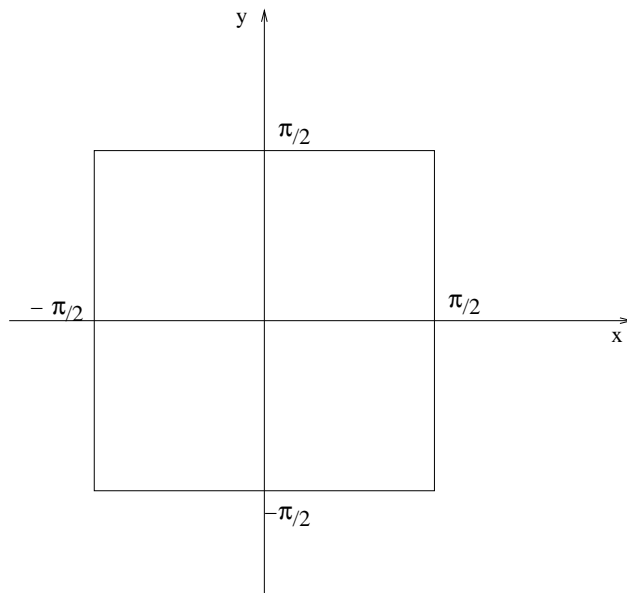


Problema da risolvere

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{f(x, y)}{\rho c K_H} = q(x, y)$$



- ✗ T [$^{\circ}C$] è la temperatura,
- ✗ K_H [m^2/s] è il coefficiente di diffusività termica
- ✗ ρ [Kg/m^3] è la densità della piastra
- ✗ c [$Cal/Kg^{\circ}C$] è il calore specifico
- ✗ $f(x, y)$ [Cal/m^2s] è il calore aggiunto o sottratto alla piastra per unità di tempo e di area.

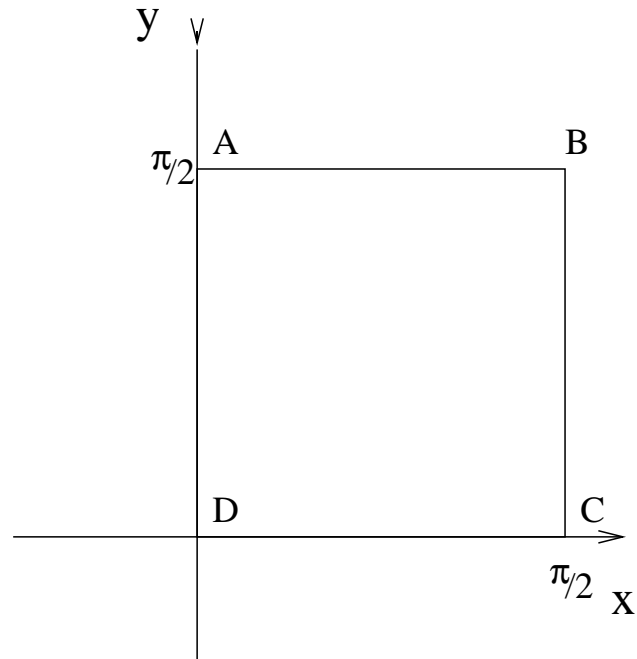
Poniamo $q(x, y) = 4$ e consideriamo le condizioni al contorno:

$$T = 0 \quad \text{per } x, y = \pm\pi/2$$

La soluzione analitica del problema è:

$$T(x, y) = -\frac{\pi^2}{2} + 2x^2 + \frac{16}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^{k+1} \cosh [(2k-1)y]}{(2k-1)^3 \cosh [(2k-1)\pi/2]} \cos [(2k-1)x]$$

Data la **simmetria** del problema, noi lo risolviamo sulla porzione di piastra del quadrante positivo di lato $\pi/2$:



Ora le condizioni al contorno diventano:

$$\left\{ \begin{array}{l} T = \bar{T} = 0 \quad \text{sui lati AB e BC: condizione di Dirichlet} \\ \frac{\partial T}{\partial n} = 0 \quad \text{sui lati AD e DC: condizione di Neumann} \\ \text{(osserviamo che è una condizione di flusso nulla)} \end{array} \right.$$

Soluzione agli Elementi Finiti

→ Il funzionale che associamo al problema da risolvere è:

$$\Omega(T) = \iint_S \frac{1}{2} \left[\left(\frac{\partial T}{\partial x} \right)^2 + \left(\frac{\partial T}{\partial y} \right)^2 \right] + qT \, dS$$

→ Applichiamo il **metodo variazionale di Ritz**: cerchiamo una soluzione approssimata \tilde{T} dove

$$\tilde{T} = \sum_{i=1}^n T_i \xi_i(x, y)$$

dove $\xi_i = \xi_i(x, y)$, $i = 1, \dots, n$ sono opportune funzioni di forma o funzioni base e i T_i corrispondono ai valori puntuali assunti da T su n punti del dominio S (le incognite del problema).

Sostituendo \tilde{T} nel funzionale Ω , calcolando le derivate di $\Omega(\tilde{T})$ rispetto ai coefficienti T_i e ponendole uguali a zero (per determinare la soluzione che minimizza il funzionale), otteniamo:

$$\frac{\partial \Omega(\tilde{T})}{\partial T_i} = \iint_S \left[\left(\frac{\partial \tilde{T}}{\partial x} \frac{\partial \xi_i}{\partial x} + \frac{\partial \tilde{T}}{\partial y} \frac{\partial \xi_i}{\partial y} \right) + q \xi_i \right] dS = 0 \quad i = 1, \dots, n$$

e, ancora,

$$\iint_S \left[\sum_{j=1}^n \left(\frac{\partial \xi_j}{\partial x} \frac{\partial \xi_i}{\partial x} + \frac{\partial \xi_j}{\partial y} \frac{\partial \xi_i}{\partial y} \right) T_j + q \xi_i \right] dS = 0 \quad i = 1, \dots, n$$

Queste n equazioni che ricaviamo costituiscono un sistema di n equazioni nelle n incognite T_1, T_2, \dots, T_n :

$$HT + f = 0$$

Il sistema, scritto come matrice x vettore incognito = termine noto è:

$$HT = -f$$

Stiamo dunque attenti al segno del termine noto!

Questo sistema possiede un'infinità di soluzioni perchè la matrice è singolare (determinante nullo). - **perchè?** -

Per rendere la matrice non singolare e determinare un'unica soluzione occorre imporre le condizioni al contorno di Dirichlet.

Utilizziamo il metodo variazionale di Ritz utilizzando polinomi di interpolazione bidimensionali continui a tratti con supporto locale ed elementi triangolari.

L'elemento i, j della matrice H è il risultato dell'assemblaggio dei contributi locali di ciascun elemento:

$$h_{ij} = \sum_e h_{ij}^{(e)} = \sum_e \iint_{S^{(e)}} \left[\frac{\partial \xi_j}{\partial x} \frac{\partial \xi_i}{\partial x} + \frac{\partial \xi_j}{\partial y} \frac{\partial \xi_i}{\partial y} \right] dS^{(e)}$$

Con la scelta fatta, H è simmetrica, definita positiva, sparsa: possiamo applicare il CGM per risolvere il sistema.

Il termine noto è:

$$f_i = \sum_e f_i^{(e)} = \sum_e \iint_{S^{(e)}} q \xi_i dS^{(e)}$$

Sulla programmazione

Il programma che dobbiamo costruire per risolvere il problema si basa su un algoritmo abbastanza complicato (o difficile) in quanto deve tener conto dell'equazione alle derivate parziali da risolvere, della griglia su cui andare ad approssimare la soluzione, della memorizzazione in forma compatta della matrice del sistema, del gradiente coniugato modificato...

Conviene perciò cercare di dare al codice una struttura modulare (utilizzando diverse subroutine, ad esempio, a seconda di ciò che si deve fare), seguendo uno schema del genere:

- ↳ dichiarazione di tutte le variabili utilizzate
- ↳ lettura dei dati di input (griglia, condizioni al contorno, caratteristiche fisiche)

- ↳ calcolo della topologia della matrice H (matrice di rigidezza) (J_A, I_A)
- ↳ calcolo della matrice dei puntatori per l'assemblaggio (TRIJA)
- ↳ calcolo e assemblaggio dei contributi locali nella matrice di rigidezza
- ↳ calcolo del vettore dei termini noti
- ↳ imposizione delle condizioni al contorno
- ↳ soluzione del sistema lineare attraverso il CGM
- ↳ stampa dei risultati ottenuti e calcolo dell'errore

La griglia

Suddividiamo il dominio in elementi triangolari (per far ciò abbiamo già il programma FORTRAN `griglia.f`): abbiamo bisogno infatti di una tabella che fornisce per ciascun elemento la successione in senso antiorario degli indici dei nodi che lo caratterizzano.

Ciascun nodo è determinato dalle coordinate cartesiane x e y .

ATTENZIONE: la successione dei nodi per ciascun triangolo è data riga per riga, in output dal programma `griglia.f` in questo modo: alla riga i -sima della tabella abbiamo il primo, il secondo e il terzo nodo del triangolo i -simo.

Per ragioni di locazione di memoria, conviene memorizzare la matrice non come $NE \times 3$ ($NE = \#$ dei triangoli) ma come $3 \times NE$.

Se chiamiamo TRIANG questa matrice (è uno dei parametri di input della subroutine `topol.f`), la lettura avverrà in questo modo

```
real*8 TRIANG(3,NEMAX)
!NEMAX e' il numero massimo di elementi
.
.
read(11,*) n,ne
!il file associato all'etichetta 11 e' quello da cui
!leggiamo i dati della griglia
do i=1,ne
    read(11,*) (triang(j,i), j=1,3)
end do
```

Analogamente, possiamo associare alle coordinate dei nodi, due vettori di dimensione n che chiamiamo x e y .

Una volta data la griglia, dobbiamo individuare le zone del contorno caratterizzate da condizioni di Dirichlet, di Neumann o miste.

Nel problema che dobbiamo risolvere, **le condizioni al contorno di Neumann sono di tipo naturale (flusso nullo)**, e quindi sono automaticamente assegnate sulla parte inferiore e sul lato sinistro del contorno e, di conseguenza, non bisogna imporre alcuna condizione in maniera esplicita.

Per **le condizioni di Dirichlet**, invece, dobbiamo noi individuare quali sono i nodi in cui si trova una condizione di Dirichlet e assegnare ad essi il valore di Dirichlet; quindi dobbiamo selezionare i nodi che cadono sul lato superiore e sul lato destro del dominio ed associare a ciascuno di essi un valore $\bar{T} = 0$.

Una via per dire quali sono i nodi di Dirichlet, una volta che sono stati individuati (vedendo la griglia e la numerazione dei nodi) è di associare ad essi un vettore.

Un esempio di come procedere:

se la mesh ha 8 triangoli i nodi del lato superiore e del lato destro sono i nodi 1, 4, 7, 8, 9. Quindi sono 5 nodi.

Diamo in input il numero totale di nodi di Dirichlet e la loro numerazione globale, vale a dire

5

1 4 7 8 9

Possiamo memorizzare in una variabile $ndir$ il numero totale dei nodi di Dirichlet e in un vettore di interi $nodedir$ i nodi di Dirichlet, per cui $nodedir(i)$ è l' i -simo nodo di Dirichlet per $i=1, ndir$. Nell'esempio

$nodedir(1)=1$ $nodedir(2)=4$ $nodedir(3)=7$

$nodedir(4)=8$ $nodedir(5)=9$

Calcolo della topologia

I vettori JA e IA sono univocamente determinati dai contatti nodali definiti dalla maglia computazionale, i quali individuano tutti gli elementi potenzialmente non nulli della matrice di rigidità. Poiché nel nostro caso la matrice è simmetrica, si memorizzano solo gli elementi della parte triangolare alta di H e cioè solo gli elementi h_{ij} per cui $j \geq i$

(subroutine `topol.f`).

Calcolo della matrice dei puntatori

Il vettore di matrici di puntatori TRIJA permette l'operazione di assemblaggio dei contributi locali nella matrice globale:

TRIJA(i, j, k) individua un indice ind in cui va aggiunto nella matrice globale il contributo locale $h_{ij}^{(k)}$.

Calcolo e assemblaggio dei contributi locali

Sul generico elemento triangolare e della griglia indichiamo con $\tilde{T}^{(e)}$ la soluzione approssimata

$$\tilde{T}^{(e)}(x, y) = \sum_{k=1}^3 T_k \xi_k(x, y)$$

Siano i, j, m i tre nodi dell'elemento e presi in senso antiorario. Le funzioni di forma ξ_k , infatti, sono calcolate utilizzando un'interpolazione lineare e sono convenientemente scelte in modo tale da:

- assumere valori non nulli solamente all'interno dell'elemento (**supporto locale**);
- valere 1 sul nodo cui sono associate e 0 sugli altri.

Abbiamo:

$$\begin{aligned}\xi_i(x, y) &= (a_i + b_i x + c_i y) / 2\Delta \\ \xi_j(x, y) &= (a_j + b_j x + c_j y) / 2\Delta \\ \xi_m(x, y) &= (a_m + b_m x + c_m y) / 2\Delta\end{aligned}$$

Δ è l'area dell'elemento:

$$\Delta = \frac{1}{2} \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix},$$

i coefficienti a_i , b_i e c_i sono dati da:

$$a_i = x_j y_m - x_m y_j$$

$$b_i = y_j - y_m$$

$$c_i = x_m - x_j$$

e gli altri si ottengono con una permutazione degli indici in senso antiorario:

$$a_j = x_m y_i - x_i y_m \quad b_j = y_m - y_i \quad c_j = x_i - x_m$$

$$a_m = x_i y_j - x_j y_i \quad b_m = y_i - y_j \quad c_m = x_j - x_i$$

Il termine generico della matrice locale del sistema FEM (detta **matrice di rigidezza locale**) vale:

$$h_{ij}^{(e)} = \iint_{S^{(e)}} \left[\frac{\partial \xi_j}{\partial x} \frac{\partial \xi_i}{\partial x} + \frac{\partial \xi_j}{\partial y} \frac{\partial \xi_i}{\partial y} \right] dS^{(e)} = \frac{1}{4\Delta} (b_i b_j + c_i c_j) \quad (1)$$

e quindi la matrice di rigidezza locale $H^{(e)}$ per un elemento triangolare a 3 nodi si può esprimere come:

$$H^{(e)} = \frac{1}{4\Delta} \left\{ \begin{array}{c} \left[\begin{array}{ccc} b_i b_i & b_i b_j & b_i b_m \\ b_j b_i & b_j b_j & b_j b_m \\ b_m b_i & b_m b_j & b_m b_m \end{array} \right] + \left[\begin{array}{ccc} c_i c_i & c_i c_j & c_i c_m \\ c_j c_i & c_j c_j & c_j c_m \\ c_m c_i & c_m c_j & c_m c_m \end{array} \right] \end{array} \right\}$$

Questi contributi vanno assemblati nella matrice di rigidezza globale H attraverso la matrice dei puntatori TRIJA, costruendo in tal modo il vettore reale SYSMAT dei termini non nulli della matrice di rigidezza. **Praticamente cosa conviene fare?**

Se, per ogni elemento IEL, individuiamo due vettori per i coefficienti b_i, b_j, b_m e per c_i, c_j, c_m (per esempio, BI(3, IEL), CI(3, IEL)) e se indichiamo con AREA(IEL) l'area dell'elemento IEL, possiamo scrivere l'algoritmo che segue:

```

001      Per iel=1,ne
002          Per k=1,3
003              knod=TRIANG(k,IEL)
004                  Per l=1,3
005                      lnod=TRIANG(l,IEL)
006                          Se lnod ≥ knod
007                              ind=TRIJA(k,l,iel)
008                                  SYSMAT(ind)=SYSMAT(ind) +
(BI(K,IEL)*BI(L,IEL)+ CI(K,IEL)*CI(L,IEL))/(4*AREA(IEL))
009                                      Fine Se
010                                          Fine Per
011                                              Fine Per
012                                                  Fine Per

```

Calcolo del termine noto

Il contributo locale per il termine noto viene da

$$f_i^{(e)} = \iint_{S^{(e)}} q \xi_i dS^{(e)} = \frac{1}{3} q \Delta$$

Globalmente, quindi, si ha:

$$f_i = \frac{1}{3} q \sum_e \Delta = \frac{4}{3} \sum_e \Delta$$

Questa sommatoria è estesa a tutti i triangoli che hanno un vertice nel nodo i -simo: $\frac{1}{3} \sum_e \Delta$ (dove la sommatoria è estesa a tutti i triangoli che condividono il nodo i) si chiama **area di afferenza** del nodo i -simo. Sarà utile predisporre un vettore contenente l'area di afferenza relativa a ciascun nodo della griglia di calcolo.

Imposizione delle condizioni al contorno

Se non imponiamo al problema le condizioni al contorno, esso ammette un'infinità di soluzioni. E difatti la matrice locale $H^{(e)}$ ha determinante nullo (cioè è singolare): la somma delle righe (o delle colonne) ha come risultato zero. Quindi anche la matrice globale è singolare.

Imponendo le condizioni al contorno di Dirichlet la matrice H diventa regolare (cioè con determinante diverso da zero).

Preassegniamo il valore della temperatura sui nodi di Dirichlet nella fase di soluzione del sistema globale.

Per far questo sarebbe necessario imporre 1 al termine diagonale di H corrispondente al nodo di Dirichlet, azzerare tutti gli elementi extradiagonali della riga e della colonna corrispondenti, ed uguagliare il termine noto al valore della condizione di Dirichlet: questa procedura è piuttosto complicata (data la simmetria e la memorizzazione in forma compatta).

Conviene sostituire l'elemento diagonale della matrice di rigidezza con un valore molto grande, ad esempio $R_{max} = 10^{25}$, ed uguagliare il termine noto corrispondente al valore di Dirichlet moltiplicato per R_{max} . Nel nostro caso, basterà moltiplicare per R_{max} l'elemento diagonale ed annullare il termine noto corrispondente (in quanto la condizione di Dirichlet vale zero). In questo modo la condizione al contorno sarebbe soddisfatta esattamente se R_{max} fosse ∞ ; in pratica, è sufficiente usare R_{max} circa 10 ordini di grandezza maggiore degli elementi di H .


```
001      Per  $k = 1, n_{dir}$   
002           $j = \text{nodedir}(k)$   
003           $\text{ind} = \text{IA}(j)$   
004           $f_j = \bar{T}(k) R_{max}$  (=0 nel nostro caso)  
005           $\text{SYSMAT}(\text{ind}) = R_{max}$   
006      Fine Per
```

Soluzione del sistema lineare

Una volta costruita la matrice H , poichè è sparsa, simmetrica, definita positiva, possiamo risolvere il sistema con il CGM.

Calcolo dell'errore

Possiamo confrontare la soluzione numerica che otteniamo dal nostro codice con la soluzione analitica del problema.

Per la soluzione analitica (poichè è costituita dalla somma di una serie e non possiamo fare la somma di infiniti termini), considereremo una somma parziale in modo da avere un sufficiente numero di cifre decimali corrette per poter fare il confronto con la soluzione numerica.

Se consideriamo la somma parziale della serie fino a $k = 8$ la soluzione presenta almeno quattro cifre decimali corrette (aumentando k le prime quattro cifre decimali restano invariate). Considerando invece, $k = 11$ sono corrette almeno le prime cinque cifre decimali. **Noi assumeremo come soluzione analitica quella ottenuta considerando la somma parziale della serie fino a $k = 11$.**

Data una certa griglia (e quindi un determinato numero di triangoli e di nodi) diremo errore puntuale sul nodo i -simo

$$err_i^{puntuale} = |\hat{T}_i - T_i|.$$

\hat{T}_i è la soluzione numerica, T_i la soluzione analitica nel nodo i -simo.

Definiamo poi **la norma euclidea dell'errore**:

$$\varepsilon = \left[\iint_{\mathcal{S}} (\hat{T} - T)^2 dS \right]^{1/2}$$

Numericamente:

$$\varepsilon = \left\{ \sum_{i=1}^n \left[(\hat{T}_i - T_i)^2 \sum_e \frac{\Delta}{3} \right] \right\}^{1/2}$$

Ritroviamo l'area di appartenenza al nodo.

Studio della convergenza

La norma euclidea dell'errore è utile non solo per verificare la bontà del risultato numerico ma anche per studiare la convergenza del metodo.

Lo studio della convergenza degli elementi finiti alla soluzione analitica consiste nel raffinare la mesh 3 o 4 volte e verificare il seguente importante risultato: la norma euclidea dell'errore diminuisce come ℓ^2 , dove ℓ è una dimensione rappresentativa della triangolazione (in pratica dell'ordine del lato di un qualsiasi triangolo).

Noi partiremo da una griglia molto grossolana: dividiamo il dominio in 2×2 parti (con il programma `griglia.f`) in modo da ottenere una griglia di 8 triangoli. Questa sarà la nostra mesh 1.

Poi dividiamo il dominio in 4×4 parti (sempre facendo girare il codice `griglia.f`) e otterremo 32 triangoli (mesh 2).

Ogni triangolo viene suddiviso in 4 triangoli quando raffiniamo la mesh.

Costruiremo, quindi almeno altre 2 mesh, con 128 e 512 triangoli (se si riesce a costruirne altre più fini, meglio!).

Con queste mesh noi avremo dei risultati da portare all'esame.

In particolare, insieme ad una relazione scritta, bisogna fare:

- **tabella di confronto** fra la soluzione numerica e quella analitica per la mesh 1 con il calcolo dell'errore puntuale;
- **diagrammi di convergenza** in scala semilogaritmica del GCM per la soluzione del sistema: si presenti un diagramma con tutti i profili di convergenza relativi alle mesh assegnate ottenuti partendo da $x_0 = K^{-1}f$ ed utilizzando entrambi i

precondizionatori studiati, $K^{-1} = D^{-1}$ e $K^{-1} = (\tilde{L}\tilde{L}^T)^{-1}$. Si ponga sull'asse delle ascisse il numero di iterazioni e sull'asse delle ordinate la norma euclidea del residuo relativo;

- **diagramma di convergenza** in scala semilogaritmica del GCM per la soluzione del sistema con 1, 2, 5, 10 e 20 iterazioni preliminari effettuate con lo schema delle Correzioni Residue: per questo test si utilizzi la mesh più fine usando $K^{-1} = (\tilde{L}\tilde{L}^T)^{-1}$;
- **tabella riassuntiva dei risultati** in cui siano riportati ad ogni raffinamento successivo della triangolazione: (a) il valore ε dell'errore, (b) il rapporto fra l'errore corrispondente alla griglia corrente ed alla griglia meno raffinata immediatamente precedente.
- **il listato completo** del codice di calcolo.