# Comparative Study of Solvers for System of Linear Equations
# ME6151 Jan-May 2022

Anand Zambare - AM21S004[1] and Shristi Singh - ME21D025[2]

[1]Department of Applied Mechanics, Indian Institute of Technology, Madras, India, 600036.
[2]Department of Mechanical Engineering, Indian Institute of Technology, Madras, India, 600036.

April 28, 2022

## Abstract

The comparison study for different solvers for solving the system of linear equations is done. The system is an output of discritization of governing partial differential equations through different methods like FDM or FVM. Elimination methods, Gradient search based methods and Multigrid methods are tested on 1-D diffusion problem with Dirichlet boundary conditions. No.of iterations required and the path to the solution are basis of comparison. It was found that Geometric Multigrid Method (GMG) performs the best in terms of no.of iterations required for converged solution. GMG takes less than $(1/10)^{th}$ the no.of iterations required for Gauss-Seidel method(GSM) which performs much better than other methods.

**Keywords**: Discretization, 1-D diffusion, Geometric Multigrid Method(GMG), Gauss-Seidel Method(GSM).

## Introduction

Finding an optimum method to solve a system of simultaneous equations with n unknowns is one of the difficulties in computational fluid dynamics, mainly when the value of n is large. Of course, not all problems have the best method, because the goodness of a method depends to some extent on the specific physics of the problem to be captured. The solving method must have the mentioned features:
(1) The method or technique should occupy the as minimum space as possible and it should be as simple as possible.
(2) A generally stable procedure is adopted while rounding off errors. In case of any needful situation, there must be an availability subroutine to assure stability. The use of previous results in the same routine to estimate a new one is done to minimize the round off errors.
(3) In general after every iteration the result should come out to be better than the previous iteration.
There are three different classes of methods that are under the study in this project:

1. Elimination methods
   - Jacobi method
   - Gauss-Seidel method
2. Gradient methods
   - Steepest descent method
   - Conjugate gradient method
3. Multi Grid methods
   - Geometric multi grid method

## Problem description

The problem is considered from assignment 1 i.e. 1-D diffusion equation. The problem statement is as follows: Consider the one-dimensional diffusion equation. Assume that there is a source term S = 8x, and that the diffusion coefficient  = 1 is constant. Solve for the steady state of the variable  where the boundary conditions are A = 100 and B = 600 at the end A and B (See Figure 1). The above mentioned problem is discretized using the finite volume method. The discretized equations are solved using:
1)Jacobi method (JM)
2)Guass-seidel method (GSM)
3) Steepest descent method(SDM)
4) Conjugate gradient method (CGM)
5) Geometric multigrid method (GMG).

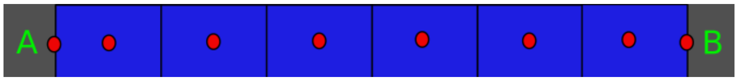

Figure 1: 1-D Diffusion problem with Dirichlet boundary conditions.

## Methods to solve system of linear equations

Following methods are considered for the solution of discretized equations of 1-D diffusion problems.

- JACOBI METHOD

Jacobi method comes under the elimination methods. Jacobi method is an iterative method for determining the solutions of a system of linear equations. The algorithm starts with guess values and other values are calculated using this guess. The process is then iterated until it converges. As the number of grid points increases, the Jacobi method is slow to converge.let's look at the algorithm.
Step 1: The system of linear equations will have matrix A and b where A is the coefficient matrix and b is the rhs vector which is known vector. A has size $n \times n$ and b has size $n \times 1$.
Step 2: Now express the matrix in $A \cdot X = b$
where $X = x_1, x_2, .......x_n$ which is the unknown vector we are solving for.
Step 3: Substituting an initial guess values of $X$ and solve for the $x_1, x_2....$ using the following equations

$$x_1{}^{k+1} = \frac{A_{12}\cdot x_2{}^k + A_{13}\cdot x_3{}^k + ........A_{1N}\cdot x_N{}^k}{A_{11}}$$

$$\begin{array}{c} . \\ . \\ . \\ . \end{array}$$

$$x_N{}^{k+1} = \frac{A_{11}\cdot x_1{}^k + A_{12}\cdot x_2{}^k + ........A_{1N}\cdot x_{N-1}{}^k}{A_{NN}}$$

Step 4: Check for the residue. If residue is less than the tolerance then solution is converged but if greater than tolerance go to step 3 again.

- Gauss Seidel Method

Gauss-Seidel method comes under the elimination methods. The Gauss-Seidel method is an iterative method used to solve a system of linear equations. The algorithm starts with guess values and other values are calculated using this guess. The process is then iterated until it converges. As the number of grid points increases, the Gauss-Seidel method is also slow to converge. In the Gauss-Seidel method and Jacobi method the difference is latest value calculated would be used. For example in solving the equation $x_2$ the $x_1$ from latest calculations is used.
Step 1: The system of linear equations will have matrix A and b where A is the coefficient matrix and b is the rhs vector which is known vector. A has size $n \times n$ and b has size $n \times 1$.
Step 2: Now express the matrix in $A \cdot X = b$
where $X = x_1, x_2, .......x_n$ which is the unknown vector we are solving for.
Step 3: Substituting an initial guess values of $X$ and solve for the $x_1, x_2....$ using the following equations

$$x_1{}^{k+1} = \frac{A_{12}\cdot x_2{}^k + A_{13}\cdot x_3{}^k + ........A_{1N}\cdot x_N{}^k}{A_{11}}$$

$$\begin{array}{c} . \\ . \\ . \\ . \end{array}$$

$$x_N{}^{k+1} = \frac{A_{11}\cdot x_1{}^{k+1} + A_{12}\cdot x_2{}^{k+1} + ........A_{1N}\cdot x_{N-1}{}^{k+1}}{A_{NN}}$$

Step4: Check for the residue. If residue is less than the tolerance then solution is converged but if greater than tolerance go to step 3 again.

- Steepest Descent Method.

Steepest descent is a method from a different class. It comes under the title of gradient search based methods. The methods were initially developed for the optimization problems. This method is effective for low dimensional problems, however for a large dimensional system, it converges slowly. The gradient method is better than the elimination method because gradient is the rate of change of the function at that point, using a gradient helps to reduce the number of iterations needed and get faster convergence. The big idea is to solve for the optimization/minimization of the problem $A \times X = b$ where it can be minimized for $r = b - A \times X$. Let's go through the algorithm in brief.
Step 1: The system of linear equations will have matrix A and b where A is the coefficient matrix and b is the rhs vector which is known vector. A has size $n \times n$ and b has size $n \times 1$.
Step 2: Calculate the residue for the guess value of X which are say $X_0$. Let the residue values are $r_0$. This is nothing but the direction of gradient for guess values. The equation should move into this direction. By what factor? This question is to be asked. The factor can be calculated so that the movement is maximum.
Step 3: Calculate the factor $\alpha_0$ which gives the magnitude of movement.
Step 4: Calculate X1 which is next iteration value of X using $X_1 = X_0 + \alpha_0 r_0$. Check for the residue. Continue in loop from step 2 to step 4 till convergence is achieved.

- Conjugate gradient Method.

The important idea is to use the gradient search and get faster convergence. The no.of steps involved in the steepest descent method is an issue with it. Here in the Conjugate gradient method the no.of calculations reduced and the gradient known is used in better way to get faster convergence. The first iteration in this scheme is same as that of the Steepest descent method. After the first step following steps are to be followed.
Step 1*: Assume $P_0 = r_0$ where $r_0$ comes from the previous steps. Calculate $r_1 = b - A \times x_1$.
Step 2*: Calculate $\beta_1$ where $beta_1 = \frac{P_0{}^T \cdot A \cdot r_1}{P_0{}^T \cdot A \cdot P_0}$
Step 3*: Calculate the $P_1$ as $P_1 = r_1 - (\beta_1 \cdot P_0)$
Step 4*: Calculate the $\alpha_1$ as $\alpha_1 = \frac{P_1{}^T \cdot r_1}{P_1{}^T \cdot A \cdot P_1}$
Step 5*: Calculate the residue and converge till tolerance with repetition from step 2*.

- Geometric Multigrid Method.

Geometric multi-grid method is a method from different class. It comes under the umbrella of Multigrid methods. This method is quite easy to implement for structured meshes. The main idea in this method is, the high frequency errors drop down faster in the case of Gauss-Seidel method but it takes very long time for low frequency errors to drop down this results into the slow convergence rate of the Gauss-Seidel method for large (see Figure 2.) $N \times N$ systems.
Now to overcome this, the solution strategy is set in such a way that the low frequency errors for the grids of our interest are transformed on the coarser grids and hence they become the high frequency errors. Here i.e. on coarser grid the error equation is solved to find the error on the coarser grid. This error is again transformed back on the finer grid i.e. the grid

with which the problem is getting solved. The transformed error is used as a correction on for the solution on that level. Let's look at the algorithm in short.

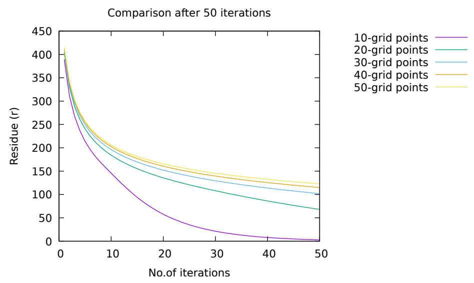

Figure 2: Comparison of residue after 50 iterations for different grid size.

Consider l = 0 is the fine mesh where the solution of unknown should be obtained. Consider l = 1 is the coarser mesh where the low frequency errors of the l = 0 become high frequency errors. For the geometric Multigrid method, there can be various levels but here let's look at two level.
Step 1: Use GSM for say 50 iterations on the fine grid. Calculate ${X_0}^k$ where k = 50 iterations of the GSM.
Step 2: Calculate the residue at the same level for k = 50. Say ${r_0}^k$. (In both steps 0 suffix denote the fineness level of the mesh.)
Step 3: Transfer ${r_0}^k$ to ${r_1}^k$ where it is the residue at level 1 which is coarser mesh.
Step 4: Solve for the error equation using the ${r_1}^k$ where error equation is given as follows:
$A \cdot e = r$ The solution of this equation can be calculated using GSM (use only 5-10 iterations) equation will be solved for 'e'. Here ${e_1}^k$ is calculated.
Step 5: Error ${e_1}^k$ should be transferred to ${e_0}^k$ so that it can be used as a correction in the ${X_0}^k$. to get ${X_0}^{k+1}$
Step 6: Calculate the residue value using the ${X_0}^k$ and ${X_0}^{k+1}$ and continue this cycle till the convergence is achieved.

# Results and Discussion

The problem proposed here is solved using the 50 grid cells, Finite Volume Method for discretization and given boundary conditions with all the above mentioned techniques for tolerance value of $1.0e^{-}6$. In case of Multigrid for this problem only two grids are used. 50 grid points is the finest level and 10 grid points is the coarser mesh. The table 1 reports the no.of iterations required for the converged solution in each method.

| Method | No.of iterations |
|---|---|
| JM | 9136 |
| GSM | 4749 |
| SDM | 9034 |
| CGM | 6274 |
| GMG | 127 |

The required no.of iterations are clear indicators for the best method. The Geometric Multi Grid method needs least no.of iterations i.e. 127 and it turns out to be the best method. In case of the elimination method, the Jacobi method takes 9136 iterations and Gauss-Seidel method takes 4749 iterations. So in the class of elimination methods Gauss-Seidel turns out to be the better method. The rate of convergence is calculated for both as follows:

$$\gamma = \frac{log(r_{n+1}/r_n)}{log(r_n/r_{n-1})}$$

where $\gamma$ is the estimation of the convergence rate.
This is found in the case of the elimination methods i.e. the Jacobi method and Gauss Seidel method. For Jacobi method the convergence rate is 0.85 and 1.55 for Guass-Seidel method. Figures 4 to 8 depict the paths to the solution for different no.of grid points from 10 to 50 using all methods discussed in this project. From all the figures again it is very clear for the first four methods that only high frequency errors drop quickly but all low frequency errors are going to drop slowly and hence the no.of iterations required for it's convergence is more.
So as discussed previously this becomes an important factor for the geometric Multigrid method. In this case only two grids are used but it is possible to work with more than two grids and even further reduce the no.of iterations required for the convergence.
Figure 8 represents the residue drop in case of Geometric Multigrid method with respect to no.of iterations. In general the residue drop may look little surprising as the residue after few iterations has jumped again. But the point where the actual multi grid starts is the point where the sudden jump in residue is seen (After 51 iterations of GSM).
For implementing the Geometric Multigrid method the coarser mesh taken is 10 cells. So there are 5 fine cells clubbed together to form the one coarser cells (see Figure 3. ). The three major steps are a)Coarse level creation, b) Restriction and c) Prolongation. In case of structured grids the coarse level creation is very easy step, just clubbing of cells (at fine level) together forms the coarser grid.
The error equation which is to be solved at the coarse level has the form $A \cdot e = r$ where A is the coefficient matrix at coarser level. It is easy to get it in case of the structured meshes for any no.of grid points. For any 1-D diffusion problem with h as $\Delta x$ it will look like as follows:

$$\begin{vmatrix} 3/h & -1/h & 0 & 0 & ---- \\ -1/h & 2/h & -1/h & 0 & ---- \\ | & | & | & | & ---- \\ ---- & 0 & 0 & -1/h & 3/h \end{vmatrix}$$

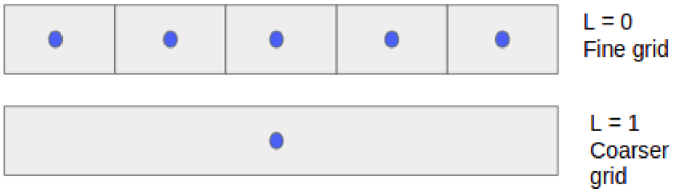

Figure 3: Various levels in Multigrid method

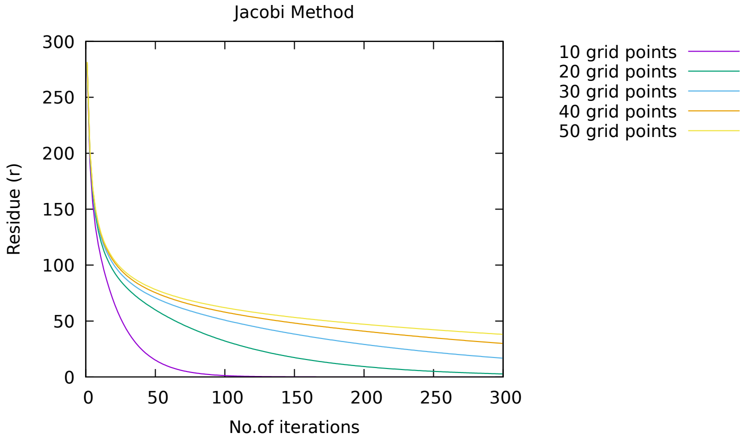

Figure 4: Convergence for the Jacobi Method (JM) with all grid sizes.

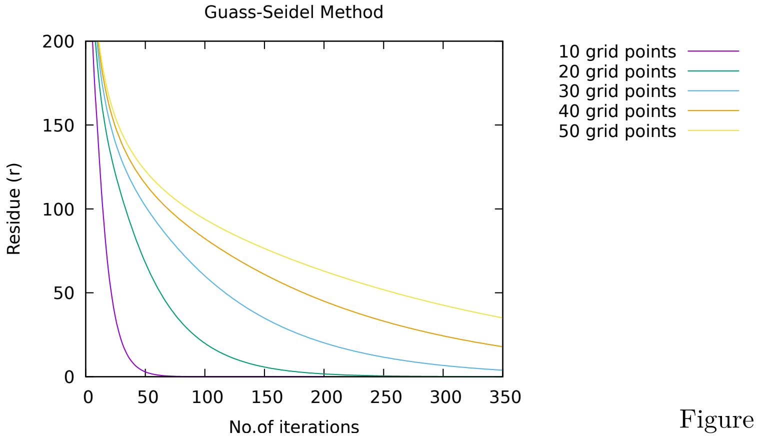

Figure 5: Convergence for the Gauss-Seidel Method (GSM) with all grid sizes.

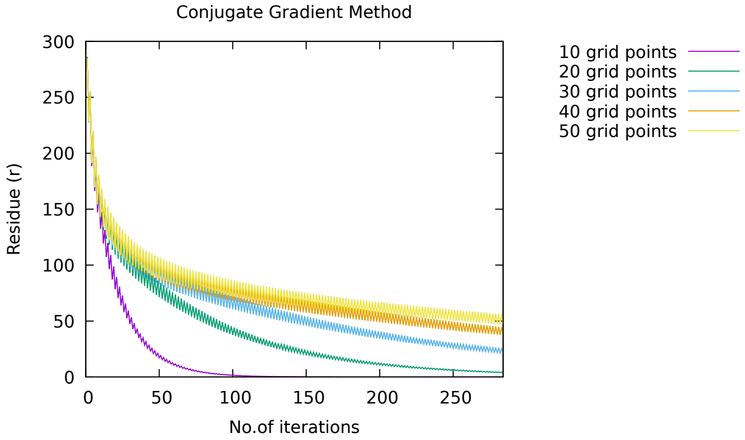

Figure 6: Convergence for the Steepest Descent Method (SDM) with all grid sizes.

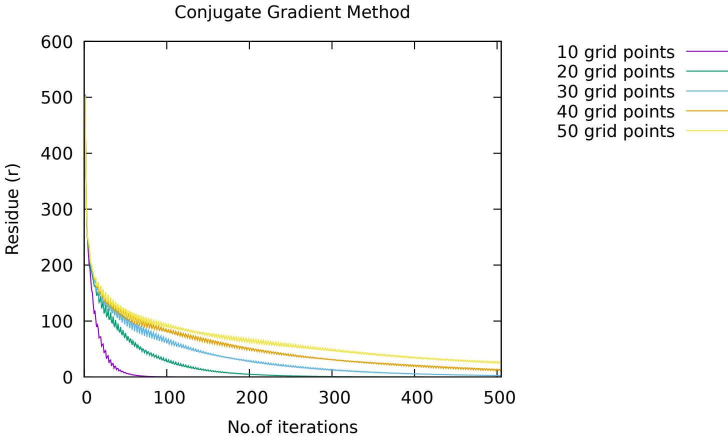

Figure 7: Convergence for the Conjugate Gradient Method (CGM) with all grid sizes.

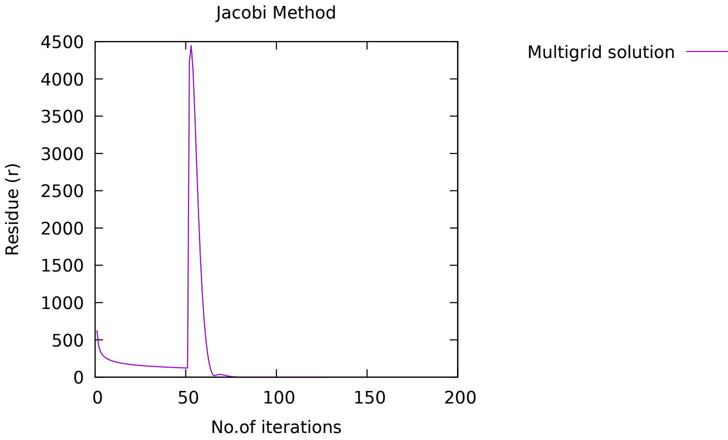

Figure 8: Multigrid cycles and convergence to the tolerance value.

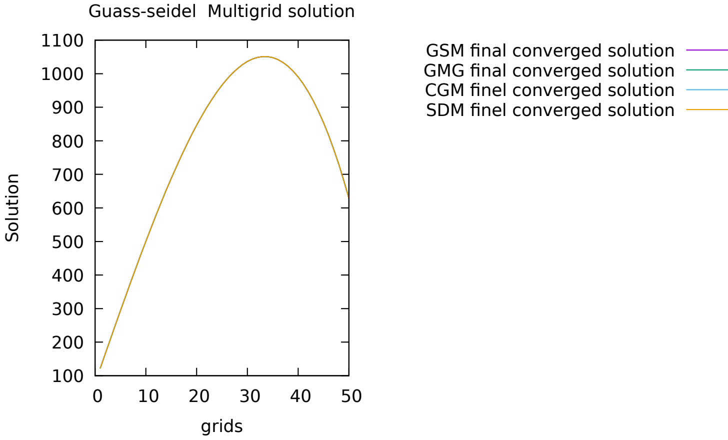

Figure 9: Final converged solution for all methods.

The restriction operation is nothing but finding out the residue at fine level and transferring it to the coarse level, for this problem very simple restrictor is used called as ”Addition/-Summation”. The last step is prolongation. In this step the error equation is solved on the coarser grid and error is transferred back to the fine grid as a correction to the unknowns for which the equations are being solved.
Figure 9 is the final converged solution for all the methods. The figure depicts that it is same for all the methods. The path to the solution is different in case of different methods but the final solution is same.
In case of elimination methods the path to solution is through the guess values. In case of the gradient methods the path to the solution is through the gradient of f which is constructed as follows:

$$\nabla f = b - (A \cdot X)$$

Gradient search based methods are solving for the minimum of gradient of f. In case of Multigrid the path is completely different as it uses the GSM for some iterations and then it is using error equations to get corrections.

# Conclusions

- Elimination methods are very slow for convergence with ’N’ unknowns (specially when N is very large) and their behaviour depends on the frequency of the error.
- Higher the frequency of error quickly it will be dropped down, lower the frequency of the error slowly it will go down (converge).
- In class of elimination method, the Gauss-Seidel method performs better than the Jacobi method. In fact it takes almost half the iterations than the Jacobi i.e. because recently calculated values are taken into account quickly (in the same iteration).
- In class of gradient search based method, the Conjugate Gradient method performs better than the steepest descent method. The no.of calculation are more in CGM but the no.of iterations are almost 1.5 times less than the SDM.
- The fastest method which gives converged solution in the least no.of iterations is Geometric Multigrid Method. It takes less than $\frac{1}{10}$ no.of iterations of GSM. Though it is implemented only with two grids. With more no.of grids involved it will take lesser no.of cycles/iterations to converge.
  Note: Here with the two grids the v-cycle is performed.
- In case of structured grids Geometric Multigrid method works efficiently than any other class of method. In case of unstructured grids the Algebraic Multigrid method is to be implemented for faster convergence which is beyond this study.

# Abbreviations

FDM : Finite Difference Method
FVM : Finite Volume Method
JM: Jacobi Method
GSM: Gauss-Seidel Method
SDM: Steepest-Descent Method
CGM: Conjugate Gradient Method
GMG: Geometric Multigrid Method
A: Coefficient Matrix
b: known vector in $A \cdot X = b$ system.
X: unknown scalar for which equations are solved.
k: iteration no.
$\alpha$: Magnitude of movement.
$\beta$: Magnitude of movement.
$\gamma$: Rate of convergence.
h: Δx which is spacing in the cells.
L: level of the refinement.
f: function.

# References

1) Magnus R Hestenes et.al,”Method of Conjugate Gradients for Solving Linear Systems”, journal of Research of the National Bureau of Standards, vol 49, 1952.
2) Numerical Recipes: The Art of Scientific Computing, $3^{rd}$ edition.
3)An Introduction to Computational Fluid Dynamics: The Finite Volume Method, second edition.