Practical 7 : SOR Method

1

1.1 SOR Method

Figure 1:

The final iterative technique that we will discuss in this section is the SOR method. An explanation for the name of the method will be provided shortly. This technique attempts to improve upon the convergence of the Gauss-Seidel method by computing $x_i^{(k+1)}$ as a weighted average of $x_i^{(k)}$ and the value produced by the Gauss-Seidel method, as given in equation (4). Let the weighting parameter, also known as a relaxation parameter, be denoted by ω. Then the analogue of equations (3) and (4) for the SOR method is

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{i,i}}\left[b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}\right]. \qquad (5)$$

Note that when $\omega = 1$, the SOR method reduces to the Gauss-Seidel method. Typically, there exists a range of ω values for which the SOR method will converge faster than the Gauss-Seidel method. The splitting associated with the SOR method is

$$M = \frac{1}{\omega}D - L \quad \text{and} \quad N = \left(\frac{1}{\omega} - 1\right)D + U.$$

Figure 2:

Therefore,

$$T_{sor} = \left(\frac{1}{\omega}D - L\right)^{-1}\left[\left(\frac{1}{\omega} - 1\right)D + U\right]$$

and

$$\mathbf{c}_{sor} = \left(\frac{1}{\omega}D - L\right)^{-1}\mathbf{b}.$$

1.2

function for SOR method

```
→ kill(all);
(%o0) done
```

(%i1)
```
sor(A, b, w, init, iter):=block(
    n:matrix_size(A)[1],

    for i:1 thru n do
    (
        for j:1 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: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),

    D3:invert((1/w)·D+L),
    N:((1/w)−1)·D−U,
    Tsor:D3.N,
    csor:D3.b,

    x0:init,
    for i:1 thru iter do
    (
        x1:Tsor.x0+csor,
        disp(i, "−−−−", x1),
        x0:x1
    )
)
;
```

(%o1) sor $(A,b,w,init,iter)$:=block$(n:(\text{matrix_size}(A))_1$, for i thru n do for j thru n do if $i>j$ then $l_{i,j}:A_{i,j}$ else $l_{i,j}:0, L:$

```
(%i5) A:matrix([5, 1, 2], [−3, 9, 4], [1, 2, −7]);
      b:matrix([10], [−14], [−33]);
      start:matrix([0.0], [0.0], [0.0]);
      sor(A, b, 0.9, start, 10);
```

(%o2) $\begin{pmatrix} 5 & 1 & 2 \\ -3 & 9 & 4 \\ 1 & 2 & -7 \end{pmatrix}$

(%o3) $\begin{pmatrix} 10 \\ -14 \\ -33 \end{pmatrix}$

(%o4) $\begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$

1

$\begin{pmatrix} 1.799999999999999 \\ -0.8599999999999999 \\ 4.253142857142857 \end{pmatrix}$

2

$\begin{pmatrix} 0.6036685714285714 \\ -3.006156571428571 \\ 3.972774269387755 \end{pmatrix}$

3

$\begin{pmatrix} 0.9712763030204081 \\ -2.998342473991836 \\ 3.994010601157784 \end{pmatrix}$

4

$\begin{pmatrix} 0.9989854592037691 \\ -2.997742850101166 \\ 3.999851029130248 \end{pmatrix}$

5

$\begin{pmatrix} 0.9995458884516973 \\ -2.999850930126706 \\ 3.999965049395661 \end{pmatrix}$

6

$\begin{pmatrix} 0.9999403384855388 \end{pmatrix}$

2

Exercise

Starting with initial vector x0 = 0, perform 6 iterations of SOR method with w = 0.9

Figure 3:

$$\text{(a)} \begin{bmatrix} 2 & -1 & 0 \\ -1 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ 3 \\ 5 \end{bmatrix}$$

$$\text{(b)} \begin{bmatrix} 3 & -1 & 1 \\ 2 & -6 & 3 \\ -9 & 7 & -20 \end{bmatrix}, \quad \begin{bmatrix} 4 \\ -13 \\ 7 \end{bmatrix}$$

$$\text{(c)} \begin{bmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 4 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$\text{(d)} \begin{bmatrix} 4 & -1 & 0 & 0 \\ 2 & 4 & -1 & 0 \\ 0 & -2 & 4 & -1 \\ 0 & 0 & -2 & 4 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 2 \\ -3 \\ 1 \end{bmatrix}$$