**Project A – Poisson Equation AP02-3**

**MECE 5397 – Computing For Engineers**

**Report By**:Ibrahim Ayyash

**PSID**: 1295286

**Due Date**: May 5, 2017

Name: Daniel Rodriguez

Lab Partner Name: Jon Marta

Course Number: MECE 3360

Experiment Date: 04/11/2015

Report Due Date: 04/24/2015

Date Submitted: 05/01/2015

Lab Session: Friday 9:00am-12:00pm

**Abstract**

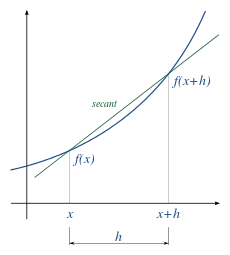
When engineers are tasked with coming up with solutions to large scale complex and seemingly insurmountable problems the first tool that is usually reach for is some sort of computing software or language. With the help of computers the new limiting factor for finding solutions becomes the power capability of the computer. In order to get as much use out of computing as possible certain simplifications are taken that produce a solution with an acceptable level of error as a result of the “shortcuts” taken. In this report solutions for a two-dimensional Poisson equations were computed using a finite difference method.

This report contain data pertaining to a computational solution carried out on Poisson’s two-dimensional equation with certain boundary and initial conditions. The report contains several sections: An introduction into finite difference methods that were used for problem simplifications that will give a more in depth description of the method. A problem description that will give a thorough explanation of the problem and how it was solved. The results that will display the computed solutions accompanied explanations of the data. And finally a conclusion discussing the results of the entire computing process.

**Finite Difference Introduction**

Certain mathematical equations can be used to model different mechanical, thermal, or chemical phenomena. True analytical solutions are extremely rare to find due to the shear difficulty of solving the problems due to them containing high order derivations. However very accurate approximations are able to be computed using finite difference approximations.

Finite difference methods gives one the ability to solve high order differential equations using a combinations of low level algebra rather than high level calculus. The main reason for the simplification is because the scale of problems in reality are quite large with non-idealized conditions. Large problems like these require computer simulations to solve, but computers are limited in their computing capabilities so they cannot preforms a large number of “expensive” calculations like integration or derivation. That’s where the finite difference approximations come in which can be performed several millions of times relatively fast.



*Figure 1: Graphical Derivative Approximation*



*Figure 2: First Order Derivative Approximation*

Figures 1 & 2 give a simple representation of finite difference methods and approximation. In the case of single variable, one-dimensional functions a derivative can be approximated by knowing two values and the distance between them. Higher accuracy of derivative approximation can be achieved with centered difference derivative approximations in cases where it’s necessary. This is a basic explanation of the most rudimentary of problems, but is crucial for understanding the system being solved for this report.

**System Description**

*PROBLEM SETUP*

The system being analyzed is a second order partial derivative in two space dimension Poisson equation which requires treading farther into finite difference methods. The introduction of partial derivatives complicates the analysis slightly, but is still essentially the same process.

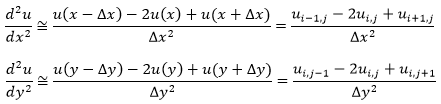
**

**

**

**

Discretizing the high order equation into a combination of linear equations is the first step in producing a solution to such problems. Using centered difference second order differential approximations yields a second order accuracy which is accurate enough for this this purpose.



Substituting in the discretized derivative approximations and solving out the equation in terms of the point of interest will produce a generalize linear equation. This is the basic equation that will dictate the values of the inner nodes.

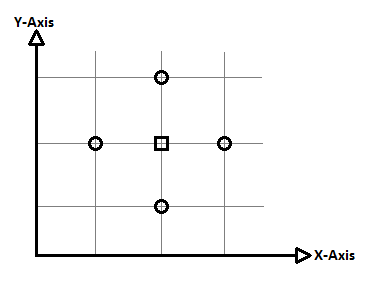
Exact analytical solutions aren’t practical for such function so an iterative solving method is used to get to a solution that with an acceptable error value. The two iterative methods that will be used to compute solution for the Poisson equation are Gauss-Sidel and Successive Over Relaxation algorithms.

*ALGORITHM EXPLANATION*

The Gauss-Sidel algorithm provides a stable convergence to a solution. The algorithm uses the previously computed values of adjacent nodes to provide a reasonable approximation to the current node. This method requires an initial prediction of adjacent values to start off iterative process that slowly corrects itself with each iterative calculation.

The Successive Over-Relaxation algorithm (SOR) also provides a stable convergence to a solution, but at a much faster rate. SOR uses the same method of predicting the value using adjacent value that also require an initial estimation. The reason SOR converges faster is because of an extra variable added onto the equation allows the algorithm to take larger “steps” with each iterative computation, resulting in less iterations.

A graphical representation of the solution shows how the algorithms work by taking adjacent values about the point of interest which is the center location with adjacent node values being the point to the top, bottom, left, and right sides. These algorithms are valid when using them to approximate internal nodes of a matrix, however they cannot be used on the boundaries of such problems. For those instances Dirichlet and Neumann conditions must be employed separately.



Xj-1

Xj

Xj+1

Yj+1

Yj

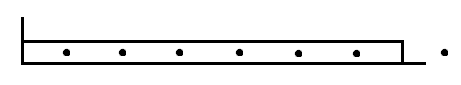
Yj-1

*Figure 3: Graphical Nodular Analysis*

*GHOST NODES*

Neumann conditions are boundary conditions that require their own discretized approximations to solve. The approximation method typically used is a centered difference discretization that utilizes a superficial “ghost node” after the boundary node. This method give a reasonable approximation with second order accuracy of the boundary value.

Solving for the ghost node value and inputting into the general equation at the given boundary eliminates the excess term that is generated after the boundary. This application of ghost nodes can be done for two-dimensional problems as well.

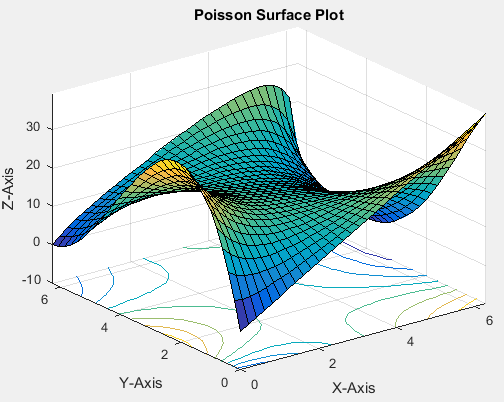


*Figure 4: Ghost Node Representation*

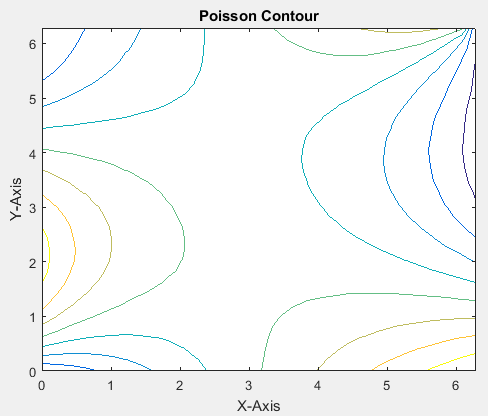
**Results**

*SOLUTION*

The solution of the Poisson equation is given in the form of a visual surface plot using a color gradient to show concentration as well as a contour plot. The solution behaves as expected with smooth consistent sloping values with no sudden spikes or discontinuities.



*Figure 5: Poisson Surface Solution*

**

*Figure 6: Poisson Contour Solution*

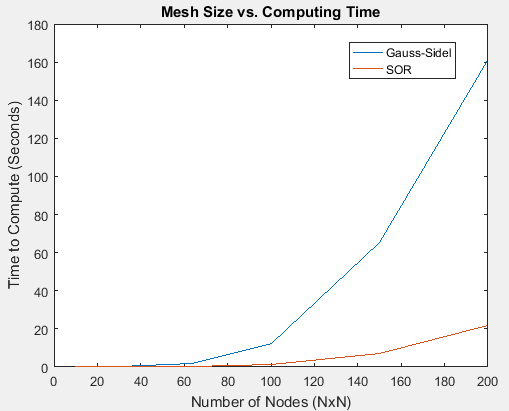
The two solver methods used for this particular problem, Gauss-Sidel and the Successive Over Relaxation, both have their respective benefits and drawbacks. A comparison of this two methods being used to solve Poisson’s equation are shown below

*Table 1: Gauss-Sidel Data Table*

|  |  |  |
| --- | --- | --- |
| **Gauss-Sidel Method (omega =1)** | | |
| *Node Number* | *Calculating Iterations* | *Time (seconds)* |
| 10x10 Mesh | 16200 | 0.011849 |
| 30x30 Mesh | 1098900 | 0.165067 |
| 64x64 Mesh | 20336640 | 1.891896 |
| 100x100 Mesh | 119190000 | 12.084551 |
| 150x150 Mesh | 624465000 | 65.33586 |
| 200x200 Mesh | 1724800000 | 160.688874 |

*Table 2: SOR Data Table*

|  |  |  |
| --- | --- | --- |
| **Successive Over Relaxation Method (omega=1.8)** | | |
| *Node Number* | *Calculating Iterations* | *Time (seconds)* |
| 10x10 Mesh | 5700 | 0.000758 |
| 30x30 Mesh | 108900 | 0.014126 |
| 64x64 Mesh | 2502656 | 0.286637 |
| 100x100 Mesh | 15290000 | 1.432763 |
| 150x150 Mesh | 80482500 | 7.251449 |
| 200x200 Mesh | 228920000 | 21.540489 |

****

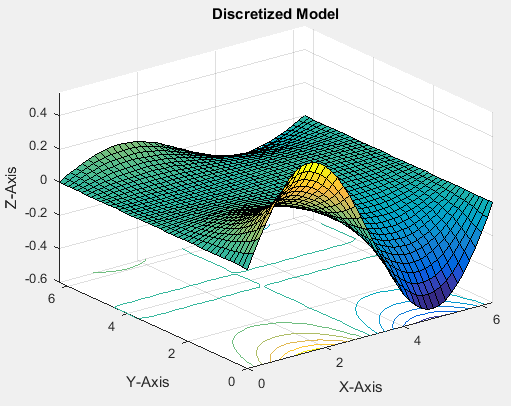
*Figure 7: Graphical Comparison of Solver Performance*

This data shows how much more beneficial using an SOR method for solving this system was. Since this computation required relatively large numbers of calculations SOR was a much better solver method.

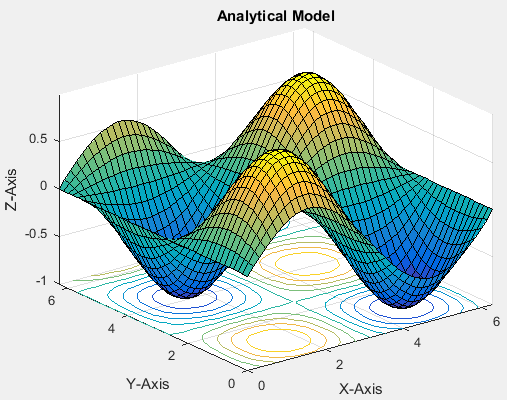
*VERIFICATION & VALIDATION*

In order to verify that this is indeed a solution the Poisson equation a verification and validation process must be taken. The validation method that was used for this report was the method of manufactured solutions.

This method of validation takes a function that is compatible with the Poisson, meaning that it is a two variable function containing both an “x” and “y” value. Plugging that equation into the Poisson equation and solving. This forces that function to be a solution which you can then discretize and solve using an SOR or Gauss-Sidel method which would then be compared to the true analytical function. If the errors are relatively small between the discretized and the analytical, the solving method used for Poisson is correct.



*Figure 8: Discrete Manufactured Solution*

**

*Figure 9: Analytical Manufactured Solution*

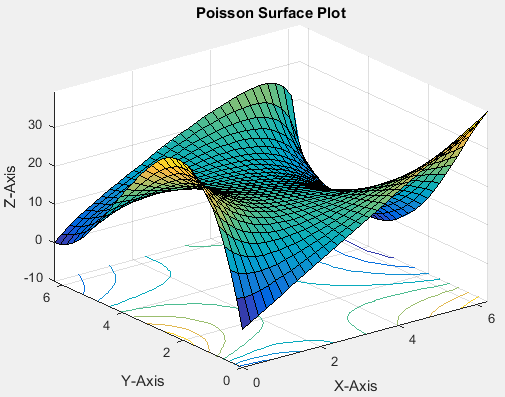
The function that was used to validate the solution was a sine, cosine elliptical function. The analytical and discrete model varied with their values suspiciously which led to the belief that possibly the discretizing solver code was incorrect, but after inspection and consultation with the instructor no error was found in the discretization. Therefore the issue may lie with the function chosen for the manufactured solution. The possible use of another function may validate the solution.



*Figure 10: Manufacture Solution Function*

*SPECIAL SCENERIO (F=0)*

This is a special circumstance in which the right hand side of the Poisson equation was set to equal zero. Based off the plot that was produced and comparing it to the original non-special scenario there was no difference.



*Figure 11: Poisson Solution (F=0 Scenario)*

The reasoning for this lack of any difference even though as substantial part of the Poisson equation was taken out can only be due to the iterative methods that were used. Gauss-Sidel and SOR tend to be self-correcting and will offset whatever incorrect values you input (to a certain degree) and slowly converge back to the solution.

There could be a different possibility as to why the values remained identical, however there is little else in the code except for those solvers.

**Summary and Conclusion**

This report covered the procedure taken in order to solve a high order partial multidimensional Poisson equation. This required an initial discretization to change represent the equation in terms of an algebraic system. That new simpler algebraic system could then be solve using an iterative method that would slowly converge to the correct solution within a small error percentage. The Iterative methods used were Gauss-Sidel and Successive Over Relaxation Method (SOR) where it was shown that for this particular equation using the SOR method allowed for faster solution convergence with less iterations.

The results of the entire computation were then displayed and analyzed where it was then proved to be the true solution by using a validation technique known as the method of manufactured solutions, wherein the known analytical solution of a function is compared to the discretized method to prove that the discretization model used was valid. The purpose of this project was to solve a complex Poisson equation using a simple discrete difference method and linear solve. This report proves this project was successful.