Practical 8: Gauss Seidel method

Gauss Seidel is an improvisation of Gauss Jacobi.

Consider the following system of linear equations:

$a_{11} x_1+ a_{12} x_2+ a_{13} x_3 = b_1$
$a_{21} x_1+ a_{22} x_2+ a_{23} x_3 = b_2$ (I)
$a_{31} x_1+ a_{32} x_2+ a_{33} x_3 = b_3$

If the system is not diagonally dominant or a diagonal element is zero, the rows and columns are interchanged to get a diagonally dominant system with non zero diagonal elements.

Above system can be written as:

$x_1= (b_1-(a_{12} x_2 - a_{13} x_3))/a_{11}$
$x_2= (b_2-(a_{21} x_1 - a_{23} x_3))/a_{22}$
(II)
$x_3= (b_3-(a_{31} x_1 - a_{32} x_2))/a_{33}$

Suppose the true solution of (I) is x = (x1, x2, … , xn). If $x1^{(k+1)}$ is a better approximation to the true value of x1 than $x1^{(k)}$ is, then it would make sense that once we have found the new value $x1^{(k+1)}$ to use it (rather than the old value $x1^{(k)}$) in finding $x2^{(k+1)}$, … , $xn^{(k+1)}$. So $x1^{(k+1)}$ is found as in Jacobi's Method, but in finding $x2^{(k+1)}$, instead of using the old value of $x1^{(k)}$ and the old values $x3^{(k)}$, … , $xn^{(k)}$, we now use the new value $x1^{(k+1)}$ and the old values $x3^{(k)}$, … , $xn^{(k)}$. Like this, we continue finding $x3^{(k+1)}$, instead of using the old value of $x1^{(k)}$, $x2^{(k)}$ and the old values $x4^{(k)}$, … , $xn^{(k)}$, we now use the new value $x1^{(k+1)}$,$x2^{(k+1)}$ and the old values $x4^{(k)}$, … , $xn^{(k)}$.

System (II) can be written using summtion for n variables as:

For i=1,2,..,n, $x_i^{(k+1)}= 1/a_{ii}(b_i-\sum a_{ij} x_j^{(k+1)}-\sum a_{ij} x_j^{(k)})$. For first summation j=1 to i-1, we use $x_j^{(k+1)}$ and for second summation j=i+1 to n, we use $x_j^{(k)}$.

Q1 Perform 8 iterations of Gauss-Seidel method to solve the following system of linear equations with initial approximation x0=[0,0,0]^T :

4x1+x2+x3= 2

x1+5x2+2x3= -6

x1+2x2+3x3= -4

Solution

Method 1

```
kill(all)$
x1:0.0;
x2:0.0;
x3:0.0;
print("itr","","","","","","","","solution")$
for i:1 thru 8 do(
x1:(2-x2-x3)/4,
x2:(-6-x1-2·x3)/5,
x3:(-4-x1-2·x2)/3,
print(i,"","","","x1=",x1," x2=",x2," x3=",x3))$ /* we are not updating
the values of x1,x2, x3 as done in Gauss Jacobi.*/
print("x1=",x1)$
print("x2=",x2)$
print("x3=",x3)$
```

(%o1) 0.0

(%o2) 0.0

(%o3) 0.0

```
itr            solution
1      x1= 0.5   x2= -1.3   x3= -0.6333333333333333
2      x1= 0.9833333333333333   x2= -1.143333333333333   x3=
 -0.898888888888889
3      x1= 1.010555555555556   x2= -1.042555555555555   x3=
 -0.9751481481481482
4      x1= 1.004425925925926   x2= -1.010825925925926   x3=
 -0.994258024691358
5      x1= 1.001270987654321   x2= -1.002550987654321   x3=
 -0.9987230041152264
6      x1= 1.000318497942387   x2= -1.000574497942387   x3=
 -0.9997231673525379
7      x1= 1.000074416323731   x2= -1.000125616323731   x3=
 -0.999941061225423
8      x1= 1.000016669387289   x2= -1.000026909387288   x3=
 -0.9999876168709038
x1= 1.000016669387289
x2= -1.000026909387288
x3= -0.9999876168709038
```

Method 2

```
(%i7) kill(all)$
      'n=n:3;
      'a=a:matrix([4,1,1],[1,5,2],[1,2,3]);
      'x=x:matrix([0],[0],[0]);'b=b:matrix([2],[-6],[-4]);
      print("itr","","","","","solution")$
      for k:1 thru 8 do(
      for i:1 thru n do (
       x[i]:float((b[i]-sum(a[i,j].x[j],j,1,i-1)-
                      sum(a[i,j].x[j],j,i+1,n))/a[i,i])) ,/* In Gauss Jacobi,
         we use y[i]:float.... in the above formula and also update the value
      x[i] using y[i] with the command "for i:1 thru n do (x[i]:y[i])"*/
      print(k,"","","",'x[1]=x[1],'x[2]=x[2],'x[3]=x[3]))$
      for p:1 thru n do print('x[p]=x[p])$
```

(%o1) $n=3$

(%o2) $a=\begin{pmatrix} 4 & 1 & 1 \\ 1 & 5 & 2 \\ 1 & 2 & 3 \end{pmatrix}$

(%o3) $x=\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

(%o4) $b=\begin{pmatrix} 2 \\ -6 \\ -4 \end{pmatrix}$

itr solution

1 $x_1=[0.5]$ $x_2=[-1.3]$ $x_3=[-0.6333333333333333]$

2 $x_1=[0.9833333333333334]$ $x_2=[-1.143333333333333]$ $x_3=[-0.8988888888888888]$

3 $x_1=[1.010555555555555]$ $x_2=[-1.042555555555555]$ $x_3=[-0.9751481481481481]$

4 $x_1=[1.004425925925926]$ $x_2=[-1.010825925925926]$ $x_3=[-0.9942580246913579]$

5 $x_1=[1.001270987654321]$ $x_2=[-1.002550987654321]$ $x_3=[-0.9987230041152262]$

6 $x_1=[1.000318497942387]$ $x_2=[-1.000574497942387]$ $x_3=[-0.9997231673525376]$

7 $x_1=[1.000074416323731]$ $x_2=[-1.000125616323731]$ $x_3=[-0.9999410612254229]$

8 $x_1=[1.000016669387289]$ $x_2=[-1.000026909387288]$ $x_3=[-0.9999876168709039]$

$x_1=[1.000016669387289]$

$x_2=[-1.000026909387288]$

$x_3=[-0.9999876168709039]$

Assignment: Do two similar questions.