Practical 6: Gauss-Jacobi method

Jacobi method is one the iterative methods for approximating the solution of a system of n linear equations in n variables. The Jacobi method is considered as an iterative algorithm which is used for determining the solutions for the system of linear equations which is diagonally dominant.
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

The system is diagonally dominant if
|a_11| > |a_12|+|a_13|,
|a_22| > |a_21|+|a_23|,
|a_33| > |a_31|+|a_32|,
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

Given initial aproximation x^0=(x_1^0,x_2^0, x_3^0), we can substitute in (II) to find x^1=(x_1^1,x_2^1, x_3^1). Again substituting x^1 in (II) we find x^2 and so on......

System (II) can be written using summation for n variables as:
For i=1,2,..,n , x_i^(k+1)= 1/a_ii(b_i-∑a_ij x_j^(k)) and j=1,2,..n, i≠j in the summation.

Q1 Perform 10 iterations of Gauss Jacobi 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

```
(%i8) kill(all)$
      x10=x10:0.0;
      x20=x20:0.0;
      x30=x30:0.0;
      print("itr","    ","","","","x1","   ","   ","    ","    ","x2","
      ","  ","  ","  ","x3")$
      for i:1 thru 10 do(
      x1:(2-x20-x30)/4,
      x2:(-6-x10-2·x30)/5,
      x3:(-4-x10-2·x20)/3,
      print(i," "," "," ",x1,"","","",x2,"","","",x3),
      x10:x1,
      x20:x2,
      x30:x3)$
      print("x1=",x1)$
      print("x2=",x2)$
      print("x3=",x3)$
```

```
(%o1) x10=0.0
(%o2) x20=0.0
(%o3) x30=0.0
      itr          x1                     x2                x3
      1         0.5      -1.2      -1.333333333333333
      2         1.133333333333333      -0.7666666666666667      -
      0.7000000000000001
      3         0.8666666666666667      -1.146666666666667      -
      1.2
      4         1.086666666666667      -0.8933333333333333      -
      0.8577777777777781
      5         0.9377777777777779      -1.074222222222222      -
      1.1
      6         1.043555555555556      -0.9475555555555555      -
      0.9297777777777778
      7         0.9693333333333334      -1.0368      -
      1.049481481481482
      8         1.02157037037037      -0.9740740740740741      -
      0.9652444444444445
      9         0.9848296296296296      -1.018216296296296      -
      1.024474074074074
      10         1.010672592592593      -0.9871762962962963      -
      0.9827990123456791
      x1= 1.010672592592593
      x2= -0.9871762962962963
      x3= -0.9827990123456791
```

Method 2

```
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","    ","","","","x1","   ","   ","    ","
    ","x2","  ","  ","  ","  ","x3")$
for k:1 thru 10 do(
for i:1 thru n do(
y[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])),/*Calculating x[i]'s a
    different iterations*/
for i:1 thru n do (x[i]:y[i]),/*Updating x[i]'s*/
print(k,"","","",x[1],"","","",x[2],"","","",x[3]))$
for p:1 thru n do print('x[p]=x[p])$;/*Printing the x[i]
values in last iteration/*
```

(%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        x1                          x2
x3
1      [0.5]      [-1.2]      [-1.333333333333333]
2      [1.133333333333333]      [-0.7666666666666667]
[-0.7]
3      [0.8666666666666667]      [-1.146666666666667]
[-1.2]
4      [1.086666666666666]      [-0.8933333333333335]
[-0.8577777777777779]
5      [0.9377777777777778]      [-1.074222222222222]
[-1.1]
6      [1.043555555555555]      [-0.9475555555555557]
[-0.9297777777777778]
7      [0.9693333333333334]      [-1.0368]      [-
1.049481481481481]
8      [1.02157037037037]      [-0.974074074074074l]      [
-0.9652444444444443]
9      [0.9848296296296296]      [-1.018216296296296]
[-1.024474074074074]
```

Assignment: Do two similar questions.