Project 2

A0109983 Liu Zhaoqiang

April 11, 2015

Problem: Use either Matlab or freefem++ to solve

$$\partial_t u + b(x,t)^T \cdot \nabla u(x,t) = \Delta u + f$$

on $[0,1]^2$ with u = g on the whole boundary. Take $b(x,t) = (1,-2)^T$. For the project, suppose the exact solution is

$$u(t,x,y) = cos(t)cos(6x)sin(6y)e^{x-y}$$

and determine f and g accordingly. Numerically integrate the equation from t = 0 to t = 1. Use P_1 element for the spatial discretization and BDF2+ extrapolation for the temporal discretization. To initialize, u_h^0 is taken to be the interpolation of u^0 and u_h^1 is computed with backward Euler for the diffusion term and explicit extrapolation for the convection term.
Solution: We have the following schemes:

$$(\frac{u_h^1 - u_h^0}{\Delta t}, v_h) + (b(t^1)\cdot \nabla u_h^0, v_h) + (\nabla u_h^1, \nabla v_h) = (f(t^1), v_h) \tag{1}$$

and

$$(\frac{3u_h^n - 4u_h^{n-1} + u_h^{n-2}}{2\Delta t}, v_h) + (b(t^n)\cdot \nabla(2u_h^{n-1} - u_h^{n-2}), v_h) + (\nabla u_h^1, \nabla v_h) = (f(t^n), v_h) \tag{2}$$

Suppose n is the number of small intervals in x-axis and y-axis, and let h be the spatial step size, then $h\cdot n = 1$. Δt is the time step size. Denote $N = n+1, M = N^2$. Let the basis functions be φ_i, i=1,2,...,M, then to use the first scheme to solve u_h^1, we assume

$$u_h^1 = \sum_{j=1}^{M} \xi_j^1 \varphi_j, u_h^0 = \sum_{j=1}^{M} \xi_j^0 \varphi_j$$

where ξ_j^0 is already known, and we need to solve ξ_j^1 via first schme.
Let

$$A = (a_{ij}), a_{ij} = (\nabla\varphi_i, \nabla\varphi_j)$$
$$B = (b_{ij}), b_{ij} = (\varphi_i, \varphi_j)$$
$$C = (c_{ij}), c_{ij} = (\varphi_i, \partial_x\varphi_j)$$
$$D = (d_{ij}), d_{ij} = (\varphi_i, \partial_y\varphi_j)$$

Then the first scheme leads to

$$B\frac{\xi^1-\xi^0}{\Delta t}+b(t^1)_1C\xi^0+b(t^1)_2D\xi^0+A\xi^1=F(t^1) \tag{3}$$

where $\xi^1=(\xi_1^1,...,\xi_M^1)^T, \xi^0=(\xi_1^0,...,\xi_M^0)^T$, $b(t^1)=(b(t^1)_1,b(t^1)_2)^T$, $F(t^1)=((f(t^1),\varphi_1),...,(f(t^1),\varphi_M)^T$. Thus, we have:

$$(B+\Delta tA)\xi^1=\Delta tF(t^1)+B\xi^0-\Delta tb(t^1)_1C\xi^0-\Delta tb(t^1)_2D\xi^0$$

If we got A,B,C,D already, it is easy to solve ξ^1 from the above equation.
When use the rectangle element as in previous project, we have:
(1) if the k-th node is a inner point,

$$A(k,k)=\frac{8}{3}, A(k,k+1)=A(k,k-1)=A(k,k+N)=A(k,k-N)=-\frac{1}{3},$$
$$A(k,k-N-1)=A(k,k-N+1)=A(k,k+N-1)=A(k,k+N+1)=-\frac{1}{3}$$
$$B(k,k)=\frac{4h^2}{9}, B(k,k+1)=B(k,k-1)=B(k,k+N)=B(k,k-N)=\frac{h^2}{9},$$
$$B(k,k-N-1)=B(k,k-N+1)=B(k,k+N-1)=B(k,k+N+1)=\frac{h^2}{36}$$
$$C(k,k)=C(k,k+N)=C(k,k-N)=0, C(k,k+1)=\frac{h}{3}, C(k,k-1)=-\frac{h}{3},$$
$$C(k,k-N-1)=C(k,k+N-1)=-\frac{h}{12}, C(k,k-N+1)=C(k,k+N+1)=\frac{h}{12}$$
$$D(k,k)=D(k,k+1)=D(k,k-1)=0, D(k,k+N)=\frac{h}{3}, D(k,k-N)=-\frac{h}{3},$$
$$D(k,k-N-1)=C(k,k-N+1)=-\frac{h}{12}, C(k,k+N-1)=C(k,k+N+1)=\frac{h}{12}$$

(2) if the k-th node is a boundary point, we just need to let $B(k,k)=1, A(k,:)=0, C(k,:)=0, D(k,:)=0$.

When use the P_1 triangle element, we have:
(1) if the k-th node is a inner point,

$$A(k,k)=4, A(k,k+1)=-1, A(k,k-1)=-1, A(k,k+N)=-1, A(k,k-N)=-1;$$
$$B(k,k)=\frac{h^2}{2}, B(k,k+1)=B(k,k-1)=B(k,k+N)=\frac{h^2}{12},$$
$$B(k,k-N)=B(k,k-N-1)=B(k,k+N+1)=\frac{h^2}{12};$$
$$C(k,k)=0, C(k,k+1)=\frac{h}{3}, C(k,k-1)=-\frac{h}{3}, C(k,k+N)=-\frac{h}{6},$$
$$C(k,k-N)=\frac{h}{6}, C(k,k-N-1)=-\frac{h}{6}, C(k,k+N+1)=\frac{h}{6};$$
$$D(k,k)=0, D(k,k+1)=-\frac{h}{6}, D(k,k-1)=\frac{h}{6}, D(k,k+N)=\frac{h}{3},$$
$$D(k,k-N)=-\frac{h}{3}, D(k,k-N-1)=-\frac{h}{6}, D(k,k+N+1)=\frac{h}{6};$$

(2) if the k-th node is a boundary point, we just need to let $B(k,k)=1, A(k,:)=0, C(k,:)=0, D(k,:)=0$.

To use the second scheme to solve $u_h^n, n\geq 2$, the procedure is similar.

Numerical results
Table1. Using rectangle element(compute to T=1)

n and Δt	**L^∞ error**	**L_2 error**
n=10,Δt=0.01	0.0262	0.0085
n=20,Δt=0.005	0.0066	0.0021
n=40,Δt=0.0025	0.0017	5.2904e-04
n=80,Δt=0.00125	4.1900e-04	1.3222e-04

Loglog graphs are shown below.

Using polyfit function in Matlab, we have:

$$log(L^\infty \quad error)=1.9856log(h)+0.9322$$
$$log(L_2 \quad error)=2.0008log(h)-0.1649$$

Table2. Using P_1 triangle element(compute to T=1)

n and Δt	**L^∞ error**	**L_2 error**
n=10,Δt=0.01	0.0240	0.0104
n=20,Δt=0.005	0.0061	0.0026
n=40,Δt=0.0025	0.0015	6.4827e-04
n=80,Δt=0.00125	3.8234e-04	1.6202e-04

Figure 1: loglog graph for L^∞ error

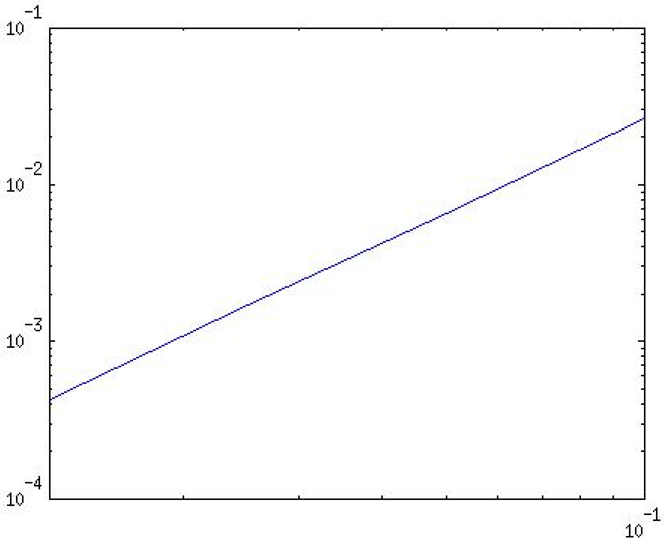
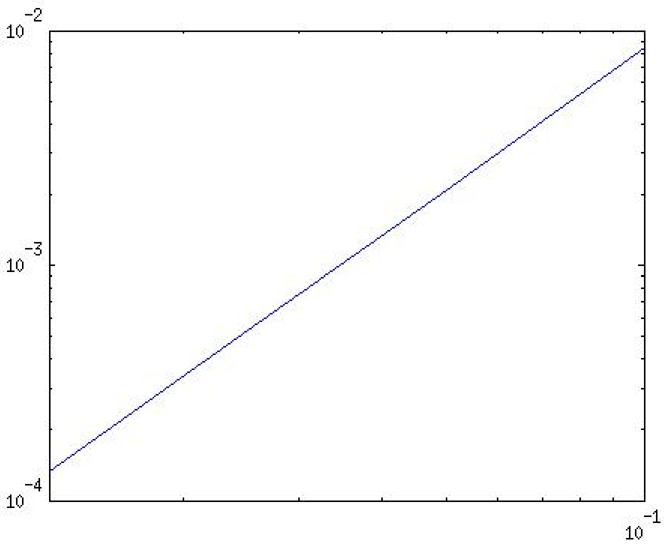


Figure 2: loglog graph for L_2 error



The loglog graphs are similar with that of using rectangle element. Here we omit the graphs.
Using polyfit function in Matlab, we have:

$$log(L^{\infty} \quad error) = 1.9940 log(h) + 0.8644$$
$$log(L_2 \quad error) = 2.0017 log(h) + 0.0434$$