Study of the effect of ignoring the boundary curvature in numerical simulations

To study the numerical artifact caused by ignoring the curvature of the boundary I will use a manufactured solution for the Poisson problem and then calculate the error norms with mesh refinement.

The problem is defined as:

$$\begin{aligned}
&-\triangle u = f(r,\theta) \qquad \frac{1}{4} \le r \le 1 \qquad 0 \le \theta \le \frac{\pi}{2} \\
&\partial_n u(r,0) = \partial_n u(r,\frac{\pi}{2}) = 0 \\
&u(\frac{1}{4},\theta) = u(1,\theta) = 0
\end{aligned}$$

Where the source term and the exact solution are defined as:

$$\begin{aligned}
&u_e(r,\theta) = \sin(\frac{4\pi}{3}(r-\frac{1}{4}))e^{\cos 2\theta} \\
&f(r,\theta) = e^{\cos\theta}\left(-\frac{4\pi}{3}\frac{1}{r}sin(\frac{\pi}{6}(5-8r)) + \frac{16\pi^2}{9}cos(\frac{\pi}{6}(5-8r)) + \frac{4}{r^2}sin(\frac{4\pi}{3}(r-\frac{1}{4}))(\cos\theta - \sin^2\theta)\right)
\end{aligned}$$

I will use a Galerkin discretization based on quadratic Lagrange elements to solve this problem on three different meshes containing 206, 724, and 2864 elements respectively. The second mesh is shown in Figure 1. The meshes are generated using gmsh [1] while the code is written using the libMesh[2] finite element library. Although the problem is presented in the polar coordinates, the code actually uses Cartesian coordinates. I will consider three cases for the boundary conditions:

C Use boundary conforming curved elements for $r = 0.25, 1$ and impose Dirichlet boundary condition with $u = 0$ on them.

LH Use linear mapping quadratic elements for $r = 0.25, 1$ and impose Dirichlet boundary condition with $u = 0$ on them. The trick here is that after the mesh is loaded into libMesh's data structure, I will loop through the elements that have an edge on the boundary, and displace the middle node of that edge to the point between the side vertices.

LE Use linear mapping quadratic elements for $r = 0.25, 1$ yet impose $u = u_e$ on them. Of course this is not possible for a real problem. However, it is good idea to have it here to compare the results. We expect correct convergence rate in this case as well.

The convergence of the numerical solution with respect to mesh refinement is shown in Figure 2. In the LH case the slope of the H^1 norm has dropped from 2 to 1.5 which is consistent with the lecture notes. The performance in terms of the L_2 norm is even worse, i.e. increasing the degrees of freedom did not result in a better result compared to linear elements. As expected the LE case performed the same as the C case, which can be viewed as a sign of verification.

Figure 3 shows the L_2 average of the error in each control volume for the midsize mesh. We can see that in the LH case the error is much bigger around the boundary, although in the interior region it has the same magnitude as the C case.

[1] http://gmsh.info/
[2] http://libmesh.github.io/

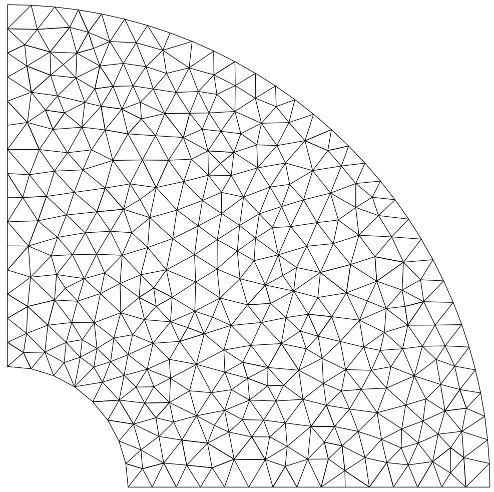

Figure 1: The 724 element mesh.

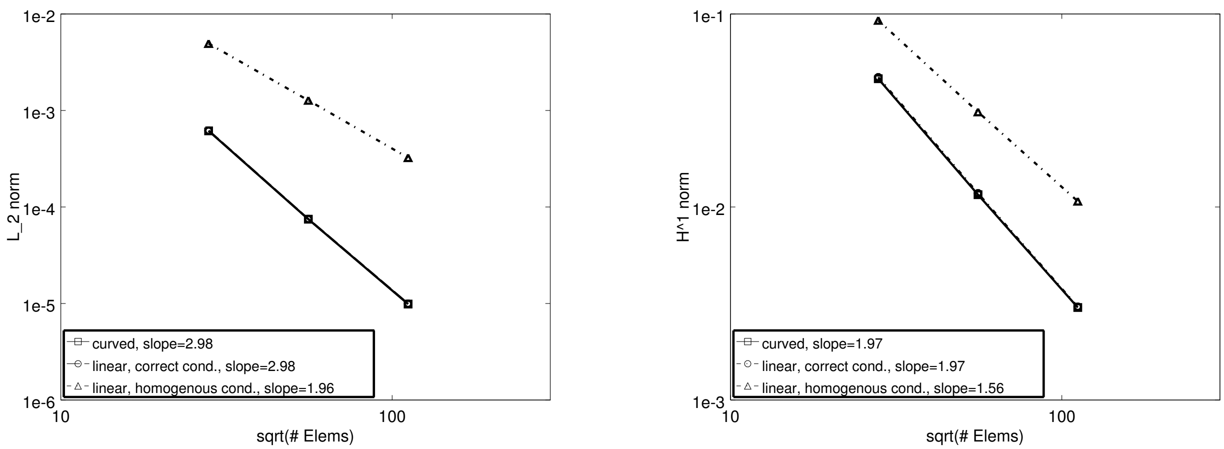

Figure 2: L_2 and H^1 norms of the solution with respect to mesh refinement.

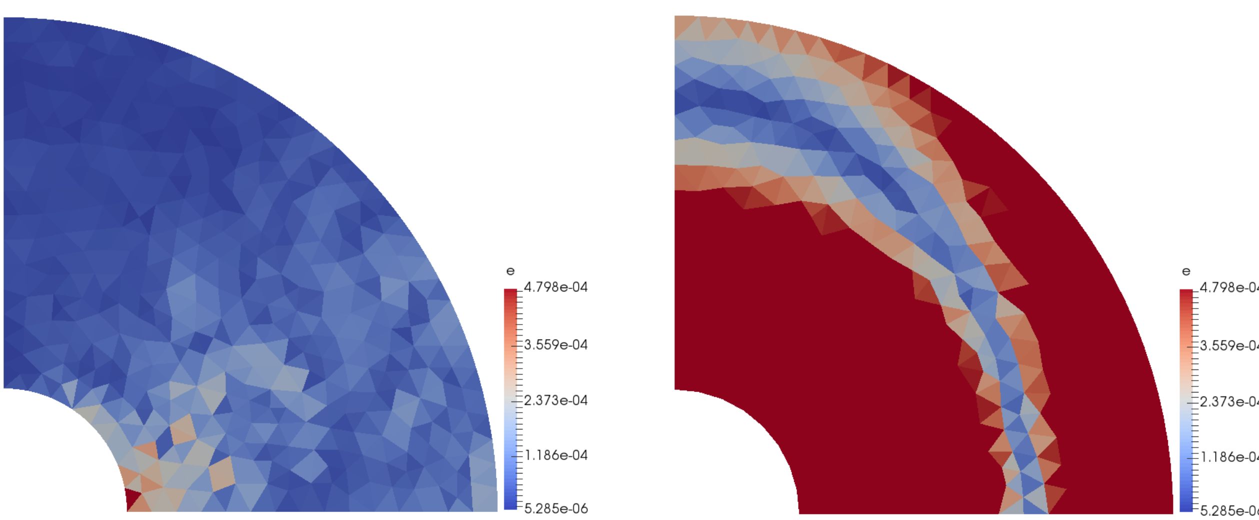

Figure 3: The L_2 average of error integrated inside each element for the 724 element mesh. left C, right LH.