**MECE 5397- Scientific Computing**

**Project A – Helmholtz Equation AHc2-5**

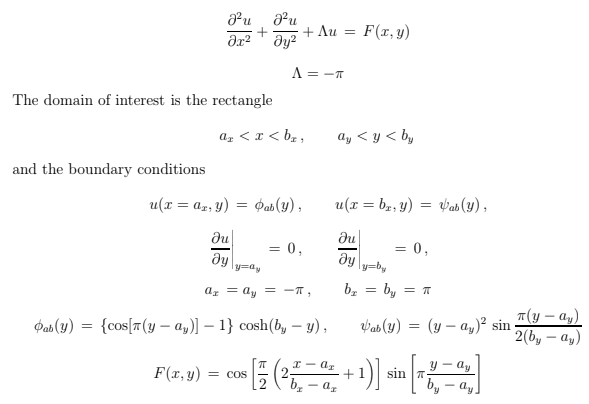
Rachel Murphy, 1351620

May 9, 2018

Abstract

The 2D Helmholtz equation is commonly used in partial differential equations regarding space and time. This equation is a time independent form of the wave equation. In the following document, two numerical methods are used to analyze the Helmholtz equation given Dirichlet and Neumann boundary conditions. The Gauss-Seidel and the Successive Over Relaxation methods are compared in terms of how quickly they converge to a final solution. The results of this study include run times for both the Gauss-Seidel and SOR methods, a grid convergence study, and the average error as the number of iterations increases.

Math Model



Equation Discretization



Assume:

Rearrange and simplify this equation.

For the Neumann Boundary condition, use the second order center difference approximation.

to get

Description of Numerical Methods:

**Gauss-Seidel Method:**

The Gauss-Seidel method is a solution process for a given set of n linear equations. If given a set of 3x3 equations in the form of [A]{X}={B}, initial guesses for x can be made and then substituted back in after every iteration until….

In this project, all boundary conditions were defined, and then all internal nodes were initially set to 0 as an initial guess. After each iteration, an error formula is generated to compare the new value with the previous value. Once enough iterations are conducted, the error should be very small and the system has converged. The pseudo code for this method is shown below.

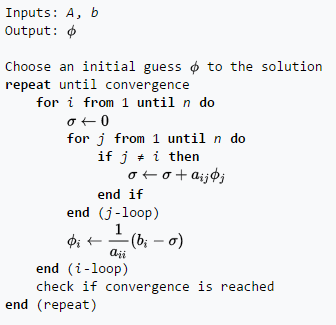


Figure 1: Gauss-Seidel Method Algorithm

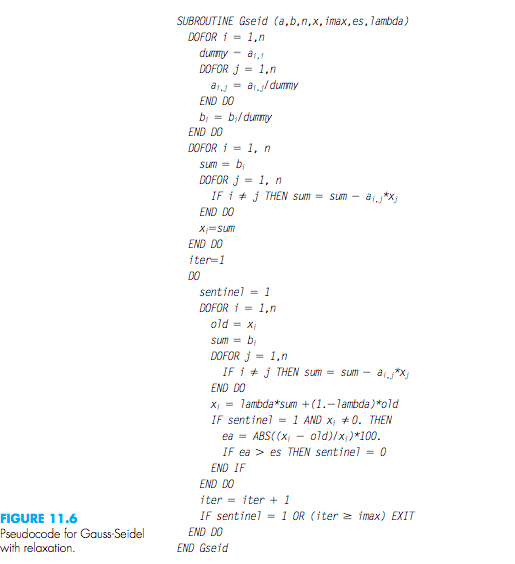
Successive Over-Relaxation Method

The SOR method is a modification of the Gauss Seidel method. This method uses the formula:



With lambda values between 1 and 2 in order to accelerate the convergence of the system.

The pseudo code for this method is shown below.



Technical Specifications of Computer Used

The computer used is an Intel ® Xeon ® CPU E5620 @ 2.40GHz with 1 core/CPU and a current CPU clock frequency of 2394 MHz (max CPU clock frequency of 2660 MHz). The machine has 64 memory channels, a DRAM total width of 32 bits, and a total DRAM per CPU of 16384 MB.

* RAM - 8 GB
* Hard Drive - 500 GB
* Graphics Card - any with DisplayPort/HDMI or DVI support - desktop only
* Monitor – Dell OptiPlex widescreen LCD with DisplayPort/HDMI or DVI support

Results:

**Gauss-Seidel**

Figure 1 is a surface plot of the Gauss-Seidel method with a mesh size of N=100.

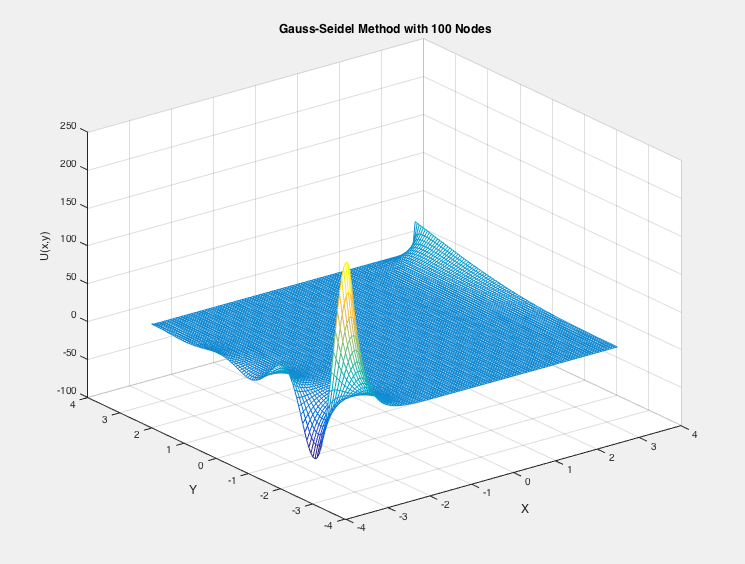


Figure 1-Surface Plot with Gauss-Seidel Method

Figure 2 is a contour plot of the Gauss-Seidel method using a mesh size of N=100.

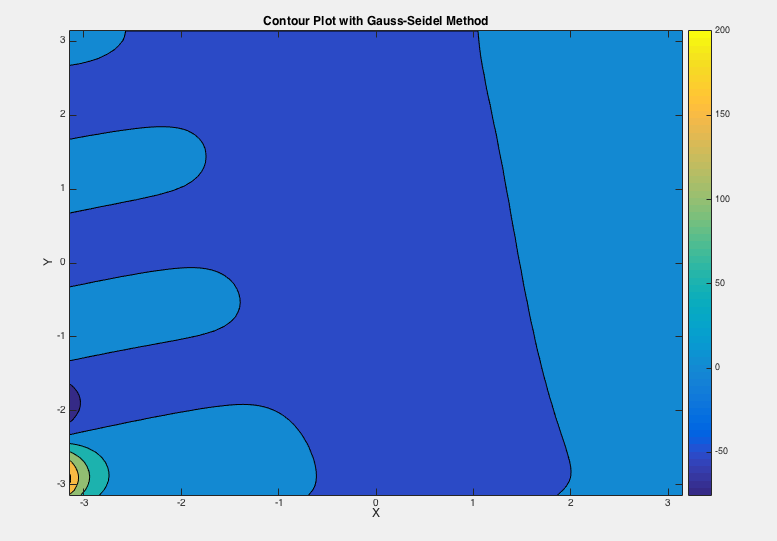


Figure 2-Contour Plot with Gauss-Seidel

Successive Over Relaxation

Figure 3 is a surface plot for the SOR method using a mesh size of N=100. This plot looks similar to the Gauss-Seidel surface plot.

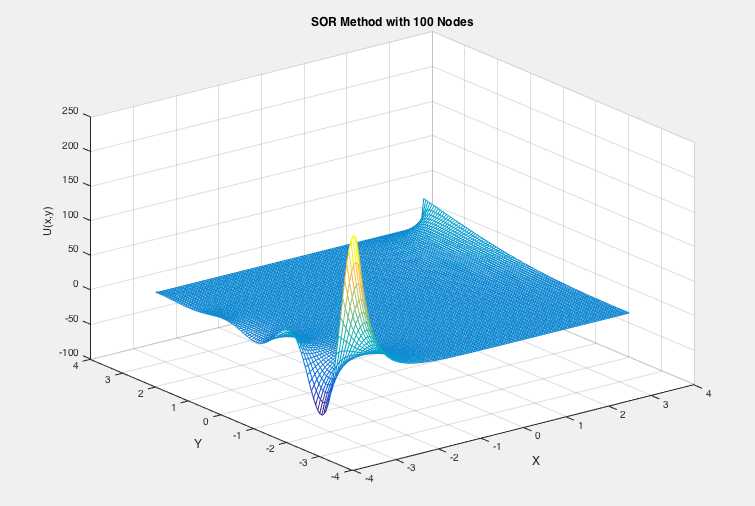


Figure 3-Surface Plot with SOR Method

Figure 4 is a contour plot of the SOR Method with a mesh size of N=100.

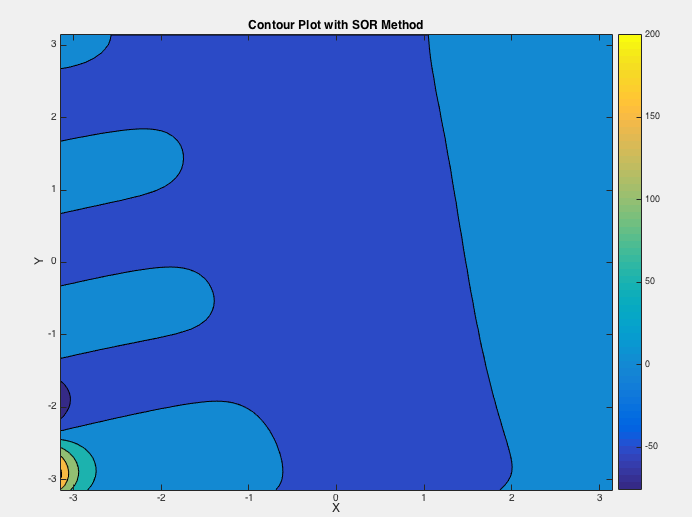


Figure 4-Contour Plot of the SOR Method

Table 1: Performance results for Gauss-Seidel Method and SOR Method

|  |  |  |  |
| --- | --- | --- | --- |
| Mesh size (N) | Number of Iterations | Run Time (Gauss-Seidel) | Run Time (SOR) |
| 10 | 20 | 0.002211 | 0.0002754 |
| 50 | 100 | 0.082787 | 0.04284174 |
| 100 | 1000 | 1.6899 | 1.6323 |
| 500 | 1000 | 44.9491 | 42.81007 |

After analyzing Table 1, one can conclude that the successive over relaxation method does run faster than the Gauss-Seidel Method. Both methods increase in runtimes as the mesh size (N) and the number of iterations (i) increases.

Table 2-Number of Iterations and the Average Error

|  |  |
| --- | --- |
| Number of Iterations (i) | Average Error (Gauss-Seidel) |
| 1 | 100 |
| 10 | 16.4965 |
| 100 | 3.2725 |
| 1000 | 0.0236 |
| 1500 | 8.8060e-04 |
| 2000 | 2.3308e-05 |

After analyzing Table 1, one can conclude that as the number of iterations increases, the average error will decrease. This analysis proves that the code used for this project is working correctly and the Gauss-Seidel is converging. The first iteration contains initial guesses for the solution. All initial guesses for the internal nodes are zero; therefore, the error will be 100%. A plot of the number of nodes (N) with respect to the error is shown in Figure 5.

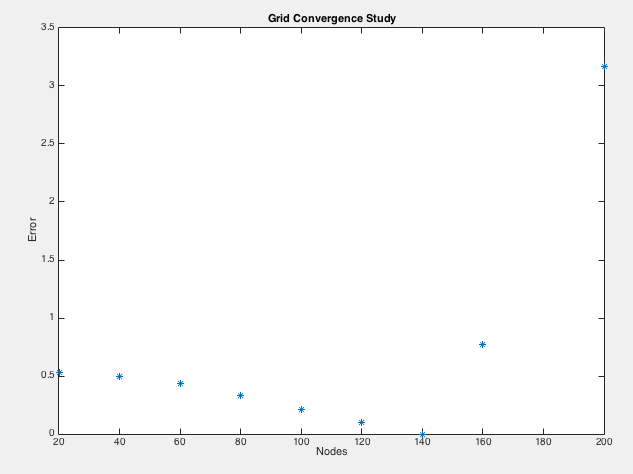


Table 3-Grid Convergence Study of the Gauss-Seidel Method

|  |  |  |
| --- | --- | --- |
| Gauss-Seidel Method | | |
| Number of Nodes (N) | Average Value of Solution | Difference to Previous Value |
| 10 | 171.86735 | 1 |
| 20 | 298.9179 | 0.52967982 |
| 40 | 508.1781 | 0.0388 |
| 60 | 712.7300 | 0.5484 |
| 80 | 908.8700 | 0.49526243 |
| 100 | 1.0907e+03 | 0.43377193 |
| 120 | 1.2562e+03 | 0.32916667 |
| 140 | 1.4063e+03 | 0.20979532 |
| 160 | 1.5431e+03 | 0.09722222 |
| 200 | 1.7850e+03 | 0 |
| 320 | 2.3549e+03 | 3.16593567 |

The grid convergence study shown in Table 1 is necessary in order to determine if additional nodes are required to obtain a more accurate solution. The number of nodes for this study begins at 10 and increments all the way up to 320 nodes and the average value of the solution was computed. The number of nodes needed for an accurate solution is around 200. If the mesh becomes any more fine, the error begins to increase.