

**Final Project 1.**

Write a Matlab function

```
[U]=my_FD_nonhom_heat (a,b,T,N,M,g,f)
```

that solves the problem

$$\begin{aligned}\frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) &= f(x,t), \quad a < x < b, \quad t > 0 \\ u(a,t) &= 0, \quad t > 0, \\ \frac{\partial u}{\partial x}(b,t) &= 0, \quad t > 0, \\ u(x,0) &= g(x), \quad a < x < b.\end{aligned}$$

by using Crank-Nicolson scheme. The input for the function is the following:

- $a, b$  – endpoints for the interval  $(a, b)$
- $T$  – final time
- $N + 1$  – number of the space points in the method
- $M + 1$  – number of the time levels
- $g(x)$  – inline function of the initial condition
- $f(x, t)$  – inline function of two variables  $x$  and  $t$  (for the RHS).

The output should be the  $(N + 1) \times (M + 1)$  matrix  $U$ .

```
% Final_project1_check

a = 0; b = 3/2; % define interval (a,b)
T = 1; % define final time T
N = 20; % number of space points
M = 20; % number of time levels
% initial and RHS functions
g = inline('sin(pi*x)','x');
f = inline('sin(pi*x)*((pi^2)*cos(3*t)-3*sin(3*t))','x','t');

% computing solution by Crank-Nicolson method
U = my_FD_nonhom_heat(a, b, T, N, M, g, f);

x = linspace(a,b,N+1);
% exact solution
U_ex = cos(3*T)*sin(pi*x);
% plotting exact and the approximate solution
figure(1)
plot(x,U(:,M+1),'r*', x,U_ex);

% to plot an animated heat evolution
t = linspace(0,T,M+1);
```

```

figure(2)
surf(x,t,U')

% check space rate of convergence
fprintf(['   dx           error           space rate of convergence           \n'] )
disp('-----')

error = zeros(10,1);
M = 1000;
N = 10;
U = my_FD_nonhom_heat(a, b, T, N, M, g, f);
x = linspace(a,b,N+1)';
U_ex = cos(3*T)*sin(pi*x);
error(1) = norm(U(:,M+1)-U_ex,inf);
for j = 2:5
    N = 10*2^(j-1);
    U = my_FD_nonhom_heat(a, b, T, N, M, g, f);
    x = linspace(a,b,N+1)';
    U_ex = cos(3*T)*sin(pi*x);
    error(j) = norm(U(:,M+1)-U_ex,inf);
    dx = 2/(10*2^(j-1));
    fprintf([' %1.2e %20.3e %20.5f \n'],...
        dx, error(j), log2(error(j-1)/error(j)) )
end
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
% check time rate of convergence
fprintf(['   dx           error           time rate of convergence           \n'] )
disp('-----')

error = zeros(10,1);
M = 10;
N = 1000;
U = my_FD_nonhom_heat(a, b, T, N, M, g, f);
x = linspace(a,b,N+1)';
U_ex = cos(3*T)*sin(pi*x);
error(1) = norm(U(:,M+1)-U_ex,inf);
for j = 2:5
    M = 10*2^(j-1);
    U = my_FD_nonhom_heat(a, b, T, N, M, g, f);
    x = linspace(a,b,N+1)';
    error(j) = norm(U(:,M+1)-U_ex,inf);
    dx = 2/(10*2^(j-1));
    fprintf([' %1.2e %20.3e %20.5f \n'],...
        dx, error(j), log2(error(j-1)/error(j)) )
end

```

Make sure that the names of your functions match. The above script can be uploaded from the course website. E-mail me your function. Print and submit the script of your function, the tables and the figures produced from running the test script

**Bonus question:** How would you speed up the computations?