MA4245 Mathematical Principles of Galerkin Methods
Project 3: 2D Poisson Equation

Prof. Frank Giraldo
Department of Applied Mathematics
Naval Postgraduate School
Monterey, CA 93943-5216

Due: May, 20 2021 at 1pm

1 Continuous Problem

The governing partial differential equation (PDE) is

$$\boldsymbol{\nabla}^2 q(x,y) = f(x,y) \qquad \forall (x,y) \in [-1,1]^2$$

where $f = -2\pi^2 \sin \pi x \sin \pi y$ which yields the analytic solution $q(x,y) = \sin \pi x \sin \pi y$. Clearly, this problem represents a Poisson problem which is a generalized elliptic equation and is, therefore, a boundary value problem.

2 Simulations

Use the CG method to solve this equation. You need to write the code to handle arbitrarily-sized elements, polynomial orders, and integration formulas. What I mean by arbitrarily-sized elements is that you should not assume that each element is of the same size. You can, however, assume that each element uses the same polynomial and integration orders.

2.1 Results You Need to Show

You must show results for linear elements $N = 1$ with increasing number of elements N_e and then show results for $N = 2$, $N = 4$, $N = 8$, and $N = 16$ with increasing numbers of elements. Plot L^2 error norms (defined below) versus number of points (N_P) and show all 5 curves on one plot. Remember that you must use a log plot for the error to capture the spectral convergence. To confirm that you have not assumed anything specific about the geometry, use the rotate_grid (matlab) or warp_grid (julia) in the driver files to rotate the grid. You should expect similar results to the unrotated grid but will not be identical. Show me a plot with the error for one grid configuration (e.g., $N = 4$, $Q = 5$, $N_e = 4$).

For the following simulations, you must turn in two plots: one for exact integration and another for inexact. Write a discussion on your findings.

N=1 Simulations For $N = 1$ use $N_e = 4, 8, 16, 24$, and 32 elements.

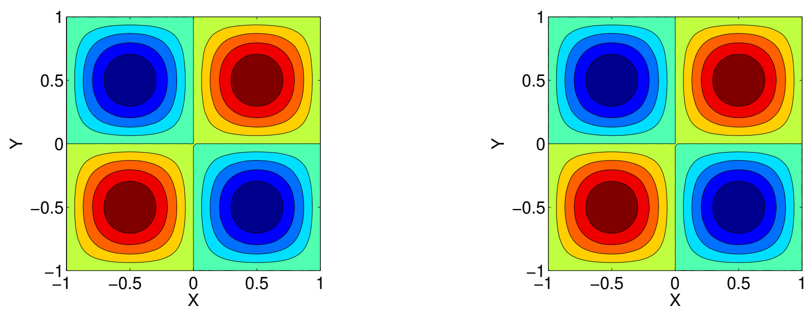
N=2 Simulations For $N = 2$ use $N_e = 2, 4, 8, 12$ and 16 elements.

N=4 Simulations For $N = 4$ use $N_e = 1, 2, 4, 6$, and 8 elements.

N=8 Simulations For $N = 8$ use $N_e = 1, 2, 3$, and 4 elements.

N=16 Simulations For $N = 16$ use $N_e = 1$ and 2 elements.

Here is what the analytic and numerical solutions should look like:

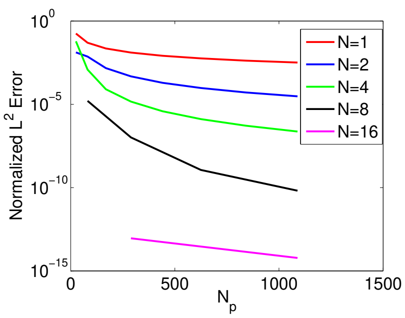


a) Analytic Solution

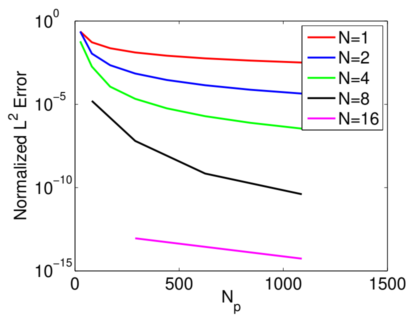
b) CG Numerical Solution

Figure 1: The 2D Elliptic equation solutions for a) analytic and b) CG numerical solutions using $n_r = 4$ elements in each direction ($N_e = 16$) and $N = 16$ order polynomials.

Here is an example of the kind of plot you should show me:



a) Exact Integration



b) Inexact Integration

Figure 2: The Convergence Rates for CG using a) exact and b) inexact integration.

3 The Write-Up

In your write-up, please include the convergence rate plots as above and give a detailed discussion of how you solved the problem. E.g, how did you construct the basis functions, mass matrix, and Laplacian matrix, and how you imposed the homogeneous Dirichlet boundary conditions. Tell me what you learned about using CG for elliptic problems. Also include a complexity analysis of your code (i.e., how many mathematical floating point operations the code required between exact and inexact integration). Assuming that you could write the code only for inexact integration, how much cheaper would that be?

4 Helpful Relations

Error Norm The normalized L2 error norm that you should use is:

$$||error||_{L^2} = \sqrt{\frac{\sum_{k=1}^{N_P} \left(q_k^{numerical} - q_k^{exact}\right)^2}{\sum_{k=1}^{N_P} \left(q_k^{exact}\right)^2}} \tag{1}$$

where $k = 1, ..., N_P$ are $N_p = (N_e N + 1)^2$ global gridpoints and $q^{numerical}$ and q^{exact} are the numerical and exact solutions.