International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 9 (2019) pp. 2234-2237

Illustration of Gauss – Seidel Method Using Matlab

Riyasdeen S[1], Abbas S[2], Lenin T[3]

[1] *Assistant Professor, P.G. and Research Department of Mathematics, Marudupandiyar College, Pillayarpatti, Thanjavur, Tamilnadu, India.*

[2] *Assistant Professor, P.G. and Research Department of Mathematics, Khadir Mohideen College, Adirampattinam, Thanjavur, Tamilnadu, India.*

[3] *Associate Professor, P.G. and Research Department of Mathematics, Khadir Mohideen College, Adirampattinam, Thanjavur, Tamilnadu, India.*

Abstract

In numerical linear algebra, a solution to a linear system is an assignment of values to the variables such that all the equations in the systems are simultaneously satisfied. One of an iterative method used to solve a linear system of equations is the Gauss–Seidel method which is also known as the Liebmann method or the method of successive displacement. It is named after the German mathematicians Carl Friedrich Gauss and Philipp Ludwig von Seidel, and is more or less similar to the Jacobi method. Further this paper gives the MATLAB code to solve the linear system of equations numerically using Gauss–Seidel method.

Keywords: *System of linear equations, Gauss-Seidel Method, MATLAB solutions*

INTRODUCTION

MATLAB

MATLAB is a very powerful software package that has many built-in tools for solving problems and for graphical illustrations (1). The name MATLAB stands for MATrix LABoratory (2). The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics. The package MATLAB provides an environment in which we can learn to programme and explore the structure of the numerical methods.

Gauss–Seidel method

The Gauss-Seidel Method allows the user to control round-off error. Most of the elimination methods are liable to suffer from round-off error. This method is modification of the Jacobi's iteration method. It is defined on matrices with non-zero diagonals, but convergence is only guaranteed if the matrix is either diagonally dominant or symmetric positive definite (3). Also: If the physics of the problem are understood, a close initial guess can be made, decreasing the number of iterations needed.

Basic Procedure:

- ✓ Algebraically solve each linear equation for the unknowns, say x_i
- ✓ Assume an initial guess solution array for the given system of linear equations
- ✓ Solve for each unknowns x_i and repeat the steps
- ✓ Use absolute relative approximate error after each iteration to check if error is within a pre-specified tolerance.

Algorithm:

By this approach suppose we have an initial guess solution array, say $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$, for the solution array x of the given system of linear equation $Ax = B$,

$$\sum_{i=1}^{n} a_{ij}\, x_i = b_j, \text{ for } j = 1, \dots, n$$

and we generate an improved solution estimate $x^{(k+1)}$, from the previous solution estimate $x^{(k)}$. To generate an improved solution estimate $x^{(k+1)}$, the Gauss-Seidel method differs from the Jacobi method in the fact that at the $(k+1)$-th step the available values of $x_j^{(k+1)}$ are being used to update the solution,

$$x_j^{(k+1)} = \frac{1}{a_{jj}}\left[b_j - \sum_{\substack{i=0 \\ i \neq j}}^{n} a_{ij}\, x_i^{(k)}\right],$$

$$\text{for } j = 1, \dots, n \quad (4)$$

Equation 1. Iteration formula of Gauss–Seidel method

At the end of each iteration, one calculates the absolute relative approximate error for each x_i as

$$|\varepsilon_a|_i = \left|\frac{x_i^{\text{new}} - x_i^{\text{old}}}{x_i^{\text{new}}}\right| x100 \quad (5)$$

Equation 2. Error for Gauss-Seidel Method

MATLAB Algorithm

- ✓ Go to file → new → M-file.
- ✓ Input the coefficient matrices *A* and *B*.
- ✓ Input the initial guess estimate for the solution of the given system of linear equations and the number of iterations to be performed.
- ✓ If diagonal elements of the matrix *A* are zero display the output enter the valid matrix to perform Gauss–Seidel method.
- ✓ Check the matrix for whether the matrix *A* is diagonally dominant and positive definite or not, since the Gauss – Seidel iteration method assures that the numerical solution for the linear system converges to the original solution for any initial starting vector if the matrix is strictly diagonally dominant and positive definite.
- ✓ Then perform Equation 1. Iteration formula of Gauss–Seidel method to get the improved solution estimate.
- ✓ Then perform Equation 2. Error for Gauss-Seidel Method to get the absolute relative approximate error at the given iteration.
- ✓ Save and run the file.

MATLAB program code

```
%Gauss Seidal
clc
format compact
A=input('Enter the Coeficient Matrix A: ');
B=input('Enter the solution Matrix b (column matrix): ');
C=[A B];
n=input('Enter the number of iteration: ');
X=input('Enter the initial guess (Column matrix): ');
x(1)=X(1,1);
y(1)=X(2,1);
z(1)=X(3,1);
for j=1:3 %checking for zero in diagonal entries
   diag=0;
   if A(j,j)==0
      diag=1;
   end
end
if diag==1
   disp('Error! Please check the input, diagonal elements of the matrix cannot be zero')
else
   eig_A = eig(A); %Checking for positive definite
   positive = 0;
   for i = 1:rank(A)
      if eig_A(i) <= 0
         positive = 1;
      end
   end

   diagdom=0; %Checking for diagonally dominant
   for j=1:length(B)
      if abs(A(j,j))>= sum(abs(A(j,:)))-abs(A(j,j))
         diagdom=1;
      end
   end

   for i=1:n %Iterative step
      x(i+1)=(1/C(1,1))*(C(1,4)-C(1,2)*y(i)-C(1,3)*z(i));
      y(i+1)=(1/C(2,2))*(C(2,4)-C(2,1)*x(i+1)-C(2,3)*z(i));
      z(i+1)=(1/C(3,3))*(C(3,4)-C(3,1)*x(i+1)-C(3,2)*y(i+1));
      error1=abs((x(i)-x(i+1))/x(i+1))*100;
      error2=abs((y(i)-y(i+1))/y(i+1))*100;
      error3=abs((z(i)-z(i+1))/z(i+1))*100;
   end
end

if positive == 1
   disp('the matrix is not positive definite hence we can not expect convergence. ')
else
   if diagdom==1
      disp('the matrix is positive definite and diagonally dominant hence, we can expect convergence.')
   else
      disp('the matrix is positive definite but not diagonally dominant hence we can not expect convergence. ')
   end
end
```

```
error=[error1; error2; error3];


disp('a1= ')
disp(x(i+1))
disp('a2= ')
disp(y(i+1))
disp('a3= ')
disp(z(i+1))
disp('Absolute relative approximate error for each a in %')
disp(error)
```

Problem

The upward velocity of a rocket is given at three different times in the following table

Time, $t\ (s)$	Velocity $v\ (m/s)$
5	106.8
8	177.2
12	279.2

Table 1. upward velocity of a rocket

The velocity data is approximated by a polynomial as

$$v(t) = a_1 t^2 + a_2 t + a_3, \qquad 5 \le t \le 12$$

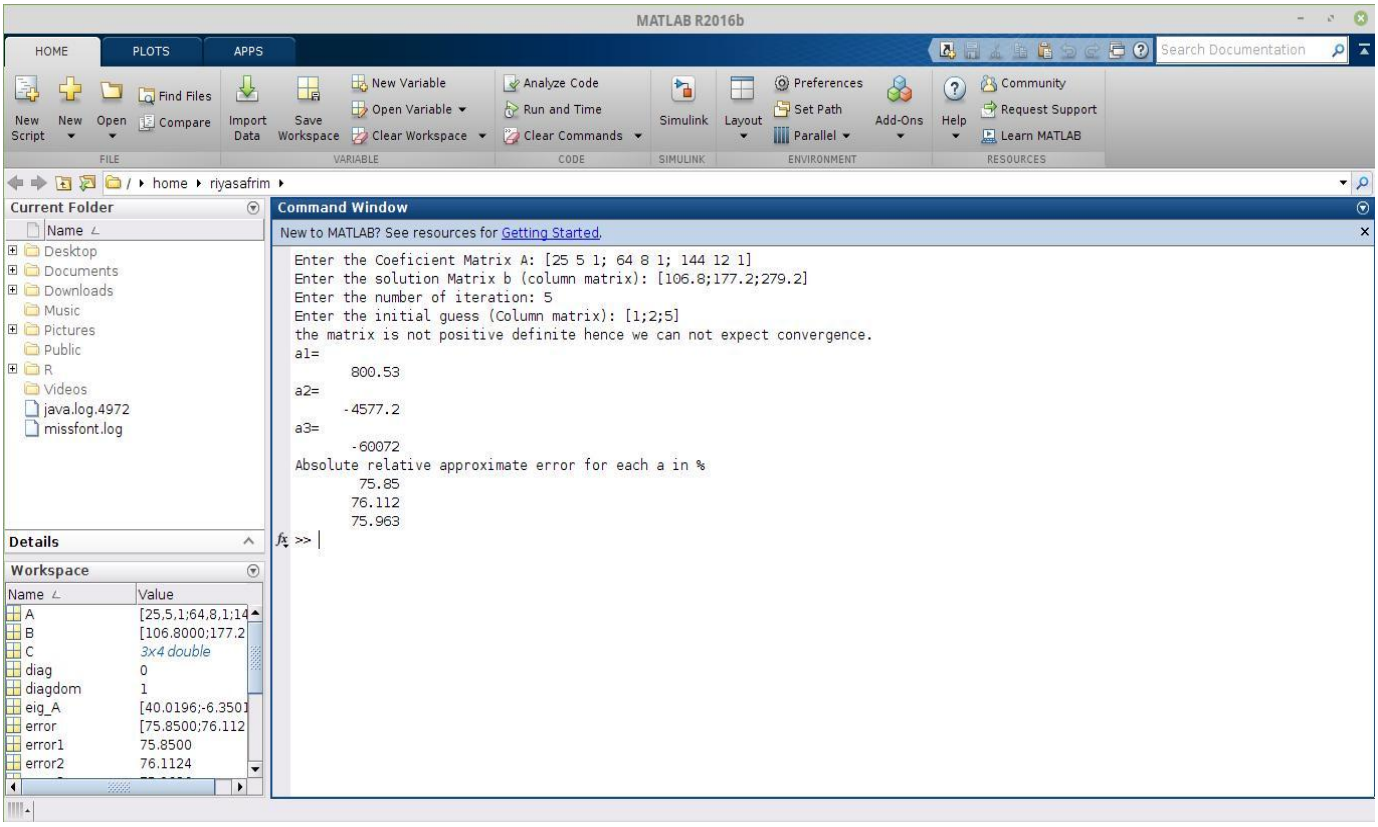
Find the values of a_1, a_2, a_3? Assume the initial guess of the solution as $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$ and conduct 5 iterations (5).

The upward velocities of the rocket is given at three different times, hence the polynomial approximation of velocity data is going through the points $(5, 106.8)$, $(8, 177.2)$ and $(12, 279.2)$ from the Table 1. upward velocity of a rocket

$\therefore v(t_i) = v_i$ for $1 \le i \le 3$

$$\Rightarrow a_1(5^2) + a_2(5) + a_3 = 106.8$$
$$a_1(8^2) + a_2(8) + a_3 = 177.2$$
$$a_1(12^2) + a_2(12) + a_3 = 279.2$$

or
$$25a_1 + 5a_2 + a_3 = 106.8$$
$$64a_1 + 8a_2 + a_3 = 177.2$$
$$44a_1 + 12a_2 + a_3 = 279.2$$



CONCLUSION

In this paper we developed MATLAB code to perform Gauss-Seidel numerical to solve the general linear system of three equations numerically. One can easily manipulate the code to perform for linear system of n equations for any natural number n. Also we have developed code to check positive definite and diagonally dominant within the above.

REFERENCES

1. **Attaway, Stormy.** *Matlab: A Practical Introduction to Programming and Problem Solving.* Canada : Elsevier, Inc., 2009.
2. **Gilat, Amos.** *MATLAB An Introduction with Applications.* Columbia, Ohio : John Wiley & Sons, Inc., 2011.
3. *Convergence of Jacobi and Gauss-Seidel Method and Error Reduction Factor.* **HarpinderKaur, KhushpreetKaur.** 2, Department of Mathematics Baba FaridCollege, Bathinda : IOSR Journal of Mathematics (IOSRJM), 2012, Vol. 2.
4. **Saleri, Alfio QuarteroniRiccardo SaccoFausto.** *Numerical Mathematics.* New York, NY : Springer, 2011. ISBN 978-1-4757-7394-1.
5. **Kaw, Autar K, Kalu, Egwalu E and Nguyen, Duc.** *Numerical Methods with Applications: Abridged.* Florida : University of South Florida, 2011.
6. **M.K. Jain, S.R.K. Iyengar, R.K. Jain.** *Numerical methods for Scientific and Engineering computaion.* New Delhi : New Age International (P) Limited, Publishers.