1 SOR Method

SOR was developed in 1950 by David Young and H. Frankel in 1950 and was developed to be used on digital computers. It was developed by modifying the Gauss-Seidel iteration model. The SOR (Successive Over Relaxation) Method is similar to Jacobi and Gauss Seidel Methods, but it uses a scaling factor to more rapidly reduce the approximation error as compare to Jacobi method and Gauss seidel method. The SOR technique is one of a class of relaxation methods that compute approximation x^k by the formula

$$(x_i)^k = (1-w)(x_i)^{k-1} + \frac{w}{a_{ij}}\left\{ b_i - \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^{n} a_{ij}x_j^{k-1} \right\} \tag{1}$$

Where w is a scaling factor which may have following effects

Case 1: When $\omega = 1$, we have the the Gauss-Seidel method.

Case 2: When $0 < w < 1$, then procedure is called underrelaxation method and can be used to obtain convergence of some system that are not convergent by the Gauss-seidel method.

Case 3: When $w > 1$, then procedures is called over-relaxation methods, which are used to accelerate the convergence for system that are convergent by the Gauss-Seidel methods techniques.

Case 4: When $w = 0$, thne there will be no iteration.

Case 5: When $w > 2$, then system will be divergent.

These methods are called SOR (Successive over relaxation) and are used for solving linear systems that occur in the numerical solutions of certain partial differential equations.

Example 1.1. *Consider a linear system $AX = B$ where*

$$A = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} -1 \\ 7 \\ 7 \end{bmatrix}$$

values of the relaxation parameter $w = 1.25$.

Solution. *Let us verify the sufficient conditions for using the SOR method .We have to check if matrix A is symmetric, positive definite.*

$$A_1 = [3], \quad det(A_1) = 3 > 0$$

$$A_2 = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}, \quad det(A_2) = 8 > 0$$

$$det(A_3) = \begin{vmatrix} 3 & -1 & 1 \\ -1 & 3 & -1 \\ 1 & -1 & 3 \end{vmatrix} = 20 > 0$$

Since determinants of leading principal sub matrices of matrix A, are positive. Therefore, matrix A is positive definite. Thus, SOR method converges for values of the relaxation parameter w within the interval $[0 < w < 2]$.

Step I

Write the system of equations

$$3x_1 - x_2 + x_3 = -1$$
$$-x_1 + 3x_2 - x_3 = 7$$
$$x_1 - x_2 + 3x_3 = -7$$

Step II

Write down the system of equations as for Gauss-seidel iterations

$$x_1^{k+1} = \frac{1}{3}\left(-1 + x_2^k - x_3^k\right)$$
$$x_2^{k+1} = \frac{1}{3}\left(7 + x_1^{k+1} + x_3^k\right)$$
$$x_3^{k+1} = \frac{1}{3}\left(-7 - x_1^{k+1} + x_2^{k+1}\right)$$

Step III

Multiplying the right hand side by the parameter 'w' and add to it the vector x^k *from the previous iteration multiply by the factor* $(1-w)$.

$$x_1^{k+1} = (1-w)x_1^k + w\left\{\frac{1}{3}\left(-1 + x_2^k - x_3^k\right)\right\}$$
$$x_2^{k+1} = (1-w)x_2^k + w\left\{\frac{1}{3}\left(7 + x_1^{k+1} + x_3^k\right)\right\}$$
$$x_3^{k+1} = (1-w)x_3^k + w\left\{\frac{1}{3}\left(-7 - x_1^{k+1} + x_2^{k+1}\right)\right\}$$

Step IV *Computing First Iteration, set* $k = 0$*, then system will be written as*

For, $k = 0$

$$\begin{aligned} x_1^1 &= (1-w)x_1^0 + w\left\{\frac{1}{3}\left(-1 + x_2^0 - x_3^0\right)\right\} \\ &= (1 - 1.25)(0) + 1.25\left\{\frac{1}{3}(-1 + 0 - 0)\right\} \\ &= -0.41667 \end{aligned}$$

$$\begin{aligned} x_2^1 &= (1-w)x_2^0 + w\left\{\frac{1}{3}\left(7 + x_1^1 + x_3^0\right)\right\} \\ &= (-0.25)(0) + 1.25\left\{\frac{1}{3}(7 - 0.41667 + 2.7131)\right\} \\ &= -1.6001 \end{aligned}$$

$$
\begin{aligned}
x_3^1 &= (1-w)x_3^0 + w\left\{\frac{1}{3}\left(-7 - x_1^1 + x_2^1\right)\right\} \\
&= (-0.25)(0) + 1.25\left\{\frac{1}{3}(-7 + 0.41667 + 2.7131)\right\} \\
&= -1.6001
\end{aligned}
$$

Solution after 1st iteration

$$x_1 = (-0.41667, 2.7431, -1.600)^T$$

Computing Second Iteration, set $k = 1$, then system will be written as

$$
\begin{aligned}
x_1^2 &= (1-w)x_1^1 + w\left\{\frac{1}{3}\left(-1 + x_2^1 - x_3^1\right)\right\} \\
&= (1-1.25)(-0.41667) + 1.25\left\{\frac{1}{3}(-1 + 2.7431 + 1.6001)\right\} \\
&= 0.10416 + 1.393 = 1.49716 \\
x_2^2 &= (1-w)x_2^1 + w\left\{\frac{1}{3}\left(7 + x_1^2 + x_3^1\right)\right\} \\
&= (1-1.25)(2.7431) + 1.25\left\{\frac{1}{3}(7 + 1.49716 - 1.6)\right\} \\
&= 2.188075 \\
x_3^2 &= (1-w)x_3^1 + w\left\{\frac{1}{3}\left(-7 - x_1^2 + x_2^2\right)\right\} = -2.2288
\end{aligned}
$$

Third Iteration

$$
\begin{aligned}
x_1^3 &= (1-1.25) * 1.4972 + 1.25 \times \frac{1}{3}\{-1 + (2.188) - (-2.2288)\} \\
&= (-0.25) * 1.4972 + 1.25 \times \frac{1}{3}\{3.4168\} = -0.3743 + 1.4237 = 1.0494 \\
x_2^3 &= (1-1.25) * 2.188 + 1.25 \times \frac{1}{3}\{7 + (1.0494) + (-2.2288)\} \\
&= (-0.25) * 2.188 + 1.25 \times \frac{1}{3}\{5.8206\} = -0.547 + 2.4252 = 1.8782 \\
x_3^3 &= (1-1.25) * -2.2288 + 1.25 \times \frac{1}{3}\{-7 - (1.0494) + (1.8782)\} \\
&= (-0.25) * -2.2288 + 1.25 \times \frac{1}{3}\{-6.1711\} = 0.5572 + -2.5713 = -2.0141
\end{aligned}
$$

Fourth Iteration

$$x_1^4 = (1-1.25)*1.0494 + 1.25 \times \frac{1}{3}\{-1+(1.8782)-(-2.0141)\}$$
$$= (-0.25)*1.0494 + 1.25 \times \frac{1}{3}\{2.8924\} = -0.2623 + 1.2051 = 0.9428$$
$$x_2^4 = (1-1.25)*1.8782 + 1.25 \times \frac{1}{3}\{7+(0.9428)+(-2.0141)\}$$
$$= (-0.25)*1.8782 + 1.25 \times \frac{1}{3}\{5.9287\} = -0.4696 + 2.4703 = 2.0007$$
$$x_3^4 = (1-1.25)*-2.0141 + 1.25 \times \frac{1}{3}\{-7-(0.9428)+(2.0007)\}$$
$$= (-0.25)*-2.0141 + 1.25 \times \frac{1}{3}\{-5.9421\} = 0.5035 + -2.4759 = -1.9723$$

Fifth Iteration

$$x_1^5 = (1-1.25)*0.9428 + 1.25 \times \frac{1}{3}\{-1+(2.0007)-(-1.9723)\}$$
$$= (-0.25)*0.9428 + 1.25 \times \frac{1}{3}\{2.9731\}$$
$$= -0.2357 + 1.2388 = 1.0031$$
$$x_2^5 = (1-1.25)*2.0007 + 1.25 \times \frac{1}{3}\{7+(1.0031)+(-1.9723)\}$$
$$= (-0.25)*2.0007 + 1.25 \times \frac{1}{3}\{6.0307\}$$
$$= -0.5002 + 2.5128 = 2.0126$$
$$x_3^5 = (1-1.25)*-1.9723 + 1.25 \times \frac{1}{3}\{-7-(1.0031)+(2.0126)\}$$
$$= (-0.25)*-1.9723 + 1.25 \times \frac{1}{3}\{-5.9905\}$$
$$= 0.4931 + -2.496 = -2.0029$$

Sixth Iteration

$$x_1^6 = (1-1.25)*1.0031 + 1.25 \times \frac{1}{3}\{-1+(2.0126)-(-2.0029)\}$$
$$= (-0.25)*1.0031 + 1.25 \times \frac{1}{3}\{3.0156\}$$
$$= -0.2508 + 1.2565 \qquad = 1.0057$$
$$x_2^6 = (1-1.25)*2.0126 + 1.25 \times \frac{1}{3}\{7+(1.0057)+(-2.0029)\}$$
$$= (-0.25)*2.0126 + 1.25 \times \frac{1}{3}\{6.0028\}$$
$$= -0.5032 + 2.5012 \qquad = 1.998$$
$$x_3^6 = (1-1.25)*-2.0029 + 1.25 \times \frac{1}{3}\{-7-(1.0057)+(1.998)\}$$
$$= (-0.25)*-2.0029 + 1.25 \times \frac{1}{3}\{-6.0077\}$$
$$= 0.5007 + -2.5032 \qquad = -2.0025$$

Seventh Iteration

$$x_1^7 = (1-1.25)*1.0057 + 1.25 \times \frac{1}{3}\{-1+(1.998)-(-2.0025)\}$$
$$= (-0.25)*1.0057 + 1.25 \times \frac{1}{3}\{3.0005\}$$
$$= -0.2514 + 1.2502 \qquad = 0.9988$$
$$x_2^7 = (1-1.25)*1.998 + 1.25 \times \frac{1}{3}\{7+(0.9988)+(-2.0025)\}$$
$$= (-0.25)*1.998 + 1.25 \times \frac{1}{3}\{5.9963\}$$
$$= -0.4995 + 2.4985 \qquad = 1.999$$
$$x_3^7 = (1-1.25)*-2.0025 + 1.25 \times \frac{1}{3}\{-7-(0.9988)+(1.999)\}$$
$$= (-0.25)*-2.0025 + 1.25 \times \frac{1}{3}\{-5.9998\}$$
$$= 0.5006 + -2.4999 \qquad = -1.9993$$

Eighth Iteration

$$
\begin{aligned}
x_1^8 &= (1-1.25)*0.9988 + 1.25 \times \frac{1}{3}\{-1+(1.999)-(-1.9993)\} \\
&= (-0.25)*0.9988 + 1.25 \times \frac{1}{3}\{2.9983\} \\
&= -0.2497 + 1.2493 && = 0.9996 \\
x_2^8 &= (1-1.25)*1.999 + 1.25 \times \frac{1}{3}\{7+(0.9996)+(-1.9993)\} \\
&= (-0.25)*1.999 + 1.25 \times \frac{1}{3}\{6.0003\} \\
&= -0.4997 + 2.5001 && = 2.0004 \\
x_3^8 &= (1-1.25)*-1.9993 + 1.25 \times \frac{1}{3}\{-7-(0.9996)+(2.0004)\} \\
&= (-0.25)*-1.9993 + 1.25 \times \frac{1}{3}\{-5.9992\} \\
&= 0.4998 + -2.4997 && = -1.9998
\end{aligned}
$$

Ninth Iteration

$$
\begin{aligned}
x_1^9 &= (1-1.25)*0.9996 + 1.25 \times \frac{1}{3}\{-1+(2.0004)-(-1.9998)\} \\
&= (-0.25)*0.9996 + 1.25 \times \frac{1}{3}\{3.0002\} \\
&= -0.2499 + 1.2501 && = 1.0002 \\
x_2^9 &= (1-1.25)*2.0004 + 1.25 \times \frac{1}{3}\{7+(1.0002)+(-1.9998)\} \\
&= (-0.25)*2.0004 + 1.25 \times \frac{1}{3}\{6.0004\} \\
&= -0.5001 + 2.5001 && = 2.0001 \\
x_3^9 &= (1-1.25)*-1.9998 + 1.25 \times \frac{1}{3}\{-7-(1.0002)+(2.0001)\} \\
&= (-0.25)*-1.9998 + 1.25 \times \frac{1}{3}\{-6.0001\} \\
&= 0.5 + -2.5001 && = -2.0001
\end{aligned}
$$

Example 1.2. *The linear system $AX = B$ of equations given by*

$$
\begin{aligned}
4x_1 + 3x_2 &= 24 \\
3x_1 + 4x_2 - x_3 &= 30 \\
-x_2 + 4x_3 &= -24
\end{aligned}
$$

has the solution $(3,4,-5)^t$. Compare the iterations from the Gauss-seidel method and the SOR method with $w = 1.25$ using $x^0 = (1,1,1)^t$ for both methods

Solution. *For each $k = 1,\ 2,\ 3,\ 4, \cdots$ the equations for the Gauss-Seidel methods are*

$$\begin{aligned} x_1^k &= -0.75x_2^{k-1} + 6 \\ x_2^k &= -0.75x_1^k + 0.25_3^{k-1} + 7.5 \\ x_3^k &= -0.25x_2^{k-1} - 6 \end{aligned}$$

and the equations for the SOR method with $w = 1.25$ are

$$\begin{aligned} x_1^k &= -0.25x_1^{k-1} - 0.9375x_2^{k-1} + 7.5 \\ x_2^k &= -0.9375x_1^k - 0.25x_2^{k-1} + 0.3125x_3^{k-1} + 9.375 \\ x_3^k &= 0.3125x_2^k - 0.25x_3^{k-1} - 7.5 \end{aligned}$$

The first seven iteration for each method are listed below (see Table 1). For each iteration to be accurate to seven decimal places, the Gauss-Seidel method requires 34 *iterations, opposed to* 14 *iterations for the SOR method with $\omega = 1.25$*

k	0	1	2	3	4	5	6	7
x_1^k	1	5.2500	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.013144
x_2^k	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.988824
x_3^k	1	−5.046875	−5.0292969	−5.0183105	−5.0114441	−5.0071526	−5.0044703	−5.002794

k	0	1	2	3	4	5	6	7
x_1^k	1	6.3125	2.6223145	3.1333027	2.9570512	3.3720011	2.9963276	3.000049
x_2^k	1	3.5195313	3.9585266	4.0102646	4.0074838	4.002925	4.0009262	4.000258
x_3^k	1	−6.6501465	−4.6004238	−5.0966863	−4.9734897	−5.0057135	−4.9982822	−5.000348

Table 1: Comparison

Theorem 1.1 (Kahan)**.** *A necessary condition for the SOR method to converge is $|\omega - 1| < 1$ (For $\omega \in \mathbb{R}$ this condition becomes $\omega \in (0,2)$)*

1.1 Applications

Successive over-relaxation (SOR) is one of the most important method for solution of large linear systems. It has applications in CFD (computational fluid dynamics), mathematical programming, medical analysis and machine learning etc. The example of applications of SOR in CFD include study of steady heat conduction, turbulent flows, boundary layer flows or chemically reacting flows. For this reason, SOR method is important for both researchers and business policymakers.