PROJECT # 2

The two-dimensional heat equation is given on a domain $[0, 1] \times [0, 1]$ by

$$\frac{\partial u}{\partial t} = \alpha \left[\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} \right] \tag{1}$$

with the Drichlet boundary conditions:

$$\psi(x, 0, t) = 0 \tag{2}$$
$$\psi(x, 1, t) = (x - x^2) \tag{3}$$
$$\psi(0, y, t) = 0 \tag{4}$$
$$\psi(1, y, t) = 0 \tag{5}$$

and initial condition of $u(x, y, 0) = 0$. The α parameter is equal to 0.1.

Use the following numerical methods

1. Galerkin finite element method with Euler explicit
2. Galerkin finite element method with Euler implicit
3. Galerkin finite element method with Crank-Nicolson

The error function is given by

$$\text{Error} = \frac{\|u_{i,j} - u_{analytic}\|_2}{\sqrt{i_{max} j_{max}}} \tag{6}$$

Employ uniform $i_{max} \times j_{max} = 41 \times 41$, 81×81 and 161×161 grid resolutions. For the time step use $4\alpha\Delta t/\Delta x^2 = 0.5$, $4\alpha\Delta t/\Delta x^2 = 1$ and $4\alpha\Delta t/\Delta x^2 = 2$. Compute the numerical solutions at $t = 1.0$. Then compute the steady state solution for $t \to \infty$. Finally, draw the error versus mesh resolution and determine the spatial convergence rate. For the solution of the linear algebraic systems use GMRES algorithm (Saad and Schultz 1986) with the incomplete LU preconditioner (optional). Compare the magnitude of error with the finite difference solution from the first project.

[0]UUT510E, Return date: 24 April 2014

Several useful MATLAB commands:
Crate a sparse matrix
i=[];
j=[];
s=[];
m=100;
n=100;
A=sparse(i, j, s, m, n);
To solve a sparse linear system
$x = A\backslash b$;
For incomplete ilu preconditioner
$[L, U]$ = luinc(A,1e-5);
For GMRES(m) solver
x=gmres(A,b,m,r_{tol},maxit,$M1$,$M2$,x_0);

Read Mesh Vetices
for $i \leftarrow 1$ **to** np **do**
  read x[i],y[i],z[i]
end

Read Mesh Connectivity
for $i \leftarrow 1$ **to** ne **do**
  read nec[i,1],nec[i,2],nec[i,3],nec[i,4]
end

Create Global Coefficient Matrix
for $i \leftarrow 1$ **to** ne **do**
  call Mass_Matrix(i,x,y,nec,[M])
  call Stiffness_Matrix(i,x,y,nec,[K])
  $[A] := [A] + [M] * \frac{1}{\Delta t} + [K]$
  $\{RHS\} := \{RHS\} + [M] * \frac{1}{\Delta t}\{u\}$
end

Impose Drichlet Boundary Conditions
for $i \leftarrow 1$ **to** np **do**
  if *Dirichlet boundary condition is valid for* i **then**
    $A[i,*] := 0$
    $A[i,i] := 1$
    $RHS[i] := 0$
  end
end

Solve Ax=RHS

Table 1: The structure of the Galerkin FEM code with Euler implicit.

for $p \leftarrow 1$ **to** $n-1$ **do**
  $d := 1/a_{p,p}$
  for $i \leftarrow p+1$ **to** n **do**
    if $(i,p) \in S$ **then**
      $e := a_{i,p} * d$
      $a_{i,p} := e$
      for $j \leftarrow p+1$ **to** n **do**
        if $(i,j) \in S$ *and* $(p,j) \in S$ **then**
          $a_{i,j} := a_{i,j} - e * a_{p,j}$
        end
      end
    end
  end
end

Table 2: The algorithm for computing ILU(0) for a n by n matrix A is given above. Here S represents the set of elements of matrix A.