MATH 666: Finite Element Methods
Homework 3

Caleb Logemann

December 4, 2018

#1 Consider the Crank-Nicolson-cG(1) FEM for the diffusion equation. Find $u_h^n \in V_h^1$ such that

$$\left\langle \frac{u_h^n - u_h^{n-1}}{\delta t}, v \right\rangle + \left\langle \nabla\left(\frac{u_h^n + u_h^{n-1}}{2}\right), v \right\rangle = \left\langle \frac{f^n + f^{n-1}}{2}, \nabla v \right\rangle \quad \forall v \in V_h^1, \quad \forall n = 1,2,3,\ldots$$
$$\left\langle u_h^0, v \right\rangle = \langle u_0, v\rangle.$$

Following the convergence proof of the Backward-Euler-cG(1) FEM from class, prove that the Crank-Nicolson-cG(1) FEM satisfies:

$$\|u_h^n - u(t^n)\| = \mathcal{O}\left(\Delta t^2 + h^2\right)$$

Proof. The initial variational problem is formulated as finding $u \in V$ such that

$$\langle u_t, v\rangle + \langle \nabla u, \nabla v\rangle = \langle f, v\rangle \quad \forall v \in V$$
$$\langle u(0,x), v\rangle = \langle u_0, v\rangle$$

The semidescrete form of the FEM can be expressed as finding $u_h \in V_h^1$ such that

$$\langle (u_h)_t, v\rangle + \langle \nabla u_h, \nabla v\rangle = \langle f, v\rangle \quad \forall v \in V_h^1 \langle u_h(0,x), v\rangle = \langle u_0, v\rangle$$

I will also make use of the following projection $R_h(u)$ which satisfies

$$\langle \nabla R_h(u), \nabla v\rangle = \langle \nabla u, \nabla v\rangle \quad \forall v \in V_h^1$$

Suppose u is the solution to the original variational problem, then for all $v \in V_h^1$,

$$\langle u_t, v\rangle + \langle \nabla u, \nabla v\rangle = \langle f, v\rangle$$
$$\langle u_t, v\rangle + \langle \nabla R_h(u), \nabla v\rangle = \langle f, v\rangle$$

This is true for every time $t > 0$, in particular it is true at times t^n and t^{n-1}, so

$$\langle u_t^n, v\rangle + \langle \nabla R_h(u^n), \nabla v\rangle = \langle f^n, v\rangle$$
$$\left\langle u_t^{n-1}, v\right\rangle + \left\langle \nabla R_h(u^{n-1}), \nabla v\right\rangle = \left\langle f^{n-1}, v\right\rangle$$
$$\left\langle \frac{u_t^n + u_t^{n-1}}{2}, v\right\rangle + \left\langle \nabla R_h\left(\frac{u^n + u^{n-1}}{2}\right), \nabla v\right\rangle = \left\langle \frac{f^n + f^{n-1}}{2}, v\right\rangle$$
$$\left\langle \nabla R_h\left(\frac{u^n + u^{n-1}}{2}\right), \nabla v\right\rangle = \left\langle \frac{f^n + f^{n-1}}{2}, v\right\rangle - \left\langle \frac{u_t^n + u_t^{n-1}}{2}, v\right\rangle$$

Consider u_h^n and u_h^{n-1}, solutions to the FEM, they satisfy

$$\left\langle \frac{u_h^n - u_h^n}{\delta T}, v\right\rangle + \left\langle \nabla \frac{u_h^n + u_h^{n-1}}{2}, \nabla v\right\rangle = \left\langle \frac{f^n + f^{n-1}}{2}, v\right\rangle$$

Let $\theta^n = R_h u^n - u_h^n$, then subtracting the last two equations gives

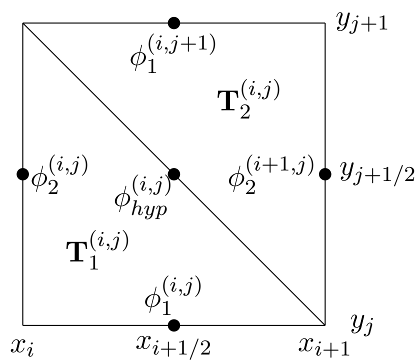
$$test$$

□

#2 Consider the non-conforming FEM for the Poisson equation using the linear Crouzeix-Raviart element discussed in class:

(a) Explicitly write out the stiffness matrix on a uniform Cartisian mesh (with triangles created by cutting each Cartesian square into two triangles) on $[0,1] \times [0,1]$.

First I will define some notation. For the Cartesian mesh, let M be the number of interior nodes along one axis, then $h = \frac{1}{M+1}$. Now the coordinates of the nodes of the Cartesian mesh will be $x_i = ih$ and $y_j = jh$, for $i, j = 0, \ldots, M+1$. We can also denote the midpoints as $x_{i+1/2} = ih + \frac{h}{2}$ and $y_{j+1/2} = jh + \frac{h}{2}$. For the Crouzeix-Raviart element, the midpoints of the edges of the triangles are the important points. These points will be at $(x_{i+1/2}, y_{j+1/2})$ for $i, j = 0, \ldots, M$, $(x_i, y_{j+1/2})$ for $i = 1, \ldots, M$ and $j = 0, \ldots, M$, and at $(x_{i+1/2}, y_j)$ for $j = 1, \ldots, M$ and $i = 0, \ldots, M$. Below is the image of how one cell in the Cartesian mesh is transformed into two Crouzeix-Raviart elements. The image also describes some notation that I will use. The lower left triangle/region, I will label $\mathbf{T}_1^{(i,j)}$ and the upper right triangle will be $\mathbf{T}_2^{(i,j)}$. The basis functions for each node are labeled as described.



Now each node of the Crouziex-Raviart element needs a basis function associated with that node. The basis function should be piecewise linear, such that it is one at its associated node and zero at all other nodes. There are three possible types, one for each side of the triangle. Suppose the node is on the hypotenuse of the right triangle, then the node is at $(x_{i+1/2}, y_{j+1/2})$ for some $0 \leq i, j \leq M$. In this case the basis function will be

$$\phi_{hyp}^{(i,j)}(x,y) = \begin{cases} 1 + \frac{2}{h}\left(x - x_{i+1/2}\right) + \frac{2}{h}\left(y - y_{j+1/2}\right) & \mathbf{T}_1^{(i,j)} \\ 1 - \frac{2}{h}\left(x - x_{i+1/2}\right) - \frac{2}{h}\left(y - y_{j+1/2}\right) & \mathbf{T}_2^{(i,j)} \\ 0 & \text{otherwise} \end{cases}$$

Suppose the node is on a leg of the right triangle such that it is at the point $(x_{i+1/2}, y_j)$ for some $0 \leq i \leq M$ and some $1 \leq j \leq M$, then the basis function will be

$$\phi_1^{(i,j)}(x,y) = \begin{cases} 1 - \frac{2}{h}(y - y_j) & \mathbf{T}_1^{(i,j)} \\ 1 + \frac{2}{h}(y - y_j) & \mathbf{T}_2^{(i,j-1)} \\ 0 & \text{otherwise} \end{cases}$$

The last case is when the node is at $(x_i, y_{j+1/2})$ for some $1 \leq i \leq M$ and some $0 \leq j \leq M$, then the basis function is

$$\phi_2^{(i,j)}(x,y) = \begin{cases} 1 - \frac{2}{h}(x - x_i) & \mathbf{T}_1^{(i,j)} \\ 1 + \frac{2}{h}(x - x_i) & \mathbf{T}_2^{(i-1,j)} \\ 0 & \text{otherwise} \end{cases}$$

The gradients of these piecewise linear functions can be computed easily.

$$\nabla\phi_{hyp}^{(i,j)}(x,y) = \begin{cases} \left[\frac{2}{h}, \frac{2}{h}\right]^T & \mathbf{T}_1^{(i,j)} \\ \left[-\frac{2}{h}, -\frac{2}{h}\right]^T & \mathbf{T}_2^{(i,j)} \\ [0,0]^T & \text{otherwise} \end{cases}$$

$$\nabla\phi_{1}^{(i,j)}(x,y) = \begin{cases} \left[0, -\frac{2}{h}\right]^T & \mathbf{T}_1^{(i,j)} \\ \left[0, \frac{2}{h}\right]^T & \mathbf{T}_2^{(i,j-1)} \\ [0,0]^T & \text{otherwise} \end{cases}$$

$$\nabla\phi_{2}^{(i,j)}(x,y) = \begin{cases} \left[-\frac{2}{h}, 0\right]^T & \mathbf{T}_1^{(i,j)} \\ \left[\frac{2}{h}, 0\right]^T & \mathbf{T}_2^{(i-1,j)} \\ [0,0]^T & \text{otherwise} \end{cases}$$

In order to compute the stiffness matrix, the integrals of the dot products of the gradients must be computed. Symmetry can be used to reduce this to 4 integrals.

$$\begin{aligned}
\int_0^1\int_0^1 \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_{hyp}^{(i,j)}\,\mathrm{d}x\,\mathrm{d}y &= \int_{\mathbf{T}_1^{(i,j)}} \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_{hyp}^{(i,j)}\,\mathrm{d}a + \int_{\mathbf{T}_2^{(i,j)}} \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_{hyp}^{(i,j)}\,\mathrm{d}a \\
&= \int_{\mathbf{T}_1^{(i,j)}} \left[-\frac{2}{h},-\frac{2}{h}\right]\cdot\left[-\frac{2}{h},-\frac{2}{h}\right]\mathrm{d}a + \int_{\mathbf{T}_2^{(i,j)}} \left[\frac{2}{h},\frac{2}{h}\right]\cdot\left[\frac{2}{h},\frac{2}{h}\right]\mathrm{d}a \\
&= \frac{8}{h^2}\int_{\mathbf{T}_1^{(i,j)}}\mathrm{d}a + \frac{8}{h^2}\int_{\mathbf{T}_2^{(i,j)}}\mathrm{d}a \\
&= \frac{8}{h^2}\frac{h^2}{2} + \frac{8}{h^2}\frac{h^2}{2} \\
&= 8 \\
\int_0^1\int_0^1 \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_1^{(i,j)}\,\mathrm{d}x\,\mathrm{d}y &= \int_0^1\int_0^1 \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_2^{(i,j)}\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_0^1\int_0^1 \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_1^{(i,j+1)}\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_0^1\int_0^1 \nabla\phi_{hyp}^{(i,j)}\cdot\nabla\phi_2^{(i+1,j)}\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_{\mathbf{T}_2^{(i,j)}} \left[-\frac{2}{h},-\frac{2}{h}\right]\cdot\left[\frac{2}{h},0\right]\mathrm{d}a \\
&= -\frac{4}{h^2}\int_{\mathbf{T}_2^{(i,j)}}\mathrm{d}a \\
&= -\frac{4}{h^2}\frac{h^2}{2} \\
&= -2 \\
\int_0^1\int_0^1 \nabla\phi_1^{(i,j)}\cdot\nabla\phi_1^{(i,j)}\,\mathrm{d}x\,\mathrm{d}y &= \int_0^1\int_0^1 \nabla\phi_2^{(i,j)}\cdot\nabla\phi_2^{(i,j)}\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_{\mathbf{T}_1^{(i,j)}} \nabla\phi_2^{(i,j)}\cdot\nabla\phi_2^{(i,j)}\,\mathrm{d}a + \int_{\mathbf{T}_2^{(i-1,j)}} \nabla\phi_2^{(i,j)}\cdot\nabla\phi_2^{(i,j)}\,\mathrm{d}a \\
&= \int_{\mathbf{T}_1^{(i,j)}} \left[-\frac{2}{h},0\right]\cdot\left[-\frac{2}{h},0\right]\mathrm{d}a + \int_{\mathbf{T}_2^{(i-1,j)}} \left[\frac{2}{h},0\right]\cdot\left[\frac{2}{h},0\right]\mathrm{d}a \\
&= \frac{4}{h^2}\int_{\mathbf{T}_1^{(i,j)}}\mathrm{d}a + \frac{4}{h^2}\int_{\mathbf{T}_2^{(i-1,j)}}\mathrm{d}a \\
&= \frac{4}{h^2}\frac{h^2}{2} + \frac{4}{h^2}\frac{h^2}{2} \\
&= 4 \\
\int_0^1\int_0^1 \nabla\phi_1^{(i,j)}\cdot\nabla\phi_2^{(i,j)}\,\mathrm{d}x\,\mathrm{d}y &= \int_{\mathbf{T}_1^{(i,j)}} \nabla\phi_1^{(i,j)}\cdot\nabla\phi_2^{(i,j)}\,\mathrm{d}a \\
&= \int_{\mathbf{T}_1^{(i,j)}} \left[-\frac{2}{h},0\right]\cdot\left[0,-\frac{2}{h}\right]\mathrm{d}a \\
&= \int_{\mathbf{T}_1^{(i,j)}} 0\,\mathrm{d}a \\
&= 0
\end{aligned}$$

Now that we have all of these values we can form the stiffness matrix. However the structure of the stiffness matrix will depend on how the nodes are numbered or listed for the matrix equation. I will number the points along the rows. If we start with M interior nodes or $M+1$ cells per row in the initial cartesian mesh, then there will be $2M-1$ rows of interior points for the Crouziex-Raviart elements. The odd numbered rows will have $2M-1$ nodes, and the even numbered rows will have M nodes. The odd numbered rows correspond to the ϕ_{hyp} and ϕ_2 basis functions while the even numbered rows only contain nodes corresponding to ϕ_1.

The stiffness matrix with this numbering will have the following block structure

$$\begin{bmatrix} A & C & & & & & \\ C^T & B & C^T & & & & \\ & C & A & C & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & C & A & C & \\ & & & & C^T & B & C^T \\ & & & & & C & A \end{bmatrix}$$

where A is $(2M-1)\times(2M-1)$ and is of the form

$$A = \begin{bmatrix} 8 & -2 & & & & & \\ -2 & 4 & -2 & & & & \\ & -2 & 8 & -2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -2 & 8 & -2 & \\ & & & & -2 & 4 & -2 \\ & & & & & -2 & 8 \end{bmatrix},$$

B is $M\times M$ and is the form

$$B = \begin{bmatrix} 4 & & \\ & \ddots & \\ & & 4 \end{bmatrix}$$

and C is $(2M-1)\times M$ and is of the form

$$C = \begin{bmatrix} -2 & & & & \\ & & & & \\ & -2 & & & \\ & & & & \\ & & \ddots & & \\ & & & & \\ & & & -2 & \\ & & & & \\ & & & & -2 \end{bmatrix}.$$

(b) Use this to solve the Poisson equation on $[0,1]\times[0,1]$ with $u=0$ on the boundary:

$$-u_{xx} - u_{yy} = 8\pi^2 \sin(2\pi x)\sin(2\pi y).$$

The following code generates the stiffness matrix and the right hand side and then solves the system.

```
function [u] = uniformSquareCrouziexRaviart(f, M, a, b)
    h = (b - a)/M;

    e = ones(2*M-1,1);
```

```
    alternating = [repmat([2; 1], M-1,1); 2];
    A = spdiags([-2*e, 4*alternating, -2*e], [-1,0,1], 2*M-1, 2*M-1);
    B = 4*speye(M,M);
    C = sparse(1:2:(2*M-1), 1:M, -2*ones(M,1), 2*M-1, M);

    % location of A matrices in D
    upperDiagonal = sparse(1:M-2,2:M-1,1,M-1,M-1);
    lowerDiagonal = sparse(2:M-1,1:M-2,1,M-1,M-1);

    D = kron(speye(M-1,M-1), [A, C; C', B]) ...
        + kron(upperDiagonal, [sparse(2*M-1,3*M-1); C', sparse(M,M)]) ...
        + kron(lowerDiagonal, [sparse(2*M-1,2*M-1), C; sparse(M,3*M-1)]);

    [nRows, nCols] = size(D);
    D = [D, [sparse(nRows-M, 2*M-1); C'] ;sparse(2*M-1,nCols-M), C, A];

    x1 = a+h/2:h/2:b-h/2;
    x2 = a+h/2:h:b-h/2;
    x = [repmat([x1,x2], 1,M-1), x1];
    y1 = ones(2*M-1,1)*x1(1:2:end);
    y2 = ones(M,1)*x1(2:2:end);
    y = [cell2mat(arrayfun(@(i) [y1(:,i)',y2(:,i)'],1:M-1,'UniformOutput',false)),
       ↪ y1(:,end)'];
    rhs = arrayfun(@(i) f(x(i), y(i))*h^2/3, 1:length(x))';

    u = D\rhs;
end
```

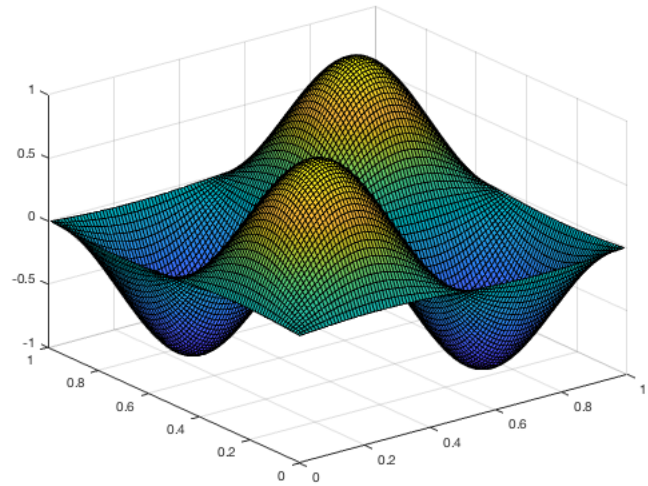
The following script uses this function to solve the given system and do the convergence study.

```
f = @(x, y) 8*pi^2*sin(2*pi*x)*sin(2*pi*y);
a = 0;
b = 1;

L2error = [];
Marray = [7, 13, 25, 101, 401];
for M = Marray
    u = uniformSquareCrouziexRaviart(f, M, a, b);
    h = (b-a)/M;
    [X, Y] = meshgrid(a+h/2:h:b-h/2);
    dr = 2*M-1 + M;
    ind = dr*floor(0:1/M:M-1/M) + repmat(1:2:2*M-1,1,M);
    uSol = reshape(u(ind), M, M);
    surf(X, Y, uSol);
    uExact = @(x, y) sin(2*pi*x).*sin(2*pi*y);
    uExactSol = uExact(X, Y);
    L2error = [L2error, norm(uExactSol- uSol)];
end

log(L2error(1:end-1)./L2error(2:end))./log(((b-a)./Marray(1:end-1))./((b-a)./Marray
    ↪ (2:end)))
```

(c) Show a plot of the solution on a reasonable resolved mesh. Note: Before plotting, you may want to evaluate the solution in element centers.



(d) Do a numerical convergence study in the L^2 norm. For simplicity, you can just compute the errors based on errors at element centers. The exact solution is

$$u(x, y) = \sin(2\pi x)\sin(2\pi y)$$

I found the order of convergence to be 3, as shown in the following table.

M	h	L^2 error	order
7	0.14	0.002	-
13	0.07	0.00037	3.04
25	0.04	0.000052	3.01
101	0.0099	0.00000078	3.002
401	0.0025	0.000000012	3.0004