Daniel Brooker

Scientific Computing

Final Project Report

Methods of Approximation for PDE's

The class of mathematical equations known as partial differential equations are extremely useful for modeling the behavior of many different natural phenomena in engineering and physics. However, the solutions to these problems vary depending on boundary conditions, initial conditions, and the specific equation. Finding an exact solution can prove difficult, especially with conflicting boundary conditions. This is where a numerical approach comes in. Creating an approximation that can work for multiple situations and has adjustable accuracy is a valuable tool. This paper presents two problems with two types of solutions to each.

The first Problem introduced is the two-dimensional Poisson equation. It has 4 boundary conditions that form a rectangle. For Dirichlet type conditions the solution values are fixed while for Neumann type, the values for derivative of the solution are fixed. When the right side is zero this indicates a homogeneous problem. In this case there is a particular solution. The equation is given as:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -F(x,y)$$

With right side equal to:

$$F(x,y) = \cos\left[\frac{\pi}{2}\left(2\frac{x-a_x}{b_x-a_x}+1\right)\right] \sin\left[\pi\frac{y-a_y}{b_y-a_y}\right]$$

I solved for both -F(x,y) = cos(0.5*x) + cos(0.5*y) and u(x,y) = -2* cos(0.5*x) + cos(0.5*y). These do satisfy the above equations, but do not hold up for the boundary conditions. Boundary conditions included three Dirichlet and one Neumann. Ghost nodes and the discretization used

$$\nabla_x^2 u_{jk} = \frac{u_{j-1,k} - 2u_{j,k} + u_{j+1,k}}{\Delta x^2}, \qquad \nabla_y^2 u_{jk} = \frac{u_{j,k-1} - 2u_{j,k} + u_{j,k+1}}{\Delta y^2}$$

is:

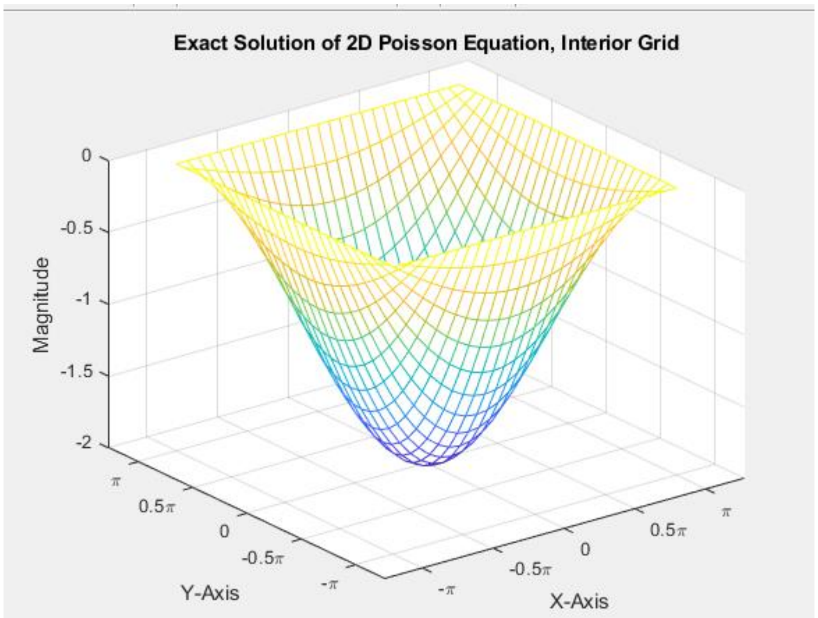
When these are substituted into the original equation, they create:

Rearranging into matrices yields the form of AU = -F. From here we can perform the two linear

$$\frac{u_{jk}^{n+1} - u_{jk}^{n}}{\Delta t} = D\left(\nabla_x^2 u_{jk}^n + \nabla_y^2 u_{jk}^n\right)$$

solving methods, Gaussian elimination and Gauss-Seidel method.

Gaussian elimination was performed with 20 nodes in x and y directions each. The average error was determined as the average difference between all points in the approximate and exact solutions. Max error is defined by the largest difference in value for a point. Gaussian elimination yields an average error of 5.9916 and max error of 124.0251. Gauss-Seidel with 20 nodes in x and y with a single iteration yielded average error of 5.1857 and max error of 124.0251. The average error appeared to max out at 5.1394 for 8 iterations. Increasing number



of nodes decreased the average error for both methods. Boundary values near x = pi, y = -pi skewed the max error. Most of approximate and exact values have magnitude close to zero.

Figure 1

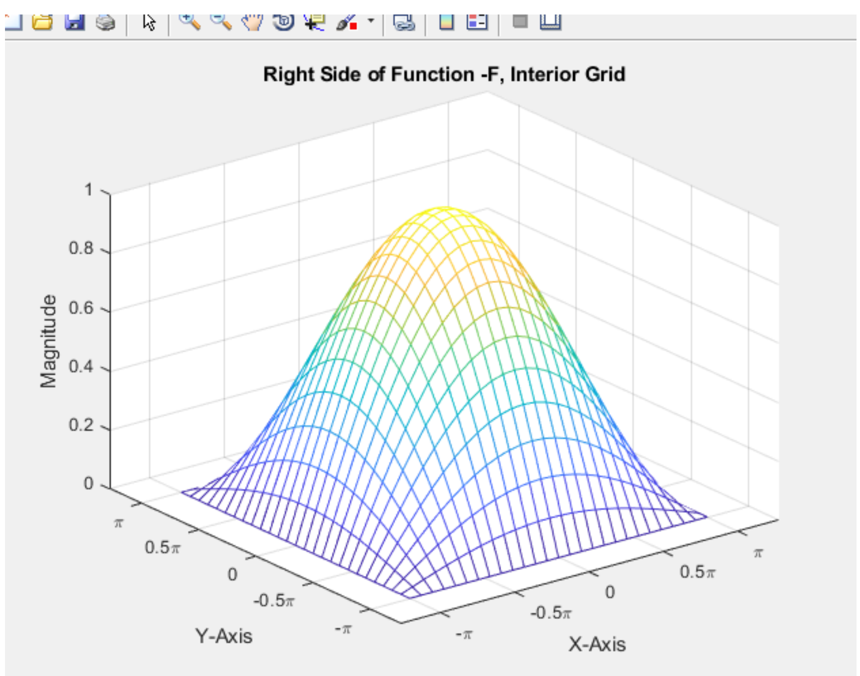


Figure 2

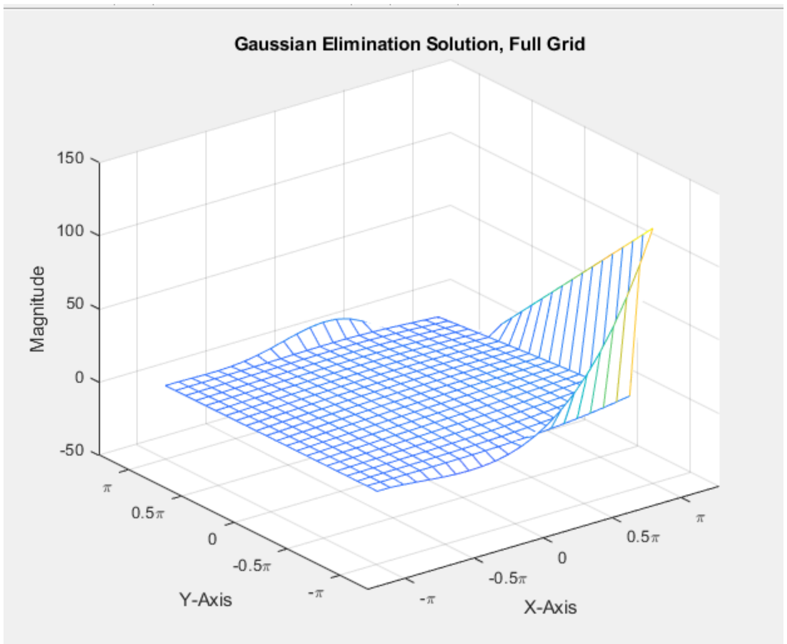


Figure 3: Note difference in z axis between plots.

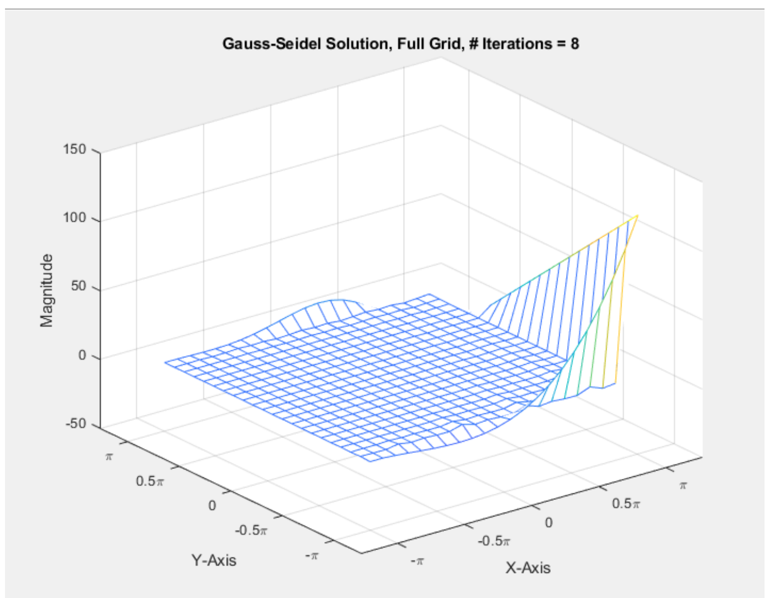


Figure 4

The second problem addressed is the two dimensional diffusion equation. The equation is given as:

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

Boundary conditions are like last time except the Neumann and one Dirichlet switched sides. The two integration methods used in this case were explicit and implicit discretization. Laplacian terms have the same form as before. The explicit discretization is:

$$\frac{u_{jk}^{n+1} - u_{jk}^{n}}{\Delta t} = D\left(\nabla_x^2 u_{jk}^n + \nabla_y^2 u_{jk}^n\right)$$

With implicit discretization being:

$$\frac{u_{jk}^{n+1} - u_{jk}^{n}}{\Delta t} = D\left(\nabla_x^2 u_{jk}^{n+1} + \nabla_y^2 u_{jk}^{n+1}\right)$$

The difference being that the first evaluates the right at some given time t, while the second evaluates at the next time step, t + delta t. Solving for an exact solution of this problem was not possible so I decided to compare the average difference in values of the integration approximations. The following includes values for different numbers of nodes:

Nodes	Average difference of approximations	Max difference for of approximate
5	0.5507	4.7732
10	0.5291	11.0486
15	0.5768	18.7567
20	0.6519	28.2294
25	0.7331	39.6915

The average difference grows slowly and remains below one. On the other hand, max difference continues to grow. I believe the reason for both is that as more nodes are included, the effect of the extreme boundary values has a greater influence on the rest of the nodes. For both PDE's I believe that having more consistent boundary conditions leads to better results. Simply increasing the number of nodes does not solve the problem.

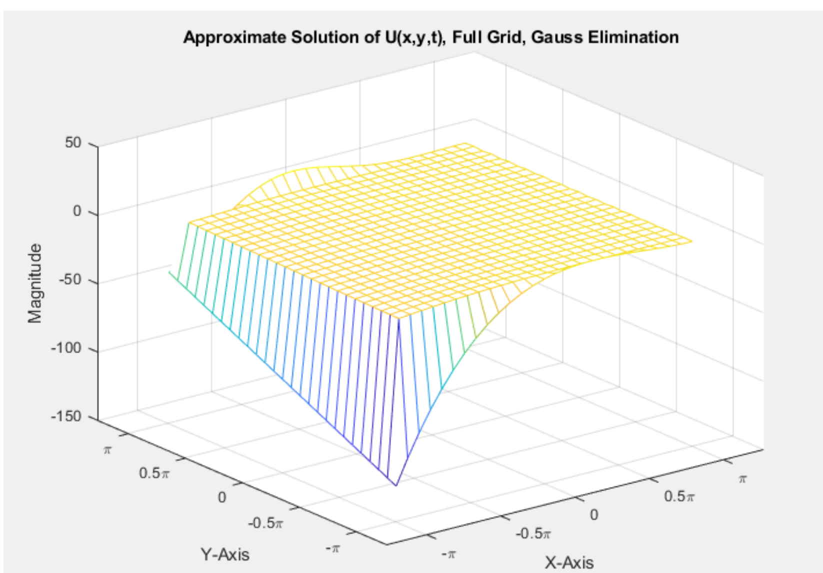


Figure 5