Consideriamo questa struttura a 4 elementi e 9 nodi, utile per descrivere la trasposizione dei *gdl* da globali a nodali:



Affrontando la th. di Kirchhoff, abbiamo visto che per ogni elemento modellato come *shell*, la relazione tra le deformazioni membranali al piano di riferimento e curvature, e i flussi degli sforzi e di momento è individuata da:

dove la matrice assemblata definisce il legame tra:

* caratteristiche di deformazione generalizzate dell’elemento

(deformazione membranale e curvature );

* caratteristiche di sollecitazione generalizzata dell’elemento

(flussi di sforzi e momenti).

È a tutti gli effetti una matrice di rigidezza del singolo elemento.

Ciò che vogliamo è però ricavare una matrice di rigidezza dell’intera struttura, ossia una matrice che premoltiplica il vettore dei *gdl* globali della struttura [[1]](#footnote-1), restituendo il vettore globale delle forze che deve agire su tale struttura per mantenerla in configurazione deformata, andando di fatto ad equilibrare le reazioni elastiche indotte dagli spostamenti e dalle rotazioni imposti.

Chiameremo il vettore globale delle forze cumulative, costituito da :

* Forze ext. applicate alla struttura;
* Forze da applicare alla struttura per contrastare le forze di reazione elastica del materiale agli spostamenti imposti, al fine di mantenere il corpo nella configurazione deformata

La sarà assemblata raccogliendo i contributi di rigidezza di tutti gli elementi costituenti la struttura.

Al fine di poter sommare i contributi di ogni singolo elemento alla rigidezza della struttura, è necessario però riferirsi ad un sistema di riferimento comune a tutti gli elementi. Si nota infatti che ogni elemento, al momento, ha una propria terna di riferimento con asse *z* normale al “piano di riferimento”.

Considereremo i nodi globali, ai quale associo un unico sistema di riferimento omonimo. Il sistema di riferimento globale mi consentirà di interfacciare due elementi che insistono sullo stesso nodo.

Pertanto, tutti i nodi globali della struttura sono associati alla terna di riferimento arbitrariamente scelta: .

Come visibile ulteriormente dalla figura, dovremo distinguere la numerazione locale dei nodi da quella globale.

In numerazione globale individueremo:

* i nodi :
* gli elementi :

In numerazione locale invece, la nomenclatura del nodo è influenzata dall’elemento considerato.

Ad esempio, per l' *elemento 1* individueremo i nodi: .

È chiaro è necessario disporre di una mappatura univoca tra numerazione locale dei nodi e numerazione globale. Tale mappatura è fornita dalla seguente tabella:



È la stessa tabella reperibile al file data input del solutore. Come leggerla?

La tabella ci dice ad esempio che il quarto nodo locale dell' *elemento 3* , , coincide con il nodo globale ; per gli altri nodi del medesimo elemento, identicamente si avrà:



Per stabilire la numerazione locale dei nodi di un generico elemento, analogamente alla convenzione adoperata dal Mentat, viene riportata una freccia (rossa in figura) sul lato che contiene i primi due nodi locali dell’elemento. L’orientazione di tale freccia indica la direzione di numerazione progressiva dei nodi locali.

Una volta trovata la mappatura tra numerazione locale e globale dei nodi, devo riuscire ad esprimere una relazione tra i *gdl* locali e quelli globali, poiché l’obiettivo è quello di arrivare a scrivere per l’intera struttura la relazione:

Prendiamo ad esempio il nodo ; il *gdl* di spostamento in direzione z locale è chiamato . Possiamo rappresentarlo sull’ *elemento 1* sottoforma di vettore spostamento:



Tale vettore è in generale scomponibile lungo le direzioni parallele agli assi del *SR* globale associato al nodo g2:

Nella figura, per come è fatta la struttura, si è scelto di considerare componente di spostamento parallela al terzo asse globale, individuato dal versore , nulla.

Analogamente posso prendere un altro *gdl* locale dell’elemento, la rotazione attorno all'asse x locale del nodo 1, , e rappresentarla in forma di vettore moltiplicandola per il versore associato (freccia verde a doppia testa in figura).

Nuovamente scomponiamo tale vettore lungo le direzioni parallele agli assi della terna globale , ottenendo:

È possibile effettuare tale operazione su tutti i *gdl* locali relativi ai nodi di ciascun elemento.

Sfruttando una relazione vettoriale [[2]](#footnote-2) abbiamo in realtà implicitamente scritto una relazione tra un *gdl* locale e un set di 3 *gdl* globali .

Come faccio dunque ad estrarre delle relazioni scalari da queste relazioni vettoriali, che mettano realmente in evidenza il legame tra i *gdl* nei due *SR*, anziché legami tra vettori?

Riconsideriamo l’equazione:

semplicemente, è sufficiente proiettare l’equazione lungo la direzione , ovvero moltiplicando scalarmente ambo i membri per il versore .

Noto che:

I prodotti scalari:

* ;
* ;
* ;

altro non sono che il coseno dell'angolo compreso tra i due versori[[3]](#footnote-3).

Abbiamo finalmente la possibilità di rappresentare un *gdl* locale come combinazione di tre *gdl* globali.

Voglio ottenere una relazione matriciale che metta in relazione il vettore dei *gdl* globali, , con il vettore dei *gdl* locali di un generico elemento . Quanto mostrato è da considerarsi valido per ogni elemento.

54 componenti

 è il vettore dei *gdl* locali del j-esimo elemento .

Ogni elemento ha 4 nodi e ogni nodo ha 6 gradi di libertà. Dunque, il vettore ha 24 componenti.

24 componenti

Cerco dunque una matrice, che moltiplicata per il vettore mi restituisca il vettore . Data la differente dimensione dei due vettori dei *gdl,* stiamo parlando di una matrice rettangolare di dimensione 24 x 54.

Definiamo tale matrice nello specifico:



Prendiamo per esempio il decimo grado di libertà del vettore :

dove *i* è l’indice che scorre le colonne della matrice, nonché le “righe” del vettore colonna .

Definire la matrice, significa determinare gli elementi della riga affinché la relazione appena citata sia rispettata.

Abbiamo visto prima che :

dunque, nel calcolo di non rientrano tutte le componenti del vettore dei gradi di libertà, ma solamente:

* : 7^ componente di ;
* : 8^ componente di ;
* : 9^ componente di .

Dunque, i termini della 10^ riga della matrice saranno tutti nulli, fatta eccezione per:

* ;
* ;
* .

NB: identico ragionamento per quanto riguarda , tredicesimo elemento del vettore .

Chiamiamo matrice di mappatura , la matrice di trasformazione da vettore dei *gdl* globali nel vettore di *gdl* locali dell’ *e1*.

Ripetendo tale procedura per tutti i j-elementi componenti la struttura, si costruiscono tutte le matrici di mappatura. Nel nostro caso, *j=4 :*

* : matrice di mappatura per l’elemento 1 ;
* : matrice di mappatura per l’elemento 2 ;
* : matrice di mappatura per l’elemento 3 ;
* : matrice di mappatura per l’elemento 4 .



Si nota che queste matrici, esclusi alcuni slot, risultano essere piene di zeri. Cosa significa? L’elemento 1 ad esempio non tocca i nodi: g3, g6, g7, g8, g9. Di fatti la matrice non interviene in alcun modo sui *gdl*, presentando un termine nullo.

Se dovessimo programmare un codice del genere su Matlab o in un qualunque altro software che preveda calcolo matriciale, non avrebbe senso utilizzare uno stoccaggio “denso” per tali matrici. L’alta concentrazione di zeri, rende inutile l’allocazione di memoria destinata alla loro memorizzazione, conviene invece fornire al manipolatore esclusivamente i termini non nulli utilizzando quello che si chiama stoccaggio “sparso”. Tutti gli elementi non citati saranno assunti implicitamente come nulli.

È possibile ricavare la relazione inversa? ossia, partendo da un *gdl* globale, possiamo ricondurci a uno o un set di *gdl* locali?

Tali considerazioni, più che per i vettori di spostamento, risultano utili per le forze. Nella figura seguente si è scelta una componente di forza globale lungo la direzione del secondo asse globale del nodo 7. Si procede dunque ad individuare il vettore nel *SR* globale.



Tale vettore globale, risulta composizione di forze locali:

In maniera del tutto analoga posso andare a ricavare una relazione inversa alla precedente, ossia da globale a locale, tramite la definizione di una matrice di mappatura inversa.

È improprio però parlare di inversa per la matrice di mappatura . Questa è infatti una matrice rettangolare!

Il criterio di generalizzazione formulato da Moore\_Penrose, nel caso in cui il problema sia sovradeterminato (nro equazioni >> nro incognite), permettere di risolvere ugualmente il problema matriciale con non quadrata. La soluzione del problema è stimata con errore ai minimi quadrati, nella forma , dove la matrice prende il nome di pseudo-inversa.

Il criterio afferma che se la matrice è:

* reale ;
* con righe ortonormali l'una rispetto all'altra, ossia le righe sono vettori a norma unitaria tra loro ortogonali;

>> la pseudo-inversa secondo Moore\_Penrose coincide con la trasposta .

Siccome gode proprio di queste proprietà, per Moore\_Penrose: .

Dunque non è un problema considerare la relazione inversa.

Ottenute le relazioni di conversione tra i due sistemi di riferimento, lavorando nel sistema globale comune a tutti gli elementi, la matrice di rigidezza della struttura sarà quindi assemblata raccogliendo i contributi di rigidezza di tutti gli elementi costituenti la struttura.

La matrice rigidezza del j-esimo elemento, , premoltiplica i *gdl* del j-esimo elemento restituendo le forze che devo applicare a tali *gdl* locali per mantenere l'elemento in configurazione deformata:

Noto che posso derivare i *gdl* locali come prodotto tra la matrice di mappatura dell'elemento j-esimo per il vettore dei gdl globali, , possiamo scrivere:

Vogliamo ora tradurre tali forze nel *SR* globale , considerando dunque le forze che a livello globale devono essere applicate per mantenere deformato l'elemento j-esimo.

Per ottenerle si sfrutta la relazione inversa da locale a globale. In virtù della pseudo-simmetrica di Moore\_Penrose , è necessario pre-moltiplicare per la trasposta della matrice di mappatura:

Chiamiamo , le forze globali da applicare per l’influenza del solo j-esimo elemento.

Raccogliendo le forze necessarie per deformare ogni singolo elemento in un vettore [[4]](#footnote-4):

L’assemblaggio di , come già anticipato, è dunque ottenuto per somma di contributi di rigidezza dei singoli elementi :

NB: essendo una matrice , mentre le delle , le matrici di mappatura che opera la trasformazione di coordinate vengono dette anche matrici di “diluizione”.

Per dare un senso a questa procedura di diluizione a gdl globali della matrice di rigidezza del j-esimo elemento e della successiva loro somma, può essere fornita una resa grafica

**Assemblaggio della matrice di rigidezza** :

* Si parte allocando una matrice con termini tutti nulli (spazi bianchi nella figura) e via via si vanno a sommare i singoli contributi di rigidezza ( colore più scuro = valore più alto).



* Alla matrice vuota sommiamo la matrice associata al primo elemento, , rappresentata in figura (a) . I termini nulli, fanno capo a nodi in alcun modo associati all’*elemento 1*  ( i nodi su cui elemento 1 insiste sono 1,2,4,5) .
* Calcoliamo la matrice associata all’*elemento 2*. Questa avrà termini non nulli solo in corrispondenza dei nodi su cui l’*elemento 2* insiste, che sommiamo alla matrice al punto precedente, ottenendo la matrice in figura (b). I contributi nulli di questa nuova matrice sono relativi ai nodi che non si interfacciano né con *e1* né con *e2 .*
* ripeto il calcolo della matrice di rigidezza ridotta a gdl globali per l’elemento 3 e sommo l’ulteriore termine.

Noto osservando le righe e colonne nulle che l’unico nodo non toccato da *e1, e2, e3* è il nodo 9 ( righe e colonne contengono blocchi, ogni blocco è una matrice 6x6 ! )

* Sommando tutti i contributi otteniamo la matrice di rigidezza completa della struttura che lega il vettore dei gdl globali alle forze da applicare a tali gdl.

 è simmetrica poiché somma di contributi simmetrici.

I termini più lontani dalla diagonale risultano tutti nulli in questo particolare caso, data la numerazione dei nodi appositamente imposta per evidenziare una caratteristica tipica di questo tipo di matrici.

Una matrice simmetrica di questo tipo prende il nome di “matrice bandata”.

***MATRICI BANDATE***

*i*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | *i* |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

* prendiamo il generico termine della diagonale principale, intercettato dalla i-riga e i-colonna
* assumo che gli unici altri termini non nulli delle righe e colonne di quella matrice siano concentrati nell’intorno sella diagonale, ossia, ogni elemento appartenente alla diagonale è circondato sia sulla riga che su la colonna che lo individua da termini (di modo che ).

Se ciò è valido per qualsiasi termine appartenente alla diagonale principale, allora la matrice si dice bandata di entità (o semibandata visto che parliamo di una matrice simmetrica)

Se dovessimo risolvere un sistema di equazioni del tipo:

il costo computazionale è:

dove n è l’ordine della matrice [[5]](#footnote-5).

Si capisce dunque che se è possibile rendere bandata la matrice di rigidezza , il vantaggio si tramuta in tempo inferiore di calcolo per tutte le trattazioni che richiedono l’utilizzo di tale matrice.

Il solutore Mentat, attraverso algoritmi di calcolo, corregge la numerazione globale dei nodi al fine di stringere la banda il più possibile e rendere il calcolo più veloce.

Ulteriori vantaggi sono legati al risparmio di memoria. Ad esempio: se una matrice è simmetrica, è sufficiente allocare uno spazio sufficiente a memorizzare la parte triangolare superiore o inferiore, ma se in aggiunta so che è anche bandata, non è necessario stoccare neppure tutta parte triangolare superiore, ma sarà sufficiente solo la striscia della banda!

Si consideri ad esempio una matrice simmetrica 6x6 con banda pari a 2:



{\displaystyle {\begin{bmatrix}A\_{11}&A\_{12}&A\_{13}&0&\cdots &0\\&A\_{22}&A\_{23}&A\_{24}&\ddots &\vdots \\&&A\_{33}&A\_{34}&A\_{35}&0\\&&&A\_{44}&A\_{45}&A\_{46}\\&sym&&&A\_{55}&A\_{56}\\&&&&&A\_{66}\end{bmatrix}}}

Tale matrice può essere memorizzata come una matrice 6 per 3:



{\displaystyle {\begin{bmatrix}A\_{11}&A\_{12}&A\_{13}\\A\_{22}&A\_{23}&A\_{24}\\A\_{33}&A\_{34}&A\_{35}\\A\_{44}&A\_{45}&A\_{46}\\A\_{55}&A\_{56}&0\\A\_{66}&0&0\end{bmatrix}}}Come di calcola la banda necessaria a stoccare ogni elemento?

La banda della matrice finale è la peggior banda di elemento. Dunque, sarà sufficiente calcolare la banda per tutti gli elementi. La peggiore penalizza la struttura.

La banda necessaria a stoccare l’ elemento j-esimo (che insiste sugli *i* nodi):

dove:

* : *label* max. tra i nodi su cui l’elemento j-esimo insiste
* : *label* min. tra i nodi su cui l’elemento j-esimo insiste
* *l*  : *gdl* di ciascun nodo

*Per esempio:*

L’*elemento 1* insiste sui nodi 1,2,4,5 .

* prendo nodo max label : 5
* prendo nodo min label : 1

è la banda necessaria per stoccare l’elemento *e1*, che connette il primo *gdl* con il trentesimo *gdl.*

Come ben si vede, la larghezza della banda, dunque conseguentemente lo sforzo computazionale, non dipende dal numero di nodi né tantomeno dai gdl di ogni nodo ma da come gli elementi realizzano il ponte elastico con i nodi, agganciandone i relativi gdl .

Assemblata dunque la matrice di rigidezza globale , le forze ext globali agenti sull’elemento:

Resta da trattare i vincoli.

**VINCOLI**

Per introdurli consideriamo una semplice struttura, tale da consentire di individuare con semplicità le possibili configurazioni di deformazione:



NON VINCOLATO ( a )

Struttura a 3 *gdl* : i moti dei tre vertici del triangolo rigido definisce tutte le possibili configurazioni.

La matrice di rigidezza della struttura lega il vettore degli spostamenti associato ai *gdl* disponibili (ogni nodo ha 1 solo gdl) al vettore delle forze ext da applicare ai nodi per mantenere la struttura nella configurazione deformata.

VINCOLATO (b)

inserisco un multi-DOF constrain, vincolo che lega i nodi 1 e 3 imponendo la relazione cinematica

* spostamento 1 e 3 opposti
* in modulo lo spostamento di 3 è tre volte lo spostamento di 1 a causa della leva

Cinematicamente è un vincolo omogeneo:

o equivalentemente, in maniera molto più comprensibile:

È un vincolo lineare. Anche se non fosse lineare, il solutore procederebbe ad una linearizzarione nell’intorno di una configurazione di primo tentativo [[6]](#footnote-6).

Introduco anche un secondo vincolo a singolo gdl imposto invece: SINGLE DOF CONSTRAIN [[7]](#footnote-7)

Lo spostamento imposto al nodo 2 è un vincolo non omogeneo:

Il F.E.M. non è in grado di implementare multipoint constrain non omogenei, ma esistono delle modalità per bypassare il problema.

I due vincoli sostanzialmente hanno la stessa forma di scrittura.

Posso scrivere il primo vincolo in forma vettoriale:

 : vettore dei gdl globali

Allo stesso modo per il secondo vincolo:

Nel caso in cui si abbia dunque un set di j vincoli, possiamo esprimerli unicamente come:



dove *i* è un indice che scorre sui *gdl* globali.

Andiamo a dare un senso fisico a quanto esposto:



Consideriamo lo spazio delle possibili configurazioni, che in base ai *gdl* disponibili nel nostro caso coincide con .

In assenza di vincolo, esistono configurazioni possibili.

Applicando il vincolo I , le configurazioni possibili sono limitate al piano individuato dall’equazione di vincolo : configurazioni possibili.

Applicando il vincolo II , le configurazioni risulterebbero invece limitate ad un piano che non passa per l’origine, poiché descritto dall’equazione non omogenea .

Poiché i vincoli esplicano la loro azione contemporaneamente a modificare le configurazioni del sistema, le configurazioni compatibili al vincolamento risulteranno limitate all’intersezione tra il piano I e II.

Tali piani sono legati ai vettori e , infatti dato un piano nello spazio definito dall’equazione:

tale piano è ortogonale al vettore con componenti:

dunque, i piani che definiscono le configurazioni possibili secondo i vincoli I e II sono rispettivamente ortogonali ai vettori e .

L’intersezione dei due piani è ortogonale sia e .

L’ortogonalità è strettamente legata ai vincoli lisci. Un vincolo è detto liscio quando le reazioni vincolari sono ortogonali ai possibili spostamenti. In virtù di questo, le reazioni vincolari compiono lavoro nullo sugli spostamenti in presenza di vincoli lisci.

1. Nel caso in esame abbiamo 54 gdl globali (9 nodi, 6 *gdl*/nodo). [↑](#footnote-ref-1)
2. ossia legando i vettori spostamento/rotazioni lungo i gdl locali con le rispettive componenti nel SR globale [↑](#footnote-ref-2)
3. Si noti in questo specifico caso che $<\hat{k}\_{e1},\hat{k}\_{g2}> =0$ poiché, per come si è scelto il *SR* globale, i due versori sono ortogonali. [↑](#footnote-ref-3)
4. Nella dispensa fornita a lezione sono stati indicati con $\overline{F}$ al posto che con $\overline{G}$ poiché $\overline{F}$ è adoperato solo per le forze ext. In questo caso invece raccogliamo sia i contributi delle forze ext. applicate, che di quelle da applicare per tenere in configurazione deformata tutta la struttura. [↑](#footnote-ref-4)
5. in caso di problema ben condizionato, l’ordine della matrice coincide con il numero delle equazioni, ovvero con il numero delle incognite nonché con il numero di termini noti. [↑](#footnote-ref-5)
6. RBE2, INSERTS, SERVOLINK , disponibili al menù LINKS di Marc sono costruiti in questa maniera, vengono inseriti in input coefficiente per coefficiente. [↑](#footnote-ref-6)
7. un fixed displacement in pratica [↑](#footnote-ref-7)