Esercizio Diffusione
Corso di LSMC, a.a. 2019-2020

Cristian Sopio
559597

28 gennaio 2022

1 Descrizione del problema

L'obiettivo è quello di risolvere un problema di diffusione su $[0,1]^2$ con condizioni di Neumann nulle al bordo, un po' diverso da quello visto a lezione. L'equazione non lineare di diffusione è

$$\begin{cases} \frac{\partial u}{\partial t} = K(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + g(u) \\ u(o,x,y) \equiv u_0(x,y) \end{cases}$$

con $g(u) = u(1-u)(u-1/2)$, $K = 10^{-4}$ e con dato iniziale discreto $U = 0.5 + \text{rand}(n)$.

2 Descrizione della sperimentazione

Per risolvere il problema abbiamo modificato come nelle slide la function `diffusione`.

3 Script e function

Si riportano di seguito le function utilizzate nella sperimentazione.

Function 1

```
function diffusione3(n)
%DIFFUSIONE_CALORE

% Discretizzazione del dominio nelle x e nelle y.
x = linspace(0, 1, n);
y = x;
```

```
% Passo di discretizzazione spaziale
h = x(2) - x(1);

% Coefficiente di diffusione per il nostro operatore laplaciano
K = 1e-4;

% Step di integrazione in tempo
dT = 0.01;

% Calcoliamo quanti step mi servono ad arrivare a t = 5.
nt = round(5 / dT);

% La sorgente di calore Ã attiva solo fino a tempo 1/2, in un angolo del
% dominio.
g=@(u) u .* (1-u) .* (u-1/2);
% Valutiamo il dato iniziale, nel nostro caso una campana esponenziale
% centrata in mezzo al dominio; per questo ci serve una griglia di punti
% che puÃ essere generata con il comando meshgrid.

u0 = 0.5+randn(n);



U = u0;

mesh(x, y, U);

% Costruiamo l'operatore Laplaciano come una matrice sparsa, per rendere i
% conti sufficientemente veloci. Questo Ã di vitale importanza!
A = K * spdiags(ones(n, 1) * [1 -2 1] , -1:1, n, n) / h^2 ;
A(1,1)=-K/h^2;
A(n,n)=-K/h^2;
AA = kron(speye(n), A) + kron(A, speye(n));
% Cominciamo ad integrare il sistema usando Eulero implicito
for j = 1 : nt
    % Risolvo il sistema (I - dT * L) u_{n+1} = u_n + dT * f + dT * BC,
    % dove f è il termine sorgente
    rhs = U(1:end, 1:end) + ...
        dT * g(U);

    Unew = (speye(size(AA)) - dT * AA) \ reshape(rhs, (n)^2, 1);

    U(1:end,1:end) = reshape(Unew, n, n);

    % Plot della soluzione
    mesh(x, y, U);
```

```
    title(sprintf('Time = %f', j * dT));
    axis([0 1 0 1 0 1])
    drawnow;
    pause(0.1);
end
end
```

4 Immagini e Commenti

Si vede che modificando K abbiamo effetti diversi, ad esempio per $K = 10^{-4}$, la g predomina, invece per $K = 10^{-2}$ la g permette la diffusione e si raggiunge quasi l'equilibrio.
Infine per K più grandi la diffusione domina su g e le soluzioni si appiattiscono sempre di più.