Equazione del calore.

Alvise Sommariva

Università degli Studi di Padova
Dipartimento di Matematica

26 maggio 2014

Consideriamo l'*equazione del calore*

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

Sia $m > 0$ intero e sia $h_x = 1/m$ ed $x_j = jh_x$ con $j = 0, 1, \ldots, m$.
Si può mostrare che per $j = 1, 2, \ldots, m-1$ e $\xi_j \in (x_{j-1}, x_{j+1})$

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

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

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

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

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

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

$$u_j(0) = f(x_j), \; j = 1, \ldots, m-1.$$

Il sistema differenziale (3) può essere riscritto matricialmente. Posto

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

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

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

$$A = \frac{1}{h_x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & \ldots & 0 \\ 1 & -2 & 1 & 0 & \ldots & 0 \\ 0 & 1 & -2 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \qquad (4)$$

otteniamo che (3) è equiv. al sistema di eq. differenziali (lineari)

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

Nota.

Osserviamo che la matrice a predominanza diagonale

$$A = \frac{1}{h_x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \tag{6}$$

è

- *A è simm. definita negativa per i teoremi di Gershgorin.*
- *Si mostra che se* $A \in \mathbb{R}^{(N-1)\times(N-1)}$, *allora i suoi autov. sono*

$$-\frac{2 - 2cos(\frac{j\pi}{N})}{h_x^2}.$$

- *Diagonalizzabile e* $I - h_t A$ *è invertibile (A è def. negativa).*

Equazione del calore.

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

$$\begin{cases} \mathbf{u}'(t) = F(t, \mathbf{u}(t)) \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases} \tag{7}$$

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

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

e quello di Eulero implicito

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + hF(t_{n+1}, \mathbf{u}_{n+1}) \\ \mathbf{u}_0 \text{ assegnato} \end{cases} \tag{9}$$

Equazione del calore.

Nel nostro caso

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

e quindi il metodo di Eulero esplicito genera la successione

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_n + h_t(A\mathbf{v}_n + \mathbf{g}(t_n)) \\ \mathbf{v}_0 \text{ assegnato} \end{cases} \tag{10}$$

mentre Eulero implicito determina

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

o equivalentemente

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

Nota.

Osserviamo che a differenza del metodo esplicito, poichè

$$\begin{cases} (I - h_t A)\mathbf{v}_{n+1} = \mathbf{v}_n + h_t \mathbf{g}(t_{n+1}) \\ \mathbf{v}_0 \ \textit{assegnato} \end{cases} \tag{13}$$

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

A partire da Eulero esplicito ed Eulero implicito si definiscono i cosidetti $\theta-$*metodi* in cui

$$\begin{aligned} \mathbf{v}_{n+1} &= (1-\theta)\left(\mathbf{v}_n + h_t(A\mathbf{v}_n + \mathbf{g}(t_n))\right) \\ &\quad +\theta\left(\mathbf{v}_n + h_t(A\mathbf{v}_{n+1} + \mathbf{g}(t_{n+1}))\right) \end{aligned} \tag{14}$$

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.

Stabilità

Consideriamo l'equazione del calore (con $G \equiv 0$)

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \ 0 < x < 1, \ t > 0 \\ u(0,t) = 0, \ u(1,t) = 0, \ t \geq 0 \\ u(x,0) = f(x), \ 0 \leq x \leq 1 \end{cases} \tag{15}$$

Si dimostra che la soluzione esatta tende a 0 e quindi si richiede che la soluzione numerica abbia la stessa proprietà . In tal caso la soluzione si dice assolutamente stabile.

Stabilità

Il problema continuo viene discretizzato come proposto precedentemente, ottenendo il sistema differenziale

$$\mathbf{u}'(t) = A\mathbf{u}(t),\ \mathbf{u}(0) = \mathbf{u}_0 \tag{16}$$

che è possibile risolvere con un generico θ-metodo

$$\mathbf{v}_{n+1} = (1-\theta)\left(\mathbf{v}_n + h_t A\mathbf{v}_n\right) + \theta\left(\mathbf{v}_n + h_t A\mathbf{v}_{n+1}\right) \tag{17}$$

con $\mathbf{v}_0$ assegnato.

Stabilità Eulero esplicito

Nel caso di Eulero esplicito abbiamo

$$\mathbf{v}_{n+1} = \mathbf{v}_n + h_t A \mathbf{v}_n = (I + h_t A)\mathbf{v}_n \tag{18}$$

Siccome A è diagonalizzabile, abbiamo che per qualche S, $A = S^{-1}\Lambda S$, con Λ diagonale. Posto $\mathbf{z}_n = S\mathbf{v}_n$,

$$\begin{aligned} \mathbf{z}_{n+1} &= S\mathbf{v}_{n+1} = S(\mathbf{v}_n + h_t A\mathbf{v}_n) = \mathbf{z}_n + h_t SA\mathbf{v}_n \\ &= \mathbf{z}_n + h_t SS^{-1}\Lambda S\mathbf{v}_n \\ &= \mathbf{z}_n + h_t \Lambda \mathbf{z}_n = (I + h_t\Lambda)\mathbf{z}_n \end{aligned} \tag{19}$$

Si noti che $\mathbf{z}_n = S\mathbf{v}_n \rightarrow 0$ implica $\mathbf{v}_n \rightarrow 0$ in quanto

$$\|\mathbf{v}_n\| = \|S^{-1}\mathbf{z}_n \leq \|S^{-1}\|\|\mathbf{z}_n\| \rightarrow 0.$$

Stabilità Eulero esplicito

Se $\mathbf{z}_n = (u_{n,i})_i$, $\Lambda_{i,i} = \lambda_i$, essendo Λ diagonale, da

$$\mathbf{z}_{n+1} = (I + h_t \Lambda)\mathbf{z}_n \tag{20}$$

la i-sima equazione diventa

$$z_{n+1,i} = (1 + h_t \lambda_i) z_{n,i}.$$

che come nel caso di Eulero esplicito scalare, converge asintoticamente a 0 se e solo se

$$|1 + h_t \lambda_i| < 1.$$

Stabilità Eulero esplicito

La condizione $|1 + h_t \lambda_i| < 1$ pone i vincoli sul passo

$$h_t \leq \frac{2}{|\lambda_i|}, \;\; i = 1, \ldots, N-1.$$

Essendo

$$\lambda_i = -\frac{2 - 2cos(\frac{j\pi}{N})}{h_x^2}, \; j = 1, \ldots, N-1$$

abbiamo che $\lambda_i < 0$ per $i = 1, \ldots, N-1$ e

$$|\min_i(\lambda_i)| = |\lambda_{N-1}| = |-\frac{2 - 2cos(\frac{(N-1)\pi}{N})}{h_x^2}| \approx \frac{|-4|}{h_x^2} = \frac{4}{h_x^2}$$

da cui la condizione sulla stabilità asintotica

$$h_t \leq \frac{h_x^2}{2}.$$

Stabilità Eulero implicito

Nel caso di Eulero implicito abbiamo

$$\mathbf{v}_{n+1} = \mathbf{v}_n + h_t A \mathbf{v}_{n+1} \tag{21}$$

da cui immediatamente

$$\mathbf{v}_{n+1} = (I - h_t A)^{-1} \mathbf{v}_n \tag{22}$$

Siccome A è diagonalizzabile, abbiamo che per qualche S, $A = S^{-1} \Lambda S$, con Λ diagonale. Posto $\mathbf{z}_n = S \mathbf{v}_n$,

$$\begin{aligned} \mathbf{z}_{n+1} &= S\mathbf{v}_{n+1} = S(I - h_t A)^{-1}\mathbf{v}_n = S(S^{-1}S - h_t S^{-1}\Lambda S)^{-1}\mathbf{v}_n \\ &= S(S^{-1}(I - h_t\Lambda)S)^{-1}\mathbf{v}_n = SS^{-1}(I - h_t\Lambda)^{-1}S\mathbf{v}_n \\ &= (I - h_t\Lambda)^{-1}\mathbf{z}_n \end{aligned} \tag{23}$$

Se $\mathbf{z}_n = (z_{n,i})_i$, $\Lambda_{i,i} = \lambda_i$, la i-sima equazione diventa

$$z_{n+1,i} = (1 - h_t \lambda_i)^{-1} z_{n,i}.$$

che quindi converge converge asintoticamente a 0 se e solo se

$$|1/(1 - h_t \lambda_i)| < 1.$$

Stabilità Eulero implicito

Siccoma A è definita negativa e simile a Λ, abbiamo che $\lambda_i < 0$ e quindi

$$|1/(1 - h_t \lambda_i)| < 1.$$

è verificata per qualsiasi scelta di h_t.

Questo significa che qualsiasi sia h_t, la successione del metodo di Eulero implicito converge asintoticamente a 0, come la soluzione dell'equazione del calore.

Stabilità Crank-Nicolson

Nel caso di Crank-Nicolson abbiamo

$$\mathbf{v}_{n+1} = (1/2)(\mathbf{v}_n + h_t A \mathbf{v}_{n+1}) + (1/2)(\mathbf{v}_n + h_t A \mathbf{v}_n) \tag{24}$$

da cui immediatamente

$$\mathbf{v}_{n+1} = \mathbf{v}_n + (1/2) h_t A \mathbf{v}_{n+1} + (1/2) h_t A \mathbf{v}_n \tag{25}$$

cioè

$$(1 - \frac{h_t}{2} A)\mathbf{v}_{n+1} = (1 + \frac{h_t}{2} A)\mathbf{v}_n. \tag{26}$$

Siccome A è diagonalizzabile, abbiamo che per qualche S, $A = S^{-1}\Lambda S$, con Λ diagonale. Posto $\mathbf{z}_n = S\mathbf{v}_n$,

$$S(1 - \frac{h_t}{2} A)\mathbf{v}_{n+1} = S(1 - \frac{h_t}{2} A) S^{-1} S \mathbf{v}_{n+1} = (1 - \frac{h_t}{2}\Lambda)\mathbf{z}_{n+1} \tag{27}$$

e pure

$$S(1 + \frac{h_t}{2} A)\mathbf{v}_n = S(1 + \frac{h_t}{2} A) S^{-1} S \mathbf{v}_n = (1 + \frac{h_t}{2}\Lambda)\mathbf{z}_n \tag{28}$$

Stabilità Crank-Nicolson

Da

$$(1 - \frac{h_t}{2}A)\mathbf{v}_{n+1} = (1 + \frac{h_t}{2}A)\mathbf{v}_n, \tag{29}$$

$$S(1 - \frac{h_t}{2}A)\mathbf{v}_{n+1} = S(1 - \frac{h_t}{2}A)S^{-1}S\mathbf{v}_{n+1} = (1 - \frac{h_t}{2}\Lambda)\mathbf{z}_{n+1} \tag{30}$$

e

$$S(1 + \frac{h_t}{2}A)\mathbf{v}_n = S(1 + \frac{h_t}{2}A)S^{-1}S\mathbf{v}_n = (1 + \frac{h_t}{2}\Lambda)\mathbf{z}_n \tag{31}$$

moltiplicando a sinistra ambo i membri di (29) per S, abbiamo

$$(1 - \frac{h_t}{2}\Lambda)\mathbf{z}_{n+1} = (1 + \frac{h_t}{2}\Lambda)\mathbf{z}_n$$

da cui

$$\mathbf{z}_{n+1} = (1 - \frac{h_t}{2}\Lambda)^{-1}(1 + \frac{h_t}{2}\Lambda)\mathbf{z}_n$$

Stabilità Crank-Nicolson

La i-sima equazione diventa

$$\mathbf{z}_{n+1,i} = (1 - \frac{h_t}{2}\lambda_i)^{-1}(1 + \frac{h_t}{2}\lambda_i)\mathbf{z}_{n,i}$$

e come nel caso scalare converge a 0 se e solo se

$$\frac{|1 + \frac{h_t}{2}\lambda_i|}{|1 - \frac{h_t}{2}\lambda_i|} < 1$$

che è sempre verificata poichè $\lambda_i < 0$.

Questo significa che qualsiasi sia h_t, la successione del metodo di Crank-Nicolson converge asintoticamente a 0, come la soluzione dell'equazione del calore.

Nota.

Si dimostra che le matrici dei sistemi lineari dei metodi di Eulero implicito e di Crank-Nicolson sono molto meglio condizionate della matrice A. Infatti si dimostra che

$$\lambda_{min}(A) \approx -4/h_x^2$$

ed essendo $1 - \cos(x) \approx x^2/2$ *per* $x \approx 0$, $Nh_x = 1$,

$$\lambda_{max}(A) = -\frac{2(1-\cos(\pi/N))}{h_x^2} \approx -\frac{2(\pi/N)^2}{2h_x^2} = -\pi^2$$

da cui $cond_2(A) = \frac{4}{\pi^2 h_x^2}$ *mentre*

$$cond_2(1 - h_t A) = \frac{1 - h_t \lambda_{min}(A)}{1 - h_t \lambda_{max}(A)} \approx \frac{1 + 4h_t/h_x^2}{1 + h_t \pi^2} \approx 1 + 4h_t/h_x^2.$$

Stabilità

Nota.

Quanto visto, presuppone che nell'equazione del calore sia $g \equiv 0$*, ma una analisi simile può essere effettuata anche per il caso in cui* $g \neq 0$*.*

Esperimento numerico.

Studiamo numericamente l'equazione del calore

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + G, \; 0 < x \; < 1, \; t > 0 \\ u(0,t) = d_0(t), \; u(1,t) = d_1(t), \; t \geq 0 \\ u(x,0) = f(x), \; 0 \leq x \leq 1 \end{cases} \tag{32}$$

per

$$\begin{cases} G(x,t) = (-0.1 + \pi^2)\,(\exp(-0.1 \cdot t)\,\sin(\pi\, x)) \\ d_0(t) = 0, \;\; d_1(t) = 0 \\ f(x) = \sin(\pi\, x) \end{cases} \tag{33}$$

avente quale soluzione

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

Esperimento numerico.

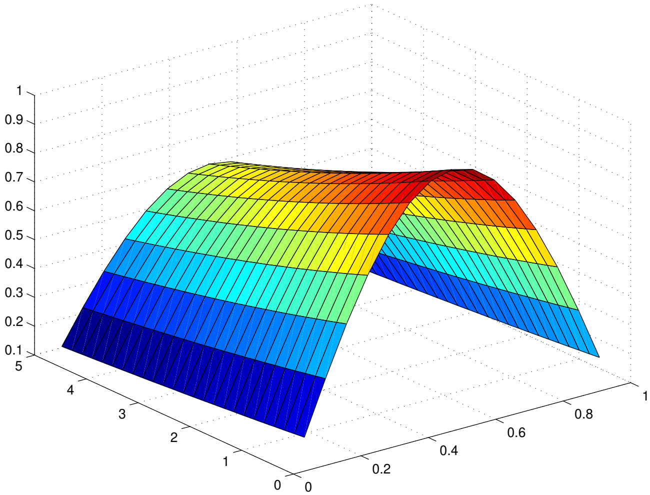

Figura: Grafico della soluzione dell'equazione del calore (32).

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     :  ISTANTE 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.
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))']) ';
lambda_matrix=(1/hx^2)*(diag(-2*ones(m-1,1))+submat+supmat);
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
```

```
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.

1. Il parametro `m` determina il passo spaziale $h_x = 1/m$;
2. se `theta=0` allora si utilizza il metodo di Eulero esplicito, mentre se `theta=1` Eulero implicito;
3. la variabile `tfin` determina l'istante finale;
4. il parametro `timestepfactor` determina il passo temporale a partire da quello spaziale; se $h_{tmax} = h_x^2/2$ per $h_x = 1/m$ il passo usato da Eulero esplicito è timestepfactor*htmax;
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

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

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

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 affinchè 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
>>
```

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'e' il comportamento del metodo? Simile ad Eulero esplicito o ad Eulero esplicito?

Esercizio (dal Quarteroni-Saleri).

Consideriamo lequazione del calore

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

con

$$G(x,t) = -\sin(x)\sin(t) + \sin(x)\cos(t)$$

$$u(x,0) = \sin(x)$$

e condizioni al bordo

$$u(0,t) = 0,$$

$$u(1,t) = \sin(1)\cos(t).$$

Esercizio.

In questo caso la soluzione

$$u(x,t) = \sin(x)\cos(t).$$

Si confrontino via plot semi-logaritmico il comportamento degli errori

$$\max_{i=0,\ldots,N} |u(x_i, 1) - u_i^M|$$

rispetto al passo temporale su una griglia uniforme lungo la direzione spaziale con $h_x = 0.002$, $h_x = 0.004$, $h_x = 0.008$ e su una griglia temporale uniforme con $h_t = 10^{-3} + 2k \cdot 10^{-3}$, $k = 0, \ldots, 10$, per un θ-metodo con $\theta = 0$, $\theta = 0.5$, $\theta = 1$.

I valori $\{u_i^M\}$ rappresentano la soluzione alle differenze finite calcolata al tempo $t^M = 1$.