

```
1: % Sheet 4 / Task B
2:
3: ne = 50;      % number of elements
4: nodes = linspace(0,1,ne+1);
5:
6: K = zeros(ne+1,ne+1); % global stiffness matrix
7: f = zeros(ne+1,1);   % load vector
8:
9: lambda = @(x) (x<1/sqrt(2))*1 + (x>=1/sqrt(2))*10; ✓
10:
11: for e = 1:ne
12:     xleft = nodes(e);
13:     xright = nodes(e+1);
14:     Ke = ne^2*[integral(lambda,xleft,xright), -integral(lambda,xleft,xright);
15:               -integral(lambda,xleft,xright), integral(lambda,xleft,xright)];
16:
17:     K(e:e+1, e:e+1) = K(e:e+1, e:e+1) + Ke;
18: end
19:
20: % disp(K)
21:
22: % adaption for dirichlet boundary
23: K(1,1) = K(1,1)*(1 + 1e6);
24: f(end) = K(end,end)*1e6;
25: K(end,end) = K(end,end)*(1 + 1e6);
26:
27: u = K\f;
28:
29: plot(nodes,u)
30: title('approximation of solution u');
31: xlabel('x values');
32: ylabel('u_h');
```

benihgt ✓  $x_1 < x_2 < x_3 \dots < x_{ne+1}$

```
1: % Sheet 4 / Task C
2: clf
3:
4: nvec = [10, 20, 30, 40, 70]; % vector with number of el
ements
5: p = 70; % parameter of ode
6:
7: for k=1:length(nvec)
8:     ne = nvec(k);
9:     nodes = linspace(0,1,ne+1);
10:
11:     K = zeros(ne+1,ne+1); % global stiffness matrix
12:     f = zeros(ne+1,1); % load vector
13:
14:     for e = 1:ne
15:         xleft = nodes(e);
16:         xright = nodes(e+1);
17:
18:         % first part of the bilinearform phi'*phi'
19:         funcone = @(x) 1 + 0*x;
20:         Ke = ne^2*[integral(funcone,xleft,xright), -integr
al(funcone,xleft,xright);
21:                 -integral(funcone,xleft,xright), integral(func
one,xleft,xright)];
22:
23:         % second part: phi'*phi
24:         func1 = @(x) xright - x;
25:         func2 = @(x) x - xleft;
26:         Ke = Ke + p*ne^2*[-integral(func1,xleft,xright), i
ntegral(func1,xleft,xright);
27:                           -integral(func2,xleft,xright), int
egral(func2,xleft,xright)];
28:
29:         K(e:e+1, e:e+1) = K(e:e+1, e:e+1) + Ke;
30:     end
31:
32:     % disp(K)
33:
34:     % adaption for dirichlet boundary
35:     K(1,1) = K(1,1)*(1 + 1e6);
36:     f(end) = K(end,end)*1e6;
37:     K(end,end) = K(end,end)*(1 + 1e6);
38:
39:     u = K\f;
40:
41:     hold on
42:     plot(nodes,u)
43:     title('approximation of solution u for p=70');
44:     xlabel('x values');
45:     ylabel('u_h');
```

```
46:     grid on;
47:     leg = legend(num2str(nvec'), 'Location', 'Northwest');
48:     title(leg, 'n =');
49:
50: end
51: saveas(gcf, 'ex_4_C_plot.jpg');
```

*Sehr gut*