Convergence of Gauss-Seidel Method

©2006 Kevin Martin, Autar Kaw, Jamie Trahan

University of South Florida

United States of America

kaw@eng.usf.edu

NOTE: This worksheet demonstrates the convergence of Gauss-Seidel method, an iterative technique used in solving a system of simultaneous linear equations.

```
> restart;
```

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 Gaussian 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 ensured only if 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:

abs(*a*[*i*, *i*]) >= *Sum*(*abs*(*a*[*i*,*j*]), *j* = 1 ..*n*)[*i* <> *j*]

$$\left(\sum_{j=1}^{n} \left|a_{i,j}\right|\right)_{i \neq j} \leq \left|a_{i,i}\right| \qquad (1.1)$$

for all *i*, and

abs(*a*[*i*, *i*]) > *Sum*(*abs*(*a*[*i*,*j*]), *j* = 1 ..*n*)[*i* <> *j*]

$$\left(\sum_{j=1}^{n} \left|a_{i,j}\right|\right)_{i \neq j} < \left|a_{i,i}\right| \qquad (1.2)$$

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 the Gauss-Seidel method, click here.

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

Section 1: Input Data

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

Input parameters:
n = number of equations
[A] = *nxn* coefficient matrix
[RHS] = *n*x*1* right hand side array
[X] = *n*x*1* initial guess of the solution vector
maxit = maximum number of iterations

```
> restart;
```

```
> n:=4;
  A:=Matrix([[12,7,3,1],[1,5,1,2],[2,7,-11,1],[9,2,1,13]]);
  RHS:=[22,7,-2,3];
  X:=[1,2,1,1];
  maxit:=8;
```

$$n := 4$$

$$A := \begin{bmatrix} 12 & 7 & 3 & 1 \\ 1 & 5 & 1 & 2 \\ 2 & 7 & -11 & 1 \\ 9 & 2 & 1 & 13 \end{bmatrix}$$

$$RHS := [22, 7, -2, 3]$$

$$X := [1, 2, 1, 1]$$

$$maxit := 8 \tag{2.1}$$

Section 2: Gauss-Seidel Procedure

Gauss-Seidel method utilizes the equation

$$x[i] = \frac{(rhs[i] - Sum(a[i,j] \cdot x[j], i = 1 .. "n")[i <> j])}{a[i,i]}$$

$$x_i = \frac{rhs_i - \left(\sum_{i=1}^{\text{"n"}} a_{i,j} x_j\right)_{i \neq j}}{a_{i,i}} \tag{3.1}$$

to compute an approximate value for a solution vector [X]. The following procedure uses 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, Maple will plot the solutions as a function of the iteration number.

Parameter names:
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

```
> gauss_seidel:=proc(A,RHS,n,Xinitial,maxit)
  local Xold,Xnew,Xstore,Xprev,i,j,k,summ,epsa,epsmax;
  #epsa is the array that stores the absolute relative approximate error at the end of each iteration.
  epsa:=Array(1..n);
  #Xnew is the solution vector after each iteration is conducted.
  Xnew:=Array(1..n);
  #epsmax is the greatest relative approximate error of all values in the solution vector that are
  generated in the given iteration.
  epsmax:=Array(1..maxit);
  #Xstore is a matrix that stores the solution vector after each iteration.
  Xstore:=Matrix(1..n,1..maxit);
  #Defining the initial guess values of the solution vector.
  Xprev:=Xinitial:
  #conducting maxit iterations.
  for k from 1 by 1 to maxit do
     epsmax[k]:=0.0;
         for i from 1 by 1 to n do
             #Initializing the series sum to zero.
             summ:=0.0;
                for j from 1 by 1 to n do
                   #Only adding i<>j terms.
                   if (i<>j) then
                      #Generating the summation term in Equation (3.1).
                      summ:=summ+A[i,j]*Xprev[j];
                   end if:
                end do:
```

```
                #Using Equation (3.1) to calculate the new [X] solution vector.
                Xnew[i]:=(RHS[i]-summ)/A[i,i];
                #Calculating the absolute relative approximate error.
                epsa[i]:=abs((Xnew[i]-Xprev[i])/Xnew[i])*100.0;
                    #Finding the maximum epsa value.
                    if epsmax[k]<=epsa[i] then
                       epsmax[k]:=epsa[i];
                    end if;
                #Updating the previous guess.
                Xprev[i]:=Xnew[i];
                #Storing each value of X for each iteration.
                Xstore[i,k]:=Xnew[i]
                end do;
         end do;
         return(Xstore,epsmax);
         end proc:
```

Section 3: Results

```
> soln:=gauss_seidel(A,RHS,n,X,maxit):
```

The following matrix stores the value of the solution for X_i in the ith row after each given iteration.

```
> Xstore:=soln[1];
```

$$Xstore := \left[\begin{array}{l}[0.3333333333, 1.220085470, 1.016817086, 0.8394342433, 0.8466193725, \\ 0.8771302825, 0.8805026500, 0.8759440883], [0.7333333334, 1.065726496, 1.327383460, \\ 1.311294467, 1.265470424, 1.261284449, 1.268262415, 1.270001019], [0.8000000001, \\ 1.065990676, 1.133223260, 1.099396804, 1.083167960, 1.086353770, 1.089523450, \\ 1.089468999], [-.1743589744, -.8598625308, -.7645649192, -.6366841477, \\ -.6333602431, -.6540842469, -.6577363177, -.6548436792]\end{array}\right] \quad (4.1)$$

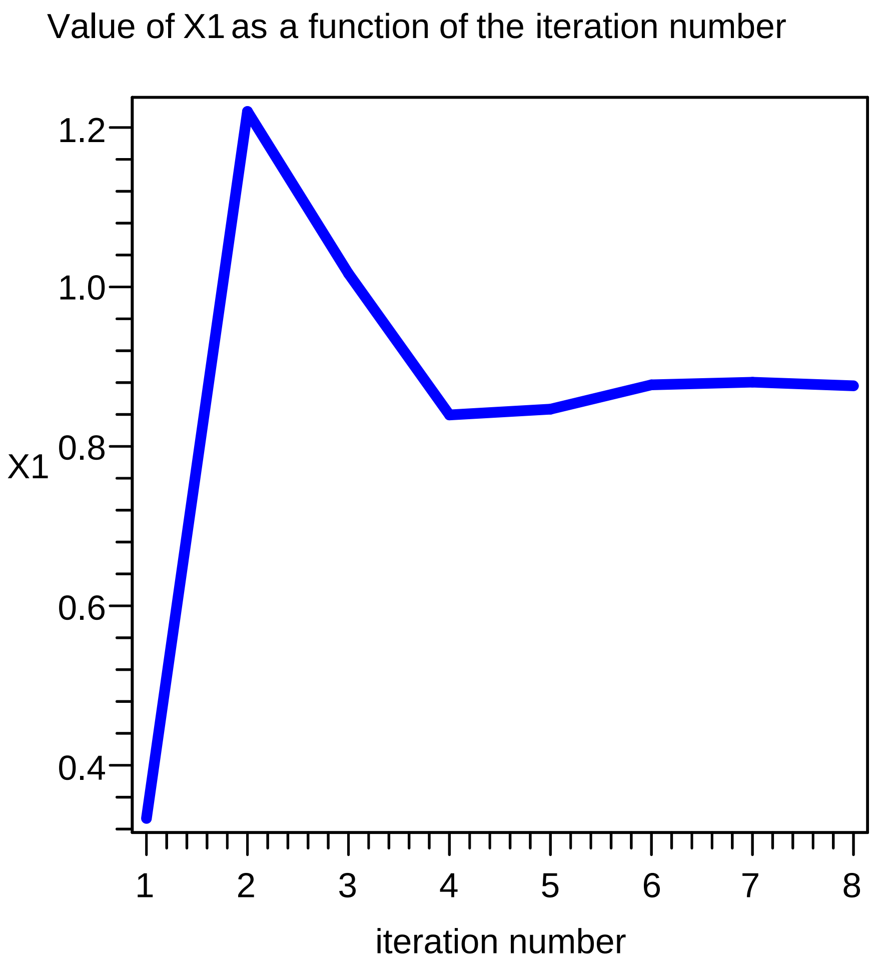
The following matrix stores the maximum absolute relative approximate error after each given iteration.

```
> epsmax:=soln[2];
epsmax := [[673.5294114, 79.72245933, 19.99065385, 21.13123739, 3.621107387,        (4.2)
    3.478492376, 0.5552484638, 0.5204169719]]
```

Section 4: Convergence Graphs

The following graphs plot the value of X_i as a function of the iteration number to demonstrate the convergence of each solution.

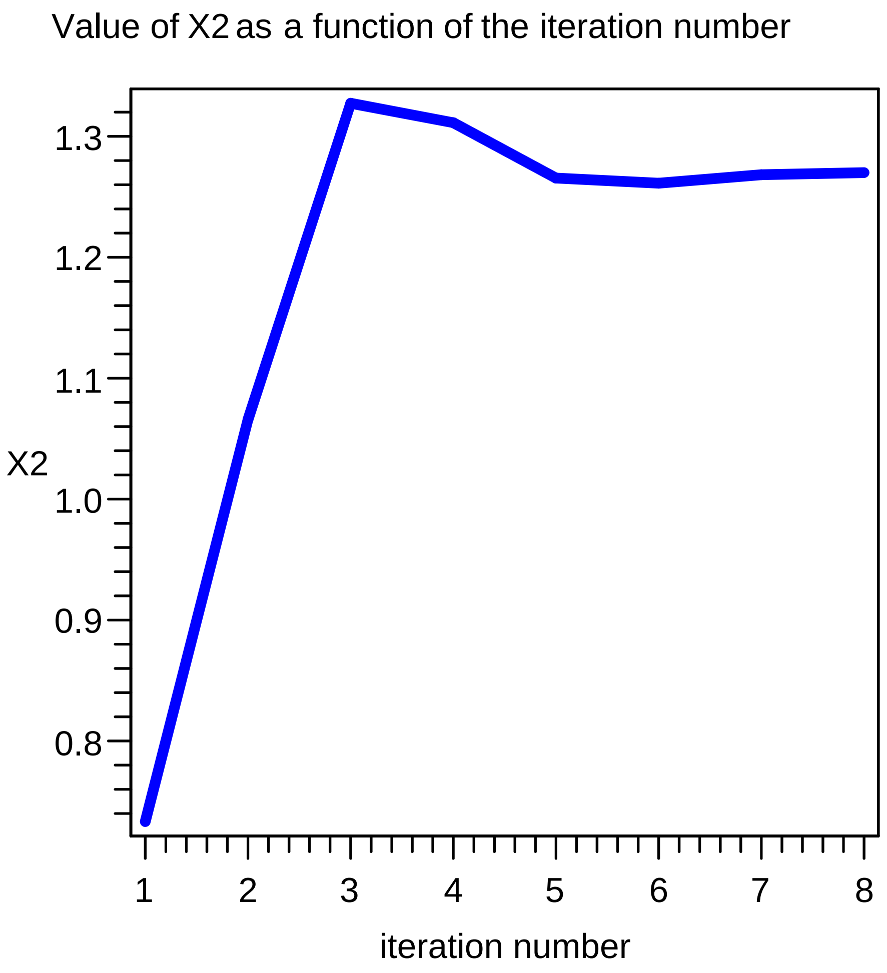
```
> with(plots):
Warning, the name changecoords has been redefined
> for i from 1 by 1 to n do
  ttl:=cat(`Value of X`,i,` as a function of the iteration
  number`):
  lbl:=cat("X",i):
  data:=[seq([k,Xstore[i,k]],k=1..maxit)];
  pointplot(data,connect=true,color=blue,axes=boxed,title=ttl,
  axes=BOXED,labels=["iteration number",lbl],thickness=3);
  end do;
                    ttl := Value of X1 as a function of the iteration number
                                          lbl := "X1"
data := [ [1, 0.3333333333 ], [2, 1.220085470 ], [3, 1.016817086 ], [4, 0.8394342433 ]
    , [5, 0.8466193725 ], [6, 0.8771302825 ], [7, 0.8805026500 ], [8, 0.8759440883 ] ]
```



ttl := *Value of X2 as a function of the iteration number*

lbl := "X2"

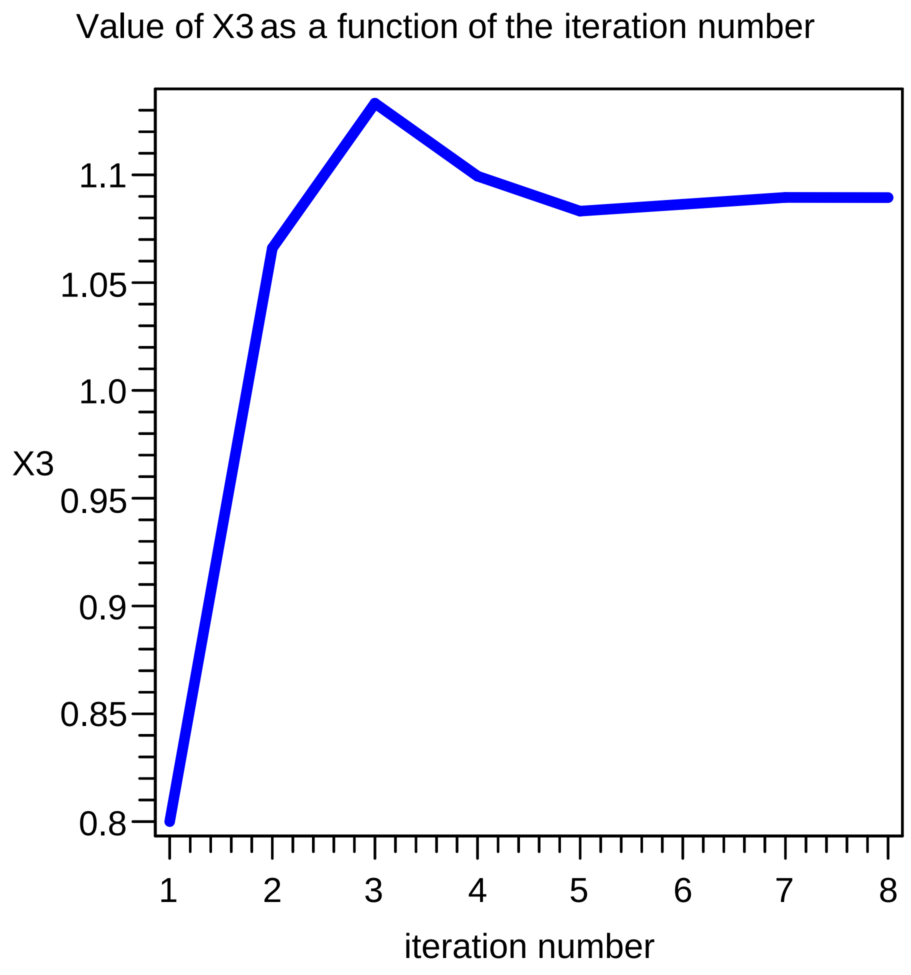
data := [[1, 0.7333333334], [2, 1.065726496], [3, 1.327383460], [4, 1.311294467]
, [5, 1.265470424], [6, 1.261284449], [7, 1.268262415], [8, 1.270001019]]



ttl := *Value of X3 as a function of the iteration number*

lbl := "X3"

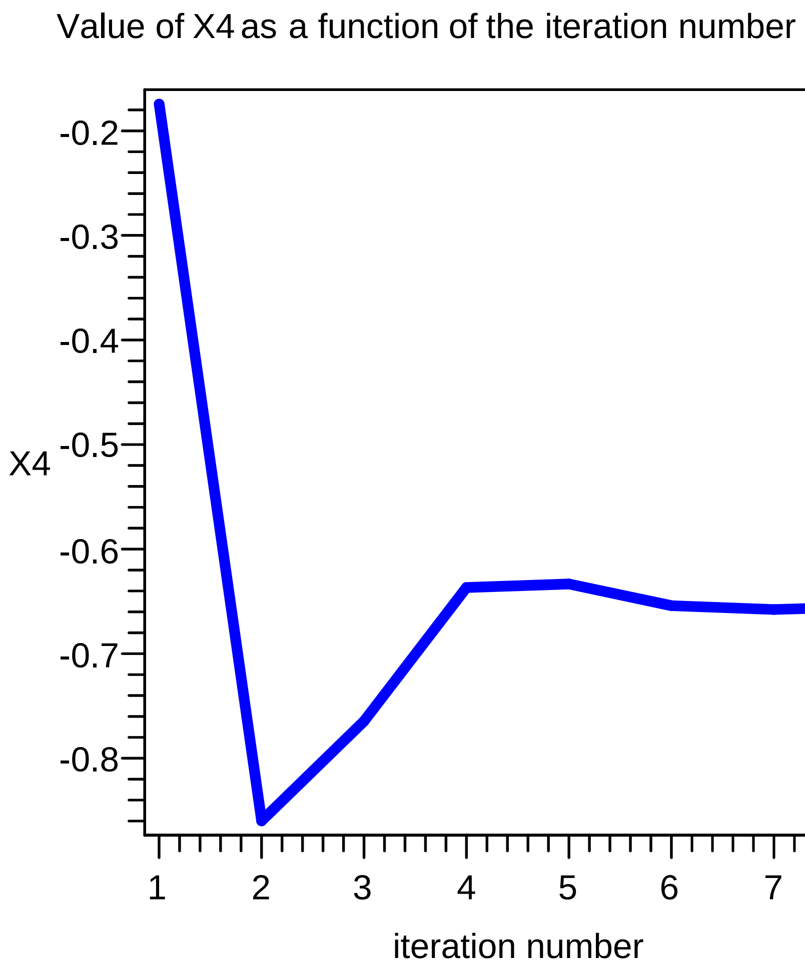
data := [[1, 0.8000000001], [2, 1.065990676], [3, 1.133223260], [4, 1.099396804]
, [5, 1.083167960], [6, 1.086353770], [7, 1.089523450], [8, 1.089468999]]



ttl := *Value of X4 as a function of the iteration number*

lbl := "X4"

data := [[1, −.1743589744], [2, −.8598625308], [3, −.7645649192], [4, −.6366841477], [5, −.6333602431], [6, −.6540842469], [7, −.6577363177], [8, −.6548436792]]

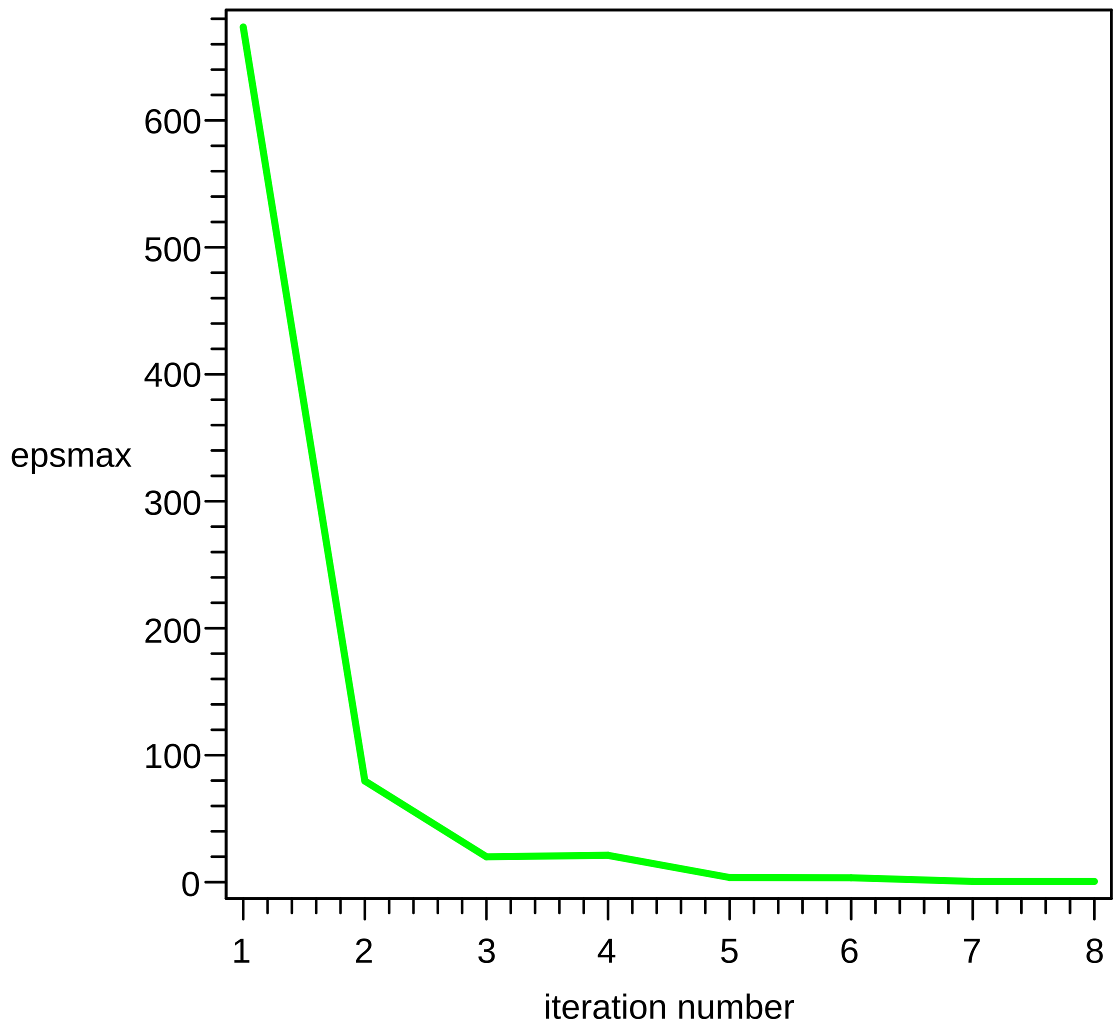


The following graph plots the maximum absolute relative approximate error as a function of the iteration number.

```
> data:=[seq([k,epsmax[k]],k=1..maxit)];
  pointplot(data,connect=true,color=green,axes=boxed,title="Value
  of maximum absolute relative approximate error as a function of
  iteration number",titlefont=[HELVETICA,28], axes=BOXED,labels=
  ["iteration number","epsmax"],thickness=2);
```

data := [[1, 673.5294114], [2, 79.72245933], [3, 19.99065385], [4, 21.13123739]
, [5, 3.621107387], [6, 3.478492376], [7, 0.5552484638], [8, 0.5204169719]]

Value of maximum absolute relative approximate error as a function of iteration number



References

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

Conclusion

Maple 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?