SOLUTION OF SYSTEMS OF SIMULTANEOUS LINEAR EQUATIONS

Gauss-Seidel Method

2006 *Jamie Trahan, Autar Kaw, Kevin Martin*
University of South Florida
United States of America
kaw@eng.usf.edu

Introduction

This worksheet demonstrates the use of Mathcad to illustrate Gauss-Seidel Method, an iterative technique used in solving a system of simultaneous linear equations.

Gauss-Seidel method is used to solve a set of simultaneous linear equations, [A] [X] = [RHS], where $[A]_{nxn}$ is the square coefficient matrix, $[X]_{nx1}$ is the solution vector, and $[RHS]_{nx1}$ is the right hand side array. The equations can be rewritten as

$$x_i := \frac{\left[rhs_i - \sum_{i=1}^{n} \left[\left(a_{i,j} \right) \cdot x_j \right], i \neq j \right]}{a_{i,i}} \qquad \text{Equation (1.1)}$$

In certain cases, such as when a system of equations is large, iterative methods of solving equations such as Gauss-Seidel method are more advantageous. Elimination methods, such as Gaussian Elimination, are prone to round-off errors for a large set of equations whereas iterative methods, such as Gauss-Seidel method, allow the user to control round-off error. Also if the physics of the problem are well known, initial guesses needed in iterative methods can be made more judiciously for faster convergence.

The steps to apply Gauss-Seidel method are:
1) Make an initial guess for the solution vector [X]. This can be based on the physics of the problem.
2) All proper values are plugged into Equation (1.1). The new x_1 value that is calculated will replace the previous guess, x_1, in the solution vector. [X] will then be used to calculate x_2. This will be done for each x_i from x_1 to x_n until a new solution vector is complete. At this point, the first iteration is done.
3) The absolute relative approximate error is calculated by comparing each new guess x_i with the previous guess. The maximum of these errors is the absolute relative approximate error at the end of the iteration.
4) The new solution vector becomes the old solution vector and Steps 2-3 are repeated until either the maximum number of iterations have been conducted or the pre-specified tolerance has been met.

Complete details of how Equation (1.1) is derived as well as the pitfalls of the method can be found **here.**

An example demonstrating Gauss-Seidel method follows.

Section 1: Input

The following are the input parameters to begin the simulation. This is the only section that requires user input. The user can change those values that are highlighted and the worksheet will use four iterations to calculate an approximate solution to the system of equations.

$ORIGIN := 1$

- Number of equations.

$n := 4$

- The *n*x*n* coefficient matrix [A]. Note that if the coefficient matrix is **diagonally dominant**, convergence of the solution is ensured. Otherwise, the solution may or may not converge.

$$A := \begin{pmatrix} 10 & 3 & 4 & 5 \\ 2 & 24 & 2 & 4 \\ 2 & 2 & 34 & 3 \\ 2 & 2 & 2 & 12 \end{pmatrix}$$

- *n*x1 right hand side array.

$$RHS := \begin{pmatrix} 22 \\ 32 \\ 41 \\ 18 \end{pmatrix}$$

- *n*x1 initial guess of the solution vector.

$$Xinit := \begin{pmatrix} 1 \\ 23 \\ 4 \\ 50 \end{pmatrix}$$

Section 2: Gauss-Seidel Iterations

Four iterations will be conducted using Gauss-Seidel method. For each iteration, the following three values will be calculated:

1) New estimate of the solution vector
2) Absolute relative approximate error (abs_ea) for each x_i
3) Maximum relative approximate error (Max_abs_ea) for the given iteration

Iteration 1

1) For the first iteration, the initial guess values of the solution vector [Xinit], as well as the proper elements of coefficient matrix [A] and the right hand side vector [RHS] are substituted into Equation (1.1) to calculate a new, approximate solution vector, denoted as [Xnew1] for the first iteration.

The following procedure calculates [Xnew1] using Equation (1.1).

Xnew1 :=	for $i \in 1..n$	Defining row elements
	$\quad sum \leftarrow 0$	Initializing the series sum to 0
	$\quad$ for $j \in 1..n$	Defining column elements
	$\quad\quad sum \leftarrow sum + A_{i,j} Xinit_j$ if $j \neq i$	Generating the summation term by only adding $i \neq j$ terms
	$\quad Xinit_i \leftarrow \dfrac{RHS_i - sum}{A_{i,i}}$	Applying equation (1.1)
	$\quad X \leftarrow Xinit$	Replacing old guess with new guess
	X	Returning new, approximate [X] vector

The new approximate solution vector is

$$Xnew1 = \begin{pmatrix} -31.3 \\ -4.725 \\ -1.08676 \\ 7.68529 \end{pmatrix}$$

2) The absolute relative percentage approximate error for each $Xnew_i$ is calculated by the following procedure

$$\text{abs\_ea1} := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{absea}_i \leftarrow \left| \dfrac{\text{Xnew1}_i - \text{Xinit}_i}{\text{Xnew1}_i} \right| \cdot 100.0 \\ \text{absea} \end{array} \right.$$

The absolute relative percentage approximate error is

$$\text{abs\_ea1} = \begin{pmatrix} 103.19489 \\ 586.77249 \\ 468.06495 \\ 550.59319 \end{pmatrix}$$

3) The maximum of these errors is the absolute relative approximate error at the end of the first iteration.

$$\text{Max\_abs\_ea1} := \left| \begin{array}{l} \text{max\_absea} \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \text{max\_absea} \leftarrow \text{abs\_ea1}_i \ \text{ if } \ \text{abs\_ea1}_i > \text{max\_absea} \\ \text{max\_absea} \end{array} \right.$$

The maximum absolute relative percentage approximate error at the end of iteration 1 is

$$\text{Max\_abs\_ea1} = 586.77249$$

Iteration 2

1) The new solution vector, [Xnew1], obtained in iteration 1 will become the old solution vector for iteration 2.

Defining the old solution vector as the solution vector that was obtained from the previous iteration:

$$Xold := Xnew1$$

$$Xold = \begin{pmatrix} -31.3 \\ -4.725 \\ -1.08676 \\ 7.68529 \end{pmatrix}$$

The new approximate solution vector for iteration 2 will now be calculated with [Xold], [A], and [RHS] by substituting the proper elements into Equation (1.1).

```
Xnew2 := | for i ∈ 1 .. n                                  Defining row elements.
         |   | sum ← 0                                      Initializing the series sum to 0.
         |   | for j ∈ 1 .. n                               Defining column elements.
         |   |   sum ← sum + A[i,j] Xold[j]  if j ≠ i       Generating the summation term by only
         |   |                                              adding i ≠ j terms.
         |   | Xold[i] ← (RHS[i] − sum) / A[i,i]            Applying Equation (1.1).
         |   | X ← Xold                                     Replacing old guess with new guess.
         | X                                                Returning new, approximate [X] vector.
```

At the end of the 2nd iteration, the new estimate of the solution vector is

$$Xnew2 = \begin{pmatrix} 0.20956 \\ 0.12555 \\ 0.50806 \\ 1.35947 \end{pmatrix}$$

2) The absolute relative percentage approximate error for the second iteration is calculated by the following procedure

$$\text{abs\_ea2} := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{absea}_i \leftarrow \left| \dfrac{\text{Xnew2}_i - \text{Xold}_i}{\text{Xnew2}_i} \right| \cdot 100.0 \\ \text{absea} \end{array} \right.$$

The absolute relative percentage approximate error is

$$\text{abs\_ea2} = \begin{pmatrix} 15036.14 \\ 3863.397 \\ 313.907 \\ 465.314 \end{pmatrix}$$

3) The maximum absolute relative approximate error for the second iteration is defined by the following procedure

$$\text{Max\_abs\_ea2} := \left| \begin{array}{l} \text{max\_absea} \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \text{max\_absea} \leftarrow \text{abs\_ea2}_i \ \text{ if } \ \text{abs\_ea2}_i > \text{max\_absea} \\ \text{max\_absea} \end{array} \right.$$

The maximum absolute relative percentage approximate error for the iteration 2 is

$$\text{Max\_abs\_ea2} = 15036.14035$$

Iteration 3

1) Again, the new estimate of the solution vector from the previous iteration will replace the old solution vector.

$$Xold := Xnew2$$

$$Xold = \begin{pmatrix} 0.20956 \\ 0.12555 \\ 0.50806 \\ 1.35947 \end{pmatrix}$$

Substituting the proper [Xold], [A], and [RHS] elements into Equation (1.1) with the following procedure will return an estimate of the solution vector after 3 iterations.

```
Xnew3 := | for i ∈ 1 .. n                          Defining row elements.
         |   | sum ← 0                              Initializing the series sum to 0.
         |   | for j ∈ 1 .. n                       Defining column elements.
         |   |   sum ← sum + A[i,j] Xold[j]  if j ≠ i   Generating the summation term by only
         |   |                                      adding i ≠ j terms.
         |   | Xold[i] ← (RHS[i] − sum) / A[i,i]    Applying Equation (1.1).
         |   | X ← Xold                             Replacing old guess with new guess.
         | X                                        Returning new estimate of solution vector [X].
```

The new guess solution vector after the third iteration is:

$$Xnew3 = \begin{pmatrix} 1.27938 \\ 0.9578 \\ 0.95433 \\ 0.96808 \end{pmatrix}$$

2) Calculating the absolute relative approximate error for the third iteration:

$$\text{abs\_ea3} := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{absea}_i \leftarrow \left| \dfrac{\text{Xnew3}_i - \text{Xold}_i}{\text{Xnew3}_i} \right| \cdot 100.0 \\ \text{absea} \end{array} \right.$$

The absolute relative percentage approximate error for iteration 3 is

$$\text{abs\_ea3} = \begin{pmatrix} 83.62023 \\ 86.89171 \\ 46.7631 \\ 40.42947 \end{pmatrix}$$

3) Defining the maximum absolute relative percentage error for iteration 3:

$$\text{Max\_abs\_ea3} := \left| \begin{array}{l} \text{max\_absea} \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \text{max\_absea} \leftarrow \text{abs\_ea3}_i \ \text{ if } \ \text{abs\_ea3}_i > \text{max\_absea} \\ \text{max\_absea} \end{array} \right.$$

The maximum absolute relative approximate error for the third iteration is

$$\text{Max\_abs\_ea3} = 86.89171$$

Iteration 4

1) The new solution vector obtained at the end of iteration 3 becomes the old guess solution vector:

$$Xold := Xnew3$$

$$Xold = \begin{pmatrix} 1.27938 \\ 0.9578 \\ 0.95433 \\ 0.96808 \end{pmatrix}$$

Substituting the proper [Xold], [A], and [RHS] elements into equation (1.1) with the following procedure will yield the final estimate of the solution vector.

```
Xnew4 := | for i ∈ 1 .. n                          Defining row elements
         |   | sum ← 0                              Initializing the series sum to 0.
         |   | for j ∈ 1 .. n                       Defining column elements.
         |   |   sum ← sum + A_{i,j} Xold_j  if j ≠ i   Generating the summation term by only
         |   |                                          adding i ≠ j terms.
         |   | Xold_i ← (RHS_i − sum) / A_{i,i}     Applying Equation (1.1).
         |   | X ← Xold                             Replacing old guess with new guess.
         | X                                        Returning new, approximate [X] vector.
```

The final estimate of the solution vector is

$$Xnew4 = \begin{pmatrix} 1.04689 \\ 1.00522 \\ 0.99975 \\ 0.99136 \end{pmatrix}$$

2) Calculating the absolute relative percentage approximate error for the fourth iteration:

$$\text{abs\_ea4} := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{absea}_i \leftarrow \left| \dfrac{\text{Xnew4}_i - \text{Xold}_i}{\text{Xnew3}_i} \right| \cdot 100.0 \\ \text{absea} \end{array} \right.$$

The absolute relative percentage approximate error for iteration 3 is

$$\text{abs\_ea4} = \begin{pmatrix} 18.17212 \\ 4.95054 \\ 4.75948 \\ 2.40428 \end{pmatrix}$$

3) Defining the maximum absolute relative percentage error for iteration 4:

$$\text{Max\_abs\_ea4} := \left| \begin{array}{l} \text{max\_absea} \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \text{max\_absea} \leftarrow \text{abs\_ea4}_i \ \text{ if } \ \text{abs\_ea4}_i > \text{max\_absea} \\ \text{max\_absea} \end{array} \right.$$

The maximum absolute relative percentage approximate error for the fourth and final iteration is

$$\text{Max\_abs\_ea4} = 18.17212$$

Section 3: Exact Solution

The exact solution to the system of linear equations can be found by using Mathcad's built-in tools.

$$\text{exactsoln} := \text{lsolve}(A, RHS)$$

The exact solution is

$$\text{exactsoln} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The following procedure calculates the absolute relative percentage true error (abs_et).

$$\text{abs\_et} := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{abs\_et}_i \leftarrow \left| \dfrac{\text{exactsoln}_i - \text{Xnew4}_i}{\text{exactsoln}_i} \right| \cdot 100.0 \\ \text{abs\_et} \end{array} \right.$$

The absolute relative percentage true error is

$$\text{abs\_et} = \begin{pmatrix} 4.68864 \\ 0.52183 \\ 0.02487 \\ 0.86427 \end{pmatrix}$$

The maximum absolute relative percentage true error is

$$\text{Max\_abs\_et} := \left| \begin{array}{l} \text{max\_abset} \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \text{max\_abset} \leftarrow \text{abs\_et}_i \ \text{ if } \ \text{abs\_et}_i > \text{max\_abset} \\ \text{max\_abset} \end{array} \right.$$

$$\text{Max\_abs\_et} = 4.68864$$

References

Autar Kaw, *Holistic Numerical Methods Institute*, *http://numericalmethods.eng.usf.edu/mws*

How does Gauss-Seidel method work?

Conclusions

Mathcad helped us apply our knowledge of Gauss-Seidel method to solve a system of *n* simultaneous linear equations. The values obtained are

Final approximate solution vector using 4 iterations

$$\text{Xnew4} = \begin{pmatrix} 1.04689 \\ 1.00522 \\ 0.99975 \\ 0.99136 \end{pmatrix}$$

Exact value

$$\text{exactsoln} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Absolute relative percentage true error

$$\text{abs\_et} = \begin{pmatrix} 4.68864 \\ 0.52183 \\ 0.02487 \\ 0.86427 \end{pmatrix}$$

Question 1: Change the coefficient matrix to one that is not diagonally dominant and see if Gauss-Seidel method converges.

Question 2: See if you can get a set of equations with a coefficient matrix that is not diagonally dominant to converge by Gauss-Seidel method.