

Equazione di Poisson

30 giugno 2007

1 Alcuni cenni sulle equazioni di Laplace e Poisson.

Si definisce equazione alle derivate parziali *quasi-lineare* il problema

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} = f(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}), \quad (x, y) \in \Omega \quad (1)$$

dove A, B e C sono costanti.

In particolare l'equazione è

- *ellittica* se $B^2 - 4AC < 0$;
- *parabolica* se $B^2 - 4AC = 0$;
- *iperbolica* se $B^2 - 4AC > 0$.

Passiamo a considerare qualche esempio notevole. Definito il dominio Ω , si consideri l'equazione di Poisson [1, p.557]

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

Nel caso particolare $g \equiv 0$, l'equazione alle derivate parziali (16), è nota come equazione di Laplace. La particolare condizione al bordo è detta di *Dirichlet*.

E' facile osservare che (16) è di tipo quasi-lineare con $A = 1, B = 0, C = 1$ e quindi ellittica.

2 Discretizzazione del Laplaciano.

Per studiare questo problema facciamo alcune osservazioni preliminari. Supponendo che $s: [a, b] \rightarrow \mathbb{R}$ sia sufficientemente regolare, dalla formula di Taylor si ha

$$s(x+h) = s(x) + hs^{(1)}(x) + \frac{h^2 s^{(2)}(x)}{2} + \frac{h^3 s^{(3)}(x)}{6} + \frac{h^4 s^{(4)}(\xi)}{24} \quad (3)$$

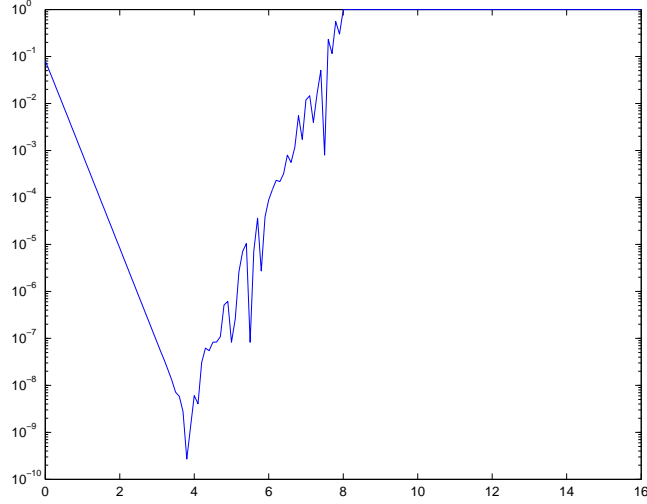


Figure 1: Grafico che illustra in scala semilogaritmica l'errore assoluto compiuto dalla formula (5) per il calcolo della derivata seconda del coseno nel punto $x = \pi$, utilizzando valori di $h = 10^{-s}$ dove s è il valore indicato nelle ascisse.

con $\xi \in \mathcal{J}(x, x+h)$ (dove $\mathcal{J}(a, b)$ è il più piccolo intervallo aperto contenente a, b). Similmente a (3) si ha quindi per $e \in \mathcal{J}(x, x-h)$

$$s(x-h) = s(x) - hs^{(1)}(x) + \frac{h^2 s^{(2)}(x)}{2} - \frac{h^3 s^{(3)}(x)}{6} + \frac{h^4 s^{(4)}(e)}{24}. \quad (4)$$

Sommando membro a membro (3), (4) abbiamo

$$s^{(2)}(x) = \frac{s(x+h) - 2s(x) + s(x-h)}{h^2} + O(h^2) \quad (5)$$

$$\approx \frac{s(x+h) - 2s(x) + s(x-h)}{h^2} \quad (6)$$

Consideriamo la funzione

$$s(x) = \cos(x)$$

e valutiamo

$$s^{(2)}(x) = -\cos(x)$$

per $x = \pi$, qualora si usino alcuni valori di $h = 10^{-s}$ con $s = 0, 0.1, 0.2, \dots, 16$. Noto che $s^{(2)}(\pi) = -\cos(\pi) = 1$ valuteremo l'errore assoluto (e al tempo stesso relativo) fornito dalla formula (5).

Un possibile codice Matlab `diff2experiments.m` è il seguente

```
s=0:0.1:16;
x=pi;
```

```
exactvalue=1;

for index=1:length(s)
    h_exp=s(index);
    h=10^(-h_exp);
    fxplus=f(x+h);
    fx=f(x);
    fxminus=f(x-h);
    f2approx=(fxplus-2*fx+fxminus)/(h^2);
    abserr(index)=abs(f2approx-exactvalue);
    fprintf('\n \t [h]: %2.2e [ABS. ERR.]: %2.2e',h,abserr(index))
end

semilogy(h_vett,abserr);
```

in cui `f.m` è data da

```
function fx=f(x)
fx=cos(x);
```

Si ottengono i seguenti risultati

```
[h]: 1.00e+000 [ABS. ERR.]: 8.06e-002
[h]: 7.94e-001 [ABS. ERR.]: 5.15e-002
[h]: 6.31e-001 [ABS. ERR.]: 3.27e-002
[h]: 5.01e-001 [ABS. ERR.]: 2.08e-002
[h]: 3.98e-001 [ABS. ERR.]: 1.31e-002
[h]: 3.16e-001 [ABS. ERR.]: 8.31e-003
...
[h]: 5.01e-004 [ABS. ERR.]: 2.09e-008
[h]: 3.98e-004 [ABS. ERR.]: 1.27e-008
[h]: 3.16e-004 [ABS. ERR.]: 7.19e-009
[h]: 2.51e-004 [ABS. ERR.]: 5.80e-009
[h]: 2.00e-004 [ABS. ERR.]: 2.77e-009
[h]: 1.58e-004 [ABS. ERR.]: 2.70e-010
[h]: 1.26e-004 [ABS. ERR.]: 1.26e-009
[h]: 1.00e-004 [ABS. ERR.]: 6.08e-009
[h]: 7.94e-005 [ABS. ERR.]: 4.04e-009
[h]: 6.31e-005 [ABS. ERR.]: 3.07e-008
[h]: 5.01e-005 [ABS. ERR.]: 6.21e-008
[h]: 3.98e-005 [ABS. ERR.]: 5.48e-008
[h]: 3.16e-005 [ABS. ERR.]: 8.27e-008
[h]: 2.51e-005 [ABS. ERR.]: 8.39e-008
[h]: 2.00e-005 [ABS. ERR.]: 1.09e-007
```

che dicono che la formula per il calcolo della derivata seconda è da usarsi con cautela, in quanto fornisce risultati accurati per h non eccessivamente piccolo altrimenti errori di arrotondamento *degradano* la convergenza della formula.

Quale esercizio si provi il caso in cui $f(x) = \exp(x)$, $x = 0$ (e quindi $f^{(2)}(0) = 1$). Si faccia attenzione a modificare correttamente la funzione f e le variabili x , `exactvalue`.

3 Risoluzione di un'equazione di Poisson.

Siamo pronti ad introdurre una *discretizzazione* dell'operatore di Laplace

$$\Delta u := u_{xx} + u_{yy} := \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

Dalla formula unidimensionale abbiamo

$$u_{xx}(x, y) = \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + O(h^2), \quad (7)$$

e

$$u_{yy}(x, y) = \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} + O(h^2), \quad (8)$$

da cui

$$\begin{aligned} \Delta u(x, y) &= \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} \\ &+ \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + O(h^2) \\ &\approx \frac{u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h) - 4u(x, y)}{h^2}. \end{aligned} \quad (9)$$

Facciamo una nuova verifica numerica. Sia

$$g(x, y) = \cos(x) \exp(2y)$$

e supponiamo di voler valutare $\Delta g(0, 0)$. E' facile osservare che

$$\Delta g(x, y) = -\cos(x) \exp(2y) + 4 \cos(x) \exp(2y)$$

e che quindi

$$\Delta g(0, 0) = -1 \cdot 1 + 4 \cdot 1 \cdot 1 = 3.$$

Scriviamo quindi le 2 funzioni Matlab/Octave `diff2Dexperiments.m` e `g.m` che ci permettono di sperimentare quanto descritto:

```
s=0:0.1:16;
x=0; y=0;

correct_value=3;
```

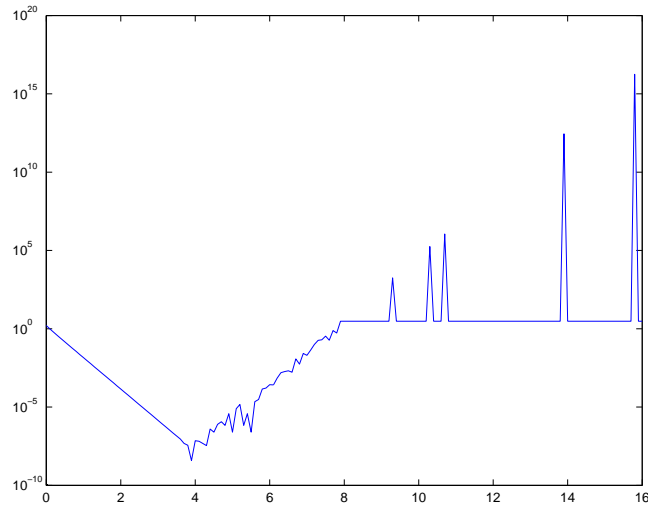


Figure 2: Grafico che illustra in scala semilogaritmica l'errore assoluto compiuto dalla formula (9) per il calcolo del laplaciano della funzione $g(x, y)$ nel punto $(0, 0)$, utilizzando valori di $h = 10^{-s}$ dove s è il valore indicato nelle ascisse.

```

for index=1:length(s)
    h_exp=s(index);
    h=10^(-h_exp);
    g2approx=( g(x+h,y)+g(x-h,y)+g(x,y+h)+g(x,y-h)-4*g(x,y) )/(h^2);
    abserr(index)=abs(g2approx-correct_value);
    fprintf('\n \t [h]: %2.2e [ABS.ERR.]: %2.2e',h,abserr(index))
end

semilogy(s,abserr);

function gxy=g(x,y)
gxy=cos(x)*exp(2*y);

```

I risultati numerici sono i seguenti

```

[h]: 1.00e+000 [ABS.ERR.]: 1.60e+000
[h]: 7.94e-001 [ABS.ERR.]: 9.67e-001
[h]: 6.31e-001 [ABS.ERR.]: 5.93e-001
[h]: 5.01e-001 [ABS.ERR.]: 3.67e-001
[h]: 3.98e-001 [ABS.ERR.]: 2.29e-001
[h]: 3.16e-001 [ABS.ERR.]: 1.43e-001

```

```

...
[h]: 5.01e-004 [ABS.ERR.]: 3.54e-007
[h]: 3.98e-004 [ABS.ERR.]: 2.22e-007
[h]: 3.16e-004 [ABS.ERR.]: 1.42e-007
[h]: 2.51e-004 [ABS.ERR.]: 8.99e-008
[h]: 2.00e-004 [ABS.ERR.]: 4.75e-008
[h]: 1.58e-004 [ABS.ERR.]: 3.62e-008
[h]: 1.26e-004 [ABS.ERR.]: 3.79e-009
[h]: 1.00e-004 [ABS.ERR.]: 7.06e-008
[h]: 7.94e-005 [ABS.ERR.]: 6.49e-008
[h]: 6.31e-005 [ABS.ERR.]: 4.75e-008
[h]: 5.01e-005 [ABS.ERR.]: 3.45e-008
[h]: 3.98e-005 [ABS.ERR.]: 3.96e-007
[h]: 3.16e-005 [ABS.ERR.]: 2.48e-007
...
[h]: 2.51e-008 [ABS.ERR.]: 1.85e-001
[h]: 2.00e-008 [ABS.ERR.]: 7.69e-001
[h]: 1.58e-008 [ABS.ERR.]: 5.36e-001
...

```

che suggerisce nuovamente che l'utilizzo di tale discretizzazione richiede che il passo h non sia eccessivamente piccolo.

Consideriamo l'equazione (cf. [1, p.557])

$$\begin{cases} \Delta u(x, y) = f(x, y), & (x, y) \in \Omega = (0, 1) \times (0, 1) \\ u(x, y) = g(x, y), & (x, y) \in \partial\Omega \end{cases} \quad (10)$$

Definita la griglia di punti $\mathcal{G} = \{(x_i, y_j)\}_{i,j=0,\dots,n+1}$

$$x_i = ih, y_j = jh, h = 1/(n+1), i, j = 0, \dots, n+1$$

risulta evidente che per $i = 0$ o $j = 0$, $i = n+1$ o $j = n+1$ abbiamo un punto del bordo e quindi in virtù delle condizioni di Dirichlet in (10), il valore di u^* è determinato. Vediamo cosa succede quando il punto della griglia \mathcal{G} è interno al quadrato $\Omega = (0, 1) \times (0, 1)$, cioè nel caso in cui (x_i, y_j) sia tale che nessuno tra gli indici i, j sia uguale a 0 oppure $n+1$.

Dalla discretizzazione dell'operatore di Laplace Δ , tralasciando i termini $O(h^2)$ abbiamo da (9)

$$\Delta u(x_i, y_j) = \frac{u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j)}{h^2} \quad (11)$$

e quindi l'equazione di Poisson, diventa

$$u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j) = h^2 f(x_i, y_j), \quad (12)$$

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

$$u(x_i, y_j) = g(x_i, y_j), \quad i = 0, j = 1, \dots, n \quad (13)$$

$$u(x_i, y_j) = g(x_i, y_j), \quad i = n+1, j = 1, \dots, n \quad (14)$$

$$u(x_i, y_j) = g(x_i, y_j), \quad i = 1, \dots, n, j = 0, j = n+1 \quad (15)$$

Purtroppo, la descrizione del sistema lineare non è troppo chiara. Vediamola scritta 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

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

Per quanto visto in (12), la matrice B è legata alla discretizzazione di $-\Delta$ e di questo fatto bisognerà tenerne conto in seguito.

Allora se b è il vettore ottenuto dai contributi dei termini dovuti a f e g in (10) e (12), definita la matrice a blocchi

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

si ricava che il sistema da risolvere è $-Au = h^2 b$. Il segno $-$ è legato alla discretizzazione di $-\Delta$ con A .

Per una implementazione della matrice di Poisson A , utilizziamo la funzione `makefish`.

4 Sull'equazione di Poisson. Il metodo alle differenze con 5 punti.

Vediamo ora un'implementazione del metodo sopra descritto, detto per ovvi motivi a 5 punti (cf. (12)).

Risulta importante ricordare che in [1, p. 558], si cita la seguente stima dell'errore, per soluzioni u dell'equazione di Poisson (10) almeno 4 volte differenziabili con continuità; se u è la soluzione esatta del problema definito nel quadrato $\Omega := [0, 1] \times [0, 1]$ ed u_h quella ottenuta 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 .

Salviamo in `poisson5pts.m` la funzione

```
function Z=poisson5pts(n,f,g_left,g_right,g_down,g_up)

% ATKINSON. P. 559.
A=makefish(n);

h=1/(n+1);

x=(h:h:1-h)'; y=x;
b_down=feval(g_down,x,y);
b_up=feval(g_up,x,y);

y=x;
b_left=feval(g_left,x,y);
b_right=feval(g_right,x,y);

b1=b_down; b1(1)=b1(1)+b_left(1); b1(n)=b1(n)+b_right(1);

bn=b_up; bn(1)=bn(1)+b_left(n); bn(n)=bn(n)+b_right(n);

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];

[X,Y]=meshgrid(x,y);
fXY=feval(f,X,Y);

b_f=(-h^2)*fXY;
b_f=b_f';
b_f=b_f(:);

b=b+b_f;

u=A\b;
Z=(reshape(u',n,n))';

e in demopoison5pts.m la demo

demoexample=1;

switch demoexample
case 1
    f=inline('zeros(size(x))','x','y');
    g_down=inline('ones(size(x))','x','y');
```



```
g_up=inline('ones(size(x))','x','y');
g_left=inline('ones(size(x))','x','y');
g_right=inline('ones(size(x))','x','y');

solution=inline('ones(size(x))','x','y');

case 2
f=inline('zeros(size(x))','x','y');
g_down=inline('exp(pi*x)','x','y');
g_up=inline('-exp(pi*x)','x','y');
g_left=inline('cos(pi*y)','x','y');
g_right=inline('((exp(1))^pi)*cos(pi*y)','x','y');

solution=inline('(exp(pi*x)).*cos(pi*y)','x','y');

case 3
f=inline('(-2*(pi^2))*sin(pi*x).*sin(pi*y)','x','y');
g_down=inline('zeros(size(x))','x','y');
g_up=inline('zeros(size(x))','x','y');
g_left=inline('zeros(size(x))','x','y');
g_right=inline('zeros(size(x))','x','y');

solution=inline('(sin(pi*x)).*sin(pi*y)','x','y');

otherwise
f=inline('ones(size(x))','x','y');
g_down=inline('zeros(size(x))','x','y');
g_up=inline('zeros(size(x))','x','y');
g_left=inline('zeros(size(x))','x','y');
g_right=inline('zeros(size(x))','x','y');

solution=inline('(sin(pi*x)).*sin(pi*y)','x','y');
end

for index=1:4
n=2^(index+1); n=n-1;

h=1/(n+1); x=(h:h:1-h)'; y=x;
[X,Y]=meshgrid(x,y);
Z=poisson5pts(n,f,g_left,g_right,g_down,g_up);

if demoexample <=3

V=feval(solution,X,Y);

err(index)=norm(V(:)-Z(:),inf);
if index == 1
fprintf('\n \t [n]: %4.0f [ERR]: %2.2e [h]: %2.2e',n,err(index),h);
else
fprintf('\n \t [n]: %4.0f [ERR]: %2.2e',n,err(index));
```

```

        fprintf(' [RATIO]: %2.2f [h]: %2.2e',err(index-1)/err(index),h);
    end
end
end

surf(X,Y,Z);

```

Alcune osservazioni:

1. Se la matrice $B \in \mathbb{R}^{n \times n}$ allora il termine noto b e il vettore soluzione u appartengono a \mathbb{R}^{n^2} . Posto $h = \frac{1}{n+1}$, dobbiamo risolvere un sistema lineare $-A = h^2 b$ con

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

e b un vettore i cui contributi dipendono dai valori che hanno sul bordo le funzioni f e g che definiscono l'equazione di Poisson

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), & (x, y) \in \Omega \\ u(x, y) = g(x, y), & (x, y) \in \partial\Omega \end{cases} \quad (16)$$

2. La funzione `meshgrid` crea a partire da un vettore x , le ascisse X e le ordinate Y dei punti facenti parte della griglia *generata* da x . Nel caso in esempio posti $x_1 = 0$, $x_2 = 0.5$, $x_3 = 1$, $y_1 = 0$, $y_2 = 0.5$, $y_3 = 1$, genera la griglia

```

(x1,y1) (x2,y1) (x3,y1)
(x1,y2) (x2,y2) (x3,y2)
(x1,y3) (x2,y3) (x3,y3)

```

Così

```

>> x=0:0.5:1;
>> y=x;
>> [X,Y]=meshgrid(x,y)

X =

    0    0.5000    1.0000
    0    0.5000    1.0000
    0    0.5000    1.0000

Y =

    0         0         0

```

```
0.5000    0.5000    0.5000
1.0000    1.0000    1.0000
```

```
>>
```

descrivendone le coordinate X , Y .

Per ulteriori chiarimenti usiamo l'help di Matlab 6.1 ottenendo

```
>> help meshgrid
```

```
MESHGRID    X and Y arrays for 3-D plots.
[X,Y] = MESHGRID(x,y) transforms the domain specified by vectors
x and y into arrays X and Y that can be used for the evaluation
of functions of two variables and 3-D surface plots.
The rows of the output array X are copies of the vector x and
the columns of the output array Y are copies of the vector y.
```

```
[X,Y] = MESHGRID(x) is an abbreviation for [X,Y] = MESHGRID(x,x).
[X,Y,Z] = MESHGRID(x,y,z) produces 3-D arrays that can be used to
evaluate functions of three variables and 3-D volumetric plots.
```

```
For example, to evaluate the function  x*exp(-x^2-y^2) over the
range  -2 < x < 2,  -2 < y < 2,
```

```
[X,Y] = meshgrid(-2:.2:2, -2:.2:2);
Z = X .* exp(-X.^2 - Y.^2);
mesh(Z)
```

```
MESHGRID is like NDGRID except that the order of the first two input
and output arguments are switched (i.e., [X,Y,Z] = MESHGRID(x,y,z)
produces the same result as [Y,X,Z] = NDGRID(y,x,z)). Because of
this, MESHGRID is better suited to problems in cartesian space,
while NDGRID is better suited to N-D problems that aren't spatially
based. MESHGRID is also limited to 2-D or 3-D.
```

```
See also SURF, SLICE, NDGRID.
```

3. Notiamo l'uso in `poisson5pts` di `reshape`, affinché la soluzione abbia una struttura uguale a quella fornita da `meshgrid`.

4.1 Problema 1.

Consideriamo un semplice esempio che illustra la risoluzione dell'equazione di Poisson con un metodo alle differenze. Come in [2, p. 284], consideriamo l'equazione di Poisson nel quadrato unitario $\Omega = [0, 1] \times [0, 1]$

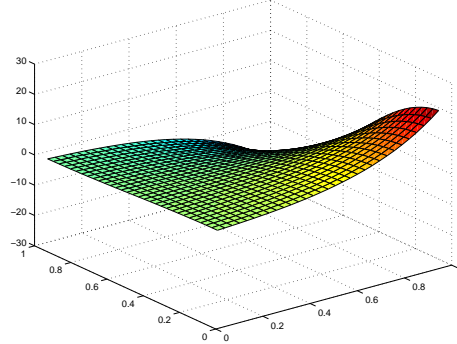


Figure 3: Soluzione del problema 2.

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, & (x, y) \in \Omega \\ u(x, y) = 1, & (x, y) \in \partial\Omega \end{cases} \quad (17)$$

la cui soluzione è $u(x, y) = 1$.

Non è difficile osservare che

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

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

Lanciamo da shell il primo esempio della demo, ottenendo

```
>> demopoisson5pts
```

```
[n]:    4 [ERR]: 3.33e-016 [RATIO]: 0.00
[n]:    8 [ERR]: 5.55e-016 [RATIO]: 0.60
[n]:   16 [ERR]: 9.99e-016 [RATIO]: 0.56
[n]:   32 [ERR]: 4.22e-015 [RATIO]: 0.24
```

```
>>
```

4.2 Problema 2.

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

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, & (x, y) \in \Omega \\ u(x, 0) = \exp(\pi x), & x \in [0, 1] \\ u(x, 1) = -\exp(\pi x), & x \in [0, 1] \\ u(0, y) = \cos(\pi y), & y \in [0, 1] \\ u(1, y) = \exp(\pi) \cdot \cos(\pi y), & y \in [0, 1] \end{cases} \quad (18)$$

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

```
>> 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`, per `demoexample=2`, abbiamo

```
>> demopoisson5pts

[n]:      3 [ERR]: 1.44e-001 [h]: 2.50e-001
[n]:      7 [ERR]: 3.90e-002 [RATIO]: 3.69 [h]: 1.25e-001
[n]:     15 [ERR]: 1.02e-002 [RATIO]: 3.82 [h]: 6.25e-002
[n]:     31 [ERR]: 2.60e-003 [RATIO]: 3.92 [h]: 3.13e-002

>>
```

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

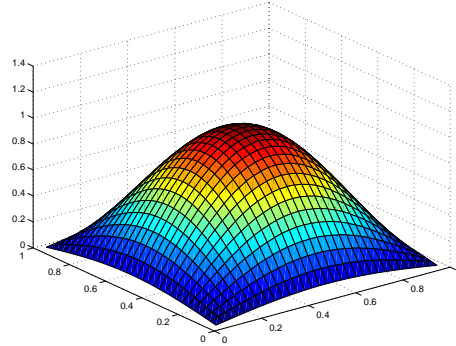


Figure 4: Soluzione del problema 3.

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

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

4.3 Problema 3.

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

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (-2\pi^2) \sin(\pi x) \sin(\pi y), & (x, y) \in \Omega \\ u(x, 0) = 0, & x \in [0, 1] \\ u(x, 1) = 0, & x \in [0, 1] \\ u(0, y) = 0, & y \in [0, 1] \\ u(1, y) = 0, & y \in [0, 1] \end{cases} \quad (19)$$

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

>>

Lanciando la demo demopoisson5pts, per demoexample=3, abbiamo

>> demopoisson5pts

```
[n]:    3 [ERR]: 5.30e-002 [h]: 2.50e-001
[n]:    7 [ERR]: 1.30e-002 [RATIO]: 4.09 [h]: 1.25e-001
[n]:   15 [ERR]: 3.22e-003 [RATIO]: 4.02 [h]: 6.25e-002
[n]:   31 [ERR]: 8.04e-004 [RATIO]: 4.01 [h]: 3.13e-002
```

>>

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 .

References

- [1] K. Atkinson, *Introduction to Numerical Analysis*, Wiley, 1989.
- [2] D. Bini, M. Capovani e O. Menchi, *Metodi numerici per l'algebra lineare*, Zanichelli, 1988.
- [3] V. Comincioli, *Analisi Numerica, metodi modelli applicazioni*, Mc Graw-Hill, 1990.
- [4] S.D. Conte e C. de Boor, *Elementary Numerical Analysis, 3rd Edition*, Mc Graw-Hill, 1980.
- [5] The MathWorks Inc., *Numerical Computing with Matlab*, <http://www.mathworks.com/moler>.
- [6] A. Quarteroni e F. Saleri, *Introduzione al calcolo scientifico*, Springer Verlag, 2006.
- [7] A. Suli e D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.