3.8, per il quale

$$\mathbf{R} = \begin{bmatrix} T_1 \\ M_{z1} \\ F_{y2} \\ M_{z2} \\ F_{y3} \\ M_{z3} \end{bmatrix} \qquad \mathbf{a}_1^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{R}_1 = \begin{bmatrix} F_{x1} \\ F_{y1} \\ M_{z1} \end{bmatrix}$$

si constata che F_{x1} ed M_{z1} contribuiscono rispettivamente alla prima ed alla seconda delle azioni nodali, mentre F_{y1} non dà alcun contributo perché assorbita direttamente dal carrello.

Le azioni esercitate agli estremi delle aste sono sempre azioni mutue tra asta e nodo; quelle esplicate dal nodo sull'asta sono quindi uguali in valore e di verso opposto rispetto alle azioni esercitate dall'asta sul nodo. Se si indica con $\mathbf{R}_{aji}$ l'azione esercitata dall'asta j sul nodo i, per un'asta di estremi i e k si ha

$$\mathbf{S}_j = - \begin{bmatrix} \mathbf{R}_{aji} \\ \mathbf{R}_{ajk} \end{bmatrix}$$

nella quale il segno meno sta proprio ad indicare la differenza di segno tra azioni dell'asta sul nodo e del nodo sull'asta.

Il contributo dell'asta j alle azioni nodali $\mathbf{R}'_a$ è somma delle aliquote dovute a $\mathbf{R}_{aji}$ ed $\mathbf{R}_{ajk}$, pari rispettivamente ad $\mathbf{a}_i^T\ \mathbf{R}_{aji}$ ed $\mathbf{a}_k^T\ \mathbf{R}_{ajk}$, ed è quindi fornito dalla relazione

$$\mathbf{R}'_a = -\left[\mathbf{a}_i^T\ \mathbf{a}_k^T\right] \mathbf{S}_j$$

nella quale si indica con $\left[\mathbf{a}_i^T\ \mathbf{a}_k^T\right]$ la matrice ottenuta unendo in sequenza le due matrici $\mathbf{a}_i^T$ ed $\mathbf{a}_k^T$.

Il contributo complessivo di tutte le aste è pari alla somma dei singoli contributi, e pertanto

$$\mathbf{R}_a = -\mathbf{a}^T\ \mathbf{S} \tag{3.11}$$

essendo $\mathbf{a}^T$ la matrice contenente le trasposte delle coppie $\mathbf{a}_i\ \mathbf{a}_k$ relative a tutte le aste e quindi essa stessa la trasposta della matrice a definita nel paragrafo 4.

Questa relazione si sarebbe potuta ricavare immediatamente dalla (3.7) per il principio di dualità dimostrato nel capitolo precedente.

7. Relazione tra carichi, azioni nodali e spostamenti nodali

La relazione (3.3) mostrata nel paragrafo 2 indica che le azioni $\mathbf{S}$ esercitate dai nodi alle estremità delle aste sono somma di una aliquota che insorge per effetto dei carichi interni alle aste quando il movimento dei loro estremi è impedito ed un'altra linearmente proporzionale agli spostamenti di tali punti. Accoppiando ad essa le espressioni (3.7) e (3.11) è possibile trovare una analoga relazione tra azioni nodali esercitate dalle aste, carichi interni e spostamenti nodali. Si ha

$$\mathbf{R}_a = -\mathbf{a}^T\ \mathbf{S} = -\mathbf{a}^T\ \overline{\mathbf{S}} - \mathbf{a}^T\ \mathbf{k}\ \mathbf{a}\ \mathbf{r}$$

e quindi sinteticamente

$$\mathbf{R}_a = \overline{\mathbf{R}}_a - \mathbf{K}\ \mathbf{r} \tag{3.12}$$

avendo indicato con $\overline{\mathbf{R}}_a$ il *vettore azioni nodali di incastro perfetto*, cioè l'insieme di azioni che vengono trasmesse dalle aste ai nodi bloccati in conseguenza ai carichi interni alle aste

$$\overline{\mathbf{R}}_a = -\mathbf{a}^T \overline{\mathbf{S}} \tag{3.13}$$

e con **K** la matrice di proporzionalità lineare tra spostamenti nodali ed azioni nodali che viene denominata *matrice di rigidezza della struttura*

$$\mathbf{K} = \mathbf{a}^T \, \mathbf{k} \, \mathbf{a} \tag{3.14}$$

In forma esplicita, per uno schema con h componenti di movimento incognite la (3.12) diviene

$$\begin{bmatrix} R_{a1} \\ \dots \\ R_{ap} \\ \dots \\ \dots \\ R_{ah} \end{bmatrix} = \begin{bmatrix} \overline{R}_{a1} \\ \dots \\ \overline{R}_{ap} \\ \dots \\ \dots \\ \overline{R}_{ah} \end{bmatrix} - \begin{bmatrix} K_{11} & \dots & \dots & K_{1q} & \dots & K_{1h} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ K_{p1} & \dots & \dots & K_{pq} & \dots & K_{ph} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ K_{h1} & \dots & \dots & K_{hq} & \dots & K_{hh} \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ \dots \\ \dots \\ r_q \\ \dots \\ r_h \end{bmatrix}$$

e la generica riga p si scrive

$$R_{ap} = \overline{R}_{ap} - (K_{p1} \; r_1 + \dots + K_{pq} \; r_q + \dots + K_{ph} \; r_h)$$

Il termine K_{pq} della matrice di rigidezza rappresenta quindi l'azione p–esima esercitata dal nodo alle aste (e, cambiando il segno, dalle aste al nodo) per effetto dello spostamento q–esimo unitario, in assenza di carichi e con tutti gli altri spostamenti nulli. Esso racchiude i contributi di tutte le aste che sono in grado di esercitare l'azione p in conseguenza allo spostamento q, ovvero delle aste che collegano i nodi cui si riferiscono l'azione e lo spostamento. Se questi sono relativi ad uno stesso nodo, i contributi sono forniti da tutte le aste che convergono in esso; se sono relativi a due nodi non collegati direttamente da aste, il termine K_{pq} è nullo.

In una qualsiasi struttura, ciascun nodo è collegato solo a pochi altri nodi. Di conseguenza i termini della matrice di rigidezza non nulli sono pochi (rispetto al numero totale di termini) e la loro posizione dipende dalla numerazione delle incognite. Il criterio assunto nel paragrafo 3 è basato sostanzialmente sulla numerazione dei nodi. Di conseguenza, se i nodi di una struttura sono conteggiati in modo che per ogni asta i nodi di estremità abbiano numeri prossimi tra loro, i termini non nulli della matrice di rigidezza si troveranno racchiusi in una ristretta banda a cavallo della diagonale principale (fig.3.11). Si denomina *ampiezza di banda* il massimo numero di termini di una riga racchiusi tra il primo e l'ultimo non nullo (estremi compresi). Con *ampiezza della semibanda* si indica invece il numero di termini compreso tra la diagonale principale e l'ultimo (o il primo) termine non nullo.

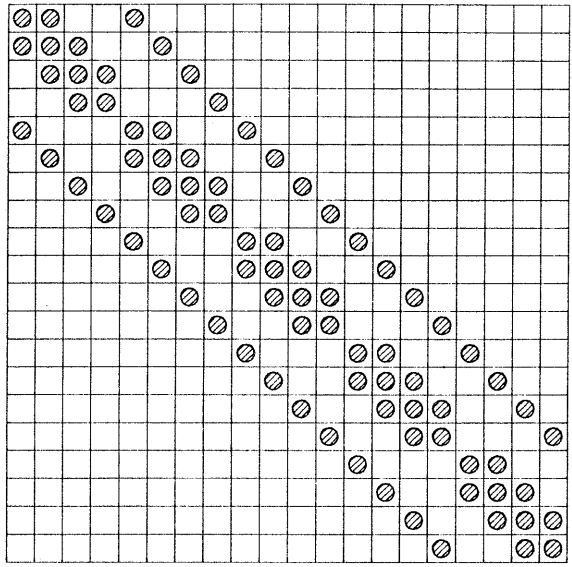


Fig. 3.11 — Matrice di rigidezza a banda

La presenza di traversi inestensibili comporta una riorganizzazione delle incognite, effettuata ponendo innanzi a tutte gli spostamenti di tali traversi. In conseguenza a ciò la matrice di rigidezza presenta ulteriori termini non nulli in due fasce laterali, lungo il bordo sinistro e quello superiore (fig.3.12), e a rigore non può più essere denominata matrice a banda. Se però si considera il vettore spostamenti nodali idealmente diviso in due gruppi, $\mathbf{r}_t$ ed $\mathbf{r}_a$

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{r}_a \end{bmatrix}$$

che contengono rispettivamente gli spostamenti dei traversi e tutte le altre incognite, e corrispondentemente la matrice di rigidezza scomposta in quattro sottomatrici, $\mathbf{K}^{tt}$, $\mathbf{K}^{ta}$, $\mathbf{K}^{at}$ e $\mathbf{K}^{aa}$, si può notare che mentre le prime tre sono sostanzialmente matrici piene (cioè con termini non nulli disposti un po' ovunque), la $\mathbf{K}^{aa}$ è inequivocabilmente una matrice a banda.

La matrice di rigidezza della struttura risulta inoltre essere:

- quadrata, cioè caratterizzata da un ugual numero di righe e di colonne, perché il numero di spostamenti nodali è esattamente pari al numero di azioni nodali.
- non singolare, e quindi invertibile, purché lo schema non sia labile.

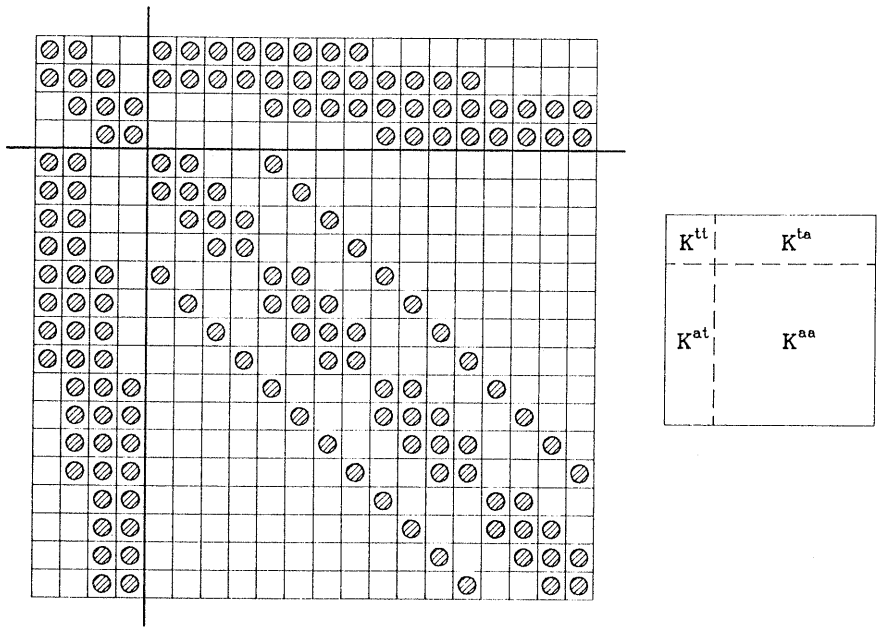


Fig. 3.12 — Scomposizione della matrice di rigidezza nel caso di vincoli mutui tra i nodi (traversi inestensibili)

La relazione lineare $\mathbf{K}\ \mathbf{r}$ associa a ciascun insieme di spostamenti un insieme di azioni, ed in particolare al vettore spostamenti nullo il vettore azioni nullo; d'altro canto, per il principio di unicità della soluzione del problema dell'equilibrio elastico, ad ogni insieme di azioni deve corrispondere un solo insieme spostamenti, e la relazione è pertanto biunivoca ed invertibile.

- simmetrica rispetto alla diagonale principale, cioè tale che $K_{pq} = K_{qp}$. Questo può essere ricavato immediatamente dal teorema di Betti; applicando infatti alla struttura gli spostamenti r_p e r_q unitari, il lavoro di K_{pq} (azione provocata in p da r_q) per r_p deve essere uguale al lavoro di K_{qp} (provocata in q da r_p) per r_q, ed essendo uguali gli spostamenti anche le due azioni devono avere uguale entità. La simmetria di $\mathbf{K}$ può essere dimostrata anche a partire dalla sua definizione (3.14).
  Si ha infatti
$$\mathbf{K}^T = (\mathbf{a}^T\ \mathbf{k}\ \mathbf{a})^T = \mathbf{a}^T\ \mathbf{k}^T\ \mathbf{a}$$
  e poiché $\mathbf{k}^T = \mathbf{k}$ deve essere anche $\mathbf{K}^T = \mathbf{K}$.
- definita positiva, cioè tale che per ogni $\mathbf{r}$ non nullo la forma quadratica

$(\mathbf{K}\,\mathbf{r})^T\,\mathbf{r}$ è maggiore di zero. Ciò deriva immediatamente dal fatto che il prodotto anzidetto rappresenta il lavoro di un insieme di azioni nodali $\mathbf{K}\,\mathbf{r}$ per gli spostamenti corrispondenti, che deve essere necessariamente positivo.

8. Equilibrio nodale e risoluzione del problema

Per l'equilibrio dei nodi, o dei traversi inestensibili, è necessario che siano nulle le azioni complessive su essi agenti, somma di un'aliquota $\mathbf{R}_e$ applicata direttamente ed un'altra $\mathbf{R}_a$ trasmessa dalle aste. Utilizzando la (3.12) la condizione di equilibrio si scrive

$$\mathbf{R}_e + \overline{\mathbf{R}}_a - \mathbf{K}\,\mathbf{r} = 0$$

ovvero

$$\overline{\mathbf{R}} = \mathbf{K}\,\mathbf{r} \tag{3.15}$$

avendo indicato con

$$\overline{\mathbf{R}} = \mathbf{R}_e + \overline{\mathbf{R}}_a \tag{3.16}$$

il vettore contenente le azioni complessive prodotte dai carichi esterni, sia applicati direttamente ai nodi che sulle aste, in presenza di vincoli che impediscono spostamenti e rotazioni dei nodi. La relazione (3.15) indica che poiché tali vincoli non sono in realtà presenti, i nodi devono subire dei cedimenti $\mathbf{r}$ di entità tale da generare un insieme di azioni $\mathbf{K}\,\mathbf{r}$ in grado di equilibrare le azioni esterne prima bilanciate dai vincoli. Questa spiegazione fisica della condizione di equilibrio è stata sfruttata in passato da numerosi procedimenti di risoluzione del problema iperstatico, che si indicano genericamente come metodi di rilassamento. Essi sono tutti basati sull'idea di rimuovere iterativamente un singolo vincolo, scrivendo per esso la condizione di equilibrio e ricavando lo spostamento, o la rotazione, necessari per rispettare tale condizione. Caratteristica di questi metodi è quindi la scrittura diretta delle equazioni di equilibrio e la loro risoluzione iterativa. L'impostazione seguita in questo testo porta ovviamente, a parità di ipotesi di partenza, allo stesso sistema di equazioni, ottenuto però per via matriciale. Coerentemente, la soluzione è fornita da

$$\mathbf{r} = \mathbf{K}^{-1}\overline{\mathbf{R}} \tag{3.17}$$

mediante l'inversione della matrice $\mathbf{K}$. Questa operazione può essere effettuata mediante procedimenti di calcolo numerico che richiedono un numero

finito, noto a priori, di operazioni, come mostrato nella seconda parte del testo. Più frequentemente, gli stessi procedimenti vengono utilizzati per effettuare direttamente la risoluzione del sistema senza passare attraverso l'inversione della matrice ed il successivo prodotto per il vettore azioni esterne. Nel caso di sistemi di ordine molto elevato è però ancora conveniente l'uso di procedimenti di risoluzione iterativi, sostanzialmente analoghi a quelli usati nelle impostazioni non matriciali.

Il procedimento da seguire per la risoluzione del problema iperstatico appare quindi concettualmente ben delineato. Occorre innanzitutto definire la matrice di rigidezza della struttura $\mathbf{K}$ ed il vettore $\overline{\mathbf{R}}$. Determinati mediante la (3.17) gli spostamenti nodali, è possibile risalire da essi alle componenti di movimento degli estremi delle aste nel riferimento globale mediante la (3.7). Utilizzando le relazioni presentate nel capitolo precedente, si passa, per ciascuna asta, da queste alle corrispondenti componenti nel sistema di riferimento locale ed infine alle caratteristiche di sollecitazione agli estremi e nei punti interni dell'asta.

9. Determinazione diretta della matrice di rigidezza

Le relazioni (3.13) e (3.14), che consentono di ricavare le azioni di incastro perfetto e la matrice di rigidezza della struttura a partire dalle corrispondenti entità relative all'insieme di aste, sono molto utili concettualmente, perché estremamente sintetiche. Esse si adattano invece meno bene alle applicazioni numeriche, perché forniscono $\overline{\mathbf{R}}_a$ e $\mathbf{K}$ mediante operazioni che coinvolgono matrici molto più grandi di quelle che si vogliono ottenere. Ad esempio, per lo schema di figura 3.8, che presenta 6 incognite, le matrici $\mathbf{a}$ e $\mathbf{k}$ sono costituite rispettivamente da 24×6 e 24×24 elementi, mentre $\overline{\mathbf{R}}_a$ e $\mathbf{K}$ sono di dimensioni 6×1 e 6×6.

Una modalità operativa più semplice scaturisce dal constatare che, essendo la matrice $\mathbf{a}$ costituita da soli 0 ed 1, i prodotti matriciali indicati hanno come unico effetto il posizionare in maniera opportuna gli elementi del vettore azioni di incastro e della matrice di rigidezza di ciascun'asta rispettivamente nel vettore e nella matrice della struttura.

Per meglio comprendere il significato di quanto ora detto, si consideri una struttura costituita da m aste ed n nodi, che presenta h spostamenti incogniti. Come spiegato in precedenza, la matrice $\mathbf{a}$ è composta da m blocchi, uno per asta, ciascuno di 6 righe ed h colonne (fig.3.13). Si indichino con r e c i numeri d'ordine (compresi tra 1 e 6) di due generiche componenti di movimento di estremità dell'asta j e con r' e c' le righe di $\mathbf{a}$

ad esse corrispondenti: $r' = 6\ (j-1) + r$, $c' = 6\ (j-1) + c$. Se la componente r è bloccata da un vincolo, tutti gli elementi della riga r' di $\mathbf{a}$ devono essere nulli. In caso contrario, la componente in esame deve corrispondere ad uno degli spostamenti nodali incogniti, che indicheremo con p. L'unico elemento non nullo della riga anzidetta sarà in tal caso quello in colonna p, $a_{r'p}$. Analogamente, se q è lo spostamento nodale corrispondente alla componente di movimento c l'unico elemento non nullo della riga c' sarà quello in colonna q, $a_{c'q}$. Quanto detto per le righe r' e c' di $\mathbf{a}$ vale ovviamente anche per le colonne r' e c' di $\mathbf{a}^T$, con esse coincidenti, che avranno come unici elementi non nulli $a^T_{pr'}$ e $a^T_{qc'}$.

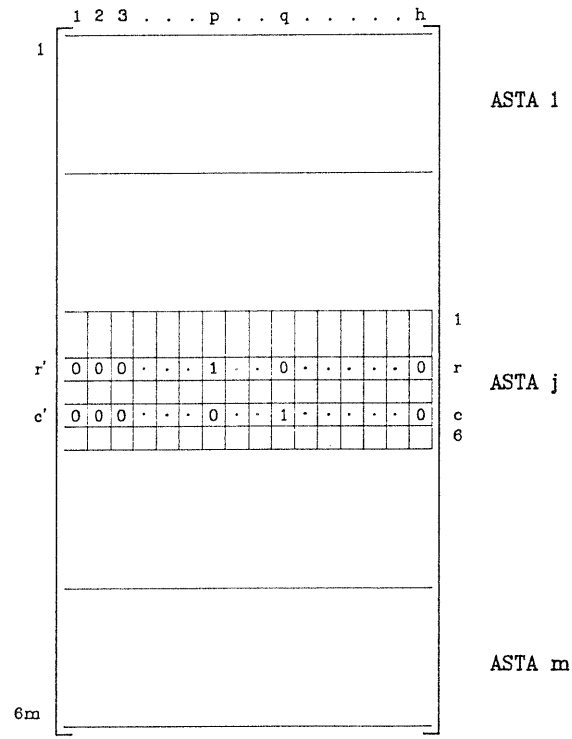


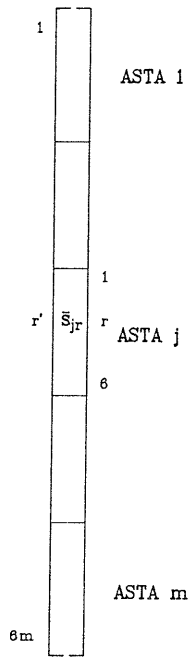
Fig. 3.13 — Matrice di congruenza $\mathbf{a}$

Anche il vettore $\overline{\mathbf{S}}$, costruito assemblando i vettori di incastro perfetto $\overline{\mathbf{S}}_j$ delle aste, è composto da m blocchi di 6 elementi; $\overline{S}_{r'}$ coincide con $\overline{S}_{jr}$ e rappresenta l'azione di incastro r–esima per l'asta j. Nell'effettuare il prodotto $\mathbf{a}^T\ \overline{\mathbf{S}}$ (fig.3.14), per ottenere a meno del segno il generico $\overline{R}_{ai}$ gli elementi della riga i di $\mathbf{a}^T$ vengono moltiplicati per i corrispondenti elementi di $\overline{\mathbf{S}}$. In tal modo, gli elementi della colonna r' vengono moltiplicati per

$$\overline{\mathbf{R}}_a = -\mathbf{a}^T \overline{\mathbf{S}}$$

$$\overline{R}_{a_i} = -\sum_{k=1}^{6m} a_{ik}^T \, \overline{S}_k$$

$$\overline{R}_{a_p} = -\sum_{k=1}^{6m} a_{pk}^T \, \overline{S}_k = \ldots - a_{pr'}^T \overline{S}_{r'} - \ldots$$



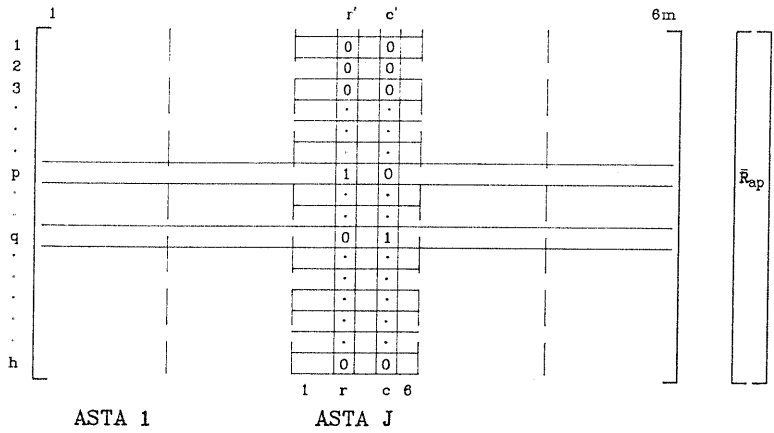


Fig. 3.14 — Prodotto matriciale $\mathbf{a}^T \ \overline{\mathbf{S}}$

$\overline{S}_{jr}$. Poiché essi sono tutti nulli, tranne al più $a_{pr'}^T$, $\overline{S}_{jr}$ contribuisce alla valutazione del solo termine $\overline{R}_{ap}$.

Analogamente, la matrice $\mathbf{k}$ relativa all'insieme di aste è stata costruita

disponendo lungo la diagonale le m matrici di rigidezza delle aste. Il suo elemento $k_{r'c'}$ coincide quindi con k_{jrc} della matrice di rigidezza dell'asta j nel riferimento globale. Nel valutare $\mathbf{K}$ mediante la (3.14), il primo prodotto matriciale, $\mathbf{a}^T\ \mathbf{k}$, fornisce una matrice $\mathbf{M}$ con un numero di righe pari a quello di $\mathbf{a}^T$ (e quindi h, quanti sono gli spostamenti incogniti) ed un numero di colonne pari a quello di k ($6m$). Nell'effettuare il prodotto vengono moltiplicati tra loro gli elementi corrispondenti di una riga di $\mathbf{a}^T$ e di una colonna di $\mathbf{k}$. Il termine $k_{r'c'} = k_{jrc}$ viene quindi moltiplicato per gli elementi della colonna r' di $\mathbf{a}^T$, tutti nulli ad eccezione di $a^T_{pr'}$; esso contribuisce pertanto solo all'elemento $M_{pc'}$ (fig.3.15). Col secondo prodotto matriciale, $\mathbf{M}\ \mathbf{a}$, il termine $M_{pc'}$ viene moltiplicato per gli elementi della riga c' di $\mathbf{a}$, anch'essi tutti nulli ad eccezione di $a_{c'q}$. Di conseguenza esso, e quindi k_{jrc}, contribuisce solo alla valutazione del termine K_{pq} (fig.3.16). Se lo spostamento r (o c) è impedito dai vincoli esterni la riga r' (o c') è tutta nulla e quindi il termine k_{jrc} non interviene nella matrice di rigidezza della struttura.

$$\mathbf{M} = \mathbf{a}^T\ \mathbf{k}$$

$$M_{i\ell} = \sum_{k=1}^{6m} a^T_{ik}\ k_{k\ell}$$

$$M_{pc'} = \sum_{k=1}^{6m} a^T_{pk}\ k_{kc'} =$$

$$= \ldots + a^T_{pr'}\ k_{r'c'} + \ldots$$

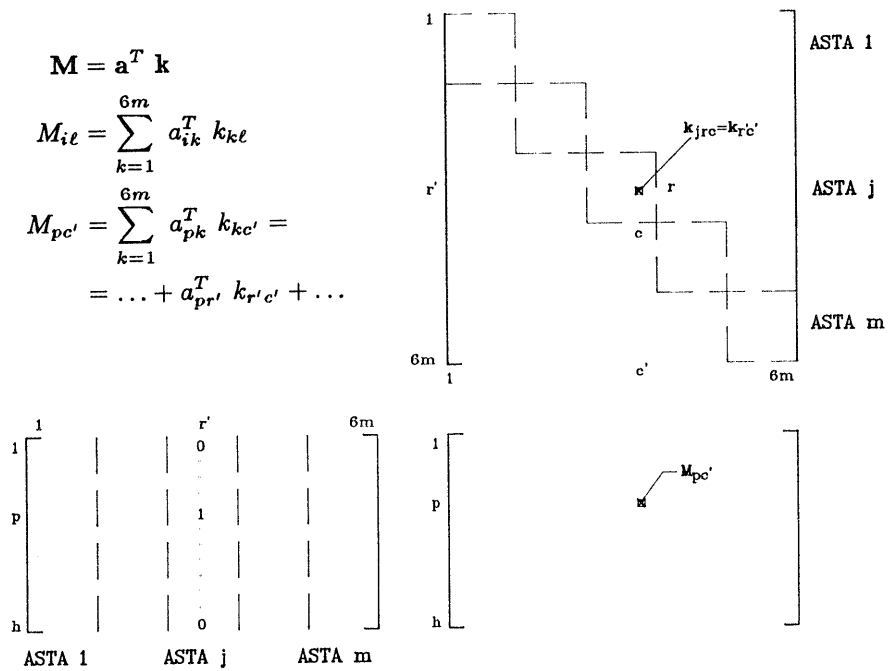


Fig. 3.15 — Prodotto matriciale $\mathbf{a}^T\ \mathbf{k}$

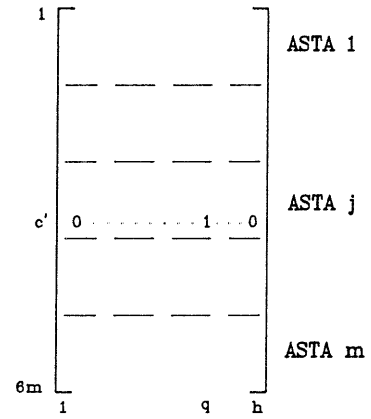
I posizionamenti effettuati mediante le (3.13) e (3.14) hanno un preciso significato fisico. Poiché l'azione r–esima esercitata agli estremi dell'asta j

$$\mathbf{K} = \mathbf{a}^T\ \mathbf{ka} = \mathbf{Ma}$$

$$K_{i\ell} = \sum_{k=1}^{6m} M_{ik}\ a_{k\ell}$$

$$K_{pq} = \sum_{k=1}^{6m} M_{pk}\ a_{kq} =$$

$$= \ldots + M_{pc'}\ a_{c'q} + \ldots$$



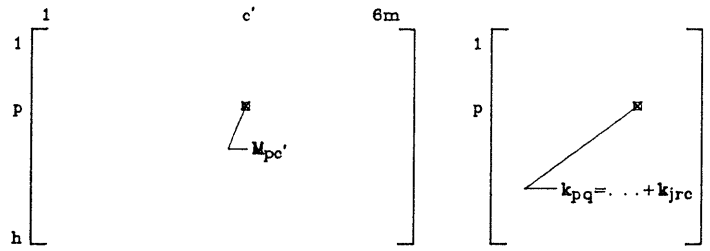


Fig. 3.16 — Prodotto matriciale $\mathbf{M}\ \mathbf{a}$

corrisponde all'azione nodale p–esima, il termine $\overline{S}_{jr}$ che indica tale azione sull'asta in assenza di movimento nodale rappresenta l'aliquota dovuta all'asta j dell'azione nodale d'incastro p–esima ed interviene quindi nella valutazione di $\overline{R}_{ap}$. Inoltre, poiché lo spostamento c–esimo dell'asta j coincide con lo spostamento nodale q–esimo, il termine k_{jrc} che indica l'azione r–esima agli estremi dell'asta j per effetto di uno spostamento c–esimo unitario degli estremi. rappresenta l'aliquota dovuta all'asta j dell'azione nodale p–esima conseguente ad uno spostamento q–esimo unitario ed interviene nella valutazione di K_{pq}.

Operativamente, si può evitare di costruire la matrice di congruenza $\mathbf{a}$, essendo ad essa equivalenti i vettori di corrispondenza $\mathbf{n}$ definiti nel paragrafo 4 insieme alla tabella di incidenza delle aste. Per determinare $\overline{\mathbf{R}}_a$, si possono passare in rassegna tutte le aste, con le relative azioni di incastro, valutando mediante la tabella di incidenza ed i vettori di corrispondenza il numero d'ordine p dello spostamento nodale associato alla componente di movimento r, che esiste solo se quest'ultima non è impedita da un vicolo. In tal caso l'azione $\overline{S}_{jr}$ viene aggiunta, con segno negativo, all'azione di incastro $\overline{R}_{ap}$. Per determinare $\mathbf{K}$, si possono passare in rassegna tutte le

aste, con le relative matrici di rigidezza nel riferimento globale, determinare per ciascuna coppia di azioni e componenti di movimento di estremità r e c le corrispondenti azioni e spostamenti nodali p e q e, se queste esistono, aggiungere il valore di k_{jrc} al termine K_{pq} della matrice di rigidezza della struttura. Questo modo di procedere è detto *metodo di determinazione diretta della matrice di rigidezza della struttura* o, con termine inglese, *direct stiffness method.*

CAPITOLO QUARTO

L'INSIEME SPAZIALE DI TELAI PIANI

1. Macroelementi e sottostrutture

Nel primo capitolo si è definita struttura un insieme di nodi rigidi collegati da elementi deformabili. Si è quindi preso in esame in particolare l'elemento asta, monodimensionale, con le sue numerose varianti (asta con cerniere, trave su suolo elastico alla Winkler, asta con tratti rigidi di estremità coassiali e non coassiali). Più in generale l'analisi matriciale può prevedere una vasta gamma di elementi: elementi bidimensionali piani, triangolari o quadrilateri, soggetti ad azioni e deformazioni nel piano (lastre) o fuori piano (piastre); elementi bidimensionali spaziali, come quadrilateri sghembi o gusci; elementi tridimensionali. L'importante è conoscere per ciascuno di essi la relazione tra le azioni applicate nei punti di estremità e le corrispondenti componenti di movimento (*matrice di rigidezza dell'elemento*) e le azioni di estremità che insorgono per effetto dei carichi interni all'elemento quando i suoi estremi sono bloccati (*azioni di incastro perfetto*).

Andando ancora avanti nella stessa ottica, si può sentire il bisogno di elementi ancora più complessi. Ad esempio, nell'analizzare un edificio in muratura farebbe comodo un elemento capace di rappresentare un'intera parete, con le aperture in essa presenti, dotato di un elevato numero di nodi di estremità per consentirne una buona connessione con altri elementi analoghi (fig. 4.1). Nello studiare i problemi di interazione tra struttura

in elevazione, trave di fondazione e terreno sarebbe utile poter considerare l'intera struttura in elevazione come un unico elemento, in cui i nodi di estremità sono rappresentati dai punti da collegare con la fondazione (fig.4.2). Nel calcolare uno schema spaziale costituito da telai piani collegati da impalcati indeformabili sarebbe comodo considerare come elemento base il singolo telaio piano, nel quale i nodi da connettere ad altri elementi sono costituiti dai traversi inestensibili dotati ciascuno di un solo grado di libertà (fig.4.3).

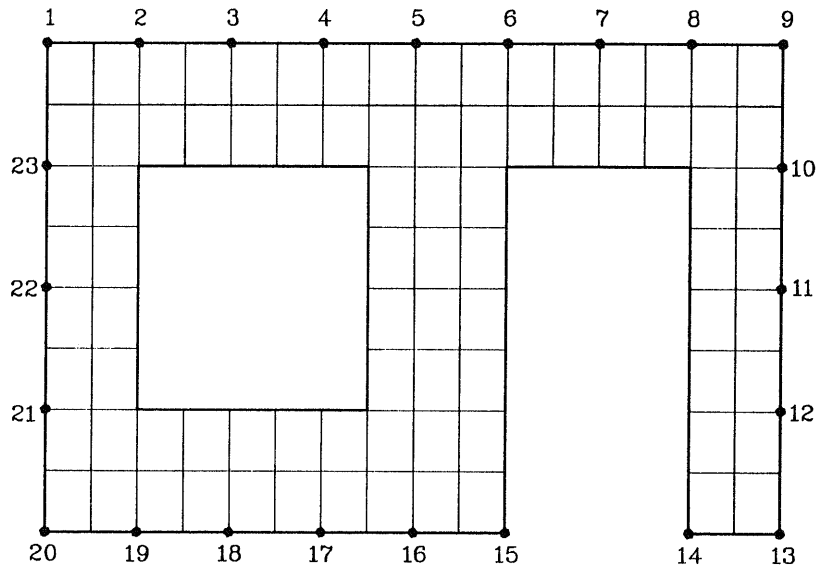


Fig. 4.1 — Macroelemento che rappresenta la parete di un edificio in muratura

Elementi tanto complessi possono essere denominati *macroelementi*. La determinazione della matrice di rigidezza e delle azioni di incastro perfetto potrebbe essere effettuata anche per essi mediante lo studio della meccanica del continuo. Così ad esempio la parete dell'edificio in muratura potrebbe ricadere nell'ambito della trattazione teorica delle lastre forate. In genere però questa via è molto complicata, mentre risulta sicuramente più agevole valutare il comportamento del macroelemento scomponendolo in parti più semplici: la parete in elementi lastra (o piastra, a seconda del carico), il telaio in elementi trave. In tal modo esso va visto a sua volta come una struttura e può essere esaminato mediante algoritmi dell'algebra lineare; lo si indica quindi più propriamente col termine *sottostruttura*, per evidenziarne contemporaneamente sia la composizione che l'appartenenza ad un insieme più complesso.

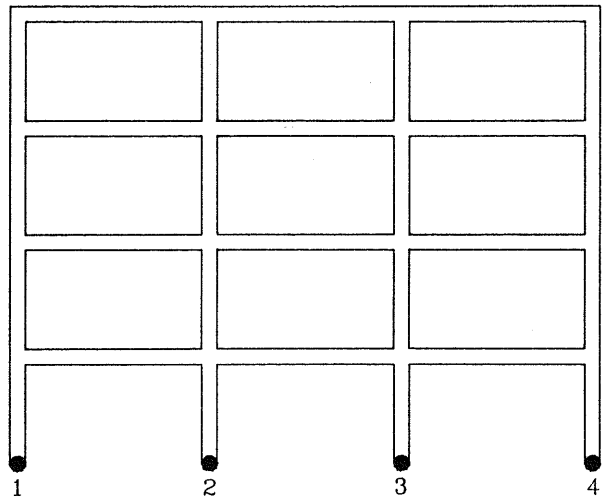


Fig. 4.2 — Macroelemento che rappresenta un telaio piano da connettere alla trave di fondazione

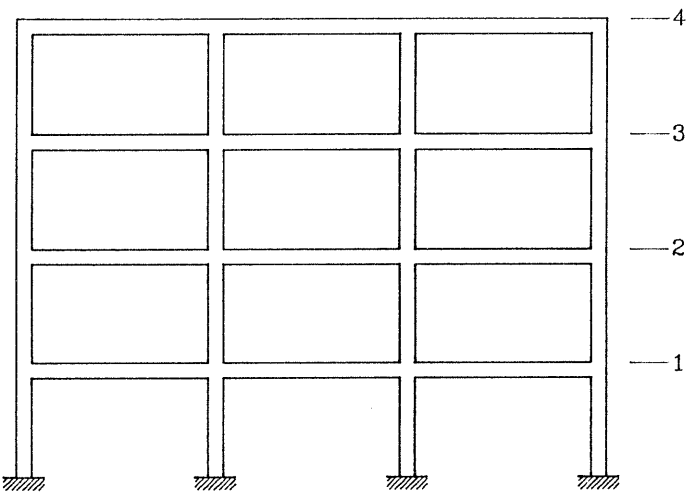


Fig. 4.3 — Macroelemento che rappresenta un telaio piano da connettere ad altri telai in corrispondenza dei traversi

Ribadendo quanto innanzi esposto, si definisce sottostruttura un insieme di nodi rigidi collegati da elementi deformabili del quale interessa mettere in evidenza la relazione tra azioni e componenti di movimento di un insieme ridotto di punti nodali e l'entità delle azioni che insorgono in tali punti per effetto dei carichi quando essi (solo essi e non anche gli altri nodi) sono bloccati, in modo da poter considerare questo insieme come macroelemento

in una struttura più complessa. Nei paragrafi che seguono viene mostrato come ricavare queste informazioni a partire dalla matrice di rigidezza e dal vettore azioni di incastro dell'insieme, operazione indicata col termine *condensazione* perché condensa tale matrice e vettore in un'altro di dimensioni minori. Questo modo di procedere viene poi particolarizzato per il caso di telaio piano visto come costituente di un insieme spaziale di telai, mostrando anche come procedere all'assemblaggio delle informazioni relative a ciascun telaio per ottenere la matrice di rigidezza ed il vettore azioni di incastro dell'insieme spaziale.

2. Condensazione della matrice di rigidezza e del vettore azioni di incastro

Nel capitolo precedente si è ricavata la relazione generale che fornisce le azioni $\mathbf{R}$ sui nodi di una struttura come somma di un'aliquota $\overline{\mathbf{R}}$ dovuta alle azioni ad essi direttamente applicate ed ai carichi sugli elementi, in assenza di movimento dei nodi, ed un'altra dovuta alle componenti di movimento nodale $\mathbf{r}$ e linearmente dipendente da queste mediante la matrice di rigidezza $\mathbf{K}$

$$\mathbf{R} = \overline{\mathbf{R}} - \mathbf{K}\,\mathbf{r} \tag{4.1}$$

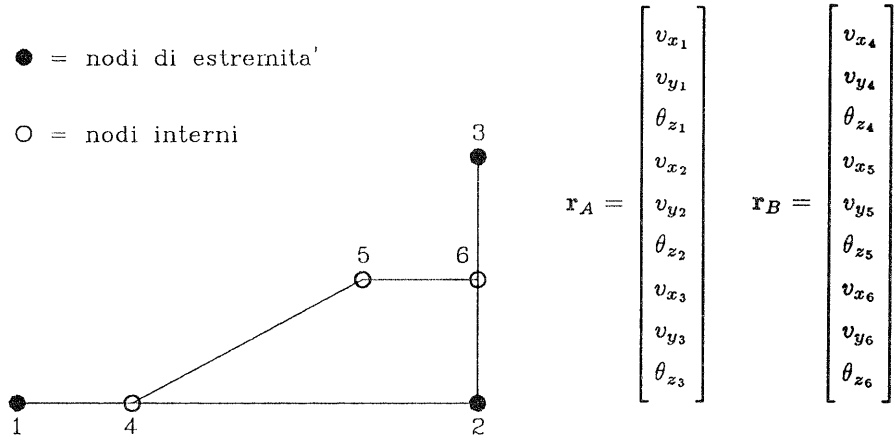


Fig. 4.4 — Sottostruttura che rappresenta il complesso costituito da trave a ginocchio, trave di piano e pilastro

Si prende ora in esame una generica sottostruttura, ad esempio l'insieme di aste mostrato in figura 4.4. Le sue componenti di movimento nodale possono essere divise in due gruppi, A e B, a seconda che esse appartengono rispettivamente ai nodi di estremità del macroelemento o ai nodi interni.

Nel numerare tali componenti è sempre possibile individuare prima quelle appartenenti ad A e poi quelle di B, in modo da poter considerare il vettore $\mathbf{r}$ come diviso nei due sottoinsiemi $\mathbf{r}_A$ ed $\mathbf{r}_B$. In maniera analoga il vettore azioni nodali risulta scomposto nei vettori $\mathbf{R}_A$ ed $\mathbf{R}_B$ e la matrice di rigidezza nelle quattro sottomatrici $\mathbf{K}_{AA}$, $\mathbf{K}_{AB}$, $\mathbf{K}_{BA}$, $\mathbf{K}_{BB}$. La relazione (4.1) può in tal caso scriversi

$$\begin{bmatrix} \mathbf{R}_A \\ \mathbf{R}_B \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{R}}_A \\ \overline{\mathbf{R}}_B \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{r}_A \\ \mathbf{r}_B \end{bmatrix} \tag{4.2}$$

ovvero

$$\begin{aligned} \mathbf{R}_A &= \overline{\mathbf{R}}_A - \mathbf{K}_{AA}\ \mathbf{r}_A - \mathbf{K}_{AB}\ \mathbf{r}_B \\ \mathbf{R}_B &= \overline{\mathbf{R}}_B - \mathbf{K}_{BA}\ \mathbf{r}_A - \mathbf{K}_{BB}\ \mathbf{r}_B \end{aligned} \tag{4.3}$$

In analogia a quanto fatto per la singola asta, si possono indicare con $\mathbf{S}$ le azioni trasmesse al macroelemento nei nodi di estremità e con $\mathbf{s}$ le componenti di movimento di questi nodi.

Si valuta innanzitutto l'entità delle azioni di estremità che nascono in conseguenza al movimento dei nodi di estremità ($\mathbf{r}_A = \mathbf{s}$) in assenza di carichi ($\overline{\mathbf{R}}_A = \overline{\mathbf{R}}_B = 0$). Le azioni $\mathbf{R}_A$ sui nodi esterni sono uguali ed opposte a quelle trasmesse al macroelemento ($\mathbf{R}_A = -\mathbf{S}$) mentre quelle sui nodi interni devono essere nulle perché questi possano essere in equilibrio ($\mathbf{R}_B = 0$). Dalle (4.3) si ha

$$\begin{aligned} -\mathbf{S} &= -\mathbf{K}_{AA}\ \mathbf{s} - \mathbf{K}_{AB}\ \mathbf{r}_B \\ 0 &= -\mathbf{K}_{BA}\ \mathbf{s} - \mathbf{K}_{BB}\ \mathbf{r}_B \end{aligned}$$

e ricavando $\mathbf{r}_B$ dalla seconda relazione e sostituendolo nella prima si ottiene la relazione lineare

$$\mathbf{S} = (\mathbf{K}_{AA} - \mathbf{K}_{AB}\ \mathbf{K}_{BB}^{-1}\ \mathbf{K}_{BA})\ \mathbf{s}$$

ovvero

$$\mathbf{S} = \mathbf{k}\ \mathbf{s}$$

dove

$$\mathbf{k} = \mathbf{K}_{AA} - \mathbf{K}_{AB}\ \mathbf{K}_{BB}^{-1}\ \mathbf{K}_{BA} \tag{4.4}$$

rappresenta la *matrice di rigidezza condensata* della sottostruttura. Il significato fisico di questa relazione può essere compreso osservando che la sottomatrice di rigidezza $\mathbf{K}_{AA}$ fornisce le azioni di estremità conseguenti

ai movimenti dei nodi di estremità quando i nodi interni sono bloccati, mentre $\mathbf{k}$ fornisce le azioni che insorgono quando questi ultimi sono liberi. Il termine $-\mathbf{K}_{AB}\ \mathbf{K}_{BB}^{-1}\ \mathbf{K}_{BA}$ deve quindi rappresentare la variazione di rigidezza, ovvero la variazione delle azioni dei nodi di estremità conseguente al movimento dei nodi interni.

Si considera poi ciò che avviene per effetto dei carichi applicati ai nodi interni o agli elementi della sottostruttura quando i nodi di estremità sono bloccati ($\mathbf{r}_A = 0$), mentre quelli interni sono liberi di spostarsi e ruotare. Per l'equilibrio dei nodi interni deve essere anche in questo caso $\mathbf{R}_B = 0$, mentre le azioni sui nodi di estremità sono uguali ed opposte alle azioni di incastro perfetto $\overline{\mathbf{S}}$ della sottostruttura, o *azioni di incastro condensate* ($\mathbf{R}_A = -\overline{\mathbf{S}}$). Dalle (4.3) si ha

$$-\overline{\mathbf{S}} = \overline{\mathbf{R}}_A - \mathbf{K}_{AB}\ \mathbf{r}_B$$
$$0 = \overline{\mathbf{R}}_B - \mathbf{K}_{BB}\ \mathbf{r}_B$$

ed operando come nel caso precedente si ottiene l'espressione del vettore azioni di incastro condensate della sottostruttura

$$\overline{\mathbf{S}} = -\overline{\mathbf{R}}_A + \mathbf{K}_{AB}\ \mathbf{K}_{BB}^{-1}\ \overline{\mathbf{R}}_B \tag{4.5}$$

La relazione complessiva tra azioni di estremità, componenti di movimento e carichi è infine fornita, per il macroelemento come già per la singola asta, dalla relazione

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{k}\ \mathbf{s} \tag{4.6}$$

Mediante le (4.4) e (4.5) è possibile ricavare tutte le informazioni relative al macroelemento a partire da quelle relative allo stesso visto come una struttura autonoma, determinabili seguendo l'impostazione illustrata nel capitolo precedente. La semplicità formale delle espressioni non mette però in evidenza la complessità operativa, costituita soprattutto dalla necessità di invertire la matrice $\mathbf{K}_{BB}$ che è di ordine tanto maggiore quanto più numerosi sono i nodi interni della sottostruttura. Lo stesso risultato può ottenersi con minor onere computazionale utilizzando il procedimento di riduzione Gaussiana descritto in dettaglio nel terzo capitolo della seconda parte del testo. La relazione matriciale (4.2) può essere riscritta per la sottostruttura evidenziando che le azioni sui nodi esterni sono opposte a quelle trasmesse da questi al macroelemento ($\mathbf{R}_A = -\mathbf{S}$) mentre quelle sui nodi interni sono nulle ($\mathbf{R}_B = 0$)

$$\begin{bmatrix} -\mathbf{S} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{R}}_A \\ \overline{\mathbf{R}}_B \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{r}_B \end{bmatrix} \tag{4.7}$$

Questa relazione è in effetti costituita da un insieme di equazioni in numero pari al totale delle componenti di movimento dei nodi (interni o di estremità). La riduzione Gaussiana consiste nell'effettuare combinazioni lineari delle equazioni in maniera tale da azzerare parte dei coefficienti della matrice $\mathbf{K}$ e trasformare questa in una matrice triangolare (cioè nella quale sono nulli tutti i termini al di sopra o al di sotto della diagonale principale). Nel caso in esame la triangolarizzazione può essere arrestata quando si è azzerata la matrice $\mathbf{K}_{AB}$ ed il triangolo superiore della $\mathbf{K}_{BB}$. Si può constatare che nel far ciò il vettore al primo membro della (4.7) non viene modificato, e la relazione matriciale diventa quindi

$$\begin{bmatrix} -\mathbf{S} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{R}}'_A \\ \overline{\mathbf{R}}'_B \end{bmatrix} - \begin{bmatrix} \mathbf{K}'_{AA} & 0 \\ \mathbf{K}'_{BA} & \mathbf{K}'_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{r}_B \end{bmatrix}$$

ovvero

$$\mathbf{S} = -\overline{\mathbf{R}}'_A + \mathbf{K}'_{AA}\,\mathbf{s}$$

$$0 = \overline{\mathbf{R}}'_B - \mathbf{K}'_{BA}\,\mathbf{s} - \mathbf{K}'_{BB}\,\mathbf{r}_B$$

La prima espressione confrontata con la (4.6) mostra che il vettore $\overline{\mathbf{R}}'_A$ (a meno del segno) e la matrice $\mathbf{K}'_{AA}$ forniti dalla riduzione rappresentano proprio il vettore azioni di incastro e la matrice di rigidezza del macroelemento

$$\begin{aligned} \overline{\mathbf{S}} &= -\overline{\mathbf{R}}'_A \\ \mathbf{k} &= \mathbf{K}'_{AA} \end{aligned} \tag{4.8}$$

La seconda espressione può anche scriversi

$$\mathbf{K}'_{BB}\,\mathbf{r}_B = \overline{\mathbf{R}}'_B - \mathbf{K}'_{BA}\,\mathbf{s} \tag{4.9}$$

e da essa è possibile ricavare le componenti di movimento dei nodi interni $\mathbf{r}_B$ in funzione dei carichi e delle componenti di movimento di estremità, mediante il procedimento di sostituzione all'indietro descritto nella seconda parte del libro.

3. Matrice di rigidezza traslante di un telaio piano

Una sottostruttura particolarmente interessante ai fini applicativi è il telaio piano, visto come costituente di un insieme spaziale di telai. Esso

viene descritto utilizzando un sistema di riferimento locale $x'y'$, con l'asse x' orizzontale ed orientato verso destra, l'asse y' verticale ed orientato verso l'alto (fig.4.5). I nodi di estremità sono costituiti dai traversi inestensibili, che nella trattazione qui svolta verranno per semplicità supposti orizzontali. Ciascuno di essi è dotato di un unico grado di libertà, la traslazione nella direzione del traverso. I vettori azioni di estremità $\mathbf{S}$ e componenti di movimento di estremità $\mathbf{s}$ racchiudono quindi rispettivamente le forze orizzontali applicate ai traversi e gli spostamenti orizzontali di questi.

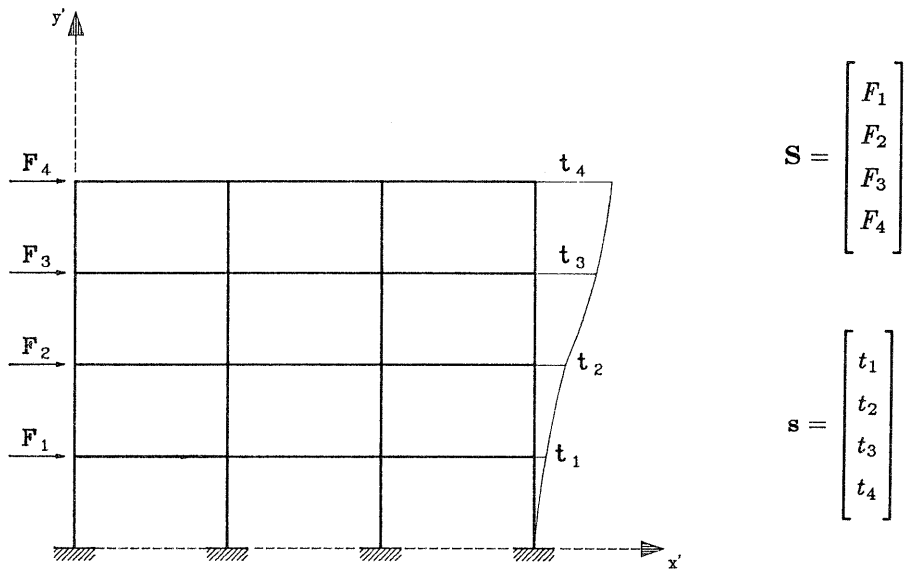


Fig. 4.5 — Sistema di riferimento locale per un telaio piano; azioni e componenti di movimento di estremità

Il criterio di numerazione delle incognite adottato nel capitolo precedente rispetta già il requisito di scindere le componenti di movimento in due gruppi, $\mathbf{r}_t$ ed $\mathbf{r}_a$, che contengono rispettivamente le traslazioni dei traversi e le componenti di movimento dei nodi interni (fig. 4.6). Corrispondentemente la matrice di rigidezza globale del telaio è stata suddivisa nelle quattro sottomatrici $\mathbf{K}^{tt}$, $\mathbf{K}^{ta}$, $\mathbf{K}^{at}$, $\mathbf{K}^{aa}$.

Mediante le operazioni matriciali indicate dalla (4.4) o col procedimento di riduzione Gaussiana che conduce alla (4.8) si determina la matrice di rigidezza $\mathbf{k}$ della sottostruttura; essa esprime la relazione lineare esistente tra forze orizzontali e traslazioni dei traversi e per tale motivo è denominata matrice di rigidezza traslante del telaio piano. Il generico termine k_{ij} di questa matrice rappresenta l'azione orizzontale F_i conseguente alla

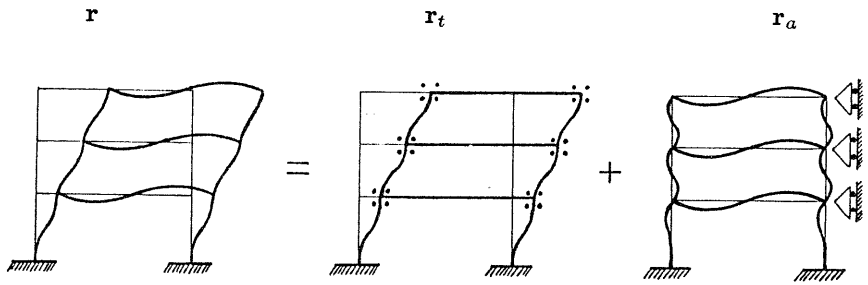


Fig. 4.6 — Componenti di movimento di estremità ed interne

traslazione t_j unitaria, quando gli spostamenti orizzontali degli altri traversi sono impediti (fig. 4.7 a). La matrice di rigidezza traslante, analogamente a quella globale della struttura, è simmetrica, definita positiva e non singolare. La sua inversa è denominata matrice di flessibilità traslante **f** ed il suo generico termine f_{ij} rappresenta lo spostamento t_i conseguente all'azione orizzontale F_j unitaria, in assenza di altre azioni applicate al telaio (fig. 4.7 b). Sfruttando il significato fisico dei termini della matrice di rigidezza o flessibilità traslante qui mostrato, alcuni autori hanno proposto nel passato la determinazione di tali matrici mediante ripetute risoluzioni dello schema di telaio a nodi fissi con uno spostamento impresso o a nodi spostabili con una forza applicata. Una simile impostazione può essere considerata interessante dal punto di vista didattico, perché rende visivamente chiara l'essenza delle matrici, ma è sicuramente obsoleta dal punto di vista computazionale e comporterebbe complicazioni e non semplificazioni operative.

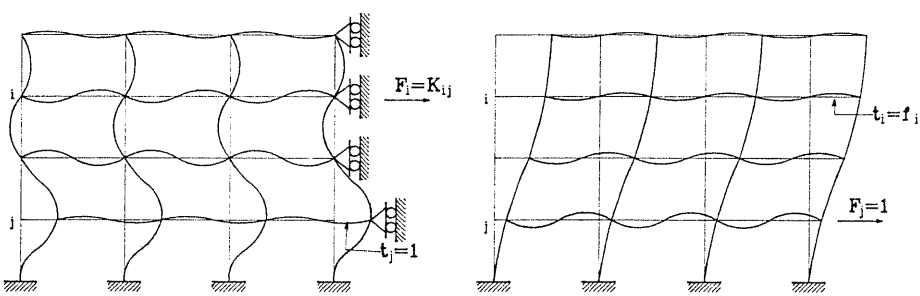


Fig. 4.7 — Significato fisico dei termini delle matrici di rigidezza e flessibilità traslante

In maniera analoga, mediante la (4.5) o con la riduzione Gaussiana e la (4.8) si determina il vettore azioni di incastro $\overline{\mathbf{S}}$ della sottostruttura. Il generico termine $\overline{S}_i$ di esso rappresenta l'azione orizzontale F_i che insorge per effetto dei carichi applicati alle aste o ai nodi interni del telaio quando i traversi sono impediti di spostarsi (fig.4.8).

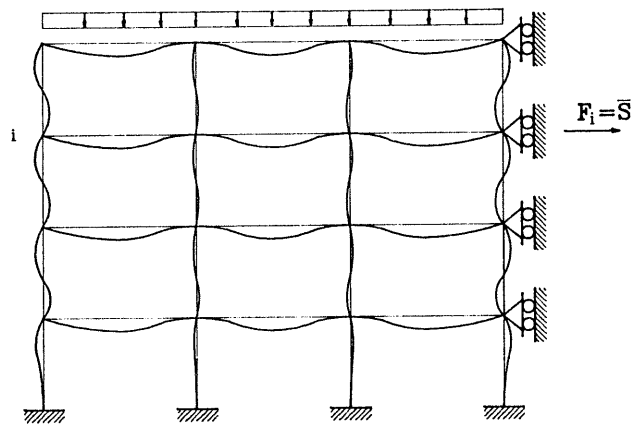


Fig. 4.8 — Significato fisico dei termini del vettore azioni di incastro

4. Matrice di rigidezza traslante di un telaio piano nel riferimento spaziale

Lo studio del comportamento traslante di un telaio piano è finalizzato al considerarlo come componente di un insieme spaziale di telai piani collegati da impalcati indeformabili planimetricamente. Per descrivere un tale insieme si userà come riferimento una terna ortogonale destrorsa xyz con gli assi x ed y orizzontali e l'asse z verticale ed orientato verso l'alto (fig.4.9). Essa viene denominata *sistema di riferimento globale* dello schema spaziale, per distinguerla dal riferimento locale utilizzato per il singolo telaio.

Nella trattazione qui svolta si ipotizza per semplicità che sia i traversi dei telai piani che gli impalcati dello schema spaziale siano tali da costituire un vincolo mutuo allo spostamento orizzontale dei nodi da essi collegati. Occorre a tale proposito osservare che un'asta inestensibile impone che i nodi di estremità traslino della stessa quantità nella direzione del suo asse; si ha l'uguaglianza dello spostamento orizzontale in due casi: quando l'asta è orizzontale o quando lo spostamento verticale dei nodi è impedito, per esempio per la presenza di pilastri indeformabili assialmente. In numerosi casi

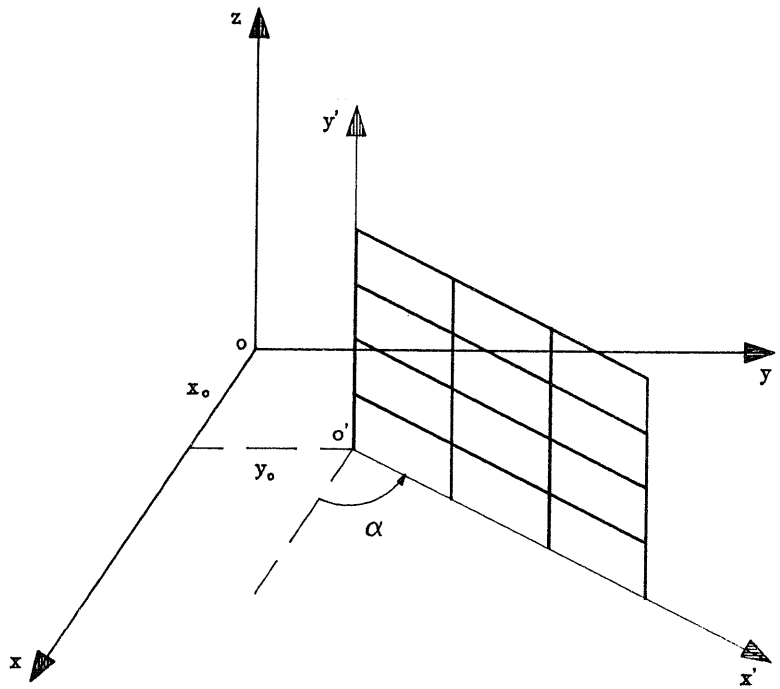


Fig. 4.9 — Sistema di riferimento globale per un insieme spaziale di telai piani

l'ipotesi può però essere formulata in via approssimata, se l'inclinazione è modesta o gli spostamenti verticali sono piccoli rispetto a quelli orizzontali.

Coerentemente con la definizione richiamata all'inizio, l'insieme spaziale di telai piani è una struttura che ha come nodi rigidi gli impalcati e come elementi deformabili i telai piani. Il generico impalcato i, indeformabile nel suo piano, possiede tre gradi di libertà. Le componenti che ne descrivono il movimento sono costituite dalle traslazioni v_{xi} e v_{yi} di un suo punto P (positive se concordi con gli assi) e dalla rotazione intorno all'asse verticale θ_{zi} (positiva se nel verso che porta l'asse x su y). Il punto P può essere scelto arbitrariamente, ed in particolare lo si considera coincidente con l'intersezione dell'asse z col piano dell'impalcato. In maniera duale le azioni ad esso applicate possono essere ricondotte alle forze F_{xi} e F_{yi} in P ed al momento M_{zi}.

Per ciascuna sottostruttura "telaio piano" si possono quindi definire i vettori azioni e componenti di movimento di estremità nel riferimento globale, $\mathbf{S}_g$ e $\mathbf{s}_g$. Nel raggruppare in $\mathbf{S}_g$ le azioni si preferisce disporre ordinatamente prima tutte le forze F_x, poi le F_y ed infine le M_z; $\mathbf{S}_g$ può

quindi considerarsi composto da tre vettori

$$\mathbf{S}_g = \begin{bmatrix} \mathbf{S}_{g_x} \\ \mathbf{S}_{g_y} \\ \mathbf{S}_{g_z} \end{bmatrix}$$

dove per un telaio con n traversi si ha

$$\mathbf{S}_{g_x} = \begin{bmatrix} F_{x_1} \\ F_{x_2} \\ \dots \\ F_{x_n} \end{bmatrix} \qquad \mathbf{S}_{g_y} = \begin{bmatrix} F_{y_1} \\ F_{y_2} \\ \dots \\ F_{y_n} \end{bmatrix} \qquad \mathbf{S}_{g_z} = \begin{bmatrix} M_{z_1} \\ M_{z_2} \\ \dots \\ M_{z_n} \end{bmatrix}$$

In maniera duale si ha per gli spostamenti

$$\mathbf{s}_g = \begin{bmatrix} \mathbf{s}_{g_x} \\ \mathbf{s}_{g_y} \\ \mathbf{s}_{g_z} \end{bmatrix}$$

con

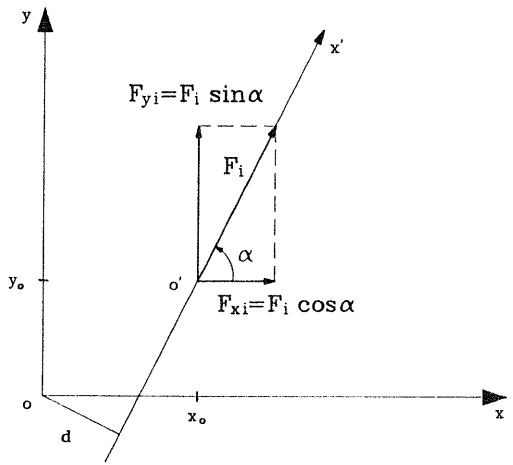
$$\mathbf{s}_{g_x} = \begin{bmatrix} v_{x_1} \\ v_{x_2} \\ \dots \\ v_{x_n} \end{bmatrix} \qquad \mathbf{s}_{g_y} = \begin{bmatrix} v_{y_1} \\ v_{y_2} \\ \dots \\ v_{y_n} \end{bmatrix} \qquad \mathbf{s}_{g_z} = \begin{bmatrix} \theta_{z_1} \\ \theta_{z_2} \\ \dots \\ \theta_{z_n} \end{bmatrix}$$

Per valutare il contributo del telaio piano allo schema spaziale occorre riesaminare le relazioni elastiche, definite per esso nel paragrafo precedente con riferimento al sistema locale $x'y'$, riconducendole a relazioni tra azioni e componenti di movimento degli impalcati nel riferimento spaziale.

Nel definire i sistemi di riferimento, si è implicitamente supposto che l'asse y' del sistema locale sia parallelo ed equiverso all'asse z del riferimento globale e l'asse x' appartenga al piano xy (ma potrebbe appartenere anche ad un piano parallelo ad esso, senza che ciò infici la trattazione del problema). La posizione della terna locale rispetto alla globale è individuata mediante le coordinate x_0, y_0 dell'origine e l'angolo α formato dall'asse x' rispetto all'asse x (positivo se nel verso che porta x su y).

All'azione F_i applicata al traverso i del telaio piano corrispondono nello schema spaziale le tre azioni F_{x_i}, F_{y_i}, M_{z_i}, legate ad essa, all'inclinazione α dell'asse x' ed alla sua distanza d dall'origine mediante le relazioni (fig. 4.10)

$$F_{x_i} = F_i \cos\alpha$$
$$F_{y_i} = F_i \sin\alpha$$
$$M_{z_i} = F_i \, d$$



$M_{zi} = F_{yi} x_o - F_{xi} y_o = F_i \, d$

con $d = x_o \sin\alpha - y_o \cos\alpha$

Fig. 4.10 — Trasformazione dell'azione su un traverso dal sistema di riferimento locale a quello globale

Queste possono essere estese a tutti i traversi ed espresse in forma matriciale per rappresentare il legame tra azioni nel riferimento locale $\mathbf{S}$ e azioni nel riferimento globale $\mathbf{S}_g$. Si ha

$$\begin{aligned} \mathbf{S}_{g_x} &= \cos\alpha \;\; \mathbf{S} \\ \mathbf{S}_{g_y} &= \sin\alpha \;\; \mathbf{S} \\ \mathbf{S}_{g_z} &= d \;\; \mathbf{S} \end{aligned}$$

ovvero

$$\mathbf{S}_g = \mathbf{T}\,\mathbf{S} \tag{4.10}$$

avendo indicato con $\mathbf{T}$ la matrice di trasformazione

$$\mathbf{T} = \begin{bmatrix} \cos\alpha & \mathbf{I} \\ \sin\alpha & \mathbf{I} \\ d & \mathbf{I} \end{bmatrix} \tag{4.11}$$

e con $\mathbf{I}$ la matrice identità di ordine n (cioè una matrice quadrata avente i termini della diagonale principale pari ad 1 e tutti gli altri nulli).

In maniera duale si ricava lo spostamento t_i del traverso conseguente alle componenti di movimento dell'impalcato

$$t_i = v_{x_i} \cos\alpha + v_{y_i} \sin\alpha + \theta_z \, d$$

ed estendendo la relazione a tutti i traversi

$$\mathbf{s} = \cos\alpha \, \mathbf{s}_{g_x} + \sin\alpha \, \mathbf{s}_{g_y} + d \, \mathbf{s}_{g_z}$$

ovvero

$$\mathbf{s} = \mathbf{T}^T \, \mathbf{s}_g \tag{4.12}$$

Utilizzando le (4.10) e (4.12), insieme alla relazione tra azioni, carichi e componenti di movimento nel riferimento locale (4.6), si ottiene l'analoga relazione valida per il sistema di riferimento globale

$$\mathbf{S}_g = \mathbf{T} \, \mathbf{S} = \mathbf{T}(\overline{\mathbf{S}} + \mathbf{k} \, \mathbf{s}) = \mathbf{T} \, \overline{\mathbf{S}} + \mathbf{T} \, \mathbf{k} \, \mathbf{T}^T \, \mathbf{s}_g$$

ovvero

$$\mathbf{S}_g = \overline{\mathbf{S}}_g + \mathbf{k}_g \, \mathbf{s}_g \tag{4.13}$$

avendo indicato rispettivamente con $\mathbf{k}_g$ la matrice di rigidezza traslante e con $\overline{\mathbf{S}}_g$ il vettore azioni di incastro del telaio nel riferimento globale

$$\begin{aligned} \mathbf{k}_g &= \mathbf{T} \, \mathbf{k} \, \mathbf{T}^T \\ \overline{\mathbf{S}}_g &= \mathbf{T} \, \overline{\mathbf{S}} \end{aligned} \tag{4.14}$$

Le (4.14) possono essere esplicitate utilizzando la definizione (4.11) della matrice di trasformazione; si ottiene così

$$\mathbf{k}_g = \begin{bmatrix} \cos^2\alpha \, \mathbf{K} & \sin\alpha \, \cos\alpha \, \mathbf{K} & d \, \cos\alpha \, \mathbf{K} \\ \sin\alpha \, \cos\alpha \mathbf{K} & \sin^2\alpha \, \mathbf{K} & d \, \sin\alpha \, \mathbf{K} \\ d \, \cos\alpha \, \mathbf{K} & d \, \sin\alpha \, \mathbf{K} & d^2 \, \mathbf{K} \end{bmatrix} \tag{4.15}$$

$$\overline{\mathbf{S}}_g = \begin{bmatrix} \cos\alpha \, \overline{\mathbf{S}} \\ \sin\alpha \, \overline{\mathbf{S}} \\ d \, \overline{\mathbf{S}} \end{bmatrix} \tag{4.16}$$

Si osservi che la trasformazione effettuata è perfettamente analoga a quella mostrata nel secondo capitolo come passaggio dagli insiemi fondamentali di azioni e componenti di movimento agli stessi espressi nel riferimento globale. Anche in questo caso gli insiemi $\mathbf{S}_g$ ed $\mathbf{s}_g$ sono "sovrabbondanti" e la loro relazione con gli insiemi fondamentali $\mathbf{S}$ ed $\mathbf{s}$ non è biunivoca; ad esempio, a ciascuna terna v_x v_y θ_z corrisponde un unico spostamento t, ma esistono infinite terne cui corrisponde lo stesso t che differiscono tra loro per una rotazione con centro appartenente all'asse x'. La relazione tra $\mathbf{S}_g$ ed $\mathbf{s}_g$, espressa dalla matrice di rigidezza $\mathbf{k}_g$, non è quindi invertibile e ciò corrisponde fisicamente al fatto che è possibile assegnare agli impalcati spostamenti arbitrari e ricavare le azioni sul telaio a essi conseguenti, ma non ha senso applicare forze arbitrarie e valutare la corrispondente deformazione del telaio perché esso può sopportare solo forze agenti nel suo piano.

5. Matrice di rigidezza dell'insieme spaziale di telai

Esaminando le tipologie di edifici per i quali è possibile adottare il modello di insieme spaziale di telai piani, si possono evidenziare situazioni nelle quali un impalcato non collega tutti i telai. Ciò si ha per esempio nel caso di edifici a piani sfalsati, esaminato numericamente nella terza parte del testo. Per descrivere lo schema spaziale occorre quindi fornire una tabella di corrispondenza tra traversi ed impalcati, che indichi il numero d'ordine dell'impalcato cui appartiene ciascun traverso. Si esclude invece la possibilità che un traverso non sia collegato ad alcun impalcato. Un tale traverso dovrebbe essere conteggiato non tra i nodi di estremità del macroelemento ma tra quelli interni e la relativa forza o spostamento dovrebbe scomparire nella condensazione.

Per quanto detto, l'insieme delle azioni o delle componenti di movimento degli impalcati si presenta in generale più ampio rispetto all'insieme delle azioni o componenti di movimento del singolo telaio nel riferimento spaziale, coincidendo con questo solo nel caso (invero molto frequente) in cui gli impalcati collegano tutti i telai. Si indica con $\mathbf{R}$ il vettore azioni e con $\mathbf{r}$ il vettore componenti di movimento, e si ordinano in essi gli elementi rispettando il criterio definito per il singolo telaio, consentendo così di evidenziare i tre sottoinsiemi $\mathbf{R}_x$ $\mathbf{R}_y$ $\mathbf{R}_z$ e $\mathbf{r}_x$ $\mathbf{r}_y$ $\mathbf{r}_z$ che contengono rispettivamente le azioni (o traslazioni) in direzione x, quelle in direzione y e i momenti (o rotazioni).

La ricerca di relazioni valide per l'insieme spaziale di telai piani, a partire da quelle del singolo telaio, procede allo stesso modo di quanto

fatto nel capitolo precedente per l'insieme piano di aste. Si giunge quindi all'espressione, formalmente identica

$$\mathbf{R} = \overline{\mathbf{R}} - \mathbf{K}\,\mathbf{r} \tag{4.17}$$

Il vettore $\overline{\mathbf{R}}$ contiene le azioni complessive sugli impalcati esercitate dai carichi, applicati sia direttamente ad essi che su aste e nodi interni dei telai, quando il movimento degli impalcati è impedito. Lo si può determinare con facilità seguendo il metodo di determinazione diretta già presentato: per ciascun telaio, il generico termine del vettore azioni di incastro $\overline{\mathbf{S}}_g$ corrispondente ad una direzione d ed un traverso t viene aggiunto al termine di $\overline{\mathbf{R}}$ che corrisponde alla stessa direzione d ed all'impalcato i cui appartiene t (cambiandone il segno, perché $\overline{\mathbf{S}}_g$ contiene le azioni sul telaio mentre $\overline{\mathbf{R}}$ quelle sugli impalcati); ad $\overline{\mathbf{R}}$ si aggiungono poi le azioni esterne applicate direttamente agli impalcati.

La matrice di rigidezza dell'insieme spaziale $\mathbf{K}$ è valutata in maniera analoga. Per ciascun telaio il termine corrispondente ad una direzione d che rappresenta l'interazione tra due traversi t_1 e t_2 viene sommato al termine di $\mathbf{K}$ corrispondente alla stessa direzione d e che rappresenta l'interazione tra gli impalcati i_1 ed i_2 cui appartengono t_1 e t_2.

6. Equilibrio degli impalcati e risoluzione del problema

Per l'equilibrio degli impalcati è necessario che siano nulle le azioni $\mathbf{R}$ su essi complessivamente esercitate. Dalla (4.17) si ha quindi

$$\overline{\mathbf{R}} = \mathbf{K}\,\mathbf{r} \tag{4.18}$$

dalla quale si ottiene la soluzione del problema iperstatico

$$\mathbf{r} = \mathbf{K}^{-1}\,\overline{\mathbf{R}} \tag{4.19}$$

Anche in questo caso conviene risolvere direttamente il sistema di equazioni (4.18), con i procedimenti illustrati nella seconda parte del testo, anziché effettuare l'inversione della matrice $\mathbf{K}$.

Il procedimento da seguire per l'analisi di un insieme spaziale di telai appare quindi concettualmente ben delineato. Occorre innanzitutto esaminare il singolo telaio piano, costruirne la matrice di rigidezza ed il vettore azioni di incastro globali ed effettuarne la condensazione per ricavare la matrice di rigidezza ed il vettore azioni di incastro traslanti. Utilizzando

le informazioni fornite dalla tabella di corrispondenza traversi–impalcati si assemblano le matrici ed i vettori dei singoli telai costruendo quelle relative all'insieme spaziale. La soluzione del sistema (4.18) fornisce i valori delle componenti di movimento degli impalcati, dalle quali si ricavano gli spostamenti dei traversi dei singoli telai. Noti questi, mediante la (4.9) si possono determinare le componenti di movimento dei nodi interni e da queste infine le caratteristiche di sollecitazione nelle aste.

Nel concludere, è opportuno ribadire i limiti di validità dello schema esaminato. I singoli telai sono del tutto indipendenti l'uno dall'altro, tranne che per gli spostamenti orizzontali dei traversi. Il modello non tiene quindi conto dell'accoppiamento flessionale e torsionale di travi di telai distinti incidenti in un unico punto, né della congruenza verticale dei nodi comuni a due telai. In particolare non ha senso utilizzarlo in una analisi che riunisca insieme telai in elevazione ed un graticcio di travi in fondazione, per la quale la congruenza verticale è essenziale. Nonostante ciò, lo schema risulta adottabile in un'ampia gamma di situazioni, perché i problemi innanzi citati sono spesso trascurabili. Inoltre, il criterio di scomporre una struttura in sottostrutture più semplici ha una potenzialità applicativa ancora più vasta perché consente di proporre ed analizzare con facilità numerosi altri modelli strutturali.

PARTE SECONDA

PROGRAMMI

CAPITOLO PRIMO

INTRODUZIONE

1. Analisi strutturale mediante il personal computer

Come ampiamente mostrato nella prima parte del testo, la notazione matriciale ivi introdotta ed il conseguente approccio all'analisi strutturale consentono una notevole sintesi espressiva negli sviluppi teorici. Un'applicazione numerica svolta per via manuale risulterebbe però estremamente onerosa anche nel caso di strutture molto semplici. L'utilizzazione di questa notazione è quindi coincisa con la diffusione degli elaboratori elettronici, che hanno liberato l'ingegnere dall'onere di un gran numero di calcoli ripetitivi.

La seconda parte del testo è pertanto dedicata a mostrare in che modo la teoria illustrata in precedenza possa essere concretizzata in programmi di calcolo automatico. L'obiettivo finale è ovviamente anche quello di giungere a programmi sufficientemente completi e di immediata applicazione, che consentano di affrontare tanto le situazioni più banali quanto quelle più complesse che si possono incontrare nella progettazione di edifici. Ma, più ancora che questo, si spera di indirizzare il lettore verso una modalità di approccio ordinata e schematica, la *programmazione strutturata*, essenziale per realizzare con fatica accettabile nuovi programmi o adattare alle proprie esigenze i programmi qui presentati.

Programmare in maniera strutturata vuol dire suddividere il programma in moduli più semplici ed organizzare in ciascuno di essi la sequenza di istruzioni per il calcolatore utilizzando esclusivamente un numero limitato

di strutture logiche elementari. Una descrizione più dettagliata delle strutture logiche e delle caratteristiche della programmazione strutturata può essere ritrovata nel primo volume di questa serie, "Introduzione alla programmazione", o in testi similari facilmente reperibili in libreria.

L'organizzazione della schema logico sequenziale è solo una delle fasi di stesura di un programma. Quella immediatamente successiva è la *codifica*, cioè la traduzione dello schema in istruzioni che il calcolatore è in grado di riconoscere, cioè che fanno parte di un *linguaggio* disponibile sul computer.

Fino a pochi anni fa, tutte le applicazioni scientifiche più serie venivano implementate su grossi calcolatori e codificate in FORTRAN, linguaggio sviluppato a partire dal 1954 dall'IBM ed organizzato in maniera specifica per le applicazioni matematiche (il nome stesso deriva da FORmula TRANslation, cioè traduzione di formule). La crescente diffusione dei personal computer ha però esteso sempre più l'uso del BASIC (Beginner's All–purpose Symbolic Instruction Code, cioè linguaggio simbolico di uso generale per principianti), nato nel 1963 con lo scopo di fornire un approccio più semplice alla codifica, ma rimasto forse per tale motivo per lungo tempo un linguaggio "di serie B", utilizzato solo per programmi di importanza minore.

La continua evoluzione dei linguaggi e la tendenza di ciascuno di essi ad inglobare le caratteristiche migliori degli altri ha portato a versioni del BASIC molto più potenti di quella iniziale e sotto molti aspetti pari o addirittura superiori al FORTRAN e ad un altro linguaggio oggi di moda, il Pascal. Si è quindi deciso di codificare i programmi illustrati nel testo mediante il Quick Basic della Microsoft, compilatore molto veloce (in inglese quick vuol dire svelto) che nelle sue ultime versioni (4.0 e 4.50) unisce alla potenza del linguaggio ed alla compatibilità col tanto usato ma ormai obsoleto GWBASIC notevoli facilitazioni in fase di scrittura e di controllo della correttezza di un programma, oltre alla possibilità di sfruttare a pieno la memoria del computer. Le sue caratteristiche principali sono richiamate nel paragrafo successivo, mentre si rinvia alla bibliografia citata alla fine del libro per una sua più dettagliata conoscenza.

2. Caratteristiche del Quick Basic

2.1. Sequenza delle istruzioni

Quando ci si riferisce all'utilizzazione del BASIC e delle sue prime versioni per personal computer, si sottointende l'uso di un programma in lin-

guaggio macchina, l'interprete BASIC, che crea un ambiente interattivo all'interno del quale si può sia eseguire calcoli in maniera diretta che scrivere ed eseguire programmi. In particolare, il testo del programma è memorizzato in un'apposita area di memoria e non compare sullo schermo a meno che non lo si richiami con il comando LIST. Le istruzioni del programma possono essere inserite e modificate in un ordine indipendente da quello che sarà poi l'ordine di esecuzione. Per tale motivo ogni istruzione deve essere preceduto da un numero che viene utilizzato dal calcolatore per definire la corretta sequenza di esecuzione.

Il Quick Basic è invece sostanzialmente un compilatore, cioè un programma in linguaggio macchina che rende eseguibile un programma già scritto per intero. Contemporaneamente esso è anche un word processor, che consente di scrivere il programma tenendolo sott'occhio sullo schermo come su un foglio di carta (ed è anzi un word processor "intelligente" perché controlla man mano la correttezza sintattica di quanto viene scritto). La sequenza standard di esecuzione è definita dall'ordine col quale le istruzioni compaiono sullo schermo e l'inserimento di nuove istruzioni avviene spostando il cursore nel punto voluto e scrivendo direttamente in tale posizione. Per tale motivo non è più necessario premettere un numero ad ogni istruzione; la numerazione delle righe diviene quindi un fatto facoltativo, ammesso soprattutto per mantenere la compatibilità col GWBASIC. In definitiva, è possibile avere righe senza numerazione, oppure che iniziano con un numero o, in maniera ancor più generale, con una etichetta, intendendo con tale termine un insieme di caratteri seguito dal simbolo due punti.

Chiunque abbia provato a scrivere programmi in BASIC molto lunghi e a fondere pezzi di differenti programmi si sarà sicuramente scontrato con problemi legati alla numerazione: difficoltà nell'inserire in un listato parecchie righe consecutive, perché i numeri delle linee di programma tra cui inserirle sono troppo vicini tra loro; errori commessi nell'unire due blocchi preparati separatamente, se in questi alcuni numeri si ripetono e quindi alcune istruzioni vengono cancellate da altre. La possibilità di rinunciare alla numerazione, offerta dal Quick Basic, appare quindi come una semplificazione notevole, che conviene senz'altro sfruttare.

Un altro motivo che impone l'uso di numeri di riga nel GWBASIC è la necessità di alterare la sequenza standard mediante istruzioni di salto. La programmazione strutturata non prevede però salti, ma solo l'uso di precise strutture logiche che nel Quick Basic sono codificate con specifiche istruzioni. L'istruzione di salto non è pertanto mai usata nei programmi presentati.

Anche l'uso di routine (istruzione GOSUB) richiede nel GWBASIC la presenza di un numero come indirizzo a cui saltare. Questa istruzione è utilizzata in più casi nei programmi del testo, ma si è preferito individuare l'inizio della routine con una etichetta (che può richiamare subito alla mente il significato del blocco di istruzioni che la costituiscono) anziché con un numero.

2.2. Codifica delle strutture logiche

Il Quick Basic è un linguaggio già orientato alla programmazione strutturata e quindi consente di codificare tutte le strutture logiche fondamentali con specifiche istruzioni.

Le strutture selettive "IF...THEN..." e "IF...THEN...ELSE ..." vengono codificate con le omonime istruzioni, che costituiscono una generalizzazione di quelle già presenti nel GWBASIC perché consentono di sviluppare i blocchi alternativi su più righe separandoli con le parole chiave IF, THEN, ELSE, END IF

```
IF condizione THEN
   ..............................
   ...  blocco di istruzioni 1 ...
   ..............................
ELSE
   ..............................
   ...  blocco di istruzioni 2 ...
   ..............................
END IF
```

La struttura selettiva "CASE...OF..." può essere codificata in due maniere distinte. Se la scelta tra i diversi casi è condizionata dal valore di una variabile conviene usare l'istruzione SELECT CASE

```
SELECT CASE variabile
   CASE valore-1
      ..............................
      ...  blocco di istruzioni 1 ...
      ..............................
   CASE valore-2
      ..............................
      ...  blocco di istruzioni 2 ...
      ..............................
   CASE valore-3
      ..............................
```

```
      ...  blocco di istruzioni 3 ...
      ..............................
   CASE ELSE
      ..............................
      ...  blocco di istruzioni 4 ...
      ..............................
END SELECT
```

Quando le condizioni che controllano la scelta sono più complesse, conviene invece utilizzare una generalizzazione dell'istruzione IF THEN ELSE

```
IF condizione-1 THEN
   ..............................
   ...  blocco di istruzioni 1 ...
   ..............................
ELSEIF condizione-2 THEN
   ..............................
   ...  blocco di istruzioni 2 ...
   ..............................
ELSEIF condizione-3 THEN
   ..............................
   ...  blocco di istruzioni 3 ...
   ..............................
ELSE
   ..............................
   ...  blocco di istruzioni 4 ...
   ..............................
END IF
```

Le strutture di ciclo iterativo possono essere codificate individuando inizio e fine del ciclo con le parole chiave DO e LOOP e ponendo la condizione di controllo del ciclo all'inizio (nel caso della struttura "WHILE...DO...")

```
DO WHILE condizione
   ..............................
   ....  blocco di istruzioni ....
   ..............................
LOOP
```

o alla fine (nel caso della struttura "REPEAT...UNTIL...")

```
DO
   ..............................
   ....  blocco di istruzioni ....
   ..............................
LOOP UNTIL condizione
```

La struttura di ciclo enumerativa "`FOR...DO...`" deve invece essere codificata con le istruzioni `FOR...NEXT...` che non presentano alcuna variazione rispetto a quelle, già note, del GWBASIC.

Il Quick Basic consente di realizzare anche strutture di ciclo interrotto, inserendo l'istruzione `EXIT DO` o `EXIT FOR` all'interno del blocco da ripetere ciclicamente.

2.3. I sottoprogrammi: routine e procedure

Il termine *sottoprogramma* indica genericamente un blocco di istruzioni separato dal programma principale. In fase di esecuzione, quando il calcolatore incontra il comando di salto al sottoprogramma passa ad eseguire le istruzioni in esso contenute per tornare poi, alla fine, all'istruzione immediatamente successiva nella sequenza principale. Questo modo di procedere, nato per evitare di scrivere più volte blocchi di istruzioni che devono essere ripetuti identicamente, è utilizzato nella programmazione strutturata anche, e forse soprattutto, per scomporre il programma in blocchi elementari, indipendentemente dal fatto che questi blocchi vengano utilizzati una o più volte. In tale ottica, l'uso di sottoprogrammi ha una duplice funzione: da un lato, quella di evidenziare l'organizzazione complessiva del programma mostrando in maniera sintetica la sequenza di elaborazioni che esso svolge; dall'altro, quella di creare dei moduli elementari che possono essere riutilizzati nella realizzazione di altri programmi.

Se si passano in rassegna i principali linguaggi disponibili per le applicazioni scientifiche (BASIC, FORTRAN, Pascal), si possono distinguere in essi due tipi di sottoprogrammi. Il primo, tipico del BASIC, è semplicemente un blocco di istruzioni scritto separatamente dal programma principale, ma che ne fa parte a tutti gli effetti ed in particolare ne condivide le variabili. È così possibile utilizzare nel sottoprogramma tutte le informazioni definite nel programma principale ed assegnare in esso nuovi valori a variabili che verranno in seguito utilizzate nella sequenza principale. Il secondo, tipico di FORTRAN e Pascal, costituisce invece, come si suol dire, un *ambiente* distinto da quello del programma principale. Tutti i nomi di variabili in esso utilizzati hanno significato solo all'interno di esso; di conseguenza, se ad esempio nel sottoprogramma si assegna un valore alla variabile `A` e nel programma principale esiste una variabile con lo stesso nome, quest'ultima è distinta dalla prima ed il suo valore non viene modificato. Lo scambio di informazioni tra l'ambiente principale e quello del sottoprogramma avviene in tal caso mediante *parametri di scambio* che

sono elencati nel richiamare il sottoprogramma.

Nel seguito, in tutti i casi nei quali sarà necessario distinguere i due tipi innanzi descritti si denominerà convenzionalmente il primo *routine* ed il secondo *procedura*.

L'uso di routine, accettabile finché si opera con programmi di media complessità, mostra in maniera evidente i suoi limiti in due situazioni. Innanzitutto, quando si costruisce un programma molto grosso, con numerose variabili: per quanto ci si sforzi, è prima o poi quasi inevitabile decidere di utilizzare per una nuova variabile un nome che è già stato usato per un'altra (con effetti imprevedibili sui risultati e con enormi difficoltà nello scoprire la causa dell'errore). In secondo luogo, quando si inserisce un modulo preparato in precedenza in un programma diverso da quello originario: anche in questo caso è molto elevato il rischio di utilizzare lo stesso nome per variabili che dovrebbero essere distinte.

L'uso di procedure risolve brillantemente questi problemi, ma è meno comodo quando si vuole semplicemente evidenziare la struttura del programma dividendolo in blocchi logici. Poiché questi devono in genere condividere gran parte delle informazioni, il numero di parametri di scambio è in questo caso molto elevato ed è quindi facile commettere errori od omissioni nell'elencarli.

Una soluzione ottimale può essere quella di utilizzare, se consentito dal linguaggio adoperato, entrambi i tipi di sottoprogrammi. In tale ottica, le routine vanno utilizzate per dividere il programma principale nei blocchi logici fondamentali (ingresso dati, elaborazioni iniziali, ecc.) rendendo in tal modo immediatamente evidente la sua organizzazione complessiva. Le procedure servono invece per creare moduli elementari di significato ben preciso, riutilizzabili anche in altri programmi (ad esempio moduli che effettuino le operazioni dell'algebra matriciale) e caratterizzati dall'avere un numero non elevato di parametri di scambio.

Il GWBASIC prevede solo il primo tipo di sottoprogramma. L'inizio del blocco di istruzioni della routine non è contrassegnato in maniera particolare; la sua fine logica è costituita dall'istruzione `RETURN` (cioè dal comando di ritorno alla sequenza principale), che per chiarezza di lettura andrebbe sempre disposta in coda al blocco. La routine viene richiamata dal programma principale mediante l'istruzione

`GOSUB` *numero-linea*

dove con *numero-linea* si intende il numero che contrassegna la prima riga

della routine.

Il Quick Basic rende disponibile anche il secondo tipo di sottoprogramma. La serie di istruzioni è in tal caso racchiusa tra le parole SUB e END SUB ed individuata mediante un nome

```
SUB nome-procedura (elenco variabili di scambio)
    ..............................
    ....  blocco di istruzioni ....
    ..............................
END SUB
```

La procedura viene richiamata dal programma principale (o da altri sottoprogrammi) mediante l'istruzione

```
CALL nome-procedura (valori o variabili di scambio)
```

Per un corretto uso delle procedure, è fondamentale chiarire bene in che modo avviene lo scambio di informazioni tra i due distinti ambienti del programma principale e del sottoprogramma. Come appena indicato, la definizione di una procedura viene effettuata facendo seguire al suo nome l'elenco dei nomi delle variabili di scambio; se tra queste deve essere incluso un intero array, se ne indica il nome seguito da due parentesi, senza indice. Ad esempio

```
SUB CalcolaValori (A, B, N, F(), V$)
```

è l'inizio di una procedura, di nome `CalcolaValori`, dotata di cinque parametri di scambio; i primi tre sono variabili numeriche semplici, il quarto un array numerico, l'ultimo una variabile alfanumerica.

Nel chiamare la procedura con l'istruzione CALL si fa seguire al nome un elenco di valori o di variabili (con nomi definiti nel programma chiamante), con un numero di termini e tipo delle informazioni che si devono accordare perfettamente con quanto previsto nella definizione della procedura. La procedura innanzi definita può ad esempio essere richiamata con

```
CALL CalcolaValori(5, (F), J, A(), "testo")
```

Lo scambio di informazioni può avvenire, come si vede, mediante scambio di valori oppure di nomi di variabili. Quando nell'elenco è indicato un valore, questo viene assegnato alla corrispondente variabile del sottoprogramma; nell'esempio mostrato ad A si assegna il valore 5, a V$ "testo", a B il valore corrente della variabile F del programma chiamante (con (F)

si indica proprio il valore di F). Quando invece si fornisce il nome di una variabile, questa coincide a tutti gli effetti con la corrispondente nell'elenco di definizione, anche se i nomi sono differenti. Nell'esempio, anziché creare una nuova area di memoria per la variabile N della procedura il nome di questa viene collegato all'area già definita per la variabile J del programma chiamante, consentendo così di leggerne o modificarne il valore. Analogamente, l'array F() viene collegato all'array A(), assumendone quindi anche il numero di indici e di elementi.

Nel descrivere le informazioni comunicate ad una procedura si è soliti distinguere tra informazioni fornite ad essa (contenute nelle variabili di ingresso) ed informazioni ottenute da essa (contenute nelle variabili di uscita). Usualmente, nel chiamare una procedura si inseriscono nell'elenco valori numerici per le variabili di ingresso e nomi di variabili per quelle di uscita. Nell'esempio, quindi, A, B e V$ potrebbero essere le variabili di ingresso, N e F() quelle di uscita.

2.4. Altre istruzioni

Il Quick Basic consente l'uso di numerose istruzioni non presenti nel BASIC standard. Per comodità del lettore, si richiamano in ordine alfabetico le istruzioni utilizzate nei programmi presentati nel testo, accompagnate da una breve spiegazione. Maggiori dettagli andranno ricercati nei manuali forniti dalla Microsoft insieme al linguaggio.

DECLARE SUB *nome-procedura (elenco variabili di scambio)*

Questa istruzione deve essere posta all'inizio del programma principale, per indicare quali sono le procedure in esso utilizzate, quanti parametri ciascuna richiede e quale è il tipo dei parametri.

DIM SHARED *nome-variabile*

La variabile indicata viene considerata come appartenente contemporaneamente al programma principale ed alle procedure, anche se il suo nome non è compreso tra i parametri di scambio.

FREEFILE

Funzione che fornisce il più piccolo valore disponibile come numero di buffer per l'apertura di un file.

LBOUND *(nome-array, n)*

Funzione che fornisce il valore minimo dell'indice *n* dell'array indicato.

RESUME

Nel caso si sia verificato un errore, questa istruzione indica di riprendere l'esecuzione del programma ripetendo la stessa istruzione per la quale è accaduto l'errore.

RESUME NEXT

Nel caso si sia verificato un errore, questa istruzione indica di riprendere l'esecuzione del programma dalla istruzione immediatamente successiva a quella nella quale l'errore è accaduto.

SWAP *variabile-1, variabile-2*

Scambia tra loro i valori delle due variabili indicate.

TIMER

Funzione che fornisce il numero di secondi trascorsi dalla mezzanotte, in base all'orologio interno del computer.

UBOUND *(nome-array, n)*

Funzione che fornisce il valore massimo dell'indice *n* dell'array indicato.

3. Contenuto del dischetto

A ciascuna copia del libro sono allegati tre dischetti da 5.25 pollici, formattati a 360 Kbytes per elaboratori con sistema operativo MS–DOS. I dischetti contengono esclusivamente i file di programmi e dati descritti nel testo; in essi non vi è quindi il sistema operativo o il compilatore Quick Basic, che si ritiene siano già in possesso del lettore. Se l'elaboratore non è dotato di disco rigido, occorre pertanto accendere il calcolatore dopo aver disposto il proprio disco di sistema nel drive A. Diventa così immediatamente possibile utilizzare i dischetti forniti insieme al libro mandando in esecuzione le versioni compilate dei programmi. Il testo BASIC originario dei programmi può invece essere utilizzato, per modificarli o per sfruttare per altre applicazioni le procedure in essi contenute, solo dopo aver caricato in memoria il linguaggio Quick Basic.

Nei dischetti è riportato innanzitutto il file HELP.EXE che una volta mandato in esecuzione fornisce indicazioni sul contenuto dei dischi. Le informazioni da visualizzare sono memorizzate in file di testo con l'estensione .HLP.

I dischetti contengono poi 10 file aventi l'estensione .BAS, che costituiscono i programmi e le procedure in Quick Basic descritti nei capitoli successivi:

ASTA.BAS	insieme di procedure che forniscono la matrice di rigidezza, le azioni di incastro perfetto e le caratteristiche di sollecitazione per diversi tipi di aste e di carichi (cap. 4).
DIAGRAM.BAS	programma esemplificativo per il tracciamento dei diagrammi delle caratteristiche di sollecitazione in un portale (cap.4).
MATRIX.BAS	insieme di procedure per la effettuazione di operazioni sulle matrici (cap. 2).
PROCOM.BAS	insieme di procedure comuni a tutti gli schemi intelaiati considerati (cap. 5).
PROGEN.BAS	insieme di procedure caratteristiche del telaio generico, utilizzate anche per il telaio spaziale (cap. 5).
PRORET.BAS	insieme di procedure caratteristiche del telaio a maglie rettangolari o trapezie, utilizzate anche per il telaio spaziale (cap. 6).
SOLSIST.BAS	insieme di procedure per la risoluzione dei sistemi di equazioni lineari (cap. 3).
TELGEN.BAS	programma principale per la risoluzione di uno schema intelaiato piano generico (cap. 5).
TELRET.BAS	programma principale per la risoluzione di uno schema intelaiato piano a maglie rettangolari o trapezie (cap.6).
TELSPA.BAS	programma principale per la risoluzione di uno schema spaziale di telai piani generici o a maglie rettangolari o trapezie (cap. 7).

I dischetti contengono inoltre, per ciascuno dei programmi innanzi indicati (DIAGRAM, TELGEN, TELRET, TELSPA), un file con l'estensione .MAK ed uno con l'estensione .EXE, che contengono rispettivamente l'elenco dei file che costituiscono il programma e la sua versione compilata, direttamente eseguibile.

Nei dischetti sono infine presenti una serie di file privi di estensione, nei quali sono contenuti i dati richiesti dai programmi per l'elaborazione dei

calcoli riportati come esempio nel testo.

4. Utilizzazione dei programmi

Come noto, nel BASIC esistono più istruzioni che consentono l'assegnazione di valori alle variabili di ingresso, cioè ai dati del problema.

L'istruzione `INPUT` si presta ad un ingresso dati conversazionale, nel quale i singoli valori siano richiesti espressamente all'utente. Non è però facilmente possibile la correzione di eventuali errori o la riutilizzazione degli stessi dati a distanza di tempo.

L'istruzione `READ` richiede invece la memorizzazione sequenziale dei dati in linee di programma contraddistinte dalla parola `DATA`. Essendo in tal modo possibile la correzione ed il riutilizzo dei dati, tale istruzione è preferibile alla precedente per quei problemi che richiedono una notevole mole di informazioni in ingresso. I dati così definiti fanno però parte integrante del programma, e non possono quindi venire modificati se esso è stato compilato.

L'istruzione `INPUT#` opera in maniera analoga alla `READ`, ma preleva i valori direttamente da un file sequenziale. Oltre a presentare i vantaggi già evidenziati per tale istruzione, essa è quindi utilizzabile anche in programmi compilati.

È infine possibile usare le istruzioni che consentono una gestione diretta della tastiera e dello schermo per realizzare maschere di ingresso dati che uniscono il vantaggio della interattività con la possibilità di correzione e riutilizzo dei valori.

Tra le quattro alternative elencate, l'ultima è indubbiamente da preferire, ma richiede un maggior onere di programmazione. Si è pertanto fatto ricorso all'istruzione `READ` nel programma `DIAGRAM`, che ha solo uno scopo esemplificativo. Nei restanti programmi si è invece utilizzata l'istruzione `INPUT#`. Per adoperarli occorre quindi aver preparato in precedenza il file dati con un qualsiasi word processor.

Per quanto riguarda l'output di dati e risultati, si è voluto consentire all'utente la scelta dell'unità cui inviare tali informazioni. Nei programmi si è pertanto adoperata l'istruzione di scrittura su file (`PRINT#`) e si è aggiunta all'inizio la richiesta del nome del file di output. Indicando per esso `CON` (cioè *console*) i valori di uscita vengono visualizzati sullo schermo. Indicando `PRN` (*printer*) i valori vengono stampati su carta. Se invece si indica il nome di un file, i risultati vengono memorizzati in esso. Quest'ultima alternativa è particolarmente comoda, perché il contenuto del file può essere successivamente esaminato sullo schermo, oppure inviato alla stampa

con l'istruzione del sistema operativo TYPE *nome-file* > PRN oppure PRINT *nome-file*.

5. Utilizzazione delle procedure e dei programmi in BASIC

Una delle caratteristiche principali del Quick Basic è il consentire che i programmi siano frazionati in più file (o, come si suol dire, in più moduli). Si è così potuto raggruppare in singoli file ciascun insieme di procedure omogenee, come ad esempio quelle relative alle operazioni matriciali, creando in tal modo una volta per tutte una biblioteca di procedure che potrà essere utilizzata in seguito anche per altri programmi. Lavorando nell'ambiente Quick Basic, per unire ad un programma che si sta realizzando un file già esistente è sufficiente caricare in memoria quest'ultimo con l'opzione "Load File..." e salvare poi il programma con l'opzione "Save All..." (nella versione italiana rispettivamente "Carica file..." e "Salva tutto..."). Il Quick Basic crea in tal caso automaticamente un file con l'estensione .MAK che indica quali file costituiscono il programma. Da questo momento in poi, per caricare in memoria tutti i moduli del programma è sufficiente utilizzare nel caricamento l'opzione "Open Program..." (in italiano "Apri programma..."). Nella fase di compilazione viene creato un file rilocabile (con l'estensione .OBJ) per ciascun modulo, e questi file vengono poi riuniti dal Linker in un unico elemento assoluto (con l'estensione .EXE).

CAPITOLO SECONDO

OPERAZIONI MATRICIALI

1. Generalità

Il linguaggio BASIC, sia nella versione standard che in quelle più sofisticate, è privo di istruzioni matriciali. Per sopperire a questa carenza, che rischierebbe di vanificare la sintesi espressiva consentita dall'uso di tale notazione nell'analisi strutturale, è stato realizzato un insieme di procedure che abilitano le istruzioni in questione.

Le procedure, contenute nel file `MATRIX.BAS`, possono essere divise in tre gruppi:

- procedure che facilitano la costruzione o l'esame di una matrice:
  `AzzeraMat`, `CostanteMat`, `DuplicaMat`,
  `IdentitaMat`, `InserMat`, `LimitiMat`;

- procedure che implementano le operazioni matriciali:
  `InversaMat`, `ProdottoCostMat`, `ProdottoMat`,
  `SommaMat`, `SottraeMat`, `TrasponeMat`;

- procedure per leggere o inviare in uscita il contenuto di una matrice:
  `InputMat`, `LprintMat`, `OutputMat`,
  `PrintMat`, `WriteMat`.

La funzione svolta da ciascuna di esse può essere facilmente dedotta dal nome attribuitole, ed è comunque esplicitamente indicata nel listato riportato nel paragrafo 3. In tale listato sono anche evidenziate le variabili di ingresso ed uscita che costituiscono i parametri di scambio tra programma chiamante e procedura.

Tutte le procedure, essendo dedicate all'analisi matriciale, operano su array ad una o due dimensioni. Gli array monodimensionali sono considerati vettori colonna, cioè costituiti da più righe ed un'unica colonna, e sono quindi del tutto equivalenti ad array bidimensionali dimensionati in modo da avere come secondo indice il valore 1.

Ciascun array deve essere dimensionato prima di essere utilizzato come parametro di scambio. Il valore minimo di ogni indice deve essere pari ad 1; valori inferiori (come lo 0, che spesso si ha come default) sono ammessi, ma i corrispondenti elementi dell'array non vengono presi in considerazione. Il valore massimo deve essere definito esattamente, in funzione delle necessità operative; le procedure operano infatti sull'intero array, valutandone automaticamente numero di dimensioni e limiti degli indici.

Si fa infine presente che le procedure sono state scritte in modo da poter effettuare, quando le regole dell'algebra matriciale lo consentano, l'esecuzione delle operazioni operando sulla stessa matrice di partenza. Ad esempio, la trasposta di una matrice rettangolare ha un numero di righe e colonne diverso (e più precisamente scambiato) rispetto alla matrice di partenza. In generale, quindi, si effettuerà la trasposizione di una matrice `A()` mediante l'istruzione

```
CALL TrasponeMat(A(), B())
```

che mette il risultato nella matrice `B()` distinta dalla precedente. Se la matrice è quadrata, è però possibile memorizzare la trasposta direttamente al posto della matrice di partenza mediante l'istruzione

```
CALL TrasponeMat(A(), A())
```

nella quale la variabile di uscita coincide con quella di ingresso.

In definitiva, nella realizzazione delle procedure si è preferito privilegiare la generalità (ammettendo l'uso di vettori colonna e la coincidenza tra matrici di ingresso ed uscita) rispetto alla sinteticità ed alla rapidità di esecuzione. Un attento esame del listato mostra le conseguenze di questa scelta: suddivisione di ciascuna procedura in più sottoprocedure, individuate dall'aggiunta di una cifra al nome base, per poter operare con array mono e bidimensionali; introduzione di matrici ausiliarie, per evitare la

perdita di informazioni in ingresso nel caso di coincidenza delle matrici.

2. Procedimenti per l'esecuzione delle operazioni

Gran parte dei procedimenti per l'esecuzione di operazioni su matrici sono molto semplici; essi possono essere facilmente compresi mediante l'esame della relativa codifica e, pertanto, non si ritiene opportuno dilungarsi sulla loro descrizione. Solo a titolo di esempio, si prende in esame la procedura `ProdottoMat`, che effettua il prodotto di due matrici, **A** e **B**, fornendo come risultato la matrice **C**.

Si ricorda dall'algebra delle matrici che, affinché il prodotto possa essere effettuato, è necessario che il numero di colonne della prima matrice sia uguale al numero di righe della seconda (condizione di compatibilità del prodotto); la matrice prodotto ottenuta avrà un numero di righe uguale a quello della prima matrice e un numero di colonne uguale a quello della seconda. Nella prima parte della procedura viene quindi richiamata tre volte la procedura `LimitiMat`, per determinare il numero di dimensioni degli array `A`, `B` e `C` ed i limiti di ciascun indice, e ne viene poi controllata la compatibilità. Si passa infine ad effettuare il prodotto, rinviando a tre distinte procedure (`ProdottoMat1`, `2` e `3`) a seconda del numero di dimensioni dei primi due array. Il valore numerico degli elementi componenti la matrice prodotto **C** è dato dalla relazione

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

ovvero, il generico elemento c_{ij} di **C** è pari alla somma dei prodotti degli elementi della riga i–esima di **A** per gli omologhi elementi della colonna j–esima di **B**. Si noti però nella procedura l'uso intermedio di un array ausiliario `D`, nel quale vengono temporaneamente conservati i valori ottenuti in modo da evitare problemi nel caso che la matrice prodotto sia coincidente con una delle matrici fattore.

L'unica operazione matriciale di una certa complessità è quella di inversione, che può essere effettuata solamente su matrici quadrate. Si definisce matrice inversa di una matrice quadrata **A** di ordine n una matrice **B**, dello stesso ordine, tale che il prodotto $\mathbf{B} \times \mathbf{A}$, ovvero il prodotto $\mathbf{A} \times \mathbf{B}$, risulti uguale alla matrice identità **I**; la matrice inversa viene denotata col simbolo $\mathbf{A}^{-1}$. Una matrice per la quale sia possibile costruire l'inversa è detta *non singolare*.

La determinazione della matrice inversa può essere effettuata operando contemporaneamente sulla matrice da invertire e sulla matrice identità, mediante combinazioni lineari di righe e colonne tali da trasformare nella matrice identità quella di partenza. Si consideri, infatti, l'uguaglianza

$$\mathbf{A} \times \mathbf{A}^{-1} = \mathbf{I}$$

che scaturisce dalla definizione innanzi richiamata. L'uguaglianza continua a sussistere se si opera contemporaneamente sulle righe di **A** ed **I**, effettuandone una qualsiasi combinazione lineare (il prodotto $\mathbf{A} \times \mathbf{A}^{-1}$ è un prodotto "righe per colonne", e quindi effettuando una combinazione lineare delle righe del primo fattore, **A**, si ottiene la stessa combinazione nelle righe del risultato, **I**). In particolare, se con tali operazioni si trasforma la **A** nella matrice identità e la **I** in una matrice **B** si avrà

$$\mathbf{I} \times \mathbf{A}^{-1} = \mathbf{B}$$

ovvero, per la definizione stessa di matrice identità

$$\mathbf{B} = \mathbf{A}^{-1}$$

Dal punto di vista operativo si presentano però due problemi. Il primo è connesso alla necessità di effettuare un controllo sul valore assoluto dei termini diagonali: valori molto piccoli possono dar luogo a rilevanti errori di approssimazione mentre valori nulli vanificano addirittura l'operazione anche quando la matrice è invertibile. Il secondo è legato all'opportunità di effettuare l'operazione di inversione sulla stessa matrice di partenza in modo da dimezzare l'ingombro di memoria richiesto.

Il primo problema può essere risolto associando al procedimento di inversione il riordinamento delle equazioni, ovvero procedendo, in ciascuna fase della trasformazione della matrice **A** in matrice identità, all'individuazione della riga e della colonna cui corrisponde l'elemento maggiore in valore assoluto.

Si considerino ad esempio le matrici di ordine n ($n = 3$)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

dove **A** è la matrice da invertire e **B** la matrice risultante, inizialmente pari alla matrice identità.

Si suppone che l'elemento maggiore, genericamente indicato con a_{rc}, sia l'a_{32}, posto nella terza riga ($r = 3$), seconda colonna ($c = 2$). Si inizia il primo passo di trasformazione rendendo unitario tale elemento, ovvero dividendo la corrispondente riga di **A** e di **B** per il valore dell'elemento stesso. Si ottiene così

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a'_{31} & 1 & a'_{33} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & b'_{33} \end{bmatrix}$$

dove i termini contrassegnati dall'apice sono stati valutati mediante le relazioni

$$\begin{aligned} a'_{rj} &= \frac{a_{rj}}{a_{rc}} \\ b'_{rj} &= \frac{b_{rj}}{a_{rc}} \end{aligned} \tag{2.1}$$

L'apice è usato solo per evidenziare gli elementi modificati, perché i nuovi valori vanno conservati al posto dei precedenti e quindi a'_{rj} diventa, a tutti gli effetti, il nuovo a_{rj}.

Si prosegue quindi sottraendo alle restanti righe un multiplo della riga r, in modo da azzerare tutti gli altri elementi della colonna c ottenendo

$$\mathbf{A} = \begin{bmatrix} a'_{11} & 0 & a'_{13} \\ a'_{21} & 0 & a'_{23} \\ a_{31} & 1 & a_{33} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 & b'_{13} \\ 0 & 1 & b'_{23} \\ 0 & 0 & b_{33} \end{bmatrix}$$

Gli elementi della generica riga i, diversa da r, sono valutati mediante le relazioni:

$$\begin{aligned} a'_{ij} &= a_{ij} - a_{ic}\ a_{rj} \\ b'_{ij} &= b_{ij} - a_{ic}\ b_{rj} \end{aligned} \tag{2.2}$$

Per la colonna r della matrice **B**, che ha l'elemento b_{rr} unitario e tutti gli altri nulli, le espressioni (2.1) e (2.2) possono particolarizzarsi nella forma

$$\begin{aligned} b'_{rr} &= \frac{1}{a_{rc}} \\ b'_{ir} &= -\frac{a_{ic}}{a_{rc}} \end{aligned} \tag{2.3}$$

Si osservi infine che in tutte le relazioni compaiono gli elementi della colonna c di $\mathbf{A}$, che è quindi necessario memorizzare temporaneamente in un vettore ausiliario $\mathbf{C}$.

Nel secondo passo di trasformazione l'elemento maggiore va ancora ricercato fra tutti quelli della matrice ad esclusione però degli elementi delle righe e colonne già esaminate (nel nostro caso quelli della terza riga e della seconda colonna).

Nell'ipotesi che l'elemento maggiore sia l'a_{21}, ripetendo le due fasi di trasformazione precedentemente definite si ottiene

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & a_{13} \\ 1 & 0 & a_{23} \\ 0 & 1 & a_{33} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \end{bmatrix}$$

Nel successivo, ultimo passo di trasformazione l'elemento maggiore della matrice va ricercato escludendo anche quelli della seconda riga e della prima colonna, ed è quindi necessariamente l'a_{13}. Ripetendo ancora una volta le fasi di trasformazione si giunge a

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

La matrice $\mathbf{A}$ così ottenuta rappresenta, a meno di uno scambio di righe, la matrice identità cercata. Il procedimento si completa quindi con tale scambio, spostando la prima riga in ultima posizione e facendo avanzare di una posizione le altre due righe, fornendo la ricercata matrice inversa

$$\mathbf{A}^{-1} = \mathbf{B} = \begin{bmatrix} b'_{11} & b'_{12} & b'_{13} \\ b'_{21} & b'_{22} & b'_{23} \\ b'_{31} & b'_{32} & b'_{33} \end{bmatrix} = \begin{bmatrix} b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{11} & b_{12} & b_{13} \end{bmatrix}$$

Il secondo problema, relativo alla opportunità di eseguire l'operazione di inversione sulla stessa matrice di partenza, può essere risolto riesaminando le operazioni effettuate.

Con riferimento all'esempio mostrato, si può notare innanzitutto che il primo passo di trasformazione ha portato la seconda colonna di $\mathbf{A}$ a valori predefiniti (0 ed 1) che è inutile memorizzare. Contemporaneamente, la particolare forma iniziale di $\mathbf{B}$ ha fatto sì che solo la terza colonna di essa risentisse delle modifiche. Appare quindi possibile sfruttare lo spazio superfluo in $\mathbf{A}$ per conservare le informazioni contenute in $\mathbf{B}$, spostando la

terza colonna di **A** al posto della seconda e memorizzando nel posto così liberato la terza colonna di **B**

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{13} & b_{13} \\ a_{21} & a_{23} & b_{23} \\ a_{31} & a_{33} & b_{33} \end{bmatrix}$$

Per mantenere il ricordo degli spostamenti effettuati si utilizza un vettore ausiliario **M**, costituito da una riga ed n colonne. Inizialmente si pone in ciascun elemento di esso un numero pari all'indice di colonna, col segno negativo per indicare convenzionalmente che la corrispondente colonna di **A** contiene elementi appartenenti ad **A**

$$\mathbf{M} = \begin{bmatrix} -1 & -2 & -3 \end{bmatrix}$$

Lo scambio tra seconda e terza colonna di **A**, effettuato nel primo passo di trasformazione, viene registrato in **M** scambiando tra loro gli elementi m_2 ed m_3. Il nuovo m_3 viene inoltre cambiato di segno, per indicare che nella terza colonna sono ora contenuti elementi di **B**. **M** diventa quindi

$$\mathbf{M} = \begin{bmatrix} -1 & -3 & 2 \end{bmatrix}$$

Si osservi che l'avere in posizione 3 il valore 2 indica che l'elemento reso unitario in questo passo è l'a_{32}, informazione necessaria alla fine per ordinare in maniera corretta le righe.

In maniera analoga, nel secondo passo è diventata superflua la prima colonna di **A** mentre è diventata significativa una nuova colonna di **B**, la seconda. Si può quindi invertire la attuale seconda colonna di **A** (ex terza colonna) con la prima e memorizzare al suo posto la seconda colonna di **B**, modificando contemporaneamente anche il vettore **M**

$$\mathbf{A} = \begin{bmatrix} a_{13} & b_{12} & b_{13} \\ a_{23} & b_{22} & b_{23} \\ a_{33} & b_{32} & b_{33} \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} -3 & 1 & 2 \end{bmatrix}$$

Si ricorda che i valori degli elementi della nuova colonna di **B** sono stati determinati mediante le (2.3), mentre quelli delle altre colonne di **A** e **B** sono stati modificati usando le (2.1) e (2.2). Queste ultime trattano allo stesso modo gli elementi di **A** e di **B** e non richiedono alcuna modifica per l'accorpamento delle due matrici in una.

Con l'ultimo passo di trasformazione si giunge infine a

$$\mathbf{A} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix}$$

Le inversioni di riga necessarie per l'ottenimento della matrice inversa sono chiaramente indicate dal vettore **M**: l'attuale prima riga va in terza posizione, la seconda in prima, la terza in seconda.

La complessità operativa del procedimento di inversione qui descritto può essere valutata calcolando il numero di operazioni che esso comporta; in particolare, può ritenersi sufficiente contare solo le moltiplicazioni e divisioni, che richiedono al calcolatore un tempo nettamente maggiore rispetto ad addizioni e sottrazioni. È facile constatare che in ciascun passo di trasformazione si devono applicare le (2.1) e (2.2) agli elementi di $n-1$ colonne, di **A** o di **B**, e le (2.3) a quelli di una colonna di **B**. Poiché ciascuna colonna contiene n elementi, ed in ogni relazione compare una sola moltiplicazione o divisione, si effettuano n^2 operazioni per passo. Essendo ancora n il numero di passi necessari per la determinazione della matrice inversa, il numero totale di operazioni richiesto è quindi n^3.

È interessante effettuare un confronto con il numero di operazioni necessario per il prodotto matriciale. Se si moltiplica una matrice $m \times p$ per una $p \times n$, la determinazione di ogni elemento della matrice prodotto richiede p moltiplicazioni. Poiché essa contiene $m \times n$ elementi, il numero totale di operazioni è pari a $p \times m \times n$. Se si moltiplicano tra loro due matrici quadrate di ordine n, occorrono quindi n^3 operazioni. L'onere numerico dell'inversione è pertanto sostanzialmente pari a quello del prodotto.

3. Codifica

```
'                              MATRIX.BAS
'
'                    procedure per le operazioni matriciali
'
'                                                          rev. 09.89
'

' ------------------ Dichiarazione delle procedure utilizzate ----------------------

DECLARE SUB AzzeraMat1 (A!(), NRA!)
DECLARE SUB AzzeraMat2 (A!(), NRA!, NCA!)
DECLARE SUB CostanteMat1 (V!, A!(), NRA!)
DECLARE SUB CostanteMat2 (V!, A!(), NRA!, NCA!)
```

```
DECLARE SUB DuplicaMat (A!(), B!())
DECLARE SUB DuplicaMat1 (A!(), NRA!, B!())
DECLARE SUB DuplicaMat2 (A!(), NRA!, B!())
DECLARE SUB DuplicaMat3 (A!(), NRA!, B!())
DECLARE SUB DuplicaMat4 (A!(), NRA!, NCA!, B!())
DECLARE SUB InputMat1 (A!(), F!, NRA!)
DECLARE SUB InputMat2 (A!(), F!, NRA!, NCA!)
DECLARE SUB InserMat1 (A!(), NRA!, RI!, B!())
DECLARE SUB InserMat2 (A!(), NRA!, RI!, CI!, B!())
DECLARE SUB InserMat3 (A!(), NRA!, NCA!, RI!, CI!, B!())
DECLARE SUB LimitiMat (A!(), DA!, LR!, LC!, NR!, NC!)
DECLARE SUB OutputMat (A!(), F!, F$)
DECLARE SUB OutputMat1 (A!(), F!, F$, NRA!)
DECLARE SUB OutputMat2 (A!(), F!, F$, NRA!, NCA!)
DECLARE SUB ProdottoCostMat1 (K!, A!(), NRA!, B!())
DECLARE SUB ProdottoCostMat2 (K!, A!(), NRA!, NCA!, B!())
DECLARE SUB ProdottoMat1 (A!(), B!(), NRA!, NCB!, C!())
DECLARE SUB ProdottoMat2 (A!(), B!(), NRA!, NCA!, C!())
DECLARE SUB ProdottoMat3 (A!(), B!(), NRA!, NCB!, NCA!, C!())
DECLARE SUB SommaMat1 (A!(), B!(), NRA!, C!())
DECLARE SUB SommaMat2 (A!(), B!(), NRA!, NCA!, C!())
DECLARE SUB SottraeMat1 (A!(), B!(), NRA!, C!())
DECLARE SUB SottraeMat2 (A!(), B!(), NRA!, NCA!, C!())
DECLARE SUB TrasponeMat1 (A!(), NRA!, B!())
DECLARE SUB TrasponeMat2 (A!(), NRA!, NCA!, B!())
DECLARE SUB WriteMat1 (A!(), F!, NRA!)
DECLARE SUB WriteMat2 (A!(), F!, NRA!, NCA!)

ErroreLimitiMat:
    RESUME NEXT

' ============================== AzzeraMat ==============================
'
'   Procedura per l'azzeramento di un array monodimensionale o bidimensionale
'
'   Variabili di ingresso:
'      A()       array da azzerare
'
'   Variabili di uscita:
'      A()       array azzerato
'
'   Procedure utilizzate:
'      LimitiMat
'
' -----------------------------------------------------------------------------
'
SUB AzzeraMat (A())

    ' determina il numero di indici e di righe e colonne dell'array A
    ' controlla che il numero di righe e colonne di A sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    IF LRA > 1 OR LCA > 1 THEN
       CLS
       PRINT "ERRORE 1 in AzzeraMat - limiti inferiori di A maggiori di 1"
       END
    END IF

    ' azzera l'array
    IF DA = 1 THEN
```

```
      CALL AzzeraMat1(A(), NRA)
   ELSE
      CALL AzzeraMat2(A(), NRA, NCA)
   END IF

END SUB

SUB AzzeraMat1 (A(), NRA)

   FOR R = 1 TO NRA
      A(R) = 0
   NEXT R

END SUB

SUB AzzeraMat2 (A(), NRA, NCA)

   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         A(R, C) = 0
      NEXT C
   NEXT R

END SUB

' ============================== CostanteMat ==============================
'
'  Procedura per la creazione di un array (monodimensionale o bidimensionale)
'  con tutti gli elementi identicamente uguali ad un valore assegnato
'
'  Variabili di ingresso:
'     V          valore da assegnare agli elementi dell'array
'
'  Variabili di uscita:
'     A()        array ottenuto
'
'  Procedure utilizzate:
'     LimitiMat
'
' -----------------------------------------------------------------------------
'
SUB CostanteMat (V, A())

   ' determina il numero di indici e di righe e colonne dell'array A
   ' controlla che il numero di righe e colonne di A sia corretto
   CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
   IF LRA > 1 OR LCA > 1 THEN
      CLS
      PRINT "ERRORE 1 in CostanteMat - limiti inferiori di A maggiori di 1"
      END
   END IF

   ' assegna il valore agli elementi dell'array
   IF DA = 1 THEN
      CALL CostanteMat1(V, A(), NRA)
   ELSE
      CALL CostanteMat2(V, A(), NRA, NCA)
   END IF
```

```
END SUB

SUB CostanteMat1 (V, A(), NRA)

    FOR R = 1 TO NRA
       A(R) = V
    NEXT R

END SUB

SUB CostanteMat2 (V, A(), NRA, NCA)

    FOR R = 1 TO NRA
       FOR C = 1 TO NCA
          A(R, C) = V
       NEXT C
    NEXT R

END SUB

' ============================== DuplicaMat ==============================
'
'   Procedura per duplicare un array monodimensionale o bidimensionale;
'   consente anche di duplicare  un vettore colonna  in una matrice con
'   una sola colonna, e viceversa
'
'   Variabili in ingresso:
'      A()       array da duplicare
'
'   Variabili in uscita:
'      B()       array duplicato di A()
'
'   Procedure utilizzate:
'      LimitiMat
'
' ------------------------------------------------------------------------
'
SUB DuplicaMat (A(), B())

    ' determina il numero di righe e colonne degli array A e B
    ' controlla che il numero di righe e colonne sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
    IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 THEN
       CLS
       PRINT "ERRORE 1 in DuplicaMat - limiti inferiori di A o B maggiori di 1"
       END
    END IF
    IF NRA <> NRB OR NCA <> NCB THEN
       CLS
       PRINT "ERRORE 2 in DuplicaMat - numero di righe e colonne di A e B"
       PRINT "                         non compatibili tra loro"
       END
    END IF

    ' duplica l'array
    IF DA = 1 AND DB = 1 THEN
       CALL DuplicaMat1(A(), NRA, B())
    ELSEIF DA = 1 AND DB = 2 THEN
```

```
      CALL DuplicaMat2(A(), NRA, B())
   ELSEIF DA = 2 AND DB = 1 THEN
      CALL DuplicaMat3(A(), NRA, B())
   ELSE
      CALL DuplicaMat4(A(), NRA, NCA, B())
   END IF

END SUB

SUB DuplicaMat1 (A(), NRA, B())

   FOR R = 1 TO NRA
      B(R) = A(R)
   NEXT R

END SUB

SUB DuplicaMat2 (A(), NRA, B())

   FOR R = 1 TO NRA
      B(R, 1) = A(R)
   NEXT R

END SUB

SUB DuplicaMat3 (A(), NRA, B())

   FOR R = 1 TO NRA
      B(R) = A(R, 1)
   NEXT R

END SUB

SUB DuplicaMat4 (A(), NRA, NCA, B())

   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         B(R, C) = A(R, C)
      NEXT C
   NEXT R

END SUB

' ============================== IdentitaMat ==============================
'
'   Procedura per la creazione di una matrice identita' ovvero una matrice
'   quadrata con  gli elementi della diagonale principale  uguali ad uno e
'   tutti gli altri nulli
'
'   Variabili di ingresso:
'      ===
'
'   Variabili di uscita:
'      A()       matrice identita'
'
'   Procedure utilizzate:
'      LimitiMat
'
' -------------------------------------------------------------------------
```

```
SUB IdentitaMat (A())

    ' determina il numero di indici e di righe e colonne dell'array A
    ' controlla che il numero di righe e colonne di A sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    IF LRA > 1 OR LCA > 1 THEN
       CLS
       PRINT "ERRORE 1 in IdentitaMat - limiti inferiori di A maggiori di 1"
       END
    END IF
    IF NRA <> NCA THEN
       CLS
       PRINT "ERRORE 2 in IdentitaMat - matrice A non quadrata"
       END
    END IF

    ' crea matrice identita'
    FOR R = 1 TO NRA
       FOR C = 1 TO NCA
          IF R = C THEN
            A(R, C) = 1
          ELSE
            A(R, C) = 0
          END IF
       NEXT C
    NEXT R

END SUB

' ============================== InputMat ==============================
'
'   Procedura per l'input di un array monodimensionale o bidimensionale;
'   l'array viene letto da un file gia' aperto nel programma principale
'
'   Variabili di ingresso:
'      A()       array da leggere
'      F         buffer del file da cui leggere l'array
'
'   Variabili di uscita:
'      ===
'
'   Procedure utilizzate:
'      LimitiMat
'
' ----------------------------------------------------------------------------
'
SUB InputMat (A(), F)

    ' determina il numero di indici e di righe e colonne dell'array A
    ' controlla che il numero di righe e colonne di A sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    IF LRA > 1 OR LCA > 1 THEN
       CLS
       PRINT "ERRORE 1 in InputMat - limiti inferiori di A maggiori di 1"
       END
    END IF

    ' effettua l'input dell'array
```

```
    IF DA = 1 THEN
       CALL InputMat1(A(), F, NRA)
    ELSE
       CALL InputMat2(A(), F, NRA, NCA)
    END IF

END SUB

SUB InputMat1 (A(), F, NRA)

    FOR R = 1 TO NRA
       INPUT #F, A(R)
    NEXT R

END SUB

SUB InputMat2 (A(), F, NRA, NCA)

    FOR R = 1 TO NRA
       FOR C = 1 TO NCA
          INPUT #F, A(R, C)
       NEXT C
    NEXT R

END SUB

' ============================== InserMat ==============================
'
'   Procedura per l'inserimento di un array monodimensionale o bidimensionale
'   in un array di dimensioni pari o maggiori
'
'   Variabili di ingresso:
'      A()       sottoarray da inserire
'      B()       array in cui inserire
'      RI        riga iniziale
'      CI        colonna iniziale
'
'   Variabili di uscita:
'      B()       array risultante
'
'   Procedure utilizzate:
'      LimitiMat
'
' ----------------------------------------------------------------------------
'
SUB InserMat (A(), B(), RI, CI)

    ' determina il numero di indici e di righe e colonne degli array A e B
    ' controlla che il numero di righe e colonne di A e B sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
    IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 THEN
       CLS
       PRINT "ERRORE 1 in InserMat - limiti inferiori di A o B maggiori di 1"
       END
    END IF
    IF DA > DB THEN
       CLS
       PRINT "ERRORE 2 in InserMat - array di dimensioni incompatibili"
```

```
      END
    END IF
    IF (NRB - RI + 1) < NRA THEN
      CLS
      PRINT "ERRORE 3 in InserMat - sottoarray non inseribile"
      END
    END IF
    IF DA = 2 AND (NCB - CI + 1) < NCA THEN
      CLS
      PRINT "ERRORE 3 in InserMat - sottoarray non inseribile"
      END
    END IF
    IF RI < 1 OR (DB = 2 AND CI < 1) THEN
      CLS
      PRINT "ERRORE 4 in InserMat - riga o colonna iniziale minore di 1"
      END
    END IF

    ' inserisce array
    IF DA = 1 AND DB = 1 THEN
      CALL InserMat1(A(), NRA, RI, B())
    ELSEIF DA = 1 AND DB = 2 THEN
      CALL InserMat2(A(), NRA, RI, CI, B())
    ELSE
      CALL InserMat3(A(), NRA, NCA, RI, CI, B())
    END IF

END SUB

SUB InserMat1 (A(), NRA, RI, B())

    FOR R = 1 TO NRA
      B(RI - 1 + R) = A(R)
    NEXT R

END SUB

SUB InserMat2 (A(), NRA, RI, CI, B())

    FOR R = 1 TO NRA
      B(RI - 1 + R, CI) = A(R)
    NEXT R

END SUB

SUB InserMat3 (A(), NRA, NCA, RI, CI, B())

    FOR R = 1 TO NRA
      FOR C = 1 TO NCA
        B(RI - 1 + R, CI - 1 + C) = A(R, C)
      NEXT C
    NEXT R

END SUB

' ========================== InversaMat ==========================
'
'   Procedura per l'inversione di una matrice A()
'
```

```
'   Variabili di ingresso:
'      A()          matrice da invertire
'
'   Variabili di uscita:
'      A()          matrice invertita
'
'   Procedure utilizzate:
'      LimitiMat
'
' ------------------------------------------------------------------------------
'
SUB InversaMat (A())

    ' determina il numero di indici e di righe e colonne dell'array A
    ' controlla che il numero di righe e colonne di A sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    IF LRA > 1 OR LCA > 1 THEN
       CLS
       PRINT "ERRORE 1 in InversaMat - limiti inferiori di A maggiori di 1"
       END
    END IF
    IF NRA <> NCA THEN
       CLS
       PRINT "ERRORE 2 in InversaMat - matrice A non quadrata"
       END
    END IF
    N = NRA
    DIM M(N), C(N)
    ' prepara il vettore di controllo
    FOR I = 1 TO N
       M(I) = -I
    NEXT I
    '
    ' ciclo da ripetere N volte
    '
    FOR K = 1 TO N
       ' individua l'elemento maggiore, in valore assoluto, della matrice
       D = 0
       FOR I = 1 TO N
          IF M(I) < 0 THEN
             FOR J = 1 TO N
                IF M(J) < 0 THEN
                   IF ABS(D) < ABS(A(I, J)) THEN
                      R = I
                      C = J
                      D = A(R, C)
                   END IF
                END IF
             NEXT J
          END IF
       NEXT I
       IF ABS(D) < 1E-30 THEN
          CLS
          PRINT "ERRORE 3 in InversaMat - matrice singolare"
          END
       END IF
       ' scambia le colonne
       M(C) = -M(C)
       SWAP M(C), M(R)
```

```
      FOR I = 1 TO N
         C(I) = A(I, C)
         SWAP A(I, C), A(I, R)
      NEXT I
      ' divide la riga per l'elemento maggiore
      FOR J = 1 TO N
         IF J <> R THEN A(R, J) = A(R, J) / D
      NEXT J
      ' azzera i restanti termini della colonna
      FOR I = 1 TO N
         IF I <> R THEN
            FOR J = 1 TO N
               IF J <> R THEN A(I, J) = A(I, J) - C(I) * A(R, J)
            NEXT J
         END IF
      NEXT I
      ' memorizza nella colonna i fattori di riduzione
      C(R) = -1
      FOR I = 1 TO N
         A(I, R) = -C(I) / D
      NEXT I
   NEXT K
   '
   ' scambia le righe
   FOR K = 1 TO N
      FOR I = 1 TO N
         IF M(I) = K THEN
            SWAP M(I), M(K)
            FOR J = 1 TO N
               SWAP A(I, J), A(K, J)
            NEXT J
         END IF
      NEXT I
   NEXT K

END SUB

' ============================== LimitiMat ==============================
'
'   Procedura per la determinazione del numero di dimensioni
'   e di righe e colonne di un array monodimensionale o bidimensionale
'
'   Variabili di ingresso:
'      A()     array in esame
'
'   Variabili di uscita:
'      DA      numero di dimensioni dell'array (1 o 2)
'      LR      primo indice di riga
'      LC      primo indice di colonna
'      NR      secondo indice di riga
'      NC      secondo indice di colonna (se l'array e' monodimensionale vale 1)
'
'   Procedure utilizzate:
'      ==
'
'   Annotazioni:
'      la procedura non controlla se il numero di dimensioni e' superiore a 2
'
' ------------------------------------------------------------------------------
```

```
SUB LimitiMat (A(), DA, LR, LC, NR, NC)

    LR = LBOUND(A, 1)
    NR = UBOUND(A, 1)
    ON ERROR GOTO ErroreLimitiMat
       LC = LBOUND(A, 2)
       NC = UBOUND(A, 2)
    ON ERROR GOTO 0
    IF LC = 0 AND NC = 0 THEN
       DA = 1
       NC = 1
    ELSE
       DA = 2
    END IF

END SUB

' ================================ LprintMat ================================
'
'   Procedura per la stampa su carta
'   di un array monodimensionale o bidimensionale
'
'   Variabili di ingresso:
'      A()       array da stampare
'      F$        stringa di formattazione per l'output
'                (se F$="" si usa il formato libero)
'
'   Variabili di uscita:
'      ===
'
'   Procedure utilizzate:
'      OutputMat
'
' ---------------------------------------------------------------------------
'
SUB LprintMat (A(), F$)

    F = FREEFILE
    OPEN "O", #F, "PRN"
    CALL OutputMat(A(), F, F$)
    CLOSE #F

END SUB

' ================================ OutputMat ================================
'
'   Procedura per l'output di un array monodimensionale o bidimensionale;
'   l'output viene inviato a un file gia' aperto nel programma principale
'
'   Variabili di ingresso:
'      A()       array da stampare
'      F         buffer del file a cui inviare l'output
'      F$        stringa di formattazione per l'output
'                (se F$="" si usa il formato libero)
'
'   Variabili di uscita:
'      ===
'
```

```
'   Procedure utilizzate:
'      LimitiMat
'
' ---------------------------------------------------------------------------
'
SUB OutputMat (A(), F, F$)

    ' determina il numero di indici e di righe e colonne dell'array A
    ' controlla che il numero di righe e colonne di A sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    IF LRA > 1 OR LCA > 1 THEN
       CLS
       PRINT "ERRORE 1 in OutputMat - limiti inferiori di A maggiori di 1"
       END
    END IF

    ' effettua l'output dell'array
    IF DA = 1 THEN
       CALL OutputMat1(A(), F, F$, NRA)
    ELSE
       CALL OutputMat2(A(), F, F$, NRA, NCA)
    END IF

END SUB

SUB OutputMat1 (A(), F, F$, NRA)

    FOR R = 1 TO NRA
       IF F$ = "" THEN
          PRINT #F, A(R)
       ELSE
          PRINT #F, USING F$; A(R)
       END IF
    NEXT R

END SUB

SUB OutputMat2 (A(), F, F$, NRA, NCA)

    FOR R = 1 TO NRA
       FOR C = 1 TO NCA
          IF F$ = "" THEN
             PRINT #F, A(R, C),
          ELSE
             PRINT #F, USING F$; A(R, C),
          END IF
       NEXT C
       PRINT #F,
    NEXT R

END SUB

' ============================== PrintMat ==============================
'
'   Procedura per la stampa su schermo
'   di un array monodimensionale o bidimensionale
'
'   Variabili di ingresso:
'      A()       array da stampare
```

```
'     F$          stringa di formattazione per l'output
'                 (se F$="" si usa il formato libero)
'
'   Variabili di uscita:
'      ===
'
'   Procedure utilizzate:
'      OutputMat
'
' ------------------------------------------------------------------------------
'
SUB PrintMat (A(), F$)

    LOCATE 25, 1
    F = FREEFILE
    OPEN "O", #F, "CON"
    CALL OutputMat(A(), F, F$)
    CLOSE #F
    PRINT

END SUB

' ============================== ProdottoCostMat ===============================
'
'   Procedura per il prodotto di una costante per un array
'
'   Variabili di ingresso:
'      K          costante da moltiplicare
'      A()        array da moltiplicare
'
'   Variabili di uscita:
'      B()        array prodotto
'
'   Procedure utilizzate:
'      DuplicaMat
'      LimitiMat
'
' ------------------------------------------------------------------------------
'
SUB ProdottoCostMat (K, A(), B())

    ' determina il numero di indici e di righe e colonne degli array A e B
    ' controlla che il numero di righe e colonne di A e B sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
    IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 THEN
       CLS
       PRINT "ERRORE 1 in ProdottoCostMat - limiti inferiori di A o B maggiori di 1"
       END
    END IF
    IF NRA <> NRB OR NCA <> NCB THEN
       CLS
       PRINT "ERRORE 2 in ProdottoCostMat - numero di righe e colonne di A e B"
       PRINT "                              non compatibili tra loro"
       END
    END IF

    ' esegue il prodotto
    IF DA = 1 AND DB = 1 THEN
```

```
      CALL ProdottoCostMat1(K, A(), NRA, B())
   ELSEIF DA = 2 AND DB = 1 THEN
      DIM E(NRA)
      CALL DuplicaMat(A(), E())
      CALL ProdottoCostMat1(K, E(), NRA, B())
   ELSEIF DA = 1 AND DB = 2 THEN
      DIM E(NRA)
      CALL ProdottoCostMat1(K, A(), NRA, E())
      CALL DuplicaMat(E(), B())
   ELSE
    ' CALL ProdottoCostMat2(K, A(), NRA, NCA, B())
   END IF

END SUB

SUB ProdottoCostMat1 (K, A(), NRA, B())

   FOR R = 1 TO NRA
      B(R) = K * A(R)
   NEXT R

END SUB

SUB ProdottoCostMat2 (K, A(), NRA, NCA, B())

   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         B(R, C) = K * A(R, C)
      NEXT C
   NEXT R

END SUB

' ============================== ProdottoMat ==============================
'
'   Procedura per il prodotto tra due array (righe per colonne)
'
'   Variabili di ingresso:
'      A()          primo array da moltiplicare
'      B()          secondo array da moltiplicare
'
'   Variabili di uscita:
'      C()          array risultante
'
'   Procedure utilizzate:
'      DuplicaMat
'      LimitiMat
'
' -------------------------------------------------------------------------
'
SUB ProdottoMat (A(), B(), C())

   ' determina il numero di indici e di righe e colonne degli array A, B e C
   ' controlla che il numero di righe e colonne di A, B e C siano corretti
   CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
   CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
   CALL LimitiMat(C(), DC, LRC, LCC, NRC, NCC)
   IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 OR LRC > 1 OR LCC > 1 THEN
      CLS
```

```
      PRINT "ERRORE 1 in ProdottoMat - limiti inferiori di A, B o C maggiori di 1"
      END
   END IF
   IF NCA <> NRB OR NRA <> NRC OR NCB <> NCC THEN
      CLS
      PRINT "ERRORE 2 in ProdottoMat - numero di righe e colonne di A, B e C"
      PRINT "                          non compatibili tra loro"
      END
   END IF

   ' calcola il prodotto
   IF DA = 1 AND DB = 2 THEN
      CALL ProdottoMat1(A(), B(), NRA, NCB, C())
   ELSEIF DA = 2 AND DB = 1 THEN
      CALL ProdottoMat2(A(), B(), NRA, NCA, C())
   ELSE
      CALL ProdottoMat3(A(), B(), NRA, NCB, NCA, C())
   END IF

END SUB

SUB ProdottoMat1 (A(), B(), NRA, NCB, C())

   DIM D(NRA, NCB)
   FOR R = 1 TO NRA
      FOR C = 1 TO NCB
         D(R, C) = A(R) * B(1, C)
      NEXT C
   NEXT R
   CALL DuplicaMat(D(), C())

END SUB

SUB ProdottoMat2 (A(), B(), NRA, NCA, C())

   DIM D(NRA)
   FOR R = 1 TO NRA
      FOR K = 1 TO NCA
         D(R) = D(R) + A(R, K) * B(K)
      NEXT K
   NEXT R
   CALL DuplicaMat(D(), C())

END SUB

SUB ProdottoMat3 (A(), B(), NRA, NCB, NCA, C())

   DIM D(NRA, NCB)
   FOR R = 1 TO NRA
      FOR C = 1 TO NCB
         FOR K = 1 TO NCA
            D(R, C) = D(R, C) + A(R, K) * B(K, C)
         NEXT K
      NEXT C
   NEXT R
   CALL DuplicaMat(D(), C())

END SUB
```

```
' ═══════════════════════ SommaMat ═══════════════════════
'
'   Procedura per la somma di due array monodimensionali o bidimensionali
'
'   Variabili di ingresso:
'      A()    primo array da sommare
'      B()    secondo array da sommare
'
'   Variabili di uscita:
'      C()    array risultante
'
'   Procedure utilizzate:
'      DuplicaMat
'      LimitiMat
'
' ---------------------------------------------------------------------------
'
SUB SommaMat (A(), B(), C())

    ' determina il numero di indici e di righe e colonne degli array A, B e C
    ' controlla che il numero di righe e colonne di A, B e C sia corretto
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
    CALL LimitiMat(C(), DC, LRC, LCC, NRC, NCC)
    IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 OR LRC > 1 OR LCC > 1 THEN
       CLS
       PRINT "ERRORE 1 in SommaMat - limiti inferiori di A, B o C maggiori di 1"
       END
    END IF
    IF NCA <> NCB OR NCB <> NCC OR NRA <> NRB OR NRB <> NRC THEN
       CLS
       PRINT "ERRORE 2 in SommaMat - numero di righe e colonne di A, B e C"
       PRINT "                       non compatibili tra loro"
       END
    END IF

    ' effettua la somma dei due array
    IF DA = 1 AND DB = 1 THEN
       CALL SommaMat1(A(), B(), NRA, C())
    ELSEIF DA = 2 AND DB = 1 THEN
       DIM E(NRA)
       CALL DuplicaMat(A(), E())
       CALL SommaMat1(E(), B(), NRA, C())
    ELSEIF DA = 1 AND DB = 2 THEN
       DIM E(NRA)
       CALL DuplicaMat(B(), E())
       CALL SommaMat1(A(), E(), NRA, C())
    ELSE
       CALL SommaMat2(A(), B(), NRA, NCA, C())
    END IF

END SUB

SUB SommaMat1 (A(), B(), NRA, C())

   DIM D(NRA)
   FOR R = 1 TO NRA
      D(R) = A(R) + B(R)
   NEXT R
```

```
   CALL DuplicaMat(D(), C())

END SUB

SUB SommaMat2 (A(), B(), NRA, NCA, C())

   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         C(R, C) = A(R, C) + B(R, C)
      NEXT C
   NEXT R

END SUB

' ============================== SottraeMat ==============================
'
'   Procedura per la sottrazione di due array
'
'   Variabili di ingresso:
'      A()       primo array
'      B()       secondo array da sottrarre al primo
'
'   Variabili di uscita:
'      C()       array risultante
'
'   Procedure utilizzate:
'      DuplicaMat
'      LimitiMat
'
' ------------------------------------------------------------------------------
'
SUB SottraeMat (A(), B(), C())

    ' determina il numero di indici e di righe e colonne degli array A, B e C
    ' controlla che il numero di righe e colonne di A, B e C siano corrette
    CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
    CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
    CALL LimitiMat(C(), DC, LRC, LCC, NRC, NCC)
    IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 OR LRC > 1 OR LCC > 1 THEN
       CLS
       PRINT "ERRORE 1 in SottraeMat - limiti inferiori di A, B o C maggiori di 1"
       END
    END IF
    IF NCA <> NCB OR NCB <> NCC OR NRA <> NRB OR NRB <> NRC THEN
       CLS
       PRINT "ERRORE 2 in SottraeMat - numero di righe e colonne di A, B e C"
       PRINT "                         non compatibili tra loro"
       END
    END IF

    ' effettua la sottrazione
    IF DA = 1 AND DB = 1 THEN
       CALL SottraeMat1(A(), B(), NRA, C())
    ELSEIF DA = 2 AND DB = 1 THEN
       DIM E(NRA)
       CALL DuplicaMat(A(), E())
       CALL SottraeMat1(E(), B(), NRA, C())
    ELSEIF DA = 1 AND DB = 2 THEN
       DIM E(NRA)
```

```
      CALL DuplicaMat(B(), E())
      CALL SottraeMat1(A(), E(), NRA, C())
   ELSE
      CALL SottraeMat2(A(), B(), NRA, NCA, C())
   END IF

END SUB

SUB SottraeMat1 (A(), B(), NRA, C())

   DIM D(NRA)
   FOR R = 1 TO NRA
      D(R) = A(R) - B(R)
   NEXT R
   CALL DuplicaMat(D(), C())

END SUB

SUB SottraeMat2 (A(), B(), NRA, NCA, C())

   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         C(R, C) = A(R, C) - B(R, C)
      NEXT C
   NEXT R

END SUB

' ============================== TrasponeMat ==============================
'
'  Procedura per la trasposizione di una matrice
'
'  Variabili di ingresso:
'     A()       matrice da trasporre
'
'  Variabili di uscita:
'     B()       matrice trasposta
'
'  Procedure utilizzate:
'     DuplicaMat
'     LimitiMat
'
' ------------------------------------------------------------------------------
'
SUB TrasponeMat (A(), B())

   ' determina il numero di indici e di righe e colonne delle matrici A e B
   ' controlla che il numero di righe e colonne di A e B siano corretti
   CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
   CALL LimitiMat(B(), DB, LRB, LCB, NRB, NCB)
   IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LCB > 1 THEN
      CLS
      PRINT "ERRORE 1 in TrasponeMat - limiti inferiori di A o B maggiori di 1"
      END
   END IF
   IF NCA <> NRB OR NRA <> NCB THEN
      CLS
      PRINT "ERRORE 2 in TrasponeMat - numero di righe e colonne di A e B"
      PRINT "                          non compatibili tra loro"
```

```
      END
   END IF

   ' effettua la trasposizione
   IF DA = 1 THEN
      CALL TrasponeMat1(A(), NRA, B())
   ELSE
      CALL TrasponeMat2(A(), NRA, NRB, B())
   END IF

END SUB

SUB TrasponeMat1 (A(), NRA, B())

   DIM D(1, NRA)
   FOR R = 1 TO NRA
      D(1, R) = A(R)
   NEXT R
   CALL DuplicaMat(D(), B())

END SUB

SUB TrasponeMat2 (A(), NRA, NCA, B())

   DIM D(NCA, NRA)
   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         D(C, R) = A(R, C)
      NEXT C
   NEXT R
   CALL DuplicaMat(D(), B())

END SUB

' ============================== WriteMat ==============================
'
'   Procedura per memorizzare su un file i valori degli elementi
'   di un array monodimensionale o bidimensionale;
'   l'array viene scritto su un file gia' aperto nel programma principale
'
'   Variabili di ingresso:
'      A()       array da inviare al file
'      F         buffer del file a cui inviare l'array
'
'   Variabili di uscita:
'      ==
'
'   Procedure utilizzate:
'      LimitiMat
'
' ----------------------------------------------------------------------------
'
SUB WriteMat (A(), F)

   ' determina il numero di indici e di righe e colonne dell'array A
   ' controlla che il numero di righe e colonne di A sia corretto
   CALL LimitiMat(A(), DA, LRA, LCA, NRA, NCA)
   IF LRA > 1 OR LCA > 1 THEN
      CLS
```

```
      PRINT "ERRORE 1 in WriteMat - limiti inferiori di A maggiori di 1"
      END
   END IF

   ' effettua l'output su file dell'array
   IF DA = 1 THEN
      CALL WriteMat1(A(), F, NRA)
   ELSE
      CALL WriteMat2(A(), F, NRA, NCA)
   END IF

END SUB

SUB WriteMat1 (A(), F, NRA)

   FOR R = 1 TO NRA
      WRITE #F, A(R)
   NEXT R

END SUB

SUB WriteMat2 (A(), F, NRA, NCA)

   FOR R = 1 TO NRA
      FOR C = 1 TO NCA
         WRITE #F, A(R, C)
      NEXT C
   NEXT R

END SUB
```

4. Esempio

Un esempio di programma principale che utilizza le procedure qui descritte è fornito dal programma per la risoluzione di un sistema di equazioni, sviluppato solo per finalità didattiche (perché il problema viene ripreso nel capitolo successivo) e listato nel seguito. Esso prevede innanzitutto la lettura dei dati (numero di equazioni N, matrice dei coefficienti delle incognite A, termini noti B) conservati in apposite linee contraddistinte dall'istruzione DATA. Dopo aver stampato i dati, la matrice A viene duplicata in A1 e quest'ultima viene invertita. Per controllo, si effettua il prodotto A $\times$ A1 che deve fornire, a meno degli errori di arrotondamento, la matrice identità (si è effettuata la duplicazione prima dell'inversione di A proprio per non perdere la matrice di partenza e per poter effettuare il controllo). Il vettore risultato X viene infine valutato moltiplicando la matrice inversa per il vettore termini noti.

RISOLUZIONE DI UN SISTEMA DI EQUAZIONI
mediante l'inversione della matrice dei coefficienti

```
DECLARE SUB DuplicaMat (A!(), B!())
DECLARE SUB InversaMat (A!())
DECLARE SUB LprintMat (A!(), F$)
DECLARE SUB ProdottoMat (A!(), B!(), C!())

OPTION BASE 1

READ N
DIM A(N, N), A1(N, N), B(N), X(N)


' lettura della matrice dei coefficienti A e del vettore termine noto B
FOR I = 1 TO N
   FOR J = 1 TO N
      READ A(I, J)
   NEXT J
NEXT I
FOR I = 1 TO N
   READ B(I)
NEXT I

' stampa dei dati
LPRINT "Matrice dei coefficienti :"
CALL LprintMat(A(), "##.###     ")
LPRINT
LPRINT "Vettore termini noti :"
CALL LprintMat(B(), "##.###     ")
LPRINT

' inversione della matrice e controllo dell'inversione
CALL DuplicaMat(A(), A1())
CALL InversaMat(A1())
LPRINT "Inversa della matrice dei coefficienti :"
CALL LprintMat(A1(), "##.####    ")
LPRINT
CALL ProdottoMat(A(), A1(), A())
LPRINT "Controllo dell'inversione - prodotto della matrice per l'inversa :"
CALL LprintMat(A(), "##.##^^^^  ")
LPRINT

' calcolo e stampa del risultato
CALL ProdottoMat(A1(), B(), X())
LPRINT "Vettore risultato :"
CALL LprintMat(X(), "##.###     ")

END


' dati per il programma
DATA  7
DATA  3, 2, 1, 0, 0, 0, 0
DATA  2,-2,-1,-2, 0, 0, 0
DATA  1,-1, 2,-1, 1, 0, 0
DATA  0,-2,-1, 4, 2,-3, 0
```

```
DATA  0, 0, 1, 2,-1, 1,-1
DATA  0, 0, 0,-3, 1, 2,-1
DATA  0, 0, 0, 0,-1,-1, 5
DATA -1,-7, 5,-7,-2, 2, 9
```

L'output fornito dal programma è riportato nel seguito. Si noti come il prodotto della matrice A per la sua inversa A1 fornisce una matrice che si discosta leggermente dall'identità. I termini che dovrebbero essere rigorosamente nulli sono invece molto piccoli (dell'ordine di 10^{-7}) ma diversi da zero, in conseguenza delle approssimazioni connesse alla modalità di memorizzazione dei numeri reali nel calcolatore.

```
Matrice dei coefficienti :

 3.000   2.000   1.000   0.000   0.000   0.000   0.000
 2.000  -2.000  -1.000  -2.000   0.000   0.000   0.000
 1.000  -1.000   2.000  -1.000   1.000   0.000   0.000
 0.000  -2.000  -1.000   4.000   2.000  -3.000   0.000
 0.000   0.000   1.000   2.000  -1.000   1.000  -1.000
 0.000   0.000   0.000  -3.000   1.000   2.000  -1.000
 0.000   0.000   0.000   0.000  -1.000  -1.000   5.000

Vettore termini noti :

-1.000
-7.000
 5.000
-7.000
-2.000
 2.000
 9.000

Inversa della matrice dei coefficienti :

 0.2272   0.1824  -0.0464   0.0680   0.1159   0.0618   0.0355
 0.1824  -0.2090  -0.1291  -0.0440  -0.1774   0.0054  -0.0344
-0.0464  -0.1291   0.3972  -0.1159   0.0070  -0.1963  -0.0379
 0.0680  -0.0440  -0.1159   0.1700   0.2898   0.1546   0.0889
 0.1159  -0.1774   0.0070   0.2898  -0.0174   0.4907   0.0947
 0.0618   0.0054  -0.1963   0.1546   0.4907   0.5951   0.2172
 0.0355  -0.0344  -0.0379   0.0889   0.0947   0.2172   0.2624

Controllo dell'inversione - prodotto della matrice per l'inversa :

 1.00E+00  2.98E-08  0.00E+00  0.00E+00  0.00E+00 0.00E+00  1.49E-08
-1.49E-08  1.00E+00  0.00E+00 -2.98E-08  0.00E+00 0.00E+00 -2.98E-08
 0.00E+00 -4.47E-08  1.00E+00  0.00E+00 -2.98E-08 0.00E+00  7.45E-09
 0.00E+00 -7.45E-09 -5.96E-08  1.00E+00  0.00E+00 0.00E+00  0.00E+00
 2.98E-08 -4.10E-08  1.49E-08  6.71E-08  1.00E+00 1.19E-07  5.96E-08
 3.73E-08 -3.35E-08 -7.45E-08 -7.45E-09 -3.73E-08 1.00E+00  0.00E+00
-2.98E-08  1.49E-08  1.49E-08 -2.98E-08  2.98E-08 1.19E-07  1.00E+00
```

```
Vettore risultato :

-2.000
 1.000
 3.000
-1.000
 1.000
 0.000
 2.000
```

CAPITOLO TERZO

SOLUZIONE DEI SISTEMI DI EQUAZIONI LINEARI

1. Generalità

Un sistema di n equazioni lineari in n incognite, esplicitamente indicato da

$$\begin{aligned}
a_{11}\,x_1 + a_{12}\,x_2 + a_{13}\,x_3 + \ldots + a_{1n}\,x_n &= b_1 \\
a_{21}\,x_1 + a_{22}\,x_2 + a_{23}\,x_3 + \ldots + a_{2n}\,x_n &= b_2 \\
a_{31}\,x_1 + a_{32}\,x_2 + a_{33}\,x_3 + \ldots + a_{3n}\,x_n &= b_3 \\
\ldots\ \ldots + \ldots\ \ldots + \ldots\ \ldots + \ldots + \ldots\ \ldots &= \ldots \\
a_{n1}\,x_1 + a_{n2}\,x_2 + a_{n3}\,x_3 + \ldots + a_{nn}\,x_n &= b_n
\end{aligned}$$

può essere sinteticamente scritto, utilizzando la notazione matriciale, nella forma

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \ldots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \ldots \\ b_n \end{bmatrix}$$

ovvero

$$\mathbf{A}\,\mathbf{x} = \mathbf{B}$$

dove $\mathbf{A}$ è la matrice quadrata di ordine n che raccoglie i coefficienti delle incognite del sistema, $\mathbf{x}$ il vettore colonna delle incognite e $\mathbf{B}$ il vettore colonna dei termini noti.

Per le regole dell'algebra matriciale la soluzione del sistema è data dalla relazione

$$\mathbf{x} = \mathbf{A}^{-1}\,\mathbf{B}$$

cioè dal prodotto dell'inversa della matrice dei coefficienti per il vettore termini noti. L'utilizzazione delle istruzioni matriciali, e quindi delle procedure illustrate nel capitolo precedente, non è però la maniera ottimale per risolvere il sistema. Il numero di moltiplicazioni e divisioni necessarie per l'inversione di una matrice di ordine n è, come mostrato in precedenza, approssimativamente pari ad n^3. Ad esse vanno aggiunte altre n^2 moltiplicazioni per ciascun vettore di termini noti, occorrenti per effettuare il prodotto matriciale $\mathbf{A}^{-1}\,\mathbf{B}$. Esistono invece procedimenti numerici, come il metodo di riduzione di Gauss illustrato nei paragrafi successivi, che consentono di risolvere direttamente il sistema con un numero di operazioni circa tre volte inferiore.

Nelle applicazioni ai problemi strutturali vi è una ulteriore motivazione contro l'uso delle istruzioni matriciali. La matrice dei coefficienti contiene infatti una grande quantità di zeri e, nella quasi totalità dei casi, i termini non nulli sono racchiusi in una ristretta banda a cavallo della diagonale principale (ed eventualmente in fasce laterali, in dipendenza alla numerazione delle incognite). L'inversa di una matrice a banda è però una matrice piena, cioè senza termini nulli; questa caratteristica non produce quindi alcuna semplificazione, né sul numero di operazioni né sull'ingombro di memoria, nella fase di inversione. I procedimenti diretti di soluzione consentono invece di tener conto della particolare conformazione della matrice, riducendo le operazioni necessarie e l'occupazione di memoria in maniera tanto più sensibile quanto più ristretta è la banda. In tal modo si aumenta la velocità di soluzione e si ampliano considerevolmente le dimensioni massime delle strutture esaminabili con un determinato calcolatore.

Nei paragrafi successivi il metodo di riduzione viene descritto innanzitutto nelle sue generalità, cioè nel caso di matrice dei coefficienti piena (non a banda). Successivamente vengono mostrati gli adattamenti necessari per le applicazioni all'analisi strutturale, nella quale la matrice è simmetrica e con una particolare conformazione di banda.

Le procedure sviluppate per la risoluzione di un sistema di equazioni mediante il metodo di riduzione sono contenute nel file `SOLSIST.BAS`. Esse possono essere divise in due gruppi:

- procedure generali, per la risoluzione di un sistema con matrice dei coefficienti piena:
  `Triang, Risolve`;
- procedure orientate all'analisi strutturale, per la risoluzione di un sistema con matrice dei coefficienti simmetrica e con la conformazione a banda descritta nella prima parte del testo:
  `Triang.K1, Risolve.K1` per banda memorizzata in un array monodimensionale
  `Triang.K2, Risolve.K2` per banda memorizzata in un array bidimensionale.

2. Il metodo di riduzione di Gauss

2.1. Le fasi del procedimento

Il metodo di riduzione (detto anche di eliminazione o di fattorizzazione) fu proposto dal matematico tedesco Karl Friedrich Gauss all'inizio del secolo scorso. Nel corso di questi quasi duecento anni ne sono state sviluppate numerose varianti, che si possono presentare più o meno utili a seconda dei casi. Tutte si riconducono sostanzialmente all'idea di sostituire ad ogni equazione del sistema una equivalente combinazione lineare, di se stessa e di altre, riducendo la matrice dei coefficienti ad una matrice triangolare superiore, cioè nella quale siano nulli tutti i termini al di sotto della diagonale principale (*prima fase — triangolarizzazione*)

$$\begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & \dots & a'_{1n} \\ 0 & a'_{22} & a'_{23} & \dots & a'_{2n} \\ 0 & 0 & a'_{33} & \dots & a'_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a'_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ \dots \\ b'_n \end{bmatrix}$$

o sinteticamente

$$\mathbf{A'\ x = B'}$$

Il sistema così ottenuto è di immediata soluzione, partendo dall'ultima equazione che contiene una sola incognita e proseguendo poi fino alla prima,

sostituendo man mano il valore trovato per ciascuna incognita nelle equazioni precedenti (*seconda fase — sostituzione all'indietro*).

Per comprendere meglio il procedimento, si esamini in dettaglio la fase di triangolarizzazione. Il procedimento inizia sottraendo multipli della prima equazione a tutte le altre in maniera tale da azzerare per esse i coefficienti dell'incognita x_1: alla seconda equazione sarà sottratta la prima moltiplicata per a_{21}/a_{11}, alla terza la prima moltiplicata per a_{31}/a_{11}, e così via. Il termine a_{11} viene denominato *pivot* del primo passo di riduzione; i rapporti innanzi indicati possono essere chiamati *fattori di riduzione*. Si passa ora alla seconda equazione. Il valore corrente del termine a_{22} è il pivot del secondo passo di riduzione. Per annullare in tutte le equazioni successive alla seconda i coefficienti di x_2, alla terza equazione sarà sottratta la seconda moltiplicata per a_{32}/a_{22}, alla quarta la seconda moltiplicata per a_{42}/a_{22}, e così via (si noti che i termini a_{22}, a_{32}, a_{42} qui utilizzati rappresentano i valori correnti dei coefficienti delle incognite, modificati rispetto a quelli iniziali per le operazioni effettuate nel primo passo). Si utilizza quindi la terza equazione per annullare i coefficienti di x_3, e si prosegue con la quarta, la quinta ... fino alla penultima, completando con essa la triangolarizzazione.

Le operazioni da effettuare possono essere espresse analiticamente in maniera generale. Quando si utilizza l'equazione k ($k = 1 \div n-1$) per azzerare il termine a_{ik}, coefficiente di x_k nell'equazione i ($i = k+1 \div n$), devono essere calcolati i nuovi valori a'_{ij} dei termini dell'equazione ($j = k+1 \div n$) ed il nuovo termine noto b'_i con le espressioni

$$\begin{aligned} a'_{ij} &= a_{ij} - a_{kj}\frac{a_{ik}}{a_{kk}} \\ b'_i &= b_i - b_k\frac{a_{ik}}{a_{kk}} \end{aligned} \tag{3.1}$$

La fase di sostituzione all'indietro parte invece dall'ultima riga, dalla quale si ricava $x_n = b'_n/a'_{nn}$. Sostituendo il valore così trovato in tutte le altre equazioni, il termine noto di ciascuna di esse verrà modificato sottraendo al suo valore corrente il prodotto dell'incognita x_n per il relativo coefficiente. Si passa quindi alla penultima equazione, nella quale rimane ora solo l'incognita x_{n-1}. Il suo valore viene determinato e sostituito nelle altre equazioni, allo stesso modo di quanto fatto per x_n. Si prosegue con l'equazione $n-2$, poi con la $n-3$ ... fino alla prima (quando si è risolta questa non è più necessaria la sostituzione del valore trovato perché non rimangono altre equazioni).

Analiticamente, si può dire che quando si utilizza l'equazione k ($k = n \div 1$) l'incognita x_k è fornita da

$$x_k = \frac{b'_k}{a'_{kk}} \tag{3.2}$$

Sostituendo questo valore nella equazione i ($i = 1 \div k-1$, per $k > 1$), il nuovo valore b''_i del termine noto è espresso da

$$b''_i = b'_i - a'_{ik} x_k \tag{3.3}$$

Per confermare la già preannunciata convenienza del procedimento, si deve calcolare il numero di operazioni necessarie (contando solo le moltiplicazioni e divisioni, che richiedono al calcolatore un tempo nettamente maggiore rispetto ad addizioni e sottrazioni).

Per azzerare il termine a_{i1}, coefficiente della x_1 nell'equazione i, occorrono una divisione per calcolare il fattore di riduzione ed $n-1$ moltiplicazioni per determinare i nuovi valori dei coefficienti delle altre incognite nell'equazione (cioè un totale di n operazioni), nonché una moltiplicazione per aggiornare il termine noto. Per azzerare i coefficienti della x_1 in tutte le $n-1$ equazioni successive alla prima, sono quindi necessarie $n(n-1) = n^2 - n$ operazioni su **A** ed $n-1$ su **B**. Analogamente, per azzerare i coefficienti di x_2 occorrono $(n-1)^2 - (n-1)$ operazioni su **A** ed $n-2$ su **B**, e così via. Nella fase di triangolarizzazione si effettuano dunque $n^2 - n + (n-1)^2 - (n-1) + \ldots = (n^3 - n)/3$ operazioni su **A** ed $n - 1 + n - 2 + \ldots = (n^2 - n)/2$ su **B**.

Nella seconda fase, occorre una divisione per determinare il valore di x_n ed $n-1$ moltiplicazioni per sostituirlo nelle restanti equazioni, cioè in tutto n operazioni. Analogamente ne servono $n-1$ per x_{n-1}, e così via, per un totale di $n + (n-1) + \ldots = (n^2 + n)/2$ operazioni su **B**.

La soluzione del sistema richiede pertanto l'effettuazione di circa $n^3/3$ operazioni su **A** ed n^2 su **B**.

2.2. Riduzione e fattorizzazione

Prima di procedere ulteriormente, è importante evidenziare alcune ulteriori caratteristiche del metodo di Gauss. Si prenda nuovamente in esame il primo passo della fase di triangolarizzazione. Si è visto che per annullare il termine a_{21} è necessario sottrarre alla seconda riga la prima moltiplicata per il fattore di riduzione a_{21}/a_{11}. Si può facilmente controllare che questa operazione equivale a premoltiplicare la matrice **A** di partenza per una

matrice elementare $\mathbf{E}_{21}$, che si differenzia dalla matrice identità solo per la presenza del termine $e_{21} = -a_{21}/a_{11}$

$$\mathbf{E}_{21} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & \dots & 0 & \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

In maniera analoga, l'azzeramento del generico termine a_{ik} equivale a premoltiplicare la matrice $\mathbf{A}$ corrente (cioè già modificata nei passi precedenti) per la matrice $\mathbf{E}_{ik}$, caratterizzata dal termine $e_{ik} = -a_{ik}/a_{kk}$. Si ha pertanto

$$\mathbf{A}' = \mathbf{E}_{n\ n-1} \dots \mathbf{E}_{ik} \dots \mathbf{E}_{31}\mathbf{E}_{21}\mathbf{A}$$

Questa relazione può essere invertita scrivendo

$$\mathbf{A} = \mathbf{E}_{21}^{-1}\mathbf{E}_{31}^{-1} \dots \mathbf{E}_{ik}^{-1} \dots \mathbf{E}_{n\ n-1}^{-1}\mathbf{A}'$$

ovvero

$$\mathbf{A} = \mathbf{L}\mathbf{A}'$$

ed è facile constatare che l'inversione di $\mathbf{E}_{ik}$ comporta soltanto il cambiare segno all'elemento e_{ik} e che il prodotto delle inverse delle matrici elementari fornisce la matrice $\mathbf{L}$, triangolare inferiore, che contiene tutti i fattori di riduzione

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & \dots & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}}{a_{22}} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \frac{a_{n1}}{a_{11}} & \frac{a_{n2}}{a_{22}} & \frac{a_{n3}}{a_{33}} & \dots & 1 \end{bmatrix}$$

La matrice $\mathbf{A}'$ può essere a sua volta scomposta nel prodotto di una matrice diagonale $\mathbf{D}$ ed una triangolare superiore con valori unitari nella diagonale principale $\mathbf{U}$

$$\mathbf{A}' = \mathbf{D}\ \mathbf{U}$$

con

$$\mathbf{D} = \begin{bmatrix} a'_{11} & 0 & 0 & \dots & 0 \\ 0 & a'_{22} & 0 & \dots & 0 \\ 0 & 0 & a'_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a'_{nn} \end{bmatrix}$$

e

$$\mathbf{U} = \begin{bmatrix} 1 & \frac{a'_{12}}{a'_{11}} & \frac{a'_{13}}{a'_{11}} & \dots & \frac{a'_{1n}}{a'_{11}} \\ 0 & 1 & \frac{a'_{23}}{a'_{22}} & \dots & \frac{a'_{2n}}{a'_{22}} \\ 0 & 0 & 1 & \dots & \frac{a'_{3n}}{a'_{33}} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

In definitiva si può scrivere

$$\mathbf{A} = \mathbf{L}\,\mathbf{D}\,\mathbf{U} \tag{3.4}$$

evidenziando così che il procedimento di riduzione equivale alla scomposizione della matrice **A** in tre fattori, una matrice triangolare inferiore **L**, una diagonale **D** ed una triangolare superiore **U**. Per tale motivo esso è anche denominato metodo di fattorizzazione.

È possibile dimostrare (ma il farlo esula dai compiti del presente testo) che questa scomposizione è unica, cioè che non esiste una differente terna di matrici dello stesso tipo che, moltiplicate nello stesso ordine, diano ancora la matrice **A**. Si noti però che è possibile effettuare il procedimento di riduzione all'incontrario, azzerando i termini al di sopra della diagonale principale. Si giunge in tal caso alla scomposizione

$$\mathbf{A} = \mathbf{U}'\,\mathbf{D}'\,\mathbf{L}' \tag{3.5}$$

essendo $\mathbf{U}'$ una matrice triangolare superiore, $\mathbf{D}'$ una matrice diagonale ed $\mathbf{L}'$ una triangolare inferiore. Queste tre matrici sono differenti dalle **L D U**, ma la scomposizione è ancora una volta unica.

2.3. Pivotaggio parziale

Il procedimento descritto cade in difetto se in un qualsiasi passo k della fase di triangolarizzazione il pivot a_{kk} è nullo. In tal caso, infatti, non ha

senso calcolare il rapporto a_{ik}/a_{kk} e non è possibile azzerare i coefficienti dell'incognita x_k nelle equazioni successive sommando ad esse multipli dell'equazione k. Se però in una di queste equazioni, ad esempio nella m, il termine a_{mk} è diverso da zero, per ottenere un nuovo pivot non nullo basta modificare l'ordine delle equazioni, scambiando tra loro la k con la m. Se invece il coefficiente di x_k è nullo in tutte le equazioni successive a k il sistema non è risolubile e la matrice dei coefficienti è una matrice singolare. Questa situazione si riscontra nelle applicazioni strutturali quando si esamina una struttura labile.

Un secondo problema che può riscontrarsi nella soluzione del sistema è legato alle inevitabili approssimazioni del calcolo. Ciascun elaboratore opera su valori memorizzati con un numero limitato di cifre significative, e quindi troncati o arrotondati. L'effettuazione delle numerose operazioni richieste per la soluzione del sistema può comportare un propagarsi ed incrementarsi degli errori di troncamento, in particolare quando i pivot hanno valori piccoli rispetto agli altri termini della matrice dei coefficienti. Nell'impostazione di una procedura generale per la risoluzione di un sistema di equazioni, oltre alla necessità di evitare pivot nulli si pone anche l'opportunità di rendere più grande possibile il loro valore. A tal fine, nel generico passo k della fase di triangolarizzazione si possono passare in rassegna tutte le equazioni, da k ad n, per individuare quella nella quale si ha il coefficiente di x_k massimo in valore assoluto. Tale equazione verrà quindi scambiata con la k, così come fatto per evitare pivot nulli.

Le manipolazioni effettuate sul sistema per ottenere pivot idonei sono denominate in generale pivotaggio. Il procedimento innanzi descritto, che richiede solo una modifica dell'ordine delle equazioni, è chiamato *pivotaggio parziale*, per distinguerlo da un procedimento più complesso, il *pivotaggio totale*. Quest'ultimo prevede che si assuma come pivot del passo k il massimo tra i coefficienti di tutte le incognite da k ad n, effettuando quindi non solo uno scambio delle equazioni ma anche un'inversione dell'ordine delle incognite. Tale tecnica può ridurre ulteriormente l'effetto degli errori di troncamento, ma è meno usata della precedente a causa dell'incremento dei tempi di soluzione conseguente al gran numero di confronti necessario per individuare ciascun pivot.

2.4. Risoluzioni multiple

La necessità di risolvere lo stesso sistema di equazioni più volte, con differenti colonne di termini noti, si presenta frequentemente nelle appli-

cazioni, in particolare per i problemi strutturali nei quali si devono usualmente valutare gli effetti di più schemi di carichi.

Un primo modo di affrontare il problema consiste nel memorizzare i termini noti in un array bidimensionale, cioè in una matrice con più colonne; è in tal caso possibile effettuare la triangolarizzazione e la sostituzione all'indietro contemporaneamente su tutte le colonne, ottenendo come risultato tutti i vettori soluzione.

Questo modo di procedere appare indubbiamente semplice e compatto, ma può risultare più o meno conveniente a seconda dell'organizzazione complessiva del programma, ed in particolare delle fasi di input e output.

L'unificazione della fase di calcolo risulta spontanea se si è scelto di presentare in maniera unitaria dati e risultati di tutte le condizioni di carico. In tal caso si procede a caricare in memoria centrale i dati dei diversi schemi di carico, preparare la matrice dei termini noti e calcolare i risultati. L'aspetto negativo è costituito dall'elevato ingombro di memoria, necessario non tanto per conservare le colonne aggiuntive, quanto per memorizzare l'intero insieme di dati di carico e di risultati di tutte le condizioni. L'ingombro può essere ridotto conservando le informazioni nella memoria di massa e richiamandole man mano, ma ciò allunga, spesso anche in misura notevole, i tempi di elaborazione.

Un'impostazione alternativa molto diffusa consiste nel presentare dati e risultati separatamente, schema per schema. In tal caso conviene operare sul singolo vettore termini noti ed è quindi necessario, per minimizzare il lavoro, scindere nettamente le due fasi del procedimento. Si può così effettuare quella di triangolarizzazione un'unica volta, solo sulla matrice dei coefficienti, memorizzando i fattori di riduzione. In un secondo momento, in maniera distinta per ciascuna colonna di termini noti, si ripetono su quest'ultima le stesse operazioni utilizzando i fattori memorizzati e si effettua quindi la sostituzione all'indietro.

Si può facilmente dimostrare che questa via non comporta in pratica alcun aggravio, di ingombro di memoria o di tempo, rispetto all'alternativa unitaria precedentemente descritta. Il generico passo k della fase di triangolarizzazione comporta l'azzeramento dei termini a_{ik} (con $i > k$) della matrice dei coefficienti, ottenuto sottraendo alla riga i la riga k moltiplicata per il fattore di riduzione a_{ik}/a_{kk}. Il valore del fattore deve essere salvato per poter poi ripetere la stessa operazione anche sul vettore termine noto. Poiché è ovviamente superfluo memorizzare il valore del termine a_{ik}, che si sa essere diventato nullo, è possibile conservare al suo posto il corrispondente fattore di riduzione. Se il procedimento prevede un pivotaggio parziale è necessario mantenere anche l'indicazione dello scambio di righe

effettuato, e quindi introdurre un ulteriore vettore `IPVT`, il cui elemento k conterrà il numero d'ordine della riga che è stata spostata in posizione k (sono sufficienti $n-1$ elementi, perché la riga posta in ultima posizione viene univocamente determinata una volta individuate tutte le altre).

Nel testo si è seguita questa seconda impostazione. Pertanto per la risoluzione di un sistema di equazioni generico, con matrice dei coefficienti piena, sono fornite le due procedure, `Triang` e `Risolve`, che effettuano rispettivamente la prima e la seconda fase del procedimento.

3. Applicazione all'analisi strutturale

3.1. Ordine di esecuzione del procedimento

Alla fine del paragrafo 2.2 si è accennato alla possibilità di procedere alla riduzione in una maniera leggermente diversa, azzerando nella fase di triangolarizzazione i termini posti al di sopra della diagonale principale anziché quelli al di sotto di essa. Ciò richiede l'inversione dell'ordine in cui si opera sulle righe; si deve partire dall'ultima, assumendo come pivot il termine a_{nn} ed azzerando nelle altre equazioni i coefficienti dell'incognita x_n; si passa poi alla penultima, e così via, fino alla seconda. Viceversa, la sostituzione all'indietro deve partire dalla prima riga, nella quale compare ora solo l'incognita x_1, e procedere con la seconda, la terza ... fino all'ultima. Le espressioni utilizzate per l'aggiornamento dei coefficienti sono però ancora le (3.1), (3.2) e (3.3).

Seguire un'ordine o l'altro per un generico sistema di equazioni è sostanzialmente indifferente ai fini operativi, perché il risultato ottenuto e l'onere computazionale sono identici. È solo per motivi "storici" che il procedimento di riduzione è presentato nell'ordine utilizzato nel paragrafo 2.1.

Passando più specificamente alle applicazioni strutturali, si può notare che la triangolarizzazione è richiesta oltre che per la risoluzione delle equazioni di equilibrio del sistema anche per la determinazione della matrice di rigidezza laterale di un telaio piano a partire dalla matrice completa. In questo secondo caso, avendo numerato le incognite ponendo in prima posizione quelle relative alla traslazione dei traversi bisogna necessariamente procedere dalla fine verso l'inizio ed arrestare anzitempo il procedimento, escludendo da esso il blocco relativo ai traversi. Nel caso in esame, con la specifica impostazione data al problema strutturale (che sarebbe potuta essere anche diversa), è quindi preferibile adottare l'ordine inverso rispetto a

quello classico ed organizzare la procedura di triangolarizzazione inserendo in essa una variabile di controllo che consenta l'arresto della riduzione ad una riga prefissata.

Si osserva inoltre che nella risoluzione di un singolo schema piano la ripetizione delle operazioni di triangolarizzazione sui termini noti e la sostituzione all'indietro vengono effettuate in maniera unitaria e completa. Il passaggio alla matrice di rigidezza laterale, necessario per l'analisi dell'insieme spaziale di telai piani, richiede invece la separazione di questi due blocchi operativi e la distinzione tra incognite globali (spostamenti e rotazioni degli impalcati) e locali (rotazioni e spostamenti verticali dei nodi). La procedura di risoluzione è quindi anch'essa gestita da una variabile di controllo che consente l'effettuazione delle sole operazioni realmente necessarie.

3.2. Riduzione di una matrice simmetrica

La simmetria è una delle principali caratteristiche delle matrici strutturali. Grazie ad essa, è possibile individuare in maniera completa una matrice conservandone soltanto la diagonale principale ed il triangolo superiore (o quello inferiore), ed è quindi dimezzato l'ingombro di memoria necessario. A prima vista, non sembrerebbe possibile sfruttare questa caratteristica nella organizzazione del procedimento di riduzione, a causa della necessità di memorizzare i fattori di riduzione nel triangolo azzerato della matrice. È però facile constatare, come conseguenza del principio di unicità della scomposizione citato nel paragrafo 2.2, che la (3.5) diventa, grazie alla simmetria [1]

$$\mathbf{A} = \mathbf{U}' \, \mathbf{D}' \, \mathbf{U}'^{T} \tag{3.6}$$

cioè che la matrice triangolare inferiore ottenuta come risultato della riduzione è la trasposta della matrice triangolare superiore che contiene i fattori di riduzione, ed è pertanto sufficiente conservare solo una delle due.

Per meglio comprendere in che modo operare su una matrice memorizzata nella diagonale principale e nel triangolo superiore, si riesamini innanzitutto la fase di triangolarizzazione. Avendo invertito l'ordine, il pivot del primo passo è il termine a_{nn}. Per annullare il coefficiente di x_n

[1] Infatti, essendo $A = U' \, D' \, L'$ deve essere, per le proprietà della trasposizione, $A^T = L'^T \, D'^T \, U'^T$; poiché per la simmetria si ha $A = A^T$, il principio di unicità comporta che $U' = L'^T$, $D' = D'^T$, $L' = L'^T$.

nell'equazione i, ai coefficienti di essa devono essere sottratti i corrispondenti termini dell'equazione n, moltiplicati per il fattore di riduzione a_i/a_n. Questo complesso di operazioni non altera la simmetria della sottomatrice ottenuta escludendo la riga e la colonna n. Infatti il nuovo valore a'_{ij} del coefficiente j–esimo dell'equazione i vale

$$a'_{ij} = a_{ij} - a_{nj}\frac{a_{in}}{a_{nn}} \tag{3.7}$$

ed è uguale al nuovo valore a'_{ji} del coefficiente i–esimo dell'equazione j

$$a'_{ji} = a_{ji} - a_{ni}\frac{a_{jn}}{a_{nn}}$$

poiché $a_{ij} = a_{ji}$, $a_{in} = a_{ni}$, $a_{jn} = a_{nj}$. Ciò evita il ricorso al triangolo inferiore e contemporaneamente consente una riduzione del numero di operazioni da effettuare. Infatti, procedendo all'azzeramento dei coefficienti di x_n facendo variare i da $n-1$ ad 1, nell'ordine, i nuovi valori dei termini a_{ij} del triangolo inferiore (per $j < i$) possono essere memorizzati in a_{ji} del triangolo superiore. Inoltre, l'espressione (3.7) deve essere applicata solo per j che va da 1 ad i, perché i termini corrispondenti a valori maggiori di j sono già stati determinati in precedenza. In generale, quindi, quando si utilizza l'equazione k ($k = n \div 2$) per azzerare il termine a_{ik}, coefficiente di x_k nell'equazione i ($i = k-1 \div 1$), si calcolano i nuovi valori a'_{ji} ($j = 1 \div i$) con l'espressione

$$a'_{ji} = a_{ji} - a_{jk}\frac{a_{ik}}{a_{kk}} \tag{3.8}$$

nella quale tutti i termini si riferiscono al triangolo superiore perché $k > i \geq j$, e si memorizza in a_{ik} il fattore di riduzione a_{ik}/a_{kk}.

L'aggiornamento della colonna dei termini noti e la successiva fase di sostituzione all'indietro richiedono l'effettuazione delle stesse operazioni già descritte in precedenza, indipendentemente dalla simmetria della matrice. Occorre solo osservare che i coefficienti a_{ji} ora determinati appartengono alla matrice $\mathbf{U}'^T$ che ha valori unitari nella diagonale principale, e quindi sono pari al rapporto tra quelli ottenuti col procedimento descritto per matrice non simmetrica ed i valori della diagonale principale. Di conseguenza, anche i coefficienti della colonna dei termini noti devono essere divisi per tali valori prima di iniziare la sostituzione all'indietro e non nel corso di questa fase.

Il numero di operazioni necessarie per la triangolarizzazione di una matrice dei coefficienti simmetrica può essere valutato riesaminando rapidamente il procedimento seguito. Nel generico passo che prende per base

l'equazione k occorre effettuarne k per azzerare $a_{k-1\ k}$ (1 divisione per calcolare il fattore di riduzione e $k-1$ moltiplicazioni per aggiornare i coefficienti delle altre incognite), $k-1$ per $a_{k-2\ k}$, e così via, fino a 2 per azzerare a_{1k}, per un totale di $(k^2+k-2)/2$. Sommando i valori che tale espressione fornisce al variare di k da n a 2 si ottiene un totale di $(n^3+3n^2-4n)/6$, che è leggermente superiore alla metà di quelle necessarie in assenza di simmetria. Rimane invece invariato il numero di operazioni da effettuare sul vettore termini noti per la triangolarizzazione e la sostituzione all'indietro (n^2).

Nel procedimento finora descritto non si è ancora accennato alla possibilità di avere pivot nulli o molto piccoli ed alla conseguente opportunità di un pivotaggio. Bisogna subito osservare che lo scambio di righe altera la simmetria della matrice se non si effettua in contemporanea anche uno scambio di colonne (e quindi un'inversione dell'ordine delle incognite), e questo renderebbe più oneroso il pivotaggio. Tutte le applicazioni strutturali hanno però il vantaggio di condurre a matrici che oltre ad essere simmetriche sono anche definite positive. Si può dimostrare che in tal caso i pivot sono sempre maggiori di zero, e pertanto non è necessario effettuare uno scambio delle righe.

3.3. Riduzione di una matrice a banda

L'altra fondamentale caratteristica delle matrici strutturali, evidenziata nel capitolo 3 della prima parte del testo, è quella di presentare un gran numero di termini nulli. Nel caso di telai piani con traversi inestensibili, in conseguenza al criterio prescelto per la numerazione delle incognite i termini diversi da zero si trovano racchiusi in una banda diagonale e due fasce laterali, lungo il bordo sinistro e quello superiore (fig.3.1). Scomponendo la matrice di rigidezza della struttura, $\mathbf{K}$, in quattro sottomatrici, $\mathbf{K}^{tt}$, $\mathbf{K}^{ta}$, $\mathbf{K}^{at}$, $\mathbf{K}^{aa}$, le prime tre risultano sostanzialmente piene mentre la quarta (che è in genere nettamente maggiore delle altre) è una tipica matrice a banda, per la quale si indica nel seguito con s_b l'ampiezza della semibanda. Si usano inoltre i simboli n_t e n_a per indicare rispettivamente l'ordine delle matrici $\mathbf{K}^{tt}$ e $\mathbf{K}^{aa}$.

La simmetria consente di memorizzare solo la metà superiore di $\mathbf{K}$, e quindi l'intera $\mathbf{K}^{ta}$ e le metà superiori di $\mathbf{K}^{tt}$ e $\mathbf{K}^{aa}$. $\mathbf{K}^{tt}$ è sempre molto piccola rispetto alle altre e si preferisce usare per essa un array bidimensionale anche se metà di questo resta inutilizzato e quindi sprecato.

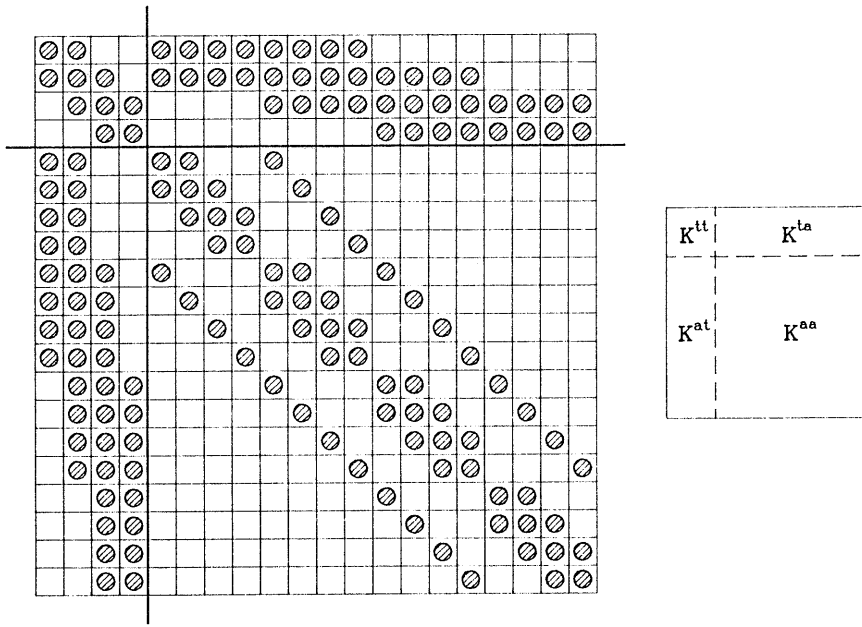


Fig. 3.1 — Scomposizione della matrice di rigidezza $\mathbf{K}$ di una struttura con traversi inestensibili

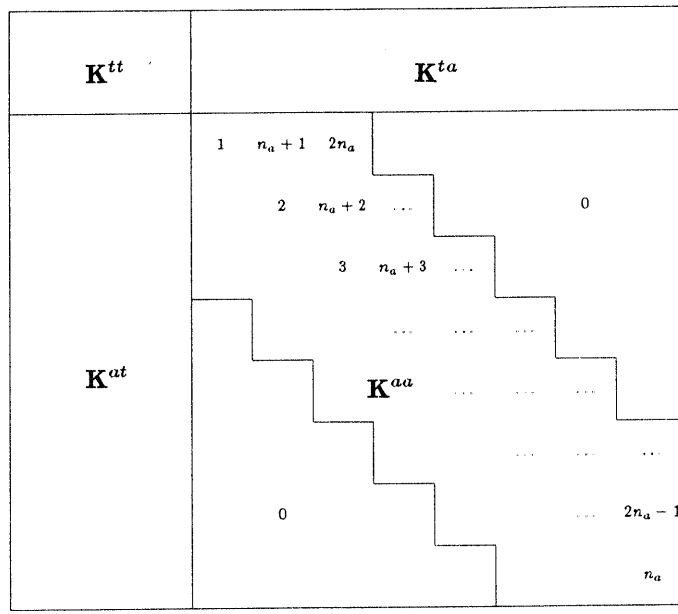


Fig. 3.2 — Numerazione progressiva degli elementi di $\mathbf{K}^{aa}$ in un array monodimensionale

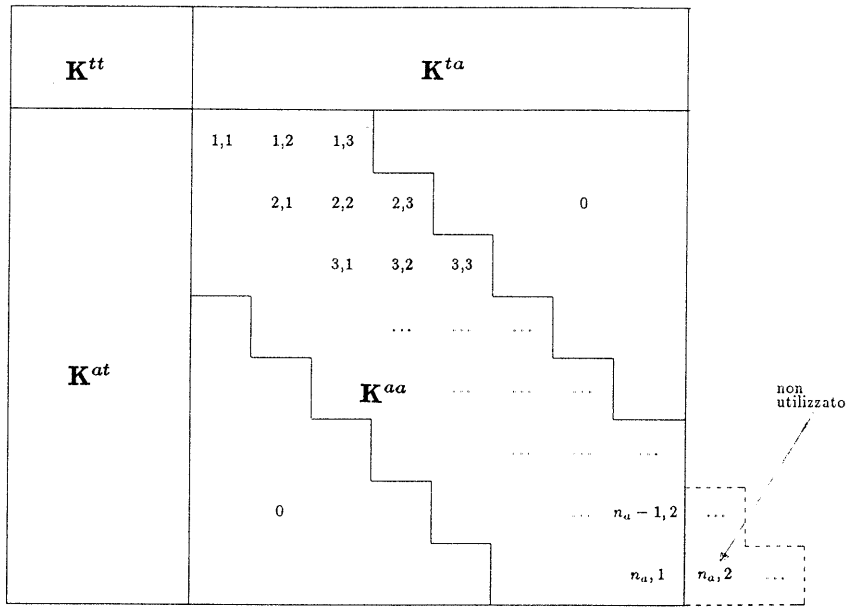


Fig. 3.3 — Numerazione progressiva degli elementi di $\mathbf{K}^{aa}$ in un array bidimensionale

$\mathbf{K}^{aa}$ è invece grande ed è quindi fondamentale sfruttare le caratteristiche di simmetria e di banda per minimizzare l'ingombro di memoria. A tal fine sono possibili due alternative: conservare gli elementi della banda in un array monodimensionale (fig.3.2) o in uno bidimensionale (fig.3.3). La prima scelta appare teoricamente la migliore, perché è necessario un numero di elementi esattamente pari a quello dei coefficienti racchiusi nella semibanda, $n_a\ s_b - s_b\ (s_b - 1)/2$; la seconda via richiede invece un array di n_a righe e s_b colonne, cioè $n_a\ s_b$ elementi, con un maggior ingombro che può diventare rilevante quando la semibanda è ampia (addirittura quasi il doppio se $s_b = n_a$). Il Quick Basic pone però un limite al massimo valore di ciascun indice e rende quindi impossibile utilizzare array monodimensionali con più di 32768 elementi, mentre con array bidimensionali se ne possono avere 32768×32768. Nel caso specifico la seconda scelta consente di sfruttare meglio la memoria disponibile del calcolatore. Essendo questa una situazione contingente, legata al linguaggio scelto e quindi suscettibile di evoluzioni future, si esaminano comunque entrambe le alternative e si riportano per esse due distinte serie di procedure (di triangolarizzazione e di risoluzione).

Nel caso dell'array monodimensionale, gli elementi sono numerati considerando prima quelli della diagonale principale e poi via via quelli delle

diagonali successive. Il generico elemento posto nella riga i, colonna k (con $d = k - i \geq 0$) è quindi memorizzato in posizione $i + d[n_a - (d-1)/2]$. Nell'array bidimensionale gli elementi rispettano invece l'organizzazione per righe della matrice $\mathbf{K}^{aa}$. L'elemento anzidetto viene quindi a trovarsi ancora nella riga i ma col secondo indice pari a $k - i + 1$.

La triangolarizzazione procede sempre dal basso verso l'alto, ed interessa quindi prima $\mathbf{K}^{aa}$, poi $\mathbf{K}^{tt}$. Per l'azzeramento di ciascun termine vale ancora sostanzialmente la relazione (3.8), che deve essere però applicata tenendo conto sia della modalità di memorizzazione prescelta che della presenza dei termini nulli.

Nel passo che prende come base la riga k di $\mathbf{K}^{aa}$ si devono annullare innanzitutto $s_b - 1$ (oppure $k-1$, se $k < s_b$) coefficienti di $\mathbf{K}^{aa}$. In particolare, per azzerare K^{aa}_{ik} la relazione (3.8), deve essere applicata a tutti gli elementi della parte di riga appartenente a $\mathbf{K}^{at}$ ed a quelli di $\mathbf{K}^{aa}$ compresi tra la colonna $k - s_b + 1$ (perché nella riga k il primo elemento della banda occupa tale posizione) e la colonna i. Si ha pertanto

$$K^{ta\prime}_{ji} = K^{ta}_{ji} - K^{ta}_{jk}\frac{K^{aa}_{ik}}{K^{aa}_{kk}}$$

$$K^{aa\prime}_{ji} = K^{aa}_{ji} - K^{aa}_{jk}\frac{K^{aa}_{ik}}{K^{aa}_{kk}} \tag{3.9}$$

Si devono inoltre annullare n_t coefficienti di $\mathbf{K}^{ta}$. Per azzerare K^{ta}_{ik} occorre aggiornare solo i primi i coefficienti della riga i di $\mathbf{K}^{tt}$, perché i successivi (e tutti quelli di $\mathbf{K}^{ta}$) sono stati già modificati nell'azzerare termini precedenti. Si ha così

$$K^{tt\prime}_{ji} = K^{tt}_{ji} - K^{ta}_{jk}\frac{K^{ta}_{ik}}{K^{aa}_{kk}} \tag{3.10}$$

Queste espressioni devono essere ulteriormente modificate per quanto riguarda $\mathbf{K}^{aa}$ per sostituire ai due indici che per esso compaiono un unico indice nel caso di memorizzazione in array monodimensionale o due nell'altro caso, con le regole innanzi definite.

La relazione da usare nella triangolarizzazione della $\mathbf{K}^{tt}$ è invece ottenuta direttamente sostituendo nelle (3.8) il simbolo K^{tt} al posto del simbolo a

$$K^{tt\prime}_{ji} = K^{tt}_{ji} - K^{tt}_{jk}\frac{K^{tt}_{ik}}{K^{tt}_{kk}} \tag{3.11}$$

La ripetizione delle operazioni sul vettore termine noto e la successiva fase di sostituzione all'indietro risentono anch'esse solo in misura modesta della divisione della matrice in quattro blocchi. Le variazioni da apportare rispetto alla impostazione base possono quindi essere riscontrate direttamente esaminando il listato delle procedure.

4. Codifica

```
'                                SOLSIST.BAS
'
'          procedure per la soluzione dei sistemi di equazioni lineari
'
'                                                              rev. 09.89
' ================================ Risolve ================================
'
'   Risoluzione di un sistema di equazioni col metodo di riduzione di Gauss
'       Seconda fase - risoluzione mediante sostituzione all'indietro
'
'       NOTA: la matrice dei coefficienti e' stata resa triangolare superiore
'             effettuando un pivotaggio parziale,  cioe' uno scambio di righe
'             per rendere massimo il termine sulla diagonale principale
'
'   Variabili di ingresso:
'      A(,)       matrice dei coefficienti delle incognite, triangolarizzata
'      B()        vettore termini noti
'      IPVT()     indice della riga del pivot K-esimo
'
'   Variabili di uscita:
'      B()        vettore soluzione
'
'   Variabili interne principali:
'      N          ordine della matrice dei coefficienti
'
' ---------------------------------------------------------------------------
'
SUB Risolve (A(), B(), IPVT())

    ' determina il numero di elementi degli array A, B ed IPVT
    LRA = LBOUND(A, 1)
    LCA = LBOUND(A, 2)
    NRA = UBOUND(A, 1)
    NCA = UBOUND(A, 2)
    LRB = LBOUND(B, 1)
    NRB = UBOUND(B, 1)
    LRI = LBOUND(IPVT, 1)
    NRI = UBOUND(IPVT, 1)

    ' controlla che le dimensioni degli array A, B ed IPVT siano corrette
    IF LRA > 1 OR LCA > 1 OR LRB > 1 OR LRI > 1 THEN
       PRINT "ERRORE 1 in Risolve - limiti inferiori di A, B o IPVT maggiori di 1"
       END
    END IF
    IF NRA <> NCA THEN
       PRINT "ERRORE 2 in Risolve - matrice dei coefficienti A non quadrata"
       END
```

```
    END IF
    IF NRB < NRA OR NRI < NRA - 1 THEN
       PRINT "ERRORE 3 in Risolve - array B o IPVT insufficiente"
       END
    END IF

    ' definisce N, ordine della matrice dei coefficienti
    N = NRA

    ' ripete su B le operazioni fatte per triangolarizzare A
    ' NOTA: il ciclo non viene effettuato nel caso che N = 1
    FOR K = 1 TO N - 1
       SWAP B(K), B(IPVT(K))
       FOR I = K + 1 TO N
          B(I) = B(I) - A(I, K) * B(K)
       NEXT I
    NEXT K
    ' effettua la sostituzione all'indietro per tutte le incognite
    FOR K = N TO 1 STEP -1
       B(K) = B(K) / A(K, K)
       ' NOTA: il ciclo non viene effettuato nel caso che K = 1
       FOR I = 1 TO K - 1
          B(I) = B(I) - A(I, K) * B(K)
       NEXT I
    NEXT K

END SUB

' ============================== Risolve.K1 ==============================
'
'   Risoluzione di un sistema di equazioni col metodo di riduzione di Gauss
'       Seconda fase - risoluzione mediante sostituzione all'indietro
'
'       NOTE: la matrice dei coefficienti K e' divisa in 4: KTT, KTA, KAT, KAA
'             la matrice dei coefficienti e' simmetrica
'             la matrice dei coefficienti e' a banda (in array monodimensionale)
'             la matrice K e' stata resa triangolare inferiore, senza pivotaggio
'
'   Variabili di ingresso:
'      KTT(,)     matrice dei coefficienti, triangolarizzata superiormente
'      KTA(,)
'      KAA()        "     "       "       , a banda (dimensione NAxSB-SBx(SB-1)/2)
'      SB         ampiezza della semibanda (compreso elemento diagonale)
'      RID        variabile di controllo del procedimento di risoluzione:
'                    RID = 0     triangolarizz. termini noti e sostituz. indietro
'                    RID = 1     triangolarizz. termini noti, esclusi i primi NT
'                    RID = 2     sostituzione indietro esclusi i primi NT termini
'      B()        vettore termini noti
'
'   Variabili di uscita:
'      B()        vettore soluzione
'
'   Variabili interne principali:
'      NT         ordine della matrice KTT
'      NA         ordine della matrice KAA
'      N          ordine della matrice dei coefficienti: N=NT+NA
'
' ------------------------------------------------------------------------
'
```

```
SUB Risolve.K1 (KTT(), KTA(), KAA(), SB, RID, B())

    ' determina il numero di elementi degli array KTT, KTA, KAA, B
    LRTT = LBOUND(KTT, 1)
    LCTT = LBOUND(KTT, 2)
    NRTT = UBOUND(KTT, 1)
    NCTT = UBOUND(KTT, 2)
    LRTA = LBOUND(KTA, 1)
    LCTA = LBOUND(KTA, 2)
    NRTA = UBOUND(KTA, 1)
    NCTA = UBOUND(KTA, 2)
    LRAA = LBOUND(KAA, 1)
    NRAA = UBOUND(KAA, 1)
    LRB = LBOUND(B, 1)
    NRB = UBOUND(B, 1)

    ' controlla che le dimensioni degli array KTT, KTA, KAA siano corrette
    ' definisce NT, NA, N, ordine delle matrici dei coefficienti
    IF LRTT > 1 OR LCTT > 1 OR LRTA > 1 OR LCTA > 1 OR LRAA > 1 THEN
       PRINT "ERRORE 1 in Risolve.K1 - limiti inferiori di KTT, KTA o KAA maggiori
              di 1"
       END
    END IF
    IF NRTT <> NCTT THEN
       PRINT "ERRORE 2 in Risolve.K1 - matrice KTT non quadrata"
       END
    END IF
    NT = NRTT
    NA = NCTA
    N = NT + NA
    IF NRAA <> NA * SB - SB * (SB - 1) / 2 THEN
       PRINT "ERRORE 3 in Risolve.K1 - dimensioni di KAA non corrette"
       END
    END IF
    IF NRTA <> NT THEN
       PRINT "ERRORE 4 in Risolve.K1 - dimensioni di KTA non corrette"
       END
    END IF
    IF LRB > 1 THEN
       PRINT "ERRORE 5 in Risolve.K1 - limiti inferiori di B maggiori di 1"
       END
    END IF
    IF NRB < N THEN
       PRINT "ERRORE 6 in Risolve.K1 - array B insufficiente"
       END
    END IF

    SELECT CASE RID
       CASE 0
          GOSUB TriangBa.K1
          GOSUB TriangBt.K1
          GOSUB DeterminaBt.K1
          GOSUB DeterminaBa.K1
       CASE 1
          GOSUB TriangBa.K1
       CASE 2
          GOSUB DeterminaBa.K1
    END SELECT
EXIT SUB
```

```
TriangBa.K1:
    ' ripete sulla seconda parte di B le operazioni fatte per triang. KAA e KTT
    FOR K = NA TO 1 STEP -1
       NK = NT + K
       C = B(NK)
       B(NK) = B(NK) / KAA(K)
       ' per l'azzeramento dei termini in KAA
       IF K < SB THEN LB = K ELSE LB = SB
       II = K - LB + 1
       DP = NA
       P = K
       FOR I = K - 1 TO II STEP -1
          NI = NT + I
          DP = DP - 1
          P = P + DP
          B(NI) = B(NI) - KAA(P) * C
       NEXT I
       ' per l'azzeramento dei termini in KTA
       FOR I = 1 TO NT
          B(I) = B(I) - KTA(I, K) * C
       NEXT I
    NEXT K
RETURN

TriangBt.K1:
    ' ripete sulla prima parte di B le operazioni fatte per triang. KAA e KTT
    FOR K = NT TO 1 STEP -1
       C = B(K)
       B(K) = B(K) / KTT(K, K)
       ' per l'azzeramento dei termini in KTT
       FOR I = 1 TO K - 1
          B(I) = B(I) - KTT(I, K) * C
       NEXT I
    NEXT K
RETURN

DeterminaBt.K1:
    ' effettua la sostituzione all'indietro per le prime NT incognite
    FOR K = 1 TO NT
       FOR I = K + 1 TO NT
          B(I) = B(I) - KTT(K, I) * B(K)
       NEXT I
    NEXT K
RETURN

DeterminaBa.K1:
    ' effettua la sostituzione all'indietro per le altre NA incognite
    FOR K = 1 TO NT
       FOR I = 1 TO NA
          NI = NT + I
          B(NI) = B(NI) - KTA(K, I) * B(K)
       NEXT I
    NEXT K
    FOR K = 1 TO NA
       NK = NT + K
       IF NA - K + 1 < SB THEN LB = NA - K + 1 ELSE LB = SB
       DP = NA
       P = K
```

```
      FOR I = K + 1 TO K + LB - 1
         NI = NT + I
         P = P + DP
         DP = DP - 1
         B(NI) = B(NI) - KAA(P) * B(NK)
      NEXT I
   NEXT K
RETURN

END SUB

' ============================== Risolve.K2 ==============================
'
'   Risoluzione di un sistema di equazioni col metodo di riduzione di Gauss
'       Seconda fase - risoluzione mediante sostituzione all'indietro
'
'       NOTE: la matrice dei coefficienti K e' divisa in 4: KTT, KTA, KAT, KAA
'             la matrice dei coefficienti e' simmetrica
'             la matrice dei coefficienti e' a banda (in array bidimensionale)
'             la matrice K e' stata resa triangolare inferiore, senza pivotaggio
'
'   Variabili di ingresso:
'      KTT(,)     matrice dei coefficienti, triangolarizzata superiormente
'      KTA(,)
'      KAA(,)        "     "       "      , a banda (dimensioni NA,SB)
'      SB         ampiezza della semibanda (compreso elemento diagonale)
'      RID        variabile di controllo del procedimento di risoluzione:
'                    RID = 0     triangolarizz. termini noti e sostituz. indietro
'                    RID = 1     triangolarizz. termini noti, esclusi i primi NT
'                    RID = 2     sostituzione indietro esclusi i primi NT termini
'      B()        vettore termini noti
'
'   Variabili di uscita:
'      B()        vettore soluzione
'
'   Variabili interne principali:
'      NT         ordine della matrice KTT
'      NA         ordine della matrice KAA
'      N          ordine della matrice dei coefficienti: N=NT+NA
'
' ------------------------------------------------------------------------------
'
SUB Risolve.K2 (KTT(), KTA(), KAA(), SB, RID, B())

    ' determina il numero di elementi degli array KTT, KTA, KAA, B
    LRTT = LBOUND(KTT, 1)
    LCTT = LBOUND(KTT, 2)
    NRTT = UBOUND(KTT, 1)
    NCTT = UBOUND(KTT, 2)
    LRTA = LBOUND(KTA, 1)
    LCTA = LBOUND(KTA, 2)
    NRTA = UBOUND(KTA, 1)
    NCTA = UBOUND(KTA, 2)
    LRAA = LBOUND(KAA, 1)
    LCAA = LBOUND(KAA, 2)
    NRAA = UBOUND(KAA, 1)
    NCAA = UBOUND(KAA, 2)
    LRB = LBOUND(B, 1)
    NRB = UBOUND(B, 1)
```

```
    ' controlla che le dimensioni degli array KTT, KTA, KAA siano corrette
    ' definisce NT, NA, N, ordine delle matrici dei coefficienti
    IF LRTT > 1 OR LCTT > 1 OR LRTA > 1 OR LCTA > 1 OR LRAA > 1 OR LCAA > 1 THEN
       PRINT "ERRORE 1 in Risolve.K2 - limiti inferiori di KTT, KTA o KAA maggiori
             di 1"
       END
    END IF
    IF NRTT <> NCTT THEN
       PRINT "ERRORE 2 in Risolve.K2 - matrice KTT non quadrata"
       END
    END IF
    NT = NRTT
    NA = NCTA
    N = NT + NA
    IF NRAA <> NA OR NCAA <> SB THEN
       PRINT "ERRORE 3 in Risolve.K2 - dimensioni di KAA non corrette"
       END
    END IF
    IF NRTA <> NT THEN
       PRINT "ERRORE 4 in Risolve.K2 - dimensioni di KTA non corrette"
       END
    END IF
    IF LRB > 1 THEN
       PRINT "ERRORE 5 in Risolve.K2 - limiti inferiori di B maggiori di 1"
       END
    END IF
    IF NRB < N THEN
       PRINT "ERRORE 6 in Risolve.K2 - array B insufficiente"
       END
    END IF

    SELECT CASE RID
       CASE 0
          GOSUB TriangBa.K2
          GOSUB TriangBt.K2
          GOSUB DeterminaBt.K2
          GOSUB DeterminaBa.K2
       CASE 1
          GOSUB TriangBa.K2
       CASE 2
          GOSUB DeterminaBa.K2
    END SELECT
EXIT SUB

TriangBa.K2:
    ' ripete sulla seconda parte di B le operazioni fatte per triang. KAA e KTT
    FOR K = NA TO 1 STEP -1
       NK = NT + K
       C = B(NK)
       B(NK) = B(NK) / KAA(K, 1)
       ' per l'azzeramento dei termini in KAA
       IF K < SB THEN LB = K ELSE LB = SB
       II = K - LB + 1
       FOR I = K - 1 TO II STEP -1
          D = K - I
          NI = NT + I
          B(NI) = B(NI) - KAA(I, D + 1) * C
       NEXT I
```

```
      ' per l'azzeramento dei termini in KTA
      FOR I = 1 TO NT
         B(I) = B(I) - KTA(I, K) * C
      NEXT I
   NEXT K
RETURN

TriangBt.K2:
   ' ripete sulla prima parte di B le operazioni fatte per triang. KAA e KTT
   FOR K = NT TO 1 STEP -1
      C = B(K)
      B(K) = B(K) / KTT(K, K)
      ' per l'azzeramento dei termini in KTT
      FOR I = 1 TO K - 1
         B(I) = B(I) - KTT(I, K) * C
      NEXT I
   NEXT K
RETURN

DeterminaBt.K2:
   ' effettua la sostituzione all'indietro per le prime NT incognite
   FOR K = 1 TO NT
      FOR I = K + 1 TO NT
         B(I) = B(I) - KTT(K, I) * B(K)
      NEXT I
   NEXT K
RETURN

DeterminaBa.K2:
   ' effettua la sostituzione all'indietro per le altre NA incognite
   FOR K = 1 TO NT
      FOR I = 1 TO NA
         NI = NT + I
         B(NI) = B(NI) - KTA(K, I) * B(K)
      NEXT I
   NEXT K
   FOR K = 1 TO NA
      NK = NT + K
      IF NA - K + 1 < SB THEN LB = NA - K + 1 ELSE LB = SB
      FOR I = K + 1 TO K + LB - 1
         D = K - I
         NI = NT + I
         B(NI) = B(NI) - KAA(K, 1 - D) * B(NK)
      NEXT I
   NEXT K
RETURN

END SUB

' ============================== Triang ==============================
'
'   Risoluzione di un sistema di equazioni col metodo di riduzione di Gauss
'       Prima fase - triangolarizzazione della matrice dei coefficienti
'
'       NOTA: la matrice dei coefficienti  viene resa  triangolare superiore
'             effettuando un pivotaggio parziale, cioe' uno scambio di righe
'             per rendere massimo il termine sulla diagonale principale
'
'   Variabili di ingresso:
```

```
'     A(,)      matrice dei coefficienti delle incognite, da triangolarizzare
'
'   Variabili di uscita:
'     A(,)      matrice dei coefficienti, triangolarizzata superiormente
'               NOTA: nel triangolo inferiore sono conservati i fattori
'                     usati per l'azzeramento
'     IPVT()    indice della riga del pivot K-esimo
'
'   Variabili interne principali:
'     N         ordine della matrice dei coefficienti
'
' ------------------------------------------------------------------------------
'
SUB Triang (A(), IPVT())

    ' determina il numero di elementi degli array A ed IPVT
    LRA = LBOUND(A, 1)
    LCA = LBOUND(A, 2)
    NRA = UBOUND(A, 1)
    NCA = UBOUND(A, 2)
    LRI = LBOUND(IPVT, 1)
    NRI = UBOUND(IPVT, 1)

    ' controlla che le dimensioni degli array A ed IPVT siano corrette
    IF LRA > 1 OR LCA > 1 OR LRI > 1 THEN
       PRINT "ERRORE 1 in Triang - limiti inferiori di A o IPVT maggiori di 1"
       END
    END IF
    IF NRA <> NCA THEN
       PRINT "ERRORE 2 in Triang - matrice dei coefficienti A non quadrata"
       END
    END IF
    IF NRI < NRA - 1 THEN
       PRINT "ERRORE 3 in Triang - array IPVT insufficiente"
       END
    END IF

    ' definisce N, ordine della matrice dei coefficienti
    N = NRA

    ' triangolarizzazione della matrice A
    ' NOTA: il ciclo non viene effettuato nel caso che N = 1
    FOR K = 1 TO N - 1
       ' cerca il pivot per la riga K
       M = K
       FOR I = K + 1 TO N
          IF ABS(A(I, K)) > ABS(A(M, K)) THEN M = I
       NEXT I
       IPVT(K) = M
       SWAP A(M, K), A(K, K)
       IF A(K, K) = 0 THEN
          PRINT "ERRORE 4 in Triang - matrice dei coefficienti A singolare"
          END
       END IF
       ' determina i coefficienti moltiplicativi delle righe
       FOR I = K + 1 TO N
          A(I, K) = A(I, K) / A(K, K)
       NEXT I
       ' scambia le righe ed azzera tutti gli elementi della colonna
```

```
      FOR J = K + 1 TO N
         SWAP A(M, J), A(K, J)
         IF A(K, J) <> 0 THEN
            FOR I = K + 1 TO N
               A(I, J) = A(I, J) - A(I, K) * A(K, J)
            NEXT I
         END IF
      NEXT J
   NEXT K

END SUB

' ============================ Triang.K1 ============================
'
'   Risoluzione di un sistema di equazioni col metodo di riduzione di Gauss
'       Prima fase - triangolarizzazione della matrice dei coefficienti
'
'       NOTE: la matrice dei coefficienti K e' divisa in 4: KTT, KTA, KAT, KAA
'             la matrice dei coefficienti e' simmetrica
'             la matrice dei coefficienti e' a banda (in array monodimensionale)
'             la matrice K viene resa triangolare inferiore, senza pivotaggio
'
'             Il procedimento di riduzione puo' essere limitato alla KAA e KTA,
'             per ottenere in KTT la matrice di rigidezza laterale
'
'   Variabili di ingresso:
'      KTT(,)    matrice dei coefficienti delle incognite, da triangolarizzare
'      KTA(,)       "     "       "
'      KAA()        "     "       "      , a banda (dimensione NAxSB-SBx(SB-1)/2)
'      SB        ampiezza della semibanda (compreso elemento diagonale)
'      RID       variabile di controllo del procedimento di riduzione:
'                   RID = 0     vengono triangolarizzate KTT e KAA
'                   RID = 1     viene triangolarizzata solo KAA
'
'   Variabili di uscita:
'      KTT(,)    matrice dei coefficienti, triangolarizzata inferiormente
'      KTA(,)       "     "       "
'      KAA()        "     "       "
'
'   Variabili interne principali:
'      NT        ordine della matrice KTT
'      NA        ordine della matrice KAA
'
' ------------------------------------------------------------------------------
'
SUB Triang.K1 (KTT(), KTA(), KAA(), SB, RID)

    ' determina il numero di elementi degli array KTT, KTA, KAA
   LRTT = LBOUND(KTT, 1)
   LCTT = LBOUND(KTT, 2)
   NRTT = UBOUND(KTT, 1)
   NCTT = UBOUND(KTT, 2)
   LRTA = LBOUND(KTA, 1)
   LCTA = LBOUND(KTA, 2)
   NRTA = UBOUND(KTA, 1)
   NCTA = UBOUND(KTA, 2)
   LRAA = LBOUND(KAA, 1)
   NRAA = UBOUND(KAA, 1)
```

```
' controlla che le dimensioni degli array KTT, KTA, KAA siano corrette
' definisce NT, NA, ordine delle matrici dei coefficienti
IF LRTT > 1 OR LCTT > 1 OR LRTA > 1 OR LCTA > 1 OR LRAA > 1 THEN
   PRINT "ERRORE 1 in Triang.K1 - limiti inferiori di KTT, KTA o KAA maggiori
          di 1"
   END
END IF
IF NRTT <> NCTT THEN
   PRINT "ERRORE 2 in Triang.K1 - matrice KTT non quadrata"
   END
END IF
NT = NRTT
NA = NCTA
IF NRAA <> NA * SB - SB * (SB - 1) / 2 THEN
   PRINT "ERRORE 3 in Triang.K1 - dimensioni di KAA non corrette"
   END
END IF
IF NRTA <> NT THEN
   PRINT "ERRORE 4 in Triang.K1 - dimensioni di KTA non corrette"
   END
END IF

' triangolarizzazione della matrice dei coefficienti
' prima fase: triangolarizzazione di KAA
FOR K = NA TO 1 STEP -1
   IF KAA(K) = 0 THEN
      PRINT "ERRORE 5 in Triang.K1 - valore 0 nella diagonale principale di KAA"
      END
   END IF
   ' ciclo di azzeramento per le righe di KAT e KAA
   IF K < SB THEN LB = K ELSE LB = SB
   II = K - LB + 1
   DP = NA
   P = K
   FOR I = K - 1 TO II STEP -1
      DP = DP - 1
      P = P + DP
      IF KAA(P) <> 0 THEN
         ' determina il coefficiente moltiplicativo relativo alla riga I
         C = KAA(P) / KAA(K)
         ' modifica tutti gli elementi della riga I
         FOR J = 1 TO NT
            KTA(J, I) = KTA(J, I) - KTA(J, K) * C
         NEXT J
         DM = NA
         M = I
         DQ = DP
         Q = P
         FOR J = I TO II STEP -1
            KAA(M) = KAA(M) - KAA(Q) * C
            DM = DM - 1
            M = M + DM
            DQ = DQ - 1
            Q = Q + DQ
         NEXT J
         KAA(P) = C
      END IF
   NEXT I
   ' ciclo di azzeramento per le righe di KTT e KTA
```

```
      FOR I = NT TO 1 STEP -1
         IF KTA(I, K) <> 0 THEN
            ' determina il coefficiente moltiplicativo relativo alla riga I
            C = KTA(I, K) / KAA(K)
            ' modifica tutti gli elementi della riga I
            FOR J = 1 TO I
               KTT(J, I) = KTT(J, I) - KTA(J, K) * C
            NEXT J
            KTA(I, K) = C
         END IF
      NEXT I
   NEXT K

   ' triangolarizzazione della matrice dei coefficienti
   ' seconda fase: triangolarizzazione di KTT
   IF RID <> 1 THEN
      FOR K = NT TO 1 STEP -1
         IF KTT(K, K) = 0 THEN
            PRINT "ERRORE 6 in Triang.K1 - valore 0 nella diagonale principale
                  di KTT"
            END
         END IF
         ' ciclo di azzeramento per le righe di KTT
         FOR I = K - 1 TO 1 STEP -1
            IF KTT(I, K) <> 0 THEN
               ' determina il coefficiente moltiplicativo relativo alla riga I
               C = KTT(I, K) / KTT(K, K)
               ' modifica tutti gli elementi della riga I
               FOR J = 1 TO I
                  KTT(J, I) = KTT(J, I) - KTT(J, K) * C
               NEXT J
               KTT(I, K) = C
            END IF
         NEXT I
      NEXT K
   END IF

END SUB

' ============================== Triang.K2 ==============================
'
'   Risoluzione di un sistema di equazioni col metodo di riduzione di Gauss
'       Prima fase - triangolarizzazione della matrice dei coefficienti
'
'       NOTE: la matrice dei coefficienti K e' divisa in 4: KTT, KTA, KAT, KAA
'             la matrice dei coefficienti e' simmetrica
'             la matrice dei coefficienti e' a banda (in array bidimensionale)
'             la matrice K viene resa triangolare inferiore, senza pivotaggio
'
'             Il procedimento di riduzione puo' essere limitato alla KAA e KTA,
'             per ottenere in KTT la matrice di rigidezza laterale
'
'   Variabili di ingresso:
'      KTT(,)    matrice dei coefficienti delle incognite, da triangolarizzare
'      KTA(,)       "     "       "
'      KAA(,)       "     "       "       , a banda (dimensioni NA,SB)
'      SB        ampiezza della semibanda (compreso elemento diagonale)
'      RID       variabile di controllo del procedimento di riduzione:
'                   RID = 0     vengono triangolarizzate KTT e KAA
```

```
'                  RID = 1      viene triangolarizzata solo KAA
'
'   Variabili di uscita:
'      KTT(,)    matrice dei coefficienti, triangolarizzata inferiormente
'      KTA(,)       "     "       "
'      KAA(,)       "     "       "
'
'   Variabili interne principali:
'      NT        ordine della matrice KTT
'      NA        ordine della matrice KAA
'
' ------------------------------------------------------------------------------
'
SUB Triang.K2 (KTT(), KTA(), KAA(), SB, RID)

    ' determina il numero di elementi degli array KTT, KTA, KAA
    LRTT = LBOUND(KTT, 1)
    LCTT = LBOUND(KTT, 2)
    NRTT = UBOUND(KTT, 1)
    NCTT = UBOUND(KTT, 2)
    LRTA = LBOUND(KTA, 1)
    LCTA = LBOUND(KTA, 2)
    NRTA = UBOUND(KTA, 1)
    NCTA = UBOUND(KTA, 2)
    LRAA = LBOUND(KAA, 1)
    LCAA = LBOUND(KAA, 2)
    NRAA = UBOUND(KAA, 1)
    NCAA = UBOUND(KAA, 2)

    ' controlla che le dimensioni degli array KTT, KTA, KAA siano corrette
    ' definisce NT, NA, ordine delle matrici dei coefficienti
    IF LRTT > 1 OR LCTT > 1 OR LRTA > 1 OR LCTA > 1 OR LRAA > 1 OR LCAA > 1 THEN
       PRINT "ERRORE 1 in Triang.K2 - limiti inferiori di KTT, KTA o KAA maggiori
             di 1"
       END
    END IF
    IF NRTT <> NCTT THEN
       PRINT "ERRORE 2 in Triang.K2 - matrice KTT non quadrata"
       END
    END IF
    NT = NRTT
    NA = NCTA
    IF NRAA <> NA OR NCAA <> SB THEN
       PRINT "ERRORE 3 in Triang.K2 - dimensioni di KAA non corrette"
       END
    END IF
    IF NRTA <> NT THEN
       PRINT "ERRORE 4 in Triang.K2 - dimensioni di KTA non corrette"
       END
    END IF

    ' triangolarizzazione della matrice dei coefficienti
    ' prima fase: triangolarizzazione di KAA
    FOR K = NA TO 1 STEP -1
       IF KAA(K, 1) = 0 THEN
          PRINT "ERRORE 5 in Triang.K2 - valore 0 nella diagonale principale di KAA"
          END
       END IF
       ' ciclo di azzeramento per le righe di KAT e KAA
```

```
      IF K < SB THEN LB = K ELSE LB = SB
      II = K - LB + 1
      FOR I = K - 1 TO II STEP -1
         D = K - I
         IF KAA(I, D + 1) <> 0 THEN
            ' determina il coefficiente moltiplicativo relativo alla riga I
            C = KAA(I, D + 1) / KAA(K, 1)
            ' modifica tutti gli elementi della riga I
            FOR J = 1 TO NT
               KTA(J, I) = KTA(J, I) - KTA(J, K) * C
            NEXT J
            FOR J = II TO I
               M = I - J + 1
               KAA(J, M) = KAA(J, M) - KAA(J, M + D) * C
            NEXT J
            KAA(I, D + 1) = C
         END IF
      NEXT I
      ' ciclo di azzeramento per le righe di KTT e KTA
      FOR I = NT TO 1 STEP -1
         IF KTA(I, K) <> 0 THEN
            ' determina il coefficiente moltiplicativo relativo alla riga I
            C = KTA(I, K) / KAA(K, 1)
            ' modifica tutti gli elementi della riga I
            FOR J = 1 TO I
               KTT(J, I) = KTT(J, I) - KTA(J, K) * C
            NEXT J
            KTA(I, K) = C
         END IF
      NEXT I
   NEXT K

   ' triangolarizzazione della matrice dei coefficienti
   ' seconda fase: triangolarizzazione di KTT
   IF RID <> 1 THEN
      FOR K = NT TO 1 STEP -1
         IF KTT(K, K) = 0 THEN
            PRINT "ERRORE 6 in Triang.K2 - valore 0 nella diagonale principale
                  di KTT"
            END
         END IF
         ' ciclo di azzeramento per le righe di KTT
         FOR I = K - 1 TO 1 STEP -1
            IF KTT(I, K) <> 0 THEN
               ' determina il coefficiente moltiplicativo relativo alla riga I
               C = KTT(I, K) / KTT(K, K)
               ' modifica tutti gli elementi della riga I
               FOR J = 1 TO I
                  KTT(J, I) = KTT(J, I) - KTT(J, K) * C
               NEXT J
               KTT(I, K) = C
            END IF
         NEXT I
      NEXT K
   END IF

END SUB
```

5. Esempio

Come esempio di programma principale che utilizza le procedure qui descritte è riportato un programma per la risoluzione di un sistema di equazioni generico mediante le procedure `Triang` e `Risolve`. Esso prevede la lettura dei dati (numero di equazioni `N`, matrice dei coefficienti `A`, termini noti `B`) conservati in apposite linee contraddistinte dall'istruzione `DATA`. Dopo aver stampato i dati, si effettuano le due fasi, di triangolarizzazione e di sostituzione all'indietro, ottenendo le incognite direttamente nel vettore `B`. Viene quindi stampata la matrice `A` modificata, contenente ora nel triangolo inferiore i fattori di riduzione, ed il vettore `IPVT` che indica l'eventuale scambio di righe nel pivotaggio parziale. Viene infine stampato il vettore `B` che racchiude i valori delle incognite.

```
                RISOLUZIONE DI UN SISTEMA DI EQUAZIONI
              mediante il metodo di riduzione di Gauss

DECLARE SUB LprintMat (A!(), F$)
DECLARE SUB Risolve (A!(), B!(), IPVT!())
DECLARE SUB Triang (A!(), IPVT!())

OPTION BASE 1

READ N
DIM A(N, N), B(N), IPVT(N - 1)

' lettura della matrice dei coefficienti A e del vettore termine noto B
FOR I = 1 TO N
   FOR J = 1 TO N
      READ A(I, J)
   NEXT J
NEXT I
FOR I = 1 TO N
   READ B(I)
NEXT I

' stampa dei dati
LPRINT "Matrice dei coefficienti :"
CALL LprintMat(A(), "##.###     ")
LPRINT
LPRINT "Vettore termini noti :"
CALL LprintMat(B(), "##.###     ")
LPRINT

' risoluzione col metodo di riduzione di Gauss
CALL Triang(A(), IPVT())
CALL Risolve(A(), B(), IPVT())

LPRINT "Matrice dei coefficienti triangolarizzata e fattori di riduzione :"
CALL LprintMat(A(), "##.###     ")
LPRINT
LPRINT "Vettore scambi righe :"
```

```
CALL LprintMat(IPVT(), "##.###      ")
LPRINT

' stampa dei risultati
LPRINT "Vettore risultato :"
CALL LprintMat(B(), "##.###      ")

END


' dati per il programma
DATA  7
DATA  3, 2, 1, 0, 0, 0, 0
DATA  2,-2,-1,-2, 0, 0, 0
DATA  1,-1, 2,-1, 1, 0, 0
DATA  0,-2,-1, 4, 2,-3, 0
DATA  0, 0, 1, 2,-1, 1,-1
DATA  0, 0, 0,-3, 1, 2,-1
DATA  0, 0, 0, 0,-1,-1, 5
DATA -1,-7, 5,-7,-2, 2, 9
```

Il programma è stato mandato in esecuzione con i dati già usati per l'esempio del capitolo precedente. L'output ottenuto è riportato qui di seguito. Pur avendo utilizzato l'impostazione generale, la matrice dei coefficienti è in realtà simmetrica e dotata di banda. Si può notare di conseguenza che la stessa banda è presente nella matrice modificata **A**. Inoltre, dividendo i termini del triangolo superiore per i valori posti nella diagonale principale si ottengono i fattori di riduzione memorizzati nel triangolo inferiore, a conferma delle già citate proprietà delle matrici simmetriche.

```
Matrice dei coefficienti :

 3.000   2.000   1.000   0.000   0.000   0.000   0.000
 2.000  -2.000  -1.000  -2.000   0.000   0.000   0.000
 1.000  -1.000   2.000  -1.000   1.000   0.000   0.000
 0.000  -2.000  -1.000   4.000   2.000  -3.000   0.000
 0.000   0.000   1.000   2.000  -1.000   1.000  -1.000
 0.000   0.000   0.000  -3.000   1.000   2.000  -1.000
 0.000   0.000   0.000   0.000  -1.000  -1.000   5.000

Vettore termini noti :

-1.000
-7.000
 5.000
-7.000
-2.000
 2.000
 9.000
```

```
Matrice dei coefficienti triangolarizzata e fattori di riduzione :

 3.000   2.000   1.000   0.000   0.000   0.000   0.000
 0.667  -3.333  -1.667  -2.000   0.000   0.000   0.000
 0.333   0.500   2.500   0.000   1.000   0.000   0.000
 0.000   0.600   0.000   5.200   2.000  -3.000   0.000
 0.000   0.000   0.400   0.385  -2.169   2.154  -1.000
 0.000   0.000   0.000  -0.577  -0.993   2.408  -1.993
 0.000   0.000   0.000   0.000   0.461  -0.828   3.811

Vettore scambi righe :

 1.000
 2.000
 3.000
 4.000
 5.000
 6.000

Vettore risultato :

-2.000
 1.000
 3.000
-1.000
 1.000
-0.000
 2.000
```

CAPITOLO QUARTO

LA SINGOLA ASTA

1. Generalità

Nel secondo capitolo della prima parte del testo è stato analizzato il comportamento di una generica asta in relazione alle sue caratteristiche geometriche, alle diverse condizioni di vincolo degli estremi ed alla presenza di una vasta gamma di tipologie di carico.

In particolare, sono state determinate: le relazioni che intercorrono, in assenza di carichi, tra azioni e componenti di movimento di estremità, sintetizzate dalla matrice di rigidezza $\mathbf{k}$ dell'asta; le azioni di incastro perfetto $\overline{\mathbf{S}}$, che insorgono per effetto dei carichi agenti sull'asta quando il movimento dei suoi estremi è impedito; le formulazioni che consentono di determinare le caratteristiche di sollecitazione in un qualsiasi punto dell'asta, in funzione dei carichi e delle azioni di estremità.

In analogia a quanto fatto per le operazioni matriciali e per la soluzione di sistemi di equazioni, anche in questo caso è stato realizzato un insieme di procedure che rendono immediatamente operativo quanto esposto nella parte teorica, in modo da facilitare il compito di chi desidera realizzare programmi di calcolo di validità generale.

Alle procedure, contenute nel file `ASTA.BAS`, sono stati assegnati nomi che richiamano alla mente quale prodotto forniscono ed il tipo di asta e di schema di carico cui si riferiscono.

Per individuare il tipo di asta si sono utilizzate le seguenti sigle:

`Cer`	=	asta con cerniera ad uno o ad entrambi gli estremi;
`Inc`	=	asta perfettamente incastrata ai nodi di estremità;
`NodRig`	=	asta collegata a nodi rigidi non puntiformi;
`Rig`	=	asta con tratti rigidi coassiali agli estremi;
`Wnk`	=	asta su suolo elastico alla Winkler.

Per i primi due tipi si è usata la sigla aggiuntiva `Tag` quando si è voluto indicare che per tale asta si tiene conto anche della deformazione conseguente al taglio (ottenendo quindi le sigle `CerTag` e `IncTag`).

Per individuare il tipo di carico si sono utilizzate le seguenti sigle:

`AzEstr`	=	azioni di estremità;
`DT`	=	distorsioni termiche;
`FM`	=	forze e momenti applicati in un punto interno all'asta;
`Qunif`	=	carichi uniformemente distribuiti su tutta l'asta;
`Qvar`	=	carichi distribuiti con legge lineare su parte dell'asta.

Le procedure possono essere divise in più gruppi:

- procedure che operano sui dati geometrici dell'asta, per fornire la matrice di rotazione `Ld` e quella di trasformazione `T`:
  `Geometria, T.Cer, T.Inc, T.Wnk;`
- procedure che forniscono la matrice di rigidezza fondamentale dell'asta:
  `Kf.Cer, Kf.CerTag, Kf.Inc, Kf.IncTag, Kf.NodRig, Kf.Rig, Kf.Wnk;`
- procedure che effettuano la trasformazione della matrice di rigidezza da un sistema di riferimento a un'altro:
  `Kl.` trasformazione da fondamentale a locale;
  `Kg.f` trasformazione da fondamentale a globale;
  `Kg.l` trasformazione da locale a globale;
- procedure che forniscono direttamente la matrice di rigidezza dell'asta nel sistema di riferimento globale:
  `Kg.Cer, Kg.CerTag, Kg.Inc, Kg.IncTag, Kg.NodRig, Kg.Rig, Kg.Wnk;`
- procedure riepilogative che conglobano gli effetti di `Geometria` e `Kg.f`, realizzate per minimizzare i tempi di esecuzione per specifici tipi di asta:
  `Asta.Inc;`
- procedure per la determinazione delle azioni di incastro perfetto:
  `Sl.Cer.Qunif, Sl.Inc.DT, Sl.Inc.FM, Sl.Inc.Qunif, Sl.Inc.Qvar, Sl.Rig.Qunif, Sl.Wnk.Qunif;`

– procedure per la valutazione delle caratteristiche della sollecitazione in un punto interno dell'asta:
`CarSol.AzEstr, CarSol.FM, Carsol.Qunif, CarSol.Qvar.`

Nel listato mostrato nel paragrafo successivo, che contiene tutte le procedure in ordine alfabetico, è riportata più specificamente per ciascuna di esse la funzione svolta, nonché le variabili di scambio in ingresso ed uscita e le procedure eventualmente utilizzate.

2. Codifica

```
'                                ASTA.BAS
'
'         procedure per la valutazione delle matrici di rigidezza,
'         dei vettori di incastro perfetto e delle caratteristiche
'         di sollecitazione nei punti interni delle aste
'
'                                                          rev. 09.89

' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file MATRIX.BAS
DECLARE SUB AzzeraMat (A!())
DECLARE SUB InserMat (A!(), B!(), RI!, CI!)
DECLARE SUB InversaMat (A!())
DECLARE SUB LimitiMat (A!(), DA!, LR!, LC!, NR!, NC!)
DECLARE SUB ProdottoCostMat (K!, A!(), B!())
DECLARE SUB ProdottoMat (A!(), B!(), C!())
DECLARE SUB TrasponeMat (A!(), B!())

' procedure di questo stesso file
DECLARE SUB Geometria (X1!, X2!, Y1!, Y2!, L!, SnAl!, CsAl!, Ld!())
DECLARE SUB Kf.Cer (ES!, E!, A!, I!, L!, Kf!())
DECLARE SUB Kf.CerTag (ES!, E!, G!, CHI!, A!, I!, L!, Kf!())
DECLARE SUB Kf.Inc (E!, A!, I!, L!, Kf!())
DECLARE SUB Kf.IncTag (E!, G!, CHI!, A!, I!, L!, Kf!())
DECLARE SUB Kf.NodRig (E!, A!, I!, L!, Sx!, Sy!, Tx!, Ty!, Kf!())
DECLARE SUB Kf.Rig (E!, A!, I!, L!, S!, T!, Kf!())
DECLARE SUB Kf.Wnk (K!, B!, E!, A!, I!, L!, Kf!())
DECLARE SUB Kg.f (Ld!(), T!(), Kf!(), Kg!())
DECLARE SUB T.Cer (ES!, L!, T!())
DECLARE SUB T.Inc (L!, T!())
DECLARE SUB T.Wnk (T!())

' ============================== Asta.Inc ==============================
'
'   Procedura per la valutazione diretta della matrice di rotazione,
'   dei parametri che dipendono solo dalla geometria dell'asta
'   e della matrice di rigidezza nel riferimento globale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'
'   NOTA: per ridurre il tempo di calcolo,  la matrice di rigidezza
'         viene valutata con le formulazioni dirette, senza passare
```

```
'        per le matrici fondamentale e locale
'
'   Variabili di ingresso:
'      X1         ascissa primo estremo
'      X2         ascissa secondo estremo
'      Y1         ordinata primo estremo
'      Y2         ordinata secondo estremo
'      E          modulo di elasticita' normale
'      A          area della sezione trasversale dell'asta
'      I          momento d'inerzia della sezione trasversale dell'asta
'
'   Variabili di uscita:
'      L          lunghezza dell'asta
'      SnAl       seno dell'angolo di inclinazione dell'asta
'      CsAl       coseno dell'angolo di inclinazione dell'asta
'      Ld()       matrice di rotazione dell'asta
'      Kg()       matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Geometria
'
' ------------------------------------------------------------------------------
'
SUB Asta.Inc (X1, X2, Y1, Y2, E, A, I, L, SnAl, CsAl, Ld(), Kg())

    CALL Geometria(X1, X2, Y1, Y2, L, SnAl, CsAl, Ld())

    EIL = E * I / L
    EIL2 = EIL / L
    EIL3 = EIL2 / L
    EAL = E * A / L
    SN2 = SnAl ^ 2
    CS2 = CsAl ^ 2

    K1 = EAL * CS2 + 12 * EIL3 * SN2
    K2 = EAL * SN2 + 12 * EIL3 * CS2
    K3 = 6 * EIL2 * SnAl
    K4 = 6 * EIL2 * CsAl
    K5 = 4 * EIL
    K6 = 2 * EIL
    K7 = (EAL - 12 * EIL3) * SnAl * CsAl

    Kg(1, 1) = K1
    Kg(1, 2) = K7
    Kg(1, 3) = -K3
    Kg(1, 4) = -K1
    Kg(1, 5) = -K7
    Kg(1, 6) = -K3

    Kg(2, 1) = K7
    Kg(2, 2) = K2
    Kg(2, 3) = K4
    Kg(2, 4) = -K7
    Kg(2, 5) = -K2
    Kg(2, 6) = K4

    Kg(3, 1) = -K3
    Kg(3, 2) = K4
    Kg(3, 3) = K5
```

```
    Kg(3, 4) = K3
    Kg(3, 5) = -K4
    Kg(3, 6) = K6

    Kg(4, 1) = -K1
    Kg(4, 2) = -K7
    Kg(4, 3) = K3
    Kg(4, 4) = K1
    Kg(4, 5) = K7
    Kg(4, 6) = K3

    Kg(5, 1) = -K7
    Kg(5, 2) = -K2
    Kg(5, 3) = -K4
    Kg(5, 4) = K7
    Kg(5, 5) = K2
    Kg(5, 6) = -K4

    Kg(6, 1) = -K3
    Kg(6, 2) = K4
    Kg(6, 3) = K6
    Kg(6, 4) = K3
    Kg(6, 5) = -K4
    Kg(6, 6) = K5

END SUB

' ========================== CarSol.AzEstr ==========================
'
'   Procedura per il calcolo di momento, taglio e sforzo normale:
'   effetto delle azioni di estremita'
'
'   Variabili di ingresso:
'      Fx        componente parallela all'asta dell'azione al primo estremo
'      Fy        componente ortogonale all'asta dell'azione al primo estremo
'      Mz        coppia concentrata Mz al primo estremo
'      X         ascissa punto calcolo delle sollecitazioni
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Sn        sforzo normale all'ascissa X
'      Mf        momento flettente all'ascissa X
'      Ta        taglio all'ascissa X
'
'   Procedure utilizzate:
'      ===
'
' ------------------------------------------------------------------------------
'
SUB Carsol.AzEstr (Fx, Fy, Mz, X, L, Sn, Mf, Ta)

    IF X < 0 OR X > L THEN
       PRINT "ERRORE 1 in CarSol.AzEstr - valori dell'ascissa non accettabili"
       END
    END IF

    Sn = -Fx
    Mf = Fy * X - Mz
    Ta = Fy
```

```
END SUB

' ================================ CarSol.FM ================================
'
'   Procedura per il calcolo di momento, taglio e sforzo normale:
'   forza F (di componenti Fx e Fy) e coppia Mz concentrate
'
'   Variabili di ingresso:
'      Fx        componente parallela all'asta della forza concentrata F
'      Fy        componente ortogonale all'asta della forza concentrata F
'      Mz        coppia concentrata M
'      XO        ascissa punto applicazione azioni concentrate
'      X         ascissa punto calcolo delle sollecitazioni
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Sn        sforzo normale all'ascissa X
'      Mf        momento flettente all'ascissa X
'      Ta        taglio all'ascissa X
'
'   Procedure utilizzate:
'      ===
'
' ------------------------------------------------------------------------------
'
SUB Carsol.FM (Fx, Fy, Mz, XO, X, L, Sn, Mf, Ta)

    IF XO < 0 OR XO > L OR X < 0 OR X > L THEN
       PRINT "ERRORE 1 in CarSol.FM - valori delle ascisse non accettabili"
       END
    END IF

    IF X <= XO THEN
       Sn = 0
       Mf = 0
       Ta = 0
    ELSE
       Sn = -Fx
       Mf = Fy * (X - XO) - Mz
       Ta = Fy
    END IF

END SUB

' ================================ CarSol.Qunif ================================
'
'   Procedura per il calcolo di momento, taglio e sforzo normale:
'   carico uniformemente distribuito
'
'   Variabili di ingresso:
'      Q         carico uniformemente distribuito ortogonale all'asta
'      N         carico uniformemente distribuito parallelo  all'asta
'      X         ascissa per il calcolo delle sollecitazioni
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Sn        sforzo normale all'ascissa X
'      Mf        momento flettente all'ascissa X
```

```
'      Ta        taglio all'ascissa X
'
'   Procedure utilizzate:
'      ===
'
' ------------------------------------------------------------------------------
'
SUB CarSol.Qunif (Q, N, X, L, Sn, Mf, Ta)

    IF X < 0 OR X > L THEN
       PRINT "ERRORE 1 in CarSol.Qunif - valori dell'ascissa non accettabili"
       END
    END IF

    Sn = -N * X
    Mf = Q * X ^ 2 / 2
    Ta = Q * X

END SUB

' ============================== CarSol.Qvar ==============================
'
'   Procedura per il calcolo di momento e taglio:
'   carico distribuito con andamento lineare su parte dell'asta
'
'   NOTA: il carico e' ortogonale all'asse e non provoca sforzo normale
'
'   Variabili di ingresso:
'      Q1        valore del carico a sinistra
'      Q2        valore del carico a destra
'      X1        ascissa inizio carico
'      X2        ascissa fine carico
'      X         ascissa per calcolo sollecitazioni
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Mf        momento flettente all'ascissa X
'      Ta        taglio all'ascissa X
'
'   Procedure utilizzate:
'      ===
'
' ------------------------------------------------------------------------------
'
SUB Carsol.Qvar (Q1, Q2, X1, X2, X, L, Mf, Ta)

    IF X1 < 0 OR X1 > L OR X2 < 0 OR X2 > L OR X < 0 OR X > L THEN
       PRINT "ERRORE 1 in CarSol.Qvar - valori delle ascisse non accettabili"
       END
    END IF
    IF X1 > X2 THEN
       PRINT "ERRORE 2 in CarSol.Qvar - ordine delle ascisse non accettabile"
       END
    END IF

    IF X <= X1 THEN
       Mf = 0
       Ta = 0
    ELSEIF X < X2 THEN
```

```
      Dx1 = X - X1
      Dx2 = X - X2
      D = X2 - X1
      U = Dx1 / D
      Mf = ((1 - U / 3) * Q1 + U / 3 * Q2) * Dx1 ^ 2 / 2
      Ta = ((1 - U / 2) * Q1 + U / 2 * Q2) * Dx1
   ELSE
      Dx2 = X - X2
      D = X2 - X1
      TO = (Q1 + Q2) / 2 * D
      MO = (2 / 3 * Q1 + 1 / 3 * Q2) * D ^ 2 / 2
      Mf = TO * Dx2 + MO
      Ta = TO
   END IF

END SUB

' ================================ Geometria ================================
'
'   Procedura  per  la valutazione  della matrice  di rotazione
'   e di parametri che dipendono solo dalla geometria dell'asta
'
'   Variabili di ingresso:
'      X1         ascissa primo estremo
'      X2         ascissa secondo estremo
'      Y1         ordinata primo estremo
'      Y2         ordinata secondo estremo
'
'   Variabili di uscita:
'      L          lunghezza dell'asta
'      SnAl       seno dell'angolo di inclinazione dell'asta
'      CsAl       coseno dell'angolo di inclinazione dell'asta
'      Ld()       matrice di rotazione dell'asta
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ---------------------------------------------------------------------------
'
SUB Geometria (X1, X2, Y1, Y2, L, SnAl, CsAl, Ld())

   L = SQR((X2 - X1) ^ 2 + (Y2 - Y1) ^ 2)
   IF L = 0 THEN
      PRINT "ERRORE 1 in Geometria - asta di lunghezza nulla"
      END
   END IF
   SnAl = (Y2 - Y1) / L
   CsAl = (X2 - X1) / L

   CALL AzzeraMat(Ld())
   Ld(1, 1) = CsAl
   Ld(1, 2) = -SnAl
   Ld(2, 1) = SnAl
   Ld(2, 2) = CsAl
   Ld(3, 3) = 1
   Ld(4, 4) = CsAl
   Ld(4, 5) = -SnAl
   Ld(5, 4) = SnAl
   Ld(5, 5) = CsAl
```

```
    Ld(6, 6) = 1

END SUB

' ============================== Kf.Cer ==============================
'
'   Procedura per la definizione della matrice di rigidezza fondamentale
'   per un'asta incernierata agli estremi
'
'   Variabili di ingresso:
'      ES        estremo incernierato
'                   1 = primo estremo
'                   2 = secondo estremo
'                   3 = entrambi gli estremi
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Kf()      matrice di rigidezza fondamentale
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB Kf.Cer (ES, E, A, I, L, Kf())

    CALL AzzeraMat(Kf())
    SELECT CASE ES
       CASE 1, 2
          Kf(1, 1) = 3 * E * I / L
          Kf(2, 2) = E * A / L
       CASE 3
          Kf(1, 1) = E * A / L
    END SELECT

END SUB

' ============================== Kf.CerTag ==============================
'
'   Procedura per la definizione della matrice di rigidezza fondamentale
'   per asta incernierata agli estremi tenendo conto della deformazione a taglio
'
'   Variabili di ingresso:
'      ES        estremo incernierato
'                   1 = primo estremo
'                   2 = secondo estremo
'                   3 = entrambi gli estremi
'      E         modulo di elasticita' normale
'      G         modulo di elasticita' tangenziale
'      CHI       fattore di taglio
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Kf()      matrice fondamentale dell'asta
```

```
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB Kf.CerTag (ES, E, G, CHI, A, I, L, Kf())

    eps = 3 * E * I * CHI / (G * A * L ^ 2)

    CALL AzzeraMat(Kf())
    SELECT CASE ES
       CASE 1, 2
          Kf(1, 1) = 3 * E * I / L / (1 + eps)
          Kf(2, 2) = E * A / L
       CASE 3
          Kf(1, 1) = E * A / L
    END SELECT

END SUB

' ========================== Kf.Inc ============================
'
'   Procedura per la definizione della matrice di rigidezza fondamentale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'
'   Variabili di ingresso:
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Kf()      matrice di rigidezza fondamentale
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB Kf.Inc (E, A, I, L, Kf())

    CALL AzzeraMat(Kf())
    EIL = 2 * E * I / L
    Kf(1, 1) = 2 * EIL
    Kf(1, 2) = EIL
    Kf(2, 1) = EIL
    Kf(2, 2) = Kf(1, 1)
    Kf(3, 3) = E * A / L

END SUB

' ========================== Kf.IncTag ============================
'
'   Procedura per la definizione della matrice di rigidezza fondamentale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'   tenendo conto della deformazione a taglio
'
'   Variabili in ingresso:
```

```
'      E          modulo di elasticita' normale
'      G          modulo di elasticita' tangenziale
'      CHI        fattore di taglio
'      A          area della sezione trasversale dell'asta
'      I          momento d'inerzia della sezione trasversale dell'asta
'      L          lunghezza dell'asta
'
'   Variabili in uscita:
'      Kf()       matrice di rigidezza fondamentale
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB Kf.IncTag (E, G, CHI, A, I, L, Kf())

    eps = 3 * E * I * CHI / (G * A * L ^ 2)

    CALL AzzeraMat(Kf())
    EIL = 2 * E * I / L
    Kf(1, 1) = 2 * EIL * (1 + eps) / (1 + 4 * eps)
    Kf(1, 2) = EIL * (1 - 2 * eps) / (1 + 4 * eps)
    Kf(2, 1) = Kf(1, 2)
    Kf(2, 2) = Kf(1, 1)
    Kf(3, 3) = E * A / L

END SUB

' ========================== Kf.NodRig ==========================
'
'   Procedura per la determinazione della matrice di rigidezza fondamentale
'   nel caso di nodi rigidi non puntiformi
'   (asta con tratti rigidi non coassiali)
'
'   Variabili di ingresso:
'      E          modulo di elasticita' normale
'      A          area della sezione trasversale
'      I          momento d'inerzia della sezione trasversale
'      L          lunghezza dell'asta
'      Sx         coefficiente lunghezza x tratto sinistro
'      Sy         coefficiente lunghezza y tratto sinistro
'      Tx         coefficiente lunghezza x tratto destro
'      Ty         coefficiente lunghezza y tratto destro
'
'   Variabili di uscita:
'      Kf()       matrice di rigidezza fondamentale
'
'   Procedure utilizzate:
'      ProdottoMat
'      TrasponeMat
'      Kf.Inc
'
' ------------------------------------------------------------------------------
'
SUB Kf.NodRig (E, A, I, L, Sx, Sy, Tx, Ty, Kf())

    C = 1 / SQR((1 + Tx - Sx) ^ 2 + (Ty - Sy) ^ 2)
    Ls = L / C
```

```
    SnAl = (Ty - Sy) * C
    CsAl = (1 - Sx + Tx) * C

    DIM C(3, 3)
    C(1, 1) = 1 + C * (Sy * SnAl + Sx * CsAl)
    C(2, 1) = -C * (Ty * SnAl + Tx * CsAl)
    C(3, 1) = C * SnAl / L
    C(1, 2) = C * (Sy * SnAl + Sx * CsAl)
    C(2, 2) = 1 - C * (Ty * SnAl + Tx * CsAl)
    C(3, 2) = C * SnAl / L
    C(1, 3) = (Sy * CsAl - Sx * SnAl) * L
    C(2, 3) = -(Ty * CsAl - Tx * SnAl) * L
    C(3, 3) = CsAl

    CALL Kf.Inc(E, A, I, Ls, Kf())
    CALL ProdottoMat(C(), Kf(), Kf())
    CALL TrasponeMat(C(), C())
    CALL ProdottoMat(Kf(), C(), Kf())

END SUB

' ================================ Kf.Rig ================================
'
'   Procedura per la definizione della matrice di rigidezza fondamentale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'   avente tratti rigidi di estremita'
'
'   Variabili di ingresso:
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      S         coefficiente lunghezza tratto rigido sinistro
'      T         coefficiente lunghezza tratto rigido destro
'
'   Variabili di uscita:
'      Kf()      matrice di rigidezza fondamentale
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB Kf.Rig (E, A, I, L, S, T, Kf())

    C = 1 / (1 - S - T)
    A1 = C * ((1 - T) ^ 2 + S * (1 - T) + S ^ 2)
    A2 = C * ((1 - S) ^ 2 + T * (1 - S) + T ^ 2)
    B = C * (1 + S + T - 2 * (S ^ 2 - S * T + T ^ 2))

    CALL AzzeraMat(Kf())
    EIL = 2 * E * I / L
    Kf(1, 1) = 2 * EIL * A1
    Kf(1, 2) = EIL * B
    Kf(2, 1) = Kf(1, 2)
    Kf(2, 2) = 2 * EIL * A2
    Kf(3, 3) = E * A / L * C

END SUB
```

```
' ================================ Kf.Wnk ================================
'
'   Procedura per la definizione della matrice di rigidezza fondamentale
'   per un'asta su suolo elastico alla Winkler
'
'   Variabili di ingresso:
'      K         costante di sottofondo
'      B         larghezza trave
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Kf()      matrice di rigidezza fondamentale
'
'   Procedure utilizzate:
'      AzzeraMat
'      InserMat
'      InversaMat
'      ProdottoCostMat
'      ProdottoMat
'
' ------------------------------------------------------------------------------
'
SUB Kf.Wnk (K, B, E, A, I, L, Kf())

    CONST en = 2.718
    lambda = (K * B / (4 * E * I)) ^ .25
    lambdaL = lambda * L
    e1 = en ^ lambdaL
    e2 = en ^ (-lambdaL)
    SnL = SIN(lambdaL)
    CsL = COS(lambdaL)
    sp = e1 * SnL
    cp = e1 * CsL
    ss = e2 * SnL
    cs = e2 * CsL

    DIM K1(4, 4)
    K1(1, 1) = 0
    K1(1, 2) = 1
    K1(1, 3) = 0
    K1(1, 4) = 1
    K1(2, 1) = lambda
    K1(2, 2) = lambda
    K1(2, 3) = lambda
    K1(2, 4) = -lambda
    K1(3, 1) = sp
    K1(3, 2) = cp
    K1(3, 3) = ss
    K1(3, 4) = cs
    K1(4, 1) = lambda * (sp + cp)
    K1(4, 2) = lambda * (-sp + cp)
    K1(4, 3) = lambda * (-ss + cs)
    K1(4, 4) = lambda * (-ss - cs)

    DIM K2(4, 4)
```

```
    K2(1, 1) = lambda
    K2(1, 2) = -lambda
    K2(1, 3) = lambda
    K2(1, 4) = lambda
    K2(2, 1) = -1
    K2(2, 2) = 0
    K2(2, 3) = 1
    K2(2, 4) = 0
    K2(3, 1) = lambda * (sp - cp)
    K2(3, 2) = lambda * (sp + cp)
    K2(3, 3) = lambda * (-ss - cs)
    K2(3, 4) = lambda * (ss - cs)
    K2(4, 1) = cp
    K2(4, 2) = -sp
    K2(4, 3) = -cs
    K2(4, 4) = ss
    CALL ProdottoCostMat(2 * E * I * lambda ^ 2, K2(), K2())

    CALL InversaMat(K1())
    CALL ProdottoMat(K2(), K1(), K2())
    CALL AzzeraMat(Kf())
    CALL InserMat(K2(), Kf(), 1, 1)
    Kf(5, 5) = E * A / L

END SUB

' ============================== Kg.Cer ==============================
'
'   Procedura per la definizione della matrice di rigidezza globale
'   per un'asta incernierata agli estremi
'
'   Variabili di ingresso:
'      ES        estremo incernierato
'                   1 = primo estremo
'                   2 = secondo estremo
'                   3 = entrambi gli estremi
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      Ld()      matrice di rotazione
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Kf.Cer
'      Kg.f
'      T.Cer
'
' ------------------------------------------------------------------------------
'
SUB Kg.Cer (ES, E, A, I, L, Ld(), Kg())

    SELECT CASE ES
       CASE 1, 2
          N = 2
       CASE 3
          N = 1
```

```
    END SELECT
    DIM Kf(N, N), T(6, N)

    CALL Kf.Cer(ES, E, A, I, L, Kf())
    CALL T.Cer(ES, L, T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())

END SUB

' ================================ Kg.CerTag ================================
'
'   Procedura per la definizione della matrice di rigidezza globale
'   per un'asta incernierata agli estremi
'   tenendo conto della deformazione a taglio
'
'   Variabili di ingresso:
'      ES        estremo incernierato
'                   1 = primo estremo
'                   2 = secondo estremo
'                   3 = entrambi gli estremi
'      E         modulo di elasticita' normale
'      G         modulo di elasticita' tangenziale
'      CHI       fattore di taglio
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      Ld()      matrice di rotazione
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Kf.CerTag
'      Kg.f
'      T.Cer
'
' ---------------------------------------------------------------------------
'
SUB Kg.CerTag (ES, E, G, CHI, A, I, L, Ld(), Kg())

    SELECT CASE ES
       CASE 1, 2
          N = 2
       CASE 3
          N = 1
    END SELECT
    DIM Kf(N, N), T(6, N), Kloc(6, 6)

    CALL Kf.CerTag(ES, E, G, CHI, A, I, L, Kf())
    CALL T.Cer(ES, L, T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())

END SUB

' ================================ Kg.f ================================
'
'   Procedura generale per la definizione della matrice di rigidezza globale
'   a partire dalla matrice di rigidezza fondamentale
'
```

```
'   Variabili di ingresso:
'      Ld()      matrice di rotazione
'      T()       matrice di trasformazione
'      Kf()      matrice di rigidezza fondamentale
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Variabili interne:
'      LT()      prodotto di Ld per T
'      LTT()     trasposta del prodotto di Ld per T
'
'   Procedure utilizzate:
'      ProdottoMat
'      TrasponeMat
'
' ------------------------------------------------------------------------------
'
SUB Kg.f (Ld(), T(), Kf(), Kg())

    CALL LimitiMat(Kf(), DK, LRK, LCK, NRK, NCK)
    DIM LT(6, NRK), LTT(NRK, 6), K1(6, NRK)

    CALL ProdottoMat(Ld(), T(), LT())
    CALL TrasponeMat(LT(), LTT())
    CALL ProdottoMat(LT(), Kf(), K1())
    CALL ProdottoMat(K1(), LTT(), Kg())

END SUB

' ============================== Kg.Inc ==============================
'
'   Procedura per la definizione della matrice di rigidezza globale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'
'   Variabili di ingresso:
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      Ld()      matrice di rotazione
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Kf.Inc
'      Kg.f
'      T.Inc
'
' ------------------------------------------------------------------------------
'
SUB Kg.Inc (E, A, I, L, Ld(), Kg())

    DIM Kf(3, 3), T(6, 3), Kloc(6, 6)
    CALL Kf.Inc(E, A, I, L, Kf())
    CALL T.Inc(L, T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())
```

```
END SUB

' ============================== Kg.IncTag ==============================
'
'   Procedura per la definizione della matrice di rigidezza globale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'   tenendo conto della deformazione a taglio
'
'   Variabili in ingresso:
'      E         modulo di elasticita' normale
'      G         modulo di elasticita' tangenziale
'      CHI       fattore di taglio
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      Ld()      matrice di rotazione
'
'   Variabili in uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Kf.IncTag
'      Kg.f
'      T.Inc
'
' ------------------------------------------------------------------------
'
SUB Kg.IncTag (E, G, CHI, A, I, L, Ld(), Kg())

    DIM Kf(3, 3), T(6, 3), Kloc(6, 6)
    CALL Kf.IncTag(E, G, CHI, A, I, L, Kf())
    CALL T.Inc(L, T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())

END SUB

' ================================ Kg.1 ================================
'
'   Procedura generale per la definizione della matrice di rigidezza globale
'   a partire dalla matrice di rigidezza nel sistema di riferimento locale
'
'   Variabili di ingresso:
'      Ld()      matrice di rotazione
'      Kloc()    matrice di rigidezza locale
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      ProdottoMat
'      TrasponeMat
'
' ------------------------------------------------------------------------
'
SUB Kg.1 (Ld(), Kloc(), Kg())

    DIM LdT(6, 6)
    CALL ProdottoMat(Ld(), Kloc(), Kg())
    CALL TrasponeMat(Ld(), LdT())
```

```
    CALL ProdottoMat(Kg(), LdT(), Kg())

END SUB

' ============================== Kg.NodRig ==============================
'
'   Procedura per la determinazione della matrice di rigidezza globale
'   nel caso di nodi rigidi non puntiformi
'   (asta con tratti rigidi non coassiali)
'
'   Variabili di ingresso:
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale
'      I         momento d'inerzia della sezione trasversale
'      L         lunghezza dell'asta
'      Sx        coefficiente lunghezza x tratto sinistro
'      Sy        coefficiente lunghezza y tratto sinistro
'      Tx        coefficiente lunghezza x tratto destro
'      Ty        coefficiente lunghezza y tratto destro
'      Ld()      matrice di rotazione
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Kf.NodRig
'      Kg.f
'      T.Inc
'
' ------------------------------------------------------------------------------
'
SUB Kg.NodRig (E, A, I, L, Sx, Sy, Tx, Ty, Ld(), Kg())

    DIM Kf(3, 3), T(6, 3), Kloc(6, 6)
    CALL Kf.NodRig(E, A, I, L, Sx, Sy, Tx, Ty, Kf())
    CALL T.Inc(L, T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())

END SUB

' ============================== Kg.Rig ==============================
'
'   Procedura per la definizione della matrice di rigidezza globale
'   per un'asta perfettamente incastrata ai nodi di estremita'
'   con tratti rigidi di estremita'
'
'   Variabili di ingresso:
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      S         coefficiente lunghezza tratto rigido sinistro
'      T         coefficiente lunghezza tratto rigido destro
'      Ld()      matrice di rotazione
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
```

```
'      Kf.Rig
'      Kg.f
'      T.Inc
'
' ------------------------------------------------------------------------------
'
SUB Kg.Rig (E, A, I, L, S, T, Ld(), Kg())

    DIM Kf(3, 3), T(6, 3), Kloc(6, 6)
    CALL Kf.Rig(E, A, I, L, S, T, Kf())
    CALL T.Inc(L, T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())

END SUB

' ============================== Kg.Wnk ==============================
'
'   Procedura per la definizione della matrice di rigidezza globale
'   per un'asta su suolo elastico alla Winkler
'
'   Variabili di ingresso:
'      K         costante di sottofondo
'      B         larghezza trave
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      L         lunghezza dell'asta
'      Ld()      matrice di rotazione
'
'   Variabili di uscita:
'      Kg()      matrice di rigidezza globale
'
'   Procedure utilizzate:
'      Kf.Wnk
'      Kg.f
'      T.Wnk
'
' ------------------------------------------------------------------------------
'
SUB Kg.Wnk (K, B, E, A, I, L, Ld(), Kg())

    DIM Kf(5, 5), T(6, 5), Kloc(6, 6)
    CALL Kf.Wnk(K, B, E, A, I, L, Kf())
    CALL T.Wnk(T())
    CALL Kg.f(Ld(), T(), Kf(), Kg())

END SUB

' ============================== Kl. ==============================
'
'   Procedura generale per la definizione della matrice di rigidezza locale
'   a partire dalla matrice di rigidezza fondamentale
'
'   Variabili di ingresso:
'      T()       matrice di trasformazione
'      Kf()      matrice di rigidezza fondamentale
'
'   Variabili di uscita:
'      Kloc()    matrice di rigidezza locale
```

```
'
'   Procedure utilizzate:
'      LimitiMat
'      ProdottoMat
'      TrasponeMat
'
' ------------------------------------------------------------------------------
'
SUB K1. (T(), Kf(), Kloc())

    CALL LimitiMat(Kf(), DK, LRK, LCK, NRK, NCK)
    DIM K1(6, NRK), TT(NRK, 6)

    CALL ProdottoMat(T(), Kf(), K1())
    CALL TrasponeMat(T(), TT())
    CALL ProdottoMat(K1(), TT(), Kloc())

END SUB

' ========================== S1.Cer.Qunif ==========================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta incernierata agli estremi: carico uniforme
'
'   Variabili di ingresso:
'      ES        estremo incernierato
'                   1 = primo estremo
'                   2 = secondo estremo
'                   3 = entrambi gli estremi
'      Q         carico uniformemente distribuito ortogonale all'asse
'      N         carico uniformemente distribuito parallelo  all'asse
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      S1()      vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB S1.Cer.Qunif (ES, Q, N, L, S1())

    CALL AzzeraMat(S1())
    S1(1, 1) = -N * L / 2
    S1(4, 1) = S1(1, 1)
    SELECT CASE ES
       CASE 1
          S1(2, 1) = -3 / 8 * L * Q
          S1(5, 1) = -5 / 8 * L * Q
          S1(6, 1) = 1 / 8 * L ^ 2 * Q
       CASE 2
          S1(2, 1) = -5 / 8 * L * Q
          S1(3, 1) = -1 / 8 * L ^ 2 * Q
          S1(5, 1) = -3 / 8 * L * Q
       CASE 3
          S1(2, 1) = -1 / 2 * L * Q
          S1(5, 1) = S1(2, 1)
    END SELECT
```

```
END SUB

' ================================ Sl.Inc.DT ================================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta perfettamente incastrata ai nodi di estremita':
'   distorsione termica trapezia
'
'   Variabili di ingresso:
'      DTu       variazione termica uniforme
'      DTf       variazione termica a farfalla
'      E         modulo di elasticita' normale
'      A         area della sezione trasversale dell'asta
'      I         momento d'inerzia della sezione trasversale dell'asta
'      H         altezza della sezione trasversale dell'asta
'      At        coefficiente di dilatazione termica
'
'   Variabili di uscita:
'      Sl()      vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      ===
'
' ---------------------------------------------------------------------------
'
SUB Sl.Inc.DT (DTu, DTf, E, A, I, H, At, Sl())

    Sl(1, 1) = E * A * At * DTu
    Sl(2, 1) = 0
    Sl(3, 1) = -E * I * At / H * DTf
    Sl(4, 1) = -Sl(1, 1)
    Sl(5, 1) = 0
    Sl(6, 1) = Sl(3, 1)

END SUB

' ================================ Sl.Inc.FM ================================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta perfettamente incastrata ai nodi di estremita':
'   forza F e coppia M concentrate
'
'   Variabili di ingresso:
'      Fx        componente della forza concentrata parallela all'asta
'      Fy        componente della forza concentrata normale all'asta
'      Mz        coppia concentrata
'      XO        distanza dal primo estremo delle azioni concentrate
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Sl()      vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      ===
'
' ---------------------------------------------------------------------------
'
SUB Sl.Inc.FM (Fx, Fy, Mz, XO, L, Sl())
```

```
    IF XO < 0 OR XO > L THEN
       PRINT "ERRORE 1 in Sl.Inc.FM - valori dell'ascissa non accettabili"
       END
    END IF

    A = XO / L
    B = 1 - A
    Sl(1, 1) = -B * Fy
    Sl(2, 1) = -B ^ 2 * (1 + 2 * A) * Fx + 6 * A * B / L * Mz
    Sl(3, 1) = -A * B ^ 2 * L * Fx + B * (2 * A - B) * Mz
    Sl(4, 1) = -A * Fy
    Sl(5, 1) = -A ^ 2 * (1 + 2 * B) * Fx - 6 * A * B / L * Mz
    Sl(6, 1) = A ^ 2 * B * L * Fx + A * (2 * B - A) * Mz

END SUB

' ================================ Sl.Inc.Qunif ================================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta perfettamente incastrata ai nodi di estremita':
'   carico uniformemente distribuito
'
'   Variabili di ingresso:
'      Q         carico uniformemente distribuito ortogonale all'asta
'      N         carico uniformemente distribuito parallelo  all'asta
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      Sl()      vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      ===
' ------------------------------------------------------------------------------
'
SUB Sl.Inc.Qunif (Q, N, L, Sl())

    Sl(1, 1) = -N * L / 2
    Sl(2, 1) = -Q * L / 2
    Sl(3, 1) = -Q * L ^ 2 / 12
    Sl(4, 1) = Sl(1, 1)
    Sl(5, 1) = Sl(2, 1)
    Sl(6, 1) = -Sl(3, 1)

END SUB

' ================================ Sl.Inc.Qvar =================================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta perfettamente incastrata ai nodi di estremita':
'   carico distribuito con andamento lineare su parte dell'asta
'
'   Variabili di ingresso:
'      Q1        valore del carico a sinistra
'      Q2        valore del carico a destra
'      L         lunghezza dell'asta
'      X1        distanza dal primo estremo dell'inizio carico
'      X2        distanza dal primo estremo della fine carico
'
```

```
'   Variabili di uscita:
'      S1()       vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      ===
'
' ------------------------------------------------------------------------------
'
SUB S1.Inc.Qvar (Q1, Q2, L, X1, X2, S1())

    IF X1 < 0 OR X1 > L OR X2 < 0 OR X2 > L THEN
       PRINT "ERRORE 1 in S1.Inc.Qvar - valori delle ascisse non accettabili"
       END
    END IF
    IF X1 > X2 THEN
       PRINT "ERRORE 2 in S1.Inc.Qvar - ordine delle ascisse non accettabile"
       END
    END IF

    r1 = 3 / 2 - (X1 + X2 / 2) / L
    s1 = 3 - 2 * r1
    r2 = (X2 + X1 / 2) / L
    s2 = 3 - 2 * r2
    T = (X2 - X1) / L

    r = r1: S = s1
    GOSUB FGH
    f1 = F: g1 = G: h1 = H

    r = r2: S = s2
    GOSUB FGH
    f2 = F: g2 = G: h2 = H

    S1(1, 1) = 0
    S1(2, 1) = -(20 * r1 * T + h1) / 60 * L * Q1 - (10 * s2 * T - h2) / 60 * L * Q2
    S1(3, 1) = -f1 * L ^ 2 / 20 * Q1 - g2 * L ^ 2 / 30 * Q2
    S1(4, 1) = 0
    S1(5, 1) = -(10 * s1 * T - h1) / 60 * L * Q1 - (20 * r2 * T - h2) / 60 * L * Q2
    S1(6, 1) = g1 * L ^ 2 / 30 * Q1 + f2 * L ^ 2 / 20 * Q2

    EXIT SUB

FGH:
    F = T / 27 * (40 * r ^ 2 * S + T ^ 2 * (2 * T - 30 * r + 15))
    G = T / 18 * (20 * r * S ^ 2 - T ^ 2 * (2 * T - 30 * r + 30))
    H = 3 * F - 2 * G
    RETURN

END SUB

' ============================== S1.Rig.Qunif ==============================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta perfettamente incastrata ai nodi di estremita'
'   avente tratti rigidi di estremita': carico uniforme
'
'   Variabili di ingresso:
'      Q          carico uniformemente distribuito ortogonale all'asta
'      N          carico uniformemente distribuito parallelo  all'asta
```

```
'      L          lunghezza asta
'      S          coefficiente lunghezza tratto rigido sinistro
'      T          coefficiente lunghezza tratto rigido destro
'
'   Variabili di uscita:
'      Sl()       vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      ===
'
' ------------------------------------------------------------------------------
'
SUB Sl.Rig.Qunif (Q, N, L, S, T, Sl())

    Z1 = (1 - S - T) ^ 2 + 6 * S * (1 - S - T) + 6 * S ^ 2
    Z2 = (1 - S - T) ^ 2 + 6 * T * (1 - S - T) + 6 * T ^ 2

    Sl(1, 1) = -N * L / 2
    Sl(2, 1) = -(1 + S - T) / 2 * L * Q
    Sl(3, 1) = -Z1 * L ^ 2 / 12 * Q
    Sl(4, 1) = Sl(1, 1)
    Sl(5, 1) = -(1 - S + T) / 2 * L * Q
    Sl(6, 1) = Z2 * L ^ 2 / 12 * Q

END SUB

' ========================== Sl.Wnk.Qunif ==========================
'
'   Procedura per il calcolo delle azioni di incastro perfetto
'   per un'asta su suolo elastico alla Winkler: carico uniforme
'
'   Variabili di ingresso:
'      Q          carico uniformemente distribuito ortogonale all'asta
'      N          carico uniformemente distribuito parallelo  all'asta
'      L          lunghezza dell'asta
'      K          costante di sottofondo
'      B          larghezza della trave
'      Kloc()     matrice di rigidezza locale
'
'   Variabili di uscita:
'      Sl()       vettore azioni di incastro perfetto
'
'   Procedure utilizzate:
'      AzzeraMat
'      ProdottoMat
'
' ------------------------------------------------------------------------------
'
SUB Sl.Wnk.Qunif (Q, N, L, K, B, Kloc(), Sl())

    CALL AzzeraMat(Sl())
    Sl(2, 1) = -1 / (K * B) * Q
    Sl(5, 1) = Sl(2, 1)
    CALL ProdottoMat(Kloc(), Sl(), Sl())
    Sl(1, 1) = Sl(1, 1) - N * L / 2
    Sl(4, 1) = Sl(4, 1) - N * L / 2

END SUB
```

```
' ================================ T.Cer ================================
'
'   Procedura per la definizione della matrice di trasformazione
'   per un'asta incernierata agli estremi
'
'   Variabili di ingresso:
'      ES        estremo incernierato
'                   1 = primo estremo
'                   2 = secondo estremo
'                   3 = entrambi gli estremi
'      L         lunghezza dell'asta
'
'   Variabili di uscita:
'      T()       matrice di trasformazione
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ---------------------------------------------------------------------------
'
SUB T.Cer (ES, L, T())

   CALL AzzeraMat(T())
   SELECT CASE ES
      CASE 1
         T(1, 2) = -1
         T(2, 1) = 1 / L
         T(4, 2) = 1
         T(5, 1) = -1 / L
         T(6, 1) = 1
      CASE 2
         T(1, 2) = -1
         T(2, 1) = 1 / L
         T(3, 1) = 1
         T(4, 2) = 1
         T(5, 1) = -1 / L
      CASE 3
         T(1, 1) = -1
         T(4, 1) = 1
   END SELECT

END SUB

' ================================ T.Inc ================================
'
'   Procedura per la definizione della matrice di trasformazione
'   per un'asta perfettamente incastrata ai nodi di estremita'
'
'   Variabili di ingresso:
'      L      lunghezza dell'asta
'
'   Variabili di uscita:
'      T()    matrice di trasformazione
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ---------------------------------------------------------------------------
'
```

```
SUB T.Inc (L, T())

    CALL AzzeraMat(T())
    T(1, 3) = -1
    T(2, 1) = 1 / L
    T(2, 2) = 1 / L
    T(3, 1) = 1
    T(4, 3) = 1
    T(5, 1) = -1 / L
    T(5, 2) = -1 / L
    T(6, 2) = 1

END SUB

' ================================ T.Wnk ================================
'
'   Procedura per la definizione della matrice di trasformazione
'   per un'asta su suolo elastico alla Winkler
'
'   Variabili di ingresso:
'      ===
'
'   Variabili di uscita:
'      T()       matrice di trasformazione
'
'   Procedure utilizzate:
'      AzzeraMat
'
' ------------------------------------------------------------------------------
'
SUB T.Wnk (T())

    CALL AzzeraMat(T())
    T(1, 5) = -1
    T(2, 1) = 1
    T(3, 2) = 1
    T(4, 5) = 1
    T(5, 3) = 1
    T(6, 4) = 1

END SUB
```

3. Esempio: diagramma delle caratteristiche di sollecitazione

3.1. Generalità

La conoscenza dell'andamento delle caratteristiche di sollecitazione lungo le aste è di fondamentale importanza per il progettista strutturale durante le fasi di verifica delle sezioni e di disposizione delle armature in strutture in cemento armato. La graficizzazione di tale andamento mediante diagrammi ne fornisce una visione immediata e globale che agevola

l'individuazione delle sezioni più sollecitate ed inoltre consente una valutazione qualitativa del comportamento della struttura, essenziale per esprimere un giudizio di accettazione dei risultati e per valutare la correttezza dei dati forniti al calcolatore.

Per i motivi citati si è ritenuto utile preparare una procedura, denominata `Diagramma`, che consente di disegnare sullo schermo il diagramma dello sforzo normale, del momento flettente o del taglio in un'asta. I valori assunti da questi enti alla generica ascissa x sono univocamente definiti dalla geometria dell'asta, dai carichi su essa agenti e dai valori delle caratteristiche di sollecitazione in un estremo (ottenuti come risultato della soluzione dello schema iperstatico).

Nel definire i possibili carichi agenti sull'asta, si è ipotizzata la presenza di:

- un carico uniformemente distribuito, con una componente parallela ed una ortogonale all'asse, applicato a tutta l'asta;
- due carichi distribuiti ortogonali all'asse, variabili con legge lineare ed applicati a parte dell'asta;
- tre insiemi di forze e momenti concentrati applicati in punti distinti all'interno dell'asta; per semplicità operativa si è supposto che questi punti siano forniti nell'ordine in cui si succedono nell'asta, inserendo comunque un controllo per verificare che tale condizione sia rispettata.

Alla presenza di azioni concentrate corrisponde una discontinuità del diagramma delle caratteristiche di sollecitazione. Per tale motivo si è divisa l'asta in quattro tratti, individuati dagli estremi e dai punti di applicazione delle azioni. Ciascun tratto è diviso in un numero intero di intervalli, di ampiezza non superiore ad $1/n$ della lunghezza totale dell'asta. Il numero n è stato conservato nella variabile `Punti`, che nel caso esaminato è stata posta uguale a 100 per ottenere un diagramma molto accurato; valori più bassi forniscono diagrammi meno precisi, ma richiedono un minor onere computazionale e quindi minori tempi di esecuzione. I valori assunti dalle caratteristiche di sollecitazione nei punti di divisione tra gli intervalli sono valutati mediante la routine `CalcolaCarSol`, che utilizza le procedure `CarSol.AzEstr`, `CarSol.FM`, `Carsol.Qunif`, `Carsol.Qvar`. La routine `TracciaDiagr`, infine, calcola le coordinate del punto rappresentativo del valore da diagrammare, ottenute spostandosi ortogonalmente all'asta di una quantità pari al valore moltiplicato per un fattore di scala `Sc` (fornito come dato d'ingresso alla procedura), ed unisce il punto in esame al precedente.

Si riporta di seguito il listato della procedura `Diagramma`, che fa parte del file `DIAGRAM.BAS`. Nel listato è contenuta anche la descrizione dei parametri

di scambio, delle principali variabili interne e delle procedure richiamate dal sottoprogramma.

```
' ═══════════════════════════ Diagramma ═══════════════════════════
'
'   Procedura per il disegno del diagramma
'   del momento flettente, del taglio o dello sforzo normale
'
'   NOTE: occorre essersi gia' posizionati nella modalita' grafica (SCREEN)
'         ed aver definito il campo di coordinate (WINDOW)
'
'   Variabili di ingresso:
'      X1         ascissa  del primo estremo dell'asta
'      Y1         ordinata del primo estremo dell'asta
'      X2         ascissa  del secondo estremo dell'asta
'      Y2         ordinata del secondo estremo dell'asta
'      S()        array che contiene le caratt. di sollecitaz. al primo estremo
'                    indice:  1 -> N,  2 -> M,  3 -> T
'      P()        array che contiene le componenti del carico uniforme sull'asta
'                    indice:  1 -> Px, 2 -> Py
'      F()        array bidimensionale che contiene le azioni concentrate
'                    primo indice:  1 -> Fx, 2 -> Fy, 3 -> Mz
'                    secondo indice: numero d'ordine della forza (1-3)
'      Xf()       vettore ascisse punti applicazione azioni concentrate (1-3)
'      Q()        array bidimensionale che contiene i carichi variabili
'                    primo indice:  1 -> Q1, 2 -> Q2
'                    secondo indice: numero d'ordine dei carichi (1-2)
'      Xq()       array bidimensionale che contiene le ascisse dei car. variabili
'                    primo indice:  1 -> X1, 2 -> X2
'                    secondo indice: numero d'ordine dei carichi (1-2)
'      Sc         fattore di scala della caratteristica da diagrammare
'      CS         caratteristica di sollecitazione da diagrammare:
'                    1 = sforzo normale
'                    2 = momento flettente
'                    3 = taglio
'
'   Variabili di uscita:
'      ==
'
'   Principali variabili interne:
'      L          lunghezza dell'asta
'      X          ascissa variabile
'      Nx,Mx,Tx   caratteristiche di sollecitazione all'ascissa X
'      XP,YP      coordinate del punto del diagramma
'      Punti      numero di punti dell'asta nei quali sono calcolati N, M, T
'
'   Procedure utilizzate:
'      CarSol.AzEstr
'      CarSol.FM
'      CarSol.Qunif
'      CarSol.Qvar
'
' ------------------------------------------------------------------------------
'
SUB Diagramma (X1, Y1, X2, Y2, S(), P(), F(), Xf(), Q(), Xq(), Sc, CS)

    Punti = 100
```

```
    ' calcola lunghezza asta e sue proiezioni
    Dx = X2 - X1
    Dy = Y2 - Y1
    L = SQR(Dx ^ 2 + Dy ^ 2)
    SnAl = Dy / L
    CsAl = Dx / L

    ' disegna l'asta
    LINE (X1, Y1)-(X2, Y2)

    ' disegna il diagramma
    PSET (X1, Y1)
    Xprec = 0
    FOR J = 1 TO 4
       IF J = 4 THEN
          Xsucc = L
       ELSE
          Xsucc = Xf(J)
       END IF
       DeltaX = Xsucc - Xprec
       IF Xsucc <> 0 AND DeltaX <> 0 THEN
          IF DeltaX < 0 THEN
             PRINT "ERRORE 1 in Diagramma - le azioni concentrate non sono assegnate"
             PRINT "                        nell'ordine in cui si succedono lungo
                                            l'asta"
             END
          END IF
          SZ = INT(DeltaX / L * Punti + 1)
          FOR S = 0 TO SZ
             X = Xprec + S * DeltaX / SZ
             GOSUB CalcolaCarSol
             GOSUB TracciaDiagr
          NEXT S
          Xprec = Xsucc
       END IF
    NEXT J
    LINE -(X2, Y2)
    EXIT SUB


CalcolaCarSol:

    ' per azioni di estremita'
    CALL Carsol.AzEstr(-S(1), S(3), -S(2), X, L, Nx, Mx, Tx)

    ' per carichi uniformemente distribuiti
    IF P(1) <> 0 OR P(2) <> 0 THEN
       CALL CarSol.Qunif(P(2), P(1), X, L, Sn, Mf, Ta)
       Nx = Nx + Sn
       Mx = Mx + Mf
       Tx = Tx + Ta
    END IF

    ' per forze e coppie concentrate
    FOR J = 1 TO 3
       IF F(1, J) <> 0 OR F(2, J) <> 0 OR F(3, J) <> 0 THEN
          CALL Carsol.FM(F(1, J), F(2, J), F(3, J), Xf(J), X, L, Sn, Mf, Ta)
```

```
            Nx = Nx + Sn
            Mx = Mx + Mf
            Tx = Tx + Ta
         END IF
      NEXT J

      ' per carichi distribuiti con legge lineare
      FOR J = 1 TO 2
         IF Q(1, J) <> 0 OR Q(2, J) <> 0 THEN
            CALL Carsol.Qvar(Q(1, J), Q(2, J), Xq(1, J), Xq(2, J), X, L, Mf, Ta)
            Mx = Mx + Mf
            Tx = Tx + Ta
         END IF
      NEXT J

RETURN


TracciaDiagr:

      SELECT CASE CS
         CASE 1
            V = Nx * Sc
         CASE 2
            V = -Mx * Sc
         CASE 3
            V = Tx * Sc
      END SELECT
      XP = X1 + X * CsAl - V * SnAl
      YP = Y1 + X * SnAl + V * CsAl
      LINE -(XP, YP)

RETURN

END SUB
```

3.2. Applicazione

La procedura `Diagramma` può essere utilizzata ciclicamente per tracciare il diagramma delle caratteristiche della sollecitazione per un intero schema intelaiato. Un esempio di ciò è costituito dal programma principale del file `DIAGRAM.BAS`, listato nelle pagine seguenti. Esso legge i dati relativi ad uno schema geometrico e a due schemi di carico e consente di scegliere quale caratteristica diagrammare, e per quale schema di carico la si vuole. Poiché il programma ha solo finalità esemplificative, non ci si è curati particolarmente della fase di input (i dati sono contenuti nello stesso programma in istruzioni `DATA`) né della individuazione di criteri generali per la scelta del fattore di scala (fornito anch'esso come dato). Il programma può comunque essere usato dal lettore sostituendo il blocco dati con nuovi valori, oppure adattato per una lettura dei dati dal disco o, meglio ancora, trasformato

in un sottoprogramma che operi in coda al programma di calcolo ricevendo automaticamente da esso le informazioni necessarie per la diagrammazione.

I dati utilizzati nel programma si riferiscono al portale mostrato in figura 4.1, soggetto a due schemi di carico. Il primo prevede un carico verticale uniformemente distribuito sui traversi, il secondo due forze concentrate verticali su un'asta ed una coppia concentrata al centro di un'altra asta. Non si sono quindi sfruttate tutte le potenzialità della procedura `Diagramma` (forze in tre punti distinti, carichi variabili), e questo è evidenziato dalla presenza di numerosi zeri tra i dati.

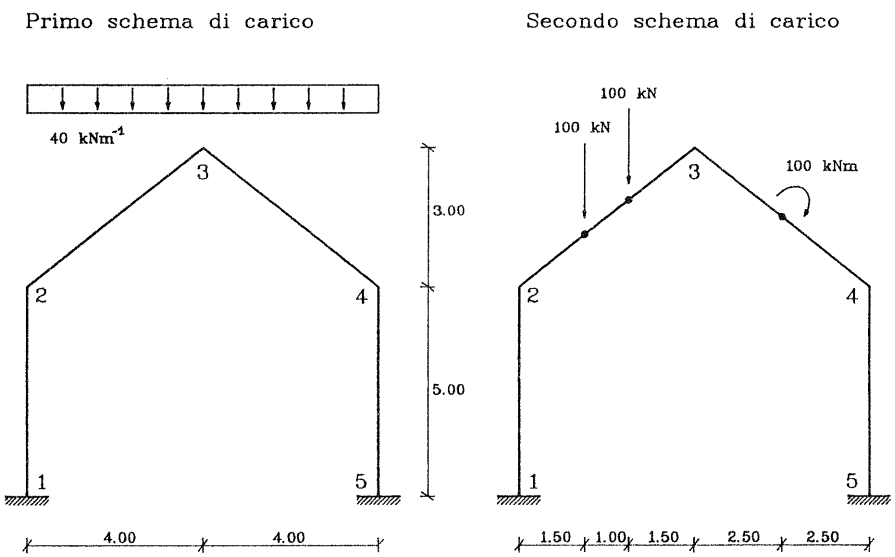


Fig. 4.1 — Portale soggetto a due schemi di carico

Nella figura 4.2 sono riuniti i sei diagrammi ottenuti come risultato dell'elaborazione grafica. Si noti che il taglio e lo sforzo normale mantengono lo stesso segno indipendentemente dal verso di percorrenza dell'asta (cioè da quale nodo è scelto come primo estremo); i loro valori positivi sono riportati nel verso positivo dell'asse locale y e quindi da un lato o dall'altro dell'asta a seconda di qual'è il primo e quale il secondo estremo.

Invece il momento flettente è riportato sempre dal lato delle fibre tese, indipendentemente dal verso dell'asta, perché invertendo l'ordine dei nodi si modifica anche il segno di esso.

Primo schema di carico

Secondo schema di carico

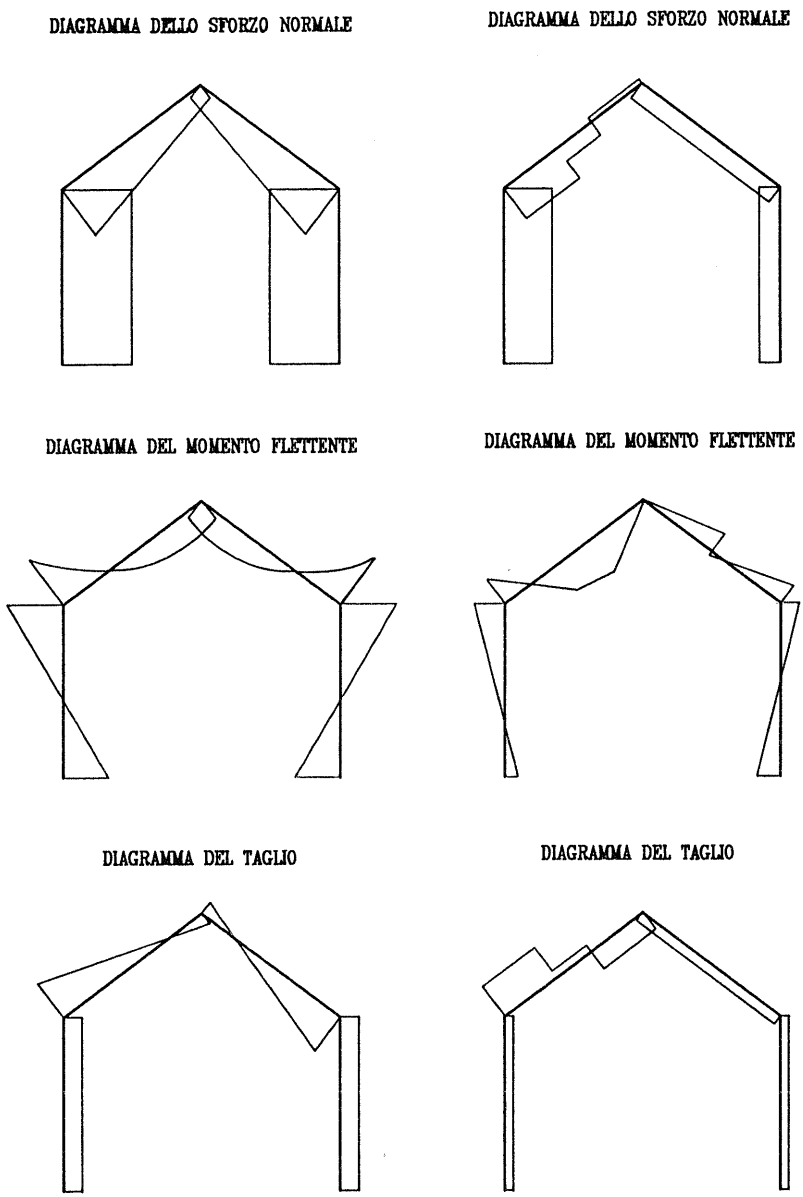


Fig. 4.2 — Diagramma delle caratteristiche di sollecitazione del portale

```
'                                DIAGRAM.BAS
'
'     tracciamento dei diagrammi delle caratteristiche di sollecitazione
'
'                                                              rev. 09.89

' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file ASTA.BAS
DECLARE SUB Carsol.AzEstr (Fx!, Fy!, Mz!, X!, L!, Sn!, Mf!, Ta!)
DECLARE SUB Carsol.FM (Fx!, Fy!, Mz!, XO!, X!, L!, Sn!, Mf!, Ta!)
DECLARE SUB CarSol.Qunif (Q!, N!, X!, L!, Sn!, Mf!, Ta!)
DECLARE SUB Carsol.Qvar (Q1!, Q2!, X1!, X2!, X!, L!, Mf!, Ta!)

' procedure di questo file
DECLARE SUB Diagramma (X1!, Y1!, X2!, Y2!, S!(), P!(), F!(), Xf!(), Q!(), Xq!(),
                       Sc!, CS!)

' --------------------------- Programma principale --------------------------------

SCREEN 0

' lettura della geometria dello schema
READ IZ, JZ                         ' numero di nodi e di aste
DIM X(IZ), Y(IZ), E1(JZ), E2(JZ), S(3), P(2), F(3, 3), Xf(3), Q(2, 2), Xq(2, 2)
FOR I = 1 TO IZ
   READ X(I), Y(I)                  ' coordinate dei nodi
NEXT I
FOR J = 1 TO JZ
   READ E1(J), E2(J)                ' nodi di estremita' delle aste
NEXT J
READ Xmin, Xmax, Ymin, Ymax         ' estremi dell'area grafica
READ Sc                             ' fattore di scala per i diagrammi

DO

   ' chiede quale diagramma visualizzare
   CLS
   LOCATE 3, 28: PRINT "DIAGRAM.BAS - rev. 09.89"
   PRINT
   PRINT "      Esempio di utilizzazione della procedura  Diagramma  per il disegno"
   PRINT "      dei diagrammi delle caratteristiche di sollecitazione in un telaio."
   PRINT
   PRINT "      Il programma e' descritto nel capitolo 4, parte seconda, del libro:"
   PRINT "      A.Ghersi, R.Coraggio,  Il personal computer nel calcolo di edifici."
   PRINT "      Analisi matriciale  di strutture intelaiate,  CUEN,  Napoli,  1990."
   PRINT
   PRINT "      ATTENZIONE: Il programma richiede scheda grafica CGA o equivalente."
   PRINT
   PRINT
   PRINT "      Scegli il diagramma da visualizzare:"
   PRINT
   PRINT "      Schema di carico 1                          Schema di carico 2"
   PRINT
   PRINT "      1 - sforzo normale                          4 - sforzo normale"
   PRINT "      2 - momento flettente                       5 - momento flettente"
   PRINT "      3 - taglio                                  6 - taglio"
   PRINT
```

```
   PRINT "                              0 - fine esecuzione"
   DO
      K$ = INKEY$
   LOOP UNTIL K$ <> ""
   K = ASC(K$) - 48
   SELECT CASE K
      CASE 1, 2, 3
         RESTORE Schema1
         CS = K
      CASE 4, 5, 6
         RESTORE Schema2
         CS = K - 3
      CASE ELSE
   END SELECT
   IF K >= 1 AND K <= 6 THEN
      SCREEN 2
      WINDOW (Xmin, Ymin)-(Xmax, Ymax)
      CLS
      LOCATE 1, 26
      SELECT CASE CS
         CASE 1
            PRINT "DIAGRAMMA  DELLO SFORZO NORMALE"
         CASE 2
            PRINT "DIAGRAMMA DEL MOMENTO FLETTENTE"
         CASE 3
            PRINT "      DIAGRAMMA DEL TAGLIO"
      END SELECT
      ' lettura dei carichi sulle aste e disegno diagramma
      FOR J = 1 TO JZ
         READ S(1), S(2), S(3)                          ' azioni di estremita'
         READ P(1), P(2)                                ' carichi uniformi
         FOR N = 1 TO 3
            READ F(1, N), F(2, N), F(3, N), Xf(N)       ' azioni concentrate
         NEXT N
         FOR N = 1 TO 2
            READ Q(1, N), Q(2, N), Xq(1, N), Xq(2, N)   ' carichi variabili
         NEXT N
         X1 = X(E1(J))
         Y1 = Y(E1(J))
         X2 = X(E2(J))
         Y2 = Y(E2(J))
         CALL Diagramma(X1, Y1, X2, Y2, S(), P(), F(), Xf(), Q(), Xq(), Sc, CS)
      NEXT J
      LOCATE 24, 1
      PRINT "Per continuare premi un tasto qualsiasi";
      DO
         K$ = INKEY$
      LOOP UNTIL K$ <> ""
   END IF

LOOP UNTIL K = 0
SCREEN 0
END

DATA 5,4
DATA 0,0,0,5,4,8,8,5,8,0
DATA 1,2,2,3,3,4,4,5
DATA -6,14,-3,10
DATA .01
```

```
Schema1:
DATA -200,128.30,-58.14,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA -166.51,-162.40,125.12,-24,-32
DATA 0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA -46.51,63.18,34.88,24,-32
DATA 0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA -200,-162.40,58.14,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0

Schema2:
DATA -141.43,36.89,-25.23,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA -105.04,-89.28,98.00,0,0
DATA -60,-80,0,1.875,-60,-80,0,3.125,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA -55.33,0.75,-31.72,0,0
DATA 0,0,-100,2.5,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA -58.57,-57.83,25.23,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
```

CAPITOLO QUINTO

L'INSIEME PIANO DI ASTE

1. Geometria dello schema e tipologia dei carichi

Nell'organizzare un programma di calcolo strutturale ci si trova sempre alle prese con due esigenze: da un lato si desidera conferire ad esso la massima generalità, per poterlo utilizzare anche in quelle situazioni particolari che ogni tanto si incontrano nell'attività professionale; dall'altro si vuol realizzare uno strumento idoneo ad un uso quotidiano e quindi facile e rapido da utilizzare. Purtroppo queste due esigenze non sono facilmente conciliabili, perché un programma generale è inevitabilmente più complesso di un programma orientato ad un problema specifico. L'approccio più corretto è quello di realizzare più programmi differenti, con diverso livello di generalizzazione. Anche in tal caso, però, si è costretti a scelte indubbiamente soggettive e quindi discutibili.

Volendo presentare un programma per la risoluzione di un insieme piano di aste come applicazione concreta della teoria esposta, ci si imbatte nella problematica citata. Il progettista di edifici incontra con la massima frequenza lo schema di telaio a maglie rettangolari; capita però ogni tanto un telaio con travi a ginocchio o altri elementi che disturbano la regolarità dello schema, oppure la necessità di inserirvi travi di fondazione, e via via, sempre più raramente, tutte le altre situazioni teoricamente esposte. Si è quindi realizzato innanzitutto un programma base, abbastanza generale, che viene descritto in questo capitolo. In esso sono inserite solo alcune delle

situazioni particolari descritte nel capitolo 2 della prima parte e codificate nel capitolo 4 della seconda parte; l'impostazione data dovrebbe però consentire al lettore di arricchire il programma con quei modelli di aste o di carico che egli riterrà necessari per le sue applicazioni. Nel capitolo successivo, invece, si affronta il caso del telaio a maglie rettangolari, mantenendo inalterata la procedura di calcolo qui sviluppata ma anteponendo ad essa un preprocessor che consente di fornire i dati in una maniera più sintetica, consona alla semplicità dello schema, ed aggiungendo in coda un postprocessor che articola i risultati nella veste schematica tipica dei telai a maglie rettangolari.

Il programma base, denominato `TELGEN`, consente di risolvere uno schema piano costituito da un insieme di aste di due tipi, ciascuno dei quali sarà nel seguito contraddistinto con una lettera:

tipo T – trave alla De Saint Venant, solidale ai nodi di estremità; nel denominare le procedure descritte nel capitolo precedente si è usata per questo tipo di asta la sigla `Inc`;

tipo W – trave collegata ad un suolo elastico alla Winkler, che esplica azioni ortogonali all'asse proporzionali allo spostamento in tale direzione; nel capitolo precedente si è usato per questo tipo di asta la sigla `Wnk`.

Per aste del primo tipo (ma non del secondo) si è prevista la possibilità di tener conto della deformazione dovuta al taglio.

Tutte le aste devono essere a sezione costante. I dati necessari per descrivere una sezione sono diversi a seconda della forma della sezione. Nel programma si sono previste due forme diverse, ciascuna contraddistinta da una lettera:

forma R – sezione di forma rettangolare, per la quale è sufficiente indicare i valori della base e dell'altezza;

forma G – sezione di forma generica, per la quale occorre indicare l'area, il rapporto tra area e fattore di taglio, il momento d'inerzia.

Il lettore potrà utilizzare le procedure descritte nel capitolo precedente per ampliare il programma, aggiungendo altri tipi di asta, ad esempio:

tipo C – trave alla De Saint Venant con cerniere ad uno o ad entrambi gli estremi;

tipo R – trave con tratti infinitamente rigidi agli estremi;

tipo N – trave con tratti rigidi non coassiali (in presenza di nodi rigidi non puntiformi).

Potrà inoltre prevedere altre forme di sezioni, come ad esempio:

forma C – sezione di forma circolare;

forma A – sezione di forma anulare;

forma T – sezione di forma a T.

Il programma consente di risolvere più schemi di carico, in ciascuno dei quali sono presenti:

- azioni concentrate sui nodi, definite mediante le componenti F_x ed F_y, parallele agli assi, ed il momento M_z.
- carichi uniformemente distribuiti sulle aste, definiti mediante le componenti p_x e p_y, parallele agli assi del sistema di riferimento globale, valutate per unità di lunghezza dell'asta.

Altre tipologie di carico possono essere aggiunte dal lettore, utilizzando in fase di risoluzione le procedure descritte nel capitolo precedente.

Le ipotesi alla base del procedimento risolutivo sono state ampiamente discusse nella prima parte del testo. Si ricorda in particolare che si è supposto:

- che gli spostamenti dei nodi della struttura, provocati dai carichi su essa agenti, siano piccoli rispetto alle sue dimensioni;
- che siano irrilevanti le non linearità geometriche (effetto $P - \delta$ o effetto instabilizzante dei carichi verticali), cioè che sia indifferente applicare i carichi sulla struttura deformata o su quella indeformata;
- che il materiale abbia un comportamento linearmente elastico e non si abbia quindi plasticizzazione delle sezioni.

L'organizzazione complessiva del programma è descritta in dettaglio nel paragrafo successivo. È però opportuno evidenziare e discutere fin da ora una scelta di base, relativa al calcolo ed alla memorizzazione delle matrici di rigidezza $\mathbf{k}_g$ e di rotazione $\mathbf{L}_d$ e dei vettori di incastro perfetto $\overline{\mathbf{S}}$ delle aste. Questi vengono utilizzati in tre fasi distinte del programma: nella valutazione della matrice di rigidezza della struttura, nella determinazione del vettore termini noti e nel calcolo delle caratteristiche di sollecitazione agli estremi delle aste. Si sono prese in considerazione tre alternative:

- calcolare una sola volta tali matrici e vettori per le aste e mantenerle tutte contemporaneamente nella memoria centrale del calcolatore;
- calcolarle una sola volta, conservarle in un file sequenziale della memoria di massa e richiamarle da esso quando necessario;
- calcolarle singolarmente ogni volta che occorrono, senza memorizzarle stabilmente.

La prima alternativa ha il vantaggio di ridurre al massimo i tempi di esecuzione; richiede però un impegno di memoria non trascurabile e quindi può comportare, rispetto alle altre, una riduzione del massimo numero di nodi ed aste esaminabili col programma.

La seconda alternativa consente di evitare l'ingombro di memoria centrale innanzi citato. Dal punto di vista della velocità di esecuzione, non è però tanto positiva quanto apparirebbe a prima vista, perché la lettura di valori da memoria di massa è lenta (specie nel caso di dischetti flessibili) ed i tempi necessari possono essere dello stesso ordine di grandezza di quelli occorrenti per ricalcolare le matrici.

La terza alternativa, infine, permette di evitare l'ingombro di memoria ma comporta tempi di esecuzione indubbiamente maggiori che nel primo caso.

La scelta tra queste diverse possibilità era veramente problematica nel passato, quando i calcolatori erano meno veloci e con capacità di memoria piuttosto bassa e quindi si doveva effettivamente limitare l'uso del programma a schemi semplici oppure accettare attese lunghissime. Il rapido progresso ha affievolito le differenze ed ha reso opinabile e contingente qualunque preferenza. Così ad esempio nell'impostare per la prima volta (non molto tempo fa) il programma si è scelta la terza alternativa, ritenendo preferibile privilegiare l'ampiezza della struttura massima risolubile anziché la velocità di esecuzione. Oggi, nel mandare il libro in stampa, la memoria standard dei personal computer ha superato il limite dei 640 K, direttamente gestibili dell'attuale versione del sistema operativo MS–DOS, e potrebbe essere preferibile adottare la seconda via, usando come disco virtuale la memoria del calcolatore eccedente i 640 K, che altrimenti resterebbe sprecata. Ma sicuramente quando questo libro verrà letto la situazione sarà ancora diversa ... e il lettore sarà così invogliato a intervenire di persona ed adattare il programma alla nuova realtà.

2. Descrizione del programma

Secondo l'impostazione tipica della programmazione strutturata, il pro-

gramma TELGEN è costituito da un programma principale, che evidenzia in maniera sintetica le operazioni da esso compiute, ed una serie di routine, le quali a loro volta richiamano altre routine o procedure, comprese quelle dei file MATRIX.BAS, SOLSIST.BAS e ASTA.BAS descritte nei capitoli precedenti. Il programma principale e le relative routine sono contenuti nel file TELGEN.BAS. Le procedure sono divise in due gruppi: quelle comuni agli altri programmi presentati nel testo (telaio a maglie rettangolari e telaio spaziale) sono memorizzate nel file PROCOM.BAS; quelle caratteristiche del telaio generico ma riutilizzate anche per l'insieme spaziale di telai stanno nel file PROGEN.BAS.

A) Programma principale.

Nel programma principale possono individuarsi sostanzialmente tre blocchi di istruzioni.

Il primo contiene la definizione di tutte le procedure richiamate dal programma TELGEN, suddivise in base al file in cui sono contenute ed elencate per ciascuno di essi in ordine alfabetico.

Nel secondo viene effettuato il dimensionamento di quelle variabili con indice che hanno un numero di elementi fisso, definito una volta per tutte (*array statici*). La maggioranza delle variabili con indice ha invece un numero di elementi che dipende dallo schema geometrico ed è quindi variabile in funzione dei dati (*array dinamici*); il dimensionamento di queste è effettuato mediante apposite routine, richiamate man mano che vengono letti i dati.

Il terzo blocco, infine, contiene il richiamo a tutti i blocchi logici nei quali il procedimento di soluzione è stato scomposto. Si può così notare in maniera immediata l'organizzazione complessiva del programma, costituita in particolare da una prima fase (lettura e stampa dei dati geometrici, preparazione e triangolarizzazione della matrice di rigidezza della struttura) ed una seconda fase (lettura e stampa dei carichi, risoluzione del sistema e stampa dei risultati); quest'ultima è ripetuta ciclicamente, per consentire la risoluzione di più schemi di carico.

B) Dimensionamento degli array dinamici (routine Dim...).

Il gruppo di routine che effettuano il dimensionamento degli array dinamici sono poste immediatamente dopo il programma principale. È opportuno evidenziare che nel caso di programmi compilati il nome di ciascun array deve essere incontrato per la prima volta in una istruzione di dimensionamento; in caso contrario esso verrebbe considerato un array statico

avente il numero di elementi definito per default.

C) Schermata di presentazione del programma (routine `Presentazione`).

Visualizza sullo schermo il nome del programma e la relativa versione, con un sintetico rinvio al presente testo per la descrizione delle ipotesi di calcolo e delle modalità operative.

D) Definizione delle unità di ingresso e uscita (procedura `DefinisceIO`).

Consente di indicare in quale file sono contenuti i dati ed a quale unità inviare l'output. Per quest'ultima si può indicare il nome di un file, oppure lo schermo (`CON`, dall'inglese console) o la stampante (`PRN`, dall'inglese printer).

E) Definizione delle modalità operative (procedura `DefinisceMO`).

Consente innanzitutto di definire se il calcolo deve essere eseguito tenendo conto o trascurando la deformabilità tagliante delle aste.

Consente inoltre di indicare se si vuole effettuare un controllo dei dati man mano che vengono letti. Conviene utilizzare questa opzione quando il programma si blocca a causa di qualche errore nei dati che non si riesce ad individuare. Il controllo consiste nell'inviare all'unità di uscita una segnalazione per ciascun valore letto, nella quale il programma indica come questo valore è stato interpretato.

F) Lettura dei dati geometrici ed elastici (routine `LeggeGeomGen`).

Utilizza la routine `InputIndGen` e le procedure `InputIntestaz`, `InputNodiGen`, `InputTraversiGen`, `InputModElast`, `InputSezioni`, `InputAsteGen` per leggere da un file su disco le informazioni geometriche generali e quelle specifiche relative a nodi, traversi orizzontali, caratteristiche del materiale, tipi di sezione, aste. L'ordine con cui i dati vengono letti è descritto in dettaglio nel paragrafo 5. Se tra i valori letti ne viene trovato qualcuno inaccettabile, l'esecuzione del programma è interrotta al termine della fase di lettura. Vengono utilizzate anche le routine `DimGeomGen`, `DimTabSez` e `DimCarGen` per dimensionare gli array dinamici in funzione del numero di elementi strettamente necessario per la struttura da calcolare.

Questo blocco, come tutti i successivi, inizia e termina con la visualizzazione sullo schermo di informazioni che consentono all'utente di seguire l'evoluzione del calcolo rendendosi conto di quali fasi vengono man mano eseguite dal calcolatore. Sono utilizzate a tale scopo le routine `Completata`, `InCorso`, `NomeProgramma`.

G) Stampa dei dati geometrici ed elastici (routine `StampaGeomGen`).

Utilizza le procedure `OutputIntestaz`, `OutputNodiGen`, `OutputAsteGen`, `OutputModElast` per inviare all'unità di uscita un elenco di tutte le informazioni geometriche ed elastiche lette nel blocco precedente.

H) Disegno dello schema del telaio (routine `Schema`).

Utilizza la procedura `DisegnaSchema` per visualizzare sullo schermo lo schema del telaio. La modalità grafica è definita automaticamente dalla procedura `ModGraf`, che la individua per tentativi. Se il calcolatore è dotato di scheda Hercules, perché questa venga riconosciuta è necessario aver eseguito preliminarmente il programma `QBHERC.COM`, fornito dalla Microsoft insieme al Quick Basic. Se l'unità di uscita è la stampante, il disegno viene inviato ad essa utilizzando la procedura `LCOPY`, valida per scheda CGA e stampante compatibile IBM.

Una descrizione dello schema di base della procedura e delle istruzioni grafiche utilizzate è riportata nel capitolo 7 del testo "Il personal computer nel calcolo di edifici. Introduzione alla programmazione" citato nella bibliografia.

I) Individuazione delle incognite (routine `DefinisceIncognite`).

Utilizza la procedura `Incognite`, contenente a sua volta le routine `Numera` ed `Ampiezza` che rispettivamente definiscono le incognite e calcolano l'ampiezza della semibanda della matrice di rigidezza della struttura. Utilizza inoltre la routine `DimMatRig`, che in base ai valori ottenuti effettua il dimensionamento degli array in cui tale matrice è suddivisa e del vettore termini noti.

La routine `Numera` effettua la numerazione delle incognite, cioè assegna ad esse un numero progressivo ed attribuisce a ciascuna componente di movimento nodale il numero dell'incognita corrispondente, o il valore `0` se la componente è impedita da vincoli esterni. Queste informazioni vengono racchiuse nell'array bidimensionale `N()`, il cui primo indice individua il nodo ed il secondo è il numero d'ordine della componente.

Per la presenza di impalcati indeformabili, il procedimento può essere diviso in due fasi. Preliminarmente si inizializza l'array `N()` con un valore di default (`-1`) che indica convenzionalmente che la componente di movimento non è ancora stata presa in esame. Nella prima fase si numerano i gradi di libertà corrispondenti agli spostamenti orizzontali degli impalcati. Per ogni impalcato, si esaminano prima i vincoli dei nodi da esso collegati (se anche per uno solo di essi è impedita la traslazione, tutto il traverso è

bloccato). Successivamente si assegna alla variabile `N(I,1)`, per tutti i nodi `I` dell'impalcato, il valore `0` se lo spostamento in direzione x è impedito, o il primo valore disponibile per il numero d'ordine delle incognite se esso è consentito.

Nella seconda fase si procede nodo per nodo. Per ciascuno di essi si esamina prima la variabile `N()` per vedere se è già stata definita nella prima fase. Si esamina quindi il codice di vincolo; in corrispondenza a ciascuna componente di movimento si assegna il valore `0` se il relativo codice di vincolo è `1`, altrimenti si assegna il primo valore disponibile per il numero d'ordine delle incognite.

La routine `Ampiezza` passa in rassegna tutte le aste, per individuare la massima differenza di numero tra le incognite dei nodi collegati da ciascuna di esse, individuando così l'ampiezza della semibanda della matrice di rigidezza della struttura.

L) Preparazione della matrice di rigidezza della struttura (routine `MatRigStruttura`).

Utilizza ciclicamente le procedure `MatAstaGlobale` e `FormaMat` per calcolare la matrice di rigidezza di ogni asta nel riferimento globale e per determinare il contributo di essa alla matrice di rigidezza della struttura. Utilizza poi la procedura `VincoliCed` per valutare il contributo di eventuali vincoli cedevoli.

La procedura `MatAstaGlobale` opera in maniera differenziata a seconda del tipo di asta, richiamando le opportune procedure del file `ASTA.BAS` per determinare le caratteristiche geometriche (lunghezza, inclinazione rispetto all'orizzontale, matrice di rotazione $\mathbf{L}_d$) e la matrice di rigidezza fondamentale $\mathbf{k}_g$ dell'asta `J`. Si noti che, per limitare l'ingombro di memoria centrale e non ricorrere alla memoria di massa, le matrici relative alle aste non vengono memorizzate stabilmente. La stessa procedura viene quindi richiamata anche in altre parti del programma, ogni volta che occorre conoscere tali matrici.

La procedura `FormaMat` determina il contributo dell'asta `J` alla matrice di rigidezza globale della struttura. Prepara innanzitutto un vettore di corrispondenza dell'asta, unendo i vettori di corrispondenza dei due nodi di estremità. Passa poi in rassegna tutti gli elementi della matrice $\mathbf{k}_g$ dell'asta; se la componente di movimento e l'azione di estremità cui un elemento si riferisce corrispondono a due incognite p e q, esso viene aggiunto in posizione p, q nella matrice di rigidezza della struttura utilizzando la procedura `Inserimento`.

La procedura `VincoliCed` esamina ciclicamente ogni componente di movimento nodale; se ad essa corrisponde l'incognita p, la cedevolezza del vincolo viene aggiunta in posizione p, p nella matrice di rigidezza della struttura, utilizzando ancora la procedura `Inserimento`.

La procedura `Inserimento` aggiunge un valore `V` nella matrice di rigidezza della struttura, in una posizione definita mediante gli indici `P` e `Q`. Poiché la matrice, a causa della sua particolare conformazione a banda, è memorizzata negli array `KTT()`, `KTA()` e `KAA()`, questa operazione richiede un preliminare controllo degli indici per individuare l'array in cui inserire il valore. Sfruttando la simmetria della matrice non vengono memorizzati i valori posti al di sotto della sua diagonale principale.

M) Riduzione della matrice di rigidezza ad una matrice triangolare (routine `Triangolarizza`).

Il procedimento di riduzione è ampiamente illustrato nel capitolo 3 della seconda parte del testo. Tra le procedure ivi presentate, viene qui utilizzata `Triang.K2` perché la sottomatrice $\mathbf{K}^{aa}$ è stata memorizzata in un array bidimensionale.

N) Lettura dei carichi nodali e sulle aste (routine `LeggeCarGen`).

Utilizza la procedura `InputCarGen` per leggere dal file dati su disco le informazioni relativi ai carichi applicati ai nodi e alle aste della struttura. L'ordine con cui essi vengono letti è descritto in dettaglio nel paragrafo 5. Se un gruppo di carichi (ad esempio quelli sulle aste) non è presente, il programma provvede automaticamente ad azzerarne i valori.

O) Stampa dei dati di carico (routine `StampaCarGen`).

Utilizza la procedura `OutputCarGen` per inviare all'unità di uscita le informazioni relative ai carichi lette nel blocco precedente.

P) Calcolo del vettore carichi nodali (routine `TerminiNoti`).

Utilizza la procedura `TermNoto` che ottiene il vettore carichi nodali $\overline{\mathbf{R}}$ come somma delle azioni esterne sui nodi e delle azioni di incastro perfetto (nel riferimento globale). Il contributo delle prime è valutato passando in rassegna tutte le azioni nodali, per individuare a quale incognita ciascuna di esse corrisponde. Quello delle seconde è calcolato solo nel caso di aste effettivamente caricate, ricorrendo a procedure del file `ASTA.BAS` ed alla procedura `AzIncLocale`.

Quest'ultima determina innanzitutto le componenti del carico parallele

ed ortogonali all'asse dell'asta J. Determina quindi il vettore $\overline{S}_l$, azioni di incastro perfetto nel riferimento locale, operando in maniera differenziata a seconda del tipo di asta ed utilizzando in alternativa le procedure `Sl.Inc.Qunif` e `Sl.Wnk.Qunif`. Poiché i vettori così ottenuti non vengono memorizzati stabilmente, la stessa procedura viene richiamata anche in altre parti del programma.

Q) Calcolo del vettore soluzione (routine `Risolvi`).

Come illustrato nel capitolo 3, la soluzione viene ottenuta mediante sostituzione all'indietro utilizzando la procedura `Risolve.K2`.

R) Calcolo delle caratteristiche di sollecitazione agli estremi delle aste e delle azioni sui vincoli (routine `CaratSollec`).

Utilizza la procedura `CarSoll`. Per ogni asta determina innanzitutto la matrice $\mathbf{k}_g$ mediante la procedura `MatAstaGlobale`. Memorizza poi nel vettore `S1()` le componenti di movimento degli estremi, in base ai valori delle incognite ed ai vettori di corrispondenza nodali, e da queste risale alle corrispondenti azioni di estremità nel riferimento globale e locale. Calcola quindi mediante la procedura `AzIncLocale` le azioni di incastro perfetto, che sommate alle precedenti forniscono le azioni effettivamente trasmesse dai nodi all'asta. Sottraendo i valori di queste, espressi nel riferimento globale, al vettore carichi nodali si ottengono le azioni sui vincoli e l'eventuale squilibrio nei nodi, conseguente agli errori di arrotondamento commessi nel calcolo. I valori delle azioni nel riferimento locale rappresentano invece le caratteristiche della sollecitazione, a meno del cambio di segno necessario per adattare alcune di esse alle convenzioni della Scienza delle Costruzioni.

S) Stampa dei risultati (routine `StampaRisultati`).

Utilizza la procedura `OutputRisGen` per inviare all'unità di uscita i risultati della risoluzione del singolo schema di carico. Indica innanzitutto i valori delle componenti di movimento dei nodi e poi quelli delle caratteristiche della sollecitazione agli estremi delle aste. Riporta quindi i valori delle azioni trasmesse dalla struttura ai vincoli esterni e identifica la eventuale presenza di azioni squilibrate (valori delle azioni nodali `F()` non nulli in assenza di vincoli). In questo caso stampa una tabella che mette in risalto l'entità di tali squilibri. Indica infine i valori delle azioni mutue orizzontali esplicate sui nodi collegati da traversi inestensibili e ne effettua la somma per evidenziare eventuali squilibri. Indica infine i valori delle azioni mutue orizzontali esplicate sui nodi collegati da traversi inestensibili e ne effettua

la somma per evidenziare eventuali squilibri. Utilizza inoltre la routine `OutputIpotesiGen` per indicare le ipotesi di base del calcolo, evidenziando in particolare se esso è stato eseguito tenendo conto della deformabilità a taglio.

3. Elenco delle variabili

Le variabili sono riunite in gruppi in base al loro significato ed al blocco di programma nel quale sono utilizzate prevalentemente o per la prima volta. Nell'ambito di ciascun gruppo sono elencate in ordine alfabetico. Per tutte le grandezze dimensionali sono riportate, tra parentesi, le unità di misura adottate.

Variabili utilizzate nella gestione generale del programma

`Contr`	indica se i dati devono essere stampati man mano che vengono letti per consentirne un controllo immediato
`DefEst`	indica se il calcolo è condotto tenendo conto della deformabilità estensionale delle aste (questa variabile è inserita solo per compatibilità con i programmi `TELRET` e `TELSPA`)
`DefTag`	indica se il calcolo è condotto tenendo conto della deformabilità a taglio delle aste
`ER`	indica un errore nei dati letti
`Squil`	indica la presenza di squilibrio nei nodi dopo la risoluzione del sistema
`Vced`	indica la presenza di vincoli elasticamente cedevoli
`Wnkl`	indica la presenza di aste su suolo alla Winkler
	queste sette variabili di controllo possono assumere solo due valori convenzionali (0=no, 1=si)
`CAR`	numero d'ordine dello schema di carico in esame
`FI$`	nome del file che contiene i dati
`FO$`	nome dell'unità cui si vuole inviare l'output (`CON`=schermo, `PRN`=stampante, oppure nome di un file)
`FT$`	nome della directory da utilizzare per file temporanei (questa variabile è inserita solo per compatibilità con il programma `TELSPA`)

Variabili utilizzate come indici degli array

I	indica un nodo generico, oppure un generico elemento dell'array NI()
J	indica un'asta generica
K	indica un generico traverso orizzontale inestensibile, oppure il numero d'ordine della componente di movimento o azione nodale come secondo indice degli array F(), KV() ed N() e come posizione nella stringa V$()
L	indica un generico tipo di sezione

Variabili utilizzate nella definizione dello schema geometrico e delle caratteristiche elastiche

B(J)	larghezza della sottobase dell'asta J, se questa è un'asta su suolo elastico alla Winkler (m)
E	modulo di elasticità normale E (kN m^{-2})
E1(J)	numero d'ordine del nodo che costituisce il primo estremo dell'asta J
E2(J)	numero d'ordine del nodo che costituisce il secondo estremo dell'asta J
EL	modulo di elasticità normale E (N mm^{-2})
EV$	singolo carattere del codice di vincolo V$(I)
FS$(L)	codice per la definizione della forma della sezione; il carattere può assumere convenzionalmente solo uno di questi valori: R = sezione di forma rettangolare G = sezione di forma generica
G	modulo di elasticità tangenziale G (kN m^{-2})
GL	modulo di elasticità tangenziale G (N mm^{-2})
IT$	intestazione del calcolo
IZ	numero totale di nodi
JZ	numero totale di aste
K(J)	costante di sottofondo dell'asta J, se questa è un'asta su suolo elastico alla Winkler (N cm^{-3})
KV(I,K)	rigidezza del vincolo corrispondente alla componente di movimento K del nodo I, con K compreso tra 1 e 3 (kN m^{-1} per K=1 e 2, kNm per K=3)
KZ	numero totale di traversi orizzontali inestensibili

`LZ`	numero totale di tipi di sezione
`NI()`	array monodimensionale che contiene la numerazione dei nodi dei traversi orizzontali inestensibili (l'elenco dei nodi di ciascun traverso termina con un valore nullo)
`NT(J)`	tipo di sezione dell'asta J
`TA$(J)`	codice per la definizione del tipo di asta; il carattere può assumere convenzionalmente solo uno di questi valori: `T` = trave alla De Saint Venant solidale ai nodi di estremità `W` = trave su suolo elastico alla Winkler
`X(I)`	ascissa del nodo `I` nel sistema di riferimento globale (m)
`Y(I)`	ordinata del nodo `I` nel sistema di riferimento globale (m)
`V$(I)`	codice di vincolo del nodo `I` la stringa è costituita da tre caratteri, uno per ciascuna componente di movimento; il carattere può assumere convenzionalmente solo uno di questi valori: `0` = componente di movimento consentita `1` = componente di movimento impedita `C` = componente di movimento soggetta ad un vincolo elastico
`ZA(L)`	area della sezione L (m^2)
`ZAT(L)`	area a taglio, cioè rapporto tra area e fattore di taglio, della sezione L (m^2)
`ZB(L)`	base della sezione L (m)
`ZH(L)`	altezza della sezione L (m)
`ZI(L)`	momento d'inerzia della sezione L (m^4)

Variabili utilizzate nel disegnare lo schema del telaio

`FX, FY`	dimensioni reali dell'area grafica (m)
`LX, LY`	dimensioni in scala dell'area grafica
`SC`	fattore di scala
`Scheda`	modalità grafica, dipendente dal tipo di scheda grafica installata nel calcolatore
`X1, X2`	valore minimo e massimo delle ascisse dei punti da rappresentare; poi valore minimo e massimo delle ascisse logiche dell'area grafica
`XC`	ascissa del punto centrale della figura da rappresentare
`Y1, Y2`	valore minimo e massimo delle ordinate dei punti da rappresentare; poi valore minimo e massimo delle ordinate logiche

	dell'area grafica
YC	ordinata del punto centrale della figura da rappresentare

Variabili utilizzate nella numerazione delle incognite

I1	posizione nell'array NI() del primo nodo del generico traverso K, durante la fase di numerazione degli spostamenti orizzontali dei traversi
N	numero d'ordine dell'incognita generica
N(I,K)	elemento K del vettore di corrispondenza del nodo I, cioè numero d'ordine dell'incognita associata alla componente di movimento K del nodo I (un valore nullo indica che la componente di movimento è impedita da un vincolo)
N1	numero più piccolo tra quelli delle incognite corrispondenti alle componenti di movimento degli estremi dell'asta J ed appartenenti alla sottomatrice $\mathbf{K}^{aa}$
N2	numero più grande tra quelli delle incognite corrispondenti alle componenti di movimento degli estremi dell'asta J ed appartenenti alla sottomatrice $\mathbf{K}^{aa}$
NL	numero di incognite corrispondenti a traslazioni dei traversi
NZ	numero totale di incognite
SB	ampiezza della semibanda della matrice $\mathbf{K}^{aa}$
VN	numero di incognita da assegnare allo spostamento orizzontale di un traverso; viene azzerato se anche un solo nodo del traverso è bloccato alla traslazione in direzione x

Variabili utilizzate nella costruzione della matrice di rigidezza dell'asta o della struttura

A	area della sezione dell'asta in esame (m^2)
AT	area a taglio della sezione dell'asta in esame (m^2)
B	larghezza della sottobase dell'asta in esame, se è un'asta su suolo elastico alla Winkler (m)
C	indice che individua la generica colonna della matrice di rigidezza globale dell'asta J
CsAl	coseno dell'angolo di inclinazione dell'asta J rispetto all'asse x
E1	numero d'ordine del nodo che costituisce il primo estremo dell'asta in esame

E2	numero d'ordine del nodo che costituisce il secondo estremo dell'asta in esame
G()	vettore di corrispondenza per l'asta J, ottenuto come unione dei vettori di corrispondenza degli estremi dell'asta
Iner	momento d'inerzia della sezione dell'asta in esame (m^4)
K	costante di sottofondo dell'asta in esame, se è un'asta su suolo elastico alla Winkler (N cm^{-3})
KAA()	array bidimensionale in cui è memorizzata la semibanda superiore di $\mathbf{K}^{aa}$, cioè della parte della matrice di rigidezza che contiene i termini relativi alle incognite diverse dagli spostamenti dei traversi
Kg()	matrice di rigidezza dell'asta J nel riferimento globale
KTA()	array bidimensionale in cui è memorizzata $\mathbf{K}^{ta}$, cioè la parte della matrice di rigidezza che contiene la banda laterale superiore
KTT()	array bidimensionale in cui è memorizzata $\mathbf{K}^{tt}$, cioè la parte della matrice di rigidezza che contiene i termini relativi agli spostamenti dei traversi; sfruttando la simmetria di $\mathbf{K}^{tt}$, si sono utilizzati solo gli elementi del triangolo superiore
Lun	lunghezza dell'asta J (m)
Ld()	matrice di rotazione $\mathbf{L}_d$ dell'asta J
P	numero di incognita corrispondente alla componente di movimento R degli estremi dell'asta J, oppure alla componente di movimento K del nodo I
Q	numero di incognita corrispondente alla componente di movimento C degli estremi dell'asta J
R	indice che individua la generica riga della matrice di rigidezza globale dell'asta J
SnAl	seno dell'angolo di inclinazione dell'asta J rispetto all'asse x
TA$	codice che individua il tipo dell'asta in esame
X1, Y1	coordinate del primo estremo dell'asta in esame (m)
X2, Y2	coordinate del secondo estremo dell'asta in esame (m)

Variabili utilizzate nella definizione dei carichi

F(I,K)	azione concentrata K applicata al nodo I; K varia da 1 a 3 per indicare nell'ordine le azioni F_x, F_y e M_z (kN per K=1 e 2, kNm per K=3)

`Q`	indica il numero d'ordine di un carico generico
`QA`	numero di aste sulle quali sono applicati carichi uniformemente distribuiti
`QN`	numero di nodi ai quali sono direttamente applicati carichi concentrati
`P(J,K)`	componente K del carico uniformemente distribuito sull'asta J; K varia in questo caso da 1 a 2 per indicare nell'ordine le componenti p_x e p_y (kN m^{-1})

Variabili utilizzate nella preparazione del vettore termini noti e nella determinazione e stampa di spostamenti e caratteristiche della sollecitazione

`F`	variabile ausiliaria usata durante la stampa dei risultati per calcolare la somma delle azioni orizzontali sui nodi di un traverso ed individuare così la presenza di un eventuale squilibrio (kN o kNm)
`F()`	dopo la risoluzione del sistema viene memorizzata in questo array, già descritto in precedenza, la somma di azioni nodali esterne ed azioni trasmesse ai nodi dalle aste, pari quindi alle azioni sui vincoli esterni e mutui ed all'eventuale squilibrio nodale conseguente agli errori di troncamento
`K1()`	matrice di rigidezza dell'asta J nel riferimento locale
`LdT()`	trasposta della matrice di rotazione dell'asta J
`M1(J)`	momento flettente al primo estremo dell'asta J (kNm)
`M2(J)`	momento flettente al secondo estremo dell'asta J (kNm)
`N1(J)`	sforzo normale al primo estremo dell'asta J (kN)
`N2(J)`	sforzo normale al secondo estremo dell'asta J (kN)
`NN`	variabile ausiliaria usata durante la stampa dei risultati per indicare il numero massimo di nodi di un traverso
`PN`	componente del carico uniformemente distribuito parallela all'asse dell'asta J (kN m^{-1})
`PQ`	componente del carico uniformemente distribuito ortogonale all'asse dell'asta J (kN m^{-1})
`R()`	vettore azioni nodali $\overline{\mathbf{R}}$, somma delle azioni applicate direttamente sui nodi e delle azioni di incastro perfetto; dopo la risoluzione del sistema contiene il vettore spostamenti nodali $\mathbf{r}$
`S()`	vettore azioni di incastro perfetto dell'asta J, valutate nel riferimento locale

`S1()`	componenti di movimento degli estremi dell'asta J
`Sg()`	vettore azioni agli estremi dell'asta J, di incastro perfetto o conseguenti alle componenti di movimento di estremità, valutate nel riferimento globale
`Sl()`	vettore azioni agli estremi dell'asta J, valutate nel riferimento locale
`T1(J)`	taglio al primo estremo dell'asta J (kN)
`T2(J)`	taglio al secondo estremo dell'asta J (kN)
`V`	variabile ausiliaria usata durante la stampa dei risultati per indicare quante componenti di movimento del nodo in esame sono impedite da un vincolo esterno, oppure sono libere ma corrispondono ad un'azione squilibrata; è usata anche per contare i nodi di un traverso
`V(K)`	variabile ausiliaria usata durante la stampa dei risultati per memorizzare il valore della componente di movimento K relativa a un nodo generico, oppure per indicare se tale componente è impedita da un vincolo esterno, o infine se è libera ma ad essa corrisponde un'azione squilibrata

4. Codifica

```
'                                  TELGEN.BAS
'
'  programma per il calcolo di strutture piane intelaiate di forma qualsiasi
'
'  tipi di aste:          T  =  trave alla De Saint Venant, incastrata ai nodi
'                         W  =  trave su suolo elastico alla Winkler
'
'  vincoli esterni:       vincoli rigidi o elasticamente cedevoli
'                         vincoli mutui allo spostamento orizzontale
'
'  carichi:               azioni concentrate sui nodi
'                         carichi uniformemente distribuiti sulle aste
'
'                                                              rev. 01.90

' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file SOLSIST.BAS
DECLARE SUB Triang.K2 (KTT!(), KTA!(), KAA!(), SB!, RID!)
DECLARE SUB Risolve.K2 (KTT!(), KTA!(), KAA!(), SB!, RID!, B!())

' procedure del file PROCOM.BAS
DECLARE SUB Aspetta (Sec!)
```

```
DECLARE SUB CarSoll (JZ!, X!(), Y!(), ZA!(), ZAT!(), ZI!(), E!, G!, E1!(), E2!(),
                     TA$(), NT!(), K!(), B!(), N!(), P!(), F!(), R!(), N1!(), T1!(),
                     M1!(), N2!(), T2!(), M2!())
DECLARE SUB DefinisceIO (Prog$, FI$, FO$, FT$)
DECLARE SUB DefinisceMD (Prog$, DefEst!, DefTag!, Contr!)
DECLARE SUB DisegnaSchema (IZ!, JZ!, X!(), Y!(), E1!(), E2!())
DECLARE SUB FormaMat (E1!, E2!, N!(), Kg!(), NL!, SB!, KTT!(), KTA!(), KAA!())
DECLARE SUB Incognite (IZ!, JZ!, KZ!, E1!(), E2!(), V$(), NI!(), N!(), NL!, NZ!,SB!)
DECLARE SUB InputIntestaz (Contr!, Intest$)
DECLARE SUB InputModElast (Contr!, EL!, GL!, E!, G!)
DECLARE SUB InputSezioni (Contr!, LZ!, FS$(), ZB!(), ZH!(), ZA!(), ZI!(),ZAT!(),ER!)
DECLARE SUB MatAstaGlobale (TA$, X1!, X2!, Y1!, Y2!, A!, ZAT!, Iner!, K!, B!, E!,
                            G!, Lun!, SnAl!, CsAl!, Ld!(), Kg!())
DECLARE SUB OutputIntestaz (FI$, Intest$)
DECLARE SUB OutputModElast (DefTag!, EL!, GL!)
DECLARE SUB TermNoto (IZ!, JZ!, X!(), Y!(), ZA!(), ZI!(), E!, E1!(), E2!(), TA$(),
                      NT!(), K!(), B!(), N!(), P!(), F!(), R!())
DECLARE SUB VincoliCed (IZ!, N!(), KV!(), NL!, SB!, KTT!(), KTA!(), KAA!())

' procedure del file PROGEN.BAS
DECLARE SUB InputAsteGen (Contr!, JZ!, LZ!, E1!(), E2!(), NT!(), TA$(), Wnkl!, B!(),
                          K!(), ER!)
DECLARE SUB InputCarGen (Contr!, IZ!, JZ!, QN!, QA!, F!(), P!(), ER!)
DECLARE SUB InputNodiGen (Contr!, IZ!, X!(), Y!(), V$(), VCed!, KV!(), ER!)
DECLARE SUB InputTraversiGen (Contr!, KZ!, NI!())
DECLARE SUB OutputAsteGen (DefTag!, Wnkl!, JZ!, NT!(), TA$(), E1!(), E2!(), FS$(),
                           ZB!(), ZH!(), ZA!(), ZAT!(), ZI!(), B!(), K!())
DECLARE SUB OutputCarGen (CAR!, IZ!, JZ!, QN!, QA!, F!(), P!())
DECLARE SUB OutputNodiGen (IZ!, X!(), Y!(), V$(), VCed!, KV!())
DECLARE SUB OutputRisGen (IZ!, JZ!, KZ!, NL!, V$(), NI!(), N!(), R!(), M1!(), M2!(),
                          T1!(), T2!(), N1!(), N2!(), F!())


' ------------------ Dimensionamento delle variabili con indice --------------------

OPTION BASE 1             ' definisce l'estremo inferiore dell'indice degli array
DIM Ld(6, 6), Kg(6, 6)    ' array statici
                          ' array dinamici: sono dimensionati in apposite routine


' ---------------------------- Programma principale --------------------------------

GOSUB Presentazione
CALL DefinisceIO("GEN", FI$, FO$, FT$)
CALL DefinisceMD("GEN", DefEst, DefTag, Contr)
OPEN "I", #1, FI$
OPEN "O", #2, FO$
GOSUB LeggeGeomGen
GOSUB StampaGeomGen
GOSUB Schema
GOSUB DefinisceIncognite
GOSUB MatRigStruttura
GOSUB Triangolarizza
CAR = 0
DO WHILE NOT EOF(1)
   CAR = CAR + 1
   GOSUB LeggeCarGen
   GOSUB StampaCarGen
   GOSUB TerminiNoti
```

```
   GOSUB Risolvi
   GOSUB CaratSollec
   GOSUB StampaRisultati
LOOP
CLOSE #1
CLOSE #2
PRINT
PRINT "elaborazione terminata"
PRINT
END


' ------------------------------ DimXXXXXX ------------------------------------
'  Gruppo di routine per il dimensionamento degli array dinamici
' -----------------------------------------------------------------------------

' ................................ DimCarGen ....................................
'  Routine per il dimensionamento degli array relativi a carichi e risultati
'
DimCarGen:
   DIM F(IZ, 3), P(JZ, 2)
   DIM M1(JZ), M2(JZ), T1(JZ), T2(JZ), N1(JZ), N2(JZ)
RETURN

' ................................ DimGeomGen ...................................
'  Routine per il dimensionamento degli array relativi alla geometria
'
DimGeomGen:
   DIM X(IZ), Y(IZ), V$(IZ), KV(IZ, 3), NI(IZ + KZ)
   DIM E1(JZ), E2(JZ), TA$(JZ), NT(JZ), B(JZ), K(JZ)
   DIM N(IZ, 3)
RETURN

' ................................ DimMatRig ....................................
'  Routine per il dimensionamento di matrice di rigidezza e vettore carico
'
DimMatRig:
   IF NL = 0 THEN
      REDIM KTT(0 TO 0, 0 TO 0), KTA(0 TO 0, NZ - NL)
   ELSE
      REDIM KTT(NL, NL), KTA(NL, NZ - NL)
   END IF
   DIM KAA(NZ - NL, SB), R(NZ)
RETURN

' ................................ DimTabSez ....................................
'  Routine per il dimensionamento degli array relativi ai tipi di sezione
'
DimTabSez:
   DIM FS$(LZ), ZB(LZ), ZH(LZ), ZA(LZ), ZI(LZ), ZAT(LZ)
RETURN


' ------------------------------ Presentazione --------------------------------
'  Routine per la visualizzazione della schermata di presentazione del
'  programma
' -----------------------------------------------------------------------------
Presentazione:
   CLS
```

```
   PRINT TAB(31); "TELGEN - rev. 01.90"
   PRINT
   PRINT "       Le ipotesi di calcolo e le modalita' operative del programma"
   PRINT "       TELGEN sono riportate nella terza parte, capitolo 5, del libro:"
   PRINT "       A.Ghersi, R.Coraggio, Il personal computer nel calcolo di edifici."
   PRINT "       Analisi matriciale di strutture intelaiate, CUEN, Napoli, 1990."
   PRINT
   PRINT
   PRINT
RETURN


' ------------------------------- Completata -----------------------------------
'  Routine per l'indicazione della fine di una fase
' ------------------------------------------------------------------------------
Completata:
   LOCATE CSRLIN - 1, 44
   PRINT "completata            "
RETURN

.
' ------------------------------- InCorso --------------------------------------
'  Routine per l'indicazione dell'inizio di una fase
' ------------------------------------------------------------------------------
InCorso:
   LOCATE CSRLIN - 1, 44
   PRINT "in corso"
RETURN


' ------------------------------- NomeProgramma --------------------------------
'  Routine per la visualizzazione dell'intestazione del programma
' ------------------------------------------------------------------------------
NomeProgramma:
   CLS
   PRINT "TELGEN - rev. 01.90"
   PRINT
RETURN


' ------------------------------- LeggeGeomGen ---------------------------------
'  Routine per la lettura dei dati geometrici ed elastici
' ------------------------------------------------------------------------------
LeggeGeomGen:
   GOSUB NomeProgramma
   PRINT "lettura dati geometrici": GOSUB InCorso

   ER = 0
   CALL InputIntestaz(Contr, IT$)
   GOSUB InputIndGen
   GOSUB DimGeomGen
   GOSUB DimTabSez
   GOSUB DimCarGen
   CALL InputNodiGen(Contr, IZ, X(), Y(), V$(), VCed, KV(), ER)
   CALL InputTraversiGen(Contr, KZ, NI())
   CALL InputModElast(Contr, EL, GL, E, G)
   CALL InputSezioni(Contr, LZ, FS$(), ZB(), ZH(), ZA(), ZI(), ZAT(), ER)
   CALL InputAsteGen(Contr, JZ, LZ, E1(), E2(), NT(), TA$(), Wnkl, B(), K(), ER)
   IF ER = 1 THEN END
```

```
   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN

' ..... InputIndGen .....
' Routine per la lettura delle indicazioni generali
'
InputIndGen:
   INPUT #1, IZ, JZ, KZ, LZ
   IF Contr = 1 THEN
      PRINT #2, "numero di nodi:"; IZ
      PRINT #2, "numero di aste:"; JZ
      PRINT #2, "numero di traversi orizzontali inestensibili:"; KZ
      PRINT #2, "numero di tipi di sezione:"; LZ
   END IF
RETURN


' ---------- StampaGeomGen ----------
' Routine per la stampa di dati geometrici ed elastici
' -----------------------------------------------------------------------------
StampaGeomGen:
   PRINT "stampa dati geometrici"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   CALL OutputIntestaz(FI$, IT$)
   CALL OutputNodiGen(IZ, X(), Y(), V$(), VCed, KV())
   CALL OutputAsteGen(DefTag, Wnkl, JZ, NT(), TA$(), E1(), E2(), FS$(), ZB(), ZH(),
                      ZA(), ZAT(), ZI(), B(), K())
   CALL OutputModElast(DefTag, EL, GL)

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN


' ---------- Schema ----------
' Routine che visualizza lo schema geometrico
' -----------------------------------------------------------------------------
Schema:
   CALL DisegnaSchema(IZ, JZ, X(), Y(), E1(), E2())
   GOSUB NomeProgramma
RETURN


' ---------- DefinisceIncognite ----------
' Routine per la numerazione delle incognite  e la valutazione della semibanda
' (gli array che costituiscono la matrice di rigidezza vengono dimensionati)
' -----------------------------------------------------------------------------
DefinisceIncognite:
   PRINT "numerazione incognite": GOSUB InCorso

   CALL Incognite(IZ, JZ, KZ, E1(), E2(), V$(), NI(), N(), NL, NZ, SB)
   GOSUB DimMatRig

   GOSUB Completata
```

```
   PRINT "- numero totale di incognite"; NZ;
   IF NL <> 0 THEN PRINT " (tra cui"; NL; "spostamenti di traversi)";
   PRINT
   PRINT "- ampiezza della semibanda  "; SB
RETURN


' ------------------------------- MatRigStruttura -------------------------------
'  Routine per la costruzione della matrice di rigidezza della struttura
' -------------------------------------------------------------------------------
MatRigStruttura:
   PRINT "costruzione matrice rigidezza": GOSUB InCorso

   FOR J = 1 TO JZ
      L = NT(J)
      CALL MatAstaGlobale(TA$(J), X(E1(J)), X(E2(J)), Y(E1(J)), Y(E2(J)), ZA(L),
                          ZAT(L), ZI(L), K(J) * 1000, B(J), E, G, Lun, SnAl, CsAl,
                          Ld(), Kg())
      CALL FormaMat(E1(J), E2(J), N(), Kg(), NL, SB, KTT(), KTA(), KAA())
      LOCATE CSRLIN - 1, 44
      PRINT USING "eseguita per il ### %"; J / JZ * 100
   NEXT J
   IF VCed = 1 THEN CALL VincoliCed(IZ, N(), KV(), NL, SB, KTT(), KTA(), KAA())

   GOSUB Completata
RETURN


' ------------------------------- Triangolarizza -------------------------------
'  Routine che triangolarizza la matrice di rigidezza della struttura
' -------------------------------------------------------------------------------
Triangolarizza:
   PRINT "triangolarizz. matrice rigidezza": GOSUB InCorso

   CALL Triang.K2(KTT(), KTA(), KAA(), SB, 0)

   GOSUB Completata
RETURN

' ------------------------------- LeggeCarGen -------------------------------
'  Routine per la lettura dei carichi nodali e sulle aste
' -------------------------------------------------------------------------------
LeggeCarGen:
   PRINT
   PRINT "    Schema di carico n."; CAR
   PRINT "lettura dati di carico": GOSUB InCorso

   ER = 0
   CALL InputCarGen(Contr, IZ, JZ, QN, QA, F(), P(), ER)
   IF ER = 1 THEN END

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN


' ------------------------------- StampaCarGen -------------------------------
'  Routine per la stampa dei carichi nodali e sulle aste
' -------------------------------------------------------------------------------
StampaCarGen:
```

```
   PRINT "stampa dati di carico"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "SCHEMA DI CARICO N."; CAR: PRINT #2,
   CALL OutputCarGen(CAR, IZ, JZ, QN, QA, F(), P())

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN


' ------------------------------- TerminiNoti ------------------------------------
'  Routine per il calcolo dei carichi nodali
' --------------------------------------------------------------------------------
TerminiNoti:
   PRINT "preparazione termini noti": GOSUB InCorso

   CALL TermNoto(IZ, JZ, X(), Y(), ZA(), ZI(), E, E1(), E2(), TA$(), NT(), K(), B(),
                 N(), P(), F(), R())

   GOSUB Completata
RETURN


' ------------------------------- Risolvi ----------------------------------------
'  Routine per la risoluzione mediante sostituzione all'indietro
' --------------------------------------------------------------------------------
Risolvi:
   PRINT "risoluzione": GOSUB InCorso

   CALL Risolve.K2(KTT(), KTA(), KAA(), SB, 0, R())

   GOSUB Completata
RETURN


' ------------------------------- CaratSollec ------------------------------------
'  Routine per il calcolo delle azioni sulle aste
' --------------------------------------------------------------------------------
CaratSollec:
   PRINT "calcolo caratt. sollecitazione": GOSUB InCorso

   CALL CarSoll(JZ, X(), Y(), ZA(), ZAT(), ZI(), E, G, E1(), E2(), TA$(), NT(), K(),
                B(), N(), P(), F(), R(), N1(), T1(), M1(), N2(), T2(), M2())

   GOSUB Completata
RETURN


' ------------------------------- StampaRisultati --------------------------------
'  Routine per la stampa dei risultati
' --------------------------------------------------------------------------------
StampaRisultati:
   PRINT "stampa risultati"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso
```

```
   CALL OutputRisGen(IZ, JZ, KZ, NL, V$(), NI(), N(), R(), M1(), M2(), T1(), T2(),
                     N1(), N2(), F())
   GOSUB OutputIpotesiGen

   IF FO$ = "CON" THEN
      LOCATE 24, 1
   ELSE
      GOSUB Completata
   END IF
RETURN

' ................................ OutputIpotesiGen ..............................
' Routine per la stampa delle note sulle ipotesi di calcolo
'
OutputIpotesiGen:
   PRINT #2, : PRINT #2,
   PRINT #2, "Il calcolo e' stato eseguito :"
   PRINT #2, "- tenendo conto della deformazione estensionale delle aste"
   IF DefTag = 1 THEN
      PRINT #2, "- tenendo conto della deformazione a taglio delle aste"
      IF Wnkl = 1 THEN
         PRINT #2, "- trascurando la deformazione a taglio delle aste su suolo alla
                    Winkler"
      END IF
   ELSE
      PRINT #2, "- trascurando la deformazione a taglio delle aste"
   END IF
   PRINT #2, : PRINT #2,
RETURN



'                                    PROCOM.BAS
'
' procedure comuni ai programmi TELGEN, TELRET, TELSPA
'
'                                                                  rev. 01.90


' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file ASTA.BAS
DECLARE SUB Asta.Inc (X1!, X2!, Y1!, Y2!, E!, A!, I!, L!, SnAl!, CsAl!, Ld!(),Kg!())
DECLARE SUB Geometria (X1!, X2!, Y1!, Y2!, L!, SnAl!, CsAl!, Ld!())
DECLARE SUB Kg.IncTag (E!, G!, CHI!, A!, I!, L!, Ld!(), Kg!())
DECLARE SUB Kg.Wnk (K!, B!, E!, A!, I!, L!, Ld!(), Kg!())
DECLARE SUB Sl.Inc.Qunif (Q!, N!, L!, Sl!())
DECLARE SUB Sl.Wnk.Qunif (Q!, N!, L!, K!, B!, Kloc!(), Sl!())

' procedure del file MATRIX.BAS
DECLARE SUB AzzeraMat (A!())
DECLARE SUB ProdottoMat (A!(), B!(), C!())
DECLARE SUB SommaMat (A!(), B!(), C!())
DECLARE SUB TrasponeMat (A!(), B!())

' procedure di questo stesso file
DECLARE SUB Aspetta (Sec!)
```

```
DECLARE SUB AzIncLocale (PX!, PY!, TA$, B!, K!, Lun!, SnAl!, CsAl!, Ld!(), LdT!(),
                         Kg!(), S!())
DECLARE SUB Inserimento (P!, Q!, V!, NL!, SB!, KTT!(), KTA!(), KAA!())
DECLARE SUB LCOPY (SCR!, DENS!)
DECLARE SUB MatAstaGlobale (TA$, X1!, X2!, Y1!, Y2!, A!, AT!, Iner!, K!, B!, E!, G!,
                            Lun!, SnAl!, CsAl!, Ld!(), Kg!())
DECLARE SUB ModGraf (Scheda!)
DECLARE SUB Scala (X1!, Y1!, X2!, Y2!)


DIM SHARED Scheda1

ModGrafErr:                 '  individuazione della modalita' grafica
SELECT CASE Scheda1
   CASE 12
      Scheda1 = 11          ' MCGA
   CASE 11
      Scheda1 = 8           ' EGA 64k
   CASE 8
      Scheda1 = 2           ' CGA
   CASE 2
      Scheda1 = 3           ' Hercules
   CASE ELSE
      Scheda1 = 0           ' scheda grafica non presente o non riconosciuta
END SELECT
RESUME

' ============================== Aspetta ==============================
'   Procedura che lascia il programma in attesa per un assegnato tempo
'   (in secondi)
'   N.B. Esce comunque se scatta la mezzanotte
' ---------------------------------------------------------------------
SUB Aspetta (Sec)

    T = TIMER
    DO UNTIL (TIMER - T > Sec) OR (TIMER - T < 0)
    LOOP

END SUB

' ============================== AzIncLocale ==============================
'   Procedura che determina il vettore di incastro perfetto S()
'   di un'asta nel sistema di riferimento locale
'   NOTA: per asta tipo W occorre avere gia' definito Ld(), LdT() e Kg()
' -------------------------------------------------------------------------
SUB AzIncLocale (PX, PY, TA$, B, K, Lun, SnAl, CsAl, Ld(), LdT(), Kg(), S())

    DIM K1(6, 6)

    PN = PX * CsAl + PY * SnAl
    PQ = PY * CsAl - PX * SnAl
    SELECT CASE TA$
       CASE "T"
          CALL S1.Inc.Qunif(PQ, PN, Lun, S())
       CASE "W"
          CALL ProdottoMat(LdT(), Kg(), K1())
          CALL ProdottoMat(K1(), Ld(), K1())
          CALL S1.Wnk.Qunif(PQ, PN, Lun, K, B, K1(), S())
    END SELECT
```

```
END SUB

' ═══════════════════════════ CarSoll ═══════════════════════════
'   Procedura per la valutazione delle caratteristiche di sollecitazione
'   nelle aste
'
'   N.B. Nel vettore F() vengono aggiunti ai carichi nodali esterni
'        le azioni trasmesse ai nodi dalle aste
' ---------------------------------------------------------------
SUB CarSoll (JZ, X(), Y(), ZA(), ZAT(), ZI(), E, G, E1(), E2(), TA$(), NT(), K(),
             B(), N(), P(), F(), R(), N1(), T1(), M1(), N2(), T2(), M2())

    DIM Ld(6, 6), LdT(6, 6), Kg(6, 6), S(6, 1), S1(6, 1), Sg(6, 1), S1(6, 1)

    FOR J = 1 TO JZ
       L = NT(J)
       CALL MatAstaGlobale(TA$(J), X(E1(J)), X(E2(J)), Y(E1(J)), Y(E2(J)), ZA(L),
                           ZAT(L), ZI(L), K(J) * 1000, B(J), E, G, Lun, SnAl, CsAl,
                           Ld(), Kg())
       FOR K = 1 TO 6
          IF K <= 3 THEN N = N(E1(J), K) ELSE N = N(E2(J), K - 3)
          IF N = 0 THEN S1(K, 1) = 0 ELSE S1(K, 1) = R(N)
       NEXT K
       CALL ProdottoMat(Kg(), S1(), Sg())
       CALL TrasponeMat(Ld(), LdT())
       CALL ProdottoMat(LdT(), Sg(), S1())
       IF P(J, 1) <> 0 OR P(J, 2) <> 0 THEN
          CALL AzIncLocale(P(J, 1), P(J, 2), TA$(J), B(J), K(J) * 1000, Lun, SnAl,
                           CsAl, Ld(), LdT(), Kg(), S())
          CALL SommaMat(S(), S1(), S1())
       END IF
       CALL ProdottoMat(Ld(), S1(), Sg())
       FOR K = 1 TO 3
          F(E1(J), K) = F(E1(J), K) - Sg(K, 1)
          F(E2(J), K) = F(E2(J), K) - Sg(K + 3, 1)
       NEXT K
       N1(J) = -S1(1, 1)
       T1(J) = S1(2, 1)
       M1(J) = -S1(3, 1)
       N2(J) = S1(4, 1)
       T2(J) = -S1(5, 1)
       M2(J) = S1(6, 1)
       '
       LOCATE CSRLIN - 1, 44
       PRINT USING "eseguito per il ### %"; J / JZ * 100
    NEXT J

END SUB

' ═══════════════════════════ DefinisceIO ═══════════════════════
'   Procedura per la definizione delle unita' di ingresso ed uscita
' ---------------------------------------------------------------
SUB DefinisceIO (Prog$, FI$, FO$, FT$)

    INPUT "       Nome del file che contiene i dati:        ", FI$
    PRINT
    PRINT "       Unita' di uscita:"
    PRINT "       (CON = schermo, PRN = stampante, oppure nome di un file)"
```

```
    LOCATE CSRLIN - 2, 50
    INPUT "", FO$
    PRINT
    PRINT
    IF Prog$ = "SPA" THEN
       INPUT "       Drive e directory per file temporanei:    ", FT$
       PRINT
    END IF
    FI$ = UCASE$(FI$)
    FO$ = UCASE$(FO$)
    FT$ = UCASE$(FT$)
    IF FO$ = "" THEN FO$ = "CON"
    IF FT$ <> "" AND RIGHT$(FT$, 1) <> "\" THEN FT$ = FT$ + "\"

END SUB

' ============================== DefinisceMO ==============================
'   Procedura per la definizione delle modalita' operative:
'      - trascurabilita' o meno della deformabilita' estensionale
'      - trascurabilita' o meno della deformabilita' a taglio
'      - stampa dei dati man mano che vengono letti
' -------------------------------------------------------------------------
SUB DefinisceMO (Prog$, DefEst, DefTag, Contr)

 IF Prog$ = "RET" OR Prog$ = "SPA" THEN
    INPUT "       Deformabilita' estensionale dei pilastri (0=no, 1=si): ",DefEst
    PRINT
 ELSE
    DefEst = 1
 END IF
 INPUT "       Deformabilita' a taglio delle aste (0=no, 1=si):       ", DefTag
 PRINT
 INPUT "       Controllo dei dati man mano che vengono letti (0=no, 1=si):",Contr

END SUB

' ============================== DisegnaSchema ==============================
'   Procedura per il disegno dello schema del telaio
'      (vedi: A.Ghersi, Introduzione alla programmazione, CUEN, Napoli, 1987)
' -------------------------------------------------------------------------
SUB DisegnaSchema (IZ, JZ, X(), Y(), E1(), E2())

    CALL ModGraf(Scheda)
    IF Scheda <> 0 THEN
       ' adatta la scheda per consentire la copia su stampante
       IF Scheda > 3 AND FO$ = "PRN" THEN
          Scheda = 2
          SCREEN Scheda
       END IF
       ' determina i valori limite di X ed Y
       X1 = X(1)
       X2 = X(1)
       Y1 = Y(1)
       Y2 = Y(1)
       FOR I = 2 TO IZ
          IF X1 > X(I) THEN X1 = X(I)
          IF X2 < X(I) THEN X2 = X(I)
          IF Y1 > Y(I) THEN Y1 = Y(I)
          IF Y2 < Y(I) THEN Y2 = Y(I)
```

```
      NEXT I
      ' individua la scala del disegno
      CALL Scala(X1, Y1, X2, Y2)
      WINDOW (X1, Y1)-(X2, Y2)
      ' disegna lo schema
      FOR J = 1 TO JZ
         LINE (X(E1(J)), Y(E1(J)))-(X(E2(J)), Y(E2(J)))
      NEXT J
      ' effettua la copia su stampante
      IF FO$ = "PRN" AND Scheda = 2 THEN
         CALL LCOPY(Scheda, 1)
      ELSE
         CALL Aspetta(2)
      END IF
      ' torna alla modalita' testo
      SCREEN 0
   END IF

END SUB

' ========================== FormaMat ==========================
'   Procedura per la valutazione del contributo di un'asta alla matrice di
'   rigidezza della struttura
' ------------------------------------------------------------------------------
SUB FormaMat (E1, E2, N(), Kg(), NL, SB, KTT(), KTA(), KAA())

   DIM G(6)

   ' prepara il vettore di corrispondenza per l'asta
   FOR K = 1 TO 3
      G(K) = N(E1, K)
      G(K + 3) = N(E2, K)
   NEXT K
   ' inserisce i termini nella matrice di rigidezza della struttura
   FOR R = 1 TO 6
      P = G(R)
      IF P <> 0 THEN
         FOR C = 1 TO 6
            Q = G(C)
            IF Q <> 0 THEN
               CALL Inserimento(P, Q, Kg(R, C), NL, SB, KTT(), KTA(), KAA())
            END IF
         NEXT C
      END IF
   NEXT R

END SUB

' ========================== Incognite ==========================
'   Procedura per la numerazione delle incognite, valutazione dell'ampiezza
'   della semibanda, ridimensionamento della matrice di rigidezza
' ------------------------------------------------------------------------------
SUB Incognite (IZ, JZ, KZ, E1(), E2(), V$(), NI(), N(), NL, NZ, SB)

   GOSUB Numera
   GOSUB Ampiezza
   EXIT SUB

' numerazione automatica delle incognite
```

```
Numera:
    ' inizializzazione N(,)
    FOR I = 1 TO IZ
       FOR K = 1 TO 3
          N(I, K) = -1
       NEXT K
    NEXT I
    ' prima fase - spostamenti impalcati
    N = 0
    IF KZ <> 0 THEN
       I1 = 1
       FOR K = 1 TO KZ
          VN = N + 1
          I = I1
          DO WHILE NI(I) <> 0
             IF MID$(V$(NI(I)), 1, 1) = "1" THEN VN = 0
             I = I + 1
          LOOP
          I = I1
          DO WHILE NI(I) <> 0
             N(NI(I), 1) = VN
             I = I + 1
          LOOP
          IF VN <> 0 THEN N = N + 1
          I1 = I + 1
       NEXT K
    END IF
    NL = N
    ' seconda fase - altre incognite
    FOR I = 1 TO IZ
       FOR K = 1 TO 3
          IF N(I, K) = -1 THEN
             IF MID$(V$(I), K, 1) = "1" THEN
                N(I, K) = 0
             ELSE
                N = N + 1
                N(I, K) = N
             END IF
          END IF
       NEXT K
    NEXT I
    NZ = N
RETURN

' calcolo dell'ampiezza della semibanda
Ampiezza:

    SB = 0
    FOR J = 1 TO JZ
       N1 = NZ
       N2 = NL
       FOR K = 1 TO 3
          N = N(E1(J), K)
          IF N > NL THEN
             IF N1 > N THEN N1 = N
             IF N2 < N THEN N2 = N
          END IF
          N = N(E2(J), K)
          IF N > NL THEN
```

```
            IF N1 > N THEN N1 = N
            IF N2 < N THEN N2 = N
         END IF
      NEXT K
      IF N2 > N1 AND SB < N2 - N1 THEN SB = N2 - N1
   NEXT J
   SB = SB + 1
RETURN

END SUB

' ============================== InputIntestaz ==============================
'   Procedura per la lettura della intestazione
' ---------------------------------------------------------------------------
SUB InputIntestaz (Contr, Intest$)

   INPUT #1, Intest$
   IF Contr = 1 THEN
      PRINT #2, "intestazione: "; CHR$(34); Intest$; CHR$(34)
   END IF

END SUB

' ============================== InputModElast ==============================
'   Procedura per la lettura del modulo di elasticita' normale ed il calcolo
'   del modulo di elasticita' tangenziale
'
'   N.B. si assume G = 0.45xE, con riferimento al calcestruzzo
' ---------------------------------------------------------------------------
SUB InputModElast (Contr, EL, GL, E, G)

   INPUT #1, EL
   IF Contr = 1 THEN PRINT #2, "modulo elastico E ="; EL
   GL = .45 * EL
   E = EL * 1000
   G = GL * 1000

END SUB

' ============================== InputSezioni ==============================
'   Procedura per la lettura dei dati delle sezioni tipo
' ---------------------------------------------------------------------------
SUB InputSezioni (Contr, LZ, FS$(), ZB(), ZH(), ZA(), ZI(), ZAT(), ER)

   ' sezioni tipo
   FOR L = 1 TO LZ
      INPUT #1, FS$(L)
      FS$(L) = UCASE$(FS$(L))
      IF Contr = 1 THEN
         PRINT #2, "tipo di sezione"; L
         PRINT #2, "   forma della sezione "; CHR$(34); FS$(L); CHR$(34)
      END IF
      ' legge dati sezioni
      IF FS$(L) = "R" THEN
         INPUT #1, ZB(L), ZH(L)
         IF Contr = 1 THEN
            PRINT #2, "   sezione rettangolare     B ="; ZB(L); "   H ="; ZH(L)
         END IF
         ZA(L) = ZB(L) * ZH(L)
```

```
            ZI(L) = ZB(L) * ZH(L) ^ 3 / 12
            ZAT(L) = ZA(L) / 1.2
         ELSEIF FS$(L) = "G" THEN
            INPUT #1, ZA(L), ZAT(L), ZI(L)
            IF Contr = 1 THEN
               PRINT #2, "   sezione generica     A ="; ZA(L);
               PRINT #2, "   A/Chi ="; ZAT(L); "   I ="; ZI(L)
            END IF
         ELSE
            ER = 1
            PRINT "Il codice che identifica la forma della sezione"; L;
            PRINT " non e' ammissibile"
         END IF
      NEXT L

END SUB

' ============================== Inserimento ==============================
'
'   Procedura per l'inserimento di un termine
'   nella matrice di rigidezza globale della struttura
'
'   Variabili di ingresso:
'      P         primo indice
'      Q         secondo indice
'      V         valore da inserire
'      NL        ordine della matrice KTT()
'      SB        ampiezza della semibanda della matrice KAA()
'      KTT()     sottomatrice Ktt della matrice di rigidezza della struttura
'      KTA()     sottomatrice Kta della matrice di rigidezza della struttura
'      KAA()     sottomatrice Kaa della matrice di rigidezza della struttura
'
'   Variabili di uscita:
'      KTT()     sottomatrice Ktt della matrice di rigidezza della struttura
'      KTA()     sottomatrice Kta della matrice di rigidezza della struttura
'      KAA()     sottomatrice Kaa della matrice di rigidezza della struttura
'
' ------------------------------------------------------------------------------
'
SUB Inserimento (P, Q, V, NL, SB, KTT(), KTA(), KAA())

    IF P > Q THEN
       ' non memorizza il valore, perche' situato nel triangolo inferiore
    ELSEIF P <= NL AND Q <= NL THEN
       KTT(P, Q) = KTT(P, Q) + V
    ELSEIF P <= NL AND Q > NL THEN
       KTA(P, Q - NL) = KTA(P, Q - NL) + V
    ELSEIF P > NL AND Q > NL AND Q - P < SB THEN
       KAA(P - NL, Q - P + 1) = KAA(P - NL, Q - P + 1) + V
    END IF

END SUB

' ============================== LCOPY ==============================
'
'   Effettua la copia dello schermo, in modalita' grafica, sulla stampante
'
'   Variabili di ingresso:
'      SCR       modalita' grafica corrente:
```

```
'                    2 = CGA
'      DENS       densita' di stampa:
'                    1 = densita' singola (piu' veloce, meno precisa)
'                    2 = densita' doppia (meno veloce, piu' precisa)
'
' ------------------------------------------------------------------------------
'
SUB LCOPY (SCR, DENS)

    SELECT CASE SCR
       CASE 2
          DEF SEG = &HB800
          LPRINT CHR$(27); CHR$(51); CHR$(24);
          WIDTH "LPT1:", 255
          IF DENS = 2 THEN
             C$ = CHR$(27) + CHR$(76) + CHR$(32) + CHR$(3)
          ELSE
             C$ = CHR$(27) + CHR$(75) + CHR$(144) + CHR$(1)
          END IF
          FOR M = 0 TO 79
             LPRINT C$;
             FOR L = 7920 TO 0 STEP -80
                P = M + L
                P1$ = CHR$(PEEK(P + 8192))
                P2$ = CHR$(PEEK(P))
                IF DENS = 2 THEN
                   LPRINT P1$; P1$; P1$; P1$; P2$; P2$; P2$; P2$;
                ELSE
                   LPRINT P1$; P1$; P2$; P2$;
                END IF
             NEXT L
             LPRINT
          NEXT M
          WIDTH "LPT1:", 80
          LPRINT CHR$(27); CHR$(51); CHR$(36);
       CASE ELSE
          ' attualmente non previsto
    END SELECT

END SUB

' ========================== MatAstaGlobale ==========================
'   Procedura che determina geometria e matrice di rigidezza globale di un'asta
' ------------------------------------------------------------------------------
SUB MatAstaGlobale (TA$, X1, X2, Y1, Y2, A, AT, Iner, K, B, E, G, Lun, SnAl, CsAl,
                    Ld(), Kg())

    SELECT CASE TA$
       CASE "T"
          IF DefTag = 1 THEN
             CALL Geometria(X1, X2, Y1, Y2, Lun, SnAl, CsAl, Ld())
             CHI = A / AT
             CALL Kg.IncTag(E, G, CHI, A, Iner, Lun, Ld(), Kg())
          ELSE
             CALL Asta.Inc(X1, X2, Y1, Y2, E, A, Iner, Lun, SnAl, CsAl, Ld(), Kg())
          END IF
       CASE "W"
          CALL Geometria(X1, X2, Y1, Y2, Lun, SnAl, CsAl, Ld())
          CALL Kg.Wnk(K, B, E, A, Iner, Lun, Ld(), Kg())
```

```
    END SELECT

END SUB

'============================== ModGraf ==============================
'
'   Definisce la migliore modalita' grafica possibile
'
'   Variabili di uscita:
'      Scheda    migliore modalita' grafica possibile; si prova nell'ordine:
'                   12 = VGA
'                   11 = MCGA
'                    8 = EGA
'                    2 = CGA
'                    3 = Hercules
'                    0 = possibile solo modalita' testo
'
' ------------------------------------------------------------------------------
'
SUB ModGraf (Scheda)

    Scheda1 = 12          ' VGA
    ON ERROR GOTO ModGrafErr
    SCREEN Scheda1
    ON ERROR GOTO 0
    Scheda = Scheda1

END SUB

' ============================== OutputIntestaz ==============================
'   Procedura per la stampa dell'intestazione
' ------------------------------------------------------------------------------
SUB OutputIntestaz (FI$, Intest$)

    PRINT #2, "File dati: "; FI$;
    PRINT #2, "           data: "; MID$(DATE$, 4, 2) + "/" + MID$(DATE$, 1, 2);
    PRINT #2, "/" + RIGHT$(DATE$, 4); "     ora: "; TIME$
    PRINT #2, : PRINT #2,
    PRINT #2, Intest$
    PRINT #2, : PRINT #2, : PRINT #2,

END SUB

' ============================== OutputModElast ==============================
'   Procedura per la stampa dei moduli di elasticita'
' ------------------------------------------------------------------------------
SUB OutputModElast (DefTag, EL, GL)

    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "Modulo di elasticita' normale =     "; EL; " N/mm^2"
    IF DefTag = 1 THEN
       PRINT #2, "Modulo di elasticita' tangenziale = "; GL; " N/mm^2"
    END IF

END SUB

' ============================== Scala ==============================
'   Procedura per la definizione della scala per rappresentazione monometrica
'      (vedi: A.Ghersi, Introduzione alla programmazione, CUEN, Napoli, 1987)
```

```
' ---------------------------------------------------------------------------
SUB Scala (X1, Y1, X2, Y2)

    FX = .125
    FY = .075
    SC = (X2 - X1) / FX
    IF SC < (Y2 - Y1) / FY THEN SC = (Y2 - Y1) / FY
    LX = FX * SC
    LY = FY * SC
    XC = (X1 + X2) / 2
    YC = (Y1 + Y2) / 2
    X1 = XC - LX / 2
    X2 = XC + LX / 2
    Y1 = YC - LY / 2
    Y2 = YC + LY / 2

END SUB

' ============================== TermNoto ==============================
'   Procedura per la valutazione del vettore termine noto R(),
'   come somma dei carichi nodali esterni e delle azioni di incastro
'   trasmesse ai nodi dalle aste
' ---------------------------------------------------------------------------
SUB TermNoto (IZ, JZ, X(), Y(), ZA(), ZI(), E, E1(), E2(), TA$(), NT(), K(), B(),
              N(), P(), F(), R())

    DIM Ld(6, 6), LdT(6, 6), Kg(6, 6), S(6, 1), Sg(6, 1)

    CALL AzzeraMat(R())
    FOR I = 1 TO IZ
       FOR K = 1 TO 3
          P = N(I, K)
          IF P <> 0 THEN R(P) = R(P) + F(I, K)
       NEXT K
    NEXT I
    FOR J = 1 TO JZ
       IF P(J, 1) <> 0 OR P(J, 2) <> 0 THEN
          CALL Geometria(X(E1(J)), X(E2(J)), Y(E1(J)), Y(E2(J)), Lun,SnAl,CsAl,Ld())
          IF TA$(J) = "W" THEN
             CALL Kg.Wnk(K(J) * 1000, B(J), E, ZA(NT(J)), ZI(NT(J)), Lun, Ld(),Kg())
             CALL TrasponeMat(Ld(), LdT())
          END IF
          CALL AzIncLocale(P(J, 1), P(J, 2), TA$(J), B(J), K(J) * 1000, Lun, SnAl,
                           CsAl, Ld(), LdT(), Kg(), S())
          CALL ProdottoMat(Ld(), S(), Sg())
          FOR K = 1 TO 6
             IF K <= 3 THEN N = N(E1(J), K) ELSE N = N(E2(J), K - 3)
             IF N <> 0 THEN R(N) = R(N) - Sg(K, 1)
          NEXT K
       END IF
    NEXT J

END SUB

' ============================== VincoliCed ==============================
'   Procedura per l'aggiunta della rigidezza dei vincoli cedevoli
'   alla matrice di rigidezza
' ---------------------------------------------------------------------------
SUB VincoliCed (IZ, N(), KV(), NL, SB, KTT(), KTA(), KAA())
```

```
    FOR I = 1 TO IZ
       FOR K = 1 TO 3
          IF N(I, K) <> 0 THEN
             P = N(I, K)
             CALL Inserimento(P, P, KV(I, K), NL, SB, KTT(), KTA(), KAA())
          END IF
       NEXT K
    NEXT I

END SUB



'                                  PROGEN.BAS
'
'  procedure del programma TELGEN, utilizzate anche nel programma TELSPA
'
'                                                                rev. 01.90


' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file MATRIX.BAS
DECLARE SUB AzzeraMat (A!())

' ================================== InputAsteGen ==================================
'   Procedura per la lettura dei dati relativi alle aste
' ----------------------------------------------------------------------------------
SUB InputAsteGen (Contr, JZ, LZ, E1(), E2(), NT(), TA$(), Wnkl, B(), K(), ER)

      ' controlla ammissibilita' dell'identificatore del tipo di asta
    Wnkl = 0
    FOR J = 1 TO JZ
       INPUT #1, E1(J), E2(J), TA$(J), NT(J)
       TA$(J) = UCASE$(TA$(J))
       IF Contr = 1 THEN
          PRINT #2, "asta"; J, "   estremi: "; E1(J); E2(J),
          PRINT #2, "asta tipo "; CHR$(34); TA$(J); CHR$(34),
          PRINT #2, "    sezione tipo"; NT(J)
       END IF
       IF TA$(J) <> "T" AND TA$(J) <> "W" THEN
          PRINT "Il codice che identifica il tipo di asta per l'asta"; J;
          PRINT " non e' ammissibile"
          ER = 1
       END IF
       L = NT(J)
       IF L < 1 OR L > LZ THEN
          PRINT "Tipo di sezione inaccettabile per la trave "; J
          ER = 1
       ELSE
          IF TA$(J) = "W" THEN
             Wnkl = 1
             INPUT #1, B(J), K(J)
             IF Contr = 1 THEN
                PRINT #2, "   asta su suolo alla Winkler    B =";B(J); "   K =";K(J)
             END IF
          END IF
```

```
      END IF
   NEXT J

END SUB

' ============================== InputCarGen ==============================
'   Procedura per la lettura dei carichi nodali e sulle aste;
'   controlla che i carichi siano applicati a nodi ed aste esistenti
' -------------------------------------------------------------------------
SUB InputCarGen (Contr, IZ, JZ, QN, QA, F(), P(), ER)

   INPUT #1, QN, QA
   CALL AzzeraMat(F())
   CALL AzzeraMat(P())
   IF QN <> 0 THEN
      IF Contr = 1 THEN PRINT #2, QN; " carichi nodali"
      FOR Q = 1 TO QN
         INPUT #1, I, F1, F2, F3
         IF I < 1 OR I > IZ THEN
            PRINT "Carico inaccettabile perche' applicato al nodo"; I
            ER = 1
         ELSE
            F(I, 1) = F1
            F(I, 2) = F2
            F(I, 3) = F3
         END IF
         IF Contr = 1 THEN
            PRINT #2, "   nodo"; I, "Fx ="; F1, "Fy ="; F2, "M ="; F3
         END IF
      NEXT Q
   END IF
   IF QA <> 0 THEN
      IF Contr = 1 THEN PRINT #2, QA; " carichi distribuiti sulle aste"
      FOR Q = 1 TO QA
         INPUT #1, J, P1, P2
         IF J < 1 OR J > JZ THEN
            PRINT "Carico inaccettabile perche' applicato all'asta"; J
            ER = 1
         ELSE
            P(J, 1) = P1
            P(J, 2) = P2
         END IF
         IF Contr = 1 THEN
            PRINT #2, "   asta"; J, "px ="; P1, "py ="; P2
         END IF
      NEXT Q
   END IF

END SUB

' ============================== InputNodiGen ==============================
'   Procedura per la lettura dei dati relativi ai nodi
' -------------------------------------------------------------------------
SUB InputNodiGen (Contr, IZ, X(), Y(), V$(), VCed, KV(), ER)

   VCed = 0
   FOR I = 1 TO IZ
      INPUT #1, X(I), Y(I), V$(I)
      V$(I) = UCASE$(V$(I))
```

```
      IF Contr = 1 THEN
         PRINT #2, "nodo"; I, "x ="; X(I), "y ="; Y(I), "vincolo "; V$(I)
      END IF
      FOR K = 1 TO 3
         EV$ = MID$(V$(I), K, 1)
         IF EV$ = "1" OR EV$ = "0" THEN
            KV(I, K) = 0
         ELSEIF EV$ = "C" THEN
            VCed = 1
            INPUT #1, KV(I, K)
            IF Contr = 1 THEN
               PRINT #2, "      costante elastica"; K; " ="; KV(I, K)
            END IF
         ELSE
            PRINT "Elemento"; K; " del codice vincolo nodo"; I; " non ammissibile"
            ER = 1
         END IF
      NEXT K
   NEXT I

END SUB

' ============================== InputTraversiGen ==============================
'   Procedura per la lettura dei dati relativi a traversi inestensibili
' ------------------------------------------------------------------------------
SUB InputTraversiGen (Contr, KZ, NI())

   IF KZ <> 0 THEN
      I = 0
      FOR K = 1 TO KZ
         IF Contr = 1 THEN
            PRINT #2, "traverso orizzontale inestensibile"; K; " :  ";
         END IF
         DO
            I = I + 1
            INPUT #1, NI(I)
            IF Contr = 1 AND NI(I) <> 0 THEN PRINT #2, NI(I);
         LOOP UNTIL NI(I) = 0
         IF Contr = 1 THEN PRINT #2,
      NEXT K
   END IF

END SUB

' ============================== OutputAsteGen ==============================
'   Procedura per la stampa dei dati delle aste
' ------------------------------------------------------------------------------
SUB OutputAsteGen (DefTag, Wnkl, JZ, NT(), TA$(), E1(), E2(), FS$(), ZB(), ZH(),
                   ZA(), ZAT(), ZI(), B(), K())

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "DATI RELATIVI ALLE TRAVI": PRINT #2,
   PRINT #2, "                                b     h     area ";
   IF DefTag = 1 THEN PRINT #2, "   A/Chi ";
   PRINT #2, "   inerzia ";
   IF Wnkl = 1 THEN PRINT #2, "   B      K  ";
   PRINT #2,
   PRINT #2, "asta  tipo  estremi  forma    m     m      m^2  ";
   IF DefTag = 1 THEN PRINT #2, "     m^2  ";
```

```
    PRINT #2, "      m4    ";
    IF Wnkl = 1 THEN PRINT #2, "   m     N/cm3";
    PRINT #2, : PRINT #2,
    FOR J = 1 TO JZ
       L = NT(J)
       PRINT #2, USING "###      !   ### ###    !  ";J,TA$(J),E1(J),E2(J),FS$(L);
       IF FS$(L) = "R" THEN
          PRINT #2, USING " ##.## ##.##"; ZB(L), ZH(L);
       ELSE
          PRINT #2, "  ----  ----";
       END IF
       PRINT #2, USING " ###.####"; ZA(L);
       IF DefTag = 1 THEN
          IF TA$(J) = "W" THEN
             PRINT #2, "    ---- ";
          ELSE
             PRINT #2, USING " ###.####"; ZAT(L);
          END IF
       END IF
       PRINT #2, USING " ###.######"; ZI(L);
       IF Wnkl = 1 THEN
          IF TA$(J) = "W" THEN
             PRINT #2, USING " ##.## ###.#"; B(J), K(J);
          ELSE
             PRINT #2, "  ----  ----";
          END IF
       END IF
       PRINT #2,
    NEXT J

END SUB

' ============================== OutputCarGen ==============================
'   Procedura per la stampa dei carichi nodali e sulle aste
' ----------------------------------------------------------------------------
SUB OutputCarGen (CAR, IZ, JZ, QN, QA, F(), P())

    IF QN <> 0 THEN
       PRINT #2, "CARICHI NODALI": PRINT #2,
       PRINT #2, "nodo        Fx (kN)      Fy (kN)      M (kNm)": PRINT #2,
       FOR I = 1 TO IZ
          IF F(I, 1) <> 0 OR F(I, 2) <> 0 OR F(I, 3) <> 0 THEN
             PRINT #2, USING "###  "; I;
             PRINT #2, USING "     #####.##"; F(I, 1), F(I, 2), F(I, 3)
          END IF
       NEXT I
       PRINT #2, : PRINT #2,
    END IF
    IF QA <> 0 THEN
       PRINT #2, "CARICHI SULLE ASTE": PRINT #2,
       PRINT #2, "asta       px (kN/m)    py (kN/m)": PRINT #2,
       FOR J = 1 TO JZ
          IF P(J, 1) <> 0 OR P(J, 2) <> 0 THEN
             PRINT #2, USING "###  "; J,
             PRINT #2, USING "     #####.##"; P(J, 1), P(J, 2)
          END IF
       NEXT J
    END IF
    PRINT #2,
```

```
END SUB

' ============================== OutputNodiGen ==============================
'   Procedura per la stampa dei dati dei nodi
' ---------------------------------------------------------------------------
SUB OutputNodiGen (IZ, X(), Y(), V$(), VCed, KV())

    PRINT #2, "COORDINATE E VINCOLI DEI NODI": PRINT #2,
    PRINT #2, "nodo       x (m)        y (m)      vincolo";
    IF VCed = 1 THEN PRINT #2, "      rigidezza del vincolo";
    PRINT #2, : PRINT #2,
    FOR I = 1 TO IZ
       PRINT #2, USING "###    #####.##    #####.##       \\  ";I, X(I),Y(I),V$(I);
       IF VCed <> 0 THEN
          FOR K = 1 TO 3
             IF KV(I, K) <> 0 THEN
                PRINT #2, USING " ##.####^^^^"; KV(I, K);
             ELSE
                PRINT #2, "       -    ";
             END IF
          NEXT K
       END IF
       PRINT #2,
    NEXT I

END SUB

' ============================== OutputRisGen ==============================
'   Procedura per la stampa dei risultati
' ---------------------------------------------------------------------------
SUB OutputRisGen (IZ, JZ, KZ, NL, V$(), NI(), N(), R(), M1(), M2(), T1(), T2(),
                  N1(), N2(), F())

    DIM V(3)

    PRINT #2, : PRINT #2,
    PRINT #2, "SPOSTAMENTI E ROTAZIONI DEI NODI": PRINT #2,
    PRINT #2, "nodo          vx (mm)      vy (mm)   rotaz.x1000": PRINT #2,
    FOR I = 1 TO IZ
       FOR K = 1 TO 3
          IF N(I, K) = 0 THEN V(K) = 0 ELSE V(K) = R(N(I, K)) * 1000
       NEXT K
       PRINT #2, USING"###       #####.###   #####.###   #####.###";I,V(1),V(2),V(3)
    NEXT I
    ' stampa caratteristiche di sollecitazione
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "CARATTERISTICHE DI SOLLECITAZIONE NELLE ASTE": PRINT #2,
    PRINT #2, "asta      M1 (kNm)   M2 (kNm)    T1 (kN)    T2 (kN)    N1 (kN)    N2
              (kN)"
    PRINT #2,
    FOR J = 1 TO JZ
       PRINT #2, USING "###   "; J;
       PRINT #2, USING "   #####.##"; M1(J), M2(J), T1(J), T2(J), N1(J), N2(J)
    NEXT J
    ' stampa azioni su vincoli esterni ed eventuale squilibrio
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "AZIONI SUI VINCOLI ESTERNI": PRINT #2,
    PRINT #2, "nodo        Fx (kN)      Fy (kN)      M (kNm)": PRINT #2,
```

```
Squil = 0
FOR I = 1 TO IZ
   V = 0
   FOR K = 1 TO 3
      IF MID$(V$(I), K, 1) = "0" THEN V(K) = 0 ELSE V(K) = 1
      V = V + V(K)
   NEXT K
   IF V <> 0 THEN
      PRINT #2, USING "###  "; I;
   END IF
   FOR K = 1 TO 3
      IF V(K) = 0 THEN
         IF N(I, K) > NL AND ABS(F(I, K)) > .01 THEN Squil = 1
         IF V <> 0 THEN PRINT #2, "          ----";
      ELSE
         PRINT #2, USING "     #####.##"; F(I, K);
         F(I, K) = 0
      END IF
   NEXT K
   IF V <> 0 THEN PRINT #2,
NEXT I
IF Squil = 1 THEN
   PRINT #2, : PRINT #2,
   PRINT #2, "SQUILIBRIO NEI NODI": PRINT #2,
   PRINT #2, "nodo        Fx (kN)      Fy (kN)      M (kNm)": PRINT #2,
   FOR I = 1 TO IZ
      V = 0
      FOR K = 1 TO 3
         IF N(I, K) > NL AND ABS(F(I, K)) > .01 THEN V(K) = 1 ELSE V(K) = 0
         V = V + V(K)
      NEXT K
      IF V <> 0 THEN
         PRINT #2, USING "###  "; I;
         FOR K = 1 TO 3
            IF V(K) = 0 THEN
               PRINT #2, "          ----";
            ELSE
               PRINT #2, USING "     #####.##"; F(I, K);
            END IF
         NEXT K
         PRINT #2,
      END IF
   NEXT I
END IF
' stampa azioni su vincoli mutui ed eventuale squilibrio
IF KZ > 0 THEN
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "AZIONI SUI VINCOLI MUTUI (TRAVERSI ORIZZONTALI)": PRINT #2,
   NN = 0
   I = 0
   FOR K = 1 TO KZ
      V = -1
      DO
         I = I + 1
         V = V + 1
      LOOP UNTIL NI(I) = 0
      IF NN < V THEN NN = V
   NEXT K
   IF NN > 4 THEN NN = 4
```

```
      PRINT #2, "traverso";
      FOR V = 1 TO NN
         PRINT #2, "   nodo   Fx (kN)";
      NEXT V
      PRINT #2, : PRINT #2,
      FOR K = 1 TO KZ
         PRINT #2, USING "  ###      "; K;
         F = 0
         V = 0
         FOR I = 1 TO IZ
            IF N(I, 1) = K THEN
               V = V + 1
               IF V = 5 THEN
                  PRINT #2, : PRINT #2, "           ";
                  V = 1
               END IF
               F = F + F(I, 1)
               PRINT #2, USING "###  #####.##    "; I, F(I, 1);
            END IF
         NEXT I
         PRINT #2,
         IF ABS(F) > .01 THEN
            PRINT #2, USING "          squilibrio = #####.## kN"; F
         END IF
      NEXT K
   END IF

END SUB
```

5. Modalità di utilizzazione del programma

I dati necessari per descrivere lo schema geometrico e di carico, elencati in dettaglio nel seguito, devono essere conservati in un file su memoria di massa, che può essere preparato mediante un qualsiasi word processor tra quelli esistenti in commercio.

Quando il programma viene mandato in esecuzione, esso richiede espressamente in maniera interattiva solo le seguenti informazioni:

- il nome del file che contiene i dati;
- il nome dell'unità alla quale inviare l'output; questa può essere lo schermo (CON), la stampante (PRN) oppure un file del quale si indicherà il nome;
- se nel calcolo si vuole tener conto della deformabilità a taglio (si risponda 1 in caso affermativo, 0 in caso negativo). La versione attuale del programma prevede la deformabilità a taglio solo per aste non collegate ad un suolo elastico; assume inoltre automaticamente un valore del modulo di elasticità tangenziale G legato a quello del modulo di Young E dalla relazione $G = 0.45\ E$, valida per $\nu = 0.1$ (cemento armato);

- se si vuole effettuare un controllo dei dati man mano che vengono letti. Conviene utilizzare questa opzione quando il programma si blocca a causa di qualche errore nei dati che non si riesce ad individuare. Il controllo consiste nell'inviare all'unità di uscita una segnalazione per ciascun valore letto, nella quale il programma indica come questo valore è stato interpretato.

Il programma passa poi a leggere dal file indicato i dati geometrici e di carico. Ciascuna esecuzione consente la risoluzione di uno schema geometrico soggetto a più condizioni di carico. I dati devono essere memorizzati nel file in maniera sequenziale, separati da virgole o da andate a capo, col seguente ordine:

- gruppo di dati che descrivono la geometria dello schema
- gruppo di dati che descrivono la prima condizione di carico
- gruppo di dati che descrivono la seconda condizione di carico

  .....
- gruppo di dati che descrivono l'ultima condizione di carico

Nel preparare lo schema geometrico, sia i nodi (liberi o vincolati) che le aste devono essere numerati progressivamente a partire dal valore 1. Si ricorda che il criterio adoperato nella numerazione automatica delle incognite è legato alla numerazione dei nodi; di conseguenza, questa condiziona l'ampiezza della banda della matrice di rigidezza della struttura e quindi il tempo di esecuzione e l'ingombro di memoria. È pertanto opportuno che l'utente eviti di assegnare numeri molto distanti tra loro a nodi direttamente collegati da aste. L'ordine di numerazione delle aste può invece essere arbitrario, perché non ha alcuna influenza sull'ampiezza della banda.

Nelle applicazioni pratiche numerose aste di uno schema presentano la stessa sezione. Si è pertanto ritenuto più comodo per la immissione dei dati che l'utente individui preliminarmente i differenti tipi di sezione, e definisca poi per ogni asta il numero d'ordine della sezione corrispondente. Per ciascun tipo di sezione deve essere definita la forma (rettangolare o generica) e le relative caratteristiche geometriche.

Il gruppo di dati che descrive la geometria dello schema deve rispettare l'ordine di seguito riportato. I valori forniti devono essere conformi alle unità di misura indicate tra parentesi:

- intestazione
- numero totale di nodi dello schema
- numero di aste presenti nello schema

- numero di traversi orizzontali indeformabili (ovvero di gruppi di nodi aventi lo stesso spostamento orizzontale)
- numero di tipi di sezioni
- per ogni nodo:
  - ascissa del nodo nel riferimento globale (m)
  - ordinata del nodo nel riferimento globale (m)
  - codice di vincolo. È costituito da una stringa con tre caratteri, che si riferiscono alle tre componenti di movimento del nodo (nell'ordine: traslazione in direzione x ed y e rotazione). Si indica con `0` una componente di movimento consentita, con `1` una impedita, con `C` una limitata da un vincolo cedevole
  - costante elastica di eventuali vincoli cedevoli (kN m^{-1} per vincoli allo spostamento, kNm per vincoli alla rotazione). La costante elastica deve essere fornita solo per quei gradi di libertà per i quali si è usato `C` come codice di vincolo
- per ogni traverso orizzontale inestensibile:
  nodi collegati dal traverso, che devono quindi avere lo stesso spostamento orizzontale. L'elenco dei nodi deve terminare con uno 0, che indica al calcolatore la fine dei nodi del traverso
- modulo di elasticità normale (N mm^{-2})
- per ogni tipo di sezione:
  - carattere che identifica la forma della sezione:
    `R` = sezione di forma rettangolare
    `G` = sezione di forma generica
  - caratteristiche geometriche della sezione:
    - base (m) ed altezza (m) per le sezioni rettangolari
    - area (m^2), rapporto tra area e fattore di taglio (m^2) e momento di inerzia (m^4) per quelle generiche
- per ogni asta:
  - primo estremo
  - secondo estremo
  - carattere che identifica il tipo di asta:
    `T` = trave alla De Saint Venant solidale ai nodi di estremità
    `W` = trave su suolo elastico alla Winkler
  - tipo di sezione
  - larghezza della sottobase, se l'asta poggia su un suolo elastico alla Winkler (m)
  - costante di sottofondo, se l'asta poggia su un suolo elastico alla Winkler (N cm^{-3})

Per ciascuno schema di carico occorre rispettare la convenzione dei segni definita nella prima parte del testo. Le forze nodali F_x e F_y sono positive se concordi col verso positivo degli assi x ed y; il momento nodale M_z è positivo se antiorario, cioè nel verso che porta l'asse x sull'asse y. Il carico uniformemente distribuito sull'asta è descritto mediante le componenti rispetto al riferimento globale p_x e p_y, positive se concordi col verso positivo degli assi x ed y. Si noti che usualmente l'asse y è considerato positivo se orientato verso l'alto; in tal caso i carichi verticali che agiscono verso il basso devono essere indicati col segno meno.

Poiché in genere solo alcuni nodi ed aste sono soggetti a carichi, si è ritenuto preferibile che l'utente indichi preliminarmente quanti nodi e quante aste sono caricati e specifichi poi solo per questi il numero d'ordine del nodo ed i relativi valori di F_x, F_y e M_z e il numero d'ordine dell'asta ed i relativi valori di p_x e p_y.

Il gruppo di dati che descrive la singola condizione di carico deve rispettare l'ordine e le unità di misura di seguito indicati:

- numero di nodi in cui sono applicati carichi concentrati
- numero di aste soggette a carico uniformemente distribuito
- per ciascun nodo caricato:
  - numero d'ordine del nodo
  - F_x, componente in direzione x della forza concentrata (kN)
  - F_y, componente in direzione y della forza concentrata (kN)
  - M_z, coppia concentrata (kNm)
- per ciascuna asta caricata:
  - numero d'ordine dell'asta
  - p_x, componente nella direzione x del riferimento globale del carico distribuito, valutato per unità di lunghezza dell'asta (kN m^{-1})
  - p_y, componente nella direzione y del riferimento globale del carico distribuito, valutato per unità di lunghezza dell'asta (kN m^{-1})

Si ricordi infine che il file dati non deve contenere dopo l'ultimo blocco di dati alcuna riga bianca, che verrebbe intepretata dal calcolatore come un valore nullo ma darebbe inizio comunque alla lettura dei dati di un ulteriore schema di carico.

6. Risultati forniti

L'output del programma contiene innanzitutto l'elenco dei dati geometrici ed elastici forniti in ingresso dall'utente, per consentirne il neces-

sario controllo. A tal fine, si è ritenuto preferibile riportare le informazioni relative al tipo di sezione a fianco di ciascuna asta, anziché rinviare ad una distinta tabella dei tipi di sezione. Non vengono stampati valori irrilevanti ai fini del calcolo: ad esempio, se non si è tenuto conto della deformabilità a taglio non viene indicato il rapporto A/χ anche se esso deve essere sempre fornito per le sezioni di forma generica. I nodi appartenenti a ciascun traverso inestensibile non sono elencati espressamente in questa fase, ma un riscontro può essere effettuato esaminando le indicazioni sui vincoli mutui poste alla fine dei risultati di ogni condizione di carico.

Per ciascuno schema di carico vengono elencati i nodi e le aste caricati e l'entità dei carichi ad essi applicati.

Sono poi riportate le componenti di movimento dei nodi ottenute come risultato del calcolo. Gli spostamenti v_x e v_y, positivi se concordi col verso positivo degli assi x ed y, sono espressi in mm. La rotazione θ_z, positiva se antioraria cioè nel verso che porta l'asse x sull'asse y, è indicata in radianti amplificata di un fattore 1000.

I valori delle caratteristiche della sollecitazione M, T, N nei due estremi sono espressi in kNm e kN. Si ricorda che il momento flettente è positivo se tende le fibre inferiori, guardando l'asta in modo da avere a sinistra il primo estremo e a destra il secondo; il taglio è positivo se la microcoppia costituita dalle azioni taglianti su due facce opposte è antioraria; lo sforzo normale è positivo se di trazione.

Sono infine riportate separatamente le azioni esplicate dalla struttura sui vincoli esterni e quelle sui vincoli mutui (traversi orizzontali). Se si riscontrano squilibri in nodi non vincolati o nei traversi orizzontali, dovuti agli inevitabili errori di troncamento del procedimento risolutivo, questi vengono espressamente segnalati per consentire all'utente di rendersi conto della loro entità e quindi della maggiore o minore validità dei risultati ottenuti.

7. Esempio

Il programma è stato utilizzato per la risoluzione di uno schema di telaio con travi a ginocchio solidale ad una trave di fondazione su suolo elastico alla Winkler (fig.5.1), soggetto a carichi verticali ed orizzontali. Di seguito sono mostrati i dati, contenuti nel file **SCALA**, ed i conseguenti risultati.

```
TELAIO CON TRAVE A GINOCCHIO E TRAVE DI FONDAZIONE
20,27,3,6
```

```
0,0,000,1.5,0,100,6.5,0,100,12.5,0,100,14,0,000
11,2.2,000,12.5,2.2,000
1.5,3.8,000,6.5,3.8,000,8,3.8,000,12.5,3.8,000
11,5.4,000,12.5,5.4,000
1.5,7.0,000,6.5,7.0,000,8,7.0,000,12.5,7.0,000
1.5,10.2,000,6.5,10.2,000,12.5,10.2,000
8,9,10,11,0,14,15,16,17,0,18,19,20,0
25000
R,.3,.6,R,.6,.3,R,.8,.24,R,.3,.5,R,.3,.6,G,.57,.33,.057307
2,8,T,1,3,9,T,1,4,7,T,2,7,11,T,2
8,14,T,1,9,15,T,1,11,13,T,2,13,17,T,2
14,18,T,1,15,19,T,1,17,20,T,2
8,9,T,3,9,10,T,5,10,11,T,5,10,6,T,5,6,7,T,5
14,15,T,3,15,16,T,5,16,17,T,5,16,12,T,5,12,13,T,5
18,19,T,3,19,20,T,4
1,2,W,6,1.4,50,2,3,W,6,1.4,50,3,4,W,6,1.4,50,4,5,W,6,1.4,50
3,12
8,30,0,0,14,60,0,0,18,90,0,0
12,0,-8,13,0,-25,14,0,-8,15,0,-20,16,0,-22
17,0,-8,18,0,-25,19,0,-8,20,0,-20,21,0,-22
22,0,-5,23,0,-5
```

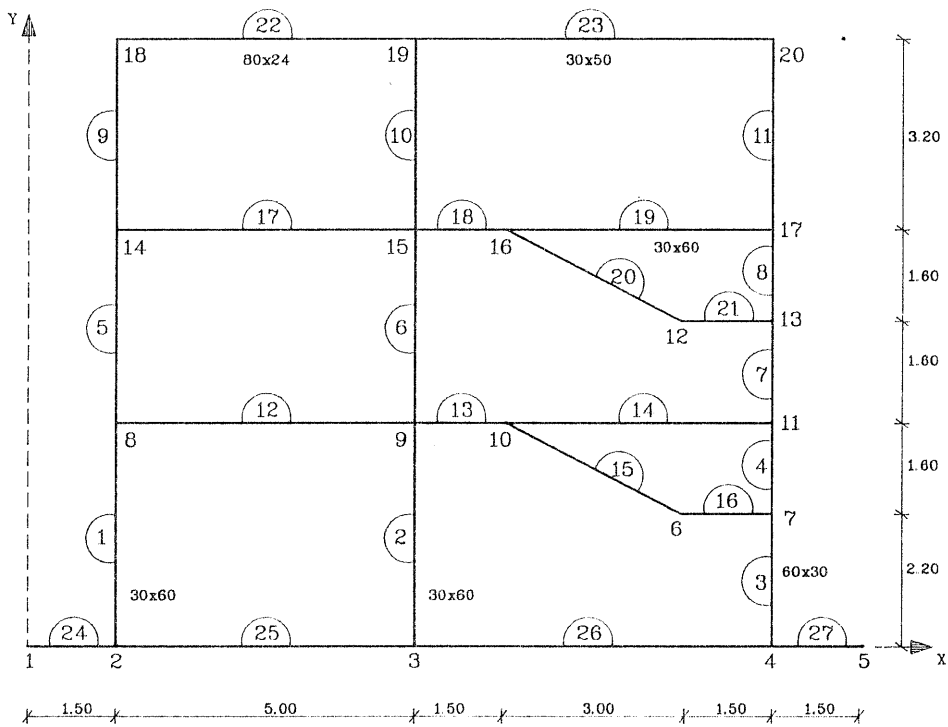


Fig. 5.1 — Schema di telaio con trave a ginocchio e trave di fondazione

File dati: SCALA data: 14/02/1990 ora: 17:24:36

TELAIO CON TRAVE A GINOCCHIO E TRAVE DI FONDAZIONE

COORDINATE E VINCOLI DEI NODI

nodo	x (m)	y (m)	vincolo
1	0.00	0.00	000
2	1.50	0.00	100
3	6.50	0.00	100
4	12.50	0.00	100
5	14.00	0.00	000
6	11.00	2.20	000
7	12.50	2.20	000
8	1.50	3.80	000
9	6.50	3.80	000
10	8.00	3.80	000
11	12.50	3.80	000
12	11.00	5.40	000
13	12.50	5.40	000
14	1.50	7.00	000
15	6.50	7.00	000
16	8.00	7.00	000
17	12.50	7.00	000
18	1.50	10.20	000
19	6.50	10.20	000
20	12.50	10.20	000

DATI RELATIVI ALLE TRAVI

asta	tipo	estremi		forma	b m	h m	area m2	inerzia m4	B m	K N/cm3
1	T	2	8	R	0.30	0.60	0.1800	0.005400	----	----
2	T	3	9	R	0.30	0.60	0.1800	0.005400	----	----
3	T	4	7	R	0.60	0.30	0.1800	0.001350	----	----
4	T	7	11	R	0.60	0.30	0.1800	0.001350	----	----
5	T	8	14	R	0.30	0.60	0.1800	0.005400	----	----
6	T	9	15	R	0.30	0.60	0.1800	0.005400	----	----
7	T	11	13	R	0.60	0.30	0.1800	0.001350	----	----
8	T	13	17	R	0.60	0.30	0.1800	0.001350	----	----
9	T	14	18	R	0.30	0.60	0.1800	0.005400	----	----
10	T	15	19	R	0.30	0.60	0.1800	0.005400	----	----
11	T	17	20	R	0.60	0.30	0.1800	0.001350	----	----
12	T	8	9	R	0.80	0.24	0.1920	0.000922	----	----
13	T	9	10	R	0.30	0.60	0.1800	0.005400	----	----
14	T	10	11	R	0.30	0.60	0.1800	0.005400	----	----
15	T	10	6	R	0.30	0.60	0.1800	0.005400	----	----
16	T	6	7	R	0.30	0.60	0.1800	0.005400	----	----
17	T	14	15	R	0.80	0.24	0.1920	0.000922	----	----
18	T	15	16	R	0.30	0.60	0.1800	0.005400	----	----
19	T	16	17	R	0.30	0.60	0.1800	0.005400	----	----
20	T	16	12	R	0.30	0.60	0.1800	0.005400	----	----

21	T	12	13	R	0.30	0.60	0.1800	0.005400	----	----
22	T	18	19	R	0.80	0.24	0.1920	0.000922	----	----
23	T	19	20	R	0.30	0.50	0.1500	0.003125	----	----
24	W	1	2	G	----	----	0.5700	0.057307	1.40	50.0
25	W	2	3	G	----	----	0.5700	0.057307	1.40	50.0
26	W	3	4	G	----	----	0.5700	0.057307	1.40	50.0
27	W	4	5	G	----	----	0.5700	0.057307	1.40	50.0

Modulo di elasticita' normale = 25000 N/mm^2

SCHEMA DI CARICO N. 1

CARICHI NODALI

nodo	Fx (kN)	Fy (kN)	M (kNm)
8	30.00	0.00	0.00
14	60.00	0.00	0.00
18	90.00	0.00	0.00

CARICHI SULLE ASTE

asta	px (kN/m)	py (kN/m)
12	0.00	-8.00
13	0.00	-25.00
14	0.00	-8.00
15	0.00	-20.00
16	0.00	-22.00
17	0.00	-8.00
18	0.00	-25.00
19	0.00	-8.00
20	0.00	-20.00
21	0.00	-22.00
22	0.00	-5.00
23	0.00	-5.00

SPOSTAMENTI E ROTAZIONI DEI NODI

nodo	vx (mm)	vy (mm)	rotaz.x1000
1	0.000	0.226	-0.112
2	0.000	0.056	-0.117
3	0.000	-0.251	-0.088
4	0.000	-1.477	-0.378
5	0.000	-1.986	-0.327
6	3.055	-1.881	0.195
7	3.018	-1.652	-0.143
8	3.478	0.037	-1.046
9	3.478	-0.338	-0.859
10	3.478	-1.275	-0.399
11	3.478	-1.741	-0.647
12	5.914	-1.974	0.209
13	5.862	-1.812	-0.315

14	6.396	0.029	-0.996
15	6.396	-0.380	-0.870
16	6.396	-1.339	-0.451
17	6.396	-1.835	-0.294
18	10.189	0.027	-1.176
19	10.189	-0.398	-0.955
20	10.189	-1.855	-0.508

CARATTERISTICHE DI SOLLECITAZIONE NELLE ASTE

asta	M1 (kNm)	M2 (kNm)	T1 (kN)	T2 (kN)	N1 (kN)	N2 (kN)
1	-104.13	38.13	37.44	37.44	-22.54	-22.54
2	-121.53	66.78	49.55	49.55	-103.36	-103.36
3	-98.70	105.91	93.00	93.00	-358.10	-358.10
4	3.00	-24.28	-17.05	-17.05	-250.49	-250.49
5	29.68	-25.43	-17.22	-17.22	-12.25	-12.25
6	-12.51	11.54	7.52	7.52	-58.33	-58.33
7	-120.75	134.78	159.70	159.70	-198.92	-198.92
8	-3.26	4.13	4.62	4.62	-66.72	-66.72
9	-32.88	17.70	15.80	15.80	-1.66	-1.66
10	-72.66	65.50	43.18	43.18	-25.61	-25.61
11	-51.89	47.37	31.02	31.02	-27.73	-27.73
12	8.45	-40.06	10.30	-29.70	0.00	0.00
13	39.23	34.08	15.32	-22.18	0.00	0.00
14	54.61	-96.47	-15.57	-51.57	-0.00	-0.00
15	-20.52	33.76	45.96	-14.04	-100.22	-132.22
16	33.76	-102.90	-74.61	-107.61	-110.06	-110.06
17	7.45	-39.62	10.59	-29.41	0.00	0.00
18	44.58	21.41	3.31	-34.19	0.00	0.00
19	38.46	-56.02	-3.00	-39.00	-0.00	-0.00
20	-17.04	35.51	45.46	-14.54	-151.52	-183.52
21	35.51	-138.04	-99.20	-132.20	-155.09	-155.09
22	17.70	-36.50	1.66	-23.34	0.00	0.00
23	29.00	-47.37	2.27	-27.73	0.00	0.00
24	0.00	-13.39	-0.00	-14.89	0.00	0.00
25	90.75	-43.33	-37.43	-0.72	0.00	0.00
26	78.19	44.72	-104.08	175.69	0.00	0.00
27	143.42	-0.00	-182.40	-0.00	0.00	0.00

AZIONI SUI VINCOLI ESTERNI

nodo	Fx (kN)	Fy (kN)	M (kNm)
2	37.44	----	----
3	49.55	----	----
4	93.00	----	----

AZIONI SUI VINCOLI MUTUI (TRAVERSI ORIZZONTALI)

traverso	nodo	Fx (kN)	nodo	Fx (kN)	nodo	Fx (kN)	nodo	Fx (kN)
1	8	-24.66	9	-42.04	10	-110.06	11	176.76
2	14	93.03	15	35.66	16	-155.09	17	26.40
3	18	74.20	19	-43.18	20	-31.02		

Il calcolo è stato eseguito :

- tenendo conto della deformazione estensionale delle aste
- trascurando la deformazione a taglio delle aste

CAPITOLO SESTO

SCHEMI PARTICOLARI

1. Preprocessor e postprocessor

Il modello di insieme piano di aste esaminato nel capitolo precedente consente di affrontare e risolvere un gran numero di problemi strutturali. Oltre ai telai per edifici, già esplicitamente richiamati, si possono esaminare con esso ad esempio le travature reticolari, le struttura ad arco (schematizzate mediante tanti brevi conci rettilinei) oppure i pali di fondazione soggetti ad azioni orizzontali. La generalità del modello può però comportare problemi nell'immissione dei dati, nel loro controllo e nell'esame dei risultati. Le informazioni da fornire al calcolatore per descrivere lo schema sono parecchie e il rischio di errori cresce enormemente nel caso di strutture complesse, con molti nodi ed aste. Il controllo dei valori immessi è lungo e noioso ed è quindi facile che qualcosa sfugga. I risultati sono presentati usando come indicatori i numeri dei nodi e delle aste, che in genere non consentono un immediato riscontro geometrico; per rendersi conto di quale sia il nodo 7 o l'asta 12 diventa necessario consultare l'elenco dei dati o qualche disegno preparato in precedenza, e questa ricerca distoglie l'attenzione dai risultati e rende difficile mantenerne una visione globale.

L'uso di modelli strutturali generali può essere reso più semplice affiancando al programma di calcolo delle apposite procedure, che fungano da intermediari tra esso e l'utente sia nell'immissione dei dati che nella presen-

tazione dei risultati. Nel primo caso si parla di preprocessor, o interfacce di ingresso, nel secondo di postprocessor, o interfacce di uscita.

Notevoli risultati possono essere ottenuti sfruttando le capacità grafiche dei personal computer. Lo schema geometrico della struttura può essere descritto disegnandolo man mano sullo schermo mediante comandi espressamente approntati, oppure utilizzando programmi generali per la grafica come Autocad. Le informazioni relative a ciascuna asta, non graficizzabili, possono essere fornite in maniera interattiva dopo aver espressamente individuato sullo schermo l'asta cui esse si riferiscono. I risultati possono essere presentati sotto forma di diagrammi delle caratteristiche di sollecitazione e degli spostamenti, consentendo un'immagine globale ed immediata dell'effetto provocato dai carichi assegnati.

Procedure di questo tipo possono giungere a livelli di sofisticazione molto elevati, tali da esulare dalle finalità del presente testo. Questa impostazione è stata però recepita nei capitoli precedenti, anche se in misura ridotta. La grafica è stata usata infatti per mostrare lo schema del telaio al termine della fase di lettura dati, consentendo così l'individuazone di errori macroscopici. Il disegno del diagramma delle caratteristiche della sollecitazione non è stato previsto nel programma TELGEN, ma il modo in cui effettuarlo è stato descritto alla fine del quarto capitolo.

Una semplificazione ancora maggiore scaturisce però dal fatto che nelle usuali applicazioni professionali un modello generale risulta realmente necessario solo in rare occasioni. Gli edifici presentano infatti nella maggior parte dei casi telai a maglie rettangolari, che possono essere descritti in maniera più sintetica (assegnando luci delle campate ed altezze dei ritti anziché le coordinate dei nodi) e con un riferimento più diretto all'edificio (individuando i pilastri con la numerazione in carpenteria e indicando a quale piano appartengono le travi).

In queste situazioni sarebbe possibile realizzare una nuova procedura di calcolo, valida esclusivamente per lo schema semplificato. È però più conveniente continuare ad utilizzare il programma di calcolo generale, aggiungendo ad esso un preprocessor orientato al problema, che richieda solo le informazioni strettamente necessarie e generi da esse l'insieme più ampio dei valori di ingresso richiesti dal modello generale. Si deve ovviamente aggiungere anche un postprocessor, che presenti i risultati facendo riferimento alla numerazione usata in carpenteria per travi e pilastri.

Più in generale, si possono individuare più schemi ricorrenti nella pratica professionale. Un primo modello è il già citato telaio a maglie rettangolari. Un secondo, ricorrente nelle zone in cui le coperture sono realizzate

con tetti a falde inclinate, è il telaio a maglie pseudo–rettangolari (ritti verticali e traversi inclinati che formano maglie trapezie o, al limite, triangolari) la cui descrizione geometrica può essere efficacemente conseguita mediante la conoscenza delle luci delle campate e delle altezze di ciascuno ritto. Un'altra tipologia è quella del telaio con travi a ginocchio, o più in generale di uno schema nel quale la regolarità della maglia rettangolare è alterata dall'aggiunta di aste inclinate; la sua descrizione geometrica può essere effettuata partendo dalle semplici informazioni richieste per il telaio a maglie rettangolari e ricorrendo ad ulteriori indicazioni solo per le aste aggiunte. In tutti i modelli innanzi descritti i ritti del primo ordine possono essere incastrati al piede oppure collegati ad una trave di fondazione.

Per ciascuno degli schemi indicati si potranno realizzare adeguati pre e postprocessor. In questo capitolo viene innanzitutto esaminato brevemente il caso del telaio a maglie rettangolari completo, cioè senza aste mancanti, che costituisce la base comune a tutti i modelli citati. Si affronta poi in dettaglio lo schema di telaio a maglie rettangolari o trapezie (non triangolari), eventualmente con aste mancanti, accoppiato ad una trave di fondazione; il corrispondente programma operativo, completo di pre e postprocessor, è denominato TELRET. La metodologia di approccio al problema utilizzata per questi casi potrà servire al lettore per affrontare il problema delle aste aggiunte e gli altri possibili schemi che incontrerà nella sua attività.

2. Il telaio a maglie rettangolari

Nell'esaminare un telaio piano a maglie rettangolari (fig. 6.1) viene spontaneo distinguere le aste orizzontali (denominate travi) da quelle verticali (dette pilastri o ritti). Si usa il termine traverso per indicare l'insieme di travi poste ad una stessa quota; i traversi sono in genere individuati con numeri progressivi, dal basso verso l'alto. Per indicare l'insieme di ritti di un allineamento verticale si usa il termine pilastrata (o semplicemente pilastro); le pilastrate possono essere individuate con numeri consecutivi, da sinistra verso destra, oppure mediante la numerazione adottata in carpenteria per consentire un riferimento più diretto all'edificio cui il telaio appartiene.

Nello schema considerato i traversi sono supposti inestensibili e quindi i nodi posti ad una stessa quota sono obbligati ad avere lo stesso spostamento in orizzontale. I pilastri sono incastrati alla base del telaio e sono considerati estensionalmente deformabili; nel seguito si mostra però anche cosa cambia nel caso se ne volesse ipotizzare l'inestensibilità. Si suppone inoltre che lo schema sia completo, cioè senza aste mancanti.

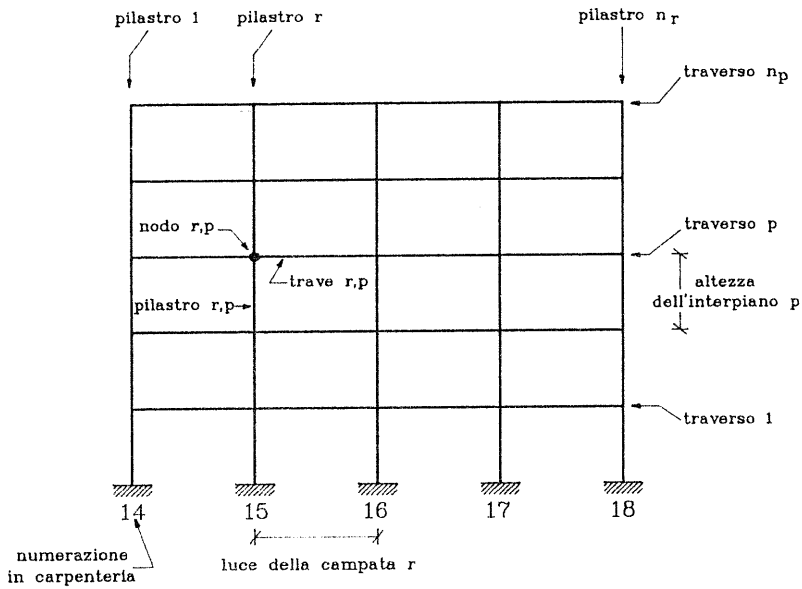


Fig. 6.1 — Telaio a maglie rettangolari

Mentre in un telaio generico i nodi sono individuati da un unico numero, la regolarità dello schema di telaio a maglie rettangolari rende preferibile identificare ciascun nodo mediante due indici, r e p, che rappresentano rispettivamente il numero d'ordine della pilastrata e del traverso cui esso appartiene. Anche le travi ed i pilastri (e le informazioni ad essi relative) vengono in questo caso individuati da due indici, corrispondenti a quelli del nodo che ne costituisce rispettivamente l'estremo sinistro o l'estremo superiore.

Lo schema geometrico è completamente definito se si conoscono il numero di pilastri n_r e di traversi n_p, le luci delle campate e le altezze degli interpiani. Sia queste che le altre informazioni necessarie (sezioni delle travi e dei pilastri, carichi verticali uniformemente distribuiti applicati sulle travi, forze orizzontali applicate ai traversi) sono descritte facendo riferimento agli indici r e p. L'interfaccia di ingresso consiste quindi in una procedura che legge i dati nella maniera tipica del telaio a maglie rettangolari e li trasforma in quella adottata per il telaio generico. Essa deve pertanto:

- individuare una corrispondenza tra i due criteri di numerazione, valida per i nodi e per le aste;
- definire le coordinate di ciascun nodo a partire dalla luce delle campate e dall'altezza degli interpiani;

- definire il codice di vincolo dei nodi;
- definire gli estremi ed il tipo di sezione di ogni asta;
- definire il numero di traversi inestensibili ed i nodi appartenenti a ciascuno di essi.

Nella figura 6.2 è mostrato il criterio prescelto per la numerazione degli elementi. I nodi vengono individuati con un numero progressivo, andando da sinistra verso destra e dal basso verso l'alto. Di conseguenza alla coppia di indici r e p corrisponde l'indice i che vale

$$i = p\, n_r + r \tag{6.1}$$

indicando convenzionalmente con $p = 0$ l'allineamento orizzontale alla base del telaio. Nel numerare le aste si conteggiano per prime le travi; per esse alla coppia di indici r e p corrisponde l'indice j che vale

$$j = (p-1)(n_r - 1) + r \tag{6.2}$$

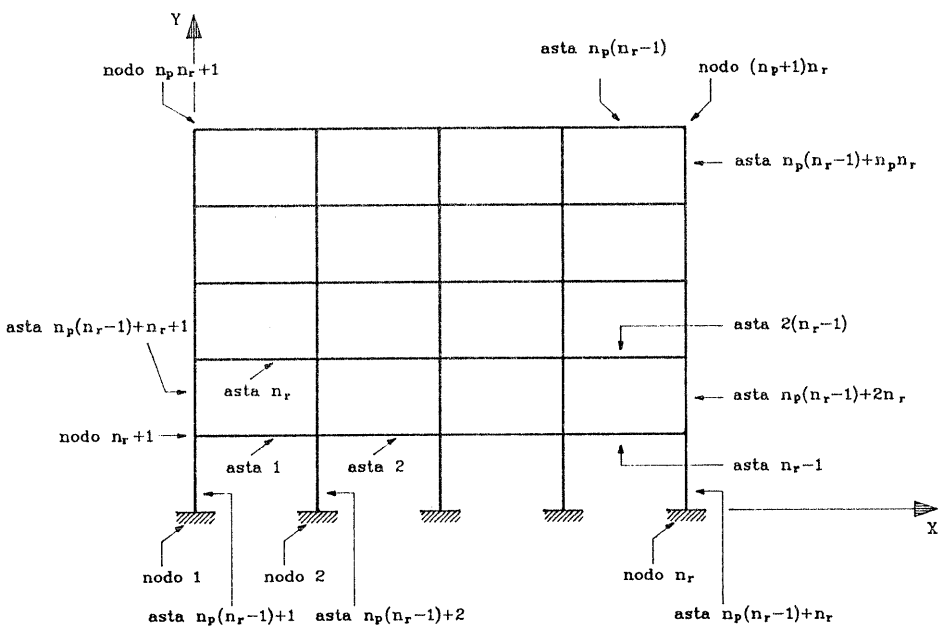


Fig. 6.2 — Telaio a maglie rettangolari: numerazione dei nodi e delle aste

Si conteggiano poi i pilastri, sempre procedendo da sinistra a destra e dal basso verso l'alto, facendo corrispondere agli indici r e p l'indice j che vale ora

$$j = n_p(n_r - 1) + (p - 1)n_r + r \tag{6.3}$$

L'origine del sistema di riferimento è assunta coincidente con il nodo 1. La generica pilastrata r è quindi posta ad una ascissa pari alla somma delle luci delle campate precedenti (da 1 ad $r - 1$). Analogamente, il traverso p è posto ad una ordinata pari alla somma delle altezze degli interpiani sottostanti (da 1 a p). Le coordinate del nodo i sono coincidenti con quelle del pilastro r e del traverso p cui esso appartiene. I primi n_r nodi, posti alla base del telaio, sono vincolati da un incastro; si deve quindi adottare per essi il codice di vincolo 111. Se i ritti sono deformabili estensionalmente i restanti nodi sono liberi ed hanno come codice di vincolo 000; se invece i ritti sono inestensibili lo spostamento verticale è impedito ed il codice dovrà essere 010.

Gli estremi di ogni asta sono definiti con facilità seguendo il criterio di corrispondenza della numerazione innanzi definito. Infine, i traversi orizzontali inestensibili previsti dal telaio generico sono ovviamente coincidenti con i traversi del telaio a maglie rettangolari ed i nodi ad essi appartenenti sono elencabili con immediatezza.

L'interfaccia di uscita consiste in una procedura che utilizza all'incontrario la corrispondenza tra i criteri di numerazione per presentare i risultati nell'ordine definito per il telaio a maglie rettangolari. Solo nel caso di ritti inestensibili è necessario effettuare una ulteriore elaborazione per ottenere gli sforzi normali nei pilastri. La presenza di vincoli allo spostamento verticale dei nodi, utilizzati per simulare l'indeformabilità estensionale, rende infatti nullo lo sforzo normale fornito dallo schema di telaio generico. Questa caratteristica di sollecitazione deve quindi essere calcolata per ciascun ritto come somma delle reazioni dei vincoli fittizi dei nodi posti al di sopra di esso.

3. Il telaio a maglie trapezie con trave di fondazione

Il telaio piano a maglie trapezie (fig. 6.3) differisce molto poco da quello a maglie rettangolari. In esso infatti sono ancora presenti pilastrate verticali e traversi continui ed inestensibili, ed è netta la distinzione tra pilastri e travi anche se queste ultime possono non essere più orizzontali. Rimangono quindi validi i criteri di numerazione definiti nel paragrafo precedente e le regole ivi indicate per il passaggio alla numerazione di nodi ed aste del telaio

generico. La non orizzontalità di un traverso implica però che le ordinate dei nodi ad esso appartenenti non siano più uguali, ma debbano essere calcolate singolarmente come somma delle altezze dei ritti sottostanti. Occorre inoltre ricordare che un'asta inestensibile costituisce per i nodi di estremità un vincolo mutuo allo spostamento nella direzione dell'asse dell'asta; se questa è inclinata, non è più corretto imporre che i nodi si spostino della stessa quantità in orizzontale. Tale ipotesi può però essere mantenuta in via approssimata, essendo ancora rigorosa se i ritti sono inestensibili ed in caso contrario vicina al vero quando l'inclinazione è modesta.

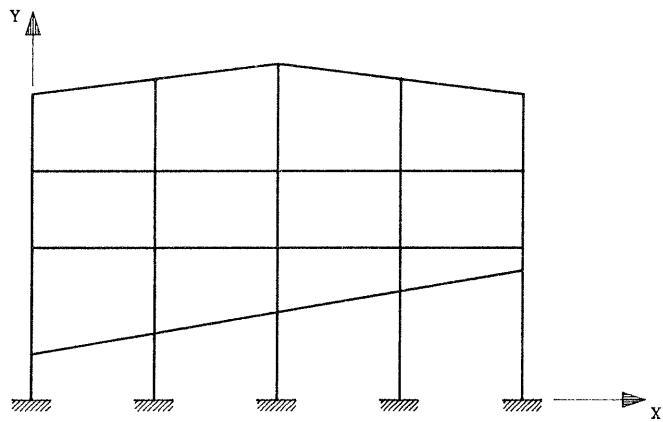


Fig. 6.3 — Telaio a maglie trapezie

L'aggiunta di una trave di fondazione dotata di sbalzi di estremità (fig. 6.4) comporta la presenza di due nodi ed alcune aste in più. I criteri di numerazione col doppio indice r,p sono ancora validi purché si indichi convenzionalmente con $p = 0$ il livello della trave di fondazione e con $r = 0$ ed $r = n_r + 1$ la posizione degli estremi dei suoi sbalzi. La corrispondenza tra questa coppia di indici e l'unico indice i richiesto per i nodi del telaio generico è fornita dalle relazioni

$$\begin{aligned} i &= r + 1 \qquad && \text{per} \quad p = 0 \\ i &= p\, n_r + r + 2 \qquad && \text{per} \quad p > 0 \end{aligned} \tag{6.4}$$

La corrispondenza con l'indice j richiesto per le aste del telaio generico

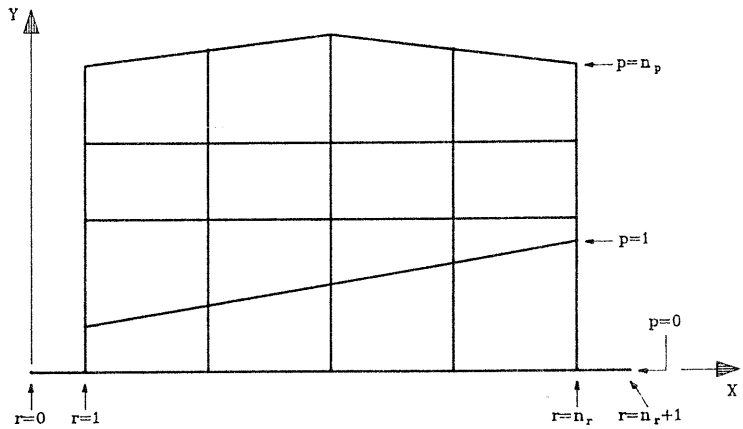


Fig. 6.4 — Telaio a maglie trapezie con trave di fondazione

è fornita dalle relazioni

$$j = r + 1$$

$$j = (p-1)(n_r - 1) + r + n_r + 1 \qquad (6.5)$$

$$j = n_p(n_r - 1) + p\, n_r + r + 1$$

valide rispettivamente per la trave di fondazione, le travi in elevazione ed i pilastri.

Nel programma descritto in questo capitolo la presenza di una trave di fondazione è considerata incompatibile con l'ipotesi di pilastri inestensibili. Nel caso di telai incastrati al piede l'indeformabilità estensionale equivale ad avere spostamenti verticali nodali nulli e se ne tiene quindi conto semplicemente con una opportuna scelta del codice di vincolo. A causa della trave di fondazione, invece, gli spostamenti verticali dei nodi di una pilastrata sono diversi da zero ed uguali al cedimento verticale della trave stessa e costituiscono quindi un'unica incognita; per affrontare correttamente il problema occorrerebbe pertanto modificare la fase di individuazione delle incognite, operando in maniera analoga a quanto fatto per imporre l'uguaglianza degli spostamenti orizzontali dei nodi di un traverso.

L'eventuale mancanza di alcune aste (fig. 6.5) non modifica i criteri di numerazione di nodi ed aste validi per lo schema completo, ma rende necessario riesaminare le modalità di corrispondenza con gli indici di nodo ed asta del telaio generico. Nel numerare i nodi, si potrebbero saltare

quelli non collegati ad alcuna asta, ma ciò invaliderebbe le relazioni (6.4). Includendoli nel conteggio ed assegnando ad essi il codice di vincolo 111 (incastro) si mantiene la validità delle relazioni senza aumentare il numero di incognite e quindi senza influire sui tempi di risoluzione. In maniera analoga sarebbe possibile conteggiare normalmente anche le aste mancanti, assegnando ad esse una sezione di area ed inerzia nulla. In questo caso il calcolatore sarebbe però costretto a valutare inutilmente le matrici di rigidezza di queste aste fittizie ed i tempi di elaborazione ne risentirebbero negativamente. Si preferisce quindi numerare progressivamente le sole aste reali (fig. 6.6), anche se ciò comporta la perdita di relazioni automatiche tra gli indici r, p e l'indice j.

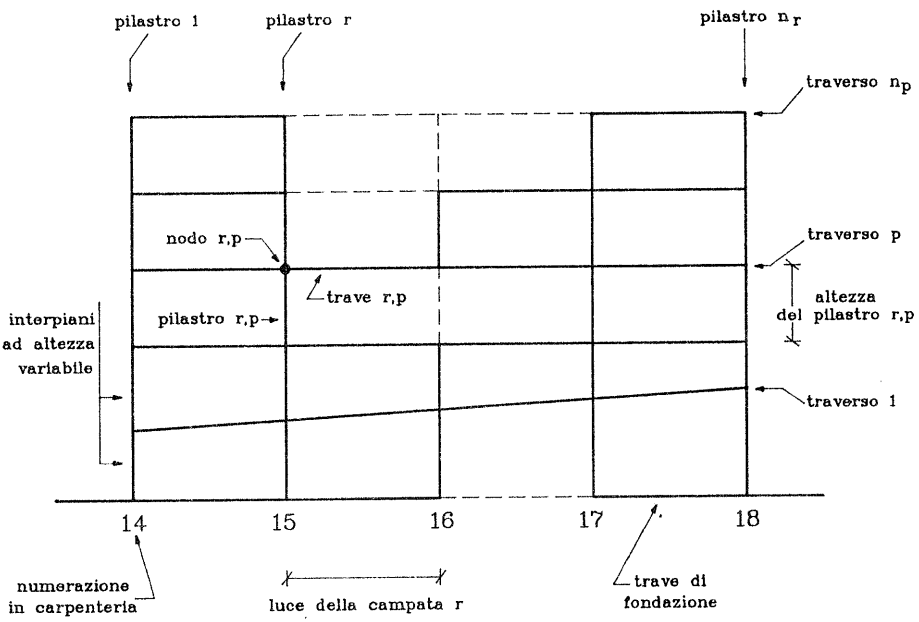


Fig. 6.5 — Telaio a maglie trapezie con trave di fondazione ed aste mancanti

L'assenza di un ritto interrompe la continuità della pilastrata. In conformità a quanto spiegato in precedenza, se si volesse ipotizzare l'indeformabilità estensionale delle aste occorrerebbe assumere come incognite uno spostamento verticale per ciascuno dei tratti in cui la pilastrata è suddivisa. Per non aumentare la complessità del programma che si presenta si è però esclusa tale possibilità, accettando quindi l'inestensibilità solo nel caso di schemi privi di travi di fondazione e con pilastrate non interrotte.

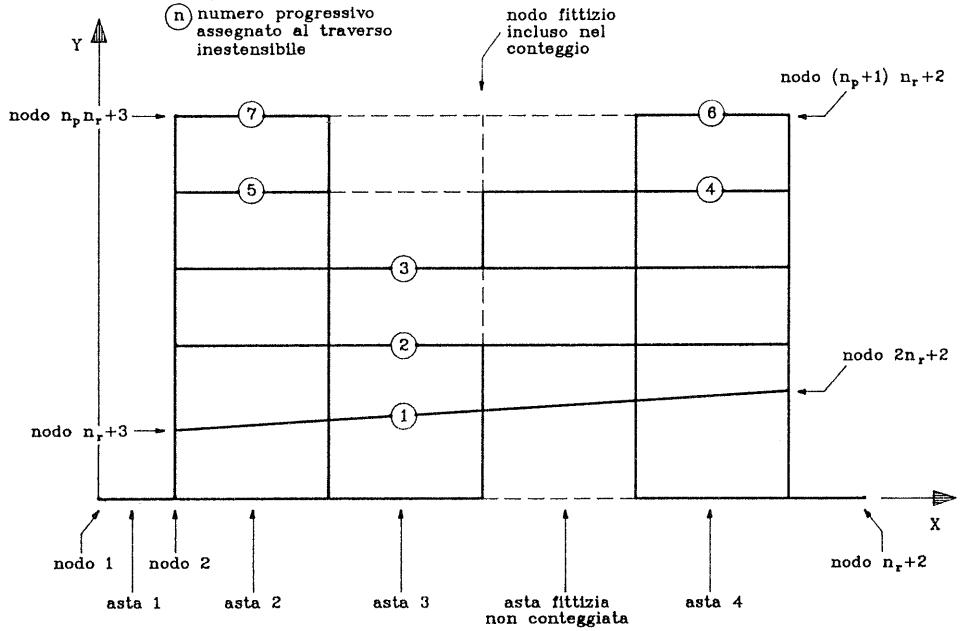


Fig. 6.6 — Telaio a maglie trapezie: numerazione dei nodi, delle aste e dei traversi inestensibili

Analogamente, l'assenza di una trave comporta l'interruzione della continuità del traverso. Una situazione del genere, per quanto rara, può verificarsi in un edificio ed è importante poterne tener conto. Occorre in questo caso distinguere tra la definizione puramente geometrica data per i traversi del telaio a maglie rettangolari o trapezie (linee di riferimento orizzontali o inclinate) e l'essenza fisica dei traversi inestensibili del telaio generico (vincoli allo spostamento mutuo dei nodi). Nella descrizione di questi ultimi diventa necessario tener presente la possibilità che un singolo traverso geometrico sia suddiviso in più traversi inestensibili distinti. La numerazione di questi procederà ancora dal basso verso l'alto e, a parità di livello, da destra verso sinistra (fig. 6.6).

Nel realizzare il programma si è presa in considerazione anche la possibilità di sostituire nello schema ad un'asta un pendolo inestensibile, ad esempio per imporre la continuità tra due traversi posti allo stesso livello, distinti nell'ambito del telaio piano ma collegati indirettamente mediante un impalcato. Si è quindi deciso di considerare convenzionalmente come pendolo inestensibile un'asta con sezione di area ed inerzia nulla. Essa viene considerata del tutto equivalente ad un'asta mancante, salvo che nell'imposizione dei vincoli mutui tra i nodi.

4. Descrizione del programma.

Il programma per la risoluzione di telai piani a maglie rettangolari o trapezie, eventualmente con trave di fondazione, è denominato TELRET. Esso è costituito da un programma principale, che evidenzia in maniera sintetica le operazioni da esso compiute, ed una serie di routine, le quali a loro volta richiamano altre routine o procedure, comprese quelle dei file MATRIX.BAS, SOLSIST.BAS, ASTA.BAS e PROCOM.BAS descritte nei capitoli precedenti. Il programma principale e le relative routine sono contenuti nel file TELRET.BAS. Le procedure caratteristiche di questo schema ma riutilizzate anche per l'insieme spaziale di telai stanno nel file PRORET.BAS.

Nel presente paragrafo si elencano tutti i blocchi logici che costituiscono il programma TELRET, ma si descrivono solo quelli aggiunti o modificati rispetto al programma TELGEN.

A) Programma principale.

Il programma principale è diviso anche in questo caso in tre parti: dichiarazione delle procedure richiamate, dimensionamento degli array statici, richiamo ai blocchi logici nei quali il procedimento di soluzione è stato scomposto. Quest'ultima in particolare mostra l'organizzazione complessiva del programma, sempre costituita da una prima fase (lettura, stampa dei dati geometrici, conversione degli stessi nel formato compatibile col nucleo di calcolo già adottato nel programma TELGEN, preparazione e triangolarizzazione della matrice di rigidezza della struttura) ed una seconda fase (lettura, stampa e conversione dei carichi, risoluzione del sistema, conversione e stampa dei risultati nel formato congruente coi dati forniti in input), ripetuta ciclicamente per consentire la risoluzione di più schemi di carico.

B) Dimensionamento degli array dinamici (routine `Dim...`).

C) Schermata di presentazione del programma (routine `Presentazione`).

D) Definizione delle unità di ingresso e uscita (procedura `DefinisceIO`).

E) Definizione delle modalità operative (procedura `DefinisceMO`).

Consente innanzitutto di definire se il calcolo deve essere eseguito tenendo conto o trascurando la deformabilità estensionale dei pilastri. Si ricorda che questa seconda possibilità è ammessa solo se non vi è trave di

fondazione e la continuità delle pilastrate non è interrotta dalla mancanza di ritti.

Consente inoltre di indicare se si vuole tener conto della deformabilità tagliante delle aste e se si vuole effettuare un controllo dei dati man mano che vengono letti.

F) Lettura dei dati geometrici ed elastici (routine `LeggeGeomRet`).

Utilizza la routine `InputIndRet` e le procedure `InputIntestaz`, `InputNumRitti`, `InputAltezze`, `InputLuci`, `InputModElast`, `InputSezioni`, `InputAsteRet` per leggere da un file su disco le informazioni geometriche generali e quelle specifiche relative a numerazione dei pilastri, altezze e luci, caratteristiche del materiale, tipi di sezione, aste. L'ordine con cui i dati vengono letti è descritto in dettaglio nel paragrafo 7. Utilizza inoltre le routine `DimGeomRet` e `DimTabSez` per dimensionare gli array dinamici in funzione del numero di elementi strettamente necessario per la struttura da calcolare. Nel caso si sia richiesta l'inestensibilità delle aste viene utilizzata la procedura `ContrRitti` per verificare che la continuità delle pilastrate non sia interrotta. Se tra i valori letti ne viene trovato qualcuno inaccettabile l'esecuzione del programma è interrotta al termine della fase di lettura.

G) Stampa dei dati geometrici ed elastici (routine `StampaGeomRet`).

Utilizza le procedure `OutputIntestaz`, `OutputPiani`, `OutputAsteRet`, `OutputTabSez`, `OutputModElast` per inviare all'unità di uscita un elenco di tutte le informazione geometriche ed elastiche lette nel blocco precedente.

H) Conversione dei dati geometrici (routine `ConvGeomRet`).

Calcola il numero di nodi e di aste in funzione del numero di pilastrate e di piani ed utilizza la procedura `ConverteGeomRet` per ricavare dai dati geometrici che descrivono in maniera sintetica il telaio a maglie rettangolari o trapezie l'insieme più generale di dati necessari per il nucleo di calcolo già adoperato per il programma `TELGEN`. Utilizza anche le routine `DimGeomGen`, `DimCarRet`, `DimCarGen` per dimensionare altri gruppi di array dinamici.

Più in dettaglio, la procedura `ConverteGeomRet` calcola le coordinate dei nodi, numerati in sequenza dal basso verso l'alto e da sinistra verso destra. Definisce le aste numerando prima le travi, in fondazione o in elevazione, e poi i pilastri, procedendo sempre dal basso verso l'alto e da sinistra verso destra ed escludendo le aste mancanti, cioè quelle cui si è assegnata una sezione di tipo 0 o con area ed inerzia nulla. Definisce poi i codici di vincolo. Assegna inizialmente agli estremi delle aste in elevazione

il codice 010 o 000, a seconda che il calcolo debba essere eseguito nell'ipotesi di aste inestensibili o deformabili estensionalmente, ed a quelli delle aste in fondazione il codice 100. Assegna quindi il codice 111, corrispondente ad un incastro perfetto, ai nodi di base del telaio (se non vi è trave di fondazione) ed a tutti i nodi che non sono collegati a nessuna asta. Procede infine alla individuazione dei traversi inestensibili e dei nodi ad essi appartenenti, procedendo dal basso verso l'alto e da destra verso sinistra. Per far ciò confronta il secondo estremo di ciascuna trave col primo dell'asta ad essa successiva nella numerazione del telaio generico: le aste appartengono ad uno stesso traverso se questi estremi coincidono in un unico nodo.

I) Disegno dello schema del telaio (routine `Schema`).

L) Individuazione delle incognite (routine `DefinisceIncognite`).

M) Preparazione della matrice di rigidezza della struttura (routine `MatRigStruttura`).

N) Riduzione della matrice di rigidezza ad una matrice triangolare (routine `Triangolarizza`).

O) Lettura dei carichi nodali e sulle aste (routine `LeggeCarRet`).

Utilizza la procedura `InputCarRet` per leggere dal file dati su disco le informazioni relative ai carichi applicati ai nodi, alle aste ed ai traversi dello schema. L'ordine con cui essi vengono letti è descritto in dettaglio nel paragrafo 7.

P) Stampa dei dati di carico (routine `StampaCarRet`).

Utilizza la procedura `OutputCarRet` per inviare all'unità di uscita le informazioni relative ai carichi lette nel blocco precedente.

Q) Conversione dei dati di carico (routine `ConvCarRet`).

Utilizza la procedura `ConverteCarRet` per ricavare dai dati di carico forniti per il telaio a maglie rettangolari o trapezie l'insieme di informazioni necessarie per il nucleo di calcolo già adoperato per il programma `TELGEN`. In particolare, le forze orizzontali agenti sui traversi inestensibili vengono considerate applicate al primo nodo di ciascun traverso.

R) Calcolo del vettore carichi nodali (routine `TerminiNoti`).

S) Calcolo del vettore soluzione (routine `Risolvi`).

T) Calcolo delle caratteristiche di sollecitazione agli estremi delle aste e delle azioni sui vincoli (routine `CaratSollec`).

U) Conversione dei risultati (routine `ConvRisultRet`).

Utilizza la procedura `ConverteRisRet` per riorganizzare i risultati in una forma congruente con quella adoperata in ingresso. In particolare gli spostamenti dei traversi vengono determinati uguagliandoli alla traslazione secondo l'asse x del loro primo nodo, individuato mediante l'analisi del vettore `NI()`. Le caratteristiche di sollecitazione vengono presentate riordinando ciascuna asta in funzione degli indici r e p della pilastrata e del piano. Se il calcolo è stato eseguito trascurando la deformabilità estensionale delle aste, cioè aggiungendo vincoli allo spostamento verticale dei nodi, lo sforzo normale in ciscun ritto è valutato come somma delle reazioni dei vincoli dei nodi posti al di sopra di esso.

V) Stampa dei risultati (routine `StampaRisultRet`).

Utilizza la procedura `OutputRisRet` per inviare all'unità di uscita i risultati della risoluzione del singolo schema di carico. Questa è suddivisa nelle routine `Spostamenti`, `RisTravi`, `RisPilastri` e `RisFond` che forniscono componenti di movimento e caratteristiche di sollecitazione e nella routine `Squilibrio` che determina l'eventuale squilibrio dei nodi o dei traversi conseguente agli errori di arrotondamento del calcolo.

Utilizza inoltre la routine `OutputIpotesiRet` per indicare le ipotesi di base del calcolo, evidenziando se esso è stato eseguito tenendo conto della deformabilità estensionale e di quella a taglio delle aste.

5. Elenco delle variabili

Le variabili sono riunite in gruppi in base al loro significato ed al blocco di programma nel quale sono utilizzate prevalentemente o per la prima volta. Si fa riferimento solo ai blocchi aggiunti o modificati rispetto al programma TELGEN. Nell'ambito di ciascun gruppo le variabili sono elencate in ordine alfabetico. Per tutte le grandezze dimensionali sono riportate, tra parentesi, le unità di misura adottate.

Variabili utilizzate nella gestione generale del programma

`AltVar`	indica se vi è un interpiano con ritti ad altezza variabile
`Contr`	indica se i dati devono essere stampati man mano che vengono letti per consentirne un controllo immediato
`DefEst`	indica se il calcolo è condotto tenendo conto della deformabilità estensionale delle aste
`DefTag`	indica se il calcolo è condotto tenendo conto della deformabilità a taglio delle aste
`ER`	indica un errore nei dati letti
`ERpil`	indica la presenza di ritti che interrompono la continuità verticale, incompatibili con l'ipotesi di pilastri inestensibili
`Squil`	indica la presenza di squilibrio nei nodi dopo la risoluzione del sistema
`Vced`	indica la presenza di vincoli elasticamente cedevoli (questa variabile è inserita solo per compatibilità con il programma `TELGEN`)
`Wnkl`	indica la presenza di una trave di fondazione su suolo elastico alla Winkler
	queste variabili di controllo possono assumere solo due valori convenzionali (0=no, 1=si)
`CAR`	numero d'ordine dello schema di carico in esame
`FI$`	nome del file che contiene i dati
`FO$`	nome dell'unità cui si vuole inviare l'output (`CON`=schermo, `PRN`=stampante, oppure nome di un file)
`FT$`	nome della directory da utilizzare per file temporanei (questa variabile è inserita solo per compatibilità con il programma `TELSPA`)

Variabili utilizzate come indici degli array

`F`	indica un generico tratto della trave di fondazione; il valore `0` indica lo sbalzo sinistro, il valore `RZ` lo sbalzo destro
`I`	indica un nodo generico, oppure un generico elemento dell'array `NI()`
`J`	indica un'asta generica
`K`	indica un generico traverso orizzontale inestensibile

L	indica un generico tipo di sezione
P	se riferito a un nodo o una trave indica il generico piano o orizzontamento, cioè l'allineamento orizzontale o inclinato (indipendentemente dalla eventuale mancanza di continuità del traverso); i piani sono numerati dal basso verso l'alto; se riferito a un ritto indica il generico interpiano, sottostante l'orizzontamento P
R	se riferito a un ritto indica il numero d'ordine del pilastro, inteso come allineamento verticale; i pilastri sono numerati in sequenza, nel verso positivo dell'asse x (da sinistra verso destra); se riferito a una trave indica il numero d'ordine della campata e coincide col numero d'ordine del pilastro posto alla sinistra di essa

Variabili utilizzate nella definizione dello schema geometrico e delle caratteristiche elastiche

BF(F)	larghezza della sottobase del tratto F della trave di fondazione (m)
DL(R)	luce della campata R (m)
E	modulo di elasticità normale E (kN m^{-2})
EL	modulo di elasticità normale E (N mm^{-2})
FS$(L)	codice per la definizione della forma della sezione; il carattere può assumere convenzionalmente solo uno di questi valori: R = sezione di forma rettangolare G = sezione di forma generica
G	modulo di elasticità tangenziale G (kN m^{-2})
GL	modulo di elasticità tangenziale G (N mm^{-2})
H(R,P)	altezza del ritto R al piano P (m)
HI(P)	altezza dell'interpiano P (m); se in esso vi sono ritti di differente altezza, si assegna HI(P)=0
IT$	intestazione del calcolo
KF(F)	costante di sottofondo del suolo alla Winkler posto sotto il tratto F della trave di fondazione (N cm^{-3})
LZ	numero totale di tipi di sezione
NB(R,P)	tipo di sezione per la trave R del piano P; si può assegnare il valore 0 per indicare che la trave è mancante

`NF(F)` tipo di sezione per il tratto `F` della trave di fondazione; si può assegnare il valore `0` per indicare che il tratto è mancante

`NP(R)` numerazione adottata in carpenteria per il pilastro `R`

`NR(R,P)` tipo di sezione per il ritto `R` del piano `P`; si può assegnare il valore `0` per indicare che il ritto è mancante

`PZ` numero totale di piani

`R1` numero d'ordine della prima campata (`1` se non vi sono travi di fondazione, `0` in caso contrario per includere lo sbalzo sinistro)

`R2` numero d'ordine dell'ultima campata (`RZ-1` se non vi sono travi di fondazione, `RZ` in caso contrario per includere lo sbalzo destro)

`RV` variabile ausiliaria usata nella stampa delle altezze dei ritti; indica quanti valori vanno stampati su una stessa riga

`RZ` numero totale di pilastri (intesi come allineamenti verticali, non come singole aste)

`ZA(L)` area della sezione `L` (m^2)

`ZAT(L)` area a taglio, cioè rapporto tra area e fattore di taglio, della sezione `L` (m^2)

`ZB(L)` base della sezione `L` (m)

`ZH(L)` altezza della sezione `L` (m)

`ZI(L)` momento d'inerzia della sezione `L` (m^4)

Variabili utilizzate nella conversione dei dati geometrici

`B(J)` larghezza della sottobase dell'asta `J`, se questa è un'asta su suolo elastico alla Winkler (m)

`E1` numero d'ordine del nodo che costituisce il primo estremo di un'asta

`E1(J)` numero d'ordine del nodo che costituisce il primo estremo dell'asta `J`

`E2` numero d'ordine del nodo che costituisce il secondo estremo di un'asta

`E2(J)` numero d'ordine del nodo che costituisce il secondo estremo dell'asta `J`

`EP` numero d'ordine dell'ultimo nodo esaminato appartenente a un traverso

`Iinf` numero d'ordine del nodo immediatamente sottostante il nodo `I`

IZ	numero totale di nodi
JF	numero di aste che costituiscono la trave di fondazione
JZ	massimo numero di aste possibile; poi effettivo numero totale di aste
K(J)	costante di sottofondo dell'asta J, se questa è un'asta su suolo elastico alla Winkler (N cm^{-3})
KV()	array bidimensionale che contiene le rigidezze dei vincoli elasticamente cedevoli (questo array è inserito solo per compatibilità col programma TELGEN)
KZ	massimo numero di traversi possibile; poi effettivo numero totale di traversi inestensibili
NI()	array monodimensionale che contiene la numerazione dei nodi dei traversi orizzontali inestensibili (l'elenco dei nodi di ciascun traverso termina con un valore nullo)
NT(J)	tipo di sezione dell'asta J
TA$(J)	codice per la definizione del tipo di asta; il carattere può assumere convenzionalmente solo uno di questi valori: T = trave alla De Saint Venant solidale ai nodi di estremità W = trave su suolo elastico alla Winkler
V$(I)	codice di vincolo del nodo I la stringa è costituita da tre caratteri, uno per ciascuna componente di movimento; il carattere può assumere convenzionalmente solo uno di questi valori: 0 = componente di movimento consentita 1 = componente di movimento impedita C = componente di movimento soggetta ad un vincolo elastico (non prevista nel programma TELRET)
X(I)	ascissa del nodo I (m)
Y(I)	ordinata del nodo I (m)

Variabili utilizzate nella definizione dei carichi

IM	indica l'esistenza di momenti concentrati nei nodi
IN	indica l'esistenza di forze verticali concentrati nei nodi
IO	indica l'esistenza di carichi orizzontali applicati ai traversi
IV	indica l'esistenza di carichi verticali sulle travi

queste quattro variabili possono assumere solo due valori convenzionali (0=non esistono, 1=esistono carichi di quel tipo)

`F(I,K)` azione concentrata `K` applicata al nodo `I`; `K` varia da 1 a 3 per indicare nell'ordine le azioni F_x, F_y e M_z (kN per `K=1` e 2, kNm per `K=3`)

`FX(K)` forza orizzontale applicata al traverso `K` (kN)

`FYF(F)` forza verticale applicata al nodo `F` della trave di fondazione (kN)

`FYN(R,P)` forza verticale applicata al nodo posto nel pilastro `R`, piano `P` (kN)

`MN(R,P)` momento concentrato applicato al nodo posto nel pilastro `R`, piano `P` (kNm)

`MNF(F)` momento concentrato applicato al nodo `F` della trave di fondazione (kNm)

`P(J,K)` componente `K` del carico uniformemente distribuito sull'asta `J`; `K` varia in questo caso da 1 a 2 per indicare nell'ordine le componenti p_x e p_y (kN m^{-1})

`PV` variabile ausiliaria usata nella stampa dei carichi; indica quanti valori vanno stampati su una stessa riga

`P1` indica qual è il primo orizzontamento del telaio (0 se vi è la trave di fondazione, 1 in caso contrario)

`Q(R,P)` carico verticale uniformemente distribuito sulla trave `R` del piano `P` (kN m^{-1})

`QF(F)` carico verticale uniformemente distribuito sulla campata `F` della trave di fondazione (kN m^{-1})

Variabili utilizzate nella conversione e stampa dei risultati

`D(K)` spostamento orizzontale del traverso `K` (m)

`F` variabile ausiliaria utilizzata durante la stampa dei risultati per calcolare la somma delle azioni orizzontali sui nodi di un traverso ed individuare così la presenza di un eventuale squilibrio (kN o kNm)

`F()` dopo la risoluzione del sistema viene memorizzata in questo array, già descritto in precedenza, la somma di azioni nodali esterne ed azioni trasmesse ai nodi dalle aste, pari quindi alle azioni sui vincoli esterni e mutui ed all'eventuale squilibrio nodale conseguente agli errori di troncamento

`M1(J)` momento flettente al primo estremo dell'asta `J` (kNm)

`M2(J)` momento flettente al secondo estremo dell'asta `J` (kNm)

`MD(R,P)`	momento flettente all'estremo destro della trave R, piano P (kNm)
`MFD(F)`	momento flettente all'estremo destro del tratto F della trave di fondazione (kNm)
`MFS(F)`	momento flettente all'estremo sinistro del tratto F della trave di fondazione (kNm)
`MP(R,P)`	momento flettente al piede del ritto R, ordine P (kNm)
`MS(R,P)`	momento flettente all'estremo sinistro della trave R, piano P (kNm)
`MT(R,P)`	momento flettente in testa al ritto R, ordine P (kNm)
`N1(J)`	sforzo normale al primo estremo dell'asta J (kN)
`N(I,K)`	elemento K del vettore di corrispondenza del nodo I, cioè numero d'ordine dell'incognita associata alla componente di movimento K del nodo I (un valore nullo indica che la componente di movimento è impedita da un vincolo)
`R()`	dopo la risoluzione del sistema contiene il vettore spostamenti nodali **r**
`ROT`	rotazione di un nodo
`SN(R,P)`	sforzo normale nel ritto R, ordine P (kN)
`T1(J)`	taglio al primo estremo dell'asta J (kN)
`T2(J)`	taglio al secondo estremo dell'asta J (kN)
`TD(R,P)`	taglio all'estremo destro della trave R, piano P (kN)
`TFD(F)`	taglio all'estremo destro del tratto F della trave di fondazione (kN)
`TFS(F)`	taglio all'estremo sinistro del tratto F della trave di fondazione (kN)
`TR(R,P)`	taglio nel ritto R, ordine P (kN)
`TS(R,P)`	taglio all'estremo sinistro della trave R, piano P (kN)
`V`	variabile ausiliaria usata durante la stampa dei risultati per indicare se almeno una componente di movimento relativa a un nodo generico è libera e ad essa corrisponde un'azione squilibrata
`V(K)`	variabile ausiliaria usata durante la stampa dei risultati per indicare se la componente di movimento K relativa a un nodo generico è libera e ad essa corrisponde un'azione squilibrata
`VY`	spostamento verticale di un nodo

6. Codifica

```
'                                  TELRET.BAS
'
'  programma per il calcolo di strutture piane intelaiate con maglie di forma
'                              rettangolare o trapezia
'
'  tipi di aste:          travi alla De Saint Venant, incastrate ai nodi
'                         travi di fondazione su suolo elastico alla Winkler
'
'  vincoli esterni:       vincoli di incastro perfetto alla base
'                         vincoli mutui allo spostamento orizzontale (traversi)
'
'  carichi:               carichi uniformemente distribuiti sulle travi
'                         momenti concentrati sui nodi
'                         forze verticali concentrate sui nodi
'                         forze orizzontali applicate ai traversi
'
'                                                                  rev. 01.90

' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file SOLSIST.BAS
DECLARE SUB Triang.K2 (KTT!(), KTA!(), KAA!(), SB!, RID!)
DECLARE SUB Risolve.K2 (KTT!(), KTA!(), KAA!(), SB!, RID!, B!())

' procedure del file PROCOM.BAS
DECLARE SUB Aspetta (Sec!)
DECLARE SUB CarSoll (JZ!, X!(), Y!(), ZA!(), ZAT!(), ZI!(), E!, G!, E1!(), E2!(),
                     TA$(), NT!(), K!(), B!(), N!(), P!(), F!(), R!(), N1!(), T1!(),
                     M1!(), N2!(), T2!(), M2!())
DECLARE SUB DefinisceIO (Prog$, FI$, FO$, FT$)
DECLARE SUB DefinisceMO (Prog$, DefEst!, DefTag!, Contr!)
DECLARE SUB DisegnaSchema (IZ!, JZ!, X!(), Y!(), E1!(), E2!())
DECLARE SUB FormaMat (E1!, E2!, N!(), Kg!(), NL!, SB!, KTT!(), KTA!(), KAA!())
DECLARE SUB Incognite (IZ!, JZ!, KZ!, E1!(), E2!(), V$(), NI!(), N!(), NL!, NZ!,SB!)
DECLARE SUB InputIntestaz (Contr!, Intest$)
DECLARE SUB InputModElast (Contr!, EL!, GL!, E!, G!)
DECLARE SUB InputSezioni (Contr!, LZ!, FS$(), ZB!(), ZH!(), ZA!(), ZI!(),ZAT!(),ER!)
DECLARE SUB MatAstaGlobale (TA$, X1!, X2!, Y1!, Y2!, A!, AT!, Iner!, K!, B!, E!, G!,
                            Lun!, SnAl!, CsAl!, Ld!(), Kg!())
DECLARE SUB OutputIntestaz (FI$, Intest$)
DECLARE SUB OutputModElast (DefTag!, EL!, GL!)
DECLARE SUB TermNoto (IZ!, JZ!, X!(), Y!(), ZA!(), ZI!(), E!, E1!(), E2!(), TA$(),
                      NT!(), K!(), B!(), N!(), P!(), F!(), R!())
DECLARE SUB VincoliCed (IZ!, N!(), KV!(), NL!, SB!, KTT!(), KTA!(), KAA!())

' procedure del file PRORET.BAS
DECLARE SUB ContrRitti (PZ!, RZ!, NP!(), NB!(), NR!(), ZA!(), ZI!(), ER!)
DECLARE SUB ConverteCarRet (RZ!, PZ!, Wnkl!, NB!(), NF!(), ZA!(), ZI!(), IV!, IM!,
                            IN!, IO!, Q!(), QF!(), MN!(), MNF!(), FYN!(), FYF!(),
                            FX!(), KZ!, V$(), NI!(), P!(), F!())
DECLARE SUB ConverteGeomRet (PZ!, RZ!, Wnkl!, DefEst!, DL!(), H!(), NB!(), NR!(),
                             NF!(), BF!(), KF!(), ZA!(), ZI!(), IZ!, JZ!, KZ!, X!(),
                             Y!(), V$(), NI!(), E1!(), E2!(), TA$(),NT!(),B!(),K!())
DECLARE SUB ConverteRisRet (RZ!, PZ!, Wnkl!, DefEst!, NB!(), NR!(), NF!(), ZA!(),
```

```
                          ZI!(), D!(), MS!(), MD!(), TS!(), TD!(), MP!(), MT!(),
                          TR!(), SN!(), MFS!(), MFD!(), TFS!(), TFD!(), KZ!,NI!(),
                          N!(), R!(), F!(), M1!(), M2!(), T1!(), T2!(), N1!())
DECLARE SUB InputAltezze (Contr!, PZ!, RZ!, AltVar!, HI!(), H!())
DECLARE SUB InputAsteRet (Contr!, Wnkl!, PZ!, RZ!, LZ!, NP!(), NB!(), NR!(), NF!(),
                          BF!(), KF!(), ER!)
DECLARE SUB InputCarRet (Contr!, Wnkl!, IV!, IM!, IN!, IO!, PZ!, RZ!, KZ!, NP!(),
                         V$(), NB!(), NF!(), ZA!(), ZI!(), Q!(), QF!(), MN!(),
                         MNF!(), FYN!(), FYF!(), FX!())
DECLARE SUB InputLuci (Contr!, Wnkl!, RZ!, DL!())
DECLARE SUB InputNumRitti (Contr!, RZ!, NP!())
DECLARE SUB OutputAsteRet (PZ!, RZ!, Wnkl!, NP!(), DL!(), NB!(), NR!(), NF!(),
                           BF!(), KF!(), FS$(), ZB!(), ZH!(), ZI!())
DECLARE SUB OutputCarRet (CAR!, Wnkl!, IV!, IM!, IN!, IO!, PZ!, RZ!, KZ!, NP!(),
                          V$(), NB!(), NF!(), ZA!(), ZI!(), Q!(), QF!(), MN!(),
                          MNF!(), FYN!(), FYF!(), FX!())
DECLARE SUB OutputPiani (AltVar!, PZ!, RZ!, NP!(), HI!(), H!())
DECLARE SUB OutputRisRet (RZ!, PZ!, Wnkl!, NP!(), NB!(), NR!(), NF!(), ZA!(), ZI!(),
                          D!(), MS!(), MD!(), TS!(), TD!(), MP!(), MT!(), TR!(),
                          SN!(), MFS!(), MFD!(), TFS!(), TFD!(), IZ!, KZ!, NL!,
                          V$(), N!(),R!(),F!())
DECLARE SUB OutputTabSez (DefTag!, LZ!, FS$(), ZB!(), ZH!(), ZA!(), ZAT!(), ZI!())


' ------------------ Dimensionamento delle variabili con indice --------------------

OPTION BASE 1             ' definisce l'estremo inferiore dell'indice degli array
DIM Ld(6, 6), Kg(6, 6)    ' array statici
                          ' array dinamici: sono dimensionati in apposite routine


  --------------------------- Programma principale --------------------------------

GOSUB Presentazione
CALL DefinisceIO("RET", FI$, FO$, FT$)
CALL DefinisceMD("RET", DefEst, DefTag, Contr)
OPEN "I", #1, FI$
OPEN "O", #2, FO$
GOSUB LeggeGeomRet
GOSUB StampaGeomRet
GOSUB ConvGeomRet
GOSUB Schema
GOSUB DefinisceIncognite
GOSUB MatRigStruttura
GOSUB Triangolarizza
CAR = 0
DO WHILE NOT EOF(1)
   CAR = CAR + 1
   GOSUB LeggeCarRet
   GOSUB StampaCarRet
   GOSUB ConvCarRet
   GOSUB TerminiNoti
   GOSUB Risolvi
   GOSUB CaratSollec
   GOSUB ConvRisultRet
   GOSUB StampaRisultRet
LOOP
CLOSE #1
CLOSE #2
```

```
PRINT
PRINT "elaborazione terminata"
PRINT
END


' ------------------------------ DimXXXXXX ------------------------------------
'  Gruppo di routine per il dimensionamento degli array dinamici
' -----------------------------------------------------------------------------

' ................................ DimCarGen .....................................
'  Routine per il dimensionamento degli array relativi a carichi e risultati
'
DimCarGen:
   DIM F(IZ, 3), P(JZ, 2)
   DIM M1(JZ), M2(JZ), T1(JZ), T2(JZ), N1(JZ), N2(JZ)
RETURN

' ................................ DimCarRet .....................................
'  Routine per il dimensionamento degli array relativi a carichi e risultati
'
DimCarRet:
   DIM Q(RZ - 1, PZ), MN(RZ, PZ), FYN(RZ, PZ), FX(KZ)
   DIM D(KZ), MD(RZ - 1, PZ), MS(RZ - 1, PZ), TD(RZ - 1, PZ), TS(RZ - 1, PZ)
   DIM MP(RZ, PZ), MT(RZ, PZ), TR(RZ, PZ), SN(RZ, PZ)
   IF Wnkl = 1 THEN
      DIM QF(0 TO RZ), MNF(0 TO RZ + 1), FYF(0 TO RZ + 1)
      DIM MFD(0 TO RZ), MFS(0 TO RZ), TFD(0 TO RZ), TFS(0 TO RZ)
   END IF
RETURN

' ................................ DimGeomGen ....................................
'  Routine per il dimensionamento degli array relativi alla geometria
'
DimGeomGen:
   DIM X(IZ), Y(IZ), V$(IZ), KV(1, 1), NI(IZ + KZ)
   DIM E1(JZ), E2(JZ), TA$(JZ), NT(JZ), B(JZ), K(JZ)
   DIM N(IZ, 3)
RETURN

' ................................ DimGeomRet ....................................
'  Routine per il dimensionamento degli array relativi alla geometria
'
DimGeomRet:
   DIM NP(RZ), HI(PZ), H(RZ, PZ), DL(0 TO RZ)
   DIM NB(RZ - 1, PZ), NR(RZ, PZ)
   IF Wnkl = 1 THEN DIM NF(0 TO RZ), BF(0 TO RZ), KF(0 TO RZ)
RETURN

' ................................ DimMatRig .....................................
'  Routine per il dimensionamento di matrice di rigidezza e vettore carico
'
DimMatRig:
   IF NL = 0 THEN
      REDIM KTT(0 TO 0, 0 TO 0), KTA(0 TO 0, NZ - NL)
   ELSE
      REDIM KTT(NL, NL), KTA(NL, NZ - NL)
   END IF
   DIM KAA(NZ - NL, SB), R(NZ)
```

```
RETURN

' ..... DimTabSez .....
' Routine per il dimensionamento degli array relativi ai tipi di sezione
'
DimTabSez:
   DIM FS$(LZ), ZB(LZ), ZH(LZ), ZA(LZ), ZI(LZ), ZAT(LZ)
RETURN


' ------------------------------ Presentazione ----------------------------------
' Routine per la visualizzazione della schermata di presentazione del
' programma
' -------------------------------------------------------------------------------
Presentazione:
   CLS
   PRINT TAB(31); "TELRET - rev. 01.90"
   PRINT
   PRINT "       Le ipotesi di calcolo e le modalita' operative del programma"
   PRINT "       TELRET sono riportate nella terza parte, capitolo 6, del libro:"
   PRINT "       A.Ghersi, R.Coraggio, Il personal computer nel calcolo di edifici."
   PRINT "       Analisi matriciale di strutture intelaiate, CUEN, Napoli, 1990."
   PRINT
   PRINT
   PRINT
RETURN


' ------------------------------- Completata ------------------------------------
' Routine per l'indicazione della fine di una fase
' -------------------------------------------------------------------------------
Completata:
   LOCATE CSRLIN - 1, 44
   PRINT "completata            "
RETURN


' ------------------------------- InCorso ---------------------------------------
' Routine per l'indicazione dell'inizio di una fase
' -------------------------------------------------------------------------------
InCorso:
   LOCATE CSRLIN - 1, 44
   PRINT "in corso"
RETURN


' ------------------------------ NomeProgramma ----------------------------------
' Routine per la visualizzazione dell'intestazione del programma
' -------------------------------------------------------------------------------
NomeProgramma:
   CLS
   PRINT "TELRET - rev. 01.90"
   PRINT
RETURN


' ------------------------------ LeggeGeomRet -----------------------------------
' Routine per la lettura dei dati geometrici ed elastici
' -------------------------------------------------------------------------------
```

```
LeggeGeomRet:
   GOSUB NomeProgramma
   PRINT "lettura dati geometrici": GOSUB InCorso

   ER = 0
   CALL InputIntestaz(Contr, IT$)
   GOSUB InputIndRet
   GOSUB DimGeomRet
   GOSUB DimTabSez
   CALL InputNumRitti(Contr, RZ, NP())
   CALL InputAltezze(Contr, PZ, RZ, AltVar, HI(), H())
   CALL InputLuci(Contr, Wnkl, RZ, DL())
   CALL InputModElast(Contr, EL, GL, E, G)
   CALL InputSezioni(Contr, LZ, FS$(), ZB(), ZH(), ZA(), ZI(), ZAT(), ER)
   CALL InputAsteRet(Contr, Wnkl, PZ, RZ, LZ, NP(), NB(), NR(), NF(), BF(), KF(),ER)
   IF ER = 1 THEN END
   IF DefEst = 0 THEN CALL ContrRitti(PZ, RZ, NP(), NB(), NR(), ZA(), ZI(), ER)
   IF ER = 1 THEN END

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN


' ..... InputIndRet .....
' Routine per la lettura delle indicazioni generali
'
InputIndRet:
   INPUT #1, RZ, PZ, LZ, Wnkl
   IF Contr = 1 THEN
      PRINT #2, "numero di pilastri: "; RZ
      PRINT #2, "numero di piani: "; PZ
      PRINT #2, "numero di tipi di sezione: "; LZ
      IF Wnkl = 1 THEN PRINT #2, "e' presente la trave di fondazione"
   END IF
   IF DefEst = 0 AND Wnkl = 1 THEN
      PRINT "Esecuzione interrotta:  il programma non consente l'ipotesi"
      PRINT "di aste inestensibili quando esiste una trave di fondazione"
      END
   END IF
RETURN


' ---------- StampaGeomRet ----------
' Routine per la stampa di dati geometrici ed elastici
' -----------------------------------------------------------------------------
StampaGeomRet:
   PRINT "stampa dati geometrici"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   CALL OutputIntestaz(FI$, IT$)
   CALL OutputPiani(AltVar, PZ, RZ, NP(), HI(), H())
   CALL OutputAsteRet(PZ, RZ, Wnkl, NP(), DL(), NB(), NR(), NF(), BF(), KF(), FS$(),
                      ZB(), ZH(), ZI())
   CALL OutputTabSez(DefTag, LZ, FS$(), ZB(), ZH(), ZA(), ZAT(), ZI())
   CALL OutputModElast(DefTag, EL, GL)

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
```

```
      GOSUB Completata
   END IF
RETURN


' ------------------------------ ConvGeomRet ------------------------------------
'   Routine per la conversione dei dati geometrici per telai a maglie
'   rettangolari
' -------------------------------------------------------------------------------
ConvGeomRet:
   PRINT "conversione dati geometrici": GOSUB InCorso

   IZ = RZ * (PZ + 1) + 2
   JZ = (2 * RZ - 1) * PZ + RZ + 1      ' massimo numero di aste possibile
   KZ = INT(RZ * PZ / 2)                ' massimo numero di traversi possibile
   GOSUB DimGeomGen
   CALL ConverteGeomRet(PZ, RZ, Wnkl, DefEst, DL(), H(), NB(), NR(), NF(), BF(),
                        KF(), ZA(), ZI(), IZ, JZ, KZ, X(), Y(), V$(), NI(), E1(),
                        E2(), TA$(), NT(), B(), K())
   VCed = 0
   GOSUB DimCarRet
   GOSUB DimCarGen

   GOSUB Completata
RETURN


' ------------------------------ Schema -----------------------------------------
'  Routine che visualizza lo schema geometrico
' -------------------------------------------------------------------------------
Schema:
   CALL DisegnaSchema(IZ, JZ, X(), Y(), E1(), E2())
   GOSUB NomeProgramma
RETURN


' ------------------------------ DefinisceIncognite -----------------------------
'  Routine per la numerazione delle incognite  e la valutazione della semibanda
'  (gli array che costituiscono la matrice di rigidezza vengono dimensionati)
' -------------------------------------------------------------------------------
DefinisceIncognite:
   PRINT "numerazione incognite": GOSUB InCorso

   CALL Incognite(IZ, JZ, KZ, E1(), E2(), V$(), NI(), N(), NL, NZ, SB)
   GOSUB DimMatRig

   GOSUB Completata
   PRINT "- numero totale di incognite"; NZ;
   IF NL <> 0 THEN PRINT " (tra cui"; NL; "spostamenti di traversi)";
   PRINT
   PRINT "- ampiezza della semibanda  "; SB
RETURN


' ------------------------------ MatRigStruttura --------------------------------
'  Routine per la costruzione della matrice di rigidezza della struttura
' -------------------------------------------------------------------------------
MatRigStruttura:
   PRINT "costruzione matrice rigidezza": GOSUB InCorso
```

```
   FOR J = 1 TO JZ
      L = NT(J)
      CALL MatAstaGlobale(TA$(J), X(E1(J)), X(E2(J)),Y(E1(J)),Y(E2(J)),ZA(L),ZAT(L),
                          ZI(L), K(J)*1000, B(J), E, G, Lun, SnAl, CsAl, Ld(), Kg())
      CALL FormaMat(E1(J), E2(J), N(), Kg(), NL, SB, KTT(), KTA(), KAA())
      LOCATE CSRLIN - 1, 44
      PRINT USING "eseguita per il ### %"; J / JZ * 100
   NEXT J
   IF VCed = 1 THEN CALL VincoliCed(IZ, N(), KV(), NL, SB, KTT(), KTA(), KAA())

   GOSUB Completata
RETURN


' ------------------------------- Triangolarizza ---------------------------------
'  Routine che triangolarizza la matrice di rigidezza della struttura
' --------------------------------------------------------------------------------
Triangolarizza:
   PRINT "triangolarizz. matrice rigidezza": GOSUB InCorso

   CALL Triang.K2(KTT(), KTA(), KAA(), SB, 0)

   GOSUB Completata
RETURN


' ------------------------------- LeggeCarRet ------------------------------------
'  Routine per la lettura dei carichi nodali e sulle aste
' --------------------------------------------------------------------------------
LeggeCarRet:
   PRINT
   PRINT "    Schema di carico n."; CAR
   PRINT "lettura dati di carico": GOSUB InCorso

   IF Contr = 1 THEN PRINT #2, "    Schema di carico n."; CAR
   CALL InputCarRet(Contr, Wnkl, IV, IM, IN, IO, PZ, RZ, KZ, NP(), V$(), NB(), NF(),
                    ZA(), ZI(), Q(), QF(), MN(), MNF(), FYN(), FYF(), FX())

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN


' ------------------------------- StampaCarRet -----------------------------------
'  Routine per la stampa dei carichi nodali e sulle aste
' --------------------------------------------------------------------------------
StampaCarRet:
   PRINT "stampa dati di carico"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "SCHEMA DI CARICO N."; CAR
   CALL OutputCarRet(CAR, Wnkl, IV, IM, IN, IO, PZ, RZ, KZ, NP(), V$(), NB(), NF(),
                     ZA(), ZI(), Q(), QF(), MN(), MNF(), FYN(), FYF(), FX())

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
```

```
   END IF
RETURN


' ------------------------------ ConvCarRet ------------------------------------
'  Routine per la conversione dei carichi sui nodi e sulle aste
' ------------------------------------------------------------------------------
ConvCarRet:
   PRINT "conversione dati di carico": GOSUB InCorso

   CALL ConverteCarRet(RZ, PZ, Wnkl, NB(), NF(), ZA(), ZI(), IV, IM, IN,IO,Q(),QF(),
                       MN(), MNF(), FYN(), FYF(), FX(), KZ, V$(), NI(), P(), F())

   GOSUB Completata
RETURN


' ------------------------------ TerminiNoti -----------------------------------
'  Routine per il calcolo dei carichi nodali
' ------------------------------------------------------------------------------
TerminiNoti:
   PRINT "preparazione termini noti": GOSUB InCorso

   CALL TermNoto(IZ, JZ, X(), Y(), ZA(), ZI(), E, E1(), E2(), TA$(), NT(), K(), B(),
                 N(), P(), F(), R())

   GOSUB Completata
RETURN


' ------------------------------ Risolvi ---------------------------------------
'  Routine per la risoluzione mediante sostituzione all'indietro
' ------------------------------------------------------------------------------
Risolvi:
   PRINT "risoluzione": GOSUB InCorso

   CALL Risolve.K2(KTT(), KTA(), KAA(), SB, 0, R())

   GOSUB Completata
RETURN


' ------------------------------ CaratSollec -----------------------------------
'  Routine per il calcolo delle azioni sulle aste
' ------------------------------------------------------------------------------
CaratSollec:
   PRINT "calcolo caratt. sollecitazione": GOSUB InCorso

   CALL CarSoll(JZ, X(), Y(), ZA(), ZAT(), ZI(), E, G, E1(), E2(), TA$(), NT(), K(),
                B(), N(), P(), F(), R(), N1(), T1(), M1(), N2(), T2(), M2())

   GOSUB Completata
RETURN


' ------------------------------ ConvRisultRet ---------------------------------
'  Routine per la conversione delle caratteristiche di sollecitazione
' ------------------------------------------------------------------------------
ConvRisultRet:
```

```
   PRINT "conversione caratt. sollecitaz.": GOSUB InCorso

   CALL ConverteRisRet(RZ, PZ, Wnkl, DefEst, NB(), NR(), NF(), ZA(), ZI(), D(),MS(),
                       MD(), TS(), TD(), MP(), MT(), TR(), SN(), MFS(), MFD(),TFS(),
                       TFD(), KZ, NI(), N(), R(), F(), M1(), M2(), T1(), T2(), N1())

   GOSUB Completata
RETURN


' ------------------------------- StampaRisultRet --------------------------------
'  Routine per la stampa dei risultati
' --------------------------------------------------------------------------------
StampaRisultRet:
   PRINT "stampa risultati"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   CALL OutputRisRet(RZ, PZ, Wnkl, NP(), NB(), NR(), NF(), ZA(), ZI(), D(), MS(),
                     MD(), TS(), TD(), MP(), MT(), TR(), SN(), MFS(), MFD(), TFS(),
                     TFD(), IZ, KZ, NL, V$(), N(), R(), F())
   GOSUB OutputIpotesiRet

   IF FO$ = "CON" THEN
      LOCATE 24, 1
   ELSE
      GOSUB Completata
   END IF
RETURN

' ................................ OutputIpotesiRet ................................
'  Routine per la stampa delle note sulle ipotesi di calcolo
'
OutputIpotesiRet:
   PRINT #2, : PRINT #2,
   PRINT #2, "Il calcolo e' stato eseguito :"
   IF DefEst = 1 THEN
      PRINT #2, "- tenendo conto della deformazione estensionale dei pilastri"
   ELSE
      PRINT #2, "- trascurando la deformazione estensionale dei pilastri"
   END IF
   IF DefTag = 1 THEN
      PRINT #2, "- tenendo conto della deformazione a taglio di travi e pilastri"
      IF Wnkl = 1 THEN
         PRINT #2,"- trascurando la deformazione a taglio della trave di fondazione"
      END IF
   ELSE
      PRINT #2, "- trascurando la deformazione a taglio delle aste"
   END IF
   PRINT #2, : PRINT #2,
RETURN



'                                   PRORET.BAS
'
'  procedure del programma TELRET, utilizzate anche nel programma TELSPA
'
'                                                                rev. 01.90
```

```
' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file MATRIX.BAS
DECLARE SUB AzzeraMat (A!())

' procedure di questo stesso file
DECLARE SUB ControllaSezione (L!, ZA!(), ZI!())
DECLARE SUB PrintCampata (F!, RZ!, NP!())
DECLARE SUB PrintPiano (P!)
DECLARE SUB PrintPilastro (R!, RZ!, NP!())
DECLARE SUB PrintSez (L!, FS$(), ZB!(), ZH!(), ZI!())

' ═══════════════════════════════ ControllaSezione ═══════════════════════════════
'
'   Procedura che controlla se a un tipo di sezione corrisponde
'   area e momento d'inerzia nullo
'
'   Variabili di ingresso:
'      L         numero d'ordine del tipo di sezione
'      ZA()      array che contiene le aree
'      ZI()      array che contiene i momenti di inerzia
'
'   Variabili di uscita:
'      L         se area e momento d'inerzia sono nulli fornisce L=0
'
' ------------------------------------------------------------------------------
'
SUB ControllaSezione (L, ZA(), ZI())

    IF L <> 0 THEN
       IF ZA(L) = 0 AND ZI(L) = 0 THEN L = 0
    END IF

END SUB

' ═══════════════════════════════ ContrRitti ═══════════════════════════════
'   Procedura per il controllo che tra i ritti non vi siano sezioni
'   inammissibili (tipo 0)
' ------------------------------------------------------------------------------
SUB ContrRitti (PZ, RZ, NP(), NB(), NR(), ZA(), ZI(), ER)

    FOR P = 1 TO PZ
       FOR R = 1 TO RZ
          ERpil = 0
          IF NR(R, P) = 0 THEN
             FOR P1 = P + 1 TO PZ
                L = NR(R, P1)
                CALL ControllaSezione(L, ZA(), ZI())
                IF L <> 0 THEN ERpil = 1: EXIT FOR
             NEXT P1
             IF R > 1 AND ERpil = 0 THEN
                FOR P1 = P TO PZ
                   L = NB(R - 1, P1)
                   CALL ControllaSezione(L, ZA(), ZI())
                   IF L <> 0 THEN ERpil = 1: EXIT FOR
                NEXT P1
             END IF
```

```
            IF R < RZ AND ERpil = 0 THEN
               FOR P1 = P TO PZ
                  L = NB(R, P1)
                  CALL ControllaSezione(L, ZA(), ZI())
                  IF L <> 0 THEN ERpil = 1: EXIT FOR
               NEXT P1
            END IF
         END IF
         IF ERpil = 1 THEN
            PRINT "ritto"; NP(R), "piano"; P, "tipo di sezione non ammissibile"
            ER = 1
         END IF
      NEXT R
   NEXT P
   IF ER = 1 THEN
      PRINT "   Esecuzione interrotta:  sezione tipo 0 non ammissibile perche'"
      PRINT "   interrompe la continuita' verticale ed e' quindi incompatibile"
      PRINT "   con l'ipotesi di indeformabilita' estensionale dei ritti"
   END IF

END SUB

' ========================== ConverteCarRet ==========================
'   Procedura per la conversione dei carichi sui nodi e sulle aste
' ------------------------------------------------------------------------------
SUB ConverteCarRet (RZ, PZ, Wnkl, NB(), NF(), ZA(), ZI(), IV, IM, IN, IO, Q(), QF(),
                    MN(), MNF(), FYN(), FYF(), FX(), KZ, V$(), NI(), P(), F())

   CALL AzzeraMat(F())
   CALL AzzeraMat(P())

   ' carichi verticali distribuiti
   IF IV = 1 THEN
      J = 0
      'trave di fondazione
      IF Wnkl = 1 THEN
         FOR F = 0 TO RZ
            L = NF(F)
            CALL ControllaSezione(L, ZA(), ZI())
            IF L <> 0 THEN
               J = J + 1
               P(J, 2) = QF(F)
            END IF
         NEXT F
      END IF
      ' travi in elevazione
      FOR P = 1 TO PZ
         FOR R = 1 TO RZ - 1
            L = NB(R, P)
            CALL ControllaSezione(L, ZA(), ZI())
            IF L <> 0 THEN
               J = J + 1
               P(J, 2) = Q(R, P)
            END IF
         NEXT R
      NEXT P
   END IF

   ' momenti concentrati
```

```
    IF IM = 1 THEN
       ' nodi della trave di fondazione
       IF Wnkl = 1 THEN
          FOR F = 0 TO RZ + 1
             I = F + 1
             IF V$(I) <> "111" THEN F(I, 3) = MNF(F)
          NEXT F
       END IF
       ' nodi in elevazione
       FOR P = 1 TO PZ
          FOR R = 1 TO RZ
             I = RZ * P + 2 + R
             IF V$(I) <> "111" THEN F(I, 3) = MN(R, P)
          NEXT R
       NEXT P
    END IF

    ' forze verticali concentrate
    IF IN = 1 THEN
       ' nodi della trave di fondazione
       IF Wnkl = 1 THEN
          FOR F = 0 TO RZ + 1
             I = F + 1
             IF V$(I) <> "111" THEN F(I, 2) = FYF(F)
          NEXT F
       END IF
       ' nodi in elevazione
       FOR P = 1 TO PZ
          FOR R = 1 TO RZ
             I = RZ * P + 2 + R
             IF V$(I) <> "111" THEN F(I, 2) = FYN(R, P)
          NEXT R
       NEXT P
    END IF

    ' forze orizzontali
    IF IO = 1 THEN
       I = 0
       FOR K = 1 TO KZ
          I = I + 1
          F(NI(I), 1) = FX(K)
          DO
             I = I + 1
          LOOP UNTIL NI(I) = 0
       NEXT K
    END IF

END SUB

' ============================== ConverteGeomRet ==============================
'   Procedura per la conversione dei dati geometrici
' -----------------------------------------------------------------------------
SUB ConverteGeomRet (PZ, RZ, Wnkl, DefEst, DL(), H(), NB(), NR(), NF(), BF(), KF(),
                     ZA(), ZI(), IZ, JZ, KZ, X(), Y(), V$(), NI(), E1(), E2(),TA$(),
                     NT(), B(), K())

    ' coordinate nodi in fondazione
    FOR I = 1 TO RZ + 2
       IF I = 1 THEN X(I) = 0 ELSE X(I) = X(I - 1) + DL(I - 2)
```

```
   Y(I) = 0
NEXT I
' coordinate nodi in elevazione
FOR P = 1 TO PZ
   FOR R = 1 TO RZ
      I = RZ * P + 2 + R
      IF P = 1 THEN Iinf = I - RZ - 1 ELSE Iinf = I - RZ
      X(I) = X(Iinf)
      Y(I) = Y(Iinf) + H(R, P)
   NEXT R
NEXT P

' definizione aste - travi in fondazione
J = 0
IF Wnkl = 1 THEN
   FOR F = 0 TO RZ
      L = NF(F)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         J = J + 1
         E1(J) = F + 1
         E2(J) = F + 2
         NT(J) = L
         B(J) = BF(F)
         K(J) = KF(F)
         TA$(J) = "W"
      END IF
   NEXT F
END IF
JF = J
' definizione aste - travi in elevazione
FOR P = 1 TO PZ
   FOR R = 1 TO RZ - 1
      L = NB(R, P)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         J = J + 1
         E1(J) = RZ * P + 2 + R
         E2(J) = E1(J) + 1
         NT(J) = L
         TA$(J) = "T"
      END IF
   NEXT R
NEXT P
' definizione aste - pilastri
FOR P = 1 TO PZ
   FOR R = 1 TO RZ
      L = NR(R, P)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         J = J + 1
         I = RZ * P + 2 + R
         IF P = 1 THEN Iinf = I - RZ - 1 ELSE Iinf = I - RZ
         E1(J) = Iinf
         E2(J) = I
         NT(J) = L
         TA$(J) = "T"
      END IF
   NEXT R
```

```
    NEXT P
    JZ = J

    ' definizione codici vincolo
    FOR J = JF + 1 TO JZ
       IF DefEst = 0 THEN
          V$(E1(J)) = "010"
          V$(E2(J)) = "010"
       ELSE
          V$(E1(J)) = "000"
          V$(E2(J)) = "000"
       END IF
    NEXT J
    IF Wnkl = 1 THEN
       FOR J = 1 TO JF
          V$(E1(J)) = "100"
          V$(E2(J)) = "100"
       NEXT J
    END IF
    FOR I = 1 TO IZ
       IF V$(I) = "" THEN V$(I) = "111"
    NEXT I
    IF Wnkl <> 1 THEN
       FOR I = 1 TO RZ + 2
          V$(I) = "111"
       NEXT I
    END IF

    ' definizione traversi
    EP = 0
    K = 0
    I = 0
    FOR P = 1 TO PZ
       FOR R = RZ - 1 TO 1 STEP -1
          L = NB(R, P)
          IF L <> 0 THEN
             E1 = RZ * P + 2 + R
             E2 = E1 + 1
             IF E2 <> EP THEN
                IF K > 0 THEN I = I + 1: NI(I) = 0
                K = K + 1
                IF V$(E2) <> "111" THEN I = I + 1: NI(I) = E2
             END IF
             IF V$(E1) <> "111" THEN I = I + 1: NI(I) = E1
             EP = E1
          END IF
       NEXT R
    NEXT P
    I = I + 1: NI(I) = 0
    KZ = K

END SUB

' ============================== ConverteRisRet ==============================
'   Procedura per la valutazione degli spostamenti dei traversi
'   e la conversione delle caratteristiche di sollecitazione
' ----------------------------------------------------------------------------
SUB ConverteRisRet (RZ, PZ, Wnkl, DefEst, NB(), NR(), NF(), ZA(), ZI(), D(), MS(),
                    MD(), TS(), TD(), MP(), MT(), TR(), SN(), MFS(), MFD(), TFS(),
```

```
                TFD(), KZ, NI(), N(), R(), F(), M1(), M2(), T1(), T2(), N1())

' spostamenti traversi
I = 0
FOR K = 1 TO KZ
   I = I + 1
   IF N(NI(I), 1) = 0 THEN D(K) = 0 ELSE D(K) = R(N(NI(I), 1))
   DO
      I = I + 1
   LOOP UNTIL NI(I) = 0
NEXT K
' caratteristiche di sollecitazione nella trave di fondazione
J = 0
IF Wnkl = 1 THEN
   FOR F = 0 TO RZ
      L = NF(F)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         J = J + 1
         MFS(F) = M1(J)
         TFS(F) = T1(J)
         MFD(F) = M2(J)
         TFD(F) = T2(J)
      END IF
   NEXT F
END IF
' caratteristiche di sollecitazione nelle travi in elevazione
FOR P = 1 TO PZ
   FOR R = 1 TO RZ - 1
      L = NB(R, P)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         J = J + 1
         MS(R, P) = M1(J)
         TS(R, P) = T1(J)
         MD(R, P) = M2(J)
         TD(R, P) = T2(J)
      END IF
   NEXT R
NEXT P
' taglio, momento flettente e sforzo normale nei pilastri
FOR P = 1 TO PZ
   FOR R = 1 TO RZ
      L = NR(R, P)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         J = J + 1
         MP(R, P) = M1(J)
         MT(R, P) = M2(J)
         TR(R, P) = T1(J)
         IF DefEst = 1 THEN
            SN(R, P) = N1(J)
         ELSE
            I = RZ * P + 2 + R
            SN(R, P) = F(I, 2)
         END IF
      END IF
   NEXT R
NEXT P
```

```
    IF DefEst = 0 THEN
       FOR R = 1 TO RZ
          FOR P = PZ - 1 TO 1 STEP -1
             SN(R, P) = SN(R, P) + SN(R, P + 1)
          NEXT P
       NEXT R
    END IF

END SUB

' ========================== InputAltezze ==========================
'   Procedura per la lettura delle altezze dei ritti
' ------------------------------------------------------------------
SUB InputAltezze (Contr, PZ, RZ, AltVar, HI(), H())

    AltVar = 0
    FOR P = PZ TO 1 STEP -1
       INPUT #1, HI(P)
       IF Contr = 1 THEN
          IF HI(P) > 0 THEN
             PRINT #2, "piano"; P, "altezza"; HI(P)
          ELSE
             PRINT #2, "piano"; P, "altezza variabile"
          END IF
       END IF
       IF HI(P) = 0 THEN AltVar = 1
    NEXT P
    FOR P = PZ TO 1 STEP -1
       IF HI(P) > 0 THEN
          FOR R = 1 TO RZ
             H(R, P) = HI(P)
          NEXT R
       ELSE
          FOR R = 1 TO RZ
             INPUT #1, H(R, P)
             IF Contr = 1 THEN
                PRINT #2, "piano"; P, "ritto"; R, "altezza"; H(R, P)
             END IF
          NEXT R
       END IF
    NEXT P

END SUB

' ========================== InputAsteRet ==========================
'   Procedura per la lettura dei dati relativi alle aste
' ------------------------------------------------------------------
SUB InputAsteRet (Contr, Wnkl, PZ, RZ, LZ, NP(), NB(), NR(), NF(), BF(), KF(), ER)

    ' travi in elevazione
    FOR P = PZ TO 1 STEP -1
       FOR R = 1 TO RZ - 1
          INPUT #1, NB(R, P)
          IF Contr = 1 THEN
             PRINT #2, "campata"; NP(R); NP(R + 1), "piano"; P,
             PRINT #2, "sezione tipo"; NB(R, P)
          END IF
          IF NB(R, P) < 0 OR NB(R, P) > LZ THEN
             PRINT "campata"; NP(R); NP(R + 1), "piano"; P,
```

```
            PRINT "sezione tipo"; NB(R, P); "  non accettabile"
            ER = 1
         END IF
      NEXT R
   NEXT P
   ' ritti
   FOR P = PZ TO 1 STEP -1
      FOR R = 1 TO RZ
         INPUT #1, NR(R, P)
         IF Contr = 1 THEN
            PRINT #2, "ritto"; NP(R), "piano"; P, "sezione tipo"; NR(R, P)
         END IF
         IF NR(R, P) < 0 OR NR(R, P) > LZ THEN
            PRINT "ritto"; NP(R), "piano"; P,
            PRINT "sezione tipo"; NR(R, P); "  non accettabile"
            ER = 1
         END IF
      NEXT R
   NEXT P
   ' trave in fondazione
   IF Wnkl = 1 THEN
      FOR F = 0 TO RZ
         INPUT #1, NF(F), BF(F), KF(F)
         IF Contr = 1 THEN
            PRINT #2, "trave di fondazione - tronco"; F,
            PRINT #2, "sezione tipo"; NF(F); "   B ="; BF(F); "   K ="; KF(F)
         END IF
         IF NF(F) < 0 OR NF(F) > LZ THEN
            PRINT "trave di fondazione - tronco"; F,
            PRINT "sezione tipo"; NF(F); "  non accettabile"
            ER = 1
         END IF
      NEXT F
   END IF

END SUB

' ========================== InputCarRet ==========================
'   Procedura per la lettura dei carichi su aste, nodi e traversi
' -----------------------------------------------------------------
SUB InputCarRet (Contr, Wnkl, IV, IM, IN, IO, PZ, RZ, KZ, NP(), V$(), NB(), NF(),
                 ZA(), ZI(), Q(), QF(), MN(), MNF(), FYN(), FYF(), FX())

   INPUT #1, IV, IM, IN, IO
   IF IV = 1 THEN GOSUB CarVert
   IF IM = 1 THEN GOSUB MomNodali
   IF IN = 1 THEN GOSUB ForzeNodali
   IF IO = 1 THEN GOSUB ForzeOriz
   EXIT SUB

CarVert:
   IF Contr = 1 THEN PRINT #2, "carichi verticali"
   FOR P = PZ TO 1 STEP -1
      FOR R = 1 TO RZ - 1
         L = NB(R, P)
         CALL ControllaSezione(L, ZA(), ZI())
         IF L <> 0 THEN
            INPUT #1, Q(R, P)
            IF Contr = 1 THEN
```

```
            PRINT #2, "campata"; NP(R); NP(R + 1), "piano"; P, "py ="; Q(R, P)
         END IF
      END IF
   NEXT R
NEXT P
IF Wnkl = 1 THEN
   FOR F = 0 TO RZ
      L = NF(F)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         INPUT #1, QF(F)
         IF Contr = 1 THEN
            PRINT #2, "trave di fondazione, tratto"; F, "py ="; QF(F)
         END IF
      END IF
   NEXT F
END IF
RETURN

MomNodali:
   IF Contr = 1 THEN PRINT #2, "momenti nodali"
   FOR P = PZ TO 1 STEP -1
      FOR R = 1 TO RZ
         I = RZ * P + 2 + R
         IF V$(I) <> "111" THEN
            INPUT #1, MN(R, P)
            IF Contr = 1 THEN
               PRINT #2, "pilastro"; NP(R), "piano"; P, "M ="; MN(R, P)
            END IF
         END IF
      NEXT R
   NEXT P
   IF Wnkl = 1 THEN
      FOR F = 0 TO RZ + 1
         I = F + 1
         IF V$(I) <> "111" THEN
            INPUT #1, MNF(F)
            IF Contr = 1 THEN
               PRINT #2, "trave di fondazione, nodo ";
               CALL PrintPilastro(F, RZ, NP())
               PRINT #2, "M ="; MNF(F)
            END IF
         END IF
      NEXT F
   END IF
RETURN

ForzeNodali:
   IF Contr = 1 THEN PRINT #2, "forze verticali nodali"
   FOR P = PZ TO 1 STEP -1
      FOR R = 1 TO RZ
         I = RZ * P + 2 + R
         IF V$(I) <> "111" THEN
            INPUT #1, FYN(R, P)
            IF Contr = 1 THEN
               PRINT #2, "pilastro"; NP(R), "piano"; P, "Fy ="; FYN(R, P)
            END IF
         END IF
      NEXT R
```

```
   NEXT P
   IF Wnkl = 1 THEN
      FOR F = 0 TO RZ + 1
         I = F + 1
         IF V$(I) <> "111" THEN
            INPUT #1, FYF(F)
            IF Contr = 1 THEN
               PRINT #2, "trave di fondazione, nodo ";
               CALL PrintPilastro(F, RZ, NP())
               PRINT #2, "Fy ="; FYF(F)
            END IF
         END IF
      NEXT F
   END IF
RETURN

ForzeOriz:
   IF Contr = 1 THEN PRINT #2, "forze orizzontali"
   FOR K = KZ TO 1 STEP -1
      INPUT #1, FX(K)
      IF Contr = 1 THEN
         PRINT #2, "traverso"; K, "F ="; FX(K)
      END IF
   NEXT K
RETURN

END SUB

' ============================== InputLuci ==============================
'   Procedura per la lettura delle luci delle campate
' -----------------------------------------------------------------------
SUB InputLuci (Contr, Wnkl, RZ, DL())

    IF Wnkl = 1 THEN
       R1 = 0
       R2 = RZ
    ELSE
       R1 = 1
       R2 = RZ - 1
    END IF
    FOR R = R1 TO R2
       INPUT #1, DL(R)
       IF Contr = 1 THEN
          SELECT CASE R
             CASE 0
                PRINT #2, "sbalzo sinistro trave fondazione", "luce"; DL(R)
             CASE 1 TO RZ - 1
                PRINT #2, "campata"; R, "luce"; DL(R)
             CASE RZ
                PRINT #2, "sbalzo destro trave fondazione", " luce "; DL(R)
          END SELECT
       END IF
    NEXT R

END SUB

' ============================== InputNumRitti ==============================
'   Procedura per la lettura della numerazione in carpenteria dei ritti
' -----------------------------------------------------------------------
```

```
SUB InputNumRitti (Contr, RZ, NP())

    FOR R = 1 TO RZ
       INPUT #1, NP(R)
       IF Contr = 1 THEN
          PRINT #2, "ritto"; R, "numerazione in carpenteria"; NP(R)
       END IF
    NEXT R

END SUB

' ========================== OutputAsteRet ==========================
'   Procedura per la stampa dei dati relativi a travi in elevazione,
'   pilastri e trave di fondazione
' ---------------------------------------------------------------------
SUB OutputAsteRet (PZ, RZ, Wnkl, NP(), DL(), NB(), NR(), NF(), BF(), KF(), FS$(),
                   ZB(), ZH(), ZI())

    ' travi in elevazione
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "LUCI E SEZIONI DELLE TRAVI": PRINT #2,
    PRINT #2, " campata   luce  piano sez.  b     h    inerzia";
    IF PZ > 1 THEN PRINT #2, "  piano sez.  b     h    inerzia";
    PRINT #2,
    PRINT #2, "             m               m     m       m4  ";
    IF PZ > 1 THEN PRINT #2, "              m     m       m4  ";
    PRINT #2, : PRINT #2,
    FOR R = 1 TO RZ - 1
       PRINT #2, USING "### -###  ##.##"; NP(R), NP(R + 1), DL(R);
       FOR P = PZ TO 1 STEP -1
          L = NB(R, P)
          IF (PZ - P) MOD 2 = 0 AND P < PZ THEN
             PRINT #2, : PRINT #2, "                ";
          END IF
          PRINT #2, USING "   ##  "; P;
          CALL PrintSez(L, FS$(), ZB(), ZH(), ZI())
       NEXT P
       PRINT #2, : PRINT #2,
    NEXT R
    ' pilastri
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "SEZIONI DEI PILASTRI": PRINT #2,
    PRINT #2, "pilastro         piano sez.  b     h    inerzia";
    IF PZ > 1 THEN PRINT #2, "  piano sez.  b     h    inerzia";
    PRINT #2,
    PRINT #2, "                             m     m       m4  ";
    IF PZ > 1 THEN PRINT #2, "              m     m       m4  ";
    PRINT #2, : PRINT #2,
    FOR R = 1 TO RZ
       PRINT #2, USING "  ###          "; NP(R);
       FOR P = PZ TO 1 STEP -1
          L = NR(R, P)
          IF (PZ - P) MOD 2 = 0 AND P < PZ THEN
             PRINT #2, : PRINT #2, "                ";
          END IF
          PRINT #2, USING "   ##  "; P;
          CALL PrintSez(L, FS$(), ZB(), ZH(), ZI())
       NEXT P
       PRINT #2, : PRINT #2,
```

```
    NEXT R
    ' trave di fondazione
    IF Wnkl = 1 THEN
       PRINT #2, : PRINT #2, : PRINT #2,
       PRINT #2, "CARATTERISTICHE DELLA TRAVE DI FONDAZIONE": PRINT #2,
       PRINT #2, " campata   luce        sez.  b     h    inerzia    B     K"
       PRINT #2, "             m               m     m       m4      m   N/cm3"
       PRINT #2,
       FOR F = 0 TO RZ
          L = NF(F)
          CALL PrintCampata(F, RZ, NP())
          PRINT #2, USING "  ##.##       "; DL(F);
          CALL PrintSez(L, FS$(), ZB(), ZH(), ZI())
          PRINT #2, USING " ##.## ####.#"; BF(F), KF(F)
       NEXT F
    END IF

END SUB

' ========================================= OutputCarRet ==========================================
'   Procedura per la stampa dei carichi su aste, nodi e traversi
' -----------------------------------------------------------------------------------------------
SUB OutputCarRet (CAR, Wnkl, IV, IM, IN, IO, PZ, RZ, KZ, NP(), V$(), NB(), NF(),
                  ZA(), ZI(), Q(), QF(), MN(), MNF(), FYN(), FYF(), FX())

    IF Wnkl = 1 THEN
       PV = PZ + 1
       R1 = 0
       R2 = RZ + 1
       P1 = 0
    ELSE
       PV = PZ
       R1 = 1
       R2 = RZ
       P1 = 1
    END IF
    IF PV > 4 THEN PV = 4

    IF IV = 1 THEN GOSUB OutCarVert
    IF IM = 1 THEN GOSUB OutMomNodali
    IF IN = 1 THEN GOSUB OutForzeNodali
    IF IO = 1 THEN GOSUB OutForzeOriz
EXIT SUB
 OutCarVert:
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "CARICHI VERTICALI": PRINT #2,
    PRINT #2, " campata";
    FOR P = 1 TO PV
       PRINT #2, "   piano  carico";
    NEXT P
    PRINT #2,
    PRINT #2, "        ";
    FOR P = 1 TO PV
       PRINT #2, "           kN/m ";
    NEXT P
    PRINT #2, : PRINT #2,
    FOR R = R1 TO R2 - 1
       CALL PrintCampata(R, RZ, NP())
       FOR P = PZ TO P1 STEP -1
```

```
         IF P = 0 OR (R >= 1 AND R <= RZ - 1) THEN
            IF (PZ - P) MOD 4 = 0 AND P < PZ AND R <> 0 AND R <> RZ THEN
               PRINT #2, : PRINT #2, "        ";
            END IF
            IF P = 0 THEN
               L = NF(R)
               Q = QF(R)
            ELSE
               L = NB(R, P)
               Q = Q(R, P)
            END IF
            CALL ControllaSezione(L, ZA(), ZI())
            PRINT #2, "  ";
            CALL PrintPiano(P)
            PRINT #2, " ";
            IF L <> 0 THEN
               PRINT #2, USING "#####.# "; Q;
            ELSE
               PRINT #2, "   ---- ";
            END IF
         END IF
      NEXT P
      PRINT #2,
      IF PZ + 1 - P1 > 4 THEN PRINT #2,
   NEXT R
RETURN

OutMomNodali:
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "MOMENTI NEI NODI ": PRINT #2,
   PRINT #2, "pilastro";
   FOR P = 1 TO PV
      PRINT #2, "   piano momento";
   NEXT P
   PRINT #2,
   PRINT #2, "        ";
   FOR P = 1 TO PV
      PRINT #2, "           kNm  ";
   NEXT P
   PRINT #2, : PRINT #2,
   FOR R = R1 TO R2
      CALL PrintPilastro(R, RZ, NP())
      FOR P = PZ TO P1 STEP -1
         IF P = 0 OR (R >= 1 AND R <= RZ) THEN
            IF (PZ - P) MOD 4 = 0 AND P < PZ AND R <> 0 AND R <> RZ + 1 THEN
               PRINT #2, : PRINT #2, "        ";
            END IF
            IF P = 0 THEN
               I = R + 1
               Q = MNF(R)
            ELSE
               I = RZ * P + 2 + R
               Q = MN(R, P)
            END IF
            PRINT #2, "   ";
            CALL PrintPiano(P)
            IF V$(I) = "111" THEN
               PRINT #2, "   ---- ";
            ELSE
```

```
            PRINT #2, USING "#####.# "; Q;
         END IF
      END IF
   NEXT P
   PRINT #2,
   IF PZ + 1 - P1 > 4 THEN PRINT #2,
NEXT R
RETURN

OutForzeNodali:
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "FORZE VERTICALI NEI NODI ": PRINT #2,
   PRINT #2, "pilastro";
   FOR P = 1 TO PV
      PRINT #2, "   piano  forza ";
   NEXT P
   PRINT #2,
   PRINT #2, "        ";
   FOR P = 1 TO PV
      PRINT #2, "            kN  ";
   NEXT P
   PRINT #2, : PRINT #2,
   FOR R = R1 TO R2
      CALL PrintPilastro(R, RZ, NP())
      FOR P = PZ TO P1 STEP -1
         IF P = 0 OR (R >= 1 AND R <= RZ) THEN
            IF (PZ - P) MOD 4 = 0 AND P < PZ AND R <> 0 AND R <> RZ + 1 THEN
               PRINT #2, : PRINT #2, "        ";
            END IF
            IF P = 0 THEN
               I = R + 1
               Q = FYF(R)
            ELSE
               I = RZ * P + 2 + R
               Q = FYN(R, P)
            END IF
            PRINT #2, "   ";
            CALL PrintPiano(P)
            IF V$(I) = "111" THEN
               PRINT #2, "   ---- ";
            ELSE
               PRINT #2, USING "#####.# "; Q;
            END IF
         END IF
      NEXT P
      PRINT #2,
      IF PZ + 1 - P1 > 4 THEN PRINT #2,
   NEXT R
RETURN

OutForzeOriz:
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "FORZE ORIZZONTALI": PRINT #2,
   PRINT #2, "  traverso    forza"
   PRINT #2, "                kN": PRINT #2,
   FOR K = KZ TO 1 STEP -1
      PRINT #2, USING "     ##    ######.#"; K, FX(K)
   NEXT K
RETURN
```

```
END SUB

' ============================== OutputPiani ==============================
'   Procedura per la stampa delle altezze dei piani
' -------------------------------------------------------------------------
SUB OutputPiani (AltVar, PZ, RZ, NP(), HI(), H())

    PRINT #2, "ALTEZZA DEI PIANI": PRINT #2,
    PRINT #2, "piano     altezza (m)": PRINT #2,
    FOR P = PZ TO 1 STEP -1
       PRINT #2, USING "###      "; P;
       IF HI(P) > 0 THEN
          PRINT #2, USING "#####.##"; HI(P)
       ELSE
          PRINT #2, "  variabile"
       END IF
    NEXT P
    IF AltVar = 1 THEN
       IF RZ > 4 THEN RV = 4 ELSE RV = RZ
       PRINT #2, : PRINT #2,
       PRINT #2, "piano   ";
       FOR R = 1 TO RV
          PRINT #2, "    pil.  altezza";
       NEXT R
       PRINT #2,
       PRINT #2, "         ";
       FOR R = 1 TO RV
          PRINT #2, "              m   ";
       NEXT R
       PRINT #2, : PRINT #2,
       FOR P = PZ TO 1 STEP -1
          IF HI(P) = 0 THEN
             PRINT #2, USING " ##      "; P;
             FOR R = 1 TO RZ
                IF R MOD 4 = 1 AND R > 1 THEN PRINT #2, : PRINT #2, "         ";
                PRINT #2, USING "    ###   ###.## "; NP(R), H(R, P);
             NEXT R
             PRINT #2, : PRINT #2,
          END IF
       NEXT P
    END IF

END SUB

' ============================== OutputRisRet ==============================
'   Procedura per la stampa dei risultati
' -------------------------------------------------------------------------
SUB OutputRisRet (RZ, PZ, Wnkl, NP(), NB(), NR(), NF(), ZA(), ZI(), D(), MS(), MD(),
                  TS(), TD(), MP(), MT(), TR(), SN(), MFS(), MFD(), TFS(), TFD(),IZ,
                  KZ, NL, V$(), N(), R(), F())

    DIM V(3)

    GOSUB Spostamenti
    GOSUB RisTravi
    GOSUB RisPilastri
    IF Wnkl = 1 THEN GOSUB RisFond
    GOSUB Squilibrio
```

```
    EXIT SUB

Spostamenti:
    ' stampa spostamenti orizzontali dei traversi
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "SPOSTAMENTO ORIZZONTALE DEI TRAVERSI": PRINT #2,
    PRINT #2, " traverso      spostamento"
    PRINT #2, "                   (mm)"
    PRINT #2,
    FOR K = KZ TO 1 STEP -1
       PRINT #2, USING "   ##          ####.###"; K, D(K) * 1000
    NEXT K
    ' stampa spostamenti verticali e rotazioni dei nodi
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI": PRINT #2,
    PRINT #2, "pilastro    piano    vy (mm)   rotaz.x1000";
    IF PZ > 1 THEN PRINT #2, "    piano    vy (mm)   rotaz.x1000";
    PRINT #2, : PRINT #2,
    IF Wnkl = 1 THEN
       PV = PZ + 1
       R1 = 0
       R2 = RZ + 1
       P1 = 0
    ELSE
       PV = PZ
       R1 = 1
       R2 = RZ
       P1 = 1
    END IF
    IF PV > 2 THEN PV = 2
    FOR R = R1 TO R2
       CALL PrintPilastro(R, RZ, NP())
       FOR P = PZ TO P1 STEP -1
          IF P = 0 OR (R >= 1 AND R <= RZ) THEN
             IF P = 0 THEN I = R + 1 ELSE I = RZ * P + 2 + R
             IF N(I, 2) <> 0 THEN VY = R(N(I, 2)) * 1000 ELSE VY = 0
             IF N(I, 3) <> 0 THEN ROT = R(N(I, 3)) * 1000 ELSE ROT = 0
             IF (PZ - P) MOD 2 = 0 AND P < PZ AND R <> 0 AND R <> RZ + 1 THEN
                PRINT #2, : PRINT #2, "          ";
             END IF
             PRINT #2, "    ";
             CALL PrintPiano(P)
             PRINT #2, USING "   ####.###    ####.###   "; VY, ROT;
          END IF
       NEXT P
       PRINT #2, : PRINT #2,
    NEXT R
RETURN

RisTravi:
    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2,"CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI": PRINT #2,
    PRINT #2," campata  piano    momento a     momento a     taglio a      taglio a"
    PRINT #2,"                  sinistra (kNm) destra (kNm) sinistra (kN) destra(kN)"
    FOR R = 1 TO RZ - 1
       PRINT #2,
       FOR P = PZ TO 1 STEP -1
          PRINT #2, USING "### -###   ##"; NP(R), NP(R + 1), P;
          L = NB(R, P)
```

```
         CALL ControllaSezione(L, ZA(), ZI())
         IF L <> 0 THEN
            PRINT #2, USING "   #######.## ";MS(R, P), MD(R, P), TS(R, P), TD(R, P)
         ELSE
            PRINT #2, USING "            "; "-----", "-----", "-----", "-----"
         END IF
      NEXT P
   NEXT R
RETURN

RisPilastri:
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2,"CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI": PRINT #2,
   PRINT #2," pilastro piano    momento       momento        taglio       sforzo"
   PRINT #2,"                  sup. (kNm)    inf. (kNm)       (kN)     normale(kN)"
   FOR R = 1 TO RZ
      PRINT #2,
      FOR P = PZ TO 1 STEP -1
         PRINT #2, USING "   ###     ##"; NP(R), P;
         L = NR(R, P)
         CALL ControllaSezione(L, ZA(), ZI())
         IF L <> 0 THEN
            PRINT #2, USING "   #######.## ";MT(R, P), MP(R, P), TR(R, P), SN(R, P)
         ELSE
            PRINT #2, USING "            "; "-----", "-----", "-----", "-----"
         END IF
      NEXT P
   NEXT R
RETURN

RisFond:
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2,"CARATTERISTICHE DI SOLLECITAZIONE NELLA TRAVE DI FONDAZIONE":PRINT #2,
   PRINT #2," campata          momento a     momento a     taglio a      taglio a"
   PRINT #2,"                 sinistra (kNm) destra (kNm) sinistra (kN) destra(kN)"
   FOR F = 0 TO RZ
      SELECT CASE F
         CASE 0
            PRINT #2, " sb.sin      ";
         CASE 1 TO RZ - 1
            PRINT #2, USING "### -###     "; NP(F), NP(F + 1);
         CASE RZ
            PRINT #2, " sb.des      ";
      END SELECT
      L = NF(F)
      CALL ControllaSezione(L, ZA(), ZI())
      IF L <> 0 THEN
         PRINT #2, USING "   #######.## "; MFS(F), MFD(F), TFS(F), TFD(F)
      ELSE
         PRINT #2, USING "            "; "-----", "-----", "-----", "-----"
      END IF
   NEXT F
RETURN

Squilibrio:
   ' controlla se vi e' squilibrio nei nodi
   Squil = 0
   FOR I = 1 TO IZ
      FOR K = 1 TO 3
```

```
      IF MID$(V$(I), K, 1) = "O" THEN
         IF N(I, K) > NL AND ABS(F(I, K)) > .01 THEN Squil = 1: EXIT FOR
      END IF
   NEXT K
   IF Squil = 1 THEN EXIT FOR
NEXT I
' stampa l'eventuale squilibrio nei nodi
IF Squil = 1 THEN
   PRINT #2, : PRINT #2,
   PRINT #2, "SQUILIBRIO NEI NODI": PRINT #2,
   PRINT #2, "pilastro  piano        Fx (kN)      Fy (kN)      M (kNm)"
   PRINT #2,
   FOR I = 1 TO IZ
      V = 0
      FOR K = 1 TO 3
         IF N(I, K) > NL AND ABS(F(I, K)) > .01 THEN V(K) = 1 ELSE V(K) = 0
         V = V + V(K)
      NEXT K
      IF V <> 0 THEN
         P = INT((I - 3) / RZ)
         IF P = 0 THEN R = I - 1 ELSE R = I - RZ * P - 2
         CALL PrintPilastro(R, RZ, NP())
         PRINT "  ";
         CALL PrintPiano(P)
         FOR K = 1 TO 3
            IF V(K) = 0 THEN
               PRINT #2, "          ----";
            ELSE
               PRINT #2, USING "    ######.##"; F(I, K);
            END IF
         NEXT K
         PRINT #2,
      END IF
   NEXT I
END IF
' controlla se vi e' squilibrio nei traversi
Squil = 0
FOR K = 1 TO KZ
   F = 0
   FOR I = 1 TO IZ
      IF N(I, 1) = K THEN
         F = F + F(I, 1)
      END IF
   NEXT I
   IF ABS(F) > .01 THEN
      Squil = 1
      EXIT FOR
   END IF
NEXT K
' stampa l'eventuale squilibrio nei traversi
IF Squil = 1 THEN
   PRINT #2, : PRINT #2,
   PRINT #2, "SQUILIBRIO NEI TRAVERSI ORIZZONTALI": PRINT #2,
   PRINT #2, "traverso    squilibrio (kN)"
   PRINT #2,
   FOR K = KZ TO 1 STEP -1
      F = 0
      FOR I = 1 TO IZ
         IF N(I, 1) = K THEN
```

```
            F = F + F(I, 1)
         END IF
      NEXT I
      IF ABS(F) > .01 THEN
         PRINT #2, USING "  ###       ######.##"; K, F
      END IF
   NEXT K
END IF
RETURN

END SUB

' ============================== OutputTabSez ==============================
'   Procedura per la stampa della tabella dei tipi di sezione
' ----------------------------------------------------------------------------
SUB OutputTabSez (DefTag, LZ, FS$(), ZB(), ZH(), ZA(), ZAT(), ZI())

    PRINT #2, : PRINT #2, : PRINT #2,
    PRINT #2, "TABELLA DEI TIPI DI SEZIONE": PRINT #2,
    PRINT #2, "                    b     h     area ";
    IF DefTag = 1 THEN PRINT #2, "   A/Chi ";
    PRINT #2, "   inerzia ": PRINT #2,
    PRINT #2, "sezione  forma     m     m      m2  ";
    IF DefTag = 1 THEN PRINT #2, "     m2  ";
    PRINT #2, "      m4   "
    FOR L = 1 TO LZ
       PRINT #2, USING " ###       !   "; L, FS$(L);
       IF FS$(L) = "R" THEN
          PRINT #2, USING " ##.## ##.##"; ZB(L), ZH(L);
       ELSE
          PRINT #2, "  ----  ----";
       END IF
       PRINT #2, USING " ###.####"; ZA(L);
       IF DefTag = 1 THEN
          PRINT #2, USING " ###.####"; ZAT(L);
       END IF
       PRINT #2, USING " ###.######"; ZI(L)
    NEXT L

END SUB

' ============================== PrintCampata ==============================
'
'   Procedura per la stampa dell'indicazione della campata (o sbalzo)
'
'   Variabili di ingresso:
'      F         numero d'ordine della campata
'      RZ        numero totale di pilastri
'      NP()      array che contiene la numerazione dei pilastri in carpenteria
'
' ----------------------------------------------------------------------------
'
SUB PrintCampata (F, RZ, NP())

    SELECT CASE F
       CASE 0
          PRINT #2, "  sb.sin";
       CASE 1 TO RZ - 1
          PRINT #2, USING "### -###"; NP(F), NP(F + 1);
```

```
      CASE RZ
         PRINT #2, "  sb.des";
   END SELECT

END SUB

' ============================== PrintPiano ==============================
'
'   Procedura per la stampa dell'indicazione del piano (o fondazione)
'
'   Variabili di ingresso:
'      P          numero d'ordine del piano (0=fondazione)
'
' ------------------------------------------------------------------------
'
SUB PrintPiano (P)

   IF P = 0 THEN
      PRINT #2, "fond.";
   ELSE
      PRINT #2, USING " ##  "; P;
   END IF

END SUB

' ============================== PrintPilastro ==============================
'
'   Procedura per la stampa dell'indicazione del pilastro
'
'   Variabili di ingresso:
'      R          numero d'ordine del pilastro
'      RZ         numero totale di pilastri
'      NP()       array che contiene la numerazione dei pilastri in carpenteria
'
' ------------------------------------------------------------------------
'
SUB PrintPilastro (R, RZ, NP())

   SELECT CASE R
      CASE 0
         PRINT #2, "estr.sin";
      CASE 1 TO RZ
         PRINT #2, USING " ####   "; NP(R);
      CASE RZ + 1
         PRINT #2, "estr.des";
   END SELECT

END SUB

' ============================== PrintSez ==============================
'
'   Procedura per la stampa di base, altezza ed inerzia della sezione L
'
'   Variabili di ingresso:
'      L          numero d'ordine del tipo di sezione
'      FS$()      array che contiene il codice di forma delle sezioni
'      ZB()       array che contiene le basi
'      ZH()       array che contiene le altezze
'      ZI()       array che contiene i momenti di inerzia
```

```
'
' ------------------------------------------------------------------------------
'
SUB PrintSez (L, FS$(), ZB(), ZH(), ZI())

    IF L = 0 THEN
       PRINT #2, USING "###  ----  ----  --------"; L;
    ELSEIF FS$(L) = "R" THEN
       PRINT #2, USING "### ##.## ##.## ##.######"; L, ZB(L), ZH(L), ZI(L);
    ELSEIF FS$(L) = "G" THEN
       PRINT #2, USING "###  ----  ---- ##.######"; L, ZI(L);
    END IF

END SUB
```

7. Modalità di utilizzazione del programma

I dati necessari per descrivere lo schema geometrico e di carico, elencati in dettaglio nel seguito, devono essere conservati in un file su memoria di massa, che può essere preparato mediante un qualsiasi word processor tra quelli in commercio.

Quando il programma viene mandato in esecuzione, esso richiede espressamente in maniera interattiva le seguenti informazioni:

- il nome del file che contiene i dati;
- il nome dell'unità alla quale inviare l'output; questa può essere lo schermo (CON), la stampante (PRN) oppure un file del quale si indicherà il nome;
- se nel calcolo si vuol tener conto della deformabilità estensionale dei pilastri (si risponda 1 in caso affermativo, 0 in caso negativo). La presenza di ritti inestensibili è simulata bloccando i nodi allo spostamento verticale; gli schemi che contengono una trave di fondazione e quelli in cui la continuità verticale dei pilastri è interrotta per la mancanza di ritti devono essere risolti tenendo conto della deformabilità estensionale;
- se nel calcolo si vuole tener conto della deformabilità a taglio (si risponda ancora 1 in caso affermativo, 0 in caso negativo). La versione attuale del programma prevede la deformabilità a taglio solo per aste non collegate ad un suolo elastico; assume inoltre automaticamente un valore del modulo di elasticità tangenziale G legato a quello del modulo di Young E dalla relazione $G = 0.45\ E$, valida per $\nu = 0.1$ (cemento armato);
- se si vuole effettuare un controllo dei dati man mano che vengono letti. Conviene utilizzare questa opzione quando il programma si blocca a causa di qualche errore nei dati che non si riesce ad individuare. Il

controllo consiste nell'inviare all'unità di uscita una segnalazione per ciascun valore letto, nella quale il programma indica come questo valore è stato interpretato.

Il programma passa poi a leggere dal file indicato i dati geometrici e di carico. Ciascuna esecuzione consente la risoluzione di uno schema geometrico soggetto a più condizioni di carico. I dati devono essere memorizzati nel file in maniera sequenziale, separati da virgole o da andate a capo, col seguente ordine:

- gruppo di dati che descrivono la geometria dello schema
- gruppo di dati che descrivono la prima condizione di carico
- gruppo di dati che descrivono la seconda condizione di carico

  .....
- gruppo di dati che descrivono l'ultima condizione di carico

Poiché il programma è orientato alla soluzione di schemi di telai estratti da edifici, nel preparare lo schema geometrico ogni pilastro deve essere contrassegnato con il numero utilizzato per esso nella carpenteria (unico per tutto l'allineamento verticale); ciascuna campata di trave viene così individuata dalla numerazione dei due pilastri di estremità. I piani degli orizzontamenti e gli ordini dei pilastri sono numerati dal basso verso l'alto. L'insieme di nodi collegati da travi poste in sequenza sono considerati appartenenti ad un traverso orizzontale inestensibile ed avranno quindi un unico spostamento orizzontale.

Nelle applicazioni pratiche numerose aste di uno schema presentano la stessa sezione. Si è pertanto ritenuto più comodo per la immissione dei dati che l'utente individui preliminarmente i differenti tipi di sezione, e definisca poi per ogni asta il numero d'ordine della sezione corrispondente. Per ciascun tipo di sezione deve essere definita la forma (rettangolare o generica) e le relative caratteristiche geometriche.

La mancanza di un'asta dello schema può essere indicata assegnando ad essa il tipo di sezione 0. Se nel calcolo si vuole trascurare la deformabilità estensionale dei pilastri, ritti di tipo 0 sono ammessi solo se non interrompono la continuità verticale, cioè solo se tutte le aste sovrastanti sono anch'esse mancanti. Una trave di tipo 0 interrompe la continuità dell'orizzontamento e lascia i nodi di estremità liberi di spostarsi in orizzontale di quantità diverse; possono crearsi in tal modo più traversi inestensibili in uno stesso orizzontamento.

Assegnare ad un'asta una sezione con area e momento d'inerzia nulli ha un significato in parte diverso rispetto al definirla mancante (cioè di

tipo 0). Se nel calcolo si vuole trascurare la deformabilità estensionale, un pilastro con sezione di area e inerzia nulla è considerato come un pendolo inestensibile ed è quindi sempre ammesso. Se invece si tiene conto della estensibilità esso è perfettamente equivalente ad un'asta di tipo 0. Una trave con sezione di area e inerzia nulla è invece sempre considerata come un pendolo inestensibile che non interrompe la continuità del traverso ed impone quindi l'uguaglianza dello spostamento orizzontale dei nodi da essa collegati.

Il gruppo di dati che descrive la geometria dello schema deve rispettare l'ordine di seguito riportato. I valori forniti devono essere conformi alle unità di misura indicate tra parentesi:

- intestazione
- numero totale di pilastri (intesi come allineamenti verticali)
- numero totale di piani ovvero di traversi geometrici, intesi come allineamenti orizzontali o inclinati indipendentemente dal fatto che la loro continuità sia mantenuta o interrotta dalla mancanza di travi e che siano quindi costituiti da uno o più traversi inestensibili
- numero di tipi di sezioni
- presenza della trave di fondazione (0=no, 1=si)
- numerazione dei pilastri in carpenteria, da sinistra verso destra
- altezza degli interpiani, dall'alto verso il basso (m); se ad un interpiano i ritti non hanno tutti la stessa altezza deve essere assegnato ad essa il valore 0
- per ciascun interpiano i cui ritti non hanno tutti la stessa altezza, dall'alto verso il basso:
  altezza di ciascun ritto, da sinistra verso destra (m)
- se vi è la trave di fondazione: lunghezza dello sbalzo sinistro (m)
- luce di ogni campata, da sinistra verso destra (m)
- se vi è la trave di fondazione: lunghezza dello sbalzo destro (m)
- modulo di elasticità normale (N mm^{-2})
- per ogni tipo di sezione:
  - carattere che identifica la forma della sezione:
    R = sezione di forma rettangolare
    G = sezione di forma generica
  - caratteristiche geometriche della sezione:
    - base (m) ed altezza (m) per le sezioni rettangolari
    - area (m^2), rapporto tra area e fattore di taglio (m^2) e momento

d'inerzia per quelle generiche

- per le travi di ciascun piano, dall'alto verso il basso:
  tipo di sezione di ogni campata, da sinistra verso destra; assegnare una sezione di tipo 0 indica che l'asta è mancante
- per i pilastri di ciascun interpiano, dall'alto verso il basso:
  tipo di sezione di ciascun ritto, da sinistra verso destra; assegnare una sezione di tipo 0 indica che l'asta è mancante
- se è presente la trave di fondazione, per ciascun tratto, inclusi gli sbalzi, da sinistra verso destra:
  - tipo di sezione; assegnare una sezione di tipo 0 indica che il tratto è mancante
  - larghezza della sottobase (m)
  - costante di sottofondo (N cm^{-3})

Nel definire i carichi, questi devono essere riuniti in quattro gruppi: carichi verticali uniformemente distribuiti sulle travi, momenti concentrati nei nodi, forze verticali concentrate nei nodi, forze orizzontali applicate ai traversi. Poiché in genere non tutti questi gruppi sono presenti, si è ritenuto preferibile che l'utente indichi preliminarmente quali gruppi di carico esistono ed assegni poi il valore di tutti gli elementi dei gruppi indicati.

Ciascun carico deve rispettare la convenzione dei segni definita nella prima parte del testo. I carichi verticali uniformemente distribuiti e le forze nodali verticali sono positivi se concordi col verso positivo dell'asse y, cioè se diretti verso l'alto; i momenti nodali sono positivi se concordi col verso che porta l'asse x sull'asse y, cioè se antiorari; le forze orizzontali sui traversi sono positive se concordi col verso dell'asse x, cioè se dirette verso destra. Si noti che con tale convenzione i carichi verticali che agiscono verso il basso devono essere indicati col segno meno.

Il gruppo di dati che descrive la singola condizione di carico deve rispettare l'ordine e le unità di misura di seguito indicati:

- indice di esistenza dei carichi verticali sulle travi (0=no, 1=si)
- indice di esistenza dei momenti concentrati nei nodi (0=no, 1=si)
- indice di esistenza delle forze verticali concentrate nei nodi (0=no, 1=si)
- indice di esistenza dei carichi orizzontali (0=no, 1=si)
- se esistono carichi verticali sulle travi (primo indice uguale a 1):
  - per ciascun piano, dall'alto verso il basso:
    carico verticale su ciascuna campata (kN m^{-1}), da sinistra verso destra, con esclusione delle aste non esistenti (cioè con sezione di

tipo 0) o con sezione avente area ed inerzia nulli
  - per la trave di fondazione, se esiste:
    carico verticale su ciascun tratto (kN m^{-1}), da sinistra verso destra, con esclusione dei tratti in cui la trave non esiste o ha sezione con area ed inerzia nulli
- se esistono momenti concentrati nei nodi (secondo indice uguale a 1):
  - per ciascun piano, dall'alto verso il basso:
    momento concentrato in ciascun nodo (kNm), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale (cioè con sezione non nulla)
  - per la trave di fondazione, se esiste:
    momento concentrato in ciascun nodo (kNm), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale
- se esistono forze verticali concentrate nei nodi (terzo indice uguale a 1):
  - per ciascun piano, dall'alto verso il basso:
    forza verticale concentrata in ciascun nodo (kN), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale
  - per la trave di fondazione, se esiste:
    forza verticale concentrata in ciascun nodo (kN), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale
- se esistono forze orizzontali sui traversi (quarto indice uguale a 1):
  - per ciascun traverso, dall'alto verso il basso (e da sinistra verso destra se esistono più traversi inestensibili ad uno stesso piano):
    forza orizzontale applicata al traverso (kN)

8. Risultati forniti

L'output del programma contiene innanzitutto l'elenco dei dati geometrici ed elastici forniti in ingresso dall'utente, per consentirne il necessario controllo. I pilastri sono individuati in base alla loro numerazione in carpenteria; le travi in base alla numerazione dei pilastri di estremità. Per ciascun'asta sono riportate in forma sintetica le informazioni relative alla sezione, rinviando ad un'apposita tabella dei tipi di sezione per informazioni più dettagliate. Non vengono stampati dati irrilevanti ai fini del calcolo: ad esempio, se non si è tenuto conto della deformabilità a taglio non viene indicato il rapporto A/χ anche se esso deve essere sempre fornito per le sezioni di forma generica.

Per ciascuno schema di carico sono indicati innanzitutto i carichi applicati. Sono poi riportati lo spostamento orizzontale dei traversi inestensibili e lo spostamento verticale e la rotazione dei nodi. Gli spostamenti, positivi se concordi col verso degli assi (cioè diretti rispettivamente verso destra e verso l'alto) sono espressi in mm. La rotazione, positiva se antioraria, è indicata in radianti amplificata di un fattore 1000. I valori delle caratteristiche di sollecitazione M, T, N nei due estremi di ciascun'asta sono espressi in kNm e kN. Si ricorda che per esse valgono le usuali convenzioni della Scienza delle Costruzioni, considerando le travi orientate da sinistra a destra, i pilastri dal basso verso l'alto. Il momento flettente è quindi positivo se tende le fibre inferiori delle travi e quelle a destra dei pilastri; il taglio è positivo se la microcoppia costituita dalle azioni taglianti su due facce opposte è antioraria, lo sforzo normale è positivo se di trazione. Se si riscontrano squilibri in nodi non vincolati o nei traversi, dovuti agli inevitabili errori di troncamento del procedimento risolutivo, questi vengono espressamente segnalati per consentire all'utente di rendersi conto della loro entità e quindi della maggiore o minore validità dei risultati ottenuti.

9. Esempio

Il programma è stato utilizzato per la risoluzione dello schema di telaio mostrato nella figura 6.7, che racchiude molte delle particolarità esaminate in questo capitolo (orizzontamenti interrotti in due traversi inestensibili, pilastri non continui, altezza dell'interpiano non costante, trave di fondazione). Di seguito sono mostrati i dati, contenuti nel file `TORRI`, ed i conseguenti risultati.

```
Telaio con piani superiori divisi in due corpi distinti
5,4,7,1
21,22,23,24,25
3.2,0,3.5,4
3.5,3.5,3,2.5,2.5
1.5,5.5,4,4,4.5,1.5
28500
R,.3,.6,R,.6,.3,R,.3,.3,R,.8,.24,R,.3,.5,R,.3,.7,G,.57,.33,.057307
5,0,0,4
5,0,0,4
6,6,6,6
6,6,6,6
1,1,0,2,2
1,1,0,2,2
1,1,3,2,2
1,1,0,2,2
7,1.4,50,7,1.4,50,7,1.4,50,7,1.4,50,7,1.4,50,7,1.4,50
```

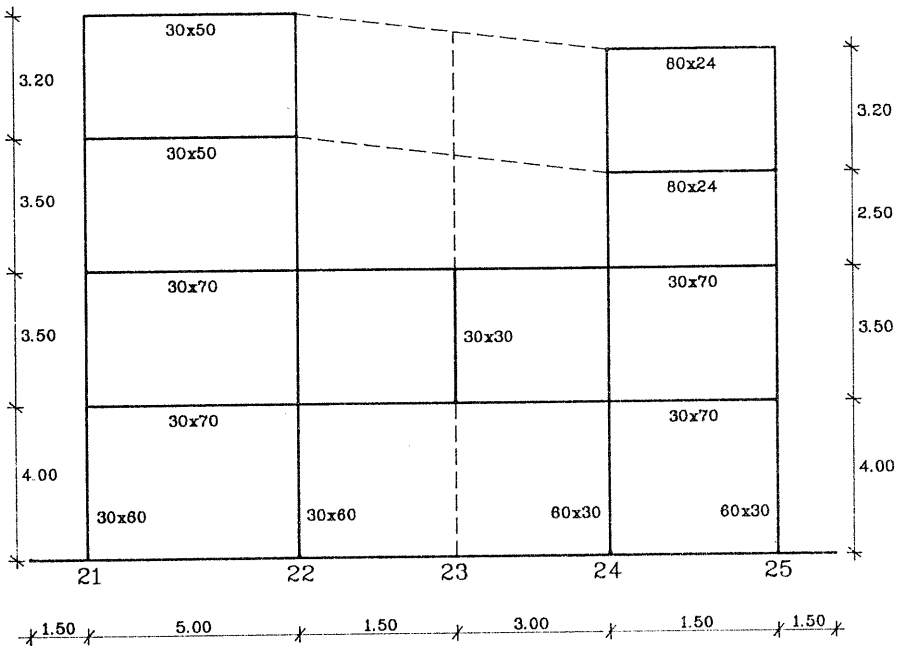


Fig. 6.7 — Schema di telaio con piani superiori divisi in due corpi distinti

```
1,0,0,1
-25,-25
-30,-30
-30,-25,-25,-30
-38,-38,-38,-38
0,0,0,0,0,0
42,39,31,29,72,44
```

```
File dati:  TORRI          data:  18/02/1990     ora:  23:38:47

Telaio con piani superiori divisi in due corpi distinti

ALTEZZA DEI PIANI

piano     altezza (m)

  4          3.20
  3        variabile
  2          3.50
  1          4.00
```

piano	pil.	altezza m	pil.	altezza m	pil.	altezza m	pil.	altezza m
3	21	3.50	22	3.50	23	3.00	24	2.50
	25	2.50						

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
21 - 22	5.50	4	5	0.30	0.50	0.003125	3	5	0.30	0.50	0.003125
		2	6	0.30	0.70	0.008575	1	6	0.30	0.70	0.008575
22 - 23	4.00	4	0	----	----	--------	3	0	----	----	--------
		2	6	0.30	0.70	0.008575	1	6	0.30	0.70	0.008575
23 - 24	4.00	4	0	----	----	--------	3	0	----	----	--------
		2	6	0.30	0.70	0.008575	1	6	0.30	0.70	0.008575
24 - 25	4.50	4	4	0.80	0.24	0.000922	3	4	0.80	0.24	0.000922
		2	6	0.30	0.70	0.008575	1	6	0.30	0.70	0.008575

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
21	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
22	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
23	4	0	----	----	--------	3	0	----	----	--------
	2	3	0.30	0.30	0.000675	1	0	----	----	--------
24	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350
25	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350

CARATTERISTICHE DELLA TRAVE DI FONDAZIONE

campata	luce m	sez.	b m	h m	inerzia m4	B m	K N/cm3
sb.sin	1.50	7	----	----	0.057307	1.40	50.0

21 - 22	5.50	7	----	----	0.057307	1.40	50.0
22 - 23	4.00	7	----	----	0.057307	1.40	50.0
23 - 24	4.00	7	----	----	0.057307	1.40	50.0
24 - 25	4.50	7	----	----	0.057307	1.40	50.0
sb.des	1.50	7	----	----	0.057307	1.40	50.0

TABELLA DEI TIPI DI SEZIONE

		b	h	area	inerzia
sezione	forma	m	m	m2	m4
1	R	0.30	0.60	0.1800	0.005400
2	R	0.60	0.30	0.1800	0.001350
3	R	0.30	0.30	0.0900	0.000675
4	R	0.80	0.24	0.1920	0.000922
5	R	0.30	0.50	0.1500	0.003125
6	R	0.30	0.70	0.2100	0.008575
7	G	----	----	0.5700	0.057307

Modulo di elasticita' normale = 28500 N/mm2

SCHEMA DI CARICO N. 1

CARICHI VERTICALI

campata	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m
sb.sin	fond.	0.0						
21 - 22	4	-25.0	3	-30.0	2	-30.0	1	-38.0
	fond.	0.0						
22 - 23	4	----	3	----	2	-25.0	1	-38.0
	fond.	0.0						
23 - 24	4	----	3	----	2	-25.0	1	-38.0
	fond.	0.0						
24 - 25	4	-25.0	3	-30.0	2	-30.0	1	-38.0
	fond.	0.0						
sb.des	fond.	0.0						

FORZE ORIZZONTALI

traverso	forza kN

6	42.0
5	39.0
4	31.0
3	29.0
2	72.0
1	44.0

RISULTATI

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
6	18.978
5	19.215
4	15.361
3	13.849
2	10.739
1	5.891

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
estr.sin	fond.	0.145	-0.345			
21	4	-0.681	-1.100	3	-0.649	-1.272
	2	-0.576	-1.091	1	-0.489	-1.325
	fond.	-0.374	-0.346			
22	4	-2.760	-0.636	3	-2.706	-1.047
	2	-2.573	-0.904	1	-2.286	-1.048
	fond.	-1.728	-0.070			
23	4	0.000	0.000	3	0.000	0.000
	2	-4.616	0.204	1	-4.644	0.222
	fond.	-0.968	0.118			
24	4	-1.966	-1.607	3	-1.941	-1.817
	2	-1.899	0.163	1	-1.709	0.162
	fond.	-1.301	-0.121			
25	4	-2.122	-0.504	3	-2.077	-1.050
	2	-1.998	-0.430	1	-1.831	-0.424
	fond.	-1.561	-0.103			
estr.des	fond.	-1.671	-0.063			

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
21 - 22	4	-7.88	-103.12	51.43	-86.07
21 - 22	3	4.33	-148.30	54.75	-110.25
21 - 22	2	101.76	-236.39	21.02	-143.98
21 - 22	1	145.82	-312.74	21.13	-187.87
22 - 23	4	-----	-----	-----	-----
22 - 23	3	-----	-----	-----	-----
22 - 23	2	-24.66	93.32	79.50	-20.50
22 - 23	1	-37.72	91.52	108.31	-43.69
23 - 24	4	-----	-----	-----	-----
23 - 24	3	-----	-----	-----	-----
23 - 24	2	145.92	-217.57	-40.87	-140.87
23 - 24	1	144.32	-252.97	-23.32	-175.32
24 - 25	4	0.00	-71.49	40.36	-72.14
24 - 25	3	2.99	-95.30	45.66	-89.34
24 - 25	2	-46.51	-119.17	51.35	-83.65
24 - 25	1	-62.00	-129.93	70.40	-100.60

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
21	4	-7.88	24.39	-10.09	-51.43
21	3	28.73	-44.68	20.97	-106.18
21	2	57.08	-36.43	26.72	-127.20
21	1	109.39	-184.77	73.54	-148.33
22	4	103.12	-63.56	52.09	-86.06
22	3	84.74	-97.36	52.03	-196.32
22	2	114.37	-101.69	61.73	-419.79
22	1	173.33	-248.59	105.48	-715.98
23	4	-----	-----	-----	-----
23	3	-----	-----	-----	-----
23	2	52.60	-52.80	30.11	20.37
23	1	-----	-----	-----	-----
24	4	0.00	5.04	-1.58	-40.36
24	3	8.03	-68.97	30.80	-86.02
24	2	102.09	-102.07	58.33	-278.25
24	1	88.90	-83.47	43.09	-523.97
25	4	71.49	-58.35	40.58	-72.14
25	3	36.95	-56.04	37.20	-161.48
25	2	63.12	-63.25	36.11	-245.13
25	1	66.68	-72.87	34.89	-345.72

CARATTERISTICHE DI SOLLECITAZIONE NELLA TRAVE DI FONDAZIONE

campata	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
sb.sin	-0.00	2.21	0.00	12.00
21 - 22	186.98	373.35	-136.33	293.27
22 - 23	621.94	-207.62	-422.70	-33.66
23 - 24	-207.62	202.64	-33.66	257.36
24 - 25	286.11	56.17	-266.61	175.54
sb.des	129.04	0.00	-170.19	-0.00

Il calcolo è stato eseguito :
- tenendo conto della deformazione estensionale dei pilastri
- trascurando la deformazione a taglio delle aste

CAPITOLO SETTIMO

L'INSIEME SPAZIALE DI TELAI PIANI

1. Generalità

Il programma che qui si presenta consente di analizzare una struttura schematizzata come un insieme spaziale di telai piani mutuamente collegati da impalcati orizzontali indeformabili. Le potenzialità ed i limiti di questo modello sono stati discussi nella prima parte del testo. Si ricorda in particolare che nell'impostare il procedimento risolutivo si è supposto:

- che gli spostamenti dei nodi della struttura, provocati dai carichi su essa agenti, siano piccoli rispetto alle sue dimensioni;
- che siano irrilevanti le non linearità geometriche (effetto $P - \delta$ o effetto instabilizzante dei carichi verticali), cioè che sia indifferente applicare i carichi sulla struttura deformata o su quella indeformata;
- che il materiale abbia un comportamento linearmente elastico e non si abbia quindi plasticizzazione delle sezioni;
- che sia trascurabile la rigidezza torsionale delle travi e dei pilastri;
- che sia irrilevante l'interazione tra aste di telai distinti che incidono in uno stesso punto nodale.

Utilizzando quanto esposto nei due capitoli precedenti, si è previsto che i telai dell'insieme possano appartenere a due diversi tipi, ciascuno dei quali sarà nel seguito contraddistinto da una lettera:

tipo G – generico insieme piano di aste, definito mediante le coordinate dei nodi e la tabella di incidenza delle aste; in esso le aste sono individuate mediante numeri progressivi;

tipo R – telaio piano a maglie rettangolari o trapezie, la cui geometria è individuata indicando le luci delle campate e le altezze degli interpiani o dei singoli ritti; in esso le aste sono individuate facendo riferimento alla numerazione dei pilastri in carpenteria.

In tal modo si potrà utilizzare nella maggior parte dei casi il tipo R, che richiede una minore quantità di informazioni; si ricorrerà al tipo G solo in casi particolari, come telai con travi a ginocchio o con piano di posa a quota non costante.

Nell'ambito dei singoli telai si è mantenuta la possibilità di inserire travi di fondazione su suolo elastico alla Winkler o vincoli elasticamente cedevoli. Questa opportunità deve però essere sfruttata con molta cautela, perché il programma non impone la congruenza degli spostamenti verticali di nodi comuni a telai distinti e non può quindi cogliere l'effetto di cedimenti differenziali.

Il programma consente di risolvere più schemi di carico, in ciascuno dei quali sono presenti:

- carichi applicati direttamente agli impalcati (forze orizzontali F_x ed F_y e coppie M_z);
- carichi applicati ai telai (azioni concentrate nei nodi o carichi distribuiti sulle aste).

Si è mantenuta la possibilità di applicare azioni orizzontali ai singoli telai piani, anche se essa non verrà normalmente utilizzata perché nell'analisi sismica di edifici le azioni orizzontali che rappresentano l'inerzia delle masse in movimento vengono usualmente considerate come concentrate negli impalcati.

L'organizzazione complessiva del programma è descritta in dettaglio nel paragrafo successivo. È però opportuno evidenziare una scelta di base relativa alla gestione delle informazioni connesse ai telai piani dell'insieme. I telai vengono esaminati in sequenza a più riprese, prima per leggerne la geometria e determinarne la matrice di rigidezza, poi per leggere i dati relativi a ciascuno schema di carico e calcolarne le azioni di incastro. In queste fasi si determinano una serie di valori (relativi a geometria, carichi e matrici di rigidezza) che comportano un ingombro di memoria non trascurabile, tale da sconsigliare di mantenerli in memoria in attesa di un uso succes-

sivo. Il problema, analogo a quello già discusso nel quinto capitolo, è stato qui risolto memorizzando le informazioni in tre file sequenziali MATR.SPA, GEOM.SPA, CAR.SPA) generati dal programma e cancellati automaticamente al termine dell'elaborazione; se l'esecuzione viene prematuramente interrotta la cancellazione dei file non avviene, ma potrà essere fatta successivamente dall'utente.

2. Descrizione del programma

Il programma per la risoluzione di un insieme spaziale di telai piani è denominato TELSPA. Esso è costituito da un programma principale, che evidenzia in maniera sintetica le operazioni da esso compiute, ed una serie di routine, le quali richiamano a loro volta sia altre routine che le procedure contenute nei file MATRIX.BAS, SOLSIST.BAS, ASTA.BAS, PROCOM.BAS, PROGEN.BAS, PRORET.BAS e descritte nei capitoli precedenti. Il programma principale e le routine sono racchiuse nel file TELSPA.BAS.

Nel presente paragrafo si elencano tutti i blocchi logici che costituiscono il programma TELSPA, descrivendo in dettaglio quelli aggiunti o modificati rispetto ai programmi TELGEN e TELSPA.

A) Programma principale.

Il programma principale è diviso anche in questo caso in tre parti: dichiarazione delle procedure richiamate, dimensionamento degli array statici, richiamo ai blocchi logici nei quali il procedimento di soluzione è stato scomposto. Quest'ultima mostra l'organizzazione complessiva del programma, evidenziando le seguenti fasi:

- richiesta delle indicazioni operative generali;
- lettura e stampa dei dati relativi alla globalità dell'insieme spaziale di telai piani;
- elaborazione dei dati geometrici ed elastici relativi ai singoli telai (lettura e stampa dei dati geometrici, preparazione della matrice di rigidezza globale e di quella traslante, valutazione del contributo del telaio alla matrice di rigidezza traslante dell'insieme);
- triangolarizzazione della matrice di rigidezza traslante dell'insieme;
- lettura, stampa ed elaborazione dei carichi sugli impalcati;
- elaborazione dei dati di carico relativi ai singoli telai (lettura e stampa dei carichi, valutazione delle azioni di incastro globali e traslanti del

telaio e del loro contributo a quelle dell'insieme);
- risoluzione del sistema e stampa dei risultati globali relativi agli impalcati;
- valutazione e stampa dei risultati relativi ai singoli telai.

Le prime quattro fasi sono eseguite una sola volta, mentre le altre quattro vengono ripetute ciclicamente per consentire la risoluzione di più schemi di carico.

B) Dimensionamento degli array dinamici (routine `Dim...`).

C) Schermata di presentazione del programma (routine `Presentazione`).

D) Definizione delle unità di ingresso e uscita (procedura `DefinisceIO`).

E) Definizione delle modalità operative (procedura `DefinisceMO`).

F) Lettura e stampa dei dati geometrici ed elastici relativi all'intero complesso spaziale.

La routine `LeggeGeomSpa` legge da un file su disco le informazioni che definiscono l'intero complesso spaziale, secondo l'ordine descritto in dettaglio nel paragrafo 5; effettua anche il dimensionamento degli array dinamici relativi alla geometria dell'insieme ed alla tabella dei tipi di sezione. L'esecuzione viene interrotta se si riscontrano valori dei dati non ammissibili. La routine `StampaGeomSpa` invia le informazioni lette all'unità di uscita prescelta.

G) Lettura, stampa ed elaborazione dei dati relativi ai singoli telai.

La routine `LeggeGeomTel` legge dal file su disco le informazioni che definiscono il telaio piano, dipendenti dal tipo di telaio ed elencate in dettaglio nel paragrafo 5. Effettua il dimensionamento degli array dinamici relativi alla geometria del telaio e, se questo è di tipo `R`, dimensiona anche gli array necessari per la conversione dei dati geometrici nella forma valida per telai generici. L'esecuzione viene interrotta se si riscontrano valori dei dati non ammissibili.

La routine `ConverteGeometria` opera su telai di tipo `R`, trasformando i dati geometrici che descrivono il telaio a maglie rettangolari o trapezie nell'insieme di informazioni necessarie per analizzare in maniera generale un insieme piano di aste.

La routine `LeggeCorrispTravImp` legge dal file su disco i valori che indicano quale impalcato dell'insieme spaziale corrisponde a ciascun traverso inestensibile del telaio piano. Si noti nel caso di telai di tipo `R` la necessità di aver effettuato preliminarmente la conversione, durante la quale si individua il numero di traversi inestensibili che non coincide necessariamente con quello dei traversi geometrici.

Le routine `StampaGeomTel` e `Schema` provvedono rispettivamente ad inviare all'unità di uscita i dati letti per il singolo telaio ed a visualizzare sullo schermo lo schema geometrico del telaio stesso. Le routine `DefinisceIncognite` e `MatRigTel` forniscono la matrice di rigidezza globale del telaio; `TriangTel` ne ricava la matrice di rigidezza traslante e `FormaMatSpa` valuta il contributo dato alla matrice di rigidezza traslante dell'insieme. Infine la routine `WriteGeomTel` memorizza in file temporanei le informazioni relative a geometria e matrici da utilizzare in fasi successive dell'elaborazione.

H) Riduzione della matrice di rigidezza traslante dell'insieme ad una matrice triangolare (routine `TriangSpa`).

Tra le procedure di riduzione presentate nel capitolo 3 della seconda parte del testo viene qui utilizzata `Triang`, perché la matrice di rigidezza traslante dell'insieme spaziale è una matrice piena, memorizzata in un unico array bidimensionale.

I) Lettura, stampa ed elaborazione dei carichi applicati agli impalcati.

La routine `LeggeCarSpa` legge dal file dati su disco i carichi agenti direttamente sugli impalcati; `StampaCarSpa` invia queste informazioni all'unità di uscita; `SommaCarSpa` valuta il contributo di questi carichi al vettore azioni traslanti di incastro dell'insieme.

L) Lettura, stampa ed elaborazione dei carichi applicati ai singoli telai.

Dopo aver riletto dai file temporanei le informazioni relative a geometria e matrici del telaio mediante la routine `ReadGeomTel`, la routine `LeggeCarTel` preleva dal file dati i carichi agenti sul telaio piano e provvede a dimensionare gli array dinamici ad essi relativi. La routine `ConverteCarichi` opera per telai di tipo `R` trasformando le informazioni relative allo schema a maglie rettangolari o trapezie nei carichi necessari per analizzare in maniera generale un insieme piano di aste. La routine `StampaCarTel`

invia all'unità di uscita le informazioni lette. Le routine `TerminiNoti`, `TriangR`, `WriteCar`, `SommaCarTel` provvedono infine a determinare il vettore azioni di incastro globali e traslanti del telaio piano e il contributo fornito al vettore azioni traslanti di incastro dell'insieme, nonché a memorizzare in un file temporaneo le informazioni relative ai carichi da utilizzare in fasi successive dell'elaborazione.

M) Calcolo e stampa del vettore componenti di movimento degli impalcati.

La routine `RisolveSistema` effettua la seconda fase della risoluzione (sostituzione all'indietro) mediante la procedura Risolve, descritta nel terzo capitolo. La routine `StampaRisultatiSpa` invia all'unità di uscita i valori delle componenti di movimento degli impalcati così determinati.

N) Determinazione e stampa delle componenti di movimento e delle caratteristiche di sollecitazione dei singoli telai.

Dopo aver riletto dai file temporanei le informazioni relative a geometria, matrici e carichi del telaio mediante le routine `ReadGeomTel` e `ReadCar`, la routine `Spostamenti` determina prima le traslazioni dei traversi inestensibili a partire dal movimento degli impalcati, poi le componenti di movimento dei nodi interni completando il procedimento di sostituzione all'indietro mediate la procedura `Risolve.K2`.

La routine `CaratSollec` determina le azioni orizzontali sui traversi inestensibili come prodotto della matrice di rigidezza traslante per il vettore traslazioni del telaio; somma quindi le forze ottenute alle forze nodali assegnate come carichi sul telaio, in modo da tenerne conto nel valutare eventuali squilibri; determina infine le caratteristiche di sollecitazione nelle aste del telaio, visto genericamente come insieme piano di aste.

Se il telaio è di tipo `R`, la routine `ConverteRisult` riorganizza i valori precedenti secondo l'impostazione tipica del telaio a maglie rettangolari o trapezie. La routine `StampaRisultTel` provvede infine a inviare i risultati all'unità di uscita.

O) Segnalazione delle ipotesi di base del calcolo (routine `OutputIpotesi`).

Invia all'unità di uscita informazioni relative alle ipotesi di base del calcolo, segnalando i limiti di validità del modello di insieme spaziale di telai piani.

P) Scrittura, lettura e cancellazione di file temporanei.

Le routine utilizzate nella gestione dei file temporanei sono raggruppate nella parte finale del listato. I file `GEOM.SPA`, `MATR.SPA` e `CAR.SPA` contengono informazioni relative rispettivamente alla geometria, alle matrici ed ai carichi dei singoli telai piani, organizzate in maniera sequenziale. La routine `EraseFiles` cancella i file temporanei prima di terminare l'elaborazione. Le routine `Read...` e `Write...` si occupano rispettivamente di prelevare e di memorizzare i valori. Esse utilizzano prevalentemente le procedure `InputMat` e `WriteMat` del file `MATRIX.BAS` operando così sull'intero array; intervengono in maniera esplicita solo per gli array alfanumerici (incompatibili con tali procedure) e per le informazioni connesse ad aste e traversi del telaio generico (si noti che nella conversione dei valori relativi a telai di tipo `R` nella forma valida per telai generici gli array sono dimensionati in base al numero massimo possibile di aste e traversi, e non in base al loro numero reale).

3. Elenco delle variabili

Le variabili sono riunite in gruppi in base al loro significato ed al blocco di programma nel quale sono utilizzate prevalentemente o per la prima volta. Si riportano qui solo le variabili utilizzate nella gestione complessiva del programma e nei blocchi specifici dell'insieme spaziale di telai, rinviando all'analogo paragrafo dei capitoli precedenti per la descrizione delle variabili già usate nei programmi `TELGEN` e `TELRET`. Nell'ambito di ciascun gruppo le variabili sono elencate in ordine alfabetico. Per tutte le grandezze dimensionali sono riportate, tra parentesi, le unità di misura adottate.

Variabili utilizzate nella gestione generale del programma

`Contr`	indica se i dati devono essere stampati man mano che vengono letti per consentirne un controllo immediato
`DefEst`	indica se il calcolo è condotto tenendo conto della deformabilità estensionale delle aste
`DefTag`	indica se il calcolo è condotto tenendo conto della deformabilità a taglio delle aste
`ER`	indica un errore nei dati letti
`Squil`	indica la presenza di squilibrio nei nodi dopo la risoluzione del sistema

`TelRet` indica se tra i telai dello schema ve ne è almeno uno a maglie rettangolari o trapezie

`TelVCed` indica se nei telai dello schema vi è almeno un vincolo cedevole

`TelWnkl` indica se nei telai dello schema vi è almeno un'asta su suolo elastico alla Winkler

queste variabili di controllo possono assumere solo due valori convenzionali (0=no, 1=si)

`CAR` numero d'ordine dello schema di carico in esame

`FI$` nome del file che contiene i dati

`FO$` nome dell'unità cui si vuole inviare l'output (`CON`=schermo, `PRN`=stampante, oppure nome di un file)

`FT$` nome della directory da utilizzare per file temporanei

Variabili utilizzate nella gestione del programma, relative ai singoli telai piani

`AltVar` indica se in un telaio di tipo `R` vi è un interpiano con ritti ad altezza variabile

`ERpil` indica la presenza in un telaio di tipo `R` di ritti che interrompono la continuità verticale, incompatibili con l'ipotesi di pilastri inestensibili

`VCed` indica la presenza di vincoli elasticamente cedevoli

`Wnkl` indica la presenza di una trave di fondazione su suolo elastico alla Winkler

anche queste variabili di controllo possono assumere solo due valori convenzionali (0=no, 1=si)

Variabili utilizzate come indici degli array direttamente relativi o comunque connessi all'insieme spaziale

`K` indica il generico traverso inestensibile di un telaio

`PS` indica il generico impalcato

`R,C` indicano riga e colonna di un generico elemento della matrice di rigidezza traslante di un telaio

`T` indica il generico telaio piano

Variabili utilizzate nella definizione globale dell'insieme spaziale di telai piani

`Al`	angolo α formato dall'asse locale x' di un telaio con l'asse globale x (radianti)
`Alfa(T)`	angolo formato dall'asse locale x', relativo al telaio `T`, con l'asse globale x (gradi)
`C1,C2,C3`	colonne della matrice di rigidezza traslante dell'insieme spaziale di telai cui fornisce un contributo la colonna `C` della matrice di rigidezza traslante del singolo telaio
`CsAl`	coseno dell'angolo α tra gli assi x' e x
`Direz$`	direzione dell'asse locale x' o angolo formato da esso con l'asse globale x
`Dist`	distanza dell'asse locale x' di un telaio dall'origine del riferimento globale
`K11...K33`	contributi della matrice di rigidezza traslante del singolo telaio a quella dell'insieme spaziale
`Ks()`	matrice di rigidezza traslante dell'insieme spaziale di telai
`KZ`	numero di traversi inestensibili di un telaio
`PI(K)`	numero d'ordine dell'impalcato corrispondente al traverso inestensibile `K` di un telaio piano
`PSZ`	numero totale di impalcati
`R1,R2,R3`	righe della matrice di rigidezza traslante dell'insieme spaziale di telai cui fornisce un contributo la riga `R` della matrice di rigidezza traslante del singolo telaio
`SnAl`	seno dell'angolo α tra gli assi x' e x
`TT$(T)`	carattere che indica il tipo del telaio `T`; puo assumere convenzionalmente solo uno di questi valori: `G` = telaio di tipo generico (insieme piano di aste) `R` = telaio a maglie rettangolari o trapezie
`TZ`	numero totale di telai
`XO(T)`	ascissa dell'origine del sistema di riferimento locale del telaio `T`, valutata nel riferimento globale (m)
`YO(T)`	ordinata dell'origine del sistema di riferimento locale del telaio `T`, valutata nel riferimento globale (m)

Variabili utilizzate nella definizione dei carichi e nella risoluzione del sistema

ICS	indica l'esistenza di carichi sugli impalcati
ICT	indica l'esistenza di carichi sui singoli telai
	queste variabili possono assumere solo due valori convenzionali (0=non esistono, 1=esistono carichi di quel tipo)
Fsx(PS)	Componente in direzione x della forza applicata all'impalcato PS (kN)
Fsy(PS)	Componente in direzione y della forza applicata all'impalcato PS (kN)
Msz(PS)	Coppia torcente intorno all'asse z applicata all'impalcato PS (kNm)
Ra()	vettore azioni di incastro sui nodi interni di un telaio piano, dopo la triangolarizzazione
Rs()	vettore azioni traslanti di incastro perfetto dell'insieme spaziale di telai; dopo la risoluzione del sistema contiene le componenti di movimento degli impalcati
Rt()	vettore azioni traslanti di incastro perfetto di un telaio piano; dopo la risoluzione del sistema contiene le traslazioni dei traversi del telaio
St()	vettore azioni traslanti di un telaio piano, che contiene le forze orizzontali che competono al telaio dalla risoluzione del sistema relativo all'insieme spaziale
Xm(PS)	ascissa del punto di applicazione della forza agente sull'impalcato PS (m)
Ym(PS)	ordinata del punto di applicazione della forza agente sull'impalcato PS (m)

4. Codifica

```
'                                  TELSPA.BAS
'
' programma per il calcolo di insiemi spaziali di strutture piane intelaiate
' generiche oppure con maglie di forma rettangolare o trapezia
'
' tipi di telai:       G  =  generici
'                      R  =  a maglie rettangolari o trapezie
'
```

```
'  tipi di aste:        T  =  trave alla De Saint Venant, incastrata ai nodi
'
'  vincoli esterni:     vincoli rigidi o elasticamente cedevoli
'                       vincoli mutui allo spostamento orizzontale
'
'  carichi:             sul complesso spaziale:
'                          forza comunque orientata e momento torcente
'
'                       sulle strutture piane componenti:
'                          azioni concentrate sui nodi
'                          carichi uniformemente distribuiti sulle aste
'
'                                                          rev. 01.90

' ------------------ Dichiarazione delle procedure utilizzate ----------------------

' procedure del file MATRIX.BAS
DECLARE SUB AzzeraMat (A!())
DECLARE SUB InputMat (A!(), F!)
DECLARE SUB InserMat (A!(), B!(), RI!, CI!)
DECLARE SUB ProdottoMat (A!(), B!(), C!())
DECLARE SUB WriteMat (A!(), F!)

' procedure del file SOLSIST.BAS
DECLARE SUB Triang (A!(), IPVT!())
DECLARE SUB Triang.K2 (KTT!(), KTA!(), KAA!(), SB!, RID!)
DECLARE SUB Risolve (A!(), B!(), IPVT!())
DECLARE SUB Risolve.K2 (KTT!(), KTA!(), KAA!(), SB!, RID!, B!())

' procedure del file PROCOM.BAS
DECLARE SUB Aspetta (Sec!)
DECLARE SUB CarSoll (JZ!, X!(), Y!(), ZA!(), ZAT!(), ZI!(), E!, G!, E1!(), E2!(),
                     TA$(), NT!(), K!(), B!(), N!(), P!(), F!(), R!(), N1!(), T1!(),
                     M1!(), N2!(), T2!(), M2!())
DECLARE SUB DefinisceID (Prog$, FI$, FO$, FT$)
DECLARE SUB DefinisceMD (Prog$, DefEst!, DefTag!, Contr!)
DECLARE SUB DisegnaSchema (IZ!, JZ!, X!(), Y!(), E1!(), E2!())
DECLARE SUB FormaMat (E1!, E2!, N!(), Kg!(), NL!, SB!, KTT!(), KTA!(), KAA!())
DECLARE SUB Incognite (IZ!, JZ!, KZ!, E1!(), E2!(), V$(), NI!(), N!(), NL!, NZ!,SB!)
DECLARE SUB InputIntestaz (Contr!, Intest$)
DECLARE SUB InputModElast (Contr!, EL!, GL!, E!, G!)
DECLARE SUB InputSezioni (Contr!, LZ!, FS$(), ZB!(), ZH!(), ZA!(), ZI!(),ZAT!(),ER!)
DECLARE SUB MatAstaGlobale (TA$, X1!, X2!, Y1!, Y2!, A!, AT!, Iner!, K!, B!, E!, G!,
                            Lun!, SnAl!, CsAl!, Ld!(), Kg!())
DECLARE SUB OutputIntestaz (FI$, Intest$)
DECLARE SUB OutputModElast (DefTag!, EL!, GL!)
DECLARE SUB TermNoto (IZ!, JZ!, X!(), Y!(), ZA!(), ZI!(), E!, E1!(), E2!(), TA$(),
                      NT!(), K!(), B!(), N!(), P!(), F!(), R!())
DECLARE SUB VincoliCed (IZ!, N!(), KV!(), NL!, SB!, KTT!(), KTA!(), KAA!())

' procedure del file PROGEN.BAS
DECLARE SUB InputAsteGen (Contr!, JZ!, LZ!, E1!(), E2!(), NT!(), TA$(), Wnkl!, B!(),
                          K!(), ER!)
DECLARE SUB InputCarGen (Contr!, IZ!, JZ!, QN!, QA!, F!(), P!(), ER!)
DECLARE SUB InputNodiGen (Contr!, IZ!, X!(), Y!(), V$(), VCed!, KV!(), ER!)
DECLARE SUB InputTraversiGen (Contr!, KZ!, NI!())
DECLARE SUB OutputAsteGen (DefTag!, Wnkl!, JZ!, NT!(), TA$(), E1!(), E2!(), FS$(),
                           ZB!(), ZH!(), ZA!(), AT!(), ZI!(), B!(), K!())
```

```
DECLARE SUB OutputCarGen (CAR!, IZ!, JZ!, QN!, QA!, F!(), P!())
DECLARE SUB OutputNodiGen (IZ!, X!(), Y!(), V$(), VCed!, KV!())
DECLARE SUB OutputRisGen (IZ!, JZ!, KZ!, NL!, V$(), NI!(), N!(), R!(), M1!(), M2!(),
                          T1!(), T2!(), N1!(), N2!(), F!())

' procedure del file PRORET.BAS
DECLARE SUB ContrRitti (PZ!, RZ!, NP!(), NB!(), NR!(), ZA!(), ZI!(), ER!)
DECLARE SUB ConverteCarRet (RZ!, PZ!, Wnkl!, NB!(), NF!(), ZA!(), ZI!(), IV!, IM!,
                            IN!, IO!, Q!(), QF!(), MN!(), MNF!(), FYN!(), FYF!(),
                            FX!(), KZ!, V$(), NI!(), P!(), F!())
DECLARE SUB ConverteGeomRet (PZ!, RZ!, Wnkl!, DefEst!, DL!(), H!(), NB!(), NR!(),
                             NF!(), BF!(), KF!(), ZA!(), ZI!(), IZ!, JZ!, KZ!, X!(),
                             Y!(), V$(), NI!(), E1!(), E2!(), TA$(),NT!(),B!(),K!())
DECLARE SUB ConverteRisRet (RZ!, PZ!, Wnkl!, DefEst!, NB!(), NR!(), NF!(), ZA!(),
                            ZI!(), D!(), MS!(), MD!(), TS!(), TD!(), MP!(), MT!(),
                            TR!(), SN!(), MFS!(), MFD!(), TFS!(), TFD!(), KZ!,NI!(),
                            N!(), R!(), F!(), M1!(), M2!(), T1!(), T2!(), N1!())
DECLARE SUB InputAltezze (Contr!, PZ!, RZ!, AltVar!, HI!(), H!())
DECLARE SUB InputAsteRet (Contr!, Wnkl!, PZ!, RZ!, LZ!, NP!(), NB!(), NR!(), NF!(),
                          BF!(), KF!(), ER!)
DECLARE SUB InputCarRet (Contr!, Wnkl!, IV!, IM!, IN!, IO!, PZ!, RZ!, KZ!, NP!(),
                         V$(), NB!(), NF!(), ZA!(), ZI!(), Q!(), QF!(),MN!(),MNF!(),
                         FYN!(), FYF!(), FX!())
DECLARE SUB InputLuci (Contr!, Wnkl!, RZ!, DL!())
DECLARE SUB InputNumRitti (Contr!, RZ!, NP!())
DECLARE SUB OutputAsteRet (PZ!, RZ!, Wnkl!, NP!(), DL!(), NB!(), NR!(), NF!(),BF!(),
                           KF!(), FS$(), ZB!(), ZH!(), ZI!())
DECLARE SUB OutputCarRet (CAR!, Wnkl!, IV!, IM!, IN!, IO!, PZ!, RZ!, KZ!, NP!(),
                          V$(), NB!(), NF!(), ZA!(), ZI!(), Q!(), QF!(), MN!(),
                          MNF!(), FYN!(), FYF!(), FX!())
DECLARE SUB OutputPiani (AltVar!, PZ!, RZ!, NP!(), HI!(), H!())
DECLARE SUB OutputRisRet (RZ!, PZ!, Wnkl!, NP!(), NB!(), NR!(), NF!(), ZA!(), ZI!(),
                          D!(), MS!(), MD!(), TS!(), TD!(), MP!(), MT!(), TR!(),
                          SN!(), MFS!(), MFD!(), TFS!(), TFD!(), IZ!, KZ!, NL!,V$(),
                          N!(), R!(), F!())
DECLARE SUB OutputTabSez (DefTag!, LZ!, FS$(), ZB!(), ZH!(), ZA!(), ZAT!(), ZI!())


' ------------------ Dimensionamento delle variabili con indice --------------------

OPTION BASE 1             ' definisce l'estremo inferiore dell'indice degli array
DIM Ld(6, 6), Kg(6, 6)    ' array statici
                          ' array dinamici: sono dimensionati in apposite routine


' --------------------------- Programma principale --------------------------------

' definizione di costanti
CONST PiGr = 3.141592654#

GOSUB Presentazione
CALL DefinisceIO("SPA", FI$, FO$, FT$)
CALL DefinisceMD("SPA", DefEst, DefTag, Contr)

OPEN "I", #1, FI$
OPEN "O", #2, FO$
GOSUB LeggeGeomSpa
GOSUB StampaGeomSpa
OPEN "O", #3, FT$ + "MATR.SPA"
```

```
OPEN "O", #4, FT$ + "GEOM.SPA"
FOR T = 1 TO TZ
   GOSUB LeggeGeomTel
   GOSUB ConverteGeometria
   GOSUB LeggeCorrispTravImp
   GOSUB StampaGeomTel
   GOSUB Schema
   GOSUB DefinisceIncognite
   GOSUB MatRigTel
   GOSUB TriangTel
   GOSUB FormaMatSpa
   GOSUB WriteGeomTel
NEXT T
CLOSE #3
CLOSE #4
GOSUB TriangSpa

CAR = 0
DO WHILE NOT EOF(1)
   CAR = CAR + 1
   INPUT #1, ICS, ICT
   IF ICS = 1 THEN
      GOSUB LeggeCarSpa
      GOSUB StampaCarSpa
   END IF
   GOSUB SommaCarSpa
   IF ICT = 1 THEN
      OPEN "I", #3, FT$ + "MATR.SPA"
      OPEN "I", #4, FT$ + "GEOM.SPA"
      OPEN "O", #5, FT$ + "CAR.SPA"
      FOR T = 1 TO TZ
         GOSUB ReadGeomTel
         GOSUB LeggeCarTel
         GOSUB ConverteCarichi
         GOSUB StampaCarTel
         GOSUB TerminiNoti
         GOSUB TriangR
         GOSUB WriteCar
         GOSUB SommaCarTel
      NEXT T
      CLOSE #3
      CLOSE #4
      CLOSE #5
   END IF
   GOSUB RisolveSistema
   GOSUB StampaRisultSpa

   OPEN "I", #3, FT$ + "MATR.SPA"
   OPEN "I", #4, FT$ + "GEOM.SPA"
   IF ICT = 1 THEN OPEN "I", #5, FT$ + "CAR.SPA"
   FOR T = 1 TO TZ
      GOSUB ReadGeomTel
      GOSUB ReadCar
      GOSUB Spostamenti
      GOSUB CaratSollec
      GOSUB ConverteRisult
      GOSUB StampaRisultTel
   NEXT T
   GOSUB OutputIpotesi
```

```
   CLOSE #3
   CLOSE #4
   IF ICT = 1 THEN CLOSE #5
LOOP
GOSUB EraseFiles
PRINT
PRINT "elaborazione terminata"
PRINT
END

' ------------------------------ DimXXXXXX ------------------------------------
'  Gruppo di routine per il dimensionamento degli array dinamici
' -----------------------------------------------------------------------------

' ................................ DimCarGen .....................................
'  Routine per il dimensionamento degli array relativi a carichi
'
DimCarGen:
   REDIM F(IZ, 3), P(JZ, 2)
RETURN

' ................................ DimCarRet .....................................
'  Routine per il dimensionamento degli array relativi a carichi
'
DimCarRet:
   REDIM Q(RZ - 1, PZ), MN(RZ, PZ), FYN(RZ, PZ), FX(KZ)
   IF Wnkl = 1 THEN REDIM QF(0 TO RZ), MNF(0 TO RZ + 1), FYF(0 TO RZ + 1)
RETURN

' ................................ DimGeomGen ....................................
'  Routine per il dimensionamento degli array relativi alla geometria
'
DimGeomGen:
   REDIM X(IZ), Y(IZ), V$(IZ), KV(IZ, 3), NI(IZ + KZ), PI(KZ)
   REDIM E1(JZ), E2(JZ), TA$(JZ), NT(JZ), B(JZ), K(JZ)
   REDIM N(IZ, 3)
RETURN

' ................................ DimGeomRet ....................................
'  Routine per il dimensionamento degli array relativi alla geometria
'
DimGeomRet:
   REDIM NP(RZ), HI(PZ), H(RZ, PZ), DL(0 TO RZ)
   REDIM NB(RZ - 1, PZ), NR(RZ, PZ)
   IF Wnkl = 1 THEN REDIM NF(0 TO RZ), BF(0 TO RZ), KF(0 TO RZ)
RETURN

' ................................ DimGeomSpa ....................................
'  Routine per il dimensionamento degli array relativi alla geometria
'  dell'insieme spaziale di telai piani
'
DimGeomSpa:
   DIM TT$(TZ), XO(TZ), YO(TZ), Alfa(TZ)
   DIM Xm(PSZ), Ym(PSZ), Fsx(PSZ), Fsy(PSZ), Msz(PSZ)
   DIM Ks(3 * PSZ, 3 * PSZ), Rs(3 * PSZ), IPVT(3 * PSZ)
RETURN

' ................................ DimMatRig .....................................
```

```
' Routine per il dimensionamento di matrice di rigidezza e vettore carico
'
DimMatRig:
   IF NL = 0 THEN
      REDIM KTT(0 TO 0, 0 TO 0), KTA(0 TO 0, NZ - NL)
   ELSE
      REDIM KTT(NL, NL), KTA(NL, NZ - NL)
   END IF
   REDIM KAA(NZ - NL, SB), R(NZ), Rt(KZ), Ra(NZ - KZ)
RETURN

' ..... DimRisGen .....
' Routine per il dimensionamento degli array relativi a risultati
'
DimRisGen:
   REDIM St(KZ), M1(JZ), M2(JZ), T1(JZ), T2(JZ), N1(JZ), N2(JZ)
RETURN

' ..... DimRisRet .....
' Routine per il dimensionamento degli array relativi a risultati
'
DimRisRet:
   REDIM D(KZ), MD(RZ - 1, PZ), MS(RZ - 1, PZ), TD(RZ - 1, PZ), TS(RZ - 1, PZ)
   REDIM MP(RZ, PZ), MT(RZ, PZ), TR(RZ, PZ), SN(RZ, PZ)
   IF Wnkl = 1 THEN REDIM MFD(0 TO RZ), MFS(0 TO RZ), TFD(0 TO RZ), TFS(0 TO RZ)
RETURN

' ..... DimTabSez .....
' Routine per il dimensionamento degli array relativi ai tipi di sezione
'
DimTabSez:
   DIM FS$(LZ), ZB(LZ), ZH(LZ), ZA(LZ), ZI(LZ), ZAT(LZ)
RETURN


' ---------- Presentazione ----------
' Routine per la visualizzazione della schermata di presentazione del
' programma
' ------------------------------------------------------------------------------
Presentazione:
   CLS
   PRINT TAB(31); "TELSPA - rev. 01.90"
   PRINT
   PRINT "       Le ipotesi di calcolo e le modalita' operative del programma"
   PRINT "       TELSPA sono riportate nella terza parte, capitolo 7, del libro:"
   PRINT "       A.Ghersi, R.Coraggio, Il personal computer nel calcolo di edifici."
   PRINT "       Analisi matriciale di strutture intelaiate, CUEN, Napoli, 1990."
   PRINT
   PRINT
   PRINT
RETURN


' ---------- Completata ----------
' Routine per l'indicazione della fine di una fase
' ------------------------------------------------------------------------------
Completata:
   LOCATE CSRLIN - 1, 44
   PRINT "completata            "
```

```
RETURN


' ------------------------------ InCorso ---------------------------------------
'  Routine per l'indicazione dell'inizio di una fase
' ------------------------------------------------------------------------------
InCorso:
   LOCATE CSRLIN - 1, 44
   PRINT "in corso"
RETURN


' ------------------------------ NomeProgramma ---------------------------------
'  Routine per la visualizzazione dell'intestazione del programma
' ------------------------------------------------------------------------------
NomeProgramma:
   CLS
   PRINT "TELSPA - rev. 01.90"
   PRINT
RETURN


' ------------------------------ LeggeGeomSpa ----------------------------------
'  Routine per la lettura dei dati geometrici ed elastici dell'insieme spaziale
' ------------------------------------------------------------------------------
LeggeGeomSpa:
   GOSUB NomeProgramma
   PRINT "lettura geometria insieme spaziale": GOSUB InCorso

   ER = 0
   CALL InputIntestaz(Contr, IT$)
   GOSUB InputIndSpa
   GOSUB DimGeomSpa
   GOSUB DimTabSez
   GOSUB InputPosTelai
   CALL InputModElast(Contr, EL, GL, E, G)
   CALL InputSezioni(Contr, LZ, FS$(), ZB(), ZH(), ZA(), ZI(), ZAT(), ER)
   IF ER = 1 THEN END

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN

' ..............................  InputIndSpa  ..................................
'  Routine per la lettura delle indicazioni generali dell'insieme spaziale;
'
InputIndSpa:
   INPUT #1, TZ, PSZ, LZ
   IF Contr = 1 THEN
      PRINT #2, "numero di telai:"; TZ
      PRINT #2, "numero di impalcati:"; PSZ
      PRINT #2, "numero di tipi di sezione:"; LZ
   END IF
RETURN

' ..............................  InputPosTelai  ................................
'  Routine per la lettura della posizione dei telai piani che compongono
'  l'insieme spaziale
'
InputPosTelai:
```

```
   TelRet = 0
   TelVCed = 0
   TelWnkl = 0
   FOR T = 1 TO TZ
      INPUT #1, TT$(T), XO(T), YO(T), Direz$
      TT$(T) = UCASE$(TT$(T))
      IF TT$(T) = "R" THEN TelRet = 1
      SELECT CASE Direz$
         CASE "X", "x"
            Alfa(T) = 0
         CASE "Y", "y"
            Alfa(T) = 90
         CASE ELSE
            Alfa(T) = VAL(Direz$)
      END SELECT
      IF Contr = 1 THEN
         PRINT #2, "Telaio: "; T
         PRINT #2, "   Tipo:"; TT$(T),
         PRINT #2, "x ="; XO(T), "y ="; YO(T), "Direzione (o angolo) ="; Direz$
      END IF
      IF TT$(T) <> "G" AND TT$(T) <> "R" THEN
         ER = 1
         PRINT "Il codice che identifica la tipologia del telaio"; T;
         PRINT " non e' ammissibile"
      END IF
   NEXT T
RETURN


' -------------------------------- StampaGeomSpa ---------------------------------
'  Routine per la stampa dei dati geometrici ed elastici dell'insieme spaziale
' --------------------------------------------------------------------------------
StampaGeomSpa:
   PRINT "stampa  geometria insieme spaziale"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   CALL OutputIntestaz(FI$, IT$)
   GOSUB OutputPosTelai
   CALL OutputTabSez(DefTag, LZ, FS$(), ZB(), ZH(), ZA(), ZAT(), ZI())
   CALL OutputModElast(DefTag, EL, GL)

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN

' ................................ OutputPosTelai ................................
'  Routine per la stampa della posizione dei telai piani che compongono
'  l'insieme spaziale
'
OutputPosTelai:
   PRINT #2, "DATI GEOMETRICI DELL'INSIEME SPAZIALE"
   PRINT #2,
   PRINT #2, "telaio   tipo    x (m)     y (m)   angolo (gradi)"
   PRINT #2,
   FOR T = 1 TO TZ
      PRINT #2, USING " ###      \\  "; T, TT$(T);
```

```
      PRINT #2, USING "###.##      ###.##        ###.##"; XO(T), YO(T), Alfa(T)
   NEXT T
RETURN


' ------------------------------- LeggeGeomTel ---------------------------------
' Routine per la lettura dei dati geometrici delle strutture piane componenti
' l'insieme spaziale
' ------------------------------------------------------------------------------
LeggeGeomTel:
   PRINT
   PRINT "telaio n."; T; "     tipo: "; TT$(T)
   PRINT "lettura dati geometrici": GOSUB InCorso

   ER = 0
   SELECT CASE TT$(T)
      CASE "G"
         ' telaio generico
         GOSUB InputIndGen
         GOSUB DimGeomGen
         CALL InputNodiGen(Contr, IZ, X(), Y(), V$(), VCed, KV(), ER)
         CALL InputTraversiGen(Contr, KZ, NI())
         CALL InputAsteGen(Contr, JZ, LZ, E1(), E2(), NT(), TA$(), Wnkl, B(),K(),ER)
      CASE "R"
         ' telaio a maglie trapezie
         GOSUB InputIndRet
         IZ = RZ * (PZ + 1) + 2
         JZ = (2 * RZ - 1) * PZ + RZ + 1      ' massimo numero di aste possibile
         KZ = INT(RZ * PZ / 2)                ' massimo numero di traversi possibile
         VCed = 0
         GOSUB DimGeomRet
         GOSUB DimGeomGen: REDIM KV(1, 1)
         CALL InputNumRitti(Contr, RZ, NP())
         CALL InputAltezze(Contr, PZ, RZ, AltVar, HI(), H())
         CALL InputLuci(Contr, Wnkl, RZ, DL())
         CALL InputAsteRet(Contr, Wnkl, PZ, RZ, LZ,NP(),NB(),NR(),NF(),BF(),KF(),ER)
         IF ER = 1 THEN END
         IF DefEst = 0 THEN CALL ContrRitti(PZ, RZ, NP(), NB(), NR(), ZA(), ZI(),ER)
   END SELECT
   IF ER = 1 THEN END
   IF VCed = 1 THEN TelVCed = 1
   IF Wnkl = 1 THEN TelWnkl = 1

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN

' ................................ InputIndGen ..................................
' Routine per la lettura delle indicazioni generali del telaio di tipo G
'
InputIndGen:
   INPUT #1, IZ, JZ, KZ
   IF Contr = 1 THEN
      PRINT #2, "numero di nodi:"; IZ
      PRINT #2, "numero di aste:"; JZ
      PRINT #2, "numero di traversi orizzontali inestensibili:"; KZ
   END IF
RETURN

' ................................ InputIndRet ..................................
```

```
'  Routine per la lettura delle indicazioni generali del telaio di tipo R
'
InputIndRet:
   INPUT #1, RZ, PZ, Wnkl
   IF Contr = 1 THEN
      PRINT #2, "numero di pilastri:"; RZ
      PRINT #2, "numero di piani:"; PZ
      IF Wnkl = 1 THEN PRINT #2, "e' presente la trave di fondazione"
   END IF
RETURN


' ------------------------------ ConverteGeometria ------------------------------
'  Routine per la conversione dei dati geometrici degli schemi piani
'  a maglie trapezie
' -------------------------------------------------------------------------------
ConverteGeometria:
   IF TT$(T) = "G" THEN RETURN
   PRINT "conversione dati geometrici": GOSUB InCorso

   SELECT CASE TT$(T)
      CASE "R"
         ' telaio a maglie trapezie
         CALL ConverteGeomRet(PZ, RZ, Wnkl, DefEst, DL(), H(), NB(), NR(), NF(),
                              BF(), KF(), ZA(), ZI(), IZ, JZ, KZ, X(), Y(), V$(),
                              NI(), E1(), E2(), TA$(), NT(), B(), K())
         REDIM PI(KZ)
   END SELECT

   GOSUB Completata
RETURN


' ------------------------------ LeggeCorrispTravImp ----------------------------
'  Routine per la lettura e stampa della corrispondenza traversi impalcati
' -------------------------------------------------------------------------------
LeggeCorrispTravImp:
   PRINT "lettura corrisp. traversi-impalcati": GOSUB InCorso

   ER = 0
   FOR K = 1 TO KZ
      INPUT #1, PI(K)
      IF Contr = 1 THEN
         PRINT #2, "traverso"; K; " appartenente all'impalcato"; PI(K)
      END IF
      IF PI(K) < 1 OR PI(K) > PSZ THEN
         PRINT "Il numero d'ordine"; PI(K);
         PRINT " dell'impalcato corrispondente al traverso"; K;" non e' accettabile"
         ER = 1
      END IF
   NEXT K
   IF ER = 1 THEN END

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN


' ------------------------------ StampaGeomTel ----------------------------------
'  Routine per la stampa dei dati geometrici delle strutture piane componenti
```

```
' l'insieme spaziale
' ------------------------------------------------------------------------------
StampaGeomTel:
   PRINT "stampa  dati geometrici"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "TELAIO N."; T
   PRINT #2,
   SELECT CASE TT$(T)
      CASE "G"
         ' telaio generico
         CALL OutputNodiGen(IZ, X(), Y(), V$(), VCed, KV())
         CALL OutputAsteGen(DefTag, Wnkl, JZ, NT(), TA$(), E1(), E2(), FS$(), ZB(),
                            ZH(), ZA(), ZAT(), ZI(), B(), K())
      CASE "R"
         ' telaio a maglie trapezie
         CALL OutputPiani(AltVar, PZ, RZ, NP(), HI(), H())
         CALL OutputAsteRet(PZ, RZ, Wnkl, NP(), DL(), NB(), NR(), NF(), BF(), KF(),
                            FS$(), ZB(), ZH(), ZI())
   END SELECT
   '
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "CORRISPONDENZA TRA TRAVERSI ED IMPALCATI"
   PRINT #2,
   PRINT #2, "traverso       impalcato"
   FOR K = KZ TO 1 STEP -1
      PRINT #2, USING "  ###            ###"; K, PI(K)
   NEXT K

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN


' ------------------------------- Schema ---------------------------------------
' Routine che visualizza lo schema geometrico
' ------------------------------------------------------------------------------
Schema:
   CALL DisegnaSchema(IZ, JZ, X(), Y(), E1(), E2())
   GOSUB NomeProgramma
   PRINT "telaio n."; T
RETURN


' ------------------------------ DefinisceIncognite ----------------------------
' Routine per la numerazione delle incognite  e la valutazione della semibanda
' (gli array che costituiscono la matrice di rigidezza vengono ridimensionati)
' ------------------------------------------------------------------------------
DefinisceIncognite:
   PRINT "numerazione incognite": GOSUB InCorso

   CALL Incognite(IZ, JZ, KZ, E1(), E2(), V$(), NI(), N(), NL, NZ, SB)
   GOSUB DimMatRig

   GOSUB Completata
```

```
   PRINT "- numero totale di incognite"; NZ;
   IF NL <> 0 THEN PRINT " (tra cui"; NL; "spostamenti di traversi)";
   PRINT
   PRINT "- ampiezza della semibanda  "; SB
RETURN


' ------------------------------- MatRigTel ------------------------------------
'  Routine per la costruzione della matrice di rigidezza delle strutture piane
'  componenti l'insieme spaziale
' ------------------------------------------------------------------------------
MatRigTel:
   PRINT "costruzione matrice rigidezza": GOSUB InCorso

   FOR J = 1 TO JZ
      L = NT(J)
      CALL MatAstaGlobale(TA$(J), X(E1(J)), X(E2(J)),Y(E1(J)),Y(E2(J)),ZA(L),ZAT(L),
                          ZI(L), K(J)*1000, B(J), E, G, Lun, SnAl, CsAl, Ld(), Kg())
      CALL FormaMat(E1(J), E2(J), N(), Kg(), NL, SB, KTT(), KTA(), KAA())
      LOCATE CSRLIN - 1, 44
      PRINT USING "eseguita per il ### %"; J / JZ * 100
   NEXT J
   IF VCed = 1 THEN CALL VincoliCed(IZ, N(), KV(), NL, SB, KTT(), KTA(), KAA())

   GOSUB Completata
RETURN


' ------------------------------- TriangTel ------------------------------------
'  Routine che triangolarizza la matrice di rigidezza delle strutture piane
'  componenti l'insieme spaziale
' ------------------------------------------------------------------------------
TriangTel:
   PRINT "triangolar. matrice rigidezza": GOSUB InCorso

   CALL Triang.K2(KTT(), KTA(), KAA(), SB, 1)
   GOSUB CompletaMat

   GOSUB Completata
RETURN

' ................................ CompletaMat ..................................
'  Routine per il riempimento del triangolo inferiore della sottomatrice Ktt
'
CompletaMat:
   FOR R = 2 TO KZ
      FOR C = 1 TO R - 1
         KTT(R, C) = KTT(C, R)
      NEXT C
   NEXT R
RETURN


' ------------------------------- FormaMatSpa ----------------------------------
'  Routine per l'inserimento dei termini delle matrici di rigidezza laterali
'  delle strutture piane componenti nella matrice di rigidezza dell'insieme
'  spaziale
' ------------------------------------------------------------------------------
FormaMatSpa:
```

```
   PRINT "inserimento termini nella matrice spaziale": GOSUB InCorso

   ' parametri geometrici del telaio
   Al = Alfa(T) * PiGr / 180
   CsAl = COS(Al)
   SnAl = SIN(Al)
   Dist = XO(T) * SnAl - YO(T) * CsAl
   ' inserisce termini nella matrice di rigidezza dell'insieme spaziale
   FOR R = 1 TO KZ
      FOR C = 1 TO KZ
         R1 = PI(R): R2 = PI(R) + PSZ: R3 = PI(R) + 2 * PSZ
         C1 = PI(C): C2 = PI(C) + PSZ: C3 = PI(C) + 2 * PSZ
         K11 = KTT(R, C) * CsAl ^ 2
         K12 = KTT(R, C) * SnAl * CsAl
         K13 = KTT(R, C) * Dist * CsAl
         K22 = KTT(R, C) * SnAl ^ 2
         K23 = KTT(R, C) * Dist * SnAl
         K33 = KTT(R, C) * Dist ^  2
         Ks(R1, C1) = Ks(R1, C1) + K11
         Ks(R1, C2) = Ks(R1, C2) + K12
         Ks(R1, C3) = Ks(R1, C3) + K13
         Ks(R2, C1) = Ks(R2, C1) + K12
         Ks(R2, C2) = Ks(R2, C2) + K22
         Ks(R2, C3) = Ks(R2, C3) + K23
         Ks(R3, C1) = Ks(R3, C1) + K13
         Ks(R3, C2) = Ks(R3, C2) + K23
         Ks(R3, C3) = Ks(R3, C3) + K33
      NEXT C
   NEXT R

   GOSUB Completata
RETURN


' ------------------------------ TriangSpa ------------------------------------
'  Routine che triangolarizza la matrice di rigidezza dell'insieme spaziale
' -----------------------------------------------------------------------------
TriangSpa:
   PRINT "triangolar. matrice rigidezza spaziale": GOSUB InCorso

   CALL Triang(Ks(), IPVT())

   GOSUB Completata
RETURN


' ------------------------------ LeggeCarSpa ----------------------------------
' Routine per la lettura dei dati che definiscono i carichi agenti
' sull'insieme spaziale
' -----------------------------------------------------------------------------
LeggeCarSpa:
   GOSUB NomeProgramma
   PRINT "    Schema di carico n."; CAR
   PRINT "lettura dati di carico insieme spaziale": GOSUB InCorso

   FOR PS = PSZ TO 1 STEP -1
      INPUT #1, Xm(PS), Ym(PS)
      INPUT #1, Fsx(PS), Fsy(PS), Msz(PS)
      IF Contr = 1 THEN
```

```
        PRINT #2, "impalcato"; PS
        PRINT #2, "  x0 ="; Xm(PS), "y0 ="; Ym(PS)
        PRINT #2, "  Fx ="; Fsx(PS), "Fy ="; Fsy(PS), "Mz ="; Msz(PS)
      END IF
   NEXT PS

   IF NOT (Contr = 1 AND FO$ = "CON") THEN GOSUB Completata
RETURN


' ------------------------------ StampaCarSpa ------------------------------------
'  Routine per la stampa dei carichi agenti sull'insieme spaziale
' --------------------------------------------------------------------------------
StampaCarSpa:
   PRINT "stampa dati di carico insieme spaziale"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "SCHEMA DI CARICO N."; CAR
   PRINT #2,
   PRINT #2, "CARICHI AGENTI SULL'INSIEME SPAZIALE"
   PRINT #2,
   PRINT #2, "Impalcato   Xm (m)     Ym (m)     Fx (kN)    Fy (kN)    Mz (kNm)"
   PRINT #2,
   FOR PS = PSZ TO 1 STEP -1
      PRINT #2, USING "   ###     ###.##     ###.##"; PS, Xm(PS), Ym(PS);
      PRINT #2, USING "     ###.###    ###.###     ###.###"; Fsx(PS),Fsy(PS),Msz(PS)
   NEXT PS

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN


' ------------------------------ SommaCarSpa -------------------------------------
'  Routine che valuta le azioni traslanti di incastro dovute
'  ai carichi applicati in un punto prefissato di ciascun impalcato
' --------------------------------------------------------------------------------
SommaCarSpa:
   IF FO$ = "CON" THEN GOSUB NomeProgramma
   PRINT "valutazione azioni carichi spaziali": GOSUB InCorso

   IF ICS = 1 THEN
      FOR PS = 1 TO PSZ
         Rs(PS) = Fsx(PS)
         Rs(PS + PSZ) = Fsy(PS)
         Rs(PS + 2 * PSZ) = Msz(PS) + Fsy(PS) * Xm(PS) - Fsx(PS) * Ym(PS)
      NEXT PS
   ELSE
      CALL AzzeraMat(Rs())
   END IF

   GOSUB Completata
RETURN
```

```
' ------------------------------ LeggeCarTel ----------------------------------
'  Routine per la lettura dei carichi nodali e sulle aste per le singole
'  strutture piane componenti l'insieme spaziale
' -----------------------------------------------------------------------------
LeggeCarTel:
   PRINT "lettura dati di carico": GOSUB InCorso

   SELECT CASE TT$(T)
      CASE "G"
         ' telaio generico
         GOSUB DimCarGen
         CALL InputCarGen(Contr, IZ, JZ, QN, QA, F(), P(), ER)
      CASE "R"
         ' telaio a maglie trapezie
         GOSUB DimCarRet
         GOSUB DimCarGen
         CALL InputCarRet(Contr, Wnkl, IV, IM, IN, IO, PZ, RZ, KZ, NP(), V$(), NB(),
                          NF(), ZA(), ZI(), Q(), QF(), MN(), MNF(),FYN(),FYF(),FX())
   END SELECT

   GOSUB Completata
RETURN


' ------------------------------ ConverteCarichi ------------------------------
'   Routine per la conversione dei dati di carico degli schemi piani
'   particolari
' -----------------------------------------------------------------------------
ConverteCarichi:
   IF TT$(T) = "G" THEN RETURN
   PRINT "conversione dati di carico": GOSUB InCorso

   SELECT CASE TT$(T)
      CASE "R"
         CALL ConverteCarRet(RZ, PZ, Wnkl, NB(), NF(), ZA(), ZI(), IV, IM, IN, IO,
                             Q(), QF(), MN(), MNF(), FYN(), FYF(), FX(), KZ, V$(),
                             NI(), P(), F())
   END SELECT

   GOSUB Completata
RETURN


' ------------------------------ StampaCarTel ---------------------------------
'  Routine per la stampa dei carichi agenti sulle singole strutture piane
'  componenti l'insieme spaziale
' -----------------------------------------------------------------------------
StampaCarTel:
   PRINT "stampa dati di carico"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   IF (TT$(T) = "G" AND QN + QA > 0) OR (TT$(T) ="R" AND IV + IM + IN + IO > 0) THEN
      PRINT #2, : PRINT #2, : PRINT #2,
      PRINT #2, "TELAIO N."; T, "SCHEMA DI CARICO N."; CAR
   END IF
   SELECT CASE TT$(T)
      CASE "G"
         ' telaio generico
          CALL OutputCarGen(CAR, IZ, JZ, QN, QA, F(), P())
```

```
      CASE "R"
         ' telaio a maglie trapezie
         CALL OutputCarRet(CAR, Wnkl, IV, IM, IN, IO, PZ, RZ, KZ, NP(), V$(), NB(),
                           NF(), ZA(), ZI(), Q(), QF(), MN(),MNF(),FYN(),FYF(),FX())
   END SELECT

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN


' ------------------------------ TerminiNoti ------------------------------------
'  Routine per il calcolo del vettore termini noti
' -------------------------------------------------------------------------------
TerminiNoti:
   PRINT "preparazione termini noti": GOSUB InCorso

   CALL TermNoto(IZ, JZ, X(), Y(), ZA(), ZI(), E, E1(), E2(), TA$(), NT(), K(), B(),
                 N(), P(), F(), R())

   GOSUB Completata
RETURN


' ------------------------------ TriangR ----------------------------------------
'  Routine per l'applicazione al vettore termini noti R delle trasformazioni
'  eseguite sulla matrice di rigidezza
' -------------------------------------------------------------------------------
TriangR:
   PRINT "triangolarizzazione termini noti": GOSUB InCorso

   CALL Risolve.K2(KTT(), KTA(), KAA(), SB, 1, R())
   FOR R = 1 TO NL
      Rt(R) = R(R)
   NEXT R
   FOR R = NL + 1 TO NZ
      Ra(R - NL) = R(R)
   NEXT R

   GOSUB Completata
RETURN


' ------------------------------ SommaCarTel ------------------------------------
'  Routine che aggiunge alle azioni traslanti di incastro sull'insieme spaziale
'  il contributo fornito dalle azioni traslanti di incastro del telaio piano
'  (conseguenti a dissimmetrie geometriche o di carico)
' -------------------------------------------------------------------------------
SommaCarTel:
   PRINT "somma azioni traslanti del telaio": GOSUB InCorso

   ' parametri caratteristici telaio
   Al = Alfa(T) * PiGr / 180
   CsAl = COS(Al)
   SnAl = SIN(Al)
   Dist = XO(T) * SnAl - YO(T) * CsAl
```

```
   FOR K = 1 TO KZ
      Rs(PI(K)) = Rs(PI(K)) + Rt(K) * CsAl
      Rs(PI(K) + PSZ) = Rs(PI(K) + PSZ) + Rt(K) * SnAl
      Rs(PI(K) + 2 * PSZ) = Rs(PI(K) + 2 * PSZ) + Rt(K) * Dist
   NEXT K

   GOSUB Completata
RETURN


' ------------------------------ RisolveSistema ---------------------------------
'  Routine per il calcolo delle componenti di movimento degli impalcati
' -------------------------------------------------------------------------------
RisolveSistema:
   PRINT "soluzione sistema": GOSUB InCorso

   CALL Risolve(Ks(), Rs(), IPVT())

   GOSUB Completata
RETURN


' ------------------------------ StampaRisultSpa --------------------------------
'  Routine per la stampa dei risultati dell'insieme spaziale
' -------------------------------------------------------------------------------
StampaRisultSpa:
   PRINT "stampa risultati insieme spaziale"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "RISULTATI"
   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "COMPONENTI DI MOVIMENTO DEGLI IMPALCATI"
   PRINT #2, : PRINT #2,
   PRINT #2, "impalcato     Vx (mm)      Vy (mm)    rotaz.x1000"
   PRINT #2,
   FOR PS = PSZ TO 1 STEP -1
      PRINT #2, USING "   ##  "; PS;
      PRINT #2, USING "      ##.####"; Rs(PS) * 1000, Rs(PS + PSZ) * 1000, Rs(PS +
            2 * PSZ) * 1000
   NEXT PS

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN


' ------------------------------ Spostamenti ------------------------------------
'  Routine per il calcolo delle componenti di movimento dei nodi
'  del singolo telaio piano
' -------------------------------------------------------------------------------
Spostamenti:
   PRINT "calcolo componenti di movimento": GOSUB InCorso

   ' aggiunge azioni traslanti di incastro
   I = 0
```

```
   FOR K = 1 TO KZ
      I = I + 1
      F(NI(I), 1) = F(NI(I), 1) - Rt(K)
      DO
         I = I + 1
      LOOP UNTIL NI(I) = 0
   NEXT K
   ' calcolo traslazioni traversi
   Al = Alfa(T) * PiGr / 180
   CsAl = COS(Al)
   SnAl = SIN(Al)
   Dist = XO(T) * SnAl - YO(T) * CsAl
   FOR K = 1 TO KZ
      Rt(K) = Rs(PI(K)) * CsAl + Rs(PI(K) + PSZ) * SnAl + Rs(PI(K) + 2 * PSZ) * Dist
   NEXT K
   ' calcolo altre componenti di movimento
   CALL InserMat(Rt(), R(), 1, 1)
   CALL InserMat(Ra(), R(), NL + 1, 1)
   CALL Risolve.K2(KTT(), KTA(), KAA(), SB, 2, R())

   GOSUB Completata
RETURN


' ------------------------------ CaratSollec ------------------------------------
'  Routine per il calcolo delle azioni sulle aste delle strutture piane
'  componenti l'insieme spaziale
' -------------------------------------------------------------------------------
CaratSollec:
   PRINT "calcolo caratt. sollecitazione": GOSUB InCorso

   GOSUB DimRisGen
   CALL ProdottoMat(KTT(), Rt(), St())
   I = 0
   FOR K = 1 TO KZ
      I = I + 1
      F(NI(I), 1) = F(NI(I), 1) + St(K)
      DO
         I = I + 1
      LOOP UNTIL NI(I) = 0
   NEXT K
   CALL CarSoll(JZ, X(), Y(), ZA(), ZAT(), ZI(), E, G, E1(), E2(), TA$(), NT(), K(),
               B(), N(), P(), F(), R(), N1(), T1(), M1(), N2(), T2(), M2())

   GOSUB Completata
RETURN


' ------------------------------ ConverteRisult ---------------------------------
'  Routine per la conversione delle caratteristiche di sollecitazione degli
'  schemi piani particolari
' -------------------------------------------------------------------------------
ConverteRisult:
   IF TT$(T) = "G" THEN RETURN
   PRINT "conversione caratt. sollecitaz.": GOSUB InCorso

   SELECT CASE TT$(T)
      CASE "R"
         GOSUB DimRisRet
```

```
        CALL ConverteRisRet(RZ, PZ, Wnkl, DefEst, NB(), NR(), NF(), ZA(), ZI(),D(),
                            MS(), MD(), TS(), TD(), MP(), MT(), TR(), SN(), MFS(),
                            MFD(), TFS(), TFD(), KZ, NI(), N(), R(), F(), M1(),
                            M2(), T1(), T2(), N1())
   END SELECT

   GOSUB Completata
RETURN


' ------------------------------ StampaRisultTel --------------------------------
'  Routine per la stampa dei risultati
' -------------------------------------------------------------------------------
StampaRisultTel:
   PRINT "stampa risultati"
   IF FO$ = "CON" THEN CLS  ELSE GOSUB InCorso

   PRINT #2, : PRINT #2, : PRINT #2,
   PRINT #2, "TELAIO N."; T
   SELECT CASE TT$(T)
      CASE "G"
         CALL OutputRisGen(IZ, JZ, KZ, NL, V$(), NI(), N(), R(), M1(), M2(), T1(),
                           T2(), N1(), N2(), F())
      CASE "R"
         CALL OutputRisRet(RZ, PZ, Wnkl, NP(), NB(), NR(), NF(), ZA(), ZI(), D(),
                           MS(), MD(), TS(), TD(), MP(), MT(), TR(), SN(), MFS(),
                           MFD(), TFS(), TFD(), IZ, KZ, NL, V$(), N(), R(), F())
   END SELECT

   IF FO$ = "CON" THEN
      CALL Aspetta(2): GOSUB NomeProgramma
   ELSE
      GOSUB Completata
   END IF
RETURN

' ------------------------------ OutputIpotesi ----------------------------------
'  Routine per la stampa delle note sulle ipotesi di calcolo
' -------------------------------------------------------------------------------
OutputIpotesi:
   PRINT #2, : PRINT #2,
   PRINT #2, "Il calcolo e' stato eseguito :"
   PRINT #2, "- trascurando l'interazione tra aste di telai distinti che incidono"
   PRINT #2, "  in uno stesso punto nodale"
   IF TelRet = 1 AND DefEst <> 1 THEN
      PRINT #2, "- trascurando la deformazione estensionale dei pilastri"
      PRINT #2, "  di telai a maglie rettangolari o trapezie"
   ELSE
      PRINT #2, "- tenendo conto della deformazione estensionale delle aste"
   END IF
   IF DefTag = 1 THEN
      PRINT #2, "- tenendo conto della deformazione a taglio delle aste"
      IF TelWnkl = 1 THEN
         PRINT #2, "- trascurando la deformazione a taglio delle aste su suolo alla
                    Winkler"
      END IF
   ELSE
      PRINT #2, "- trascurando la deformazione a taglio delle aste"
   END IF
```

```
   PRINT #2,
   PRINT #2, "Attenzione:"
   PRINT #2, "- l'effettivo sforzo normale in ciascun pilastro dello schema spaziale"
   PRINT #2, "  deve essere valutato  come somma  degli sforzi normali dei telai cui"
   PRINT #2, "  esso appartiene"
   IF TelVCed = 1 OR TelWnkl = 1 THEN
      PRINT #2,"- la correttezza dei risultati ottenuti potrebbe essere inficiata da"
      PRINT #2,"  cedimenti differenziali dovuti alla presenza di vincoli cedevoli o"
      PRINT #2,"  travi di fondazione"
   END IF
   PRINT #2, : PRINT #2,
RETURN


' ------------------------------ EraseFiles ------------------------------------
'   Routine per la cancellazione dei files temporanei
' ------------------------------------------------------------------------------
EraseFiles:
   KILL FT$ + "MATR.SPA"
   KILL FT$ + "GEOM.SPA"
   IF ICT = 1 THEN KILL FT$ + "CAR.SPA"
RETURN


' ------------------------------ ReadCar ---------------------------------------
'  Routine per la lettura dei vettori carichi telaio generico
'  dai file temporanei su memoria di massa
' ------------------------------------------------------------------------------
ReadCar:
   PRINT "lettura carichi": GOSUB InCorso

   GOSUB DimCarGen
   IF ICT = 1 THEN
      CALL InputMat(F(), 5)
      CALL InputMat(P(), 5)
      CALL InputMat(Ra(), 5)
      CALL InputMat(Rt(), 5)
   ELSE
      CALL AzzeraMat(F())
      CALL AzzeraMat(P())
      CALL AzzeraMat(Ra())
      CALL AzzeraMat(Rt())
   END IF

   GOSUB Completata
RETURN


' ------------------------------ ReadGeomTel -----------------------------------
'  Routine per la lettura dei dati geometrici e della matrice di rigidezza
'  del singolo telaio nei file temporanei su memoria di massa
' ------------------------------------------------------------------------------
ReadGeomTel:
   GOSUB NomeProgramma
   PRINT "    Schema di carico n."; CAR
   PRINT
   PRINT "    telaio n."; T
   PRINT "lettura da file temporanei": GOSUB InCorso
```

```
   IF TT$(T) = "R" THEN
      GOSUB ReadGeomRet
   END IF
   GOSUB ReadGeomGen
   GOSUB ReadK

   GOSUB Completata
RETURN

' ..... ReadGeomRet .....
' Routine per la lettura dei dati geometrici per telai a maglie rettangolari
'
ReadGeomRet:
   INPUT #4, RZ, PZ, Wnkl
   GOSUB DimGeomRet
   CALL InputMat(NP(), 4)
   CALL InputMat(DL(), 4)
   CALL InputMat(NB(), 4)
   CALL InputMat(NR(), 4)
   IF Wnkl = 1 THEN
      CALL InputMat(NF(), 4)
      CALL InputMat(BF(), 4)
      CALL InputMat(KF(), 4)
   END IF
RETURN

' ..... ReadGeomGen .....
' Routine per la lettura dei dati geometrici per telai generici
'
ReadGeomGen:
   INPUT #4, IZ, JZ, KZ, VCed, Wnkl
   GOSUB DimGeomGen
   CALL InputMat(X(), 4)
   CALL InputMat(Y(), 4)
   FOR I = 1 TO IZ
      INPUT #4, V$(I)
   NEXT I
   IF VCed = 1 THEN CALL InputMat(KV(), 4)
   CALL InputMat(N(), 4)
   FOR J = 1 TO JZ
      INPUT #4, E1(J), E2(J), TA$(J), NT(J)
      IF Wnkl = 1 THEN INPUT #4, B(J), K(J)
   NEXT J
   FOR J = 1 TO IZ + KZ
      INPUT #4, NI(J)
   NEXT J
   CALL InputMat(PI(), 4)
RETURN

' ..... ReadK .....
' Routine per la lettura della matrice di rigidezza del singolo telaio
'
ReadK:
   INPUT #3, NZ, NL, SB
   GOSUB DimMatRig
   CALL InputMat(KTT(), 3)
   CALL InputMat(KTA(), 3)
   CALL InputMat(KAA(), 3)
RETURN
```

```
' ------------------------------ WriteCar ------------------------------------
'  Routine per la memorizzazione dei vettori carichi telaio generico
'  nei file temporanei su memoria di massa
' ----------------------------------------------------------------------------
WriteCar:
   PRINT "memorizzazione carichi": GOSUB InCorso

   CALL WriteMat(F(), 5)
   CALL WriteMat(P(), 5)
   CALL WriteMat(Ra(), 5)
   CALL WriteMat(Rt(), 5)

   GOSUB Completata
RETURN


' ------------------------------ WriteGeomTel --------------------------------
'  Routine per la memorizzazione dei dati geometrici e della matrice di
'  rigidezza del singolo telaio nei file temporanei su memoria di massa
' ----------------------------------------------------------------------------
WriteGeomTel:
   PRINT "memorizzazione in file temporanei": GOSUB InCorso

   IF TT$(T) = "R" THEN
      GOSUB WriteGeomRet
   END IF
   GOSUB WriteGeomGen
   GOSUB WriteK

   GOSUB Completata
RETURN

' ............................... WriteGeomRet ..................................
'  Routine per la memorizzazione dei dati geometrici per telai a maglie
'  rettangolari
'
WriteGeomRet:
   WRITE #4, RZ, PZ, Wnkl
   CALL WriteMat(NP(), 4)
   CALL WriteMat(DL(), 4)
   CALL WriteMat(NB(), 4)
   CALL WriteMat(NR(), 4)
   IF Wnkl = 1 THEN
      CALL WriteMat(NF(), 4)
      CALL WriteMat(BF(), 4)
      CALL WriteMat(KF(), 4)
   END IF
RETURN

' ............................... WriteGeomGen ..................................
'  Routine per la memorizzazione dei dati geometrici per telai generici
'
WriteGeomGen:
   WRITE #4, IZ, JZ, KZ, VCed, Wnkl
   CALL WriteMat(X(), 4)
   CALL WriteMat(Y(), 4)
   FOR I = 1 TO IZ
```

```
      WRITE #4, V$(I)
   NEXT I
   IF VCed = 1 THEN CALL WriteMat(KV(), 4)
   CALL WriteMat(N(), 4)
   FOR J = 1 TO JZ
      WRITE #4, E1(J), E2(J), TA$(J), NT(J)
      IF Wnkl = 1 THEN WRITE #4, B(J), K(J)
   NEXT J
   FOR J = 1 TO IZ + KZ
      WRITE #4, NI(J)
   NEXT J
   CALL WriteMat(PI(), 4)
RETURN

' ..... WriteK .....
' Routine per la memorizzazione della matrice di rigidezza del singolo telaio
'
WriteK:
   WRITE #3, NZ, NL, SB
   CALL WriteMat(KTT(), 3)
   CALL WriteMat(KTA(), 3)
   CALL WriteMat(KAA(), 3)
RETURN
```

5. Modalità di utilizzazione del programma

I dati necessari per descrivere lo schema geometrico e di carico, elencati in dettaglio nel seguito, devono essere conservati in un file su memoria di massa, che può essere preparato mediante un qualsiasi word processor tra quelli esistenti in commercio.

Quando il programma viene mandato in esecuzione, esso richiede espressamente in maniera interattiva le seguenti informazioni:

- il nome del file che contiene i dati;
- il nome dell'unità alla quale inviare l'output; questa può essere lo schermo (CON), la stampante (PRN) oppure un file del quale si indicherà il nome;
- il nome del drive e della directory nella quale memorizzare i file temporanei creati dal programma per conservare informazioni relative alla geometria, ai carichi ed alle matrici di rigidezza dei singoli telai. Se si indica un drive con dischetti flessibili, è opportuno controllare che nel dischetto vi sia spazio sufficiente. Se la capacità di memoria disponibile lo consente, l'impostazione ottimale dal punto di vista dei tempi di esecuzione consiste nell'utilizzare per tale scopo una parte della memoria configurata come disco virtuale;
- se nel calcolo si vuol tener conto della deformabilità estensionale dei pilastri appartenenti a telai a maglie rettangolari o trapezie (si risponda

1 in caso affermativo, 0 in caso negativo). In questo tipo di telai la presenza di ritti inestensibili è simulata bloccando i nodi allo spostamento verticale; se in essi è presente una trave di fondazione, o la continuità verticale dei pilastri è interrotta per la mancanza di ritti, è necessario tener conto della deformabilità estensionale;

- se nel calcolo si vuole tener conto della deformabilità a taglio (si risponda ancora 1 in caso affermativo, 0 in caso negativo). La versione attuale del programma prevede la deformabilità a taglio solo per aste non collegate ad un suolo elastico; assume inoltre automaticamente un valore del modulo di elasticità tangenziale G legato a quello del modulo di Young E dalla relazione $G = 0.45E$, valida per $\nu = 0.1$ (cemento armato);
- se si vuole effettuare un controllo dei dati man mano che vengono letti. Conviene utilizzare questa opzione quando il programma si blocca a causa di qualche errore nei dati che non si riesce ad individuare. Il controllo consiste nell'inviare all'unità di uscita una segnalazione per ciascun valore letto, nella quale il programma indica come questo valore è stato interpretato.

Il programma passa poi a leggere dal file indicato i dati geometrici e di carico. Ciascuna esecuzione consente la risoluzione di uno schema geometrico soggetto a più condizioni di carico. I dati devono essere memorizzati nel file in maniera sequenziale, separati da virgole o da andate a capo, col seguente ordine:

- gruppo di dati che descrivono la geometria dell'intero complesso spaziale
- gruppo di dati che descrivono la geometria dei singoli telai
- gruppo di dati che descrivono la prima condizione di carico
- gruppo di dati che descrivono la seconda condizione di carico

  .....
- gruppo di dati che descrivono l'ultima condizione di carico

Nel definire l'insieme spaziale di telai, occorre numerare progressivamente sia i telai che gli impalcati a partire dal valore 1. L'ordine col quale questi numeri vengono attribuiti non ha alcuna influenza sui tempi di risoluzione. Per ciascun telaio è definito un riferimento locale $x'y'$, individuato mediante la posizione dell'origine e l'inclinazione dell'asse x' rispetto all'asse x del sistema di riferimento globale.

Nelle applicazioni pratiche numerose aste di uno schema presentano la stessa sezione. Si è pertanto ritenuto più comodo per la immissione dei dati che l'utente individui preliminarmente i differenti tipi di sezione, in una tabella comune a tutti i telai. Per ciascun tipo di sezione deve essere

definita la forma (rettangolare o generica) e le relative caratteristiche geometriche. Il gruppo di dati che descrive globalmente il complesso spaziale deve rispettare l'ordine di seguito riportato. I valori forniti devono essere conformi alle unità di misura indicate tra parentesi:

- intestazione
- numero totale di telai dello schema
- numero di impalcati presenti nello schema
- numero di tipi di sezioni
- per ogni telaio:
  - carattere che identifica il tipo di telaio:
    G = telaio generico, descritto nel capitolo 5
    R = telaio a maglie rettangolari o trapezie, descritto nel capitolo 6
  - ascissa e ordinata dell'origine del riferimento locale rispetto alla terna globale (m)
  - direzione o inclinazione del telaio; è possibile indicare una lettera (x o y) quando il telaio è parallelo a uno degli assi, oppure l'angolo formato dall'asse locale x' rispetto all'asse globale x (espresso in gradi e positivo se nel verso che porta x su y)
- modulo di elasticità normale (N mm^{-2})
- per ogni tipo di sezione:
  - carattere che identifica la forma della sezione:
    R = sezione di forma rettangolare
    G = sezione di forma generica
  - caratteristiche geometriche della sezione:
    - base (m) ed altezza (m) per le sezioni rettangolari
    - area (m^2), rapporto tra area e fattore di taglio (m^2) e momento di inerzia (m^4) per quelle generiche

I dati che descrivono la geometria di ciascun telaio piano dipendono dal tipo di telaio. I criteri con i quali preparare lo schema geometrico sono stati descritti nel paragrafo "Modalità di utilizzazione del programma" dei due capitoli precedenti e per brevità non vengono qui ripetuti. In aggiunta a quanto ivi indicato, si deve definire a quale impalcato appartiene ciascun traverso. È possibile indicare che più traversi appartengono ad uno stesso impalcato, mentre non è previsto che un traverso inestensibile sia svincolato da tutti gli impalcati. Si deve quindi intendere come traverso un gruppo di nodi del telaio piano che appartengono tutti allo stesso impalcato dello

schema spaziale. Il numero di traversi del singolo telaio può essere minore del numero di impalcati dell'insieme spaziale.

Si noti che si è mantenuta la possibilità di imporre vincoli cedevoli o travi di fondazione su suolo alla Winkler. Questa opportunità deve però essere sfruttata con molta cautela, ricordando che il modello di insieme spaziale di telai piani non impone la congruenza degli spostamenti verticali di nodi comuni a telai distinti.

Il gruppo di dati che descrive la geometria di un telaio di tipo generico appartenente all'insieme deve rispettare l'ordine di seguito riportato:

- numero totale di nodi del telaio
- numero di aste presenti nel telaio
- numero di traversi orizzontali indeformabili (ovvero di gruppi di nodi aventi lo stesso spostamento orizzontale)
- per ogni nodo:
  - ascissa del nodo nel riferimento locale del telaio (m)
  - ordinata del nodo nel riferimento locale del telaio (m)
  - codice di vincolo. È costituito da una stringa con tre caratteri, che si riferiscono alle tre componenti di movimento del nodo (nell'ordine: traslazione in direzione x ed y e rotazione). Si indica con `0` una componente di movimento consentita, con `1` una impedita, con `C` una limitata da un vincolo cedevole
  - costante elastica di eventuali vincoli cedevoli (kN m^{-1} per vincoli allo spostamento, kNm per vincoli alla rotazione). La costante elastica deve essere fornita solo per quei gradi di libertà per i quali si è usato `C` come codice di vincolo
- per ogni traverso orizzontale inestensibile: nodi collegati dal traverso, che devono quindi avere lo stesso spostamento orizzontale. L'elenco dei nodi deve terminare con uno 0, che indica al calcolatore la fine dei nodi del traverso
- per ogni asta:
  - primo estremo
  - secondo estremo
  - carattere che identifica il tipo di asta:

    `T` = trave alla De Saint Venant solidale ai nodi di estremità

    `W` = trave su suolo elastico alla Winkler
  - tipo di sezione

  - larghezza della sottobase, se l'asta poggia su un suolo elastico alla Winkler (m)
  - costante di sottofondo, se l'asta poggia su un suolo elastico alla Winkler (N cm^{-3})
- per ogni traverso inestensibile: numero d'ordine dell'impalcato ad esso corrispondente

Il gruppo di dati che descrive la geometria di un telaio a maglie rettangolari o trapezie appartenente all'insieme deve rispettare l'ordine di seguito riportato:

- numero totale di pilastri (intesi come allineamenti verticali)
- numero totale di piani, intesi come allineamenti orizzontali o inclinati indipendentemente dal fatto che la loro continuità sia mantenuta o interrotta dalla mancanza di travi e che siano quindi costituiti da uno o più traversi inestensibili
- presenza della trave di fondazione (0=no, 1=si)
- numerazione dei pilastri in carpenteria, da sinistra verso destra
- altezza degli interpiani, dall'alto verso il basso (m); se ad un interpiano i ritti non hanno tutti la stessa altezza deve essere assegnato ad essa il valore 0
- per ciascun interpiano i cui ritti non hanno tutti la stessa altezza, dall'alto verso il basso: altezza di ciascun ritto, da sinistra verso destra (m)
- se vi è la trave di fondazione: lunghezza dello sbalzo sinistro (m)
- luce di ogni campata, da sinistra verso destra (m)
- se vi è la trave di fondazione: lunghezza dello sbalzo destro (m)
- per le travi di ciascun piano, dall'alto verso il basso: tipo di sezione di ogni campata, da sinistra verso destra; assegnare una sezione di tipo 0 indica che l'asta è mancante
- per i pilastri di ciascun interpiano, dall'alto verso il basso: tipo di sezione di ciascun ritto, da sinistra verso destra; assegnare una sezione di tipo 0 indica che l'asta è mancante
- se è presente la trave di fondazione, per ciascun tratto, inclusi gli sbalzi, da sinistra verso destra:
  - tipo di sezione; assegnare una sezione di tipo 0 indica che il tratto è mancante
  - larghezza della sottobase (m)

- costante di sottofondo (N cm^{-3})
- per ogni traverso inestensibile: numero d'ordine dell'impalcato ad esso corrispondente

Ciascuno schema di carico può essere costituito sia da forze e momenti concentrati applicati direttamente agli impalcati che da carichi, concentrati o distribuiti, agenti sui singoli telai; poiché non sempre questi carichi coesistono, si è ritenuto preferibile che l'utente indichi preliminarmente se sono presenti gli uni, gli altri o entrambi.

Le azioni concentrate su ogni impalcato possono essere applicate ad un punto qualsiasi, individuato mediante l'ascissa e ordinata nel riferimento globale; sono definite mediante le forze F_x e F_y, positive se concordi col verso positivo degli assi x ed y, e la coppia M_z, positiva se tende a far ruotare l'impalcato nel verso che porta x su y.

I carichi applicati sui singoli telai sono descritti con riferimento al sistema di assi locale $x'y'$. Nel caso di telai generici sono previste forze e coppie nodali, $F_{x'}$ $F_{y'}$ $M_{z'}$, e carichi distribuiti sulle aste, $p_{x'}$ $p_{y'}$. Nel caso di telai a maglie rettangolari o trapezie sono previste forze verticali e coppie nodali, carichi verticali uniformemente distribuiti sulle travi, forze orizzontali applicate ai traversi inestensibili. Le forze e i carichi sono positivi se concordi col verso degli assi cui sono paralleli, le coppie nodali se nel verso che porta x' su y' cioè antiorarie nel piano del telaio. Si noti che, avendo definito come verso positivo dell'asse y' quello diretto verso l'alto, i carichi verticali agenti verso il basso devono essere indicati col segno meno. Si osservi inoltre che si è mantenuta la possibilità di applicare azioni orizzontali ai singoli telai, ma che essa dovrà essere sfruttata solo per situazioni particolari perché usualmente nell'analisi sismica di edifici le masse vengono considerate come concentrate sugli impalcati ed a questi quindi devono essere applicate le conseguenti azioni inerziali orizzontali.

Il gruppo di dati che descrive la singola condizione di carico deve rispettare l'ordine e le unità di misura di seguito indicati:

- indice di esistenza di carichi applicati agli impalcati (0=no, 1=si)
- indice di esistenza di carichi applicati ai telai (0=no, 1=si)
- informazioni che descrivono i carichi applicati agli impalcati (se il primo indice uguale a 1)
- per ciascun telaio: informazioni che descrivono i carichi applicati ad esso (se il secondo indice è uguale a 1)

Per individuare i carichi sugli impalcati occorre indicare:

- per ciascun impalcato:

  - ascissa e ordinata del punto di applicazione delle azioni concentrate rispetto al sistema di riferimento globale (m)
  - F_x, componente in direzione x dell'azione applicata all'impalcato (kN)
  - F_y, componente in direzione y dell'azione applicata all'impalcato (kN)
  - M_z, coppia concentrata (kNm)

Per individuare i carichi su un telaio generico (tipo `G`) occorre indicare:

- numero di nodi in cui sono applicati carichi concentrati
- numero di aste soggette a carico uniformemente distribuito
- per ciascun nodo caricato:
  - numero d'ordine del nodo
  - $F_{x'}$, componente in direzione x' della forza concentrata (kN)
  - $F_{y'}$, componente in direzione y' della forza concentrata (kN)
  - $M_{z'}$, coppia concentrata (kNm)
- per ciascuna asta caricata:
  - numero d'ordine dell'asta
  - $p_{x'}$, componente del carico distribuito nella direzione x' del riferimento locale del telaio, valutata per unità di lunghezza dell'asta ($\mathrm{kNm^{-1}}$)
  - $p_{y'}$, componente del carico distribuito nella direzione y' del riferimento locale del telaio, valutata per unità di lunghezza dell'asta ($\mathrm{kNm^{-1}}$)

Per individuare i carichi su un telaio a maglie rettangolari o trapezie (tipo `R`) occorre indicare:

- indice di esistenza dei carichi verticali sulle travi (0=no, 1=si)
- indice di esistenza dei momenti concentrati nei nodi (0=no, 1=si)
- indice di esistenza delle forze verticali concentrate nei nodi (0=no, 1=si)
- indice di esistenza dei carichi orizzontali (0=no, 1=si)
- se esistono carichi verticali sulle travi (primo indice uguale a 1):
  - per ciascun piano, dall'alto verso il basso: carico verticale su ciascuna campata ($\mathrm{kN\ m^{-1}}$), da sinistra verso destra, con esclusione delle aste non esistenti (cioè con sezione di tipo 0) o con sezione avente area ed inerzia nulli

  - per la trave di fondazione, se esiste: carico verticale su ciascun tratto (kN m^{-1}), da sinistra verso destra, con esclusione dei tratti in cui la trave non esiste o ha sezione con area ed inerzia nulli
- se esistono momenti concentrati nei nodi (secondo indice uguale a 1):
  - per ciascun piano, dall'alto verso il basso: momento concentrato in ciascun nodo (kNm), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale (cioè con sezione non nulla)
  - per la trave di fondazione, se esiste: momento concentrato in ciascun nodo (kNm), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale
- se esistono forze verticali concentrate nei nodi (terzo indice uguale a 1):
  - per ciascun piano, dall'alto verso il basso: forza verticale concentrata in ciascun nodo (kN), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale
  - per la trave di fondazione, se esiste: forza verticale concentrata in ciascun nodo (kN), da sinistra verso destra, con esclusione dei nodi nei quali non converge alcuna asta reale
- se esistono forze orizzontali sui traversi (quarto indice uguale a 1):
  - per ciascun traverso, dall'alto verso il basso (e da sinistra verso destra se esistono più traversi inestensibili ad uno stesso piano): forza orizzontale applicata al traverso (kN)

Si ricordi infine che il file dati non deve contenere dopo l'ultimo blocco di dati alcuna riga bianca, che verrebbe intepretata dal calcolatore come un valore nullo ma darebbe inizio comunque alla lettura dei dati di un ulteriore schema di carico.

6. Risultati forniti

L'output del programma contiene innanzitutto l'elenco dei dati geometrici ed elastici forniti in ingresso dall'utente, per consentirne il necessario controllo. Sono riportate innanzitutto le informazioni relative all'insieme nel suo complesso, che mostrano il tipo e la posizione di ciascun telaio, i tipi di sezione previsti per le aste, le caratteristiche del materiale. Sono poi descritti i singoli telai, in maniera diversa in base al loro tipo secondo quanto già illustrato nei capitoli precedenti; alle informazioni geometriche è aggiunta l'elencazione degli impalcati corrispondenti ai singoli traversi inestensibili del telaio piano.

Per ciascuno schema di carico vengono riportate, se presenti, sia le azioni

applicate direttamente agli impalcati che quelle agenti sui singoli telai.

Sono poi riportate le componenti di movimento degli impalcati ottenute come risultato del calcolo. Gli spostamenti v_x e v_y, positivi se concordi col verso positivo degli assi x ed y del riferimento globale, sono espressi in mm. La rotazione θ_z, positiva se nel verso che porta x su y, è indicata in radianti amplificata di un fattore 1000.

Per ciascun telaio sono riportate le componenti di movimento dei nodi, riferite al sistema locale $x'y'$, con le stesse unità di misura e convenzione dei segni. I valori delle caratteristiche della sollecitazione M, T, N nei due estremi sono espressi in kNm e kN. Si ricorda che per esse valgono le usuali convenzioni della Scienza delle Costruzioni considerando le aste orientate dal primo al secondo estremo; in particolare, per telai di tipo R le travi sono orientate da sinistra a destra, i pilastri dal basso verso l'alto. Il momento flettente è quindi positivo se tende le fibre inferiori dell'asta, guardandola nel piano del telaio in modo da avere a sinistra il primo estremo e a destra il secondo; il taglio è positivo se la microcoppia costituita dalle azioni taglianti su due facce opposte è antioraria, lo sforzo normale è positivo se di trazione. Se si riscontrano squilibri in nodi non vincolati o nei traversi orizzontali, dovuti agli inevitabili errori di troncamento del procedimento risolutivo, questi vengono espressamente segnalati per consentire all'utente di rendersi conto della loro entità e quindi della maggiore o minore validità dei risultati ottenuti. Si ricorda infine che i risultati, forniti per i singoli telai, devono essere riguardati in maniera globale, sia per utilizzarli nella verifica delle sezioni che per valutare l'attendibilità del modello. Non si può infatti dimenticare che, pur avendo schematizzato l'edificio come insieme di telai collegati solo dagli impalcati, nella realtà ciascun pilastro appartiene contemporaneamente a due telai distinti. L'effettivo sforzo normale è quindi somma delle aliquote fornite per ciascun telaio; esso inoltre è soggetto a flessione in piani distinti e può essere verificato a pressoflessione retta solo se la flessione in un piano prevale nettamente sull'altra. Analogamente, ciascun nodo della struttura spaziale appartiene in realtà contemporaneamente a due telai piani. L'esame dell'abbassamento e della rotazione forniti per l'uno e per l'altro consentono di rendersi conto se è stato corretto, o no, trascurare le azioni mutue che dovrebbero insorgere tra i distinti telai nei punti di contatto.

PARTE TERZA

APPLICAZIONI

EDIFICIO PER 54 APPARTAMENTI NEL COMUNE DI GIUGLIANO

IL PROGETTO

Fabio Ghersi

1. Criteri generali di progettazione

L'impostazione generale del progetto è stata determinata tanto dai vincoli imposti dalla committenza – contenimento dei costi, massimo sfruttamento delle potenzialità edificatorie del terreno – quanto dalle caratteristiche dell'area assegnata. Essa è infatti localizzata nel comune di Giugliano, immediatamente a nord di Napoli, in una zona che, inghiottita dall'espansione metropolitana, rivela tutte le caratteristiche comuni alle nostre periferie: frammentazione, degrado, assenza di qualità urbane ed edilizie. È certo vero che il *progetto* non può risolvere questi problemi; esso tuttavia non può nemmeno ignorarli. Al contrario – ed è questo lo spirito che ha guidato le scelte di fondo – esso può *rappresentarli*, nella ricerca di una strada capace di indicare nella qualità del progetto l'obiettivo da perseguire.

La forma irregolare del terreno – una L con il braccio maggiore disposto ortogonalmente alla strada di accesso – ha suggerito l'adozione di una tipologia in linea articolata in due blocchi disposti rispettivamente in ognuno dei due "bracci" dell'area.

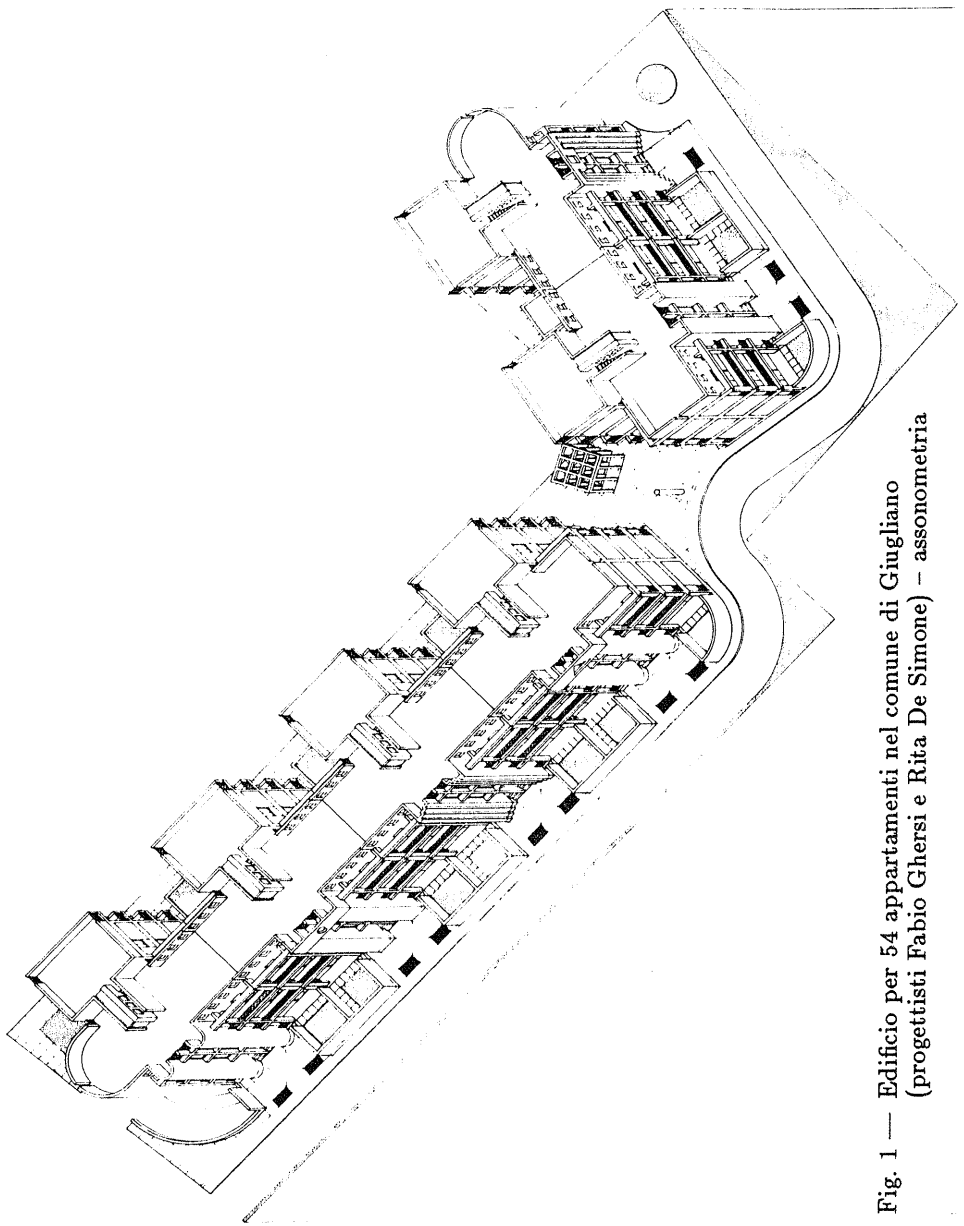


Fig. 1 — Edificio per 54 appartamenti nel comune di Giugliano (progettisti Fabio Ghersi e Rita De Simone) – assonometria

I due blocchi sono stati disegnati come parti di un unico oggetto spezzato e disposti secondo un angolo che contraddice l'esatta ortogonalità dei comparti dell'area. Questa scelta ha permesso di distribuire tutti gli ingressi mediante una strada a cul–de–sac che costeggia su di un lato entrambi i blocchi; ha inoltre permesso la creazione di una piccola piazza "urbana" tra le testate degli edifici stessi. Sul fondale di questo spazio, nel centro virtuale di rotazione dei blocchi, un "colonnato" a pianta quadrata marca il sistema di scale che portano al sottostante piano dei garage.

Tra la strada di distribuzione e il fronte degli edifici, gli slarghi antistanti gli ingressi di ciascun blocco scala si alternano ai giardini privati annessi agli appartamenti a piano terra, rialzati dal terreno e isolati da muri alti 2.10 m. Sul fronte opposto degli edifici una sistemazione continua a verde con percorsi e slarghi pavimentati costituisce un giardino sicuro e completamente isolato dalla viabilità veicolare.

2. Gli edifici

Ciascun edificio è generato dalla ripetizione di un modulo tipologico a T costituito da una scala centrale che distribuisce tre appartamenti per ogni livello. L'edificio maggiore è costituito da quattro di questi moduli, il minore da due. In ciascun edificio uno dei moduli di testata si conclude in forma cilindrica, mentre l'altro resta "sezionato" a manifestare la continuità tra i due blocchi.

Ognuno dei sei moduli tipologici è risolto sul fronte di ingresso nella forma di un grande portale alto come tutto l'edificio. Questi portali, tutti "uguali ma diversi", sono costruiti a partire dalle figure elementari – cerchio, quadrato e triangolo – nelle sei combinazioni di ciascuna forma con le altre due. Memori della "colonna" Loosiana del Chicago Tribune, caricaturali nel porsi "fuori scala", essi costruiscono nel loro susseguirsi il fronte principale degli edifici.

Ad affermare un'immagine analoga di frammento–memoria di una classicità che non può essere proclamata, ma nemmeno taciuta, il fronte opposto è stato risolto nella facciata "intermittente" delle cortine di mattoni che chiudono le logge degli appartamenti di interpiano.

I due edifici sono collegati a livello dei garage, a quota -3.30 m sotto il livello del terreno: una circolazione anulare connessa alla quota di ingresso mediante due rampe che circondano la testata cilindrica dell'edificio maggiore distribuisce i box, uno per appartamento, tutti i vani scala ed i locali tecnici necessari. L'aereazione e l'illuminazione del piano dei garage sono

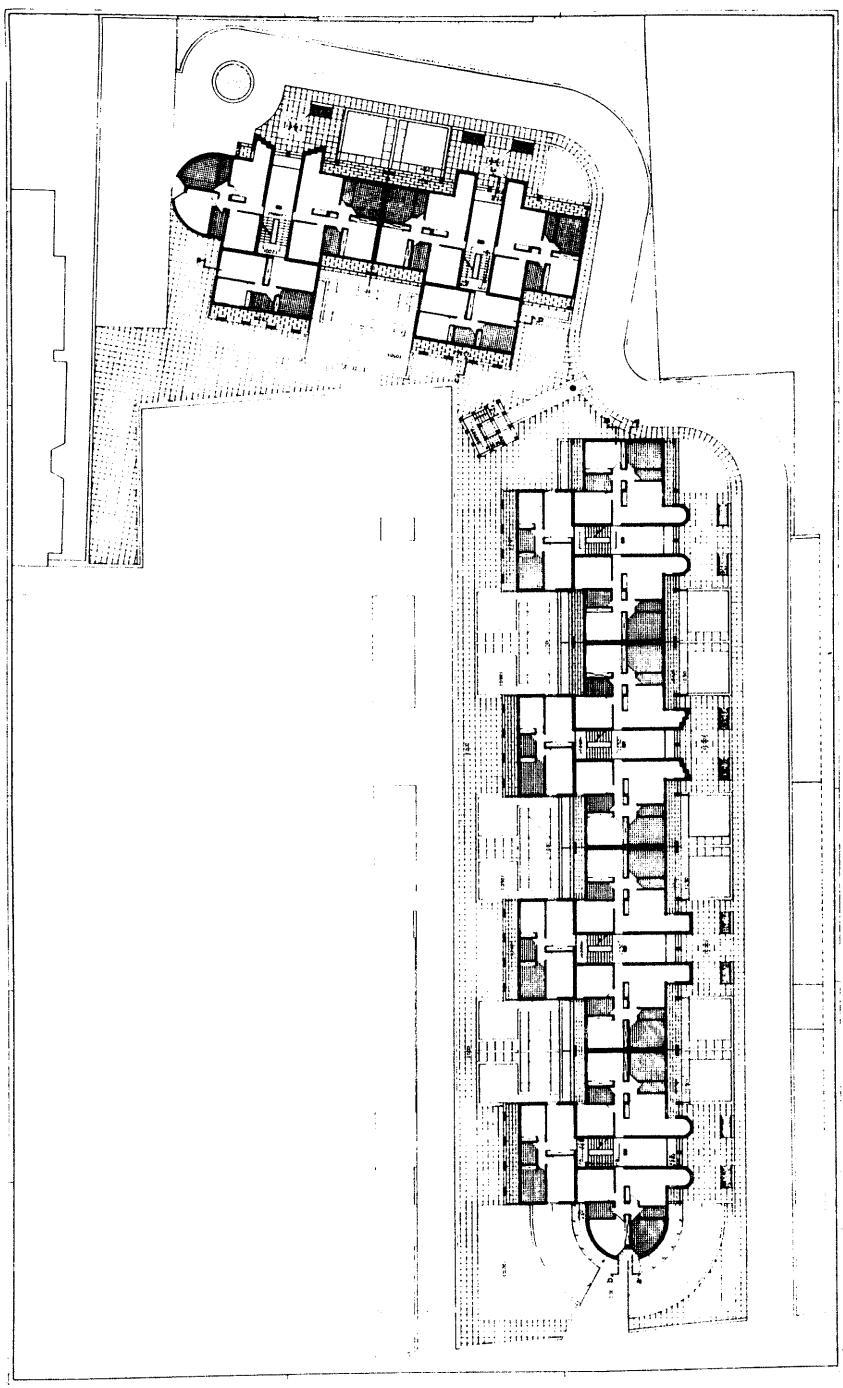


Fig. 2 — Planimetria generale

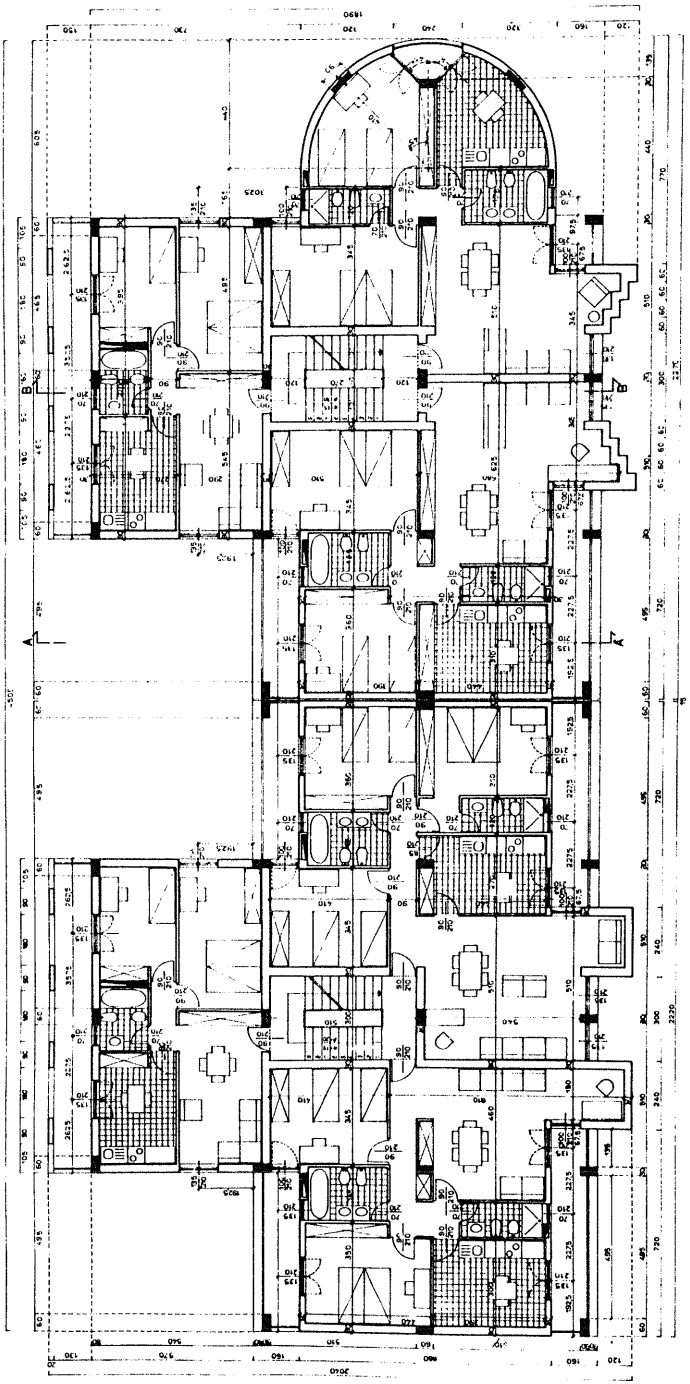


Fig. 3 — Pianta

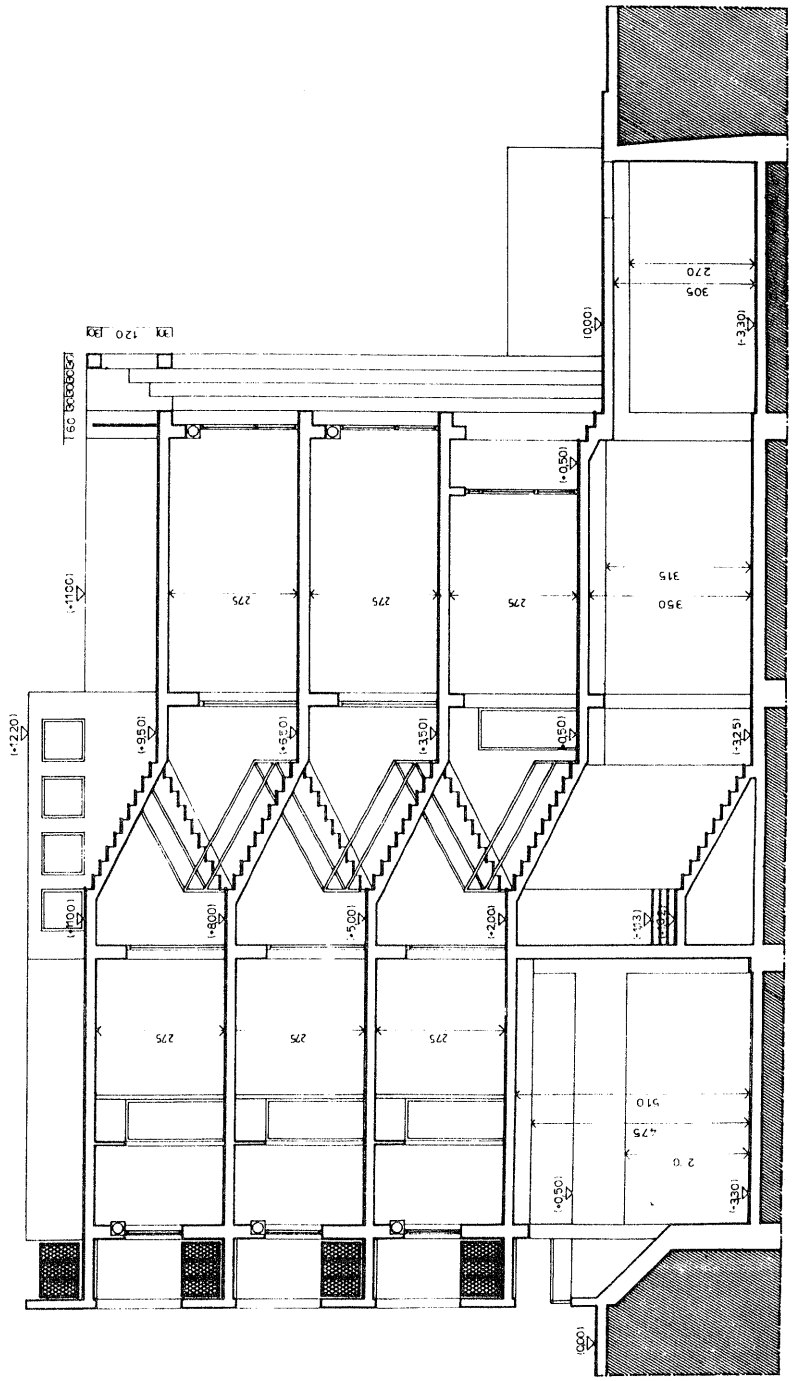


Fig. 4 — Sezione

assicurate da fori protetti da grate sul fronte degli ingressi e dalle aperture continue derivanti dallo sfalsamento dei solai sul fronte opposto.

3. Il modulo tipologico

Ogni modulo tipologico è costituito da un blocco scala che, per ogni livello, distribuisce due appartamenti sul piano principale – a destra e a sinistra della scala stessa – ed un appartamento sull'interpiano. I solai che formano l'asta centrale della T sul lato opposto al fronte di ingresso sono pertanto sfalsati di mezzo piano.

Gli appartamenti posti sui piani principali, affacciati su entrambi i fronti loggiati dell'edificio, sono stati differenziati in quattro tipi, da 80 a circa 100 m^2; le variazioni di superficie sono state determinate, oltre che dalle soluzioni di testata, dalla posizione del muro divisorio tra gli appartamenti. Le diverse forme planimetriche dei "portali" – quadrata, circolare, scalettata – generano inoltre, negli spazi di soggiorno di questi appartamenti, ulteriori differenziazioni qualitative. Gli appartamenti posti sull'interpiano, affacciati su tre lati, costituiscono invece il tipo più piccolo, con superficie di circa 60 m^2.

L'incidenza degli spazi di distribuzione nella divisione interna degli alloggi è compresa, secondo i tipi, tra il 3.4 e il 5.1% della superficie netta. Tutti gli appartamenti sono provvisti di logge da 20 m^2 per il primo tipo e da 15 m^2 per il secondo, con la sola eccezione degli alloggi posti sulle testate cilindriche (4 m^2). Gli appartamenti a piano terra usufruiscono in aggiunta di un giardino privato.

Ogni modulo tipologico si sviluppa su tre livelli e contiene pertanto sei appartamenti del primo tipo (80/100 m^2) e tre del secondo (60 m^2).

Da un punto di vista strutturale, ciascun modulo tipologico costituisce un telaio indipendente. Esso è formato da 18 pilastri di sezione 30x60 cm, orientati alternativamente in direzioni ortogonali, per conferire alla struttura un diffuso irrigidimento nei confronti delle azioni sismiche. Sulle travi poggiano solai in cemento armato e laterizi gettati in opera, sfalsati di 1.50 m. La scala che collega gli impalcati sfalsati è costituita da una soletta rampante. Date le buone caratteristiche meccaniche del terreno, costituito da sabbie mediamente addensate, si sono previste fondazioni dirette realizzate con un reticolo di travi rovesce con piano di posa posto ad una profondità di -4.90 m dal livello del terreno.

I 18 pilastri della struttura intelaiata sono ripartiti in quattro campate sul fronte per tre campate in profondità. Questa ripartizione ha svincolato

il sistema strutturale dalla logica distributiva dell'edificio, peraltro marcata dal disegno dei "portali". Infatti da un punto di vista architettonico–distributivo il fronte dell'edificio è diviso nel sistema appartamento/ingresso–scala/appartamento, cioè in un numero di campate dispari (con la classica campata centrale coincidente al varco di ingresso). Esso dunque si sovrappone al sistema pari – di quattro campate – della ripartizione strutturale così che la scala, ad esempio, viene a trovarsi a cavallo della fila centrale di pilastri.

Questa scelta di apparente contraddizione dei sistemi ha consentito la creazione di una ossatura strutturale estremamente semplificata con un basso numero di pilastri e delle dimensioni di campata molto regolari, a tutto vantaggio dell'economia generale dell'opera, dell'efficienza distributiva, della semplicità e della chiarezza spaziale.

In modo del tutto analogo – è in definitiva l'idea Corbusieriana del plan livre – la cortina di mattoni che forma il fronte dell'edificio si tiene sempre staccata dal filo della struttura (in cemento armato a vista); essa avanza o arretra creando le logge e disegnando i portali.

EDIFICIO PER 54 APPARTAMENTI NEL COMUNE DI GIUGLIANO

RELAZIONE GENERALE DI CALCOLO

1. Normativa di riferimento

La progettazione degli edifici è stata eseguita in conformità alle norme tecniche vigenti relative alle opere in conglomerato cementizio armato:

- legge 5/11/1971 n.1086, Norme per la disciplina delle opere in conglomerato cementizio armato normale e precompresso ed a struttura metallica;
- legge 2/2/1974 n.64, Provvedimenti per le costruzioni con particolari prescrizioni per le zone sismiche;
- D.M. 12/2/1982, Aggiornamento delle norme tecniche relative ai "Criteri generali per la verifica di sicurezza delle costruzioni e dei carichi e sovraccarichi";
- D.M. 27/7/1985, Norme tecniche per l'esecuzione delle opere in cemento armato normale e precompresso e per le strutture metalliche;
- CNR 10012/85, Istruzioni per la valutazione delle azioni sulle costruzioni;
- D.M. 24/1/1986, Norme tecniche relative alle costruzioni sismiche;
- CNR 10024/86, Analisi mediante elaboratore: impostazione e redazione delle relazioni di calcolo;
- D.M. 11/3/1988, Nuove norme tecniche per terreni, opere di sostegno e fondazioni.

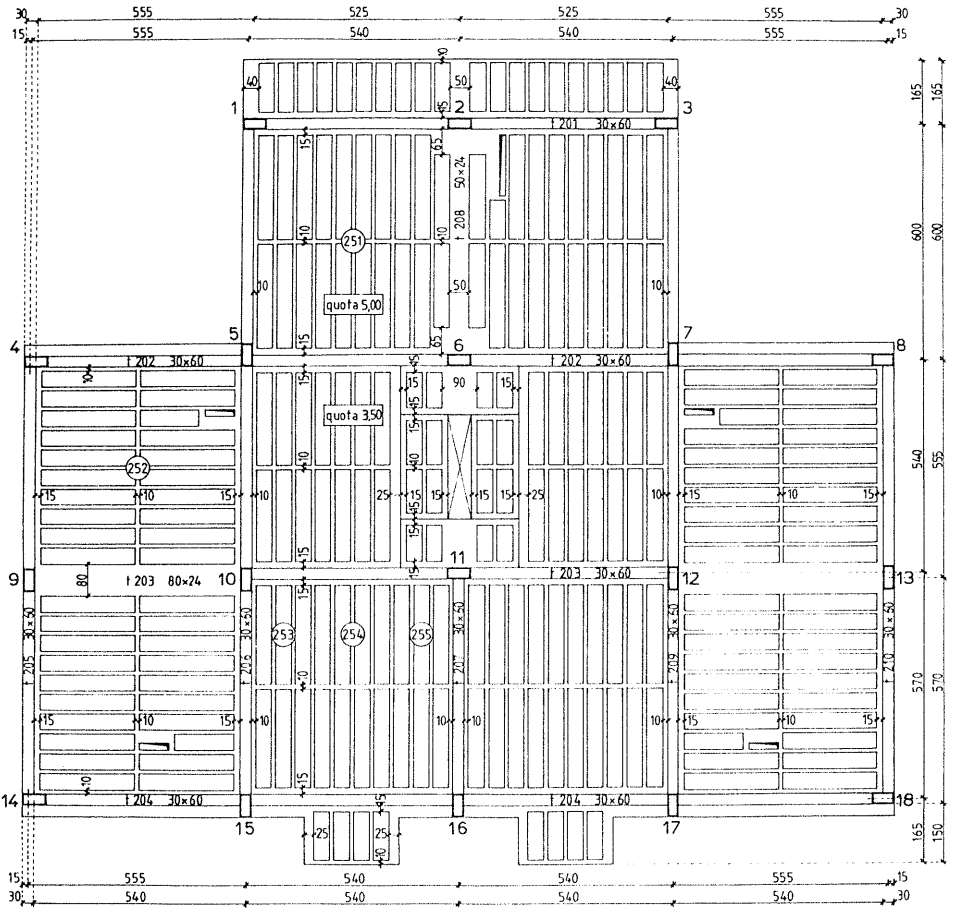


Fig. 5 — Carpenteria; i pilastri hanno sezione 30×60 costante a tutti i piani

2. Schematizzazione della struttura

Le caratteristiche meccaniche del terreno hanno consentito l'adozione di una fondazione superficiale costituita da un reticolo di travi rovesce. Le sezioni utilizzate rendono la fondazione notevolmente più rigida rispetto alla struttura in elevazione. Si è pertanto ritenuto accettabile analizzare quest'ultima indipendentemente dalla prima, ipotizzando la presenza di vincoli di incastro perfetto al piede dei pilastri del primo ordine, in corrispondenza delle sezioni di attacco con le travi. A sua volta, il reticolo di fondazione è stato esaminato in maniera autonoma, schematizzandolo come insieme di travi su suolo elastico alla Winkler collegate in punti nodali nei quali è imposta la congruenza degli abbassamenti; si è ritenuta trascurabile

la rigidezza torsionale delle travi e non si è quindi imposta la congruenza rotazionale tra travi ortogonali. I carichi applicati al reticolo sono costituiti dalle azioni trasmesse dalla struttura in elevazione ai vincoli di incastro.

La struttura in elevazione è costituita da pilastri mutuamente collegati ad ogni impalcato mediante travi nelle due direzioni. Sulle travi poggiano solai in cemento armato e laterizi gettati in opera. Gli impalcati sono sfalsati di 1.50 m l'uno dall'altro e sono collegati, oltre che dai pilastri, anche dalla soletta rampante delle scale. Si è ipotizzato che ciascun impalcato sia indeformabile nel suo piano, mentre si è ritenuto trascurabile il collegamento costituito dalla soletta rampante, che ha dimensioni planimetriche modeste rispetto a quelle degli impalcati.

Sulla base di quanto esposto, si può adottare per l'ossatura in cemento armato il modello geometrico di struttura intelaiata spaziale con impalcati infinitamente rigidi nel loro piano. La regolarità dell'orditura consente di semplificare ulteriormente lo schema, individuando due famiglie di telai piani, ortogonali tra loro. Si è ipotizzata trascurabile la deformabilità estensionale dei pilastri, ottenendo così automaticamente la congruenza verticale dei nodi comuni a telai distinti. Si è inoltre ritenuto di poter trascurare la rigidezza torsionale delle travi e quindi anche l'accoppiamento flesso–torsionale tra travi incidenti in uno stesso nodo. In definitiva, la struttura è stata schematizzata come insieme spaziale di telai piani, mutuamente vincolati da impalcati indeformabili. La particolarità dell'edificio, costituita dallo sfalsamento dei piani, non inficia tale schema, ma semplicemente comporta che ciascun telaio è collegato solo ad alcuni degli impalcati.

Si è contemporaneamente supposto che, nei confronti dei carichi verticali, i solai siano notevolmente più deformabili rispetto alle travi su cui poggiano; per racchiudere il loro comportamento in una fascia adeguata li si è schematizzati come travi continue vincolate con appoggi fissi o con incastri parziali.

Gli elementi non propriamente strutturali, tramezzi e tompagni, sono disposti nella maggior parte dei casi lungo allineamenti distinti da quelli delle travi, ed i pochi inseriti in maglie della struttura presentano aperture di dimensioni rilevanti. Si è quindi ritenuto pienamente accettabile trascurare dal punto di vista numerico il loro contributo alla statica dell'edificio, pur riconoscendo che essi possono svolgere un ruolo positivo nella dissipazione di energia durante una scossa sismica.

Per tutti i modelli geometrici si è ipotizzato un comportamento linearmente elastico, scartando quindi la possibilità di non linearità geometriche (effetto $P - \delta$ o effetto instabilizzante dei carichi verticali) o del materiale (plasticizzazione delle sezioni).

3. Tipo di analisi

L'attuale normativa relativa alle costruzioni sismiche impone l'analisi modale solo per strutture molto deformabili, con periodo proprio di oscillazione superiore a 1.4 secondi. È comunque opportuno ricorrere ad essa nel caso di strutture che presentino un comportamento spiccatamente tridimensionale, con forti rotazioni degli impalcati, in modo da superare i limiti dell'analisi statica che è stata sviluppata con riferimento a schemi piani. Più in generale, l'analisi modale può risultare utile per una migliore comprensione del comportamento di tutti gli schemi irregolari, anche a comportamento piano. Si deve però ricordare che le irregolarità mandano in crisi una struttura non tanto per il contributo dei modi di oscillazione superiori al primo, quanto piuttosto per gli incrementi locali di tensione e deformazione e la conseguente riduzione della duttilità globale, problemi che andrebbero pittosto affrontati con la ben più complessa analisi dinamica non lineare al passo.

L'edificio in oggetto potrebbe definirsi "moderatamente irregolare". I pilastri sono disposti secondo un reticolo estremamente semplice ed orientati in modo da conferire alla struttura un irrigidimento diffuso. L'unica vera irregolarità è costituita dallo sfalsamento degli impalcati e dalla conseguente differenza di altezza dei ritti. Poiché il grado di sismicità della zona in cui il fabbricato è ubicato è basso ($S = 6$) e l'effetto del sisma risulta non particolarmente gravoso si è ritenuto inutile affrontare il maggior onere computazionale connesso all'analisi modale e si è quindi fatto ricorso a quella statica.

4. Schematizzazione delle azioni

Le azioni considerate sulla costruzione comprendono carichi verticali fissi ed accidentali e forze sismiche orizzontali. I carichi unitari sono stati determinati in base ai pesi dei materiali utilizzati ed ai sovraccarichi imposti dalla normativa per locali di abitazione.

Nell'effettuare l'analisi dei carichi verticali sulle travi, il carico dovuto a solai e sbalzi orditi perpendicolarmente alla trave è stato valutato incrementando il valore corrispondente alla singola campata di solaio isolata ($ql/2$) mediante un coefficiente che tiene conto della continuità strutturale del solaio; il coefficiente di continuità è stato assunto pari a 1.2 per l'appoggio centrale dei solai a due campate, ad 1 per gli appoggi di estremità. Nel caso di solai orditi parallelamente alla trave, si è considerato gravante su di essa

una fascia di 50 cm di solaio, per tener conto del collegamento costituito da armature e travetti di ripartizione.

Le azioni orizzontali sono state valutate in funzione delle masse, supposte concentrate negli impalcati, e delle quote di questi rispetto allo spiccato di fondazione. Nel calcolare l'entità delle masse, si sono utilizzati i seguenti coefficienti di riduzione dei sovraccarichi: $s = 0.33$ per i solai e gli sbalzi, $s = 1.00$ per le scale. Le masse presenti in maniera diffusa in un interpiano (tramezzi, tompagni, pilastri, scale) sono state ripartite tra gli impalcati adiacenti. Si è per semplicità ipotizzato che il punto di applicazione delle forze (baricentro delle masse) coincida a ogni livello col baricentro geometrico dell'impalcato.

Per ciascuno modello geometrico sono state esaminate più condizioni di carico, in modo da massimizzare le sollecitazioni in ciascuna sezione. Per i solai, schematizzati come travi continue, si sono considerate tre condizioni, che prevedono rispettivamente la presenza dei sovraccarichi accidentali in tutte le campate, in quelle dispari ed in quelle pari. Si sono inoltre utilizzati i modelli limite di singola campata incastrata o semplicemente appoggiata, con minori coefficienti di sicurezza.

Per la struttura intelaiata si sono risolte le tre condizioni base, forze in direzione x, forze in direzione y e carichi verticali totali. Queste sono poi state combinate nella maniera più gravosa per determinare le massime caratteristiche di sollecitazione. Alla stessa maniera si è proceduto per il reticolo di travi di fondazione, considerando per esso come carichi le azioni M ed N determinate al piede dei pilastri del telaio spaziale.

5. Schematizzazione dei materiali

Sia per il calcestruzzo che per l'acciaio si è ipotizzato un comportamento elastico lineare, con moduli di Young pari rispettivamente a 28500 N mm^{-2} e 206000 N mm^{-2}. Si è previsto un calcestruzzo con resistenza caratteristica a 28 giorni $R_{ck} = 25$ N mm^{-2}, cui corrispondono le tensioni ammissibili $\bar{\sigma}_c = 8.5$ N mm^{-2} e $\bar{\tau}_{c1} = 1.69$ N mm^{-2}. L'acciaio è del tipo $Fe\ B\ 38k$ controllato in stabilimento, con tensione ammissibile $\bar{\sigma}_s = 215$ N mm^{-2}.

6. Criteri di verifica

La verifica delle sezioni è stata effettuata mediante il metodo delle tensioni ammissibili. Per i pilastri, dato il netto prevalere del momento flettente nel piano di azione del sisma rispetto a quello nel piano ortogonale si

sono effettuate verifiche a pressoflessione retta (e non deviata). Lo sforzo normale dovuto ai carichi verticali è stato determinato in maniera indipendente dal calcolo dello schema spaziale, mediante il criterio di aree di influenza; ad esso si è aggiunto algebricamente lo sforzo prodotto dalle azioni orizzontali.

Si sono verificate le ipotesi poste alla base della modellazione geometrica del telaio spaziale. In particolare, si è valutata la deformazione degli impalcati conseguente alle azioni sismiche distribuite ed alle reazioni concentrate dei pilastri, controllando che essa è effettivamente molto piccola rispetto agli spostamenti relativi degli impalcati. Si sono inoltre valutate in maniera approssimata le azioni e le tensioni che insorgono nella soletta rampante in conseguenza del moto relativo degli impalcati cui essa è collegata.

Si sono controllati gli spostamenti degli impalcati del telaio spaziale conseguenti alle azioni orizzontali. La modesta entità della rotazione conferma l'inutilità dell'effettuazione dell'analisi modale, per lo meno al fine della corretta valutazione del comportamento torsionale dell'edificio.

Si è infine verificata la correttezza della separazione tra reticolo di fondazione e trave in elevazione. Per far ciò si sono presi in esame i cedimenti differenziali dei punti di attacco telaio–trave ottenuti dalla risoluzione del reticolo, constatando la loro modesta entità, sia in assoluto che in termini di variazione presumibile dello stato di sollecitazione nella struttura in elevazione ad essi conseguente.

EDIFICIO PER 54 APPARTAMENTI NEL COMUNE DI GIUGLIANO

ELABORATI NUMERICI

Riportare in dettaglio tutti gli elaborati di calcolo esula dagli obiettivi che ci si è prefissi con questo libro. Il lettore interessato può trovare una diffusa trattazione, teorica e numerica, della progettazione di fabbricati in zona sismica nel volume "Edifici antisismici con struttura intelaiata in cemento armato", appartenente alla stessa collana del presente testo. Si riporta qui solo una sintetica tabella dei carichi unitari, delle masse e delle azioni orizzontali, mentre i valori dei carichi verticali sulle travi possono essere letti direttamente nei tabulati relativi alla risoluzione dello schema spaziale. Ci si sofferma invece sulla preparazione dei dati per l'insieme spaziale di telai e sulla interpretazione dei risultati conseguiti.

1. Carichi unitari

	per carichi verticali $g + p$	ai fini sismici $g + s\,p$
A) SOLAIO		
A1) solaio del piano tipo	7.50 kN m^{-2}	6.16 kN m^{-2}
A2) solaio di copertura	6.70 kN m^{-2}	5.36 kN m^{-2}
A3) solaio del torrino scale	4.70 kN m^{-2}	4.03 kN m^{-2}
B) SBALZI		
B1) sbalzo del piano tipo	8.50 kN m^{-2}	5.82 kN m^{-2}

B2) sbalzo di copertura	8.70 kN m^{-2}	6.02 kN m^{-2}
C) SCALA	9.50 kN m^{-2}	9.50 kN m^{-2}
D) TRAVI (maggior peso rispetto al solaio)		
D1) travi a spessore 50x24	1.35 kN m^{-1}	1.35 kN m^{-1}
D2) travi a spessore 80x24	2.16 kN m^{-1}	2.16 kN m^{-1}
D3) travi a spessore 90x24	2.43 kN m^{-1}	2.43 kN m^{-1}
D4) travi emergenti 30x60	1.20 kN m^{-1}	1.20 kN m^{-1}
E) TOMPAGNO h = 2.75 m	6.00 kN m^{-1}	6.00 kN m^{-1}
F) PILASTRI h = 2.75 m	12.40 kN	12.40 kN

2. Azioni sismiche orizzontali

Gli impalcati sono stati numerati progressivamente dal basso verso l'alto. A causa dello sfalsamento, vengono a trovarsi uno sull'altro gli impalcati con numero dispari e quelli con numero pari. Si è considerato come nono impalcato la copertura del torrino scale. L'edificio è stato riferito in pianta ad un sistema di assi cartesiani xy, con l'origine coincidente con lo spigolo inferiore sinistro del fabbricato, l'asse x orizzontale ed orientato verso destra, l'asse y verticale ed orientato verso l'alto.

Nel determinare l'entità delle azione orizzontali, si sono assunti i seguenti valori per i coefficienti previsti dalla normativa:

$$C = (S - 2)/100 = 0.04$$
$$R = 1 \qquad \varepsilon = 1 \qquad \beta = 1 \qquad I = 1$$

Si sono ricavati i valori riassunti nella seguente tabella:

Impalcato	Area	W_i	h_i	γ_i	F_i	x_i	y_i
IX	20.5 m^2	148.2 kN	15.95 m	1.784	10.58 kN	11.10 m	8.85 m
VIII	94.3 m^2	717.5 kN	14.75 m	1.650	47.35 kN	11.10 m	15.30 m
VII	265.5 m^2	1944.1 kN	13.25 m	1.482	115.24 kN	11.10 m	5.86 m
VI	101.9 m^2	934.3 kN	11.75 m	1.325	49.11 kN	11.10 m	15.30 m
V	265.5 m^2	2367.8 kN	10.25 m	1.146	108.58 kN	11.10 m	5.86 m
IV	101.9 m^2	934.3 kN	8.75 m	0.979	36.57 kN	11.10 m	15.30 m
III	265.5 m^2	2367.8 kN	7.25 m	0.811	76.80 kN	11.10 m	5.86 m
II	101.9 m^2	1049.9 kN	5.75 m	0.643	27.01 kN	11.10 m	15.30 m
I	265.5 m^2	2510.1 kN	4.25 m	0.475	47.73 kN	11.10 m	5.86 m

3. Preparazione dei dati per lo schema di insieme spaziale di aste

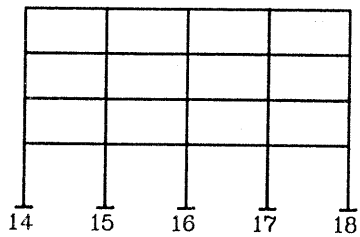
Per l'analisi dell'ossatura spaziale si è utilizzato il programma TELSPA, descritto nella seconda parte del testo. Il sistema di riferimento globale ha il piano xy coincidente col piano di estradosso della fondazione; la posizione e l'origine degli assi è stata già definita nel paragrafo precedente.

La struttura è costituita da nove telai, quattro paralleli all'asse x e cinque all'asse y, mostrati schematicamente nella figura 6. Il sistema di riferimento locale di ciascuno di essi ha l'asse y' coincidente con l'asse del suo primo pilastro. Nonostante lo sfalsamento degli impalcati tutti i telai possono essere ricondotti a schemi a maglie rettangolari, anche se con numerose aste mancanti (indicate in figura 7 con linea tratteggiata). Il numero di traversi inestensibili varia da telaio a telaio, da un minimo di 4 a un massimo di 9.

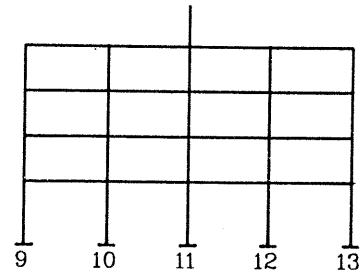
La definizione degli schemi non presenta in generale problemi. Nel caso in esame è però interessante evidenziare una difficoltà in cui ci si è imbattuti, dovuta alla conformazione molto particolare del nono impalcato (torrino scale), costituito esclusivamente da un doppio sbalzo che grava sulla trave 6–11. I pilastri 6 e 11 formano due singole mensole all'ultimo piano dei telai 2 e 3. Nell'impostazione data al telaio a maglie rettangolari si è individuato come traverso inestensibile un insieme di nodi contigui di un traverso geometrico collegati da aste; un nodo su quale non converge alcuna trave è stato invece considerato come nodo isolato, non appartenente ad un traverso inestensibile. Per forzare il programma a riconoscere i nodi in testa ai pilastri 6 e 11 come appartenenti ad un traverso (e quindi al nono impalcato) è stato necessario aggiungere travi fittizie di sezione 0x0, irrilevanti dal punto di vista statico ma non per la definizione dei traversi.

I dati che descrivono la geometria dello schema e le tre condizioni di carico esaminate (forze in direzione x, forze in direzione y e carichi verticali totali) sono memorizzati nel file EDIFICIO e qui di seguito riportati. Si noti che i carichi verticali sono indicati col segno meno perché diretti verso il basso e quindi in senso opposto rispetto a quello definito positivo per l'asse locale y'.

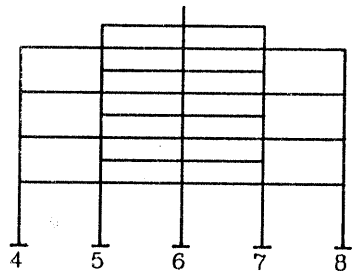
```
EDIFICIO PER 54 APPARTAMENTI NEL COMUNE DI GIUGLIANO
9,9,6
R, 0.30, 0.45,X
R, 0.15, 6.15,X
R, 0.30,11.55,X
R, 5.85,17.55,X
R, 0.15, 0.45,Y
```



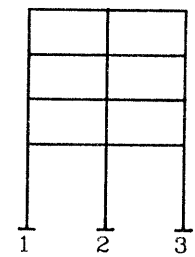
Telaio 1



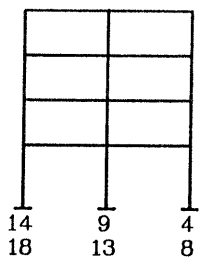
Telaio 2



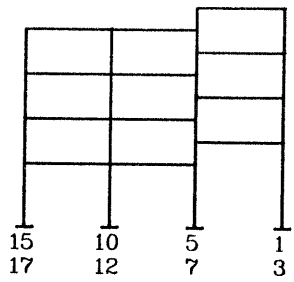
Telaio 3



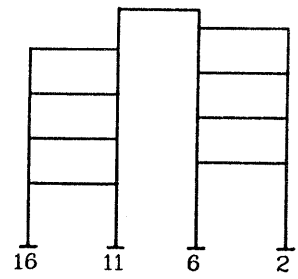
Telaio 4



Telaio 5 e 9



Telaio 6 e 8



Telaio 7

Fig. 6 — Schemi dei telai

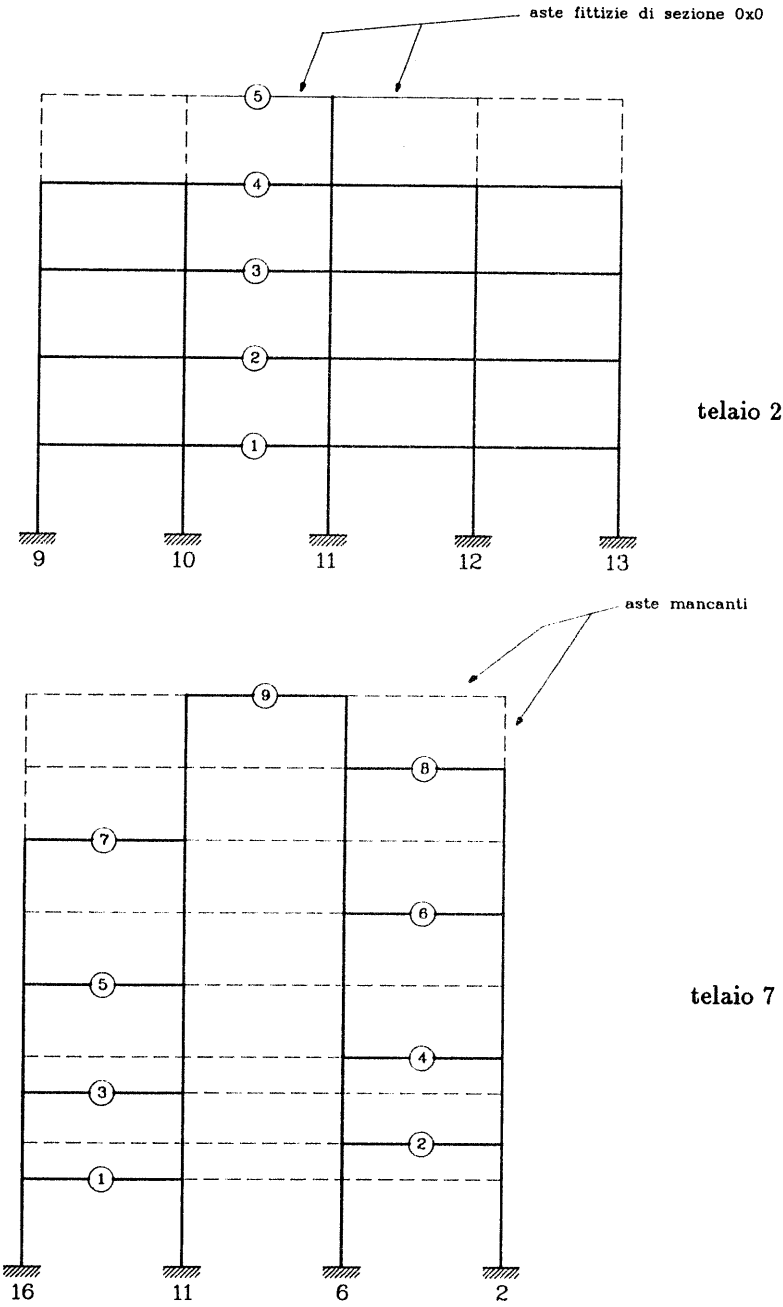


Fig. 7 — Modellazione dei telai come schemi a maglie rettangolari

```
R, 5.70, 0.30,Y
R,11.10, 0.30,Y
R,16.50, 0.30,Y
R,22.05, 0.45,Y
28500
R,0.30,0.60
R,0.60,0.30
R,0.50,0.24
R,0.80,0.24
R,0.90,0.24
R,0.00,0.00
5,4,0
14,15,16,17,18
3.00,3.00,3.00,4.25
5.40,5.40,5.40,5.40
1,1,1,1,  1,1,1,1,  1,1,1,1,  1,1,1,1
1,2,2,2,1,1,2,2,2,1,1,2,2,2,1,1,2,2,2,1
1,3,5,7
5,5,0
9,10,11,12,13
2.70,3.00,3.00,3.00,4.25
5.55,5.40,5.40,5.55
0,6,6,0,  4,1,1,4,  4,1,1,4,  4,1,1,4,  4,1,1,4
0,0,1,0,0,2,2,1,2,2,2,2,1,2,2,2,2,1,2,2,2,2,1,2,2
1,3,5,7,9
5,9,0
4,5,6,7,8
1.20,1.50,1.50,1.50,1.50,1.50,1.50,1.50,4.25
5.40,5.40,5.40,5.40
0,6,6,0,  0,1,1,0,  1,1,1,1,  0,1,1,0
1,1,1,1,  0,1,1,0,  1,1,1,1,  0,1,1,0,  1,1,1,1
0,0,1,0,0,0,2,1,2,0,1,2,1,2,1,1,2,1,2,1
1,2,1,2,1,1,2,1,2,1,1,2,1,2,1,1,2,1,2,1,1,2,1,2,1
1,2,3,4,5,6,7,8,9
3,4,0
1,2,3
3.00,3.00,3.00,5.75
5.25,5.25
1,1,  1,1,  1,1,  1,1
1,1,1,1,1,1,1,1,1,1,1,1
2,4,6,8
3,4,0
14,9,4
3.00,3.00,3.00,4.25
5.55,5.55
1,1,  1,1,  1,1,  1,1
2,1,2,2,1,2,2,1,2,2,1,2
1,3,5,7
4,8,0
15,10,5,1
1.50,1.50,1.50,1.50,1.50,1.50,1.50,4.25
5.70,5.70,5.85
0,0,1,  1,1,0,  0,0,1,  1,1,0,  0,0,1,  1,1,0,  0,0,1,  1,1,0
0,0,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2
1,2,3,4,5,6,7,8
4,9,0
16,11,6,2
1.20,1.50,1.50,1.50,1.50,1.50,1.50,1.50,4.25
5.85,5.40,6.00
```

```
0,5,0,  0,0,3,  1,0,0,  0,0,3,  1,0,0,  0,0,3,  1,0,0,  0,0,3,  1,0,0
0,2,2,0,0,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2
1,2,3,4,5,6,7,8,9
4,8,0
17,12,7,3
1.50,1.50,1.50,1.50,1.50,1.50,1.50,1.50,4.25
5.70,5.70,5.85
0,0,1,  1,1,0,  0,0,1,  1,1,0,  0,0,1,  1,1,0,  0,0,1,  1,1,0
0,0,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1,1,2
1,2,3,4,5,6,7,8
3,4,0
18,13,8
3.00,3.00,3.00,4.25
5.55,5.55
1,1,  1,1,  1,1,  1,1
2,1,2,2,1,2,2,1,2,2,1,2
1,3,5,7
1,0
11.10, 8.85, 10.58,0,0
11.10,15.30, 47.35,0,0
11.10, 5.86,115.24,0,0
11.10,15.30, 49.11,0,0
11.10, 5.86,108.58,0,0
11.10,15.30, 36.57,0,0
11.10, 5.86, 76.80,0,0
11.10,15.30, 27.01,0,0
11.10, 5.86, 47.73,0,0
1,0
11.10, 8.85,0, 10.58,0
11.10,15.30,0, 47.35,0
11.10, 5.86,0,115.24,0
11.10,15.30,0, 49.11,0
11.10, 5.86,0,108.58,0
11.10,15.30,0, 36.57,0
11.10, 5.86,0, 76.80,0
11.10,15.30,0, 27.01,0
11.10, 5.86,0, 47.73,0
0,1
1,0,0,0
-10.8,-23.3,-23.3,-10.8
-10.6,-32.0,-32.0,-10.6
-10.6,-32.0,-32.0,-10.6
-10.6,-32.0,-32.0,-10.6
1,0,0,0
-14.3,-45.6,-45.6,-14.3
-15.7,-50.0,-50.0,-15.7
-15.7,-50.0,-50.0,-15.7
-15.7,-50.0,-50.0,-15.7
1,0,0,0
-27.7,-27.7, -10.8,-23.7,-23.7,-10.8
-33.1,-33.1, -10.6,-25.2,-25.2,-10.6
-33.1,-33.1, -10.6,-25.2,-25.2,-10.6
-33.1,-33.1, -10.6,-25.2,-25.2,-10.6
1,0,0,0
-39.1,-39.1
-47.4,-47.4
-47.4,-47.4
-47.4,-47.4
1,0,0,0
```

```
-19.3,-19.3
-27.5,-27.5
-27.5,-27.5
-27.5,-27.5
1,0,0,0
-6.6, -24.7,-24.7
-7.8, -27.5,-27.5
-7.8, -27.5,-27.5
-7.8, -27.5,-27.5
1,0,0,0
-19.4
-12.6,  -9.9
-12.6, -11.0
-12.6, -11.0
-12.6, -11.0
1,0,0,0
-6.6, -24.7,-24.7
-7.8, -27.5,-27.5
-7.8, -27.5,-27.5
-7.8, -27.5,-27.5
1,0,0,0
-19.3,-19.3
-27.5,-27.5
-27.5,-27.5
-27.5,-27.5
```

4. Risultati ottenuti

Si mostra nelle pagine seguenti una sintesi del tabulato fornito dall'esecuzione del programma, indicando con una serie di puntini le parti non riportate.

File dati: EDIFICIO data: 05/03/1990 ora: 00:09:09

EDIFICIO PER 54 APPARTAMENTI NEL COMUNE DI GIUGLIANO

DATI GEOMETRICI DELL'INSIEME SPAZIALE

telaio	tipo	x (m)	y (m)	angolo (gradi)
1	R	0.30	0.45	0.00
2	R	0.15	6.15	0.00
3	R	0.30	11.55	0.00
4	R	5.85	17.55	0.00
5	R	0.15	0.45	90.00
6	R	5.70	0.30	90.00
7	R	11.10	0.30	90.00
8	R	16.50	0.30	90.00
9	R	22.05	0.45	90.00

TABELLA DEI TIPI DI SEZIONE

sezione	forma	b m	h m	area m2	inerzia m4
1	R	0.30	0.60	0.1800	0.005400
2	R	0.60	0.30	0.1800	0.001350
3	R	0.50	0.24	0.1200	0.000576
4	R	0.80	0.24	0.1920	0.000922
5	R	0.90	0.24	0.2160	0.001037
6	R	0.00	0.00	0.0000	0.000000

Modulo di elasticita' normale = 28500 N/mm2

TELAIO N. 1

ALTEZZA DEI PIANI

piano	altezza (m)
4	3.00
3	3.00
2	3.00
1	4.25

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
14 - 15	5.40	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
15 - 16	5.40	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
16 - 17	5.40	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
17 - 18	5.40	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
14	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
15	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350

	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350
16	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350
17	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350
18	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
4	7
3	5
2	3
1	1

TELAIO N. 2

ALTEZZA DEI PIANI

piano	altezza (m)
5	2.70
4	3.00
3	3.00
2	3.00
1	4.25

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
9 - 10	5.55	5	0	----	----	--------	4	4	0.80	0.24	0.000922
		3	4	0.80	0.24	0.000922	2	4	0.80	0.24	0.000922
		1	4	0.80	0.24	0.000922					
10 - 11	5.40	5	6	0.00	0.00	0.000000	4	1	0.30	0.60	0.005400
		3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
		1	1	0.30	0.60	0.005400					
11 - 12	5.40	5	6	0.00	0.00	0.000000	4	1	0.30	0.60	0.005400
		3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
		1	1	0.30	0.60	0.005400					
12 - 13	5.55	5	0	----	----	--------	4	4	0.80	0.24	0.000922
		3	4	0.80	0.24	0.000922	2	4	0.80	0.24	0.000922
		1	4	0.80	0.24	0.000922					

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
9	5	0	----	----	--------	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					
10	5	0	----	----	--------	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					
11	5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
	3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
	1	1	0.30	0.60	0.005400					
12	5	0	----	----	--------	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					
13	5	0	----	----	--------	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
5	9
4	7
3	5
2	3
1	1

TELAIO N. 3

ALTEZZA DEI PIANI

piano	altezza (m)
9	1.20
8	1.50
7	1.50
6	1.50
5	1.50
4	1.50
3	1.50
2	1.50
1	4.25

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
4 - 5	5.40	9	0	----	----	--------	8	0	----	----	--------
		7	1	0.30	0.60	0.005400	6	0	----	----	--------
		5	1	0.30	0.60	0.005400	4	0	----	----	--------
		3	1	0.30	0.60	0.005400	2	0	----	----	--------
		1	1	0.30	0.60	0.005400					
5 - 6	5.40	9	6	0.00	0.00	0.000000	8	1	0.30	0.60	0.005400
		7	1	0.30	0.60	0.005400	6	1	0.30	0.60	0.005400
		5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
		3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
		1	1	0.30	0.60	0.005400					
6 - 7	5.40	9	6	0.00	0.00	0.000000	8	1	0.30	0.60	0.005400
		7	1	0.30	0.60	0.005400	6	1	0.30	0.60	0.005400
		5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
		3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
		1	1	0.30	0.60	0.005400					
7 - 8	5.40	9	0	----	----	--------	8	0	----	----	--------
		7	1	0.30	0.60	0.005400	6	0	----	----	--------
		5	1	0.30	0.60	0.005400	4	0	----	----	--------
		3	1	0.30	0.60	0.005400	2	0	----	----	--------
		1	1	0.30	0.60	0.005400					

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
4	9	0	----	----	--------	8	0	----	----	--------
	7	1	0.30	0.60	0.005400	6	1	0.30	0.60	0.005400
	5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
	3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
	1	1	0.30	0.60	0.005400					
5	9	0	----	----	--------	8	2	0.60	0.30	0.001350
	7	2	0.60	0.30	0.001350	6	2	0.60	0.30	0.001350
	5	2	0.60	0.30	0.001350	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					
6	9	1	0.30	0.60	0.005400	8	1	0.30	0.60	0.005400
	7	1	0.30	0.60	0.005400	6	1	0.30	0.60	0.005400
	5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
	3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
	1	1	0.30	0.60	0.005400					
7	9	0	----	----	--------	8	2	0.60	0.30	0.001350
	7	2	0.60	0.30	0.001350	6	2	0.60	0.30	0.001350
	5	2	0.60	0.30	0.001350	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					

8	9	0	----	----	--------	8	0	----	----	--------
	7	1	0.30	0.60	0.005400	6	1	0.30	0.60	0.005400
	5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
	3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
	1	1	0.30	0.60	0.005400					

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
9	9
8	8
7	7
6	6
5	5
4	4
3	3
2	2
1	1

TELAIO N. 4

ALTEZZA DEI PIANI

piano	altezza (m)
4	3.00
3	3.00
2	3.00
1	5.75

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
1 - 2	5.25	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
2 - 3	5.25	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
1	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400

2	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
3	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
4	8
3	6
2	4
1	2

TELAIO N. 5

ALTEZZA DEI PIANI

piano	altezza (m)
4	3.00
3	3.00
2	3.00
1	4.25

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
14 - 9	5.55	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
9 - 4	5.55	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
		2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
14	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350
9	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
4	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
4	7
3	5
2	3
1	1

TELAIO N. 6

ALTEZZA DEI PIANI

piano	altezza (m)
8	1.50
7	1.50
6	1.50
5	1.50
4	1.50
3	1.50
2	1.50
1	4.25

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
15 - 10	5.70	8	0	----	----	--------	7	1	0.30	0.60	0.005400
		6	0	----	----	--------	5	1	0.30	0.60	0.005400
		4	0	----	----	--------	3	1	0.30	0.60	0.005400
		2	0	----	----	--------	1	1	0.30	0.60	0.005400
10 - 5	5.70	8	0	----	----	--------	7	1	0.30	0.60	0.005400
		6	0	----	----	--------	5	1	0.30	0.60	0.005400
		4	0	----	----	--------	3	1	0.30	0.60	0.005400
		2	0	----	----	--------	1	1	0.30	0.60	0.005400
5 - 1	5.85	8	1	0.30	0.60	0.005400	7	0	----	----	--------
		6	1	0.30	0.60	0.005400	5	0	----	----	--------
		4	1	0.30	0.60	0.005400	3	0	----	----	--------
		2	1	0.30	0.60	0.005400	1	0	----	----	--------

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
15	8	0	----	----	--------	7	1	0.30	0.60	0.005400
	6	1	0.30	0.60	0.005400	5	1	0.30	0.60	0.005400

	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
10	8	0	----	----	--------	7	1	0.30	0.60	0.005400
	6	1	0.30	0.60	0.005400	5	1	0.30	0.60	0.005400
	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
5	8	1	0.30	0.60	0.005400	7	1	0.30	0.60	0.005400
	6	1	0.30	0.60	0.005400	5	1	0.30	0.60	0.005400
	4	1	0.30	0.60	0.005400	3	1	0.30	0.60	0.005400
	2	1	0.30	0.60	0.005400	1	1	0.30	0.60	0.005400
1	8	2	0.60	0.30	0.001350	7	2	0.60	0.30	0.001350
	6	2	0.60	0.30	0.001350	5	2	0.60	0.30	0.001350
	4	2	0.60	0.30	0.001350	3	2	0.60	0.30	0.001350
	2	2	0.60	0.30	0.001350	1	2	0.60	0.30	0.001350

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
8	8
7	7
6	6
5	5
4	4
3	3
2	2
1	1

TELAIO N. 7

ALTEZZA DEI PIANI

piano	altezza (m)
9	1.20
8	1.50
7	1.50
6	1.50
5	1.50
4	1.50
3	1.50
2	1.50
1	4.25

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
16 - 11	5.85	9	0	----	----	--------	8	0	----	----	--------

		7	1	0.30	0.60	0.005400	6	0	----	----	--------
		5	1	0.30	0.60	0.005400	4	0	----	----	--------
		3	1	0.30	0.60	0.005400	2	0	----	----	--------
		1	1	0.30	0.60	0.005400					
11 - 6	5.40	9	5	0.90	0.24	0.001037	8	0	----	----	--------
		7	0	----	----	--------	6	0	----	----	--------
		5	0	----	----	--------	4	0	----	----	--------
		3	0	----	----	--------	2	0	----	----	--------
		1	0	----	----	--------					
6 - 2	6.00	9	0	----	----	--------	8	3	0.50	0.24	0.000576
		7	0	----	----	--------	6	3	0.50	0.24	0.000576
		5	0	----	----	--------	4	3	0.50	0.24	0.000576
		3	0	----	----	--------	2	3	0.50	0.24	0.000576
		1	0	----	----	--------					

SEZIONI DEI PILASTRI

pilastro	piano	sez.	b m	h m	inerzia m4	piano	sez.	b m	h m	inerzia m4
16	9	0	----	----	--------	8	0	----	----	--------
	7	1	0.30	0.60	0.005400	6	1	0.30	0.60	0.005400
	5	1	0.30	0.60	0.005400	4	1	0.30	0.60	0.005400
	3	1	0.30	0.60	0.005400	2	1	0.30	0.60	0.005400
	1	1	0.30	0.60	0.005400					
11	9	2	0.60	0.30	0.001350	8	2	0.60	0.30	0.001350
	7	2	0.60	0.30	0.001350	6	2	0.60	0.30	0.001350
	5	2	0.60	0.30	0.001350	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					
6	9	2	0.60	0.30	0.001350	8	2	0.60	0.30	0.001350
	7	2	0.60	0.30	0.001350	6	2	0.60	0.30	0.001350
	5	2	0.60	0.30	0.001350	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					
2	9	0	----	----	--------	8	2	0.60	0.30	0.001350
	7	2	0.60	0.30	0.001350	6	2	0.60	0.30	0.001350
	5	2	0.60	0.30	0.001350	4	2	0.60	0.30	0.001350
	3	2	0.60	0.30	0.001350	2	2	0.60	0.30	0.001350
	1	2	0.60	0.30	0.001350					

CORRISPONDENZA TRA TRAVERSI ED IMPALCATI

traverso	impalcato
9	9
8	8
7	7
6	6
5	5

4	4
3	3
2	2
1	1

. . . . .

SCHEMA DI CARICO N. 1

CARICHI AGENTI SULL'INSIEME SPAZIALE

Impalcato	Xm (m)	Ym (m)	Fx (kN)	Fy (kN)	Mz (kNm)
9	11.10	8.85	10.580	0.000	0.000
8	11.10	15.30	47.350	0.000	0.000
7	11.10	5.86	115.240	0.000	0.000
6	11.10	15.30	49.110	0.000	0.000
5	11.10	5.86	108.580	0.000	0.000
4	11.10	15.30	36.570	0.000	0.000
3	11.10	5.86	76.800	0.000	0.000
2	11.10	15.30	27.010	0.000	0.000
1	11.10	5.86	47.730	0.000	0.000

RISULTATI

COMPONENTI DI MOVIMENTO DEGLI IMPALCATI

impalcato	Vx (mm)	Vy (mm)	rotaz.x1000
9	7.726	-0.736	0.066
8	6.811	0.055	-0.005
7	6.720	0.001	-0.000
6	6.437	0.007	-0.001
5	6.037	0.022	-0.002
4	5.494	0.059	-0.005
3	4.857	0.081	-0.007
2	4.039	0.122	-0.011
1	3.145	0.170	-0.015

. . . . .

TELAIO N. 3

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
9	6.959
8	6.868

7	6.721
6	6.444
5	6.059
4	5.555
3	4.942
2	4.165
1	3.322

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
4	9	0.000	0.000	8	0.000	0.000
	7	0.000	-0.126	6	0.000	-0.238
	5	0.000	-0.245	4	0.000	-0.411
	3	0.000	-0.347	2	0.000	-0.565
	1	0.000	-0.633			
5	9	0.000	0.000	8	0.000	-0.044
	7	0.000	-0.044	6	0.000	-0.160
	5	0.000	-0.095	4	0.000	-0.269
	3	0.000	-0.152	2	0.000	-0.405
	1	0.000	-0.113			
6	9	0.000	-0.084	8	0.000	-0.059
	7	0.000	-0.116	6	0.000	-0.170
	5	0.000	-0.249	4	0.000	-0.283
	3	0.000	-0.392	2	0.000	-0.408
	1	0.000	-0.536			
7	9	0.000	0.000	8	0.000	-0.044
	7	0.000	-0.044	6	0.000	-0.160
	5	0.000	-0.095	4	0.000	-0.269
	3	0.000	-0.152	2	0.000	-0.405
	1	0.000	-0.113			
8	9	0.000	0.000	8	0.000	0.000
	7	0.000	-0.126	6	0.000	-0.238
	5	0.000	-0.245	4	0.000	-0.411
	3	0.000	-0.347	2	0.000	-0.565
	1	0.000	-0.633			

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
4 - 5	9	-----	-----	-----	-----
4 - 5	8	-----	-----	-----	-----
4 - 5	7	16.89	-12.25	-5.40	-5.40
4 - 5	6	-----	-----	-----	-----
4 - 5	5	33.37	-24.77	-10.77	-10.77
4 - 5	4	-----	-----	-----	-----
4 - 5	3	48.24	-37.12	-15.81	-15.81

4 - 5	2	-----	-----	-----	-----
4 - 5	1	78.67	-49.03	-23.65	-23.65
5 - 6	9	-----	-----	-----	-----
5 - 6	8	8.34	-9.25	-3.26	-3.26
5 - 6	7	11.70	-15.79	-5.09	-5.09
5 - 6	6	27.94	-28.52	-10.45	-10.45
5 - 6	5	24.97	-33.76	-10.88	-10.88
5 - 6	4	46.80	-47.57	-17.48	-17.48
5 - 6	3	39.70	-53.38	-17.24	-17.24
5 - 6	2	69.43	-69.62	-25.75	-25.75
5 - 6	1	43.44	-67.48	-20.54	-20.54
6 - 7	9	-----	-----	-----	-----
6 - 7	8	9.25	-8.34	-3.26	-3.26
6 - 7	7	15.79	-11.70	-5.09	-5.09
6 - 7	6	28.52	-27.94	-10.45	-10.45
6 - 7	5	33.76	-24.97	-10.88	-10.88
6 - 7	4	47.57	-46.80	-17.48	-17.48
6 - 7	3	53.38	-39.70	-17.24	-17.24
6 - 7	2	69.62	-69.43	-25.75	-25.75
6 - 7	1	67.48	-43.44	-20.54	-20.54
7 - 8	9	-----	-----	-----	-----
7 - 8	8	-----	-----	-----	-----
7 - 8	7	12.25	-16.89	-5.40	-5.40
7 - 8	6	-----	-----	-----	-----
7 - 8	5	24.77	-33.37	-10.77	-10.77
7 - 8	4	-----	-----	-----	-----
7 - 8	3	37.12	-48.24	-15.81	-15.81
7 - 8	2	-----	-----	-----	-----
7 - 8	1	49.03	-78.67	-23.65	-23.65

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
4	9	-----	-----	-----	-----
4	8	-----	-----	-----	-----
4	7	16.89	6.13	7.18	5.40
4	6	6.13	-4.63	7.18	5.40
4	5	28.73	5.22	15.68	16.16
4	4	5.22	-18.30	15.68	16.16
4	3	29.94	14.69	10.17	31.97
4	2	14.69	-0.57	10.17	31.97
4	1	78.10	-123.97	47.54	55.62
5	9	-----	-----	-----	-----
5	8	8.34	-8.29	11.09	3.26
5	7	15.65	-9.73	16.92	2.95
5	6	18.21	-21.56	26.51	13.40
5	5	28.19	-19.23	31.61	13.52
5	4	27.57	-33.57	40.76	30.99
5	3	43.25	-30.28	49.01	32.42
5	2	39.16	-54.11	62.18	58.17
5	1	38.35	-40.41	18.53	55.06

6	9	0.00	-6.35	5.29	0.00
6	8	12.15	-0.48	8.42	0.00
6	7	31.09	-20.02	34.08	-0.00
6	6	37.01	-20.86	38.58	-0.00
6	5	46.66	-39.71	57.58	-0.00
6	4	55.44	-32.98	58.94	-0.00
6	3	73.78	-70.50	96.19	-0.00
6	2	68.73	-42.65	74.25	-0.00
6	1	92.31	-131.07	52.56	-0.00
7	9	-----	-----	-----	-----
7	8	8.34	-8.29	11.09	-3.26
7	7	15.65	-9.73	16.92	-2.95
7	6	18.21	-21.56	26.51	-13.40
7	5	28.19	-19.23	31.61	-13.52
7	4	27.57	-33.57	40.76	-30.99
7	3	43.25	-30.28	49.01	-32.42
7	2	39.16	-54.11	62.18	-58.17
7	1	38.35	-40.41	18.53	-55.06
8	9	-----	-----	-----	-----
8	8	-----	-----	-----	-----
8	7	16.89	6.13	7.17	-5.40
8	6	6.13	-4.63	7.17	-5.40
8	5	28.73	5.22	15.68	-16.16
8	4	5.22	-18.30	15.68	-16.16
8	3	29.94	14.69	10.17	-31.97
8	2	14.69	-0.57	10.17	-31.97
8	1	78.10	-123.97	47.54	-55.62

TELAIO N. 4

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
4	6.898
3	6.448
2	5.587
1	4.231

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
1	4	0.000	-0.079	3	0.000	-0.171
	2	0.000	-0.277	1	0.000	-0.489
2	4	0.000	-0.053	3	0.000	-0.132
	2	0.000	-0.230	1	0.000	-0.323
3	4	0.000	-0.079	3	0.000	-0.171
	2	0.000	-0.277	1	0.000	-0.489

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
1 - 2	4	12.40	-10.86	-4.43	-4.43
1 - 2	3	27.80	-25.55	-10.16	-10.16
1 - 2	2	45.97	-43.20	-16.99	-16.99
1 - 2	1	76.23	-66.48	-27.18	-27.18
2 - 3	4	10.86	-12.40	-4.43	-4.43
2 - 3	3	25.55	-27.80	-10.16	-10.16
2 - 3	2	43.20	-45.97	-16.99	-16.99
2 - 3	1	66.48	-76.23	-27.18	-27.18

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
1	4	12.40	-3.01	5.14	4.43
1	3	24.79	-13.88	12.89	14.59
1	2	32.09	-10.38	14.16	31.58
1	1	65.85	-92.01	27.45	58.76
2	4	21.73	-13.58	11.77	0.00
2	3	37.53	-27.54	21.69	-0.00
2	2	58.86	-49.34	36.06	-0.00
2	1	83.63	-100.90	32.09	0.00
3	4	12.40	-3.01	5.14	-4.43
3	3	24.79	-13.88	12.89	-14.59
3	2	32.09	-10.38	14.16	-31.58
3	1	65.85	-92.01	27.45	-58.76

. . . . .

TELAIO N. 6

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
8	0.027
7	0.001
6	0.003
5	0.011
4	0.029
3	0.040
2	0.059
1	0.083

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
15	8	0.000	0.000	7	0.000	0.001
	6	0.000	0.003	5	0.000	0.005
	4	0.000	0.010	3	0.000	0.013
	2	0.000	0.019	1	0.000	-0.002
10	8	0.000	0.000	7	0.000	0.002
	6	0.000	0.004	5	0.000	0.003
	4	0.000	0.011	3	0.000	0.010
	2	0.000	0.019	1	0.000	-0.001
5	8	0.000	-0.018	7	0.000	-0.008
	6	0.000	0.004	5	0.000	0.008
	4	0.000	0.009	3	0.000	0.007
	2	0.000	0.017	1	0.000	0.003
1	8	0.000	0.002	7	0.000	-0.012
	6	0.000	-0.002	5	0.000	0.012
	4	0.000	0.005	3	0.000	0.015
	2	0.000	-0.004	1	0.000	-0.012

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
15 - 10	8	-----	-----	-----	-----
15 - 10	7	-0.28	0.33	0.11	0.11
15 - 10	6	-----	-----	-----	-----
15 - 10	5	-0.67	0.60	0.22	0.22
15 - 10	4	-----	-----	-----	-----
15 - 10	3	-1.91	1.75	0.64	0.64
15 - 10	2	-----	-----	-----	-----
15 - 10	1	0.32	-0.28	-0.11	-0.11
10 - 5	8	-----	-----	-----	-----
10 - 5	7	0.15	-0.68	-0.15	-0.15
10 - 5	6	-----	-----	-----	-----
10 - 5	5	-0.79	1.04	0.32	0.32
10 - 5	4	-----	-----	-----	-----
10 - 5	3	-1.45	1.31	0.48	0.48
10 - 5	2	-----	-----	-----	-----
10 - 5	1	0.00	0.24	0.04	0.04
5 - 1	8	1.76	-0.68	-0.42	-0.42
5 - 1	7	-----	-----	-----	-----
5 - 1	6	-0.37	0.04	0.07	0.07
5 - 1	5	-----	-----	-----	-----
5 - 1	4	-1.24	1.04	0.39	0.39
5 - 1	3	-----	-----	-----	-----
5 - 1	2	-1.61	0.52	0.36	0.36
5 - 1	1	-----	-----	-----	-----

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
15	8	-----	-----	-----	-----
15	7	-0.28	-0.16	-0.07	-0.11
15	6	-0.16	-0.05	-0.07	-0.11
15	5	-0.72	-0.42	-0.20	-0.33
15	4	-0.42	-0.12	-0.20	-0.33
15	3	-2.03	0.77	-1.87	-0.97
15	2	0.77	3.57	-1.87	-0.97
15	1	3.90	-4.06	1.87	-0.86
10	8	-----	-----	-----	-----
10	7	-0.19	-0.04	-0.10	0.25
10	6	-0.04	0.10	-0.10	0.25
10	5	-1.29	-0.33	-0.64	0.15
10	4	-0.33	0.62	-0.64	0.15
10	3	-2.57	0.58	-2.10	0.31
10	2	0.58	3.72	-2.10	0.31
10	1	4.01	-4.12	1.91	0.16
5	8	1.76	-3.90	3.78	0.42
5	7	-3.22	0.77	-2.66	0.27
5	6	0.40	-1.14	1.03	0.20
5	5	-2.19	1.95	-2.76	0.52
5	4	0.71	-0.31	0.68	0.14
5	3	-1.62	-0.40	-0.81	0.62
5	2	-2.01	4.90	-4.60	0.26
5	1	4.66	-4.44	2.14	0.30
1	8	0.68	0.06	0.42	-0.42
1	7	0.06	-0.57	0.42	-0.42
1	6	-0.61	-0.09	-0.35	-0.35
1	5	-0.09	0.43	-0.35	-0.35
1	4	-0.61	0.11	-0.48	0.04
1	3	0.11	0.84	-0.48	0.04
1	2	0.32	0.14	0.12	0.40
1	1	0.14	-0.37	0.12	0.40

. . . . .

SCHEMA DI CARICO N. 2

CARICHI AGENTI SULL'INSIEME SPAZIALE

Impalcato	Xm (m)	Ym (m)	Fx (kN)	Fy (kN)	Mz (kNm)
9	11.10	8.85	0.000	10.580	0.000
8	11.10	15.30	0.000	47.350	0.000
7	11.10	5.86	0.000	115.240	0.000
6	11.10	15.30	0.000	49.110	0.000
5	11.10	5.86	0.000	108.580	0.000
4	11.10	15.30	0.000	36.570	0.000
3	11.10	5.86	0.000	76.800	0.000
2	11.10	15.30	0.000	27.010	0.000
1	11.10	5.86	0.000	47.730	0.000

RISULTATI

COMPONENTI DI MOVIMENTO DEGLI IMPALCATI

impalcato	Vx (mm)	Vy (mm)	rotaz.x1000
9	0.000	7.587	0.000
8	0.000	7.362	0.000
7	0.000	7.075	0.000
6	0.000	6.752	0.000
5	0.000	6.295	0.000
4	0.000	5.699	0.000
3	0.000	4.971	0.000
2	0.000	4.132	0.000
1	0.000	3.108	0.000

. . . . .

TELAIO N. 5

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
4	7.075
3	6.295
2	4.971
1	3.108

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
14	4	0.000	-0.060	3	0.000	-0.155
	2	0.000	-0.239	1	0.000	-0.272
9	4	0.000	-0.115	3	0.000	-0.238
	2	0.000	-0.356	1	0.000	-0.495
4	4	0.000	-0.060	3	0.000	-0.155
	2	0.000	-0.239	1	0.000	-0.272

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
14 - 9	4	12.98	-16.05	-5.23	-5.23
14 - 9	3	30.40	-34.98	-11.78	-11.78
14 - 9	2	46.27	-52.77	-17.84	-17.84
14 - 9	1	57.61	-70.02	-23.00	-23.00
9 - 4	4	16.05	-12.98	-5.23	-5.23
9 - 4	3	34.98	-30.40	-11.78	-11.78
9 - 4	2	52.77	-46.27	-17.84	-17.84
9 - 4	1	70.02	-57.61	-23.00	-23.00

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
14	4	12.98	-10.53	7.84	5.23
14	3	19.87	-17.72	12.53	17.01
14	2	28.55	-27.72	18.76	34.86
14	1	29.89	-34.81	15.22	57.85
9	4	32.10	-19.48	17.19	0.00
9	3	50.48	-38.33	29.61	-0.00
9	2	67.20	-52.91	40.04	-0.00
9	1	87.13	-123.01	49.44	-0.00
4	4	12.98	-10.53	7.84	-5.23
4	3	19.87	-17.72	12.53	-17.01
4	2	28.55	-27.72	18.76	-34.86
4	1	29.89	-34.81	15.22	-57.85

. . . . .

TELAIO N. 3 SCHEMA DI CARICO N. 3

CARICHI VERTICALI

campata	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m
4 - 5	9	----	8	----	7	-10.8	6	----
	5	-10.6	4	----	3	-10.6	2	----
	1	-10.6						
5 - 6	9	----	8	-27.7	7	-23.7	6	-33.1
	5	-25.2	4	-33.1	3	-25.2	2	-33.1
	1	-25.2						

6 - 7	9	----	8	-27.7	7	-23.7	6	-33.1
	5	-25.2	4	-33.1	3	-25.2	2	-33.1
	1	-25.2						
7 - 8	9	----	8	----	7	-10.8	6	----
	5	-10.6	4	----	3	-10.6	2	----
	1	-10.6						

TELAIO N. 4 SCHEMA DI CARICO N. 3

CARICHI VERTICALI

campata	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m
1 - 2	4	-39.1	3	-47.4	2	-47.4	1	-47.4
2 - 3	4	-39.1	3	-47.4	2	-47.4	1	-47.4

. . . . .

TELAIO N. 6 SCHEMA DI CARICO N. 3

CARICHI VERTICALI

campata	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m	piano	carico kN/m
15 - 10	8	----	7	-24.7	6	----	5	-27.5
	4	----	3	-27.5	2	----	1	-27.5
10 - 5	8	----	7	-24.7	6	----	5	-27.5
	4	----	3	-27.5	2	----	1	-27.5
5 - 1	8	-6.6	7	----	6	-7.8	5	----
	4	-7.8	3	----	2	-7.8	1	----

. . . . .

RISULTATI

COMPONENTI DI MOVIMENTO DEGLI IMPALCATI

impalcato	Vx (mm)	Vy (mm)	rotaz.x1000
9	0.000	-0.074	0.000
8	0.000	0.097	0.000
7	0.000	0.132	0.000

6	0.000	0.116	0.000
5	0.000	0.080	0.000
4	0.000	0.062	0.000
3	0.000	0.052	0.000
2	0.000	0.027	0.000
1	0.000	0.012	0.000

. . . . .

TELAIO N. 3

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
9	0.000
8	0.000
7	0.000
6	0.000
5	0.000
4	0.000
3	0.000
2	0.000
1	0.000

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
4	9	0.000	0.000	8	0.000	0.000
	7	0.000	-0.075	6	0.000	0.025
	5	0.000	-0.026	4	0.000	0.015
	3	0.000	-0.034	2	0.000	0.019
	1	0.000	-0.041			
5	9	0.000	0.000	8	0.000	-0.312
	7	0.000	0.004	6	0.000	-0.249
	5	0.000	-0.020	4	0.000	-0.246
	3	0.000	-0.020	2	0.000	-0.240
	1	0.000	-0.057			
6	9	0.000	0.000	8	0.000	-0.000
	7	0.000	-0.000	6	0.000	-0.000
	5	0.000	-0.000	4	0.000	-0.000
	3	0.000	-0.000	2	0.000	-0.000
	1	0.000	-0.000			
7	9	0.000	0.000	8	0.000	0.312
	7	0.000	-0.004	6	0.000	0.249
	5	0.000	0.020	4	0.000	0.246
	3	0.000	0.020	2	0.000	0.240
	1	0.000	0.057			

8	9	0.000	0.000	8	0.000	0.000
	7	0.000	0.075	6	0.000	-0.025
	5	0.000	0.026	4	0.000	-0.015
	3	0.000	0.034	2	0.000	-0.019
	1	0.000	0.041			

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
4 - 5	9	-----	-----	-----	-----
4 - 5	8	-----	-----	-----	-----
4 - 5	7	-17.96	-30.05	26.92	-31.40
4 - 5	6	-----	-----	-----	-----
4 - 5	5	-21.69	-29.49	27.18	-30.06
4 - 5	4	-----	-----	-----	-----
4 - 5	3	-20.76	-29.97	26.91	-30.33
4 - 5	2	-----	-----	-----	-----
4 - 5	1	-17.84	-34.58	25.52	-31.72
5 - 6	9	-----	-----	-----	-----
5 - 6	8	-31.78	-85.08	64.92	-84.66
5 - 6	7	-58.04	-57.37	64.11	-63.87
5 - 6	6	-52.00	-94.65	81.47	-97.27
5 - 6	5	-58.97	-62.37	67.41	-68.67
5 - 6	4	-52.44	-94.43	81.59	-97.15
5 - 6	3	-58.96	-62.37	67.41	-68.67
5 - 6	2	-53.11	-94.09	81.78	-96.96
5 - 6	1	-54.76	-64.48	66.24	-69.84
6 - 7	9	-----	-----	-----	-----
6 - 7	8	-85.08	-31.78	84.66	-64.92
6 - 7	7	-57.37	-58.04	63.87	-64.11
6 - 7	6	-94.65	-52.00	97.27	-81.47
6 - 7	5	-62.37	-58.97	68.67	-67.41
6 - 7	4	-94.43	-52.44	97.15	-81.59
6 - 7	3	-62.37	-58.96	68.67	-67.41
6 - 7	2	-94.09	-53.11	96.96	-81.78
6 - 7	1	-64.48	-54.76	69.84	-66.24
7 - 8	9	-----	-----	-----	-----
7 - 8	8	-----	-----	-----	-----
7 - 8	7	-30.05	-17.96	31.40	-26.92
7 - 8	6	-----	-----	-----	-----
7 - 8	5	-29.49	-21.69	30.06	-27.18
7 - 8	4	-----	-----	-----	-----
7 - 8	3	-29.97	-20.76	30.33	-26.91
7 - 8	2	-----	-----	-----	-----
7 - 8	1	-34.58	-17.84	31.72	-25.52

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
4	9	-----	-----	-----	-----
4	8	-----	-----	-----	-----
4	7	-17.96	-2.51	-10.30	-26.92
4	6	-2.51	12.94	-10.30	-26.92
4	5	-8.75	0.42	-6.12	-54.10
4	4	0.42	9.59	-6.12	-54.10
4	3	-11.16	0.37	-7.69	-81.01
4	2	0.37	11.90	-7.69	-81.01
4	1	-5.94	2.97	-2.10	-106.53
5	9	-----	-----	-----	-----
5	8	-31.78	15.59	-31.58	-64.92
5	7	-12.39	25.39	-25.19	-160.43
5	6	-26.61	14.84	-27.63	-241.91
5	5	-14.64	26.22	-27.24	-339.38
5	4	-26.22	14.65	-27.25	-420.97
5	3	-14.34	25.61	-26.64	-518.71
5	2	-27.50	18.12	-30.42	-600.49
5	1	-2.06	1.03	-0.73	-698.45
6	9	0.00	0.00	-0.00	0.00
6	8	-0.00	0.00	-0.00	-169.32
6	7	-0.00	0.00	-0.00	-297.05
6	6	0.00	-0.00	0.00	-491.59
6	5	-0.00	0.00	-0.00	-628.93
6	4	-0.00	0.00	-0.00	-823.22
6	3	0.00	-0.00	0.00	-960.57
6	2	-0.00	-0.00	-0.00	-1154.48
6	1	0.00	-0.00	0.00	-1294.16
7	9	-----	-----	-----	-----
7	8	31.78	-15.59	31.58	-64.92
7	7	12.39	-25.39	25.19	-160.43
7	6	26.61	-14.84	27.63	-241.91
7	5	14.64	-26.22	27.24	-339.38
7	4	26.22	-14.65	27.25	-420.97
7	3	14.34	-25.61	26.64	-518.71
7	2	27.50	-18.12	30.42	-600.49
7	1	2.06	-1.03	0.73	-698.45
8	9	-----	-----	-----	-----
8	8	-----	-----	-----	-----
8	7	17.96	2.51	10.30	-26.92
8	6	2.51	-12.94	10.30	-26.92
8	5	8.75	-0.42	6.12	-54.10
8	4	-0.42	-9.59	6.12	-54.10
8	3	11.16	-0.37	7.69	-81.01
8	2	-0.37	-11.90	7.69	-81.01
8	1	5.94	-2.97	2.10	-106.53

TELAIO N. 4

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
4	-0.000
3	-0.000
2	-0.000
1	-0.000

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
1	4	0.000	-0.236	3	0.000	-0.134
	2	0.000	-0.137	1	0.000	-0.221
2	4	0.000	0.000	3	0.000	0.000
	2	0.000	0.000	1	0.000	0.000
3	4	0.000	0.236	3	0.000	0.134
	2	0.000	0.137	1	0.000	0.221

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
1 - 2	4	-62.14	-103.64	94.73	-110.54
1 - 2	3	-93.19	-116.71	119.94	-128.91
1 - 2	2	-92.76	-116.93	119.82	-129.03
1 - 2	1	-83.00	-121.81	117.03	-131.82
2 - 3	4	-103.64	-62.14	110.54	-94.73
2 - 3	3	-116.71	-93.19	128.91	-119.94
2 - 3	2	-116.93	-92.76	129.03	-119.82
2 - 3	1	-121.81	-83.00	131.82	-117.03

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
1	4	-62.14	51.65	-37.93	-94.73
1	3	-41.54	41.92	-27.82	-214.68
1	2	-50.84	59.38	-36.74	-334.50
1	1	-23.62	11.81	-6.16	-451.53

2	4	0.00	-0.00	0.00	-221.09
2	3	-0.00	0.00	-0.00	-478.90
2	2	-0.00	0.00	-0.00	-736.95
2	1	-0.00	0.00	-0.00	-1000.59
3	4	62.14	-51.65	37.93	-94.73
3	3	41.54	-41.92	27.82	-214.68
3	2	50.84	-59.38	36.74	-334.50
3	1	23.62	-11.81	6.16	-451.53

. . . . . .

TELAIO N. 6

SPOSTAMENTO ORIZZONTALE DEI TRAVERSI

traverso	spostamento (mm)
8	0.097
7	0.132
6	0.116
5	0.080
4	0.062
3	0.052
2	0.027
1	0.012

SPOSTAMENTO VERTICALE E ROTAZIONE DEI NODI

pilastro	piano	vy (mm)	rotaz.x1000	piano	vy (mm)	rotaz.x1000
15	8	0.000	0.000	7	0.000	-0.199
	6	0.000	0.048	5	0.000	-0.098
	4	0.000	0.037	3	0.000	-0.108
	2	0.000	0.044	1	0.000	-0.148
10	8	0.000	0.000	7	0.000	0.001
	6	0.000	-0.023	5	0.000	-0.012
	4	0.000	-0.009	3	0.000	-0.008
	2	0.000	-0.016	1	0.000	-0.006
5	8	0.000	-0.069	7	0.000	0.118
	6	0.000	-0.101	5	0.000	0.099
	4	0.000	-0.087	3	0.000	0.107
	2	0.000	-0.103	1	0.000	0.135
1	8	0.000	0.130	7	0.000	-0.048
	6	0.000	0.101	5	0.000	-0.074
	4	0.000	0.089	3	0.000	-0.073
	2	0.000	0.132	1	0.000	0.016

CARATTERISTICHE DI SOLLECITAZIONE NELLE TRAVI

campata	piano	momento a sinistra (kNm)	momento a destra (kNm)	taglio a sinistra (kN)	taglio a destra (kN)
15 - 10	8	-----	-----	-----	-----
15 - 10	7	-45.47	-77.52	64.77	-76.02
15 - 10	6	-----	-----	-----	-----
15 - 10	5	-63.28	-81.01	75.26	-81.49
15 - 10	4	-----	-----	-----	-----
15 - 10	3	-62.39	-81.16	75.08	-81.67
15 - 10	2	-----	-----	-----	-----
15 - 10	1	-58.16	-83.05	74.01	-82.74
10 - 5	8	-----	-----	-----	-----
10 - 5	7	-73.31	-54.12	73.76	-67.03
10 - 5	6	-----	-----	-----	-----
10 - 5	5	-78.50	-64.44	80.84	-75.91
10 - 5	4	-----	-----	-----	-----
10 - 5	3	-79.34	-63.35	81.18	-75.57
10 - 5	2	-----	-----	-----	-----
10 - 5	1	-81.12	-60.23	82.04	-74.71
5 - 1	8	-18.41	-8.78	20.95	-17.66
5 - 1	7	-----	-----	-----	-----
5 - 1	6	-16.99	-16.89	22.83	-22.80
5 - 1	5	-----	-----	-----	-----
5 - 1	4	-17.77	-17.43	22.87	-22.76
5 - 1	3	-----	-----	-----	-----
5 - 1	2	-18.40	-13.72	23.61	-22.02
5 - 1	1	-----	-----	-----	-----

CARATTERISTICHE DI SOLLECITAZIONE NEI PILASTRI

pilastro	piano	momento sup. (kNm)	momento inf. (kNm)	taglio (kN)	sforzo normale (kN)
15	8	-----	-----	-----	-----
15	7	-45.47	-5.18	-26.86	-64.77
15	6	-5.18	35.11	-26.86	-64.77
15	5	-28.17	0.52	-19.12	-140.04
15	4	0.52	29.20	-19.12	-140.04
15	3	-33.19	2.08	-23.51	-215.12
15	2	2.08	37.34	-23.51	-215.12
15	1	-20.82	10.10	-7.28	-289.13
10	8	-----	-----	-----	-----
10	7	4.21	0.65	2.37	-149.78
10	6	0.65	-2.91	2.37	-149.78
10	5	-0.40	-0.19	-0.14	-312.11
10	4	-0.19	0.02	-0.14	-312.11
10	3	1.83	-0.14	1.31	-474.95
10	2	-0.14	-2.11	1.31	-474.95
10	1	-0.18	-0.22	0.01	-639.74

```
5     8     -18.41     -19.87      0.97     -20.95
5     7      34.26      10.52     15.82     -87.98
5     6      -6.47     -34.41     18.63    -110.81
5     5      30.03       8.11     14.61    -186.72
5     4      -9.67     -30.16     13.66    -209.60
5     3      33.20       9.83     15.58    -285.16
5     2      -8.57     -40.11     21.03    -308.78
5     1      20.11     -10.37      7.17    -383.49

1     8       8.78       0.37      5.60     -17.66
1     7       0.37      -8.04      5.60     -17.66
1     6       8.85       0.15      5.80     -40.46
1     5       0.15      -8.55      5.80     -40.46
1     4       8.88      -0.55      6.29     -63.21
1     3      -0.55      -9.99      6.29     -63.21
1     2       3.73       2.25      0.99     -85.23
1     1       2.25      -1.96      0.99     -85.23

. . . . .

Il calcolo e' stato eseguito :
- trascurando l'interazione tra aste di telai distinti che incidono
  in uno stesso punto nodale
- trascurando la  deformazione estensionale dei pilastri
  di telai a maglie rettangolari o trapezie
- trascurando la deformazione a taglio delle aste

Attenzione:
- l'effettivo sforzo normale in ciascun pilastro dello schema spaziale
  deve essere valutato  come somma  degli sforzi normali dei telai cui
  esso appartiene
```

I risultati ottenuti per le singole condizioni di carico sono stati combinati nel modo più gravoso per verificare la correttezza del dimensionamento strutturale. Ad esempio, per il primo estremo della trave 4–5 del telaio 3 al primo impalcato si ha un momento di 78.67 kNm per forze in direzione x e −17.84 kNm per carichi verticali. Il valori limite per il momento flettente in tale sezione sono quindi −96.51 kNm e +60.83 kNm.

Nel verificare i pilastri si è ovviamente tenuto conto del fatto che ciascuno di essi appartiene contemporaneamente a due telai. Ad esempio, il pilastro 1 appartiene al telaio 4 (parallelo all'asse x) e al telaio 6 (parallelo ad y). Utilizzando i pedici x ed y per indicare le aliquote determinate rispettivamente per il primo ed il secondo telaio, per la sezione al piede del primo ordine si ha:

- per forze in direzione x:

$$M_x = -92.01 \text{ kNm} \qquad M_y = -0.37 \text{ kNm}$$
$$N = N_x + N_y = 58.76 + 0.40 = 59.16 \text{ kN}$$

– per forze in direzione y:

$M_x = 0 \qquad M_y = -26.33$ kNm

$N = N_x + N_y = 0 - 50.64 = -50.64$ kN

– per carichi verticali:

$M_x = 11.81$ kNm $\qquad M_y = -1.96$ kNm

$N = N_x + N_y +$ peso proprio pilastro $= -451.53 - 85.23 - 66.38 = -603.14$ kN

L'effetto combinato di carichi verticali e sisma in direzione x porta ai valori limite:

$M_x = -80.20$ kNm	$M_y = -2.33$ kNm	$N = -543.98$ kN
$M_x = 103.82$ kNm	$M_y = -0.59$ kNm	$N = -662.30$ kN

La netta differenza tra M_x ed M_y consente di verificare a pressoflessione retta la sezione, che ha base 30 cm e altezza 60 cm e richiede quindi una armatura sul lato corto pari ad almeno 5 cm^2.

In maniera analoga si è proceduto per l'altra direzione e per tutte le sezioni degli elementi strutturali. Si sono riscontrate sollecitazioni elevate in alcuni pilastri del primo ordine, dovute principalmente all'elevato sforzo normale; si è ritenuto opportuno incrementare per essi il lato minore della sezione, ripetendo quindi la risoluzione dello schema spaziale.

pag.	riga	ERRATA	CORRIGE
11	28	Rita De Simone	Rita Simone
36	eq.(2.9)	$\frac{dy}{dx}$	$\frac{du}{dx}$
69	eq.(2.62)	invertire F_x con F_y	
72	eq.(2.66) in basso	$-\frac{EI\alpha}{H}$	$\frac{EI\alpha}{H}$
221	31	`Sl(6,1)=Sl(3,1)`	`Sl(6,1)=-Sl(3,1)`
222	da 7 a 12	invertire `Fx` con `Fy`	
333	23	`PRINT "  ";`	`PRINT#2, "  ";`
254	1 e 12	inserire `DefTag` come parametro di scambio nelle procedure `CarSoll` e `MatAstaGlobale`	
258	13	inserire `DefTag` come parametro di scambio nella procedura `MatAstaGlobale`	
259	34	inserire `DefTag` come parametro di scambio nella procedura `CarSoll`	
262	9	inserire `DefTag` come parametro di scambio nella procedura `CarSoll`	
268	42	inserire `DefTag` come parametro di scambio nella procedura `MatAstaGlobale`	

Ai programmi sono state apportate ulteriori modifiche, qui non indicate. In particolare, a partire dalla versione 1.1 è possibile utilizzare come unità di misura il sistema tecnico. A partire dalla versione 2.0 è inoltre possibile effettuare un'analisi modale dello schema. Maggiori informazioni sulle modifiche apportate e su come sfruttare queste nuove potenzialità dei programmi sono riportate nel dischetto.

BIBLIOGRAFIA

1. A. S. Hall, R. W. Woodhead, *Frame analysis*, John Wiley & sons, New York, 1961.
2. V. Franciosi, *Scienza delle costruzioni*, Liguori, Napoli, 1965.
3. M. Cugiani, *Metodi dell'analisi numerica*, UTET, Torino, 1967.
4. J. S. Przemieniecki, *Theory of matrix structural analysis*, McGraw–Hill, New York, 1968.
5. J. L. Meek, *Matrix structural analysis*, McGraw–Hill Kogakusha Ltd., Tokio, 1971.
6. K. J. Bathe, E. L. Wilson, *Numerical methods in finite element analysis*, Prentice–Hall, Englewood Cliffs, 1976.
7. G. Strang, *Linear algebra and its applications*, second edition, Academic Press, Orlando, 1976.
8. M. Capurso, *Introduzione al calcolo automatico delle strutture*, Cremonese, Roma, 1977.
9. D. Clarke, *Computer aided structural design*, John Wiley & sons, New York, 1978.
10. M. Como, G. Lanni, *Elementi di costruzioni antisismiche*, Cremonese, Roma, 1979.
11. M. Pagano, *Teoria degli edifici, 3° vol. 1° tomo. Telai per edifici a maglie rettangolari*, Liguori, Napoli, 1979.
12. B. P. Demidovic, I. A. Maron, *Fondamenti di calcolo numerico*, Mir, Mosca, 1981.
13. G. Toniolo, *Analisi strutturale con l'elaboratore elettronico*, Masson, Milano, 1982.
14. V. Franciosi, *Il calcolo automatico degli insiemi piani di travi*, Liguori, Napoli, 1983.
15. M. Capurso, *Edifici soggetti a forze orizzontali: calcolo automatico*, Cremonese, Roma, 1984.

16. A. Ghersi, *Il personal computer nel calcolo di edifici. Introduzione alla programmazione*, 2^a ediz., CUEN, Napoli, 1987.

17. *Learning and using Microsoft Quick Basic*, Microsoft, 1987.

18. *Microsoft Quick Basic: basic language reference*, Microsoft, 1987.

19. A. Ghersi, *Edifici antisismici con struttura intelaiata in cemento armato*, 4^a ediz., CUEN, Napoli, 1988.

20. A. Ghersi, P. Lenza, *Teoria degli edifici, 3^o vol. 3^o tomo. Telai spaziali per edifici regolari a piani rigidi*, Liguori, Napoli, 1988.

21. G. Ghinassi, *Analisi matriciale e calcolo automatico delle strutture*, 2^a ediz., Esa, Milano, 1988.

22. R. Coraggio, A. Ghersi, *Teoria degli edifici, appendice al vol.3^o, tomo 1^o e 2^o. Telai per edifici: calcolo automatico*, Liguori, Napoli, 1989.

Nella stessa collana:

A. GHERSI, *Introduzione alla programmazione. Il personal computer nel calcolo di edifici*, pp. 248, lire 30.000

A. GHERSI, *Edifici antisismici con struttura intelaiata in cemento armato*, pp. 444, lire 48.000

M. COMO, P. MALANGONE, *Elementi di calcolo matriciale e automatico delle strutture*, pp. 250, lire 19.000

M. DE MATTEO, *Sul calcolo delle strutture piane a maglia di forma qualsiasi*, pp. 168, lire 12.500

M. DE MATTEO - A. MODUGNO, *Sul calcolo delle strutture spaziali a maglie di forma qualsiasi*, pp. 240, lire 35.000

A. GILIBERTI, *Una più razionale struttura della scala per edifici in zona sismica*, pp. 64, lire 10.000

L. NUNZIANTE - R. OCONE, *The limit design of frames subjected to sismic loads*, pp. 352, lire 27.000

In preparazione:

A. GHERSI, *Verifica e progetto di sezioni in c.a. Il personal computer nel calcolo di edifici*

A. GHERSI, *Telai piani e spaziali a maglie regolari. Il personal computer nel calcolo di edifici*

finito di stampare nel mese di aprile 1995
da La Buona Stampa (Ercolano) per conto della CUEN s.r.l.