TMA373 Assignment 5

Edoardo Mangia
edoardom@student.chalmers.se

Bibiana Farinha
bibiana@student.chalmers.se

February 2025

Contents

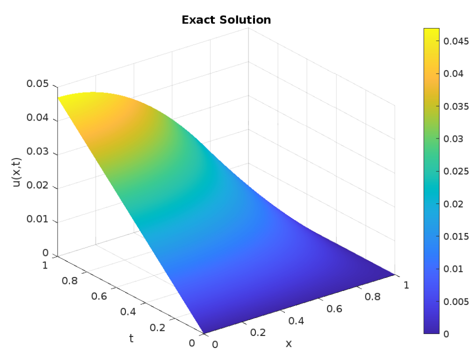
1 Task 1

In this assignment, we will implement and study the FE approximation of the heat equation.

$$\begin{cases} u_t(x,t) + u_{xx}(x,t) = f(x,t), & 0 < x < 1,\ 0 < t < T, \\ u_x(0,t) = 0, \quad u(1,t) = 0, & 0 < t < T, \\ u(x,0) = u_0(x), & 0 < x < 1. \end{cases} \tag{1}$$

To be able to test our implementation, we will need to first calculate $f(x,t)$ and $u_0(x)$. Assume that $u(x,t) = -\frac{\text{epsSTUD}}{2}(tx^2 - t)$, with $\text{epsSTUD} = 47/10^3$, from (1), we can determine the functions:

$$f(x,t) = \frac{\text{epsSTUD}}{2}(1 - x^2 - 2t), \quad u_0(x) = 0.$$



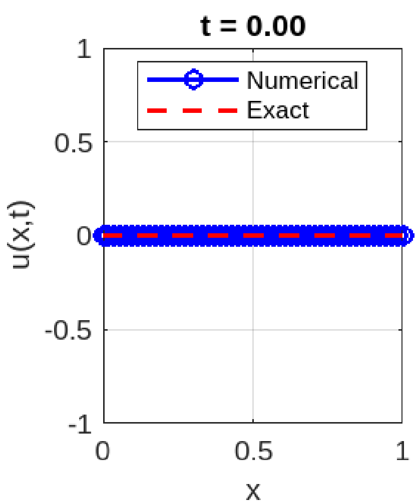
From the lessons, we know the values of the mass matrix M and the stiffness matrix S. In the implementation below we calculate an approximation of the load vector F by using the trapezoidal rule for each of the elements of the vector. The only task missing is solving the linear equation (12) for each time step.

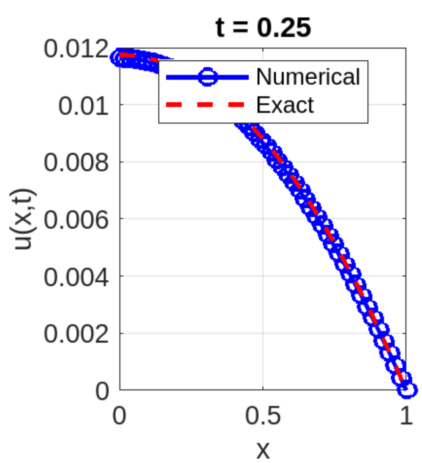
Implementation

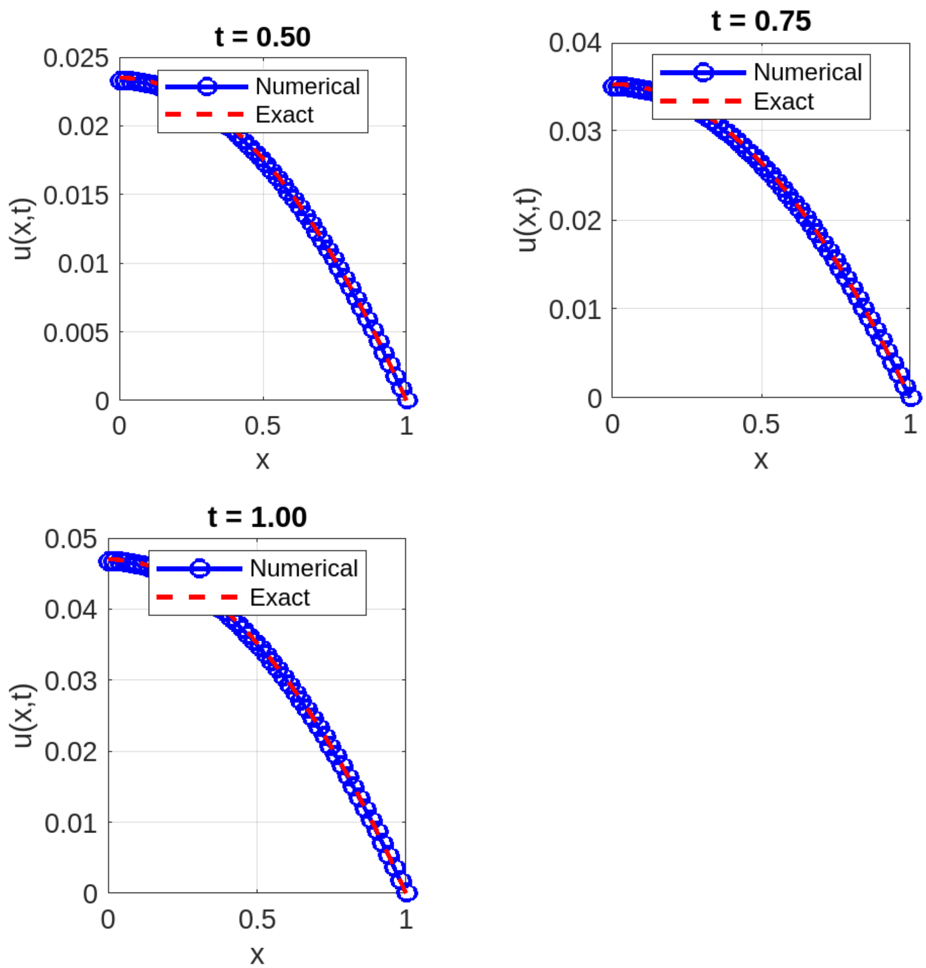
```
% Template for linear heat equation on (0,1)
% Neumann BC at x=0, Dirichlet BC at x=1

% number of nodes and mesh size and grid points
m = 50;
h = 1/(m+1);
x = 0:h:1;

% time interval, nbr of steps, stepsize
T = 1; n = 100;
k = T/n;
t = linspace(0, T, n+1);

% for the solution matrix with all discrete space and time points (inc. BC)
U = zeros(m+2, n+1);

% IC and BC for time t=0
epsSTUD = 47/10^3;
U(:,1) = 0;
U(end,1) = 0;

% construction of mass and stiffness matrices
M = zeros(m+1, m+1);
S = zeros(m+1, m+1);
M_local = (h/6)*[2 1; 1 2];
S_local = (1/h)*[1 -1; -1 1];
for elem = 1:m+1
    if elem <= m+1 && elem+1 <= m+1
        M(elem:elem+1, elem:elem+1) = M(elem:elem+1, elem:elem+1) + M_local;
        S(elem:elem+1, elem:elem+1) = S(elem:elem+1, elem:elem+1) + S_local;
    elseif elem <= m+1
        M(elem, elem) = M(elem, elem) + M_local(1,1);
        S(elem, elem) = S(elem, elem) + S_local(1,1);
    end
end

A = M + k * S;

% time discretisation
for l = 1:n
    tl = t(l+1);
    f_fun = @(x) -epsSTUD*(x.^2 - 1) + 2*epsSTUD*tl;
    F_vec = zeros(m+1,1);
    for i = 1:(m+1)
        if i == 1
            xi_local = x(1:2);
```

```
            phi = @(xi) (xi - x(1))/h;
            F_vec(i) = trapz(xi_local,
                f_fun(xi_local).*phi(xi_local));
        elseif i == m+1
            xi_local = x(m+1:end);
            phi = @(xi) (x(end) - xi)/h;
            F_vec(i) = trapz(xi_local,
                f_fun(xi_local).*phi(xi_local));
        else
            xi_local1 = x(i:i+1);
            phi1 = @(xi) (xi - x(i))/h;
            I1 = trapz(xi_local1,
                f_fun(xi_local1).*phi1(xi_local1));
            xi_local2 = x(i+1:i+2);
            phi2 = @(xi) (x(i+2) - xi)/h;
            I2 = trapz(xi_local2,
                f_fun(xi_local2).*phi2(xi_local2));
            F_vec(i) = I1 + I2;
        end
    end
    xi_old = U(1:m+1, l);
    xi_new = A \ (M*xi_old + k * F_vec);
    U(1:m+1, l+1) = xi_new;
    U(end, l+1) = 0;
end

% ... plotting
```

Visualization






t = 0.50
t = 0.75
t = 1.00
Numerical
Exact
u(x,t)
x
0.025
0.02
0.015
0.01
0.005
0.05
0.04
0.03
0
0.5
1