

Equazione del calore.

Alvise Sommariva

Università degli Studi di Padova
Dipartimento di Matematica

10 maggio 2013

Consideriamo l'*equazione del calore*

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + G, & 0 < x < 1, t > 0 \\ u(0, t) = d_0(t), u(1, t) = d_1(t), & t \geq 0 \\ u(x, 0) = f(x), & 0 \leq x \leq 1 \end{cases} \quad (1)$$

Sia $m > 0$ intero e sia $h_x = 1/m$ ed $x_j = jh_x$ con $j = 0, 1, \dots, m$.

Si può mostrare che per $j = 1, 2, \dots, m-1$ e $\xi_j \in (x_{j-1}, x_{j+1})$

$$\frac{\partial^2 u}{\partial x^2}(x_j, t) = \frac{u(x_{j+1}, t) - 2u(x_j, t) + u(x_{j-1}, t))}{h_x^2} - \frac{h_x^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_j, t) \quad (2)$$

Equazione del calore.

Tralasciando il termine finale e posto $u_j(t) := u(x_j, t)$ otteniamo quindi per $j = 1, \dots, m-1$ il sistema di equazioni differenziali

$$u_j'(t) = \frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{h_x^2} + G(x_j, t) \quad (3)$$

Risolto (3), si avrà una approssimazione della soluzione dell'equazione del calore per $x_j = jh_x$ e $t \geq 0$. Il procedimento appena descritto è noto in letteratura come *metodo delle linee*. Nel risolvere il sistema dobbiamo far attenzione alle condizioni sul bordo

$$u_0(t) = d_0(t), \quad u_m(t) = d_1(t)$$

e ricordare che la condizione iniziale del sistema di equazioni differenziali è

$$u_j(0) = f(x_j), \quad j = 1, \dots, m-1.$$

Equazione del calore.

Il sistema differenziale (3) può essere riscritto matricialmente.

Posto

$$\mathbf{u}(t) := [u_1(t), \dots, u_{m-1}(t)]^T$$

$$\mathbf{u}_0(t) := [f(x_1), \dots, f(x_{m-1})]^T$$

$$\mathbf{g}(t) := \left[\frac{1}{h_x^2} d_0(t), \dots, \frac{1}{h_x^2} d_1(t) \right]^T + [G(x_1, t), 0, \dots, 0, G(x_{m-1}, t)]^T$$

$$\Lambda = \frac{1}{h_x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \quad (4)$$

otteniamo che (3) è equivalente al sistema di equazioni differenziali (lineari)

$$\mathbf{u}'(t) = \Lambda \mathbf{u}(t) + \mathbf{g}(t), \quad \mathbf{u}(0) = \mathbf{u}_0(t) \quad (5)$$

Tra i metodi più comuni nel risolvere il problema differenziale (di Cauchy)

$$\begin{cases} \mathbf{u}'(t) = F(t, \mathbf{u}(t)) \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases} \quad (6)$$

citiamo il metodo di **Eulero esplicito** (posto $\mathbf{u}_{n+1} = \mathbf{u}(t_{n+1})$)

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + hF(t_n, \mathbf{u}_n) \\ \mathbf{u}_0 \text{ assegnato} \end{cases} \quad (7)$$

e quello di **Eulero implicito**

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + hF(t_{n+1}, \mathbf{u}_{n+1}) \\ \mathbf{u}_0 \text{ assegnato} \end{cases} \quad (8)$$

Equazione del calore.

Nel nostro caso

$$F(t, \mathbf{v}(t)) := \Lambda \mathbf{v}(t) + \mathbf{g}(t)$$

e quindi il metodo di Eulero esplicito genera la successione

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_n + h_t(\Lambda \mathbf{v}_n + \mathbf{g}(t_n)) \\ \mathbf{v}_0 \text{ assegnato} \end{cases} \quad (9)$$

mentre Eulero implicito determina

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_n + h_t(\Lambda \mathbf{v}_{n+1} + \mathbf{g}(t_{n+1})) \\ \mathbf{v}_0 \text{ assegnato} \end{cases} \quad (10)$$

o equivalentemente

$$\begin{cases} (I - h_t \Lambda) \mathbf{v}_{n+1} = \mathbf{v}_n + h_t \mathbf{g}(t_{n+1}) \\ \mathbf{v}_0 \text{ assegnato} \end{cases} \quad (11)$$

Osserviamo che a differenza del metodo esplicito, poichè

$$\begin{cases} (I - h_t \Lambda) \mathbf{v}_{n+1} = \mathbf{v}_n + h_t \mathbf{g}(t_{n+1}) \\ \mathbf{v}_0 \text{ assegnato} \end{cases} \quad (12)$$

ad ogni iterazione si richiede la soluzione di un'equazione (che nel nostro caso è lineare). Usando i primi due teoremi di Gerschgorin, si può mostrare che la matrice $(I - h_t \Lambda)$ è **definita positiva** (e quindi non singolare).

A partire da Eulero esplicito ed Eulero implicito si definiscono i cosiddetti **θ -metodi** in cui

$$\begin{aligned} \mathbf{v}_{n+1} = & (1 - \theta) (\mathbf{v}_n + h_t (\Lambda \mathbf{v}_n + \mathbf{g}(t_n))) \\ & + \theta (\mathbf{v}_n + h_t (\Lambda \mathbf{v}_{n+1} + \mathbf{g}(t_{n+1}))) \end{aligned} \quad (13)$$

con $\mathbf{v}_0$ assegnato. Per $\theta = 0$ si ottiene il metodo di Eulero esplicito mentre per $\theta = 1$ si ottiene il metodo di Eulero implicito.

Studiamo numericamente l'equazione del calore

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + G, & 0 < x < 1, t > 0 \\ u(0, t) = d_0(t), u(1, t) = d_1(t), & t \geq 0 \\ u(x, 0) = f(x), & 0 \leq x \leq 1 \end{cases} \quad (14)$$

per

$$\begin{cases} G(x, t) = (-0.1 + \pi^2) (\exp(-0.1 \cdot t) \sin(\pi x)) \\ d_0(t) = 0, d_1(t) = 0 \\ f(x) = \sin(\pi x) \end{cases} \quad (15)$$

avente quale soluzione

$$u(x, t) = \exp(-0.1 \cdot t) \sin(\pi x).$$

Esperimento numerico.

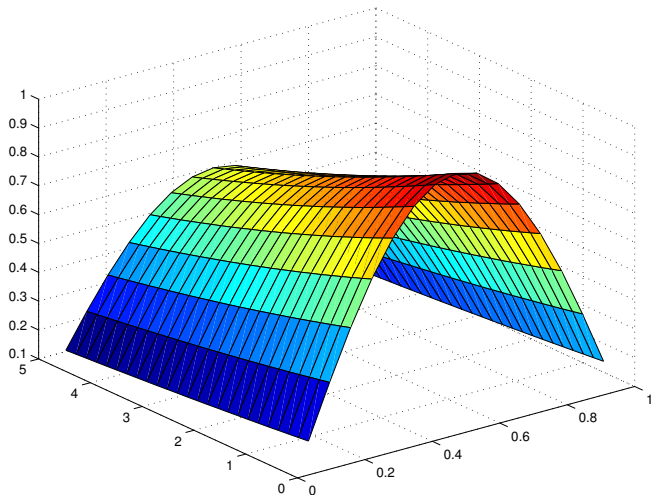


Figura : Grafico della soluzione dell'equazione del calore (14).

Esperimento numerico.

```
function gt=g(t,x,delta,d0,d1,G)
% CALCOLA
% g(t)=(1/hx^2)*(d0(t),0,...,0,d1(t))+[G(x1,t),...,G(x_(m-1)
    ,t)].
gt=feval(G,x,t);
gt(1)=gt(1)+(1/delta^2)*feval(d0,t);
gt(length(gt))=gt(length(gt))+(1/delta^2)*feval(d1,t);
```

Esperimento numerico.

```
function [V_hist,x_mid,t_mid,hx,ht]=cnheat(theta,tfin,m,f,d0
,d1,G,time_step_factor)
% INPUT.
% theta : PARAMETRO DEL THETA METODO (TRA 0 ED 1,
%        POSSIBILMENTE 0,1 INCLUSI).
% tfin   : ISTANCE FINALE.
% m      : DETERMINA IL PASSO SPAZIALE CON PASSO h_x=1/m.
% f      : u(x,0)=f(x)
% d0, d1 : u(0,t)=d0(t), u(1,t)=d1(t).
% G      : (D_t)u=(D^2_x)u+G.
% time_step_factor: IL PASSO TEMPORALE E' time_step_factor*(
%                   hx^2)/2.
% OUTPUT.
% V_hist : SOLUZIONE ISTANTE PER ISTANTE. RIGHE INIZIALI ->
%         BASSO t.
% x_mid  : PUNTI INTERNI NELLA VARIABILE x.
% t_mid  : PUNTI INTERNI NELLA VARIABILE t.
% hx     : PASSO SPAZIALE.
% ht     : PASSO TEMPORALE.
```

Esperimento numerico.

```
hx=1/m; % DELTA=hx
matorder=m-1;

ht=time_step_factor*(hx^2)/2; % STEP TEMPORALE.

x=(0:hx:1)';
x_mid=x(2:length(x)-1,:); % PUNTI INTERNI.

u0=feval(f,x_mid); % VETTORE SOLUZIONE AL TEMPO
    "0".

t=(0:ht:tfin)'; % TEMPI DA ANALIZZARE.
t_mid=t(2:length(t),1); % TEMPI t > 0.

% COSTRUZIONE MATRICE LAMBDA.
submat=[zeros(1,matorder); eye(matorder-1) (zeros(1,matorder-1))'];
supmat=submat';
lambda_matrix=(1/hx^2)*(diag(-2*ones(m-1,1))+submat+supmat);
```

Esperimento numerico.

```
if theta < 1
    gt_prev=g(0,x_mid,hx,d0,d1,G);
end

V_old=u0;
V_hist=[V_old'];
err_hist=[];
```

Esperimento numerico.

```
for index=2:length(t)
    % CALCOLO  $u'(t_{curr}) = \Lambda u(t_{curr}) + g(t_{curr})$ ,
    t_curr=t(index);
    gt_curr=g(t_curr,x_mid,hx,d0,d1,G);
    switch theta
    case 1
        A=eye(size(lambda_matrix))-ht*lambda_matrix;
        b=V_old+ht*gt_curr; V_new=A\b;
    case 0
        V_new=V_old+ht*(lambda_matrix*V_old+gt_prev);
        gt_prev=gt_curr;
    otherwise
        A=eye(size(lambda_matrix))-(ht*theta)*...
            lambda_matrix;
        b=V_old+(ht*(1-theta))*lambda_matrix*V_old+...
            (ht*(1-theta))*gt_prev+(ht*theta)*gt_curr;
        V_new=A\b; gt_prev=gt_curr;
    end
    V_hist=[V_hist; V_new']; V_old=V_new;
end
```

Esperimento numerico.

- 1 Il parametro m determina il passo spaziale $h_x = 1/m$;
- 2 se $\text{theta}=0$ allora si utilizza il metodo di Eulero esplicito, mentre se $\text{theta}=1$ Eulero implicito;
- 3 la variabile t_{fin} determina l'istante finale;
- 4 il parametro timestepfactor determina il passo temporale a partire da quello spaziale; se $h_{t_{\text{max}}} = h_x^2/2$ per $h_x = 1/m$ il passo usato da Eulero esplicito è $\text{timestepfactor} * h_{t_{\text{max}}}$;
- 5 in seguito la demo valuta uno dei metodi per $m = 2^k$ con $k = 2, 3, 4$, calcolando le ratio e_{2h_x}/e_{h_x} dove si è posto

$$e_{h_x} = \max_i |v_i(t_{\text{fin}}) - u_{x_i, t_{\text{fin}}})|$$

in cui $v_i(t) := v^{(h_x)}(x_i, t_{\text{fin}})$ è la soluzione ottenuta dal metodo scegliendo il parametro temporale uguale a $h_x = 1/m$;

- 6 il parametro m_{vect} all'interno dello switch iniziale, determina gli m da analizzare.

Esperimento numerico.

```
% demoexample: 1 o 2. theta 0:EE.; 0.5: CN. 1:EI.
demoexample=2; theta=1; mvect=[4 8 16 32 64];
time_step_factor_vect=ones(size(mvect));
switch demoexample
case 1
    tfin=10; % TEMPO FINALE.
    G=inline('(-0.1+pi^2)*(exp((-0.1)*t).*sin(pi*x))','x','t');
    d0=inline('zeros(size(t))','t');
    d1=inline('exp((-0.1)*t).*sin(pi)','t');
    f=inline('sin(pi*x)','x');
    solution=inline('exp((-0.1)*t).*sin(pi*x)','x','t');
case 2
    tfin=0.1; % TEMPO FINALE.
    G=inline('zeros(size(x))','x','t');
    d0=inline('zeros(size(t))','t');
    d1=inline('zeros(size(t))','t');
    f=inline('sin(pi*x)','x');
    solution=inline('exp((-pi^2)*t).*sin(pi*x)','x','t');
end
```

Esperimento numerico.

```
err_hist_prev_m=[];
fprintf('\n \t [THETA]:%3.3f [TFIN]:%3.3f',theta,tfin);
for mindex=1:length(mvect)
    err_hist=[]; m=mvect(mindex); % m IN UNA LISTA.
    time_step_factor=time_step_factor_vect(mindex);
    [V_hist,x_mid,t_mid,hx,ht]=cnheat(theta,tfin,m,f,d0,d1,G,
        time_step_factor);
    [X, Y]=meshgrid(x_mid,t_mid); % VALUTAZIONE ERRORI.
    U=feval(solution,X,Y); % SOL. ESATTA NELLA GRIGLIA.
    % ERRORE COMPIUTO PER "m" FISSATO.
    err=norm(U(size(U,1),:)-V_hist(size(V_hist,1),:),inf);
    err_hist=[err_hist; err];
    fprintf('\n \t [m]: %3.0f [ERROR]: %2.2e [hx]: %2.2e [ht]:
        %2.2e', m, err,hx,ht);
    if length(err_hist_prev_m) > 0
        fprintf(' [RATIO]: %2.2f', err_hist_prev_m(size(
            err_hist_prev_m,1))/err );
    end
    err_hist_prev_m=err_hist;
end
```

Esperimento numerico: Eulero esplicito.

Per motivi di stabilità tipici di Eulero esplicito, il passo temporale h_t deve essere inferiore o uguale a $h_x^2/2$. Vediamo su vari esempi cosa succede numericamente. Dopo aver settato in `demoheat` il parametro `theta=0` scegliamo per esempio `timestepfactor=1.5`. Quindi dalla shell di Matlab/Octave digitiamo quanto segue

```
>> demoheatcn
[THETA]: 0.000 [TFIN]: 5.000 [TIME STEP FACTOR]: 1.50e+000
[m]: 4 [ERROR]: 1.40e+004 [hx]: 2.50e-001 [ht]: 4.69e-002
[m]: 8 [ERROR]: 7.32e+100 [hx]: 1.25e-001 [ht]: 1.17e-002
[RATIO]: 0.00
[m]: 16 [ERROR]: NaN [hx]: 6.25e-002 [ht]: 2.93e-003 [RATIO
]: NaN
>>
```

Esperimento numerico: Eulero esplicito.

Evidentemente bisogna scegliere un parametro `timestepfactor` più piccolo. Proviamo ad esempio `timestepfactor=1.1`.

```
>> demoheatcn
[THETA]:0.000 [TFIN]:5.000 [TIME STEP FACTOR]:1.10e+000
[m]: 4 [ERROR]:3.25e-002 [hx]:2.50e-001 [ht]:3.44e-002
[m]: 8 [ERROR]:2.89e+011 [hx]:1.25e-001 [ht]:8.59e-003 [
  RATIO]:0.00
[m]: 16 [ERROR]:3.77e+149 [hx]:6.25e-002 [ht]:2.15e-003 [
  RATIO]:0.00
>>
```

Esperimento numerico: Eulero esplicito.

Il metodo non fornisce evidentemente risultati apprezzabili.

Scegliamo ora **timestepfactor=1.0**: il metodo di Eulero esplicito finalmente converge.

```
>>> demoheatcn
[THETA]: 0.000 [TFIN]: 5.000 [TIME STEP FACTOR]: 1.00e+000
[m]: 4 [ERROR]: 3.25e-002 [hx]: 2.50e-001 [ht]: 3.13e-002
[m]: 8 [ERROR]: 7.93e-003 [hx]: 1.25e-001 [ht]: 7.81e-003
[RATIO]: 4.10
[m]: 16 [ERROR]: 1.97e-003 [hx]: 6.25e-002 [ht]: 1.95e-003
[RATIO]: 4.02
>>>
```

In definitiva affinché il metodo di Eulero esplicito converga, il passo temporale dev'essere scelto dell'ordine di $h_x^2/2$, che in molti casi risulta essere troppo piccolo e rende il metodo non competitivo dal punto di vista computazionale.

Esperimento numerico: Eulero implicito.

Vediamo in questa sezione il comportamento di Eulero implicito. Osserviamo che a differenza di Eulero esplicito richiede la soluzione di sistemi lineari tridiagonali, ma ciò non è un problema dal punto di vista computazionale (il costo è di $5m$ per ogni t_i).

Proviamo il comportamento per **timestepfactor=1.5**, dopo aver posto $\theta=1$. Il metodo di Eulero implicito, a differenza di Eulero esplicito, **converge**. Infatti

```
>> demoheatcn
[THETA]:1.000 [TFIN]:5.000 [TIME STEP FACTOR]:1.50e+000
[m]:4 [ERROR]:3.26e-002 [hx]:2.50e-001 [ht]:4.69e-002
[m]:8 [ERROR]:7.95e-003 [hx]:1.25e-001 [ht]:1.17e-002
      [RATIO]:4.11
[m]:16 [ERROR]:1.97e-003 [hx]:6.25e-002 [ht]:2.93e-003
      [RATIO]:4.03
>>
```

Esperimento numerico: Eulero implicito.

Per curiosità proviamo per `timestepfactor=10`, quindi con un passo temporale h_t relativamente grande, ottenendo

```
>> demoheatcn
[THETA]:1.000 [TFIN]:5.000 [TIME STEP FACTOR]:1.00e+001
[m]:4 [ERROR]:3.26e-002 [hx]:2.50e-001 [ht]:3.13e-001
[m]:8 [ERROR]:7.96e-003 [hx]:1.25e-001 [ht]:7.81e-002
      [RATIO]: 4.10
[m]:16 [ERROR]:1.98e-003 [hx]:6.25e-002 [ht]:1.95e-002
      [RATIO]: 4.02
>>
```

Esperimento numerico: Eulero implicito.

Si può mostrare che in effetti, il metodo è **A-stabile**, e quindi non richiede alcun vincolo sullo step temporale. Ciò significa che per ogni valore di h_x e h_t la propagazione dell'errore avanzando nel tempo è *sotto controllo* o come si dice il metodo è **incondizionatamente stabile**. Inoltre se tanto la soluzione quanto d_0 , d_1 , g ed f sono sufficientemente regolari allora con ovvia notazione

$$\max_{i,k} |u(x_i, t_k) - u_{i,k}| = O(h_t + h_x^2).$$

Eseguire gli stessi test eseguiti per Eulero esplicito col θ -metodo, per $\theta = 0.25$, $\theta = 0.5$, $\theta = 0.75$. Com'è il comportamento del metodo? Simile ad Eulero esplicito o ad Eulero esplicito?