Report for MAT 7990 Directed Study

Lewei Zhao, Wayne State University, USA

May 9, 2017

1 Code Development of Finite Element Mesh in 2D by MATLAB

1. Uniform Mesh

   (a) Partition of an arbitrary triangle through linking the midpoints of each side.

```
function [node,elem,bdFlag]=TriMesh(m)
node=rand(3,2);
elem=[1,2,3];
bdFlag = setboundary(node,elem,'Dirichlet');
for i=1:m
[node,elem,bdFlag] = uniformrefine(node,elem,bdFlag);
end
node=node;
elem=elem;
bdFlag=bdFlag;
showmesh(node,elem);
```

   where the program uniformrefine is given by MATLAB software package IFEM developed by Prof.Long Chen

   (b) Partition of an arbitrary quadrangle through dividing into two triangles.

   (c) Partition of an arbitrary convex polygon

   (d) Uniform mesh of a circle

      i. Quasi-uniform mesh program circlemesh is given by MATLAB software package IFEM developed by Prof.Long Chen

      ii. Mesh generator DistMesh in MATLAB

2. Refine the Mesh

   (a) Refine a vertex of a triangle

   (b) Refine a circle near the boundary by constructing a proper distance function for DistMesh using rejection method

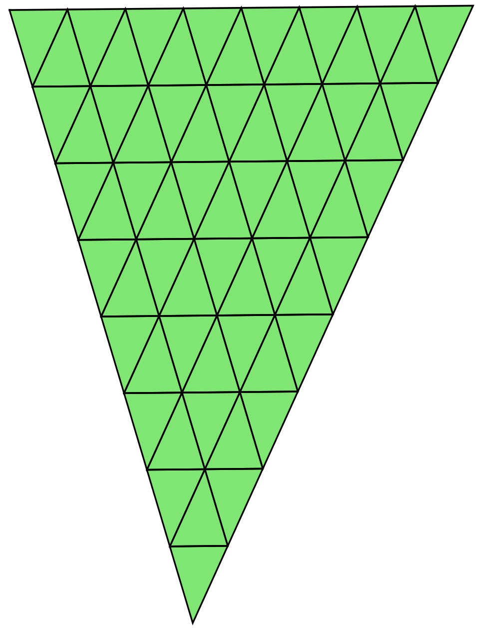


Figure 1: Uniform mesh for an arbitrary triangle

```
function [node,elem,bdFlag]=Mesh4side(m)
node=rand(4,2);
x=node(:,1);
y=node(:,2);
cx=mean(x);
cy=mean(y);
a=atan2(y-cy,x-cx);
[~,order]=sort(a);
node(:,1)=x(order);
node(:,2)=y(order);
elem=[1,2,3;1,3,4];
if m > 1
for i=1:m-1
[node,elem] = uniformrefine(node,elem);
end
else
end
bdFlag = setboundary(node,elem,'Neumann');
showmesh(node,elem);
```

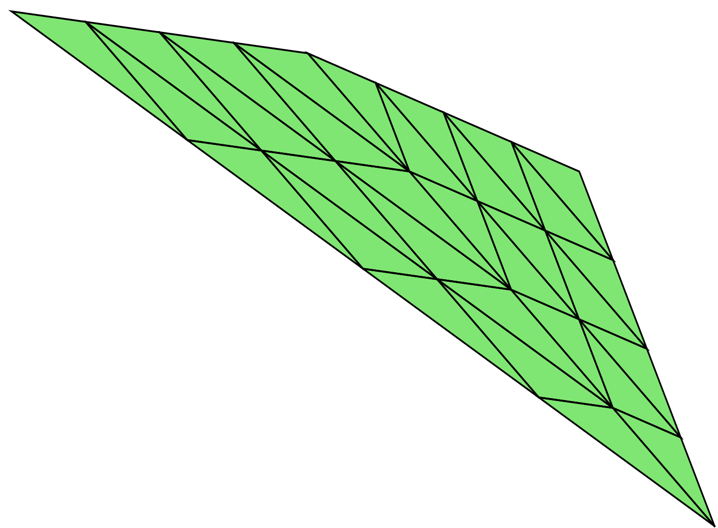


Figure 2: Uniform mesh for an arbitrary quadrangle

```
function [node,elem,bdFlag]=Meshnside(n,m)
a = rand(n,1);
a = 2*pi*(rand+cumsum(a/sum(a)));
r = rand(n,1);
r = r/sum(r);
x = cumsum(r.*cos(a));
y = cumsum(r.*sin(a));
r = cumsum(r);
x = rand+[0;x-x(n)*r];
y = rand+[0;y-y(n)*r];
xo=(x(1)+x(fix(n/2)+1))/2;
yo=(y(1)+y(fix(n/2)+1))/2;
node=[xo yo;x(1:n) y(1:n)];
elem=[ones(n-1,1) (2:1:n)' (3:1:n+1)';1 n+1 2];
if m > 1
for i=1:m-1
[node,elem] = uniformrefine(node,elem);
end
else
end
bdFlag = setboundary(node,elem,'Dirichlet');
showmesh(node,elem);
```

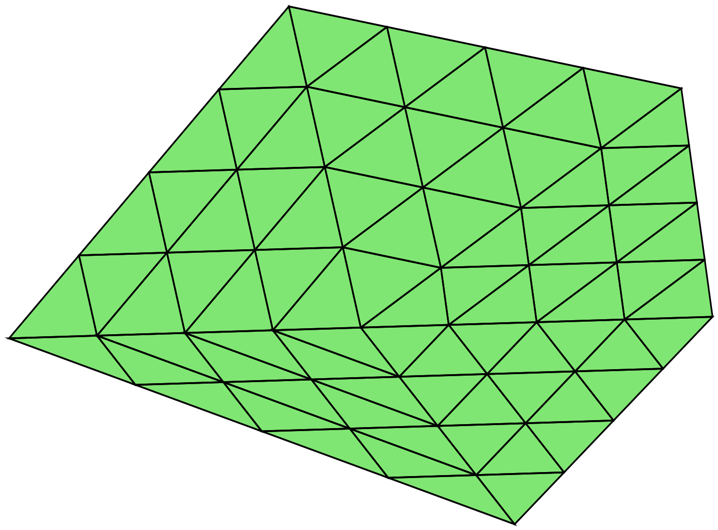


Figure 3: Uniform mesh for an arbitrary pentagon

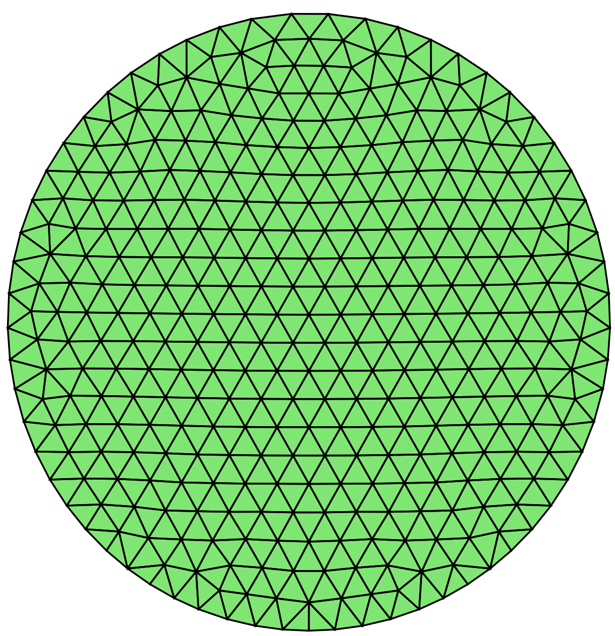


Figure 4: Uniform mesh for a unit circle with size 0.1 by IFEM

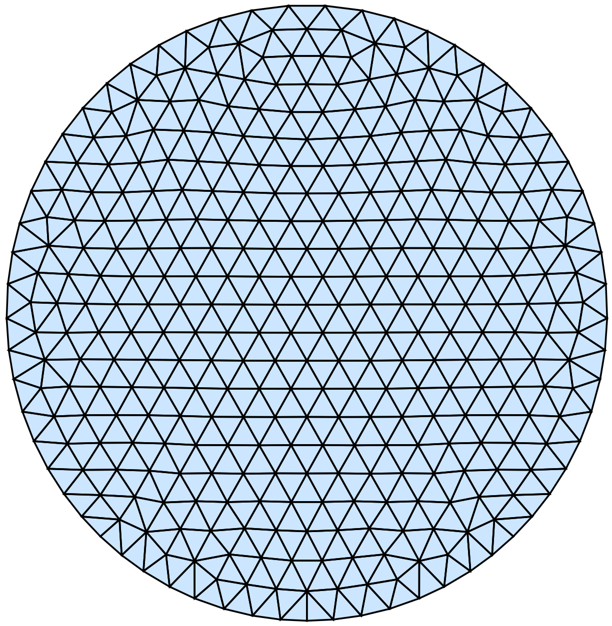


Figure 5: Uniform mesh for a unit circle with size 0.1 by DistMesh

```
function [node,elem,bdFlag]=TriRefine(m,lambda)
node=rand(3,2);
elem=[1,2,3];
bdFlag = setboundary(node,elem,'Dirichlet');
for i=1:m
[node,relem,bdFlag] = uniformrefine(node,elem,bdFlag);
node(relem(1,2),:)=(lambda*node(1,:)+node(elem(1,2),:))/(1+lambda);
node(relem(1,3),:)=(lambda*node(1,:)+node(elem(1,3),:))/(1+lambda);
elem=relem
end
showmesh(node,elem);
```

```
function [p,t]=Circlerefine
a=0.01;
b=0.04;
h0=a+b;
fd=@(p) sqrt(sum(p.^2,2))-1;
fh=@(p) a+b*(1-sqrt(sum(p.^2,2)));
[p,t]=distmesh2d(fd,fh,h0,[-1,-1;1,1],[]);
```

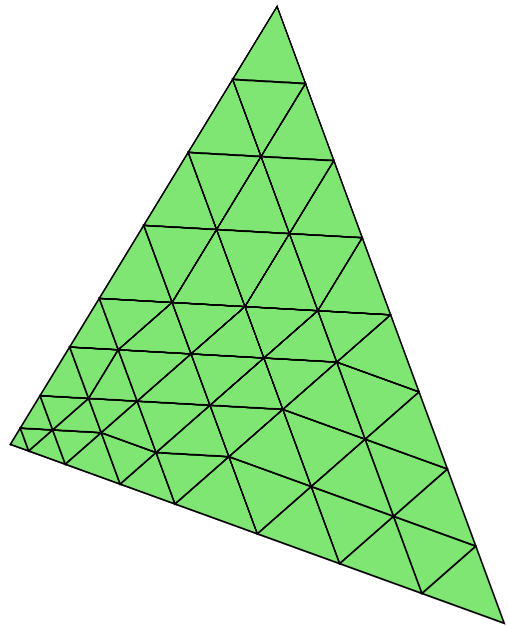


Figure 6: Refine a vertex of a triangle

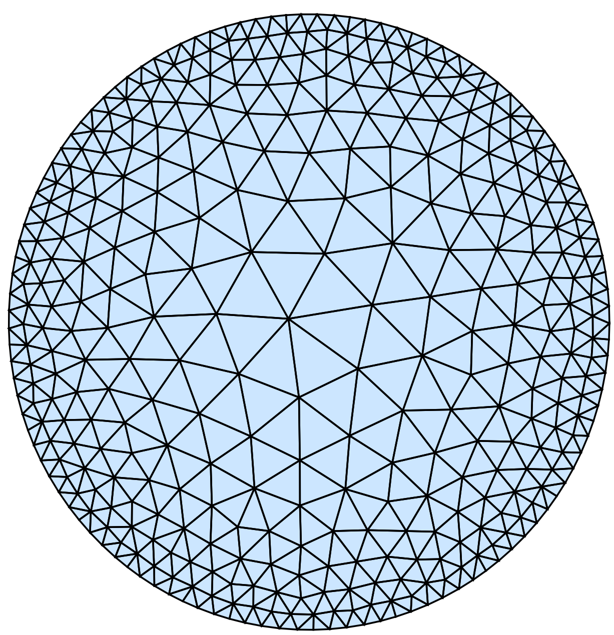


Figure 7: Refine a circle near the boundary

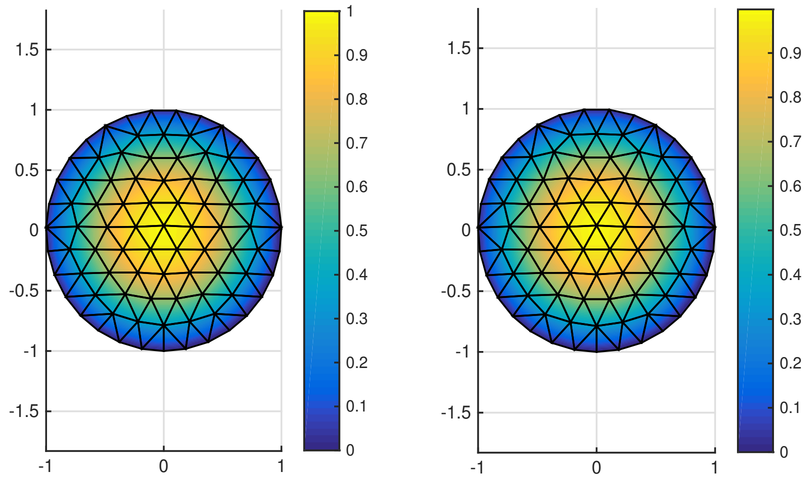
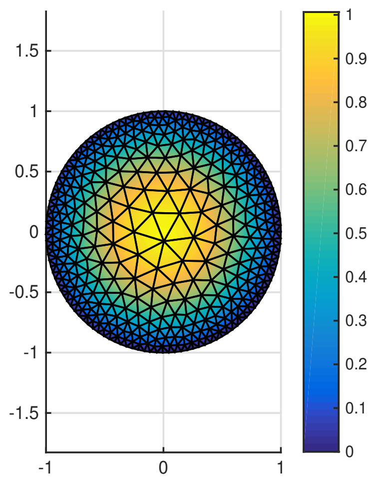


Figure 8: Comparison of numerical solution in uniform mesh u_h (Left , size h=0.2) and exact solution $u = 1 - x^2 - y^2$ (right). The condition number of stiffness matrix is 40.1023. $\|u_h - u\|_{l^1} = 0.1057$, $\|u_h - u\|_{l^2} = 0.1056$ and $\|u_h - u\|_{l^\infty} = 0.073$

2 Solution to Poisson Equation $-\Delta u = 4$ on a Unite Circle in Dirichlet Boundary Condition



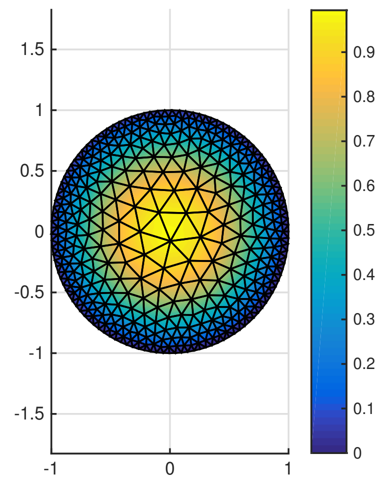


Figure 9: Comparison of numerical solution in refined mesh near the boundary u_h (Left , initial size h=0.2) and exact solution u (right). The condition number of stiffness matrix is 98.4177. $\|u_h - u\|_{l^1} = 0.4287$, $\|u_h - u\|_{l^2} = 0.0387$ and $\|u_h - u\|_{l^\infty} = 0.0123$

3 Application to Diffusion of Optically Pumped Atoms on a Square Domain

We consider a large number of atoms confined in a container. Each atom has an internal state (e.g, quantum mechanical energy levels), denoted by $|\psi>$. The probability distribution of an atom on this state is described by a function $\rho(x,t)$. We consider the following processes :

1. Quantum evolution : $\frac{\partial \rho}{\partial t} = \rho$
2. Spatial diffusion: In general, the atomic state also depends on spatial coordinate r , ie , $\rho = \rho(r)$. The atoms will diffuse from the location with higher density to lower density, which is described by standard diffusion equation.

$$\frac{\partial \rho}{\partial t} = \Delta \rho$$

In summary, within the container (denoted by a domain Ω), the full atomic evolution is governed by

$$\frac{\partial \rho}{\partial t} = \Delta \rho + \rho \quad for \quad r \in \Omega$$

The boundary condition is of Dirichlet type

$$\rho = 1 \quad for \quad r \in \partial\Omega$$

The initial condition is

$$\rho(t=0) = 1 \quad for \quad r \in \Omega$$

We want to find the solution $\rho(r,t)$ of the equation above using finite element method on a unit square domain .

For time direction, we use Back Euler Finite Difference Scheme

$$\frac{\rho^{n+1} - \rho^n}{\tau} = \Delta \rho^{n+1} + \rho^{n+1} \quad for \quad r \in \Omega$$

For each time step, we solve a Helmholtz equation

$$-\tau \Delta \rho^{n+1} + (1-\tau)\rho^{n+1} = \rho^n \quad for \quad r \in \Omega$$

By Principle of Virtual Work

$$\int_\Omega [\tau \nabla \rho^{n+1} \nabla v + (1-\tau)\rho^{n+1} v] dxdy = \int_\Omega \rho^n v dxdy \quad for \quad v \in V$$

where the linear element space on triangulation $\mathbf{T}$ of Ω is

$$V = \{v \in C(\overline{\Omega}) : \quad v|_T \in P_1 \quad \forall T \in \mathbf{T}\}$$

P_1 is the space of linear polynomial. For each vertex v_i of $\mathbf{T}$, let ϕ_i be the piecewise linear function such that $\phi_i(v_j) = \delta_{ij}$. We want to find the numerical solution $\rho_h = \sum_{i=1}^N \rho_i \phi_i$

$$K_{ij} = \sum_T \int_T [\tau \nabla\phi_i \nabla\phi_j + (1-\tau)\phi_i\phi_j] dxdy$$

We now transfer the computation to a reference triangle $\hat{T}$ (spanned by $\hat{v}_1 = (1,0)$, $\hat{v}_2 = (0,1)$ and $\hat{v}_3 = (0,0)$)through an affine map, on computing of the local stiffness matrix. In the reference triangle, $\hat{\phi}_1 = x$, $\hat{\phi}_2 = y$ and $\hat{\phi}_3 = 1 - x - y$. The MATLAB code is as following

```
function Learn(NT)
M=3;
[p,t,bdFlag] = Mesh4side(M);
N=size(p,1);
T=size(t,1);
K=sparse(N,N);
F=zeros(N,1);
dt=0.1;
Area = 0.5/(M + 1)^2;
L=[1/12 1/24 1/24;1/24 1/12 1/24;1/24 1/24 1/12];
U=ones(N,1);
for Nt=1:NT
for e=1:T
nodes=t(e,:);
Pe=[ones(3,1),p(nodes,:)];
C=inv(Pe);
grad=C(2:3,:);
Ke=dt*Area*grad'*grad+(1-dt)*Area*L;
K(nodes,nodes)=K(nodes,nodes)+Ke;
Fe=Area/3*U(nodes);
F(nodes)=F(nodes)+Fe;
end
b=unique(boundedges(p,t));
K(b,:)=0;
K(:,b)=0;
F(b)=1;
K(b,b)=speye(length(b),length(b));
Kb=K;
Fb=F;
U = Kb\Fb;
end
Uh=U;
trisurf(t,p(:,1),p(:,2),0*p(:,1),Uh,'edgecolor','k','facecolor','interp');
view(2),axis([0 1 0 1]),axis equal,colorbar
```

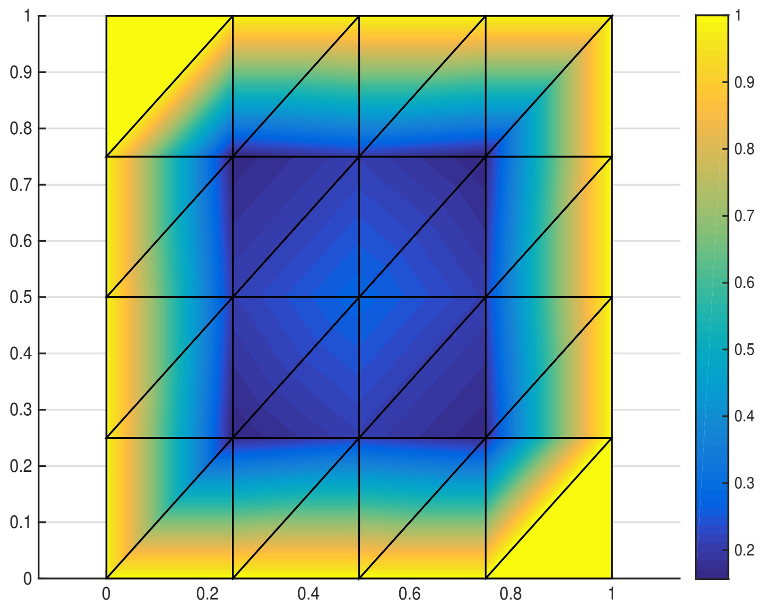


Figure 10: Probability distribution of the atom when t=5 with time step $\tau = 0.1$

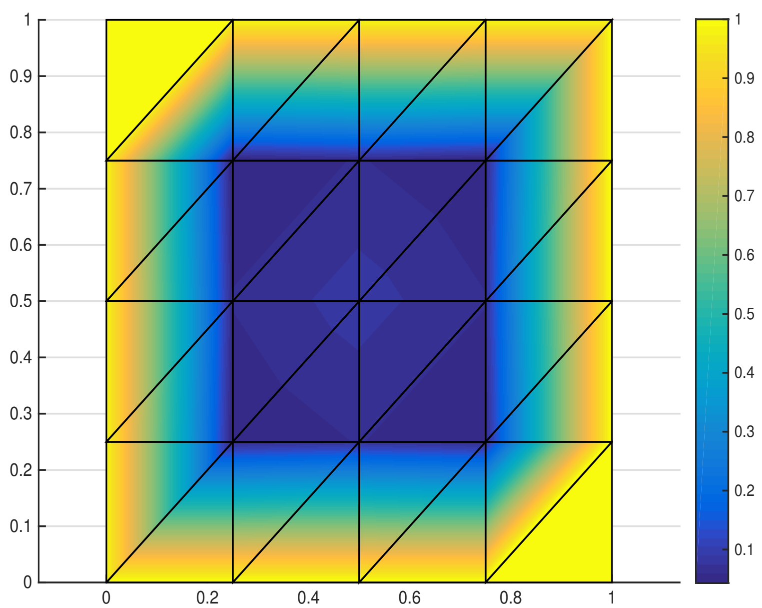


Figure 11: Probability distribution of the atom when t=50 with time step $\tau = 0.1$

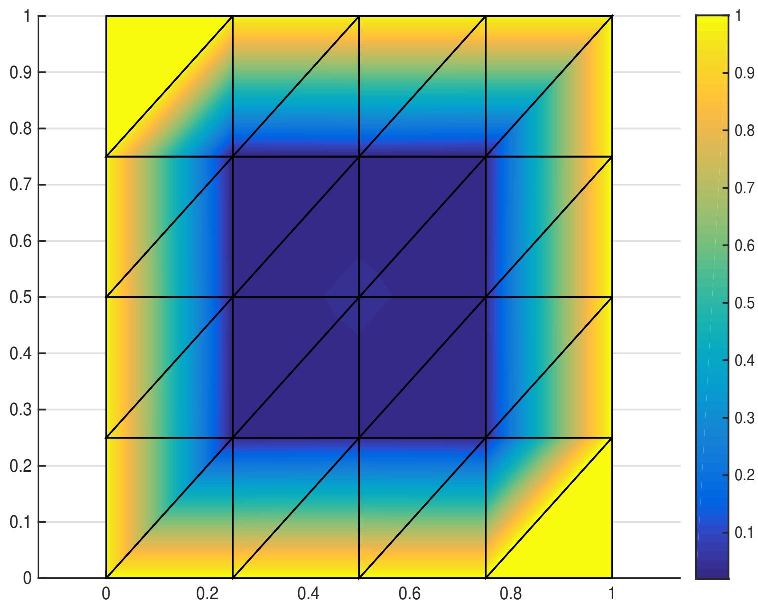


Figure 12: Probability distribution of the atom when t=200 with time step $\tau = 0.1$

4 Finite Element Convergence of Elliptic Problem for Singular Data

We consider the finite element solution u_h of the Dirichlet problem $\Delta u = \delta$ in two dimensions.

$$u(x) = \int_\Omega \delta(y) G(x,y) dy$$

where G(x,y) is corresponding Green's function.

The exact solution u should behave as log r by fundamental solution, which is not belong to H^1. So we consider the L^2 error $\|u - u_h\|_{L^2}$.

We construct a sequence $\{\delta_h\}$ to approximate δ. Let U_h be the finite element solution of the Dirichlet problem $\Delta u = \delta_h$, then by triangle inequality

$$\|u - u_h\|_{L^2} \leq \|u - U_h\|_{L^2} + \|U_h - u_h\|_{L^2}$$

Through the proper construction of $\{\delta_h\}$, we can have $\|U_h - u_h\|_{L^2} \leq Ch$. And we use the more arcane data space as following than the negative Sobolev space to get $\|u - U_h\|_{L^2} \leq Ch$.

$$\Xi^r = \{v \in L^2 \|v\|_{\Xi^r} = \sum_{|\alpha| \leq r} \|\rho^{|\alpha|} D^\alpha v\|_{L^2} < \infty\}$$

Finally, we have $\|u - u_h\|_{L^2} \leq Ch$

5 Optimal Order Convergence of Parabolic Problem for Nonsmooth Initial Data

We assume that S_h is a family of finite dimensional subspaces of L_2, and T_h a family of operators : $L_2 \to S_h$, approximating the exact solution operator of the Dirichlet problem $-\Delta u = f$ such that

1. T_h is selfadjoint, positive semidefinite on L_2, and positive definite on S_h.

2. There is a positive integer $r \geq 2$ such that $\|(T_h - T)f\| \leq Ch^s \|f\|_{s-2}$ for $2 \leq s \leq r$ and $f \in H^{s-2}$

The semidiscrete analogue of the initial boundary value problem for the homogeneous parabolic equation

$$\begin{aligned} u_t - \Delta u = 0 \quad & in \quad \Omega, \quad for \quad t > 0 \\ u = 0 \quad & on \quad \partial\Omega, \quad for \quad t > 0 \\ u(.,0) = v \quad & in \quad \Omega \end{aligned} \tag{1}$$

is then defined as $T_h u_{h,t} + u_h = 0$ for $t > 0$ with $u_h(0) = v_h$.

Then we have for $t > 0$ the error

$$\|u_h(t) - u(t)\|_{L_2} \leq Ch^r t^{-\frac{r}{2}} \|v\|_{L_2}$$

References

[1] John Coady *2D Finite Element Method in MATLAB* FEM-PIC, 2012

[2] Ridgway Scott *Finite Element Convergence for Singular Data* Numer.Math.21 317-327 (1973)

[3] Vidar Thomee *Galerkin Finite Element Methods for Parabolic Problems* Springer, 2006, Second Edition