Introduction to Biological Imaging (Homework 2)

Johann Brenner, Mat. 03662733

Winter term 2018/19

1 Assignment 1. Finite Difference Method

1.1 Come up with a simple one dimensional differential equation $\mathbf{f'(x) = g(x)}$ on $\mathbf{\Omega = [0, 1]}$ by specifying $\mathbf{g(x)}$ and setting appropriate boundary condition (e. g. $\mathbf{f(0) = 0}$).

Initial assumption:

$$g(x) = \cos^2(\pi x) \tag{1}$$

Dirichlet boundary condition:

$$f(0) = 0 \tag{2}$$

1.2 Solve the obtained boundary value problem analytically. Show your solution as well as its plot.

$$\int f'(x)\ dx = \int \cos^2(\pi x)\ dx \tag{3}$$

$$f(x) = \frac{x}{2} + \frac{\sin(2\,\pi\,x)}{4\,\pi} + c_1 \tag{4}$$

$$f(0) = \frac{0}{2} + \frac{\sin(2\,\pi\,0)}{4\,\pi} + c_1 \overset{!}{=} 0 \tag{5}$$

$$\Rightarrow c_1 = 0 \tag{6}$$

The corresponding MATLAB code was imeplemented as follwing:

```
x_cont = linspace(0,1,1000);
x_interval = [0,1];
%% Plot ODE f'(x) = cos(pi x)^2 and its solution
syms f(x)
ODE = diff(f,x) == (cos(pi.*x)).^2;
bc = f(0) == 0;
sol = dsolve(ODE,bc);

figure;
yyaxis right
plot(x_cont,(cos(pi.*x_cont)).^2,'DisplayName',['ODE (f''(x) =' ...
    'cos(x \pi)^2)']); hold on
xlabel('x');
ylabel('f''(x)');
yyaxis left
fplot(sol,x_interval,'DisplayName',['Analytical Solution ' ...
    '(f(0)=0, ' newline 'f(x)=^{x}/_{2}+sin(2\pix)/4\pi )']);
legend;
xlabel('x');
ylabel('f(x)');
```

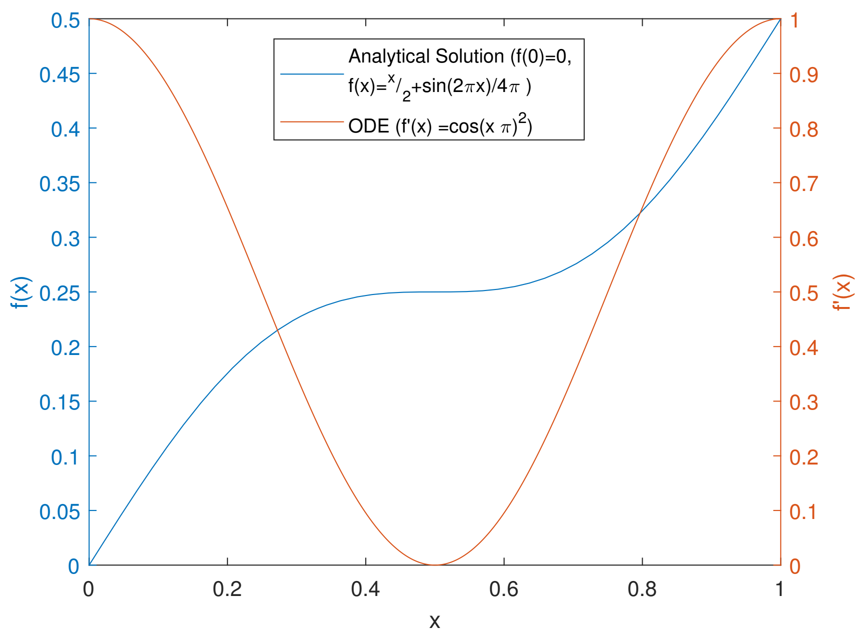

Figure 1: Original ODE with analytical solution

1.3 Solve the boundary value problem using the Finite Difference Method for step h = 0.2, 0.1 and 0.01. Pick the discretization scheme yourself (forward/backward/central difference), discretize the equation and put it into matrix form. Use linsolve() to solve the resulting system of linear equations. Hint: unless you know what you are doing, avoid central difference scheme.

Choosing the *forward Method* for the Finite Differences Method, where $x_i = i \cdot h$ with $i = 1, 2, ..., N+1$ and $N + 1 = \frac{1}{h}$. For given h values of 0.2, 0.1 and 0.01 the dimension M is 5, 10 and 100.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{7}$$

$$\frac{f(x+h) - f(x)}{h} = \cos^2(\pi x) \tag{8}$$

$$\text{Discretizing}: \frac{f(x_{i+1}) - f(x_i)}{h} = \cos^2(\pi x_i) \tag{9}$$

$$\text{Rewriting}: f(x_{i+1}) - f(x_i) = h\cos^2(\pi x_i) \tag{10}$$

$$\text{Using Eq. 2}: \; f_1 = f(0) = 0 \tag{11}$$

$$\underbrace{\begin{bmatrix} 1 & & & & & 0 \\ -1 & 1 & & & & \\ & & \ddots & \ddots & & \\ & & & -1 & 1 & \\ 0 & & & & -1 & 1 \end{bmatrix}}_{M \times M} \underbrace{\begin{bmatrix} f_2 \\ f_3 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}}_{M \times 1} = h \underbrace{\begin{bmatrix} \cos^2(\pi x_1) \\ \cos^2(\pi x_2) \\ \vdots \\ \cos^2(\pi x_{N-2}) \\ \cos^2(\pi x_{N-1}) \end{bmatrix}}_{M \times 1} \tag{12}$$

The Finite Differences were computed by the following MATLAB script:

```
%% Finite Difference Method
% Initialize Matrices
h = [0.2 0.1 0.01];
mat_size = 1./h;
A = cell(length(mat_size),1);
b = cell(length(mat_size),1);
f = cell(length(mat_size),1);
x = cell(length(mat_size),1);
fin_diff = cell(length(mat_size),1);

figure;
fplot(sol,x_interval,'DisplayName','Analytical Solution');hold on
% Populate A and b
for i = 1:length(mat_size)
    x{i} = linspace(0,1,mat_size(i))';
    A{i} = eye(mat_size(i));
    for j = 2:(mat_size(i))
        A{i}(j,j-1) = -1;
    end

    b{i} = double(h(i) .* ones(mat_size(i),1) .* (cos(pi .* x{i})).^2);

    % Evaluate Af = b
    f{i} = linsolve(double(A{i}),b{i});

    % Plot solution
    plot(x{i},[0;f{i}(1:end-1)],'DisplayName',['Finite Difference h = '
        num2str(h(i))]);
end
legend({},'Location','northwest');
xlabel('x_i');
ylabel('f(x_i)');
```

1.4 Plot the resulting solutions along with the exact solution you have computed in step 2

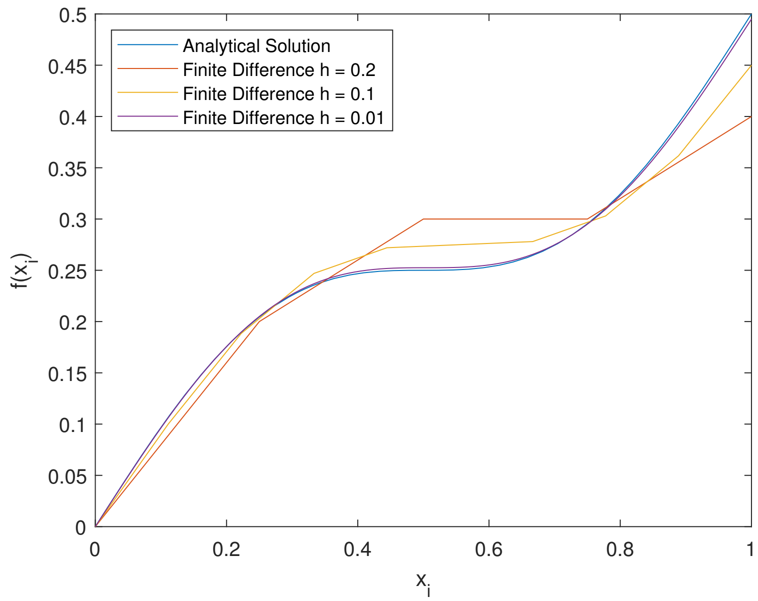

Figure 2: Analytical and Finite Difference solutions of the ODE

2 Assignment 2. Finite Element Method

2.1 Derive quadratic interpolation functions for the finite element method. Plot them in local coordinates. Show your derivation.

As shown in the lecture, for the square shape functions, we can write our trial solution as:

$$\widehat{T} = a_0 + a_1 x + a_2 x^2 \tag{13}$$

$$= \underbrace{\begin{bmatrix} 1 & x & x^2 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}^T}_{A} \tag{14}$$

$$T_{i_1} = T(0) = a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 \tag{15}$$

$$T_{i_2} = T(1/2) = a_0 + \frac{a_1}{2} + \frac{a_2}{2^2} \tag{16}$$

$$T_{i_3} = T(1) = a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 \tag{17}$$

$$\underbrace{\begin{bmatrix} T_{i_1} \\ T_{i_2} \\ T_{i_3} \end{bmatrix}}_{T} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2^2} \\ 1 & 1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}}_{A} \tag{18}$$

$$XA = NLA \tag{19}$$

$$N = XL^{-1} \tag{20}$$

$$= \begin{bmatrix} 2\,x^2 - 3\,x + 1 & 4\,x - 4\,x^2 & 2\,x^2 - x \end{bmatrix} \tag{21}$$

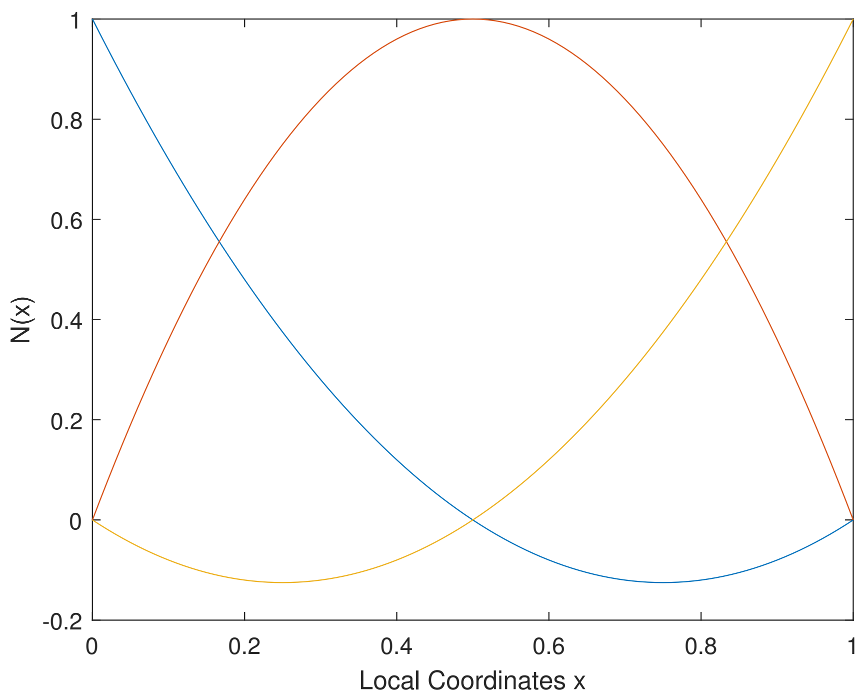

Figure 3: Quadratic FEM Interpolation Functions

2.2 Consider a 1D steady state heat conduction problem, where k and S are constant. How would the stiffness matrix for one element look like for the shape functions derived in step 1?

Problem:

$$kT'' + S = 0 \text{ on } \Omega \in [0,1] \tag{22}$$

$$T(0) = 0,\ T(1) = 0 \tag{23}$$

$$k_{stiff}(i,j) = \int_0^1 N'(i) \cdot N'(j) dx \tag{24}$$

$$k_{stiff} = \begin{bmatrix} 2.\bar{3} & -2.\bar{6} & 0.\bar{3} \\ -2.\bar{6} & 5.\bar{3} & -2.\bar{6} \\ 0.\bar{3} & -2.\bar{6} & 2.\bar{3} \end{bmatrix} \tag{25}$$

```
%% 3D Finite Element Method

% Local x
x_local = 0:0.01:1;
L = [1,0,0; 1,0.5,0.25;1,1,1];
L_inv = inv(sym(L));

syms x
X = [1,x,x^2];
N = X*L_inv;

figure; %visualize cubic shape functions
for i = 1:length(N)
    plot(x_local, subs(N(i),x_local),'DisplayName',['Square Shape Function
        ' num2str(i)]); hold on;
end
xlabel('Local Coordinates x');
ylabel('N(x)');
%% Stiffness Matrix
k_stiff = zeros(length(N));

for i = 1:length(N)
    for j = 1:length(N)
        k_stiff(i,j) = double(int( diff(N(i), x).*diff(N(j), x), x, 0, 1 ))
            ;
    end
end

nodes = 3; % number of elements
Ke = double(k_stiff);
RHSe = int(N, x, 0, 1)';


L = 1; % domain length
k = 1; %for k = [100,10,1,0.1] %; % thermal conductivity,1
S = 10; %     for S = [100,10,1,0.1]% % source,100
l = L/nodes; % node spacing assuming equidistant nodes
dim = nodes*(nodes - 1) +1; %dimension of stiffness matrix
```

```
K = zeros(dim); %initialize stiffness matrix with zeros
RHS = zeros([dim, 1]); %initialize load vector with zeros

for i = 1:nodes %cycle through elements

    %compute indices of entries for a given element
    ind_start = 1+(i-1)*(nodes-1);
    ind_end = ind_start + (nodes-1);

    % update stiffness matrix with the stiffness matrix of a single element
    K(ind_start:ind_end, ind_start:ind_end) = K(ind_start:ind_end, ...
        ind_start:ind_end) + Ke;

    %update load vector
    RHS(ind_start:ind_end) = RHS(ind_start:ind_end)+RHSe;

end

K = (k/l)*K; %multiply by constants
F = S*l*RHS;
large_number = 1e6; %a trick not to exclude T(0)
K(1,1) = K(1,1)+large_number;
T_FEM = linsolve(K, F)% solve for T
%
x_coord_FEM = 0:1/(nodes-1):1; %coordinates of nodes
figure;
plot(x_coord_FEM, T_FEM,'k--','DisplayName',['FEM, S=',num2str(S),' k=',...
    num2str(k)]); hold on %plot FEM solution
x_coord_solution = 0:0.01:1; %a grid to compute direct solution on
solution = -S/(2*k)*x_coord_solution.^2+S/k*x_coord_solution; %values of the direct
plot(x_coord_solution, solution,'DisplayName', ['Solution, S=',num2str(S),' k=',...
    num2str(k)]); hold on %visualize the direct solution
xlabel('x');
ylabel('T(x)');
legend({},'Location','northwest');
```

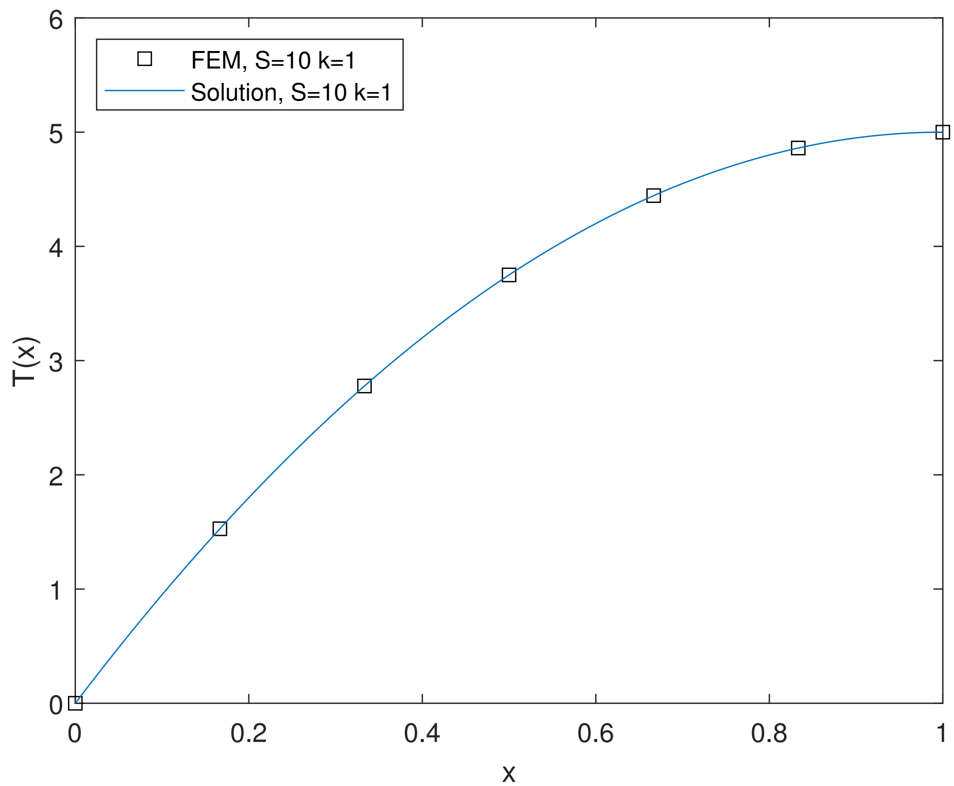

Figure 4: Analytical Solution and FEM Approximation