Convergence of Gauss-Seidel Method

Jamie Trahan, Autar Kaw, Kevin Martin

University of South Florida

United States of America

kaw@eng.usf.edu

Introduction

Gauss-Seidel method is an advantageous approach to solving a system of simultaneous linear equations because it allows the user to control round-off error that is inherent in elimination methods such as Guassian Elimination. However, this method is not without its pitfalls. Gauss-Seidel method is an iterative technique whose solution may or may not converge. Convergence is only ensured is the coefficient matrix, $[A]_{nxn}$,is diagonally dominant, otherwise the method may or may not converge.

A diagonally dominant square matrix [A] is defined by the following:

$$\left| a_{i,i} \right| \geq \left(\sum_{j=1}^{n} |a_{i,j}|\right)_{i \neq j} \quad \text{for all } i \text{, and}$$

$$\left| a_{i,i} \right| > \left(\sum_{j=1}^{n} |a_{i,i}|\right)_{i \neq j} \quad \text{for at least one } i.$$

Fortunately, many physical systems that result in simultaneous linear equations have diagonally dominant coefficient matrices, or with the exchange of a few equations, the coefficient matrix can become diagonally dominant. To learn more about diagonally dominant matrices as well as how to perform Gauss-Seidel method, click here**.**

The following simulation illustrates the convergence of Gauss-Seidel method.

Section 1: Input

The following are the input parameters for the simulation. The user may change values only in this section. Once entered, *Mathematica* will produce plots that demonstrate the convergence of each solution X_i as a function of the iteration number.

- Number of equations, *n*

```
n = 3
```

```
3
```

- *n*x*n* coefficient matrix, [A]

```
A = Table[{{12, 7, 3, 1}, {1, 5, 1, 2}, {2, 7, -11, 1}, {9, 2, 1, 13}}];
A // MatrixForm
```

$$\begin{pmatrix} 12 & 7 & 3 & 1 \\ 1 & 5 & 1 & 2 \\ 2 & 7 & -11 & 1 \\ 9 & 2 & 1 & 13 \end{pmatrix}$$

- *n*x1 right hand side array, [RHS]

```
RHS = Table[{22, 7, -2, 3}];
RHS // MatrixForm
```

$$\begin{pmatrix} 22 \\ 7 \\ -2 \\ 3 \end{pmatrix}$$

- *n*x1 initial guess of the solution vector

```
X = Table[{1, 2, 1, 3.001}];
X // MatrixForm
```

$$\begin{pmatrix} 1 \\ 2 \\ 1 \\ 3.001 \end{pmatrix}$$

- maximum number of iterations

```
maxit = 8
```

```
8
```

Section 2: Gauss-Seidel Procedure

The following procedure will use Gauss-Seidel method to calculate the value of the solution for the above system of equations using *maxit* iterations. It will then store each approximate solution, X_i, from each iteration in a matrix with *maxit* columns. Thereafter, *Mathematica* will plot the solutions as a function of the iteration number.

Variable Parameters:
A = *n*x*n* coefficient matrix
RHS = *n*x1 right hand side array
n = number of equations
Xinitial = *n*x1 initial guess solution vector
maxit = maximum number of iterations

```
gaussseidel[A_, RHS_, n_, Xinitial_, maxit_] := Module[{},
   epsa = Array[x, n];
   Xnew = Array[x, n];
   epsmax = Array[x, maxit];
   Xstore = Array[s, {n, maxit}];
   Xprev = Xinitial;
   For[k = 1, k ≤ maxit, k++,
    epsmax[[k]] = 0;
    For[i = 1, i ≤ n, i++,
     summ = 0;
     For[j = 1, j ≤ n, j++,
      If[i ≠ j, summ = summ + A[[i, j]] * Xprev[[j]]]];
     Xnew[[i]] = N[(RHS[[i]] - summ) / A[[i, i]]];
     epsa[[i]] = Abs[(Xnew[[i]] - Xprev[[i]]) / Xnew[[i]]] * 100;
     If[epsmax[[k]] ≤ epsa[[i]], epsmax[[k]] = epsa[[i]]];
     Xprev[[i]] = Xnew[[i]];
     Xstore[[i, k]] = Xnew[[i]];]];];
```

Section 3: Results

```
gaussseidel[A, RHS, n, X, maxit];
```

The following table displays the value of the solution for X_i in the i^{th} row after each given iteration.

```
Xvar = Table["X"J, {J, 1, n, 1}];
Iter = Table["iteration"[i], {i, 1, maxit, 1}];
TableForm[Xstore, TableHeadings → {Xvar, Iter}]
```

	iteration[1]	iteration[2]	iteration[3]	iteration[4]	iteration
X_1	0.416667	0.939899	0.989083	0.99899	0.999793
X_2	1.11667	1.01838	1.00203	1.00034	1.00003
X_3	0.968182	1.00077	0.999306	1.00003	0.999984

The following matrix stores the maximum absolute relative approximate error after each iteration.

```
Iter = Table["iteration"[i], {i, 1, maxit, 1}];
Print["Maximum absolute relative approximate error for each iteration:"]
TableForm[epsmax, TableDirections → {Row}, TableHeadings → {Iter}]
```

Maximum absolute relative approximate error for each iteration:

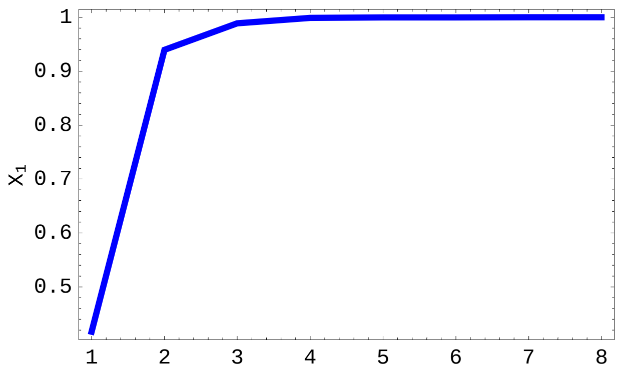
iteration[1]	iteration[2]	iteration[3]	iteration[4]	iteration[5]	iteration[6]	ite
140.	55.669	4.97271	0.991654	0.0803311	0.0190671	0.

Section 4: Convergence Graphs

The graphs below plot the value of X_i as a function of the iteration number in order to display the convergence of each solution.

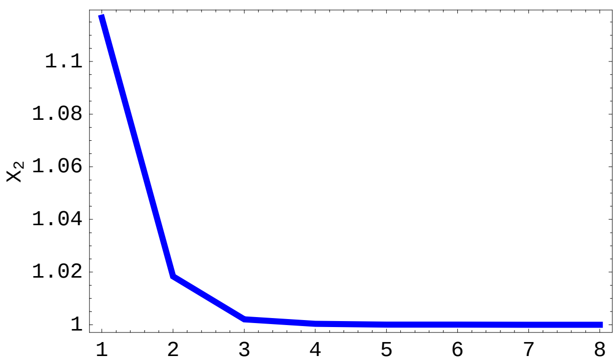
```
For[i = 1, i ≤ n, i++,
 Print["  "];
 ttl = Print["Value of", " ", "X"i, " ", "as a function of the iteration number"];
 data = Table[{k, Xstore[[i, k]]}, {k, 1, maxit}];
 ListPlot[data,
  PlotJoined → True,
  PlotRange → All,
  Frame → True,
  FrameLabel → "X"i,
  PlotStyle → {Thickness[0.012], RGBColor[0, 0, 1]}];
 Print[data];
 Print["  "]; Print["_____________________________________"]; Print["  "]]
```

Value of X_1 as a function of the iteration number



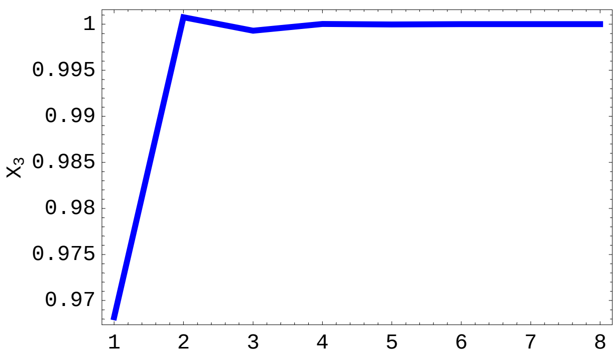
{{1, 0.416667}, {2, 0.939899}, {3, 0.989083}, {4, 0.99899}, {5, 0.999793}, {6, 0.999984}, {7, 0.999996}, {8, 1.}}

Value of X_2 as a function of the iteration number



{{1, 1.11667}, {2, 1.01838}, {3, 1.00203}, {4, 1.00034}, {5, 1.00003}, {6, 1.00001}, {7, 1.}, {8, 1.}}

Value of X_3 as a function of the iteration number

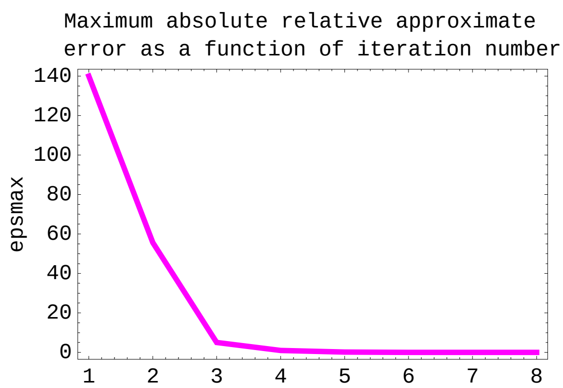


{{1, 0.968182}, {2, 1.00077}, {3, 0.999306}, {4, 1.00003}, {5, 0.999984}, {6, 1.}, {7, 1.}, {8, 1.}}

The graph below plots the value of the maximum absolute relative approximate error as a function of the iteration number.

```
data2 = Table[{k, epsmax[[k]]}, {k, 1, maxit, 1}]
plot2 = ListPlot[data2,
  PlotLabel →
   "Maximum absolute relative approximate \nerror as a function of iteration number",
  PlotJoined → True,
  Frame → True,
  FrameLabel → "epsmax",
  PlotStyle → {Thickness[0.012], RGBColor[1, 0, 1]}]
```

```
{{1, 140.}, {2, 55.669}, {3, 4.97271}, {4, 0.991654},
 {5, 0.0803311}, {6, 0.0190671}, {7, 0.00124142}, {8, 0.000374353}}
```




```
- Graphics -
```

Conclusion

Mathematica helped us to study the convergence of Gauss-Seidel method.

Question: Solve a set of equations for which the coefficient matrix is not diagonally dominant. For example,

$5x + 6y + 7z = 18$
$6x + 3y + 9z = 18$
$7x + 9y + 10z = 26$

Choose an initial solution vector guess of [2, 5, 7]. Does the solution converge? Now choose [0.99, 0.995, 0.997] as the initial guess of the solution vector. Does the solution converge now?

References

[1] Autar Kaw, *Holistic Numerical Methods Institute, http://numericalmethods.eng.usf.edu/mws,* See
How does Gauss-Seidel method work?