

Equazione di Poisson.

Alvise Sommariva

Università degli Studi di Padova
Dipartimento di Matematica

2 maggio 2020

Equazione di Poisson, caso bivariato.

Da

$$\Delta u(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j)}{h^2} \quad (1)$$

ricaviamo la discretizzazione dell'equazione

$$\begin{cases} \Delta u(x, y) = f(x, y), & (x, y) \in \Omega = (0, 1) \times (0, 1) \\ u(x, y) = g(x, y), & (x, y) \in \partial\Omega \end{cases} \quad (2)$$

nei punti (x_i, y_j) , $x_i = ih$, $y_j = jh$, $h=1/(n+1)$

$u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j) = h^2 f(x_i, y_j),$

(3)

per $i, j = 1, \dots, n$, con le condizioni al contorno

$$u(x_i, y_j) = g(x_i, y_j), \quad i = 0, j = 1, \dots, n \quad (4)$$

$$u(x_i, y_j) = g(x_i, y_j), \quad i = n+1, j = 1, \dots, n \quad (5)$$

$$u(x_i, y_j) = g(x_i, y_j), \quad i = 1, \dots, n, j = 0, j = n+1. \quad (6)$$

Equazione di Poisson, caso bivariato.

Il sistema lineare ottenuto può essere descritto matricialmente. Sia B la matrice $n \times n$

$$B = \begin{pmatrix} -4 & 1 & 0 & \dots & 0 \\ 1 & -4 & 1 & 0 & \dots \\ 0 & 1 & -4 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & -4 \end{pmatrix}$$

ed I la matrice identica di ordine n del tipo

$$I = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Equazione di Poisson, caso bivariato.

Allora

- se b è il vettore ottenuto dai contributi dei termini dovuti a f e g in (2) e (3),
- A la matrice a blocchi

$$A = \begin{pmatrix} B & I & 0 & \dots \\ I & B & I & \dots \\ \dots & \dots & \dots & \dots \\ \dots & 0 & I & B \end{pmatrix}$$

si ricava che il sistema da risolvere è $Au = b$, usando ad esempio

- il metodo di Jacobi, o
- Gauss-Seidel, SOR o
- il gradiente coniugato.

Equazione di Poisson, caso bivariato.

È possibile utilizzare la `gallery` di Matlab, per determinare la generica matrice di Poisson, composta da n blocchi, ognuno $n \times n$.
Ad esempio:

```
>> A=gallery('poisson',2);  
>> A=full(A)  
A =  
     4     -1     -1     0  
     -1     4     0     -1  
     -1     0     4     -1  
     0     -1     -1     4  
>> % NB: nei nostri esperimenti serve "-A".
```

Osserviamo che non è proprio facile determinare, fissato i, j , quali siano i punti adiacenti a (x_i, y_j) che essendo sul bordo hanno valore della soluzione noto a priori e quindi tali da contribuire attivamente al termine noto.

Equazione di Poisson: un esempio bivariato.

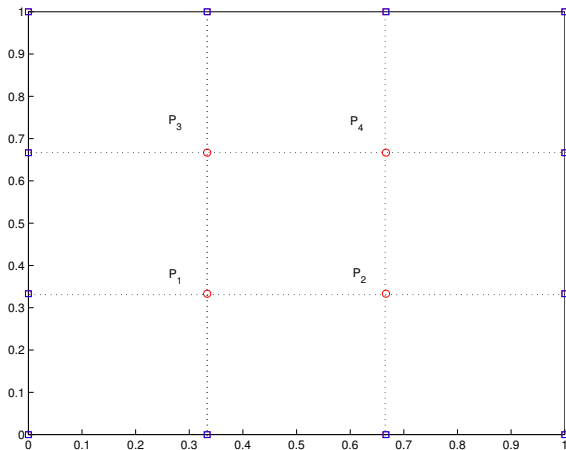


Figura: Una griglia avente passo $h = 1/3$ relativamente al quadrato $[0, 1] \times [0, 1]$.

Equazione di Poisson: un esempio bivariato.

Facciamo un esempio sulla risoluzione dell'equazione di Poisson via metodo alle differenze con 5 punti.

Sia $\Omega = [0, 1] \times [0, 1]$, $h = 1/3$ e siano

$$P_{i,j} = (ih, jh), \quad i, j = 0, 1, 2, 3.$$

E' chiaro che per

- per $i = 0$ i punti $P_{0,j}$ sono sull'asse $x = 0$ (cioè l'asse y),
- per $i = 3$ i punti $P_{3,j}$ sono sull'asse $x = 1$,
- per $j = 0$ i punti $P_{i,0}$ sono sull'asse $y = 0$ (cioè l'asse x)
- per $j = 3$ i punti $P_{i,3}$ sono sull'asse $y = 1$.

Date le condizioni al contorno, la soluzione in questi punti è nota ed è uguale a $u_{i,j} = g(x_i, y_j)$.

Equazione di Poisson: un esempio bivariato.

I rimanenti punti $P_{i,j}$, con $i, j = 1, 2$ sono interni a Ω ed è

$$u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j) = h^2 f(x_i, y_j), \quad (7)$$

Poniamo ora

$$b_1 := h^2 f(x_1, y_1) - g(x_0, y_1) - g(x_1, y_0),$$

$$b_2 := h^2 f(x_1, y_2) - g(x_0, y_2) - g(x_1, y_3),$$

$$b_3 := h^2 f(x_2, y_1) - g(x_3, y_1) - g(x_2, y_0),$$

$$b_4 := h^2 f(x_2, y_2) - g(x_3, y_2) - g(x_2, y_3),$$

ordiniamo i punti da sinistra a destra, e dal basso verso l'alto (ordine lessicografico)

$$P_1 = (x_1, y_1), P_2 = (x_2, y_1), P_3 = (x_1, y_2), P_4 = (x_2, y_2),$$

poniamo $u_1 = u(x_1, y_1)$, $u_2 = u(x_2, y_1)$, $u_3 = u(x_1, y_2)$, $u_4 = u(x_2, y_2)$, ricavando

$$u_2 + u_3 - 4u_1 = b_1,$$

$$u_1 + u_4 - 4u_2 = b_2,$$

$$u_4 + u_1 - 4u_3 = b_3,$$

$$u_3 + u_2 - 4u_4 = b_4,$$

da cui posto A la matrice di Poisson composta da 2 blocchi di ordine 2, basta risolvere il sistema $Au = b$ per ottenere $u(x_1, y_1)$, $u(x_2, y_1)$, $u(x_1, y_2)$, $u(x_2, y_2)$.

Equazione di Poisson: un esempio bivariato.

- Notiamo che la matrice calcolata dalla gallery è $-A$

```
>>> A=gallery('poisson',2);  
>>> A=full(A)  
A =  
     4     -1     -1     0  
    -1     4      0    -1  
    -1     0      4    -1  
     0     -1     -1     4  
>>>
```

- il termine noto b può essere facilmente determinato dopo aver notato che
 - 1 i termini $-g(x_i, y_0)$ sono presenti nelle componenti b_i ;
 - 2 posto $n = 1/h$, per $i = 1, \dots, n-1$, i termini $-g(x_i, y_n)$ sono presenti nelle componenti $b_{(n-1)^2 - (n-1) + i}$;
 - 3 per $j = 1, \dots, n-1$, i termini $-g(x_0, y_j)$ sono presenti nelle componenti b_s con $s \equiv 1, \text{ mod } n-1$;
 - 4 per $j = 1, \dots, n-1$, i termini $-g(x_n, y_j)$ sono presenti nelle componenti b_s con $s \equiv 0, \text{ mod } n-1$.

Equazione di Poisson: una implementazione.

Implementiamo il metodo sopra descritto, detto a 5 punti (cf. (3)).

Risulta importante ricordare la seguente stima dell'errore:

Teorema

Se

- 1** u è soluzione dell'equazione di Poisson (2) ed è almeno 4 volte differenziabile con continuità nel quadrato $\Omega := [0, 1] \times [0, 1]$,
- 2** u_h l'approssimazione ottenuta col metodo alle differenze con 5 punti, utilizzando una griglia $\mathcal{G} = \{(x_i, y_j)\}$ con $x_i = i h$, $y_j = j h$, $h = 1/(n+1)$

allora

$$|u(x_i, y_j) - u_h(x_i, y_j)| \leq c h^2$$

con

$$c = (1/24) \left(\max_{(x,y) \in \Omega} \left| \frac{\partial^4 u(x,y)}{\partial x^4} \right| + \max_{(x,y) \in \Omega} \left| \frac{\partial^4 u(x,y)}{\partial y^4} \right| \right)$$

Ci si aspetta quindi dai test numerici che effettueremo un errore dell'ordine di h^2 .

Il comando reshape

Un importante comando Matlab è reshape. Dall' help di Matlab

```
>> help reshape
reshape Reshape array.
  reshape(X,M,N) or reshape(X,[M,N]) returns the M-by-N matrix
  whose elements are taken columnwise from X. An error results
  if X does not have M*N elements.

  ... ..
>>
```

Il comando, da un vettore x di dimensione $m \cdot n \times 1$, produce una matrice di dimensione $m \times n$, i cui elementi sono tratti da x .

```
>> x=1:6'; % vettore colonna
>> reshape(x,2,3)
ans =
     1     3     5
     2     4     6
>> reshape(x,3,2)
ans =
     1     4
     2     5
     3     6
>>
```

Equazione di Poisson: una implementazione.

Salviamo in `poisson5pts.m` la funzione

```
function [x_grid,y_grid,u_grid]=poisson5pts(n,f,g_left,g_right,g_down,g_up)
A=gallery('poisson',n); A=full(A);
h=1/(n+1); x=(h:h:1-h)'; y=x;
% CONTRIBUTI DI "g".
x_loc=x; y_loc=zeros(size(x_loc)); b_down=feval(g_down,x_loc,y_loc); % BASSO
y_loc=ones(size(x_loc)); b_up=feval(g_up,x_loc,y_loc); % ALTO
y_loc=x_loc; x_loc=zeros(size(x_loc)); b_left=feval(g_left,x_loc,y_loc); % SX
x_loc=ones(size(x_loc)); b_right=feval(g_right,x_loc,y_loc); % DX
% COMPOSIZIONE TERMINE NOTO.
b1=b_down;
b1(1)=b1(1)+b_left(1); b1(n)=b1(n)+b_right(1); % PRIMA RIGA IN BASSO.
bn=b_up; bn(1)=bn(1)+b_left(n); bn(n)=bn(n)+b_right(n); % PRIMA RIGA IN ALTO.
% RIGHE INTERMEDIE.
bj=[];
for j=2:(n-1)
    bjloc=zeros(n,1);
    bjloc(1)=bjloc(1)+b_left(j);
    bjloc(n)=bjloc(n)+b_right(j);
    bj=[bj; bjloc];
end
b=[b1; bj; bn];
% GRIGLIA LESSICOGRAFICA (METODO STANDARD).
lunghezza_x=length(x); x_grid=[]; y_grid=[];
for index=1:lunghezza_x
    x_grid=[x_grid; x]; y_grid=[y_grid; y(index)*ones(size(x))];
end
fgrid=feval(f,x_grid,y_grid); b_f=(h^2)*fgrid; b=b_f-b; u_grid=A\b;
```

Equazione di Poisson: una implementazione.

Salviamo in demopoisson5pts.m la demo

```
function demopoisson5pts(demo_example)

switch demo_example
case 1
    f=@(x,y) zeros(size(x));
    g_down=@(x,y) ones(size(x)); g_up=@(x,y) ones(size(x));
    g_left=@(x,y) ones(size(x)); g_right=@(x,y) ones(size(x));
    solution=@(x,y) ones(size(x));
case 2
    f=@(x,y) zeros(size(x));
    g_down=@(x,y) exp(pi*x); g_up=@(x,y)-exp(pi*x);
    g_left=@(x,y) cos(pi*y); g_right=@(x,y) ((exp(1))^pi)*cos(pi*y);
    solution=@(x,y) (exp(pi*x)).*cos(pi*y);
case 3
    f=@(x,y) (-2*(pi^2))*sin(pi*x).*sin(pi*y);
    g_down=@(x,y) zeros(size(x)); g_up=@(x,y) zeros(size(x));
    g_left=@(x,y) zeros(size(x)); g_right=@(x,y) zeros(size(x));
    solution=@(x,y) (sin(pi*x)).*sin(pi*y);
otherwise
    f=@(x,y) ones(size(x));
    g_down=@(x,y) zeros(size(x)); g_up=@(x,y) zeros(size(x));
    g_left=@(x,y) zeros(size(x)); g_right=@(x,y) zeros(size(x));
    solution=@(x,y) (sin(pi*x)).*sin(pi*y);
end
```

Equazione di Poisson: una implementazione.

```
for index=2:5
    n=2^index;
    [x_grid,y_grid,u_grid]=poisson5pts(n,f,g_left,g_right,g_down,g_up);

    if demo_example <=3
        sol_grid=fval(solution,x_grid,y_grid);
        err(index)=max(abs(sol_grid-u_grid));
        if index == 1
            fprintf('\n \t [n]: %4.0f [ERR]: %2.2e',n,err(index));
        else
            fprintf('\n \t [n]: %4.0f [ERR]: %2.2e [RATIO]: %2.2f',...
                n,err(index),err(index-1)/err(index));
        end
    end
end

% trasformazione da vettore a griglia "immagazzinata" come matrice
X=reshape(x_grid,n,n); Y=reshape(y_grid,n,n); Z=reshape(u_grid,n,n);

% plot soluzione
surf(X,Y,Z);

fprintf('\n \n');
```

Equazione di Poisson: esempio 1.

Consideriamo l'equazione di Poisson nel quadrato unitario $\Omega = [0, 1] \times [0, 1]$

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, & (x, y) \in \Omega \\ u(x, y) = 1, & (x, y) \in \partial\Omega \end{cases} \quad (8)$$

la cui soluzione è $u(x, y) = 1$.

Non è difficile osservare che

$$\max_{(x,y) \in \Omega} \left| \frac{\partial^4 u(x, y)}{\partial x^4} \right| = 0, \quad \max_{(x,y) \in \Omega} \left| \frac{\partial^4 u(x, y)}{\partial y^4} \right| = 0$$

e quindi ci si aspetta che per qualsiasi h si abbia un errore dell'ordine della precisione di macchina.

Equazione di Poisson: esempio 1.

Lanciamo da shell il primo esempio della demo, ottenendo

```
>> demopoisson5pts(1)
[n]:      4 [ERR]: 3.33e-16 [RATIO]: 0.00
[n]:      8 [ERR]: 6.66e-16 [RATIO]: 0.50
[n]:     16 [ERR]: 1.78e-15 [RATIO]: 0.38
[n]:     32 [ERR]: 2.78e-15 [RATIO]: 0.64
>>
```

Per *ratio* si intende il rapporto dell'errore tra due iterate successive.

Equazione di Poisson: esempio 2.

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (x, y) \in \Omega \\ u(x, 0) = \exp(\pi x), \quad x \in [0, 1] \\ u(x, 1) = -\exp(\pi x), \quad x \in [0, 1] \\ u(0, y) = \cos(\pi y), \quad y \in [0, 1] \\ u(1, y) = \exp(\pi) \cdot \cos(\pi y), \quad y \in [0, 1] \end{array} \right. \quad (9)$$

la cui soluzione è $u(x, y) = \exp(\pi x) \cdot \cos(\pi y)$.

Equazione di Poisson: esempio 2.

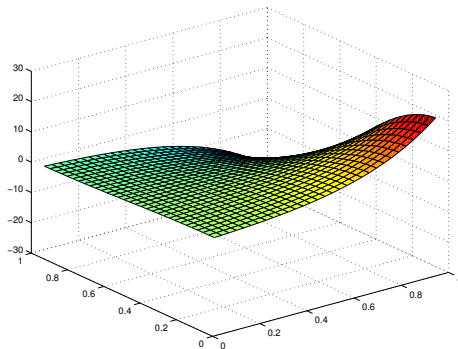


Figura: Soluzione del problema 2.

Equazione di Poisson: esempio 2.

Si nota subito che per $x, y \in [0, 1]$ si ha

$$\left| \frac{\partial^4 u}{\partial x^4}(x, y) \right| = \pi^4 |\exp(\pi x) \cdot \cos(\pi y)| \leq \pi^4 \exp(\pi) \approx 2254.1$$

$$\left| \frac{\partial^4 u}{\partial y^4}(x, y) \right| = \pi^4 |\exp(\pi x) \cdot \cos(\pi y)| \leq \pi^4 \exp(\pi) \approx 2254.1$$

da cui

$$c \leq (1/24) \cdot 2254.1 \cdot 2 \approx 187.8428.$$

Quindi quale maggiorazione dell'errore assoluto in norma infinito, per $n = 3, 7, 15, 31$, avremo i valori immagazzinati qui sotto nel vettore `err`

Equazione di Poisson: esempio 2.

```
>> format short e
>> c=187.8428; err=[];
>> for n=2:5, N=2^n-1; h=1/(N+1); h2=h^2; err=[err; c*h2]; end
>> err
err =
    1.1740e+001
    2.9350e+000
    7.3376e-001
    1.8344e-001
>>
```

Lanciando la demo demopoisson5pts con parametro 2, abbiamo

```
>> demopoisson5pts(2)
[n]:      4 [ERR]: 9.75e-02 [RATIO]: 0.00
[n]:      8 [ERR]: 3.20e-02 [RATIO]: 3.04
[n]:     16 [ERR]: 9.05e-03 [RATIO]: 3.54
[n]:     32 [ERR]: 2.45e-03 [RATIO]: 3.69
>>
```

Equazione di Poisson: esempio 2.

- 1 Come ci si aspettava la maggiorazione è realizzata, ma purtroppo come stima è abbastanza conservativa.
- 2 Nella colonna [RATIO] abbiamo indicato il rapporto e_{2h}/e_h dove e_h è l'errore assoluto compiuto dal metodo a 5 punti con passo h (ovvero la quantità esposte nella colonna [ERR] nella stessa riga di h). Il fatto che la *ratio* sia 4 non è sorprendente. Infatti se l'errore decresce come h^2 si può supporre che sia $e_h \approx \hat{c}h^2$ per qualche $\hat{c}$ indipendente da h e quindi

$$\frac{e_{2h}}{e_h} \approx \frac{\hat{c}(2h)^2}{\hat{c}h^2} \approx 4.$$

Equazione di Poisson: esempio 3.

Consideriamo un la risoluzione dell'equazione di Poisson nel quadrato unitario $\Omega = [0, 1] \times [0, 1]$ con un metodo alle differenze.

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (-2\pi^2) \sin(\pi x) \sin(\pi y), \quad (x, y) \in \Omega \\ u(x, 0) = 0, \quad x \in [0, 1] \\ u(x, 1) = 0, \quad x \in [0, 1] \\ u(0, y) = 0, \quad y \in [0, 1] \\ u(1, y) = 0, \quad y \in [0, 1] \end{array} \right. \quad (10)$$

Equazione di Poisson: esempio 3.

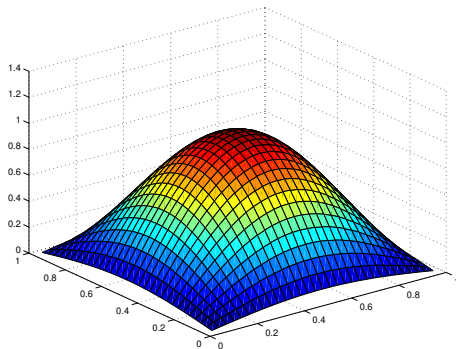


Figura: Soluzione del problema 3.

Equazione di Poisson: esempio 3.

la cui soluzione è $u(x, y) = \sin(\pi x) \sin(\pi y)$. Ripetendo la stima basata sulle derivate quarte della soluzione u abbiamo facilmente

$$c \leq (1/24) 2 \pi^4 \approx 8.1174$$

da cui

```
>> c=8.1174;
>> err=[]; for n=2:5, N=2^n; N=N-1; h=1/(N+1); h2=h^2; err=[err; c*h2]; end
>> format short e
>> err
err =
  5.0734e-001
  1.2683e-001
  3.1709e-002
  7.9271e-003
>>
```

Equazione di Poisson: esempio 3.

Lanciando la demo `demopoisson5pts` con parametro 3, abbiamo

```
>> demopoisson5pts(3)
[n]:    4 [ERR]: 3.04e-002 [RATIO]: 2.39
[n]:    8 [ERR]: 9.91e-003 [RATIO]: 3.06
[n]:   16 [ERR]: 2.83e-003 [RATIO]: 3.51
[n]:   32 [ERR]: 7.54e-004 [RATIO]: 3.75
>>
```

Rispetto al caso precedente la stima è più precisa, e la ratio di circa 4 ci dice che la convergenza è ancora dell'ordine di h^2 .

Bibliografia



V. Comincioli, *Analisi Numerica, metodi modelli applicazioni*, Mc Graw-Hill, 1990.