Practical 6: Gauss-Jacobi Method

1

Gauss jacobi Method

Figure 1:

The Jacobi Method, Gauss-Seidel Method, and SOR Method

To identify the splittings associated with the Jacobi method, the Gauss-Seidel method, and the SOR method, first express the coefficient matrix A in the form

$$A = D - L - U.$$

Here, D is the diagonal part of A, -L is the strictly lower triangular part of A, and -U is the strictly upper triangular part. It is important to keep in mind that the matrices L and U used here are in no way related to the LU decomposition of the coefficient matrix. As an example, suppose

$$A = \left[\begin{array}{rrrr} 5 & 1 & 2 \\ -3 & 9 & 4 \\ 1 & 2 & -7 \end{array} \right].$$

Then

$$D = \left[\begin{array}{ccc} 5 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & -7 \end{array} \right], \quad L = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 3 & 0 & 0 \\ -1 & -2 & 0 \end{array} \right] \quad \text{and} \quad U = \left[\begin{array}{ccc} 0 & -1 & -2 \\ 0 & 0 & -4 \\ 0 & 0 & 0 \end{array} \right].$$

The Jacobi method is based on the splitting M=D and N=L+U. In order for M to be nonsingular, it must be the case that, for each $i, d_{i,i} \equiv a_{i,i} \neq 0$. If this relationship does not hold for even a single value of i, then the equations in the system must be reordered before the Jacobi method can be applied. With

Figure 2:

the specific choice of splitting indicated above, the iteration scheme for the Jacobi method is defined by

$$\mathbf{x}^{(k+1)} = T_{jac}\mathbf{x}^{(k)} + \mathbf{c}_{jac},$$
 (2)

where

$$T_{jac} = D^{-1}(L+U)$$
 and $\mathbf{c}_{jac} = D^{-1}\mathbf{b}$.

Taking into account the structure of the iteration matrix, T_{jac} , and the vector \mathbf{c}_{jac} , the individual components of equation (2) can be written as

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left[b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k)} - \sum_{j=i+1}^n a_{i,j} x_j^{(k)} \right]. \tag{3}$$

Hence, the Jacobi method is equivalent to solving the *i*-th equation in the system for the unknown x_i .

2

2.1

Figure 3:

Consider the system of equations

The Jacobi method, when applied to this system, will produce the sequence of approximations $\{\mathbf{x}^{(k)}\}$ according to the rules

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{5} \left[10 - x_2^{(k)} - 2x_3^{(k)} \right] \\ x_2^{(k+1)} &= \frac{1}{9} \left[-14 + 3x_1^{(k)} - 4x_3^{(k)} \right] \\ x_3^{(k+1)} &= -\frac{1}{7} \left[-33 - x_1^{(k)} - 2x_2^{(k)} \right] . \end{aligned}$$

2.2

function for Gauss-Jacobi method

(%i1) kill(all);

(%00) done

```
(%i1) gauss jacobi(A, b,
                               iter):=block(
          n:matrix size(A)[1],
          for i:1 thru n do
             for j:1 thru n do
                if(i > j) then (I[i, j]:A[i, j])
                else ( l[i, j]:0)
             )
          ),
          L:genmatrix(I, n, n),
          for i:1 thru n do
          (
             for j:1 thru n do
               if(i < j) then (u[i, j]:A[i, j])
                else ( u[i, j]:0)
          ),
          U:genmatrix(u, n, n),
          for i:1 thru n do
             for j:1 thru n do
               if(i = j) then (d[i, j]:A[i, j])
                else ( d[i, j]:0)
             )
          ),
          D:genmatrix(d, n, n),
          D1:invert(D),
          Tgj:D1.(-L-U),
          cgj:D1.b,
          for i:1 thru n do t[i, 1]:0.0,
          x0:genmatrix(t, n, 1),
          for i:1 thru iter do
          (
             x1:Tgj.x0+cgj
             disp(i, "----", x1),
             x0:x1
          )
(%o1) gauss_jacobi (A, b, iter):=block (n: (matrix_size (A))<sub>1</sub>, for i
        thru n do for j thru n do if i > j then I_{i,j}: A_{i,j} else I_{i,j}: 0, L:
       genmatrix (1, n, n), for i thru n do for j thru n do if i < j then
       u_{i,j}:A_{i,j} else u_{i,j}:0,U:genmatrix(u,n,n),for i thru n do for j
       thru n do if i=j then d_{i,j}:A_{i,j} else d_{i,j}:0, D: genmatrix (d,n,n),
       D1:invert(D), Tgj:D1. (-L-U), cgj:D1. b, for i thru n do t_{i,1}
```

:0.0,x0:genmatrix(t,n,1),for i thru iter do

(%i3) A1:matrix([5, 1, 2], [-3, 9, 4], [1, 2, -7]); b1:matrix([10], [-14], [-33]);

$$(\%02) \begin{pmatrix} 5 & 1 & 2 \\ -3 & 9 & 4 \\ 1 & 2 & -7 \end{pmatrix}$$
$$(\%03) \begin{pmatrix} 10 \\ -14 \\ -33 \end{pmatrix}$$

```
(%i4) gauss_jacobi(A1, b1, 14);
               2.0
        -1.555555555555555
        4.714285714285714
       0.4253968253968252
        -2.984126984126984
        4.5555555555555
       0.7746031746031745
        -3.438447971781305
        3.922448979591836
        1.118710002519526
        -3.040665154950869
        3.842529604434366
        1.071121189216427
        -2.890443156686543
        4.005339956088256
       6
       0.9759526489020063
        -2.97866625074486
        4.041462125120478
        0.9791484001007809
        -3.026443394863988
        4.002660021058898
        1.004224670549238
        -3.008132764881472
```

3.989465944338972

Note:

disp(i, "----", transpose(x1))

can be used for printing as row vectors

2.3

check

Figure 4:

k	$\mathbf{x}^{(k)}$	
0	0.000000 0.00000	$\begin{bmatrix} 0 & 0.000000 \end{bmatrix}^T$
1	[2.000000 -1.5555	$[556 4.714286]^T$
2	[0.425397 -2.9841	$\begin{bmatrix} 27 & 4.555556 \end{bmatrix}^T$
3	[0.774603 -3.4384	$\begin{bmatrix} 448 & 3.922449 \end{bmatrix}^T$
4	[1.118710 -3.0406]	$\begin{bmatrix} 3.842530 \end{bmatrix}^T$
5	$\begin{bmatrix} 1.071121 & -2.8904 \end{bmatrix}$	
6	[0.975953 -2.9786]	
7	[0.979148 -3.0264]	
8	[1.004225 -3.0081]	
9	[1.005840 -2.9939	
10	[0.999470 -2.9972]	
11	[0.998428 -3.0013	
12	[0.999985 -3.0008]	1.
13	$\begin{bmatrix} 1.000408 & -2.9997 \end{bmatrix}$, ,
14	$\begin{bmatrix} 1.000044 & -2.9997 \end{bmatrix}$	$\begin{bmatrix} 57 & 4.000133 \end{bmatrix}^T$

(%i5) kill(all); (%o0) done

```
iter):=block(
(%i1) gauss jacobi(A, b,
                             init,
          n:matrix size(A)[1],
          for i:1 thru n do
             for j:1 thru n do
             (
               if(i > j) then (I[i, j]:A[i, j])
               else ( l[i, j]:0)
             )
          ),
          L:genmatrix(I, n, n),
          for i:1 thru n do
          (
             for j:1 thru n do
               if(i < j) then (u[i, j]:A[i, j])
               else ( u[i, j]:0)
          ),
          U:genmatrix(u, n, n),
          for i:1 thru n do
             for j:1 thru n do
               if(i = j) then (d[i, j]:A[i, j])
               else ( d[i, j]:0)
          ),
          D:genmatrix(d, n, n),
          D1:invert(D),
          Tgj:D1.(-L-U),
          cgj:D1.b,
          x0:init,
          for i:1 thru iter do
             x1:Tgj.x0+cgj,
             disp(i, "----", transpose(x1)),
             x0:x1
          )
       );
(\%o1) gauss_jacobi (A, b, init, iter):=block (n:(matrix_size (A))<sub>1</sub>,
       for i thru n do for j thru n do if i > j then l_{i,j}:A_{i,j} else l_{i,j}:0,L
       genmatrix (l, n, n), for i thru n do for j thru n do if i < j then
       u_{i,j}: A_{i,j} else u_{i,j}: 0, U: genmatrix (u,n,n), for i thru n do for j
       thru n do if i=j then d_{i,j}:A_{i,j} else d_{i,j}:0, D: genmatrix (d,n,n),
       D1:invert(D).Tai:D1:(-L-U).cai:D1:b.x0:init.for i thru
```

```
(%i9) A1:matrix([5, 1, 2], [-3, 9, 4], [1, 2, -7]);
     b1:matrix([10], [-14], [-33]);
     start:matrix([0.0], [0.0], [0.0]);
     gauss_jacobi(A1, b1, start, 14);
(\%06)
      -394
       10
(\%07)
      -14
       -33
      0.0
      0.0
      0.0
      3
      (0.7746031746031745 -3.438447971781305 3.922448979591836)
      (1.118710002519526 - 3.040665154950869 3.842529604434366)
      5
      (1.071121189216427 - 2.890443156686543 4.005339956088256)
      6
      0.9759526489020063 -2.97866625074486 4.041462125120478
      7
      (0.9791484001007809 -3.026443394863988 4.002660021058898)
      8
      (1.004224670549238 - 3.008132764881472 3.989465944338972)
      9
      (1.005840175240705 - 2.993909973967575 3.998279877255185)
      10
```

4

Exercise

Starting with initial vector x0 = 0, perform 6 iterations of Gauss-Jacobi method

Figure 5:

(a)
$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ 5 \end{bmatrix}$$
(b)
$$\begin{bmatrix} 3 & -1 & 1 \\ 2 & -6 & 3 \\ -9 & 7 & -20 \end{bmatrix}, \begin{bmatrix} 4 \\ -13 \\ 7 \end{bmatrix}$$
(c)
$$\begin{bmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$
(d)
$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ 2 & 4 & -1 & 0 \\ 0 & -2 & 4 & -1 \\ 0 & 0 & -2 & 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ -3 \\ 1 \end{bmatrix}$$