**Dr. Andrea Prosperetti & Dr. Amit Amritkar**

**Project A - Helmholtz Equation**

**Solving the Helmholtz Equation Utilizing**

**Gauss-Seidel & Over-Relaxation in MATLAB**

**MECE 5397 – Scientific Computing**

**Maria Violeta Paez, 1174656**

**Friday May 5th, 2017**

Contents

[Abstract 3](#_Toc481675210)

[Mathematical Statement of Project 4](#_Toc481675211)

[Discretized Version of the Equations 5](#_Toc481675212)

[Numerical Method – Pseudo Code 6](#_Toc481675213)

[Gauss-Seidel Method 6](#_Toc481675214)

[Successive Over-Relaxation Method 8](#_Toc481675215)

[Technical Specification of the Computer Used 10](#_Toc481675216)

[Results 11](#_Toc481675217)

[Gauss-Seidel Method 11](#_Toc481675218)

[Successive Over-Relaxation Method 14](#_Toc481675219)

[Verification 16](#_Toc481675220)

[Result for F=0, 16](#_Toc481675221)

Figures

[Figure 1 Gauss-Seidel Method Algorithm - Wikipedia 7](#_Toc481679235)

[Figure 2 Successive Over-Relaxation Method Algorithm, Wikipedia 8](#_Toc481679236)

[Figure 3 Contour Plot of Helmholtz, Gauss-Seidel using N=100 10](#_Toc481679237)

[Figure 4 Surface Plot of Helmholtz, Gauss-Seidel using N=100, 11](#_Toc481679238)

[Figure 5 Contour Plot of Solution, SOR N=100 13](#_Toc481679239)

[Figure 6 Surface Plot of Solution, SOR N=100 14](#_Toc481679240)

[Figure 7 Mesh size 10 & Iteration 500 16](#_Toc481679241)

[Figure 8 Mesh size 10 & Iteration 500 16](#_Toc481679242)

[Figure 9 Mesh size 10 & Iteration 500 16](#_Toc481679243)

[Figure 10 Mesh size 10 & Iteration 500 16](#_Toc481679244)

[Figure 11 Laplace 17](#_Toc481679245)

[Figure 12 Helmholtz 17](#_Toc481679246)

[Figure 13 Laplace 17](#_Toc481679247)

[Figure 14 Helmholtz 17](#_Toc481679248)

Tables

[Table 1 Verification for Mesh size: 10 15](#_Toc481679249)

[Table 2 Verification for Mesh size: 100 15](#_Toc481679250)

[Table 3 Verification for Mesh size:200 15](#_Toc481679251)

# Abstract

This document is an official report that contains detailed information on the solution of the Helmholtz Equation, utilizing different numerical methods in the MATLAB for the project assigned in MECE 5397 Scientific Computing. The report contains the mathematical statement of the project, an explanation on the discretization version of the equation, the pseudocode, the specification on the computer used, and the results. The solution presented in the document, showcase the Gauss-Seidel method and compares it to the Successive Over-Relaxation method, utilizing graph and tables that illustrates the rate of conversation between the two methods. After analyzing the performance statistics and the grid convergence the Successive Over-Relaxation method is concluded to be the optimal method to solve the Helmholtz Equation.

# Mathematical Statement of Project

This reports will cover the different test conducted, utilizing numerical methods to solve the Helmholtz Equation. In the subject of mathematics, the Helmholtz equation is the partial differential equation that represents the time-independent wave equation. Below is Helmholtz Equation, where is the Laplacian, k is the wavenumber, and A is the amplitude.

The problem assigned asked to solve the Helmholtz equation on a rectangular surface with the following boundary conditions: 3 nonhomogeneous Dirichlet, 1 homogenous Neumann, and a function of F(x,y).

1. )2
2. 2
3. )-

|  |
| --- |
|  |
|  |
|  |

# Discretized Version of the Equations

Discretization is the transformation of continuous differential equation transforming them into discrete difference equation, which are fit for numerical computing. The partial derivatives given in the problem, are approximated by linear combinations of functions values at the grid point. Then, the second order center difference approximation is applied to both the x and y second derivatives at all points in the mesh.



Now, the equation can be replaced into the original wave equation to approximate the x and y second derivatives of us at the mesh (i,j). Center difference approximation shown below:

After rearranging the equation above, one can find the discretized Helmholtz Equation, solved for ui,j to be solved in MATLAB:

)

# Numerical Method – Pseudo Code

## **Gauss-Seidel Method**

The Gauss-Seidel method is a mathematical procedure that generates a sequence of improving approximate solution for problems like the one presented in this report. The pseudocode for the method is shown below:

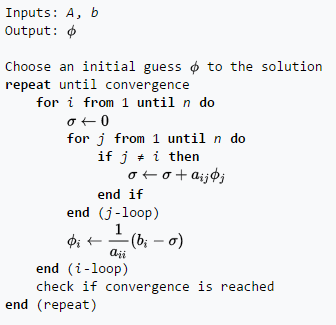


Figure 1 Gauss-Seidel Method Algorithm - Wikipedia

One must start with a guess for a solution to utilize this method. For the Helmholtz Equation project, assume all interior nodes are initially zero. Then, the linear system of equation, form in the zero matrix, can be solved using the error formula provided below:

This process continues until it the numerical method reaches convergence, which is found when the maximum error is less than the user-input tolerance, 1e-06.

The following discretized equation is the used in MATLAB:

## **Successive Over-Relaxation Method**

The Successive Over-Relaxation (SOR) method is another iterative method, similar to Gauss-Seidel method, however, it convergences faster. The pseudocode for the method is the following

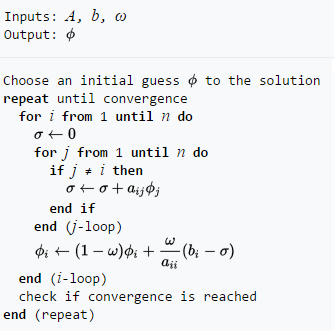


Figure 2 Successive Over-Relaxation Method Algorithm, Wikipedia

Like the Gauss-Seidel method the process starts with a guess, for the Helmholtz Equation project, assume all interior nodes are initially zero. Then, the linear system of equation can be solved using the initial guess. The process differs because the SOR method utilizes a coefficient B, which values bounded to 1<B<2. For the purpose of this project, B=1.5. Then, the error in the relation to previous value of the solution can be found utilizing equation below:

This process continues until it the numerical method reaches convergence, which is found when the maximum error is less than the user-input tolerance, 1e-06.

The following discretized equation is the used in MATLAB:

# Technical Specification of the Computer Used

The University of Houston provides computers for the engineering students at the Engineering Computing Center. The machine 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.

Processor: Intel® Core ™ i7-3770S CPU @ 3.10GHz

Installed memory RAM: 8.00 GB (7.88 GB Usable)

System type: 64- bit Operating System

# Results

## **Gauss-Seidel Method**

|  |  |
| --- | --- |
| Figure 3 Contour Plot of Helmholtz, Gauss-Seidel using N=100 |  |

The contour plot for the numerical solution to the Helmholtz Equation is shown in Figure 3, this imagine can be obtained using the Gauss-Seidel method. The mesh used to obtain the plot was of N=100, by observing the boundaries there is a correct correlation to the boundary conditions.

|  |
| --- |
| Figure 4 Surface Plot of Helmholtz, Gauss-Seidel using N=100, |

Now, Figure 4 shows the surface plot for the numerical solution to the Helmholtz Equation utilizing Gauss Seidel method with a mesh of N=100. Comparing Figure 3 with Figure 4, one can see the plots match.

Table 1 showcase the performance based on the number of iterations used in the code, as well as the running time to reach convergence.

|  |  |  |
| --- | --- | --- |
| **Mesh Size (N)** | **Number of Iterations**  **Tolerance=1e-6** | **Time (Sec)** |
| 10 | 94 | 0.0679 |
| 20 | 383 | 0..2542 |
| 40 | 1595 | 1.4546 |
| 80 | 7639 | 15.2433 |
| 160 | 26146 | 143.3888 |
| 320 | Excessive time consumed | |
| 640 |
| 1280 |

The number of iterations and the time for the code to run increases as the mesh size increases, therefore there is a distinct correlation between the size of the mesh and the performance of the numerical methods in terms of convergence speed. The numerical method does convergence to a unique solution, this validates the written code in MATLAB provided.

## **Successive Over-Relaxation Method**

|  |
| --- |
| Figure 5 Contour Plot of Solution, SOR N=100 |

The contour plot for the numerical solution to the Helmholtz Equation is shown in Figure 5, this imagine can be obtained using the SOR method. The mesh used to obtain the plot was of N=100, by observing the boundaries there is a correct correlation to the boundary conditions.

|  |
| --- |
| Figure 6 Surface Plot of Solution, SOR N=100 |

Now, Figure 6 shows the surface plot for the numerical solution to the Helmholtz Equation utilizing SOR method with a mesh of N=100. Comparing Figure 5 with Figure 6, one can see the plots match.

|  |  |  |
| --- | --- | --- |
| **Mesh Size (N)** | **Number of Iterations**  **Tolerance=1e-6** | **Time (Sec)** |
| 10 | 35 | 0.0486 |
| 20 | 155 | 0.1094 |
| 40 | 567 | 0.5023 |
| 80 | 2590 | 4.9234 |
| 160 | 9404 | 53.4259 |

## **Corrected Boundary Condition**

|  |  |
| --- | --- |
| Figure Helmholtz Equation SOR, Neumann boundary condition was inaccurate |  |

Figure 7 shows the Neumann boundary conditions used throughout the project. However, after further analysis and correction of the code, Figure 8 shows the correct boundary condition behavior.

## **Verification**

A comparison of error between result must be done to verify the approximation made in the project are accurate. This can be done with fixed iteration method, which shows that as specific size of mesh increases, the accuracy in results will increase accordingly. SOR method was utilized in this section, since it was proved that this method converges faster.

Table 1 Verification for Mesh size: 10

|  |  |  |
| --- | --- | --- |
| **Mesh Size: 10** | | |
| **Iterations** | **Elapsed Time** | **Error of u(i) and u(i-1)** |
| 5 | 0.0099 | 1.5440 |
| 50 | 0.0396 | 2.1334e-11 |
| 500 | 0.4358 | 6.9689e-15 |
| 5000 | 4.4842 | 6.9689e-15 |

Table 2 Verification for Mesh size: 100

|  |  |  |
| --- | --- | --- |
| **Mesh Size: 100** | | |
| **Iterations** | **Elapsed Time** | **Error of u(i) and u(i-1)** |
| 5 | 0.3964 | 1.1413e+03 |
| 50 | 4.3351 | 1.1956e+03 |
| 500 | 43.4408 | 1.7004 |
| 5000 | 451.4009 | 1.5663e-09 |

Table 3 Verification for Mesh size:200

|  |  |  |
| --- | --- | --- |
| **Mesh Size: 200** | | |
| **Iterations** | **Elapsed Time** | **Error of u(i) and u(i-1)** |
| 5 | 1.6038 | 405.3127 |
| 10 | 3.3889 | 1.6949e+03 |
| 100 | 36.7292 | 4.7203e+03 |
| 1000 | 601.9287 | 21.0621 |

As the iteration number increases, the error between the u(i) and the u(i-1) becomes smaller. Whenever using a small mesh, with large iteration, may seem like the error is small, this does not accurate represent the equation as having a larger mesh.

|  |  |
| --- | --- |
| Figure 8 Mesh size 10 & Iteration 500 | Figure 9 Mesh size 10 & Iteration 500 |

|  |  |
| --- | --- |
| Figure 10 Mesh size 10 & Iteration 500 | Figure 11 Mesh size 10 & Iteration 500 |

# Result for F=0,

After carrying out all simulations, now one must be show one last simulation where F=0 and and include the results in the report. Changing the forcing function, alters the Helmholtz Equation into Laplace’s equation. Below there is Figure 11-15 that compares Meshes of 100 between Helmholtz Equation and Laplace’s equation using SOR, for faster convergence.

|  |  |
| --- | --- |
| Figure 12 Laplace | Figure 13 Helmholtz |

|  |  |
| --- | --- |
| Figure 14 Laplace | Figure 15 Helmholtz |

The only changes between Laplace and the Helmholtz is that the Laplace equation convergence faster. The graph above show that they look very similar for N=50.