

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 antepo-
nendo 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 $\bar{\mathbf{S}}$ 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 `Nu-mera` 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 `Nu-mera` 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 L_d) e la matrice di rigidezza fondamentale 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 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 `KT()`, `KTA()` e `KA()`, 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 `Triangularizza`).

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 $\bar{\mathbf{R}}$ 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 \bar{S}_i , azioni di incastro perfetto nel riferimento locale, operando in maniera differenziata a seconda del tipo di asta ed utilizzando in alternativa le procedure `S1.Inc.Qunif` e `S1.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 `CarSol1`. Per ogni asta determina innanzitutto la matrice 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
Wnk1	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
- CsA1 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 ($N\text{ cm}^{-3}$)
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
SnA1	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 $\bar{\mathbf{R}}$, 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


```

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!, Ium!, SnaI!, CsaI!, Ld!(), Kg!())
DECLARE SUB OutputIntestaz (FI$, Intest$)
DECLARE SUB OutputModElast (DefTag!, EL!, GL!)
DECLARE SUB TermNota (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 PROGGEN.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.Gherzi, 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$(), VCed, KV())
CALL OutputAsteGen(DefTag, Wnk1, JZ, NT(), TA$(), 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 ML <> 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, Sna1, Csa1,
                    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

' ----- TerminNoti -----
' Routine per il calcolo dei carichi nodali
' -----
TerminNoti:
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, O, 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!, SnA1!, CsA1!, Ld!( ), Kg!( ))
DECLARE SUB Geometria (X1!, X2!, Y1!, Y2!, L!, SnA1!, CsA1!, 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 AzInLocale (PX!, PY!, TA$, B!, K!, Lum!, SnAl!, CsAl!, Ld!(), LdT!(),
                        Kg!(), S!())
DECLARE SUB Inserimento (P!, Q!, V!, NL!, SB!, KTT!(), KTA!(), KAA!())
DECLARE SUB LCOPY (SCR!, DENS!)
DECLARE SUB MatAstaGlobale (TA$, X1!, X2!, Y1!, Y2!, A!, AT!, Iner!, K!, B!, E!, G!,
                           Lum!, SnAl!, CsAl!, Ld!(), Kg!())
DECLARE SUB ModGraf (Scheda!)
DECLARE SUB Scala (X1!, Y1!, X2!, Y2!)

```

```
DIM SHARED Schedal
```

```

ModGrafErr:      ' individuazione della modalita' grafica
SELECT CASE Schedal
CASE 12
  Schedal = 11   ' MCGA
CASE 11
  Schedal = 8    ' EGA 64k
CASE 8
  Schedal = 2    ' CGA
CASE 2
  Schedal = 3    ' Hercules
CASE ELSE
  Schedal = 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
```

```

' ===== AzInLocale =====
' 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 AzInLocale (PX, PY, TA$, B, K, Lum, 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 Sl.Inc.Qunif(PQ, PN, Lum, S())
CASE "W"
  CALL ProdottoMat(LdT(), Kg(), K1())
  CALL ProdottoMat(K1(), Ld(), K1())
  CALL Sl.Wnk.Qunif(PQ, PN, Lum, 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

```

* ===== DefinisceMD =====
*   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 DefinisceMD (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.Gherzi, 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
  NEXT K
  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)
    END IF

```

```

      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

```

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, Lum, SnAl, CsAl,
  Ld(), Kg())

  SELECT CASE TA$
    CASE "T"
      IF DefTag = 1 THEN
        CALL Geometria(X1, X2, Y1, Y2, Lum, SnAl, CsAl, Ld())
        CHI = A / AT
        CALL Kg.IncTag(E, G, CHI, A, Iner, Lum, Ld(), Kg())
      ELSE
        CALL Asta.Inc(X1, X2, Y1, Y2, E, A, Iner, Lum, SnAl, CsAl, Ld(), Kg())
      END IF
    CASE "W"
      CALL Geometria(X1, X2, Y1, Y2, Lum, SnAl, CsAl, Ld())
      CALL Kg.Wnk(K, B, E, A, Iner, Lum, Ld(), Kg())
  END SELECT

```

```
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)
```

```

    Schedal = 12      ' VGA
    ON ERROR GOTO ModGrafErr
    SCREEN Schedal
    ON ERROR GOTO 0
    Scheda = Schedal

```

```
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/mm2"
    IF DefTag = 1 THEN
        PRINT #2, "Modulo di elasticita' tangenziale = "; GL; " N/mm2"
    END IF

```

```
END SUB
```

```

===== Scala =====
'
'   Procedura per la definizione della scala per rappresentazione monometrica
'   (vedi: A.Gherzi, 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, KIT(), 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$( ), Wnk1, B(), K(), ER)

  ' controlla ammissibilita' dell'identificatore del tipo di asta
  Wnk1 = 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
        Wnk1 = 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 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      m2 ";
  IF DefTag = 1 THEN PRINT #2, "  m2 ";

```

```

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

```

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, "trasverso";
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 "      equilibrio = #####.## 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 specifici 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 interpretata 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 esplicitate 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.

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

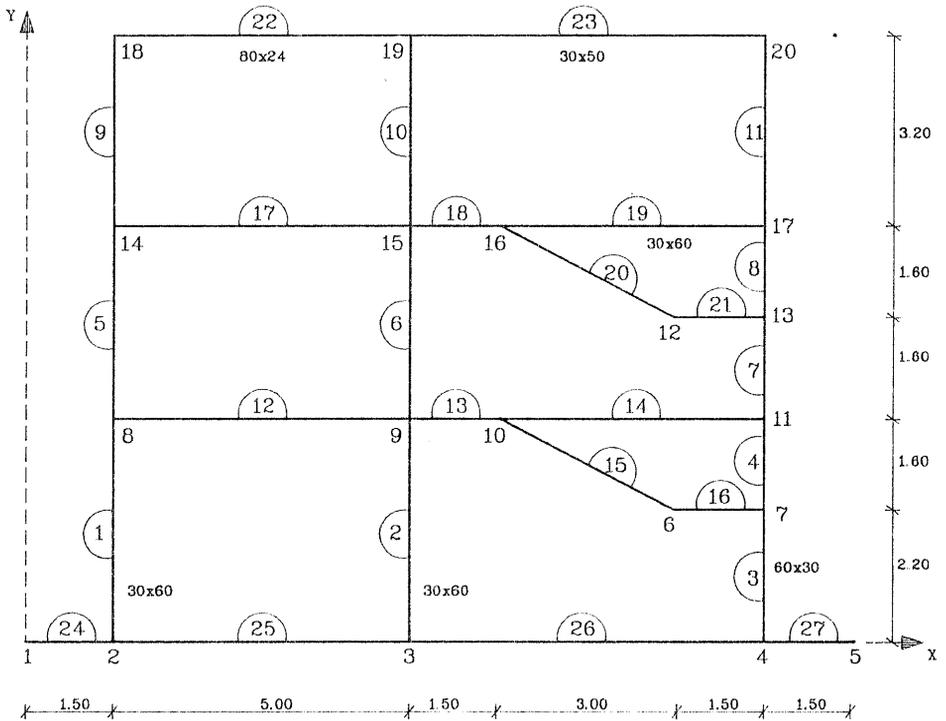


Fig. 5.1 — Schema di telaio con trave a ginocchio e trave di fondazione

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/mm²

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)						
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 strutture 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'individuazione 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 ritzi 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 ritte 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 ritte). 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 ritte 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'ineestensibilità. Si suppone inoltre che lo schema sia completo, cioè senza aste mancanti.

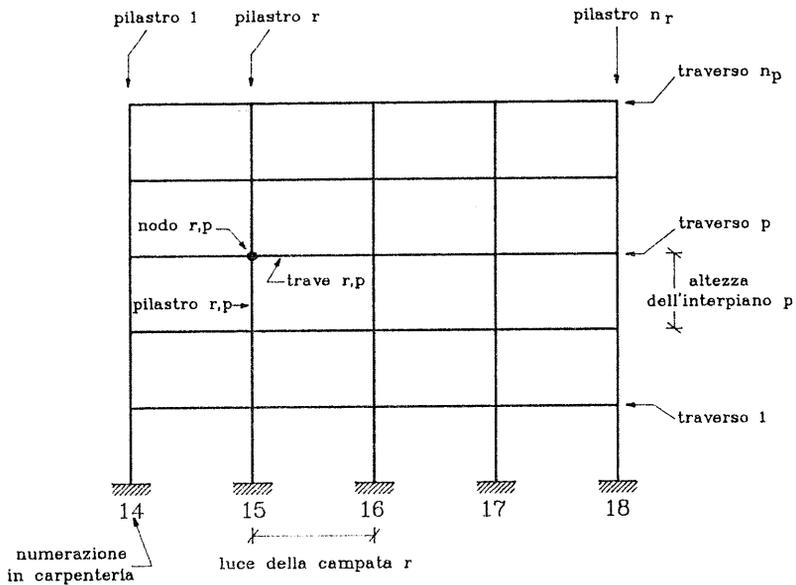


Fig. 6.1 — Telaio a maglie rettangolari

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

$$j = (p - 1)(n_r - 1) + r \tag{6.2}$$

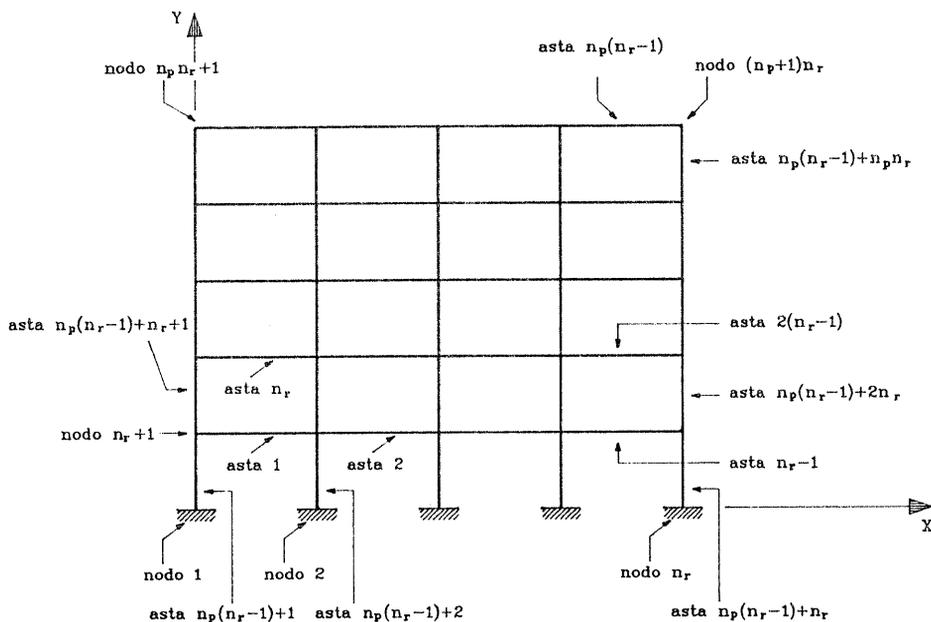


Fig. 6.2 — Telaio a maglie rettangolari: numerazione dei nodi e delle aste

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

$$j = n_p(n_r - 1) + (p - 1)n_r + r \quad (6.3)$$

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 ritzi 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 ritzi sono inestensibili ed in caso contrario vicina al vero quando l'inclinazione è modesta.

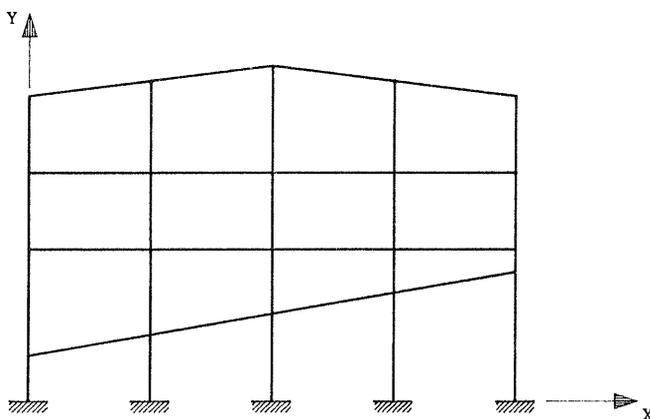


Fig. 6.3 — Telaio a maglie trapezie

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

$$\begin{aligned}
 i &= r + 1 && \text{per } p = 0 \\
 i &= p n_r + r + 2 && \text{per } p > 0
 \end{aligned}
 \tag{6.4}$$

La corrispondenza con l'indice j richiesto per le aste del telaio generico

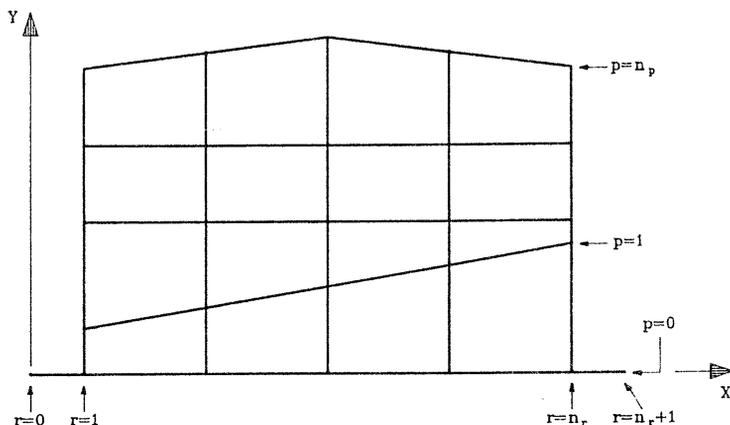


Fig. 6.4 — Telaio a maglie trapezie con trave di fondazione

è fornita dalle relazioni

$$j = r + 1$$

$$j = (p - 1)(n_r - 1) + r + n_r + 1 \quad (6.5)$$

$$j = n_p(n_r - 1) + p n_r + r + 1$$

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 rigidità 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 .

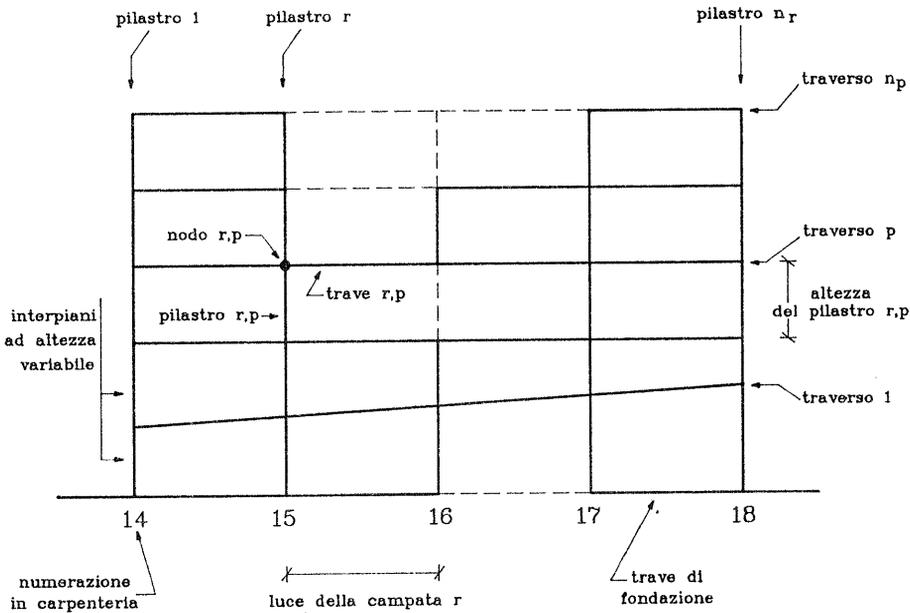


Fig. 6.5 — Telaio a maglie trapeziche con trave di fondazione ed aste mancanti

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.

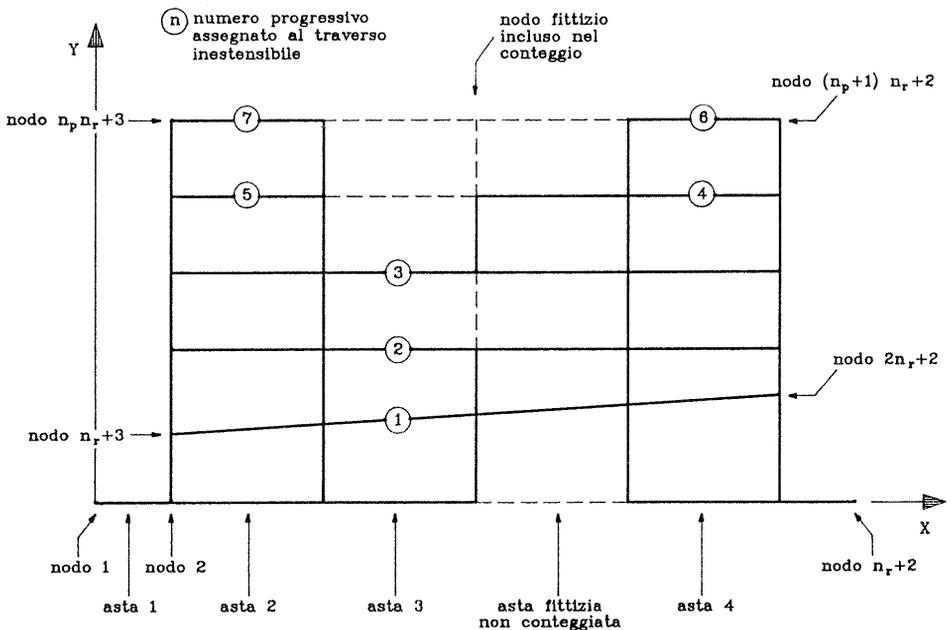


Fig. 6.6 — Telaio a maglie trapezie: numerazione dei nodi, delle aste e dei traversi inestensibili

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'assenza 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'inesistibilità 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 informazioni 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 *TELGEM*. 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. Procedo 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 *ConvertCarRet* 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 ciascun 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 `TELGEM`. 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)
Wnk1	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 ritto di differente altezza, si assegna $\text{HI}(P)=0$
- 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 ($N\text{ cm}^{-3}$)
KV()	array bidimensionale che contiene le rigidzze 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


```

                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!(), HI!())
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(O TO RZ), MNF(O TO RZ + 1), FYF(O TO RZ + 1)
    DIM MFD(O TO RZ), MFS(O TO RZ), TFD(O TO RZ), TFS(O 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(O TO RZ)
  DIM NB(RZ - 1, PZ), NR(RZ, PZ)
  IF Wnkl = 1 THEN DIM NF(O TO RZ), BF(O TO RZ), KF(O TO RZ)
RETURN

```

```

* ..... DimMatRig .....
* Routine per il dimensionamento di matrice di rigidezza e vettore carico
*

```

```

DimMatRig:
  IF NL = 0 THEN
    REDIM KTT(O TO O, O TO O), KTA(O TO O, 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.Ghersì, 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 ConvertGeomRet(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, Sna1, Csa1, 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, Wnk1, 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, Wnk1, 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 ConvertCarRet(RZ, PZ, Wnk1, 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 TermNota(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, O, 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 ConverterRisRet(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 TELREF, utilizzate anche nel programma TELSPA
'

```

```

' ----- 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
        NEXT R
    NEXT P
END SUB

```

```

      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 Wnk1 = 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 Wnk1 = 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, Wnk1, 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 Wnk1 = 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

```

```

' ===== ConvertereRisRet =====
' Procedura per la valutazione degli spostamenti dei traversi
' e la conversione delle caratteristiche di sollecitazione
' -----
SUB ConvertereRisRet (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 trasversi
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 Wnk1 = 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, Wnk1, 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
' ritto
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 Wnk1 = 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
    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$(I) <> "111" THEN
                INPUT #1, MN(R, P)
                IF Contr = 1 THEN
                    PRINT #2, "pilastro"; NP(R), "piano"; P, "M ="; MN(R, P)
                END IF
            END IF
        END IF
    NEXT R
NEXT P
IF Wnk1 = 1 THEN
    FOR F = 0 TO RZ + 1
        I = F + 1
        IF V$(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
    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
        END IF
    NEXT R

```

```

NEXT P
IF Wnk1 = 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, Wnk1, RZ, DL())

```

  IF Wnk1 = 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, Wnk1, 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 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 #3, "           ";
            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 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(), TFSD(), 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 Wnk1 = 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

```

RisPilastr:

```

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) = "0" 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 ritzi 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 ritzi 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. Convieni 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 ($N\text{ mm}^{-2}$)
- 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 assegna 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
```

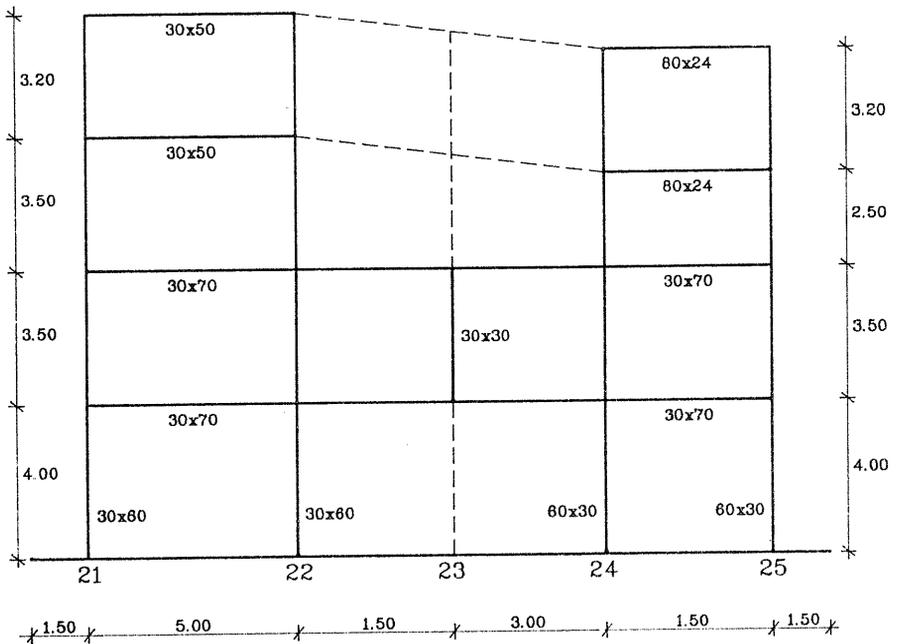


Fig. 6.7 — Schema di telaio con piani superiori divisi in due corpi distinti

1,0,0,1
 -25,-25
 -30,-30
 -30,-25,-25,-30
 -38,-38,-38,-38
 0,0,0,0,0
 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 25	3.50 2.50	22	3.50	23	3.00	24	2.50

LUCI E SEZIONI DELLE TRAVI

campata	luce m	piano	sez.	b m	h m	inerzia m ⁴	piano	sez.	b m	h m	inerzia m ⁴
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 m ⁴	piano	sez.	b m	h m	inerzia m ⁴
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 m ⁴	B m	K N/cm ³
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

sezione	forma	b		h	area	inerzia
		m	m	m	m ²	m ⁴
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	236.11	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 rigidità 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 ConvertGeometria 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 `ConvertCarichi` 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 mediante 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
- TelWnk1 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
- Wnk1 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

- A1 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
- CsA1 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
- SnA1 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 (KIT!(), KTA!(), KAA!(), SB!, RID!)
DECLARE SUB Risolve (A!(), B!(), IPVT!())
DECLARE SUB Risolve.K2 (KIT!(), KTA!(), KAA!(), SB!, RID!, B!())

```

* procedure del file PROCCM.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!, KIT!(), 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!,
    Lum!, Sna1!, Csa1!, Ld!(), Kg!())
DECLARE SUB OutputIntestaz (FI$, Intest$)
DECLARE SUB OutputModElast (DefTag!, EL!, GL!)
DECLARE SUB TermNota (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!, KIT!(), KTA!(), KAA!())

```

* procedure del file PROGEN.BAS

```

DECLARE SUB InputAsteGen (Contr!, JZ!, LZ!, E1!(), E2!(), NT!(), TA$((), Wnk1!, B!(),
    K!(), ER!)
DECLARE SUB InputCarGen (Contr!, IZ!, JZ!, QN!, QA!, F!(), P!(), ER!)
DECLARE SUB InputNodiGen (Contr!, IZ!, X!(), Y!(), V$((), VCed!, KV!(), ER!)
DECLARE SUB InputTraversiGen (Contr!, KZ!, NI!())
DECLARE SUB OutputAsteGen (DefTag!, Wnk1!, 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 ConvertCarRet (RZ!, PZ!, Wnk1!, NB!(), NF!(), ZA!(), ZI!(), IV!, IM!,
    IN!, IO!, Q!(), QF!(), MN!(), MNF!(), FYN!(), FYF!(),
    FX!(), KZ!, V$( ), NI!(), P!(), F!())
DECLARE SUB ConvertGeomRet (PZ!, RZ!, Wnk1!, DefEst!, DL!(), H!(), NB!(), NR!(),
    NF!(), BF!(), KF!(), ZA!(), ZI!(), IZ!, JZ!, KZ!, X!(),
    Y!(), V$( ), NI!(), E1!(), E2!(), TA$( ), NT!(), B!(), K!())
DECLARE SUB ConvertRisRet (RZ!, PZ!, Wnk1!, DefEst!, NB!(), NR!(), NF!(), ZA!(),
    ZI!(), D!(), MS!(), MD!(), TS!(), TD!(), MP!(), MT!(),
    TR!(), SN!(), MFS!(), MFD!(), TFS!(), TFD!(), KZ!, NI!(),
    N!(), R!(), F!(), M1!(), M2!(), T1!(), T2!(), N1!())
DECLARE SUB InputAltezze (Contr!, PZ!, RZ!, AltVar!, HI!(), H!())
DECLARE SUB InputAsteRet (Contr!, Wnk1!, PZ!, RZ!, LZ!, NP!(), NB!(), NR!(), NF!(),
    BF!(), KF!(), ER!)
DECLARE SUB InputCarRet (Contr!, Wnk1!, IV!, IM!, IN!, IO!, PZ!, RZ!, KZ!, NP!(),
    V$( ), NB!(), NF!(), ZA!(), ZI!(), Q!(), QF!(), MN!(), MNF!(),
    FYN!(), FYF!(), FX!())
DECLARE SUB InputLuci (Contr!, Wnk1!, RZ!, DL!())
DECLARE SUB InputNumRitti (Contr!, RZ!, NP!())
DECLARE SUB OutputAsteRet (PZ!, RZ!, Wnk1!, NP!(), DL!(), NB!(), NR!(), NF!(), BF!(),
    KF!(), FS$( ), ZB!(), ZH!(), ZI!())
DECLARE SUB OutputCarRet (CAR!, Wnk1!, IV!, IM!, IN!, IO!, PZ!, RZ!, KZ!, NP!(),
    V$( ), 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!, Wnk1!, 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
DIM Ld(6, 6), Kg(6, 6)

```

' definisce l'estremo inferiore dell'indice degli array
' 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$ + "GEDM.SPA"
FOR T = 1 TO TZ
  GOSUB LeggeGeomTel
  GOSUB ConvertteGeometria
  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$ + "GEDM.SPA"
    OPEN "O", #5, FT$ + "CAR.SPA"
    FOR T = 1 TO TZ
      GOSUB ReadGeomTel
      GOSUB LeggeCarTel
      GOSUB ConvertteCarichi
      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$ + "GEDM.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 ConvertteRisult
    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 Wnk1 = 1 THEN REDIM QF(O TO RZ), MNF(O TO RZ + 1), FYF(O 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(O TO RZ)
  REDIM NB(RZ - 1, PZ), NR(RZ, PZ)
  IF Wnk1 = 1 THEN REDIM NF(O TO RZ), BF(O TO RZ), KF(O 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 Wnki = 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.Gherzi, 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 OutputPosTelaio
  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

```

```

' ..... OutputPosTelaio .....
' Routine per la stampa della posizione dei telai piani che compongono
' l'insieme spaziale
'

```

```

OutputPosTelaio:
  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$( ), Wnk1, 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, Wnk1, RZ, DL())
CALL InputAsteRet(Contr, Wnk1, 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 Wnk1 = 1 THEN TelWnk1 = 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, Wnk1, 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, Wnk1, 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 trasversi)";
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, Sna1, Csa1, 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, " xO ="; Xm(PS), "yO ="; 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, Wnk1, 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

```

```

' ----- TerminNoti -----
' 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) * CsA1
  Rs(PI(K) + PSZ) = Rs(PI(K) + PSZ) + Rt(K) * SnA1
  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

```

```

' ----- ConverterResult -----
' Routine per la conversione delle caratteristiche di sollecitazione degli
' schemi piani particolari
' -----

```

```

ConverterResult:
IF TT$(T) = "G" THEN RETURN
PRINT "conversione caratt. sollecitaz.": GOSUB InCorso

SELECT CASE TT$(T)
CASE "R"
  GOSUB DimRisRet

```

```

CALL ConvertRisRet(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 rigidità
' 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, Wnk1
CALL WriteMat(NP(), 4)
CALL WriteMat(DL(), 4)
CALL WriteMat(NB(), 4)
CALL WriteMat(NR(), 4)
IF Wnk1 = 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, Wnk1
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 Wnk1 = 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(KTI(), 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 ritzi 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 ritzi, 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. Convieni 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 ritzi non hanno tutti la stessa altezza deve essere assegnato ad essa il valore 0
- per ciascun interpiano i cui ritzi 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 impalcato 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 impalcato 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 impalcato ($0=\text{no}$, $1=\text{si}$)
- indice di esistenza di carichi applicati ai telai ($0=\text{no}$, $1=\text{si}$)
- informazioni che descrivono i carichi applicati agli impalcato (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 impalcato 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 (kNm^{-1})
 - $p_{y'}$, componente del carico distribuito nella direzione y' del riferimento locale del telaio, valutata per unità di lunghezza dell'asta (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 (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 interpretata 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

applicare 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 taglienti 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.