```
clear
clc
disp('Finite Differenzen:');
u = @(x) x.*(x-1).*(2*x-1);
p = @(x) 1-(x-0.5).^2;
q = @(x) 1+(x-0.5).^2;
f = @(x) (14 + x.*(25 + x.*(-163 + x.*(122 + x.*(-20 + 8*x)))))/4;
erI = [];
erA = [];
N   = 32;
l   = 5;

for i = 1:l
      N = 2*N;
      h = 1/N;
      x = linspace(0,1,N+1)'; %Schnittstellen
      x_= zeros(N,1);
      for k = 1:N
            x_(k) = (x(k+1)+x(k))/2; %Zellzentren mit Randzellen
      end
      xx = x(2:N,1); %Schnittstellen ohne Randpunkte
      Q = spdiags([-ones(N-1,1),ones(N-1,1)],[0,1],N-1,N)/h;
      Dp = spdiags(p(x_),0,N, N);     %Diagp in Zellzentren
      Dq = spdiags(q(xx),0,N-1,N-1);  %Diagq in Schnittstellen
      A = Q*Dp*Q'+Dq;
      ff = f(xx);                     %f mit inneren Schnittstellen
      ut = A\ff;
      du = u(xx) - ut;
      Adu = A*du;

      subplot(2,4,1); hold on;               % grafische Darstellung der Loesung
      plot(xx,u(xx),'b',xx,ut)
      subplot(2,4,2); hold on;
      plot(xx,abs(du))
      subplot(2,4,3); hold on;
      plot(xx,abs(Adu))

      erI = [erI,sqrt(h)*norm(du)];          % 2,h-Norm
      erA = [erA,sqrt(du'*Adu)];             % A-Norm
  end
  deI = erI(1:(l-1))./erI(2:l);
  deA = erA(1:(l-1))./erA(2:l);
  ordI = log2(mean(deI));                    % Abschaetzung der Ordnung a)
  ordA = log2(mean(deA));                    % b)

  fprintf('Ord in 2,h-Norm = %0.10f\n',ordI);
  fprintf('Ord in A-Norm   = %0.10f\n',ordA);

  subplot(2,4,1)                             % grafische Darstellung der Fehler
  title('Finite Differenzen')
  subplot(2,4,2)
  title('Fehler')
  subplot(2,4,3)
  title('A*Fehler')
  subplot(2,4,4)
  plot(1:l,log2(erI),1:l,log2(erA))
  legend('erI','erA')
  title('Ordnung der Abschätzung')

disp('Finite Elemente:');
erI2 = [];
erA2 = [];
N   = 32;
l   = 5;
for i=1:l
    N   = 2*N;                             % N = 64, 128, 256, 512, 1024
    h   = 1/N;
    x = linspace(0,1,N+1)'; %Schnittstellen
    x_= zeros(N,1);
    for k = 1:N
        x_(k) = (x(k+1)+x(k))/2; %Zellzentren mit Randzellen
    end
```

```
    xx = x(2:N,1); %Schnittstellen ohne Randpunkte

    Q   = spdiags([-ones(N,1),ones(N,1)],[0,1],N-1,N)/h;
    R   = spdiags(ones(N,2),[0,1],N-1,N);  % Durchschnittsabbildung

    Dq  = spdiags(q(x_),0,N,N);            % diag(q) in Zellzentren
    Dp  = spdiags(p(x_),0,N,N);            % diag(p) in Zellzentren
    T   = spdiags(R*q(x_),0,N-1,N-1);
    F   = R*f(x_)/2;                      % f in Zellzentren

    A   = Q*Dp*Q' + (R*Dq*R' + T)/6;
    ut  = A\F;                            % Approximation in inneren Schnittstellen

    du  = u(xx)-ut;                       % Fehler in inneren Schnittstellen
    Adu = A*du;

    subplot(2,4,5); hold on;              % grafische Darstellung der Loesung
    plot(xx,u(xx),'b',xx,ut)
    subplot(2,4,6); hold on;
    plot(xx,abs(du))
    subplot(2,4,7); hold on;
    plot(xx,abs(Adu))

    erI2 = [erI2,sqrt(h)*norm(du)];          % 2,h-Norm
    erA2 = [erA2,sqrt(du'*Adu)];             % A-Norm
end

deI = erI2(1:(l-1))./erI2(2:l);
deA = erA2(1:(l-1))./erA2(2:l);

subplot(2,4,5)                             % grafische Darstellung der Fehler
title('Finite Elemente')
subplot(2,4,6)
title('Fehler')
subplot(2,4,7)
title('A*Fehler')
subplot(2,4,8)
plot(1:l,log2(erI),1:l,log2(erA2))
legend('erI','erA')
title('Ordnung der Abschätzung')
ordI = log2(mean(deI));                    % Abschaetzung der Ordnung c)
ordA = log2(mean(deA));                    %d)
fprintf('Ord in 2,h-Norm = %0.10f\n',ordI);
fprintf('Ord in A-Norm   = %0.10f\n',ordA);
```

Christoph Raunjak	Blatt 12	Numerik I SS20

Aufgabe 4

☑ a.) ☑ b.) ☐ c.) ☐ d.)

a.)

Numerik:

4.a) z.z.: $\lim\limits_{\varepsilon \to 0} A_\varepsilon(u, s_h) = \lim\limits_{\varepsilon \to 0} F_\varepsilon(s_h) = s_h(\tfrac{1}{2}) \quad \forall s_h \in S_0^1(\Omega) \quad \Omega = (0,1)$

$$\lim_{\varepsilon \to 0} A_\varepsilon(u, s_h) = \lim_{\varepsilon \to 0} \int_0^1 [\varepsilon \cdot u \cdot s_h + p \cdot u' \cdot s_h'] \, dx = \underbrace{\lim_{\varepsilon \to 0} \varepsilon \int_0^1 u \cdot s_h \, dx}_{0} + \int_0^1 p \cdot u' \cdot s_h' \, dx =$$

$$= \int_0^1 p \cdot u' \cdot s_h' \, dx = *$$

$u(x) = 2 - 2|x - \tfrac{1}{2}| - |x - \tfrac{1}{4}| - |x - \tfrac{3}{4}|$ $\qquad s_h(0) = s_h(1) = 0$

$$p(x) = \begin{cases} \tfrac{1}{8} & x \in (0, \tfrac{1}{4}] \vee x \in [\tfrac{3}{4}, 1) \\ \tfrac{1}{4} & x \in (\tfrac{1}{4}, \tfrac{3}{4}) \end{cases}$$

$\Rightarrow x \in (0, \tfrac{1}{4}] \Rightarrow u(x) = 4x \Rightarrow u'(x) = 4 \mid p(x) = \tfrac{1}{8}$

$x \in (\tfrac{1}{4}, \tfrac{1}{2}] \Rightarrow u(x) = \tfrac{1}{2} + 2x \Rightarrow u'(x) = 2 \mid p(x) = \tfrac{1}{4}$

$x \in (\tfrac{1}{2}, \tfrac{3}{4}] \Rightarrow u(x) = \tfrac{5}{2} - 2x \Rightarrow u'(x) = -2 \mid p(x) = \tfrac{1}{4}$

$x \in (\tfrac{3}{4}, 1] \Rightarrow u(x) = 4 - 4x \Rightarrow u'(x) = -4 \mid p(x) = \tfrac{1}{8}$ in Integral einsetzen

$$* = \int_0^{\frac{1}{4}} \tfrac{1}{2} s_h' \, dx + \int_{\frac{1}{4}}^{\frac{1}{2}} \tfrac{1}{2} s_h'(x) \, dx + \int_{\frac{1}{2}}^{\frac{3}{4}} -\tfrac{1}{2} s_h' \, dx + \int_{\frac{3}{4}}^{1} -\tfrac{1}{2} s_h' \, dx = \tfrac{1}{2} \cdot (s_h(\tfrac{1}{4}) - \underbrace{s_h(0)}_{0}) + \tfrac{1}{2} \cdot (s_h(\tfrac{1}{2}) - s_h(\tfrac{1}{4})) -$$

$$- \tfrac{1}{2} \cdot (s_h(\tfrac{3}{4}) - s_h(\tfrac{1}{2})) - \tfrac{1}{2} \cdot (\underbrace{s_h(1)}_{0} - s_h(\tfrac{3}{4})) = \tfrac{1}{2}(\cancel{s_h(\tfrac{1}{4})} + s_h(\tfrac{1}{2}) - \cancel{s_h(\tfrac{1}{4})} - \cancel{s_h(\tfrac{3}{4})} + s_h(\tfrac{1}{2}) + \cancel{s_h(\tfrac{3}{4})}) =$$

$$= \tfrac{1}{2} \cdot (2 \cdot s_h(\tfrac{1}{2})) = s_h(\tfrac{1}{2}) \checkmark$$

$$\lim_{\varepsilon \to 0} F_\varepsilon(s_h) = \lim_{\varepsilon \to 0} \int_0^1 f_\varepsilon \cdot s_h \, dx = \left| f_\varepsilon(x) = \varepsilon \cdot u_\varepsilon(x) + \begin{cases} 0 & x \notin (\tfrac{1}{2} - \varepsilon, \tfrac{1}{2} + \varepsilon) \\ \tfrac{1}{2\varepsilon} & \in (\tfrac{1}{2} - \varepsilon, \tfrac{1}{2} + \varepsilon) \end{cases} \right| =$$

$$= \lim_{\varepsilon \to 0} \Big(\underbrace{\int_0^{\frac{1}{2}-\varepsilon} \varepsilon \cdot u_\varepsilon(x) s_h \, dx}_{0} + \int_{\frac{1}{2}-\varepsilon}^{\frac{1}{2}+\varepsilon} s_h \cdot (\varepsilon \cdot u_\varepsilon(x) + \tfrac{1}{2\varepsilon}) \, dx + \underbrace{\int_{\frac{1}{2}+\varepsilon}^{1} \varepsilon \cdot u_\varepsilon(x) s_h \, dx}_{0} \Big) = \lim_{\varepsilon \to 0} \Big(\underbrace{\int_{\frac{1}{2}-\varepsilon}^{\frac{1}{2}+\varepsilon} \varepsilon \cdot u_\varepsilon(x) \cdot s_h \, dx}_{0} + \int_{\frac{1}{2}-\varepsilon}^{\frac{1}{2}+\varepsilon} \tfrac{1}{2\varepsilon} \cdot s_h \, dx \Big) =$$

$$= \lim_{\varepsilon \to 0} \frac{\int_{\frac{1}{2}-\varepsilon}^{\frac{1}{2}+\varepsilon} s_h(x) \, dx}{2\varepsilon} \overset{\text{l'Hospital}}{=} \lim_{\varepsilon \to 0} \frac{s_h(\tfrac{1}{2} + \varepsilon) - (s_h(\tfrac{1}{2} - \varepsilon) \cdot (-1))}{2} = \lim_{\varepsilon \to 0} \frac{s_h(\tfrac{1}{2} + \varepsilon) + s_h(\tfrac{1}{2} - \varepsilon)}{2} = \frac{s_h(\tfrac{1}{2}) + s_h(\tfrac{1}{2})}{2} = s_h(\tfrac{1}{2})$$

$\square$

b.)

4) b) Damit b) aus a) folgt ist noch zu zeigen: $v(x) \in S_0^1(\Omega)$

$S_0^1(x)$ ... Raum der stetigen, stückweise linearen Funktionen

$v(x) = 2 - 2|x - \frac{1}{2}| - |x - \frac{1}{4}| - |x - \frac{3}{4}|$

beginne mit $N=4$

4 Teilintervalle $\Rightarrow$ 3 innere Schnittstellen bei $\frac{1}{4}$, $\frac{1}{2}$ und $\frac{3}{4}$

betrachte $v\big|_{(0,\frac{1}{4})}$:

$$v_1 = v(x)\big|_{(0,\frac{1}{4})} = 2 - 2\underbrace{|x - \tfrac{1}{2}|}_{<0} - \underbrace{|x - \tfrac{1}{4}|}_{\leq 0} - \underbrace{|x - \tfrac{3}{4}|}_{<0} = 2 - 2(\tfrac{1}{2} - x) - (\tfrac{1}{4} - x) - (\tfrac{3}{4} - x) = 4x \text{ ... linear}$$

$$v_2 = v(x)\big|_{(\frac{1}{4},\frac{1}{2})} = 2 - 2\underbrace{|x - \tfrac{1}{2}|}_{<0} - \underbrace{|x - \tfrac{1}{4}|}_{\geq 0} - \underbrace{|x - \tfrac{3}{4}|}_{<0} = 2 - 2(\tfrac{1}{2} - x) - x + \tfrac{1}{4} - (\tfrac{3}{4} - x) = \tfrac{1}{2} + 2x \text{ ... linear affin}$$

$$v_3 = v(x)\big|_{(\frac{1}{2},\frac{3}{4})} = 2 - 2\underbrace{|x - \tfrac{1}{2}|}_{\geq 0} - \underbrace{|x - \tfrac{1}{4}|}_{>0} - \underbrace{|x - \tfrac{3}{4}|}_{\leq 0} = 2 - 2(x - \tfrac{1}{2}) - x + \tfrac{1}{4} - (\tfrac{3}{4} - x) = \tfrac{5}{2} - 2x \text{ ... linear affin}$$

$$v_4 = v(x)\big|_{(\frac{3}{4},1)} = 2 - 2\underbrace{|x - \tfrac{1}{2}|}_{>0} - \underbrace{|x - \tfrac{1}{4}|}_{>0} - \underbrace{|x - \tfrac{3}{4}|}_{\geq 0} = 2 - 2(x - \tfrac{1}{2}) - x + \tfrac{1}{4} - x + \tfrac{3}{4} = 4 - 4x \text{ ... linear affin}$$

Überprüfe auf Sprungstellen: $v_1(\frac{1}{4}) = 1 = v_2(\frac{1}{4})$ ✓

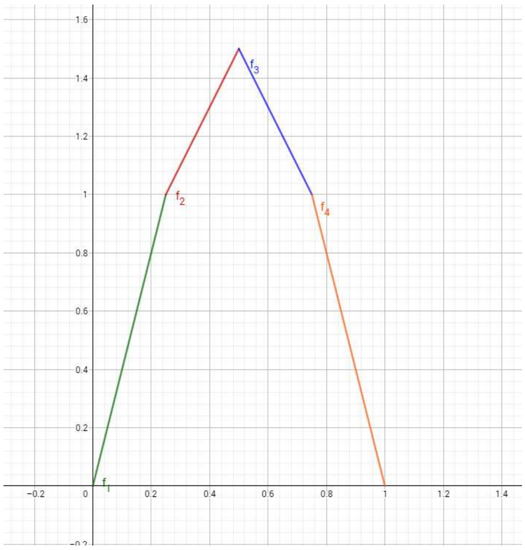
$v_2(\frac{1}{2}) = \frac{3}{2} = v_3(\frac{1}{2})$ ✓

$v_3(\frac{3}{4}) = 1 = v_4(\frac{3}{4})$ ✓

$\Rightarrow v$ ist stückchenweise konstant und hat keine Sprungstellen $\Rightarrow v$ ist stetig

$\forall N \in 2^{\mathbb{N}}$ bleibt $v \in S_0^1(\Omega)$, denn wenn die einzelnen Teilfunktionen wieder bei der Hälfte unterteilt werden, bleiben sie linear

Weil a) korrekt ist und $v \in S_0^1(\Omega) \Rightarrow$ b) ist korrekt



c.) d.)

```
Finite Differenzen:
Ord in 2,h-Norm = 1.0000726771
Ord in A-Norm   = 0.4999984761
Finite Elemente:
Ord in 2,h-Norm = 1.9971845011
Ord in A-Norm   = 1.4987646307
```

Code:

```
%***************************************************************************
%*                         Boundary value problems                        *
%*                                                                        *
%*  Solve boundary value problems with Dirichlet boundary conditions      *
%*  with the the finite differences method and the finite elements method.*
%*                                                                        *
%*  Author: Christoph Raunjak                                             *
%           christoph.raunjak@edu.uni-graz.at                             *
%*  20.1.2021 vs1.0                                                       *
%***************************************************************************

clear;clc;clf;
h1 = figure(1); close(h1); h1 = figure(1);
set(h1,'Position',[40 40 1200 600]);

%define functions
epsilon = 1/(2^5);
u = @(x) (2-2*abs(x-1/2)-abs(x-1/4)-abs(x-3/4));
ce = @(x) ((3/2-epsilon)-((x-1/2).^2)/epsilon);
ue = @(x) ((abs(x-1/2)>= epsilon).*u(x) + (abs(x-1/2)<epsilon).* ce(x));
p = @(x) ((1/8).*(abs(x-1/2)>= 1/4) + (1/4).*(abs(x-1/2) < 1/4));
q = @(x) ones(size(x))*epsilon;
f = @(x) (epsilon.*ue(x) + (1/(2*epsilon))*(abs(x-1/2)<epsilon));

disp('Finite Differenzen:')
erI = [];
erA = [];
N   = 32;
l   = 5;
for i=1:5
    N   = 2*N;                          % N = 64, 128, 256, 512, 1024
    h   = 1/N;
    x   = linspace(0,1,N+1)';           % grid points
    xz  = x(1:N) + 0.5*h;               % cell centers
    xi  = x(2:N);                       % inner grid points

    Q   = spdiags([-ones(N-1,1),ones(N-1,1)],[0,1],N-1,N)/h;
    Dp  = spdiags(p(xz),0,N,N);         % diag(p) p in cell centers
    Dq  = spdiags(q(xi),0,N-1,N-1);     % diag(q) q in inner grid points
    F   = f(xi);                        % f in cell centers

    A   = Q*Dp*Q.' + Dq;                % equation (69)
    ua  = A\F;                          % approximation in inner grid points

    du  = ue(xi)-ua;                    % error in cell centers
    Adu = A*du;

   erI = [erI,sqrt(h)*norm(du)];       % 2,h-norm
   erA = [erA,sqrt(du'*Adu)];          % A-norm
end
deI = erI(1:(l-1))./erI(2:l);
deA = erA(1:(l-1))./erA(2:l);
ordI = log2(mean(deI));
ordA = log2(mean(deA));
disp(sprintf('Ord in 2,h-Norm = %0.10f',ordI));
disp(sprintf('Ord in A-Norm   = %0.10f',ordA));
```

```
disp('Finite Elemente:')
erI = [];
erA = [];
N   = 32;
l   = 5;
for i=1:l
    N   = 2*N;                          % N = 64, 128, 256, 512, 1024
    h   = 1/N;
    x   = linspace(0,1,N+1)';           % grid points
    xz  = x(1:N) + 0.5*h;               % cell centers
    xi  = x(2:N);                       % inner grid points

    Q   = spdiags([-ones(N,1),ones(N,1)],[0,1],N-1,N)/h;
    R   = spdiags(ones(N,2),[0,1],N-1,N);

    Dq  = spdiags(q(xz),0,N,N);         % diag(q) p in cell centers
    Dp  = spdiags(p(xz),0,N,N);         % diag(p) q in cell centers
    T   = spdiags(R*q(xz),0,N-1,N-1);
    F   = R*f(xz)/2;                    % f in cell centers

    A   = Q*Dp*Q.' + (R*Dq*R.' + T)/6;  % equations (71)
    ut  = A\F;                          % approximation in inner grid points

    du  = ue(xi)-ut;                    % error in inner grid points
    Adu = A*du;

    erI = [erI,sqrt(h)*norm(du)];       % 2,h-Norm
    erA = [erA,sqrt(du'*Adu)];          % A-Norm
end

deI = erI(1:(l-1))./erI(2:l);
deA = erA(1:(l-1))./erA(2:l);
ordI = log2(mean(deI));
ordA = log2(mean(deA));
disp(sprintf('Ord in 2,h-Norm = %0.10f',ordI));
disp(sprintf('Ord in A-Norm   = %0.10f',ordA));
```

```
%bsp 6 Blatt 12

u  = @(x) x.*x.*(1-x).^2;
p  = @(x) 1-(x-1/2).^2;
q  = @(x) 1+(x-1/2).^2;
f  = @(x) (-40+x.*(576 + x.*(-571+x.*(-14 + x.*(17+x.*(-12 +4*x))))))/4;


disp('Finite Differenzen:')
erI = [];
erA = [];
N   = 32;
l   = 5;
for i=1:l
    N   = 2*N;                          % N = 64, 128, 256, 512, 1024
    h   = 1/N;
    x   = linspace(0,1,N+1)';           % Schnittstellen
    xz  = x(1:N) + 0.5*h;               % Zellzentren
    xi  = x(2:N);                       % innere Schnittstellen
                                        % d/dx
    Q   = spdiags([-ones(N-1,1),ones(N-1,1)],[0,1],N-1,N)/h;
    Dp  = spdiags(p(xi),0,N-1,N-1);         % diag(p) in Zellzentren
    Dq  = spdiags(q(xi),0,N-1,N-1);     % diag(q) in inneren Schnittstellen
                                       % f in inneren Schnittstellen
    T=zeros(N-1,N-1);
    T(1,1)=5/h^4;
    T(1,2)=-8/h^4;
    T(1,3)=3/h^4;
    T(N-1,N-1)=5/h^4;
    T(N-1,N-2)=-8/h^4;
    T(N-1,N-3)=3/h^4;

    M   = spdiags(ones(N+1,2)/2,[-1,0],N-1,N-2);
    L   = spdiags([-ones(N-1,1),-ones(N-1,1), 2*ones(N-1,1)],[1,-1,0],N-1,N-1)/h^2;
    S   = spdiags([ones(N-1,1)],[0],N-1,N-1);
    S(1,1)=0;
    S(N-1,N-1)=0;
    %R   = spdiags([ones(N-2,1),4*ones(N-2,1),ones(N-2,1)],[0,1,2],N-2,N);
    F   = S*f(xi);
    A   = S*L'*Dp*L+S*Dq+T;
    A0   = L'*Dp*L + Dq;                 % Gleichung (83)
    ua  = A\F;                          % Approximation in inneren
                                        %   Schnittstellen
    du  =u(xi) - ua;                     % Fehler in inneren Schnittstellen
    Adu = A0*du;

    subplot(2,4,1); hold on;            % grafische Darstellung der Loesung
    plot(xi,u(xi),'b',xi,ua)
    subplot(2,4,2); hold on;
    plot(xi,abs(du))
    subplot(2,4,3); hold on;
    plot(xi,abs(Adu))

    erI = [erI,sqrt(h)*norm(du)];       % 2,h-Norm
    erA = [erA,sqrt(du'*Adu)];          % A-Norm
end
subplot(2,4,1)                          % grafische Darstellung der Fehler
title('Finite Differenzen')
subplot(2,4,2)
title('Fehler')
subplot(2,4,3)
title('A0*Fehler')
subplot(2,4,4)
deI = erI(1:(l-1))./erI(2:l);
deA = erA(1:(l-1))./erA(2:l);
plot(1:l,log2(erI),1:l,log2(erA))
legend('erI','erA')
title('Ordnung Abschätzung')
ordI = log2(mean(deI));                 % Abschaetzung der Ordnung
ordA = log2(mean(deA));
disp(sprintf('Ord in 2,h-Norm = %0.10f',ordI));
disp(sprintf('Ord in A0-Norm   = %0.10f',ordA));
```

```
disp('Finite Elemente:')
erI = [];
erA = [];
N   = 32;
l   = 5;
for i=1:l
    N   = 2*N;                              % N = 64, 128, 256, 512, 1024
    h   = 1/N;
    x   = linspace(0,1,N+1)';               % Schnittstellen
    xz  = x(1:N) + 0.5*h;                   % Zellzentren
    xi  = x(2:N);                           % innere Schnittstellen
                                            % d/dx
   L   = spdiags([-ones(N-2,1),-ones(N-2,1), 2*ones(N-2,1)],[0,2,1],N-2,N)/h^2;
   T  = spdiags([ones(N-2,1),-23*ones(N-2,1),ones(N-2,1) ],[0,1,2],N-2,N);
   S  = spdiags([ones(N-2,1),13*ones(N-2,1),ones(N-2,1)],[0,1,2],N-2,N);
    R   = spdiags([ones(N-2,1),4*ones(N-2,1),ones(N-2,1)],[0,1,2],N-2,N);
    M = spdiags(ones(N+1,2)/2,[-1,0],N-1,N-2);
    Dq  = spdiags(q(xz),0,N,N);             % diag(q) in Zellzentren
    Dp  = spdiags(p(xz),0,N,N);             % diag(p) in Zellzentren
    Tq  = spdiags(T*q(xz),0,N-2,N-2);
    F   = R*f(xz)/6;                        % f in Zellzentren

    A   = L*Dp*L' + S*Dq*S'/120+ Tq/24;       % Gleichung (84)
    ua  = A\F;                              % Approximation in inneren
                                            %   Schnittstellen
    du  = u(xi)- M* ua;                       % Fehler in inneren Schnittstellen
    L0   = spdiags([-ones(N-1,1),-ones(N-1,1), 2*ones(N-1,1)],[1,-1,0],N-1,N-1)/h^2;
    Dp0  = spdiags(p(xi),0,N-1,N-1);            % diag(p) in Zellzentren
    Dq0  = spdiags(q(xi),0,N-1,N-1);
    A0= L0'*Dp0*L0 + Dq0;
    Adu = A0*du;

    subplot(2,4,5); hold on;                % grafische Darstellung der Loesung
    plot(xi,u(xi),'b',xi,M*ua)
    subplot(2,4,6); hold on;
    plot(xi,abs(du))
    subplot(2,4,7); hold on;
    plot(xi,abs(Adu))

    erI = [erI,sqrt(h)*norm(du)];           % 2,h-Norm
    erA = [erA,sqrt(du'*Adu)];              % A-Norm
end
subplot(2,4,5)                              % grafische Darstellung der Fehler
title('Finite Elemente')
subplot(2,4,6)
title('Fehler')
subplot(2,4,7)
title('A0*Fehler')
subplot(2,4,8)
deI = erI(1:(l-1))./erI(2:l);
deA = erA(1:(l-1))./erA(2:l);
plot(1:l,log2(erI),1:l,log2(erA))
legend('erI','erA')
title('Ordnung Abschätzung')
ordI = log2(mean(deI));                     % Abschaetzung der Ordnung
ordA = log2(mean(deA));
disp(sprintf('Ord in 2,h-Norm = %0.10f',ordI));
disp(sprintf('Ord in A0-Norm   = %0.10f',ordA));

for i=1:8
    subplot(2,4,i)
    hold off
end
```

```
u = @(x,y) kron(x.*(x-1).*(2*x-1), y .*(y-1).*(2*y-1));
p = @(x,y) kron((1-(x-1/2).^2),(1-(y-1/2).^2));
q = @(x,y) kron((1+(x-1/2).^2),(1+(y-1/2).^2));
ux = @(x,y) kron((1-6*x+6*x.^2),(y-3*y.^2+2*y.^3));
uxx = @(x,y) kron((-1+2*x),(y.*(1-3*y+2*y.^2)))*6;
uy = @(x,y) kron((x-3*x.^2+2*x.^3),(1-6*y+6*y.^2));
uyy = @(x,y) kron(x.*(1-3*x+2*x.^2),(-1+2*y))*6;
px = @(x,y) kron((-1+2*x),(-3-4*y+4*y.^2))/4;
py = @(x,y) kron((-3-4*x+4*x.^2), (-1+2*y))/4;
f = @(x,y) -(px(x,y) .* ux(x,y) + p(x,y) .* uxx(x,y)) -(py(x,y) .*
uy(x,y) +p(x,y) .* uyy(x,y)) + q(x,y) .* u(x,y);
%Wenn man bereits hier das Kronecker tensor product verwendet, spart
man
%sich viele Zeilen Code

disp('Finite Differenzen:')
N = 8;
l = 5;
ErI = [];
ErA = [];
for i = 1:l

    h = 1/N;
    x = linspace(0,1,N+1)'; %x-Schnittstellen; transponiert,
                            %damit Kronecker-Produkt in
                            %Funktionsvorschrift
                            %funktioniert
    y = linspace(0,1,N+1); % y-Schnittstellen
    xi = x(2:N); %innere x-Schnittstellen
    xz = x(1:N) + h/2; %x-Zellzentren
    yi = y(2:N); %innere y-Schnittstellen
    yz = y(1:N) + h/2; %y-Zellzentren

    %Sammlung aller Vektoren/Matrizen, die wir fuer Gleichung (101)
    %brauchen
    Q = spdiags([-ones(N-1,1), ones(N-1,1)], [0,1], N-1,N)/h;
    Q1 = kron(speye(N-1),Q);
    Q2 = kron(Q, speye(N-1));

    pu = reshape(p(xi,yz), (N-1)*N,1); %Speicherung in x-Schnittstellen
    pd = reshape(p(xz,yi), (N-1)*N,1); %Speicherung in y-Schnittstellen
    Pu = spdiags(pu, 0, (N-1)*N,(N-1)*N);
    Pd = spdiags(pd, 0, (N-1)*N,(N-1)*N);

    qq = reshape(q(xi,yi),(N-1)*(N-1),1); %Speicherung in inneren
Zelleneckstellen
    Dq = spdiags(qq, 0, (N-1)*(N-1), (N-1)*(N-1));

    ff = reshape(f(xi,yi),(N-1)*(N-1),1);

    A = Q1*Pd*Q1' + Q2*Pu*Q2' + Dq; %Matrix A laut Anhang
    u2 = A\ff; %Gleichung (101)
```

```
    u1 = reshape(u(xi,yi),(N-1)*(N-1),1); %exakte Loesung in inneren
Zelleneckstellen

    du = u1 - u2; %Fehler der Approximation
    Adu = A*du;

    ErI = [ErI, h * norm(du)]; %Speicherung der 2, h^2-Norm
    ErA = [ErA, sqrt(du'*Adu)]; %Speicherung der A-Norm

    %Grafik
    subplot(1,3,2)
    surf(xi,yi,u(xi,yi))
    title('exakt')
    subplot(1,3,1)
    surf(xi,yi,reshape(u2,N-1,N-1))
    title('Finite Differenzen')
    pause(0.5);


    N = 2*N; % N = 8, 16, 32, 64, 128

end

%Berechnung der F_i laut Angabe
deI = ErI(1:(l-1))./ErI(2:l);
deA = ErA(1:(l-1))./ErA(2:l);

%Antwortmoeglichkeiten
a = log2(mean(deI))
b = log2(mean(deA))


%-----------------------------------------------------------------------
disp('Finite Elemente:')

N = 8;
l = 5;
ErI = [];
ErA = [];
for i = 1:l %wie oben
    h = 1/N;
    x = linspace(0,1,N+1)';
    y = linspace(0,1,N+1);
    xi = x(2:N);
    xz = x(1:N) + h/2;
    yi = y(2:N);
    yz = y(1:N) + h/2;

    %Definition aller Matrizen fuer Gleichung (102)
    Q = spdiags([-ones(N-1,1), ones(N-1,1)], [0,1], N-1,N)/h;
    Q1 = kron(speye(N-1),Q);
    Q2 = kron(Q, speye(N-1));
```

```
    R = spdiags([ones(N-1,1), ones(N-1,1)], [0,1], N-1,N);
    R1 = kron(speye(N-1),R);
    R2 = kron(R, speye(N-1));

    S1 = kron(speye(N),R);
    S2 = kron(R, speye(N));
    T = kron(R,R);

    %Speicherung in Zellzentren
    pp = reshape(p(xz,yz), N*N,1);
    ff = reshape(f(xz,yz), N*N,1);
    qq = reshape(q(xz,yz), N*N,1);
    Dp = spdiags(pp, 0, N*N, N*N);
    Dq = spdiags(qq, 0, N*N, N*N);

    S1p = S1*pp;
    S2p = S2*pp;
    S1q = S1*qq;
    S2q = S2*qq;
    Tq = T * qq;
    DS1p = spdiags(S1p, 0, (N-1)*N, (N-1)*N);
    DS2p = spdiags(S2p, 0, (N-1)*N, (N-1)*N);
    DS1q = spdiags(S1q, 0, (N-1)*N, (N-1)*N);
    DS2q = spdiags(S2q, 0, (N-1)*N, (N-1)*N);
    DTq = spdiags(Tq, 0, (N-1)*(N-1), (N-1)*(N-1));

    %Konstruktion von A laut Anhang
    A = Q1 * S2 * Dp * S2' * Q1';
    A = A + Q2 * S1 * Dp * S1' * Q2';
    A = A + Q1 * DS2p * Q1' + Q2 * DS1p * Q2';
    A = A + (R1 * DS2q * R1' + R2 * DS1q * R2')/6;
    A = A + (T * Dq * T' + DTq)/6;
    A = A/6;

    b = T*ff/4;
    u2 = A\b; %Gleichung (102)

    u1 = reshape(u(xi,yi),(N-1)*(N-1),1); %exakte Loesung in inneren
Zelleneckstellen

    du = u1 - u2; %Fehler der Approximation
    Adu = A*du;
    ErI = [ErI, h * norm(du)]; %Speicherung der 2, h^2-Norm
    ErA = [ErA, sqrt(du'*Adu)]; %Speicherung der A_Norm

    %Grafik
    subplot(1,3,3)
    surf(xi,yi,reshape(u2,N-1,N-1))
    title('Finite Elemente')
    subplot(1,3,2)
    surf(xi,yi,u(xi,yi))
    title('Exakt')
    axis([0 1 0 1 -0.01 +0.01])
```

```
    pause(0.5);

    N = 2 * N; %wie oben

end

%Berechnung der F_i laut Angabe
deI = ErI(1:(l-1))./ErI(2:l);
deA = ErA(1:(l-1))./ErA(2:l);

%Antwortmoeglichkeiten
c = log2(mean(deI))
d = log2(mean(deA))
```