

Dispense del corso di
Metodi Numerici per le Equazioni Differenziali

Progetto numerico al calcolatore - Parte I
Topologia e assemblaggio della matrice di rigidità

Mario Putti
Dipartimento di Matematica – Università di Padova

16 marzo 2015

Indice

1	Memorizzazione compatta di matrici sparse	1
1.1	Implementazione del prodotto matrice-vettore	3
2	Costruzione della topologia della matrice di rigidità	5
3	Assemblaggio della matrice di rigidità dai contributi locali	11
3.1	La matrice di puntatori TRIJA	12

1 Memorizzazione compatta di matrici sparse

Esistono molti modi per memorizzare in maniera efficiente una matrice sparsa. Il modo descritto qui è detto CRS (Compressed Row Storage). Vediamo come si procede nel caso di matrice simmetrica e nel caso di matrice non simmetrica.

Caso non simmetrico. Data una generica matrice A quadrata di ordine n , la tecnica di memorizzazione CRS prevede la generazione di 3 vettori:

1. **SYSMAT**: vettore di numeri reali contenente gli nt coefficienti non nulli della matrice A memorizzati in successione per righe;
2. **JA**: vettore di numeri interi contenente gli nt indici di colonna dei corrispondenti elementi memorizzati in **SYSMAT**;
3. **IA**: vettore di numeri interi con $n + 1$ componenti, contenente le posizioni in cui si trova in **SYSMAT** il primo elemento di ciascuna riga di A .

Il vettore **IA** è chiamato “vettore topologico” e talvolta indicato anche come **TOPOL**. L’uso congiunto di **IA** e **JA** consente di individuare qualsiasi elemento non nullo a_{ij} memorizzato in **SYSMAT**. Infatti, l’elemento a_{ij} si troverà in una posizione k del vettore **SYSMAT** compresa nell’intervallo $\text{IA}(i) \leq k \leq \text{IA}(i + 1) - 1$ e tale per cui $\text{JA}(k) = j$. Queste due condizioni permettono di individuare univocamente k per cui $\text{SYSMAT}(k) = a_{ij}$. Si noti che l’occupazione di memoria si riduce da n^2 numeri reali (generalmente in doppia precisione) a nt numeri reali (tipicamente $nt < 0.05n^2$) e $nt + n + 1$ numeri interi.

Facciamo un esempio numerico per rendere più chiaro il concetto. Si consideri la seguente matrice A di dimensione 6×6 , sparsa e non simmetrica:

$$A = \begin{bmatrix} 4 & 0 & 7 & 0 & 0 & 0 \\ 3 & 4 & 0 & 2 & 0 & 0 \\ 0 & 2 & 4 & 0 & 4 & 0 \\ 0 & 0 & 7 & 4 & 0 & 1 \\ 1 & 0 & 0 & 3 & 4 & 0 \\ 1 & 1 & 0 & 0 & 3 & 4 \end{bmatrix}.$$

Gli elementi non nulli sono $nt = 18$, mentre $n^2 = 36$, pertanto il grado di sparsità di A è il 50%. Trascurando i coefficienti nulli, la matrice A diventa:

$$A' = \begin{bmatrix} 4 & & 7 & & & \\ 3 & 4 & & 2 & & \\ & 2 & 4 & & 4 & \\ & & 7 & 4 & & 1 \\ 1 & & & 3 & 4 & \\ 1 & 1 & & & 3 & 4 \end{bmatrix}.$$

Il vettore **SYSMAT** ha dimensione 18 e risulta:

$$\overbrace{4 \ 7 \ 3 \ 4 \ 2 \ 2 \ 4 \ 4 \ 7 \ 4 \ 1 \ 1 \ 3 \ 4 \ 1 \ 1 \ 3 \ 4}^{nt} .$$

Il vettore **JA** ha la stessa dimensione di **SYSMAT** e per ogni coefficiente di quest'ultimo individua l'indice di colonna:

$$\overbrace{1 \ 3 \ 1 \ 2 \ 4 \ 2 \ 3 \ 5 \ 3 \ 4 \ 6 \ 1 \ 4 \ 5 \ 1 \ 2 \ 5 \ 6}^{nt} .$$

Il vettore **IA**, infine, individua la posizione in **SYSMAT** del primo coefficiente di ciascuna riga:

$$\overbrace{1 \ 3 \ 6 \ 9 \ 12 \ 15 \ 19}^{n+1} .$$

Si noti che che le componenti di **IA** sono necessariamente ordinate in senso strettamente crescente e che $\text{IA}(n+1) = nt + 1$. Ciò si deduce immediatamente dal fatto che per individuare un elemento a_{ij} dell'ultima riga di A si deve cercare l'indice k della componente in **SYSMAT** nell'intervallo $\text{IA}(n) \leq k \leq \text{IA}(n+1) - 1$.

Caso simmetrico Nel caso in cui la matrice A sia simmetrica, come accade ad esempio nella soluzione agli elementi finiti dell'equazione differenziale ellittica di Laplace, il risparmio computazionale derivato dal sistema CRS viene incrementato memorizzando la sola parte triangolare alta, inclusa la diagonale principale, di A . In altri termini, vengono memorizzati solo i coefficienti $a_{ij} \neq 0$ con $j \geq i$ e si sfrutta la proprietà A secondo cui $a_{ij} = a_{ji}$.

La memorizzazione di A viene effettuata sempre mediante i tre vettori **SYSMAT**, **JA** e **IA** definiti nel paragrafo precedente, in cui tuttavia nt rappresenta il numero di coefficienti non nulli della triangolare alta e **IA** contiene le posizioni in cui si trova in **SYSMAT** l'elemento diagonale di ciascuna riga di A .

Anche in questo caso facciamo un esempio numerico per rendere più chiaro il concetto. Si consideri la seguente matrice A di dimensione 7×7 , sparsa e simmetrica:

$$A = \begin{bmatrix} 10 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 & 0 & -1 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 20 \end{bmatrix} .$$

Gli elementi non nulli sono 19, mentre se fosse piena ne avremmo 49. Tuttavia i coefficienti da memorizzare sono solamente quelli relativi alla parte superiore di A , di seguito evidenziati:

$$A' = \begin{bmatrix} \boxed{10} & & & & \boxed{1} & & & \boxed{2} \\ & \boxed{1} & & \boxed{3} & & \boxed{-1} & & \\ & & \boxed{4} & & & & & \boxed{1} \\ 1 & & 3 & \boxed{2} & \boxed{1} & & & \\ & & -1 & 1 & \boxed{5} & & & \\ 2 & & & & & \boxed{1} & & \boxed{20} \end{bmatrix} .$$

Il vettore **SYSMAT** avrà perciò dimensione $nt = 13$ (7 elementi diagonali più 6 extra-diagonali) anziché 19, con un risparmio di memoria superiore al 30%, e sarà dato da:

$$\overbrace{10 \ 1 \ 2 \ 1 \ 3 \ -1 \ 4 \ 1 \ 2 \ 1 \ 5 \ 1 \ 20}^{nt} .$$

Il vettore **JA** con gli indici di colonna corrispondenti ha la stessa dimensione di **SYSMAT** e risulta:

$$\overbrace{1 \ 5 \ 7 \ 2 \ 4 \ 6 \ 3 \ 7 \ 4 \ 5 \ 5 \ 6 \ 7}^{nt} .$$

Infine, il vettore **IA** possiede sempre dimensione $n + 1$ ed individua la posizione in **SYSMAT** degli elementi diagonali:

$$\overbrace{1 \ 4 \ 7 \ 9 \ 11 \ 12 \ 13 \ 14}^{n+1} .$$

Ad esempio, $\text{IA}(3)=7$ significa che a_{33} è memorizzato in **SYSMAT**(7), come confermato anche dal fatto che $\text{JA}(7)=3$.

Come nel caso di matrici non simmetriche, si pone $\text{IA}(n + 1) = nt + 1$. Si noti anche che dovrà sempre essere $\text{IA}(1)=1$ e $\text{IA}(n) = nt$, nonché $\text{JA}(1)=1$ e $\text{JA}(nt) = n$.

1.1 Implementazione del prodotto matrice-vettore

Lo schema del GCM necessita dell'operazione del prodotto matrice-vettore. L'implementazione di tale operazione al calcolatore risulta del tutto banale nel caso di memorizzazione tabellare della matrice, mentre è necessaria qualche attenzione qualora si utilizzi il sistema di rappresentazione CRS. Nel seguito distingueremo fra l'implementazione del prodotto matrice-vettore per matrici non simmetriche, più intuitivo, e per matrici simmetriche in cui si memorizza la sola triangolare alta.

Caso non simmetrico Si vuole calcolare il prodotto matrice-vettore:

$$Av = w \tag{1.1}$$

con A matrice quadrata non simmetrica di ordine n , v e w vettori in \mathbb{R}^n . Si ricorda che la componente i -esima di w è pari alla somma dei prodotti degli elementi della riga i di A per i corrispondenti elementi di v :

$$w_i = \sum_{j=1}^n a_{ij}v_j \tag{1.2}$$

Se la matrice è memorizzata in modo compatto, gli elementi a_{ij} vanno opportunamente ricercati in **SYSMAT** mediante l'uso di **IA**, mentre gli indici j relativi alle colonne si trovano nel vettore intero **JA**. In particolare, gli elementi di A appartenenti alla riga i sono memorizzati in corrispondenza agli indici k del vettore **SYSMAT** compresi, per definizione di **IA**, nell'intervallo $\text{IA}(i) \leq k \leq \text{IA}(i+1) - 1$. Gli indici di colonna j , di conseguenza, sono memorizzati in $\text{JA}(k)$. Il prodotto matrice-vettore con A non simmetrica e memorizzata in forma compatta può quindi essere calcolato implementando il seguente algoritmo:

```

001      Per  $i = 1, n$ 
002          azzero  $w_i$ 
003          Per  $k = \text{IA}(i), \text{IA}(i+1) - 1$ 
004               $j := \text{JA}(k)$ 
005               $w_i := w_i + \text{SYSMAT}(k) \cdot v_j$ 
006          Fine Per
007      Fine Per

```

Si noti che è sempre utile azzerare il vettore soluzione prima di procedere all'implementazione del ciclo di sommatoria (riga 2), al fine di evitare l'uso improprio di valori precedentemente contenuti in w .

Caso simmetrico Si vuole ora calcolare il prodotto (1.1) in cui la matrice simmetrica A è memorizzata in formato CRS come descritto prima (e quindi memorizzando solo la parte triangolare. Poiché gli elementi a_{ij} con $j < i$ non sono ora immediatamente disponibili nella sequenza di coefficienti della riga i , conviene scrivere la definizione (1.2) come:

$$w_i = \sum_{j=1}^{i-1} a_{ji}v_j + \sum_{j=i}^n a_{ij}v_j \tag{1.3}$$

avendo sfruttato la condizione per cui $a_{ij} = a_{ji}$. Dalla (1.3) si deduce che il contributo a w_i relativo agli elementi con $j \geq i$ può essere implementato in maniera del tutto analoga a quanto fatto nel paragrafo precedente, mentre il contributo relativo agli elementi con $j < i$ può essere determinato selezionando in SYSMAT le componenti k per cui $\text{JA}(k) = i$, cioè appartenenti alla colonna i -esima. E' del tutto evidente, tuttavia, che questo modo di procedere, seppure intuitivo, risulta inapplicabile da un punto di vista pratico. Sarebbe, infatti, necessario procedere ad una ricerca su nt componenti per n volte, con un dispendio che renderebbe inutile il vantaggio fornito dalla memorizzazione compatta della matrice e, in certi casi, dalla convergenza accelerata del GCM.

Conviene osservare che gli elementi della triangolare alta della riga i , oltre a contribuire al calcolo di w_i , entrano in gioco, in virtù della simmetria di A , anche nel calcolo di w_j , con j pari all'indice di colonna dell'elemento considerato. All'interno del medesimo ciclo sulle righe i si aggiornerà pertanto non solo w_i ma anche tutti i w_j corrispondenti. L'algoritmo descritto nel precedente paragrafo viene quindi modificato nel modo seguente:

```

001      Per  $i = 1, n$ 
002          azzero  $w_i$ 
003      Fine Per
004      Per  $i = 1, n$ 
005           $k := \text{IA}(i)$ 
006           $w_i := w_i + \text{SYSMAT}(k) \cdot v_i$ 
007          Per  $k = \text{IA}(i) + 1, \text{IA}(i + 1) - 1$ 
008               $j := \text{JA}(k)$ 
009               $w_i := w_i + \text{SYSMAT}(k) \cdot v_j$ 
010               $w_j := w_j + \text{SYSMAT}(k) \cdot v_i$ 
011          Fine Per
012      Fine Per

```

Si noti che, a differenza dell'algoritmo utilizzato per matrici non simmetriche, in questo caso l'azzeramento del vettore prodotto w va fatto con un ulteriore ciclo (righe 1-3) esterno a quello di calcolo di w . Inoltre, il contributo a w_i dato dall'elemento diagonale a_{ii} viene conteggiato a parte (riga 6) per evitare di considerarlo due volte nel ciclo successivo (corrisponde infatti al caso in cui $i = j$).

2 Costruzione della topologia della matrice di rigidezza

Il metodo degli elementi finiti applicato alla soluzione di un'equazione differenziale lineare e stazionaria, cioè indipendente dal tempo, genera un sistema lineare la cui soluzione rappresenta

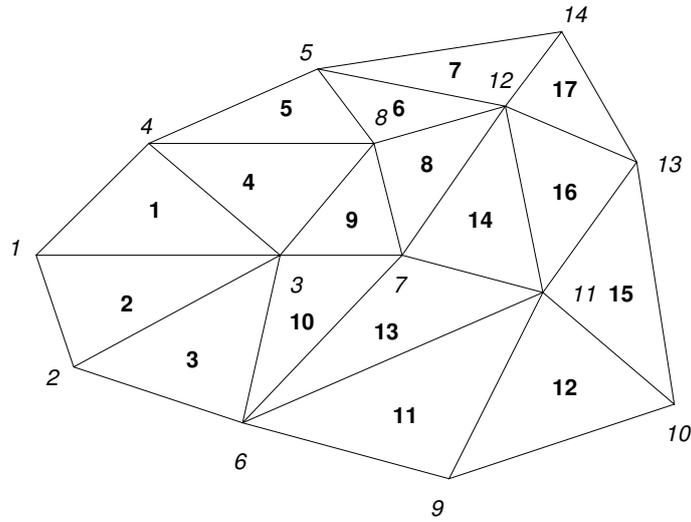


Figura 2.1: Patch di elementi triangolari. In corsivo sono indicati gli indici dei nodi ed in grassetto gli indici degli elementi.

il vettore dei valori che la funzione incognita assume sui nodi con i quali è stato discretizzato il dominio di integrazione. La matrice di tale sistema è generalmente chiamata *matrice di rigidezza* e nel seguito verrà indicata con il simbolo H . Essa è sparsa e si adatta bene alle tecniche CSR.

La topologia della matrice di rigidezza, vale a dire gli indici i, j dei coefficienti non nulli in H è determinata univocamente dalla mesh di calcolo con cui si è discretizzato il dominio di interesse. In particolare, è la rete dei contatti nodali nella maglia computazionale a stabilire la posizione degli elementi non nulli di H . Gli elementi non nulli della riga i corrispondono infatti agli indici di colonna j dei nodi con cui i risulta in contatto.

Per chiarire il concetto, ci si riferisca al patch di elementi riportato in Figura 2.1. Ad esempio, nella riga 7, relativa al valore assunto dalla funzione incognita nel nodo 7, gli elementi di H non nulli saranno:

$$h_{7,3} \ h_{7,6} \ h_{7,7} \ h_{7,8} \ h_{7,11} \ h_{7,12}$$

Ciò è naturale conseguenza del fatto che i polinomi che interpolano la soluzione approssimata in ciascun elemento finito hanno supporto locale. È ovvio che se il nodo i è in contatto con il nodo j , anche j sarà in contatto con i , e quindi la matrice H avrà una topologia simmetrica. Si osservi che, tuttavia, la simmetria della topologia non implica necessariamente la simmetria di H che, invece, nei casi di nostro interesse è assicurata dalla forma dei contributi locali, come si vedrà meglio in seguito. Risulta pertanto sufficiente memorizzare le posizioni $h_{i,j}$ per cui $j \geq i$. A partire dalla tabella dei contatti nodali, definita dalla maglia computazionale, è dunque possibile costruire i vettori \mathbf{JA} e \mathbf{IA} che individuano la topologia di H . Vediamo nel dettaglio l'implementazione di questa procedura sull'esempio di Figura 2.2. La mesh di calcolo viene

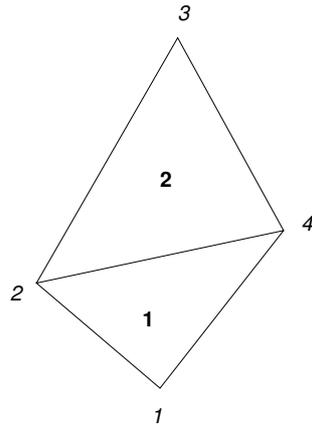


Figura 2.2: Patch di due elementi triangolari.

descritta dalla successione dei nodi e dalla topologia degli elementi. Avremo pertanto in input le seguenti strutture dati:

- coordinate nodali (es. file `coord` generato da MeshMaker):

```

4          : numero di nodi n
1  x1  y1
2  x2  y2
3  x3  y3
4  x4  y4

```

- successione dei nodi in senso antiorario e indice di zona per ciascun elemento (es. file `topol` generato da MeshMaker):

```

2          : numero di elementi ne
1  2  1  4  1
2  4  3  2  1

```

Dai contatti nodali, si deduce che la matrice di rigidezza, di dimensione $n \times n$, avrà la seguente struttura:

$$\begin{bmatrix} h_{1,1} & h_{1,2} & 0 & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} \\ h_{4,1} & h_{4,2} & h_{4,3} & h_{4,4} \end{bmatrix}$$

Grazie alla simmetria di H , si memorizza la sola parte triangolare alta inclusa la diagonale principale. Si verifica immediatamente che il vettore `JA` degli indici di colonna di $nt = 9$ componenti risulta:

$$\text{JA} = 1 \ 2 \ 4 \ 2 \ 3 \ 4 \ 3 \ 4 \ 4$$

mentre il vettore **IA** di $n + 1 = 5$ componenti contenente le posizioni in **SYSMAT**, e quindi in **JA**, degli elementi diagonali è:

$$\mathbf{IA} = 1 \ 4 \ 7 \ 9 \ 10$$

Applichiamo ora una procedura automatica che consenta di costruire **JA** dal file `topo1`. Definiamo $n1$ come il massimo numero di contatti nodali ammessi dalla griglia del problema, cioè il massimo numero di elementi non nulli che possiamo trovare su ciascuna riga di H . Nel caso in esame, $n1 = 4$ e di conseguenza il massimo numero di elementi non nulli da memorizzare sarà $nt = n1 \cdot n = 16$. Il vettore **JA** viene generato come una successione di nt componenti intere, inizialmente nulle, che possiamo interpretare come n celle, corrispondenti alle n righe di H , ciascuna composta di $n1$ elementi. In esse andranno inseriti gli indici di colonna degli elementi non nulli incontrati in ciascuna riga:

$$\boxed{0 \ 0 \ 0 \ 0} \mid \boxed{0 \ 0 \ 0 \ 0} \mid \boxed{0 \ 0 \ 0 \ 0} \mid \boxed{0 \ 0 \ 0 \ 0}$$

Il primo elemento di ogni cella corrisponde all'elemento diagonale, il cui indice di colonna j è uguale all'indice di riga i . Ciò significa che la prima componente di ogni cella deve essere pari all'indice della cella stessa, cioè sarà 1 nella prima cella, 2 nella seconda, e così via:

$$\boxed{1 \ 0 \ 0 \ 0} \mid \boxed{2 \ 0 \ 0 \ 0} \mid \boxed{3 \ 0 \ 0 \ 0} \mid \boxed{4 \ 0 \ 0 \ 0}$$

Per determinare i vari contatti nodali si esegue un ciclo sugli elementi della griglia di calcolo. Si consideri il triangolo **1** definito dalla successione di nodi *2, 1, 4*. Poiché nella cella i vanno inseriti solamente i contatti nodali relativi alla parte triangolare alta di H , cioè gli indici j tali che $j \geq i$, conviene ordinare gli indici dei nodi in senso crescente:

$$2 \ 1 \ 4 \Rightarrow 1 \ 2 \ 4$$

Si osservi che tale successione ordinata è utile solamente ai fini della determinazione della topologia di H , mentre per il calcolo dei contributi locali va mantenuta la successione originaria in senso antiorario. È consigliabile, pertanto, fare uso di un vettore ausiliario locale **I1(3)** in cui memorizzare provvisoriamente la sequenza nodale ordinata in senso crescente. Con un ciclo sulle componenti di **I1** appare ora chiaro che nella prima cella vanno aggiunti i contatti nodali 2 e 4, nella seconda va aggiunto il contatto 4 e nella quarta non va aggiunto nulla:

$$\boxed{1 \ 2 \ 4 \ 0} \mid \boxed{2 \ 4 \ 0 \ 0} \mid \boxed{3 \ 0 \ 0 \ 0} \mid \boxed{4 \ 0 \ 0 \ 0}$$

Seguiamo la medesima procedura per il triangolo **2**. Viene creato il vettore **I1** con la sequenza nodale ordinata in senso crescente:

$$4 \ 3 \ 2 \Rightarrow 2 \ 3 \ 4$$

Nella seconda cella vanno aggiunti i contatti ai nodi 3 e 4. A causa del modo in cui vengono memorizzati gli indici di colonna in **JA**, l'intero 3 va inserito fra il 2 ed il 4 già presenti, spostando

quindi l'ultimo indice avanti di una posizione. Inoltre, poiché il contatto con il nodo 4 è già stato individuato, null'altro va aggiunto alla seconda cella. Infine, nella terza cella va introdotto il contatto con il nodo 4:

1	2	4	0	2	3	4	0	3	4	0	0	4	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Va da sé che nell'ultima cella non andrà mai aggiunto nessun contatto. Si noti ora che il vettore così determinato corrisponde, a meno degli zeri, al vettore JA cercato.

La procedura seguita per la costruzione di JA può quindi essere riassunta nel seguente algoritmo:

```

001      nt := n1 · n
002      Per i = 1, nt
003          JA(i) := 0
004      Fine Per
005      Per i = 1, nt con passo n1
006          JA(i) := (i + n1 - 1) / n1
007      Fine Per
008      Per k = 1, ne
009          ordinamento nodi in senso crescente in I1(3)
010          Per j = 1, 2
011              m := (I1(j) - 1) · n1 + 1
012              Per i = j + 1, 3
013                  m := m + 1
014                  Se JA(m) = 0
015                      JA(m) := I1(i)
016                      vai all'istruzione 030
017                  Fine Se
018                  Se JA(m) < I1(i)
019                      vai all'istruzione 013
020                  Fine Se
021                  Se JA(m) > I1(i)
022                      sposta gli elementi della cella una posizione in avanti
023                      controlla che la posizione dell'ultimo elemento sia < n1 · I1(i)
024                      JA(m) := I1(i)
025                      vai all'istruzione 030
026                  Fine Se
027                  Se JA(m) = I1(i)
028                      vai all'istruzione 030
029                  Fine Se
030                      continua
031          Fine Per
032      Fine Per
033  Fine Per

```

La costruzione del vettore \mathbf{IA} è ora quasi immediata. La posizione dell'elemento diagonale della riga i corrisponde, infatti, al primo elemento della cella i in \mathbf{JA} , vale a dire $(i - 1) \cdot n1 + 1$. Il valore così determinato va quindi depurato del numero di zeri rimasti nelle celle precedenti. Il corrispondente algoritmo può pertanto essere scritto nel modo seguente:

```

001      IA(1):=1
002      Per  $i = 1, nt$  con passo  $n1$ 
003           $m := 0$ 
004          Per  $j = 1, n1$ 
005              Se  $\mathbf{JA}(i + j - 1) \neq 0$ 
006                   $m := m + 1$ 
007              Fine Se
008          Fine Per
009           $k := (i + n1 - 1)/n1 + 1$ 
010           $\mathbf{IA}(k) := \mathbf{IA}(k - 1) + m$ 
011      Fine Per

```

Si noti che mediante l'algoritmo appena riportato la componente $n+1$ -esima di \mathbf{IA} viene calcolata automaticamente.

Infine, è necessario compattare il vettore \mathbf{JA} eliminando tutti gli zeri in esso contenuti. Per far questo è sufficiente scorrere le componenti di \mathbf{JA} e, quando si incontra una componente nulla, spostare tutte le successive indietro di una posizione. Il numero di termini non nulli di \mathbf{JA} , che alla fine della procedura sono raggruppati nelle prime posizioni, è il vero nt .

Per eseguire in modo più efficiente questa operazione conviene, mano a mano che si scorrono le componenti di \mathbf{JA} , contare i termini non nulli e spostare l' m -esimo termine diverso da zero in posizione m . Tale posizione sarà necessariamente minore o tutt'al più uguale all'indice della componente raggiunta nello scorrimento. Per effettuare la compattazione di \mathbf{JA} si può pertanto utilizzare il seguente algoritmo:

```

001       $m := 0$ 
002      Per  $i = 1, nt$ 
003          Se  $\mathbf{JA}(i) \neq 0$ 
004               $m := m + 1$ 
005               $\mathbf{JA}(m) := \mathbf{JA}(i)$ 
006          Fine Se
007      Fine Per
008       $nt := m$ 

```

Poiché sappiamo che $\mathbf{IA}(n + 1)$ deve essere uguale a $nt + 1$, ma i due termini sono stati calcolati indipendentemente uno dall'altro, è sempre consigliabile controllare la consistenza della procedura verificando che effettivamente tale condizione sia rispettata.

Un'implementazione efficiente degli algoritmi per la costruzione della topologia della matrice di rigidezza è proposta nella subroutine `TOPOL.F`, messa a disposizione dello studente.

3 Assemblaggio della matrice di rigidezza dai contributi locali

Definita la topologia della matrice di rigidezza H mediante la costruzione dei vettori \mathbf{JA} e \mathbf{IA} , è ora necessario ricavare il vettore dei coefficienti non nulli `SYSMAT`. Come si vedrà meglio in seguito, la matrice globale H si ottiene da un processo di assemblaggio dei contributi locali calcolati su ciascun elemento finito che viene generalmente indicato con la seguente simbologia:

$$H_{i,j} = \sum_e H_{i,j}^{(e)}$$

I contributi locali sono contenuti in una matrice $H^{(e)}$ di dimensione $n^{(e)} \times n^{(e)}$, dove $n^{(e)}$ è il numero di nodi individuati nell'elemento finito prescelto. Ad esempio, con una griglia triangolare la matrice di rigidezza locale $H^{(e)}$ ha dimensione 3×3 .

Si consideri il generico elemento triangolare di nodi i, j, m , in cui i è il primo nodo, j il secondo ed m il terzo. L'operazione di assemblaggio viene eseguita collegando a ciascun elemento della matrice locale la posizione corrispondente nella matrice globale. Ad esempio, il coefficiente locale $h_{1,1}^{(e)}$, che si riferisce al primo nodo dell'elemento e , andrà collegato al coefficiente diagonale globale $h_{i,i}$, mentre il coefficiente $h_{1,2}^{(e)}$, che si riferisce all'interazione del primo col secondo nodo di e , va collegato ad $h_{i,j}$, e così via. Poiché ad uno stesso nodo concorrono più elementi finiti, ci sono più termini locali provenienti da diversi elementi che andranno collegati al medesimo termine globale. Essi perciò andranno sommati ai valori precedentemente accumulati nella matrice di rigidezza globale.

Per chiarire il concetto, si consideri nuovamente il patch di elementi di Figura 2.1. Supponiamo di eseguire l'assemblaggio del contributo locale dell'elemento **13** caratterizzato dalla successione nodale $7, 6, 11$. La matrice di rigidezza locale si collega ai nodi globali secondo il seguente schema:

	7	6	11
7	$h_{1,1}^{(13)}$	$h_{1,2}^{(13)}$	$h_{1,3}^{(13)}$
6	$h_{2,1}^{(13)}$	$h_{2,2}^{(13)}$	$h_{2,3}^{(13)}$
11	$h_{3,1}^{(13)}$	$h_{3,2}^{(13)}$	$h_{3,3}^{(13)}$

da cui si evince immediatamente che $h_{1,1}^{(13)}$ andrà collegato all'elemento $h_{7,7}$, $h_{1,2}^{(13)}$ all'elemento $h_{7,6}$, $h_{1,3}^{(13)}$ all'elemento $h_{7,11}$, e così via. Si osservi che, peraltro, a causa della simmetria di H ,

vengono memorizzati solo gli elementi relativi alla parte triangolare alta della matrice globale, per cui del contributo locale dato dal triangolo **13** interesseranno solamente gli elementi diagonali e quelli relativi ad $h_{6,7}$, $h_{6,11}$ ed $h_{7,11}$.

Come precedentemente accennato, per calcolare i termini globali di H si sommano i termini locali che concorrono al medesimo nodo. Ad esempio, l'assemblaggio del termine diagonale $h_{7,7}$ viene effettuato sommando tutti i contributi locali derivanti dagli elementi che afferiscono al nodo 7. Nel caso in esame, si tratta degli elementi **8**, **9**, **10**, **13** e **14**, descritti nel file `topo1` nel modo seguente:

```

...
8 8 7 12 1
9 8 3 7 1
10 3 6 7 1
...
13 7 6 11 1
14 7 11 12 1
...

```

L'elemento globale $h_{7,7}$ risulterà pertanto:

$$h_{7,7} = h_{2,2}^{(8)} + h_{3,3}^{(9)} + h_{3,3}^{(10)} + h_{1,1}^{(13)} + h_{1,1}^{(14)}$$

Ad esempio, l'elemento extra-diagonale $h_{7,11}$ riceverà un contributo dai soli elementi che hanno in comune il lato definito dai nodi 7 e 11, cioè i triangoli **13** e **14**. Dunque si avrà:

$$h_{7,11} = h_{1,3}^{(13)} + h_{1,2}^{(14)}$$

3.1 La matrice di puntatori TRIJA

L'operazione di assemblaggio della matrice di rigidezza globale è resa più complessa dalla memorizzazione compatta di H . L'elemento globale $h_{i,j}$ va, infatti, sempre cercato nel vettore `SYSMAT` a partire dalla posizione `IA(i)` attraverso lo scorrimento delle componenti di `JA`. Per facilitare tale operazione, si può definire una matrice tridimensionale di puntatori denominata `TRIJA` (Figura 3.1).

La matrice tridimensionale `TRIJA` è un insieme di matrici 3×3 generate per ciascun elemento triangolare. L'elemento `TRIJA(i, j, k)` individua l'indice *ind* della componente del vettore `SYSMAT` relativo alla matrice globale H a cui va aggiunto il contributo locale $h_{i,j}^{(k)}$, cioè il termine in posizione (i, j) della matrice locale generata sull'elemento k .

Determiniamo, ad esempio, la componente della matrice `TRIJA` relativa all'elemento **2** della patch riportata in Figura 2.2. Data la successione dei nodi $4, 3, 2$, il contributo locale alla matrice di rigidezza possiede la seguente struttura:

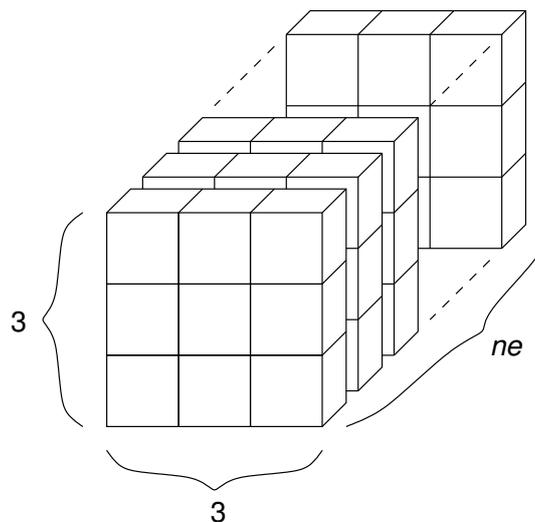


Figura 3.1: Struttura della matrice tridimensionale dei puntatori TRIJA.

	4	3	2
4	$h_{1,1}^{(2)}$	$h_{1,2}^{(2)}$	$h_{1,3}^{(2)}$
3	$h_{2,1}^{(2)}$	$h_{2,2}^{(2)}$	$h_{2,3}^{(2)}$
2	$h_{3,1}^{(2)}$	$h_{3,2}^{(2)}$	$h_{3,3}^{(2)}$

La corrispondente matrice $\text{TRIJA}(i, j, 2)$ avrà coefficienti non nulli solo nei termini per cui, detti ii e jj gli indici globali dei nodi, vale $jj \geq ii$. Ad esempio, l'elemento $h_{1,1}^{(2)}$ corrisponde ad $ii = jj = 4$, cioè all'elemento diagonale della matrice globale $h_{4,4}$. La posizione ind in **SYSMAT** di tale elemento è individuata da $\text{IA}(ii)$, vale a dire 9 nel caso in esame. Agli elementi $h_{1,2}^{(2)}$, $h_{1,3}^{(2)}$ e $h_{2,3}^{(2)}$ corrispondono indici $jj < ii$ e pertanto gli elementi ad essi collegati in **TRIJA** sono nulli. L'elemento $h_{2,1}^{(2)}$ contribuisce, invece, al termine globale $h_{3,4}$, la cui posizione ind in **SYSMAT** è determinata partendo da $\text{IA}(3)$ e scorrendo **JA** fintantoché si trova l'indice 4. Nel caso in esame, quindi, l'elemento corrispondente in **TRIJA** è 8. Proseguendo in questo modo per gli altri contributi locali, si trova $\text{TRIJA}(i, j, 2)$:

9	0	0
8	7	0
6	5	4

La procedura appena descritta può essere implementata mediante il seguente algoritmo:

```

001     Per  $k = 1, ne$ 
002         trasferimento nodi dell'elemento  $k$  in  $\text{I2}(3)$ 

```

```

003     Per  $i = 1, 3$ 
004          $ii := I2(i)$ 
005         Per  $j = 1, 3$ 
006              $jj := I2(j)$ 
007              $TRIJA(i, j, k) := 0$ 
008             Se  $jj \geq ii$ 
009                  $ind := IA(ii)$ 
010                 Se  $JA(ind) = jj$ 
011                      $TRIJA(i, j, k) := ind$ 
012                 Altrimenti
013                      $ind := ind + 1$ 
014                     vai all'istruzione 010
015                 Fine Se
016             Fine Se
017         Fine Per
018     Fine Per
019 Fine Per

```

Determinata la matrice dei puntatori `TRIJA`, l'operazione di assemblaggio dei contributi locali sulla matrice globale diventa pressoché immediata.