Università della Svizzera italiana

Facoltà di scienze informatiche

Introduction to PDEs (LAB) **Academic Year 2015/2016**

Instructor: Prof. Rolf Krause TA: Drosos Kourounis

Assignment 3 - Finite Element Solution of Poisson's equation in 2D

Due date: Thursday 8 October 2015, 10:30

Solution of Poisson's equation

We seek the discrete solution of Poisson's equation

$$\begin{aligned} -\nabla^2 u(x,y) &= f(x,y), \quad \in \Omega \\ \frac{\partial u}{\partial n} &= 0, \quad (x,y) \in \partial\Omega, \end{aligned}$$

where $\Omega = [0,1]^2$.

1. Derive the analytical expression of $f(x,y)$ so that the exact solution of the PDE is

$$u_0(x,y) = x^2\, y^2\ (2\,x-3)\ (2\,y-3)\,.$$

2. Verify that the exact solution satisfies the homogeneous Neumann conditions at the boundary $\partial\Omega$.

3. Write MATLAB code that solves the PDE implementing the steps following[1]

1. Mesh generation

Write a MATLAB function that generates a $N_x \times N_y$ quadrilateral grid. The input arguments of the function should be `N_x`, `N_y`, `grid_type`. If third argument `grid_type` is equal to `'triangles'` then you can easilly generate a triangular grid by dividing each quadrilateral to 2 triangles drawing the diagonal connecting the local points 1 and 3. The output of the function should be the matrix of the connectivity of the grid (`Elements`) and the matrix of the coordinates of the grid points (`Points`). The matrix `Elements` should by of size $N_e \times N_{\text{v}}$, where the N_e number of elements and N_{v} the number of vertices. The matrix `Points`, should be of size $N \times d$, where N the number of points and d the number of space dimensions.

[1] You should include in your submission only the implementations of the functions marked with red and not the whole MATLAB code.

```
function [mesh] = makeGrid(L_x, L_y, N_x, N_y, grid_type)
```

The output `mesh` is a MATLAB structure that contains all the information that are needed for the mesh generation and mesh description, namely, `delta_x`, `delta_y`, `N_x`, `N_y`, `L_x`, `L_y`, `N_e`, `N_v`, `N`, `Elements`, `Points`, `PointMarkers`. The array `PointMarkers` should be zero at every point except those points that belong to the boundary. We will need it later to enforce Dirichlet boundary conditions to our Discrete system of equations. Use the MATLAB function `writeMeshAsVTKFile.m` in the `matlab` folder and call it passing the structure mesh and the name of the output file. This will dump the grid as a VTK file that you can visualize using visualization package ViSiT. The declarations follow.

```
function writeMeshAsVTKFile(mesh, vtkfile)
```

2. Finite element assembly

Use the following code sample for the assembly of your discrete operators.

```
function [M,K,b] = assembleDiscreteOperators(mesh)
  N   = size(mesh.Points, 1);
  Ne  = size(mesh.Elements,1)
  Nv  = size(mesh.Elements,2)
  M   = zeros(N,N); K = zeros(N,N);
  b   = zeros(N,1);
  for e=1:Ne
    Me = makeMe(e, mesh);
    Ke = makeKe(e, mesh);
    fp = makebe(e, mesh);
    for i=1:Nv
      I = Elements(e, i);
      for j=1:Nv
        J = Elements(e, j);
        M(I, J) = M(I, J) + Me(i, j);
        K(I, J) = K(I, J) + Ke(i, j);
      end % j loop
      be   = Me*fp;
      b(I) = b(I) + be(i);
    end % i loop
  end % e loop
end
```

Provide implementations of the individual functions needed for forming the element mass matrix `M_e`, element Laplacian `K_e` and the element rhs `b_e`.

```
function Me = makeMe(e, mesh);
function Ke = makeKe(e, mesh);
function fp = makebe(e, mesh);
```

Keep in mind that `f_p` should hold the values of the function $f(x, y)$ evaluated at the nodes of the eth element, for example $f_p = [f(x_1^e, y_1^e)\ f(x_2^e, y_2^e)\ f(x_3^e, y_3^e)]^T$.

3. Solution

Provide a function that solves the linear system $Ku_h = b$, and use it to obtain the solution of the linear system you constructed in the previous step. Can the system be solved? Any ideas why? Think of a way to enforce in the discrete system $Ku_h = b$

the Dirichlet boundary condition $u_h(1) = 0$ at the first point of the grid, the point (0,0). Is this condition consistent with the exact solution? If yes why?

```
function x = solve(K, b)
```

4. Convergence study

Compare the solution you obtained with the exact one in the first part $u_0(x, y)$ by computing the L^2 and H^1 norms of the error and plot them using logarithmic scale on both axes as a function of $h = \max(\delta x, \delta y)$ for $N_x = N_y = 5, 10, 20, 40$. Note that the discrete L^2 and H^1 norms are easilly obtained from

$$\|u - u_h\|_{L^2} = \sqrt{(u - u_h)^T M (u - u_h)}$$
$$\|u - u_h\|_{H^1} = \sqrt{(u - u_h)^T M (u - u_h) + (u - u_h)^T K (u - u_h)},$$

where u_h is the vector of the discrete finite element solution you obtained by solving the linear system $Ku_h = b$ and u is the vector of the exact values of the solution. The exact solution is given by $u_0(x, y)$ evaluated at the grid points.