Soluzioni II prova in itinere

21 giugno 2022

Test 1

Si ricorda che il polinomio interpolatore, nella notazione Matlab, è memorizzato come $\Pi_n(x) = c_1\, x^n + c_2\, x^{n-1} + \ldots + c_{n-1}\, x^2 + c_n\, x + c_{n+1}$ e pertanto $\dfrac{d\Pi_n}{dx}(x) = n\, c_1\, x^{n-1} + (n-1)\, c_2\, x^{n-2} + \ldots + 2\, c_{n-1}\, x + c_n$. I valori richiesti possono essere calcolati tramite i comandi

```
x  = [0,  0.5,  1];
y  = [2,    1,  1.5];
c  = polyfit(x,y,2);
p  = polyval(c,0.25)
c1 = [2*c(1),  c(2)];
p1 = polyval(c1,0.25)
```

Risulta pertanto $\Pi_2(0.25) = 1.3125$ e $\dfrac{d\Pi_2}{dx}(0.25) = -2.0000$.

Test 2

Il valore richiesto può essere determinato senza calcoli, osservando che l'interpolante polinomiale lineare a tratti assume il massimo e il minimo valore in corrispondenza del massimo e minimo valore dei dati, $\{y_i\}_{i=0}^{n}$, rispettivamente. Pertanto si ha che

$$\min_{x\in[0,1]} \Pi_1^H(x) = \min_{i=0,\ldots,n} y_i = -0.5.$$

Test 3

Si osservi che l'interpolante polinomiale lineare a tratti è definito indipendentemente su ogni intervallo "macroscopico". Il punto di valutazione, 1.6, appartiene al quarto intervallo, $[1.5, 2]$. Pertanto, per costruire Π_1^H occorre interpolare le due coppie di

dati, $x = [1.5, 2]$ e $y = [f(1.5), f(2)] = [-1, 0]$. Questo definisce $\Pi_1^H(x) = 2(x - 2)$. Risulta quindi, $\Pi_1^H(1.6) = -0.8$.
Alternativamente, tramite Matlab, si possono usare i comandi

```
x = linspace(0,2,5);
y = sin(pi*x);
interp1(x,y,1.6)
```

Test 4

Lo scarto quadratico può essere calcolato tramite i comandi

```
x = linspace(0,1,5);
y = [3, 0.5, 1.5, -0.5, 1];
c = polyfit(x,y,2);
sum((polyval(c,x) - y).^2)
```

che fornisce il valore 2.4143. Si osservi che, in generale, il polinomio approssimante ai minimi quadrati non interpola i dati e pertanto lo scarto quadratico risulta non nullo.

Test 5

L'approssimazione dell'integrale con la formula dei trapezi composita può essere ottenuta tramite i comandi

```
f = @(x) sqrt(2 + abs(x));
x = linspace(-1,2,4);
y = f(x);
H = 1;
It = H*(0.5*y(1) + y(2) + y(3) + 0.5*y(4))
```

che implementano la formula generale su M intervalli uniformi

$$I_t^H(f) = (0.5f(x_0) + \sum_{i=1}^{M-1} f(x_i) + 0.5f(x_M))H,$$

con $H = 1$ e $M = 3$. Si ottiene $I_t^H(f) = 5.0123$.

Test 6

L'errore di quadratura, in valore assoluto, $|E^H(f)|$, di una formula composita, può essere sempre maggiorato da

$$|E^H(f)| \leq C\,(b-a)H^p \max_{x\in[a,b]} |f^{(r+1)}(x)| = C\,(b-a)^{p+1}M^{-p} \max_{x\in[a,b]} |f^{(r+1)}(x)|,$$

dove C è una costante positiva, H è l'ampiezza della partizione (supposta uniforme), $M = (b-a)/H$ è il numero di sottointervalli, p è l'ordine di accuratezza e r il grado di esattezza. Scrivendo due relazioni analoghe, per un numero di intervalli M_1 e M_2, associati alle due ampiezze H_1 e H_2, rispettivamente, e facendo il quoziente membro a membro, si ottiene che

$$\frac{e_1(f)}{e_2(f)} = \frac{|E^{H_1}(f)|}{|E^{H_2}(f)|} \simeq \left(\frac{M_2}{M_1}\right)^p.$$

Si ricava quindi che

$$M_2 = \left(\frac{e_1(f)}{e_2(f)}\right)^{1/p} M_1.$$

Con i valori forniti nel testo, si ha

$$M_2 = \left(\frac{10^{-1}}{10^{-3}}\right)^{1/2} 20 = 200.$$

Test 7

Tramite i seguenti comandi

```
f     = @(t,y) -1/81*(1+ sin(t))*y^2;
df_dy = @(t,y) -2/81*(1+ sin(t))*y;
tf = 5;
h  = 0.2;
y0 = 9;
[t_h,u_h,it_n]=eulero_indietro_newton (f, df_dy, tf, y0, h);
u_h(end)
```

è possibile calcolare il valore richiesto, che risulta $u_{N_t} = 5.5989$.

Test 8

Il problema di diffusione-reazione con condizioni al bordo miste Dirichlet-Neumann è implementato nello script

```
h = 0.1;
N = 1/h - 1;
x = linspace(0,1,N+2);
alpha = 1; % bc Dirichlet left
gamma = 0; % bc Neumann   right
mu    = 1; % coeff. diffusione
sigma = 2; % coeff. reazione
f = @(x) 3 + x; % forzante
Ad = mu/h^2*(2*eye(N+1) - diag(ones(N,1),1) - diag(ones(N,1)
    ,-1));
Ar = sigma*eye(N+1);
A = Ad + Ar;
A(end,[end-1,end]) = [-1,1]/h;
b = zeros(N+1,1);
b(1:N) = f(x(2:N+1));
b(1) = b(1) + alpha*mu/h^2;
b(end) = gamma;
U = [alpha;A\b];
U(end)
```

e fornisce $U_{N+2} = 1.4207$.

Test 9

Il problema differenziale è di diffusione-trasporto con condizioni al bordo di Dirichlet. Il coefficiente convettivo vale $\eta = -50 < 0$. Petanto, l'approssimazione del termine di trasporto è data dalla differenza finita in avanti

$$\eta\, u'(x_i) \simeq \eta\, \frac{U_{i+1} - U_i}{h}.$$

L'implementazione è descritta nel seguente script

```
h = 0.1;
N = 1/h - 1;
x = linspace(0,1,N+2); % in questo caso non serve
alpha =   0; % bc Dirichlet left
beta  =   3; % bc Dirichlet right
mu    =   1; % coeff. diffusione
eta   = -50; % coeff. trasporto
Ad =  mu/h^2*(2*eye(N) - diag(ones(N-1,1),1) - diag(ones(N
    -1,1),-1));
```

```
At = eta/h  *( -eye(N) + diag(ones(N-1,1),1));
A = Ad + At;
b = zeros(N,1);
b(1)   = b(1)   + alpha* mu/h^2;
b(end) = b(end) +  beta*(mu/h^2 -eta/h);
U = [alpha; A\b; beta];
U(2)
```

Il valore richiesto è quindi, $U_1 = 2.5000$.

Test 10

Il problema differenziale rappresenta l'equazione del calore completata da condizioni al bordo di Dirichlet nulle e da una assegnata condizione iniziale. Il coefficiente di diffusione vale $\mu = 3$. Senza fare calcoli complessi, la condizione di assoluta (asintotica) stabilità richiesta per il metodo di Eulero in avanti e differenze finite centrate in spazio è garantita per

$$\Delta t < \frac{h^2}{2\,\mu} = \frac{0.2^2}{2 \times 3} = 0.0067. \tag{1}$$

In alternativa, è possibile derivare la stima di stabilità in modo più preciso, seppur più laborioso. Se si indica con U_i^n l'approssimazione di $u(x_i, t_n)$, per $i = 0, \ldots, 5$ e $n \geq 0$, il metodo numerico richiesto si scrive

$$\begin{cases} \dfrac{U_i^{n+1} - U_i^n}{\Delta t} + \dfrac{\mu}{h^2}(2U_i^n - U_{i-1}^n - U_{i+1}^n) = 0, & i = 1, \ldots, 4 \\ U_0^n = U_5^n = 0, & n \geq 0 \\ U_i^0 = 6\sin(\pi x_i), & i = 1, \ldots, 4. \end{cases}$$

In forma vettoriale, questo corrisponde a

$$\begin{cases} \dfrac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \dfrac{\mu}{h^2} A\,\mathbf{U}^n = \mathbf{0} & n \geq 0 \\ \mathbf{U}^0 = \mathbf{U}_0 = 6[\sin(\pi x_1), \sin(\pi x_2), \sin(\pi x_3), \sin(\pi x_4)]^T, \end{cases}$$

dove si è indicato con $\mathbf{U}^n = [U_1^n, U_2^n, U_3^n, U_4^n]^T$ il vettore delle incognite al tempo t_n e con A la matrice tridiag$(-1, 2, 1)$. Risolvendo per $\mathbf{U}^{n+1}$, si ottiene

$$\mathbf{U}^{n+1} = \Big(I - \frac{\mu\Delta t}{h^2}\,A\Big)\mathbf{U}^n, \quad n \geq 0, \text{ con } \mathbf{U}^0 = \mathbf{U}_0,$$

in cui I indica la matrice identità di ordine 4. Questa è una relazione ricorsiva che ha per soluzione

$$\mathbf{U}^n = \Big(I - \frac{\mu \Delta t}{h^2} A\Big)^n \mathbf{U}_0.$$

Affinché il metodo risulti stabile, è necessario che la matrice $B = I - \frac{\mu\Delta t}{h^2} A$ abbia raggio spettrale minore di uno, i.e., $\rho(B) < 1$. Dato che A è sdp, la condizione diventa

$$\rho(B) < 1 \iff \Big|1 - \frac{\mu\Delta t}{h^2}\lambda\Big| < 1 \iff -1 < 1 - \frac{\mu\Delta t}{h^2}\lambda < 1,$$

dove si è indicato con λ un generico autovalore (positivo) di A, i.e, $\lambda \in \mathrm{sp}(A)$. Risulta quindi

$$0 < \Delta t < \frac{2h^2}{\mu\lambda}, \quad \forall \lambda \in \mathrm{sp}(A). \tag{2}$$

La condizione più restrittiva, corrispondente all'autovalore λ maggiore, può essere determinata con lo script Matlab

```
h = 0.2;
mu = 3;
n = 4;
A = 2*eye(n)-diag(ones(n-1,1),-1)-diag(ones(n-1,1),1);
e = eig(A);
2*h^2/(mu*max(e))
```

e fornisce $\Delta t < 0.0074$. Tale condizione è meno restrittiva della stima (1), che di fatto corrisponde al caso limite della (2) per $n \to \infty$ con il vincolo $hn = 1$.

Esercizio - Punto 1

Il problema differenziale del testo, che qui riportiamo per averlo sotto mano:

$$\begin{cases} \dfrac{d\mathbf{y}}{dt}(t) &= A(\mathbf{y}(t))\mathbf{y}(t) + \mathbf{g}(t), \quad t \in (0, t_f) \\ \mathbf{y}(0) &= \mathbf{y}_0, \end{cases} \tag{3}$$

con $A = A(\mathbf{y}(t)) \in \mathbb{R}^{2\times 2}$, $\mathbf{g}(t) : (0, t_f) \to \mathbb{R}^2$, $\mathbf{y}_0 \in \mathbb{R}^2$ il vettore iniziale e $t_f > 0$ il tempo finale.

Introduciamo la quantità $\mathbf{u}_n \in \mathbb{R}^2$ per indicare l'approssimazione del vettore $\mathbf{y}(t_n)$, e indichiamo con N_h il numero di intervalli in cui viene suddiviso l'intervallo $[0, t_f]$, cui corrisponde il passo temporale, $h = t_f/N_h$. Il metodo di EA applicato al problema (3) risulta pertanto: trova $\mathbf{u}_{n+1}$ tale che

$$\begin{cases} \mathbf{u}_{n+1} &= \mathbf{u}_n + h(A(\mathbf{u}_n)\mathbf{u}_n + \mathbf{g}(t_n)), \quad \text{per } n = 0, \dots, N_h - 1, \\ \mathbf{u}_0 &= \mathbf{y}_0. \end{cases} \tag{4}$$

Per definire la zero-stabilità, occorre introdurre il metodo di EA perturbato: trova $\mathbf{z}_{n+1}$ tale che

$$\begin{cases} \mathbf{z}_{n+1} & = & \mathbf{z}_n + h(A(\mathbf{z}_n)\mathbf{z}_n + \mathbf{g}(t_n)) + \delta_{n+1}, \qquad \text{per } n = 0, \ldots, N_h - 1, \\ \mathbf{z}_0 & = & \mathbf{y}_0 + \delta_0, \end{cases} \tag{5}$$

dipendente dalle perturbazioni (vettoriali), δ_n, per $n = 0, \ldots, N_h$. La definizione di zero-stabilità, che ricalca quella nel caso scalare per una $f(t, y)$ generica, confronta la soluzione di (4) con quella di (5) e sarà pertanto la seguente:
Esistono $h_0 > 0$ e $\varepsilon_0 > 0$ tali che

$$\max_{n=0,\ldots,N_h} ||\mathbf{z}_n - \mathbf{u}_n|| \leq C\varepsilon,$$

sotto la condizione che $||\delta_n|| \leq \varepsilon$, per $n = 0, \ldots, N_h$, per ogni $\varepsilon \in (0, \varepsilon_0]$, per ogni $h \in (0, h_0]$, con la costante C indipendente da h, e dove $|| \cdot ||$ indica una opportuna norma vettoriale.

Esercizio - Punto 2

Lo script seguente

```
h = 1e-2;
K = 1;
A = @(y) [-1 -(147/16 + K*y(2)); 1 0];
g = @(t) [-9*exp(-t/2)*(sin(3*t)^2-1)...
          -9/2*exp(-t/4)* sin(3*t);0];
f = @(t,y) A(y)*y + g(t);
y0 = [-3/4;3];
tf = 5;
Nh = tf/h;
tv = [0, tf];
[ t, u ] = eulero_avanti_sistemi( f, tv, y0, Nh );
u(:,[2,end])
[m,mm] = min(u(2,:));
m, t(mm)
```

implementa il metodo di Eulero in avanti e permette di calcolare le quantità: $\mathbf{u}_1 = (-1.0181, 2.9925)^T$ e $\mathbf{u}_{N_h} = (-1.6865, -0.7539)^T$, $u_{2,\min} = -2.4321$ e $t_{\min} = 1.0300$.

Esercizio - Punto 3

Il calcolo degli errori può essere effettuato attraverso il seguente script

```
yex = @(t) 3*exp(-t/4)*[-1/4*cos(3*t)-3*sin(3*t);cos(3*t)];
K = 1;
A = @(y) [-1 -(147/16 + K*y(2)); 1 0];
g = @(t) [-9*exp(-t/2)*(sin(3*t)^2-1)...
        -9/2*exp(-t/4)* sin(3*t);0];
f = @(t,y) A(y)*y + g(t);
y0 = [-3/4;3];
tf = 5;
tv = [0, tf];
H = [1e-3, 5e-4, 2.5e-4, 1.25e-4];
E = zeros(1,4);
for i = 1:4
    h = H(i);
    Nh = tf/h;
    [ t, u ] = eulero_avanti_sistemi( f, tv, y0, Nh );
    E(i) = norm(u(:,end) - yex(tf));
end
format long
E
```

I valori ottenuti sono: 0.0181789, 0.0090431, 0.0045101 e 0.0022522.

Esercizio - Punto 4

Per stimare algebricamente l'ordine di convergenza del metodo numerico, basta osservare che l'errore può essere scritto come $E_h \simeq Ch^p$, per una certa costante positiva C e con p l'ordine di convergenza. Dal confronto fra gli errori associati a due passi diversi è possibile eliminare C e stimare l'ordine. Infatti

$$\frac{E_{h_1}}{E_{h_2}} \simeq \left(\frac{h_1}{h_2}\right)^p \Longrightarrow p \simeq \frac{\log \frac{E_{h_1}}{E_{h_2}}}{\log \frac{h_1}{h_2}}.$$

Applicando questa procedura al caso in esame, confrontando gli errori a due a due, otteniamo che

$$\begin{aligned}
\frac{E_{h_1}}{E_{h_2}} &= 2.0102 \Longrightarrow p \simeq \tfrac{\log 2.0102}{\log 2} = 1.0074,\\
\frac{E_{h_2}}{E_{h_3}} &= 2.0051 \Longrightarrow p \simeq \tfrac{\log 2.0051}{\log 2} = 1.0037,\\
\frac{E_{h_3}}{E_{h_4}} &= 2.0025 \Longrightarrow p \simeq \tfrac{\log 2.0025}{\log 2} = 1.0018.
\end{aligned}$$

Consideriamo come riferimento l'ultimo valore, perché associato ai valori più piccoli del passo. Possiamo quindi assumere che $p \simeq 1.0018$. Questo valore è coerente con quanto ci si aspetta dalla teoria del metodo di EA, che prevede infatti ordine 1.

Esercizio - Punto 5

Il metodo RK proposto è un metodo esplicito a due stadi con coefficienti: $b_1 = 1/4$, $b_2 = 3/4$, $c_1 = 0$, $c_2 = 2/3$, $a_{21} = 2/3$, e tutti gli altri a_{ij} nulli. In forma estesa, si scrive: assegnato $\mathbf{u}_0 = \mathbf{y}_0$, per $n = 0, \ldots, N_h - 1$, calcola

$$\begin{cases} \mathbf{K}_1 = \mathbf{f}(t_n, \mathbf{u}_n) \\ \mathbf{K}_2 = \mathbf{f}(t_n + 2h/3, \mathbf{u}_n + (2h/3)\mathbf{K}_1) \\ \mathbf{u}_{n+1} = \mathbf{u}_n + \dfrac{h}{4}(\mathbf{K}_1 + 3\mathbf{K}_2). \end{cases}$$

Si osservi che il metodo risulta correttamente definito anche per quantità vettoriali. Lo script seguente

```
clearvars
K = 1;
A = @(y) [-1  -(147/16 + K*y(2)); 1  0];
g = @(t) [-9*exp(-t/2)*(sin(3*t)^2-1)...
          -9/2*exp(-t/4)* sin(3*t);0];
f = @(t,y) A(y)*y + g(t);
y0 = [-3/4;3];
tf = 5;
h = 1e-2;
Nh = tf/h;
th = linspace(0,tf,Nh+1);
K1 = @(t,y) f(t,y);
K2 = @(t,y) f(t + 2/3*h, y + 2/3*h*K1(t,y));
u = zeros(2,Nh+1);
u(:,1) = y0;
for i = 1:Nh
    u(:,i+1) = u(:,i) + h*(K1(th(i),u(:,i)) ...
                        + 3*K2(th(i),u(:,i)))/4;
end
u(:,[2,end])
```

implementa il metodo proposto e fornisce $\mathbf{u}_1 = (-1.0171, 2.9912)^T$ e $\mathbf{u}_{N_h} = (-1.5104, -0.6534)^T$.

Esercizio - Punto 6

Con $K = 0$ e $\mathbf{g} = \mathbf{0}$, la funzione matriciale A si riduce ad una matrice algebrica costante, $A = [-1, -147/16; 1, 0]$ e il problema differenziale (3) diventa

$$\begin{cases} \dfrac{d\mathbf{y}}{dt}(t) & = & A\,\mathbf{y}(t), \quad t \in (0, t_f) \\ \mathbf{y}(0) & = & \mathbf{y}_0, \end{cases} \tag{6}$$

che rappresenta un sistema di equazioni differenziali di ordine uno a coefficienti costanti. Il metodo di Heun applicato a tale problema si scrive

$$\begin{cases} \mathbf{u}^*_{n+1} & = & \mathbf{u}_n + h\,A\,\mathbf{u}_n \\ \mathbf{u}_{n+1} & = & \mathbf{u}_n + \dfrac{h}{2}\big(A\mathbf{u}_n + A\mathbf{u}^*_{n+1}\big), \quad n \geq 0 \\ \mathbf{u}_0 & = & \mathbf{y}_0. \end{cases}$$

Sostituendo $\mathbf{u}^*_{n+1}$ dalla prima nella seconda relazione, si ottiene

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \frac{h}{2}\Big(A\mathbf{u}_n + A\big(\mathbf{u}_n + h\,A\,\mathbf{u}_n\big)\Big) = \Big(I + hA + \frac{(hA)^2}{2}\Big)\mathbf{u}_n,$$

dove I è la matrice identità di ordine due. Questa relazione ricorsiva ha come soluzione

$$\mathbf{u}_n = \Big(I + hA + \frac{(hA)^2}{2}\Big)^n \mathbf{y}_0, \quad n \geq 0.$$

La condizione di assoluta stabilità per il metodo di Heun equivale dunque a richiedere che $\rho(B) < 1$ con $B = I + hA + \dfrac{(hA)^2}{2}$, essendo $\rho(\cdot)$ il raggio spettrale. Si ha quindi

$$\rho(B) < 1 \iff \left|1 + h\lambda + \frac{(h\lambda)^2}{2}\right| < 1 \quad \forall\lambda \in \mathrm{sp}(A).$$

Dato che gli autovalori della matrice A sono complessi coniugati, conviene passare a Matlab. Lo script

```
A = [-1  -147/16;  1  0];
e = eig(A);
h = linspace(0,1,1000);
R = abs(1 + h*e(1) + (h*e(1)).^2/2);
plot(h,R)
grid
```

costruisce la Figura 1 in cui è rappresentata la quantità $R(h\lambda) = \left|1 + h\lambda + \frac{(h\lambda)^2}{2}\right|$ al variare di h, per il primo autovalore di A (il secondo autovalore produrrebbe lo stesso grafico). Zoomando nell'intorno dell'ascissa 0.4 si può stimare che l'attraversamento della quota unitaria si ha per $h \simeq 0.4247$. In definitiva, la condizione di assoluta stabilità per il metodo di Heun è $0 < h < 0.4247$.

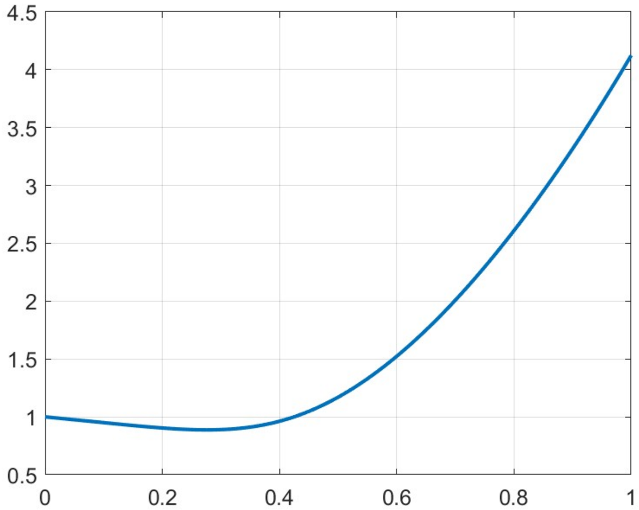

Figura 1: Grafico di $R(h\lambda)$ per il metodo di Heun.

Esercizio - Punto 7

L'applicazione del metodo di Crank-Nicolson al sistema (6) fornisce

$$\begin{cases} \mathbf{u}_{n+1} & = & \mathbf{u}_n + \dfrac{h}{2}\big(A\mathbf{u}_n + A\mathbf{u}_{n+1}\big), \quad n \geq 0 \\ \mathbf{u}_0 & = & \mathbf{y}_0, \end{cases}$$

che definisce il vettore $\mathbf{u}_{n+1}$ come soluzione del sistema lineare

$$\big(I - \frac{h}{2}A\big)\mathbf{u}_{n+1} = \big(I + \frac{h}{2}A\big)\mathbf{u}_n, \qquad \text{con } \mathbf{u}_0 = \mathbf{y}_0.$$

Si osservi che la matrice dei coefficienti di tali sistemi, $M_1 = I - \frac{h}{2}A$, non cambia al variare di n. Una strategia computazionalmente efficiente per risolvere tali sistemi lineari consiste quindi nel fattorizzare tale matrice una volta per tutte prima del ciclo `for`. Dato che M_1 non è sdp, ma è 2×2, e quindi tridiagonale, la fattorizzazione più opportuna è quella di Thomas che, però, in questo caso coincide con la fattorizzazione LU (si trova inoltre che non avviene pivotazione, cioè si ha $P = I$). Lo script

```
clearvars
A = [-1  -147/16; 1  0];
h = 1e-2;
tf = 5;
```

```
Nh = tf/h;
th = linspace(0,tf,Nh+1);
u = zeros(2,Nh+1);
y0 = [-3/4; 3];
u(:,1) = y0;
M1 = eye(2) - h/2*A;
M2 = eye(2) + h/2*A;
[L,U,P] = lu(M1); % P = I, in questo caso
for i = 1:Nh
    aux      = fwsub(L,P*(M2*u(:,i)));
    u(:,i+1) = bksub(U,aux);
end
u(:,[2,end])
```

implementa il metodo proposto e fornisce $\mathbf{u}_1 = (-1.0164, 2.9912)^T$ e $\mathbf{u}_{N_h} = (-0.4706, -0.1642)^T$.