SOR and SSOR

Wang Zhibin June 6, 2012

Successive Over Relaxation

Description of Algorithm

The successive over relaxation method is an iteration method used for finding the solution of elliptic differential equations. Successive over relaxation has been devised to accelerate the convergence of **Gauss-Seidel** and **Jacobi** by introducing a new parameter, w, referred to as the relaxation factor.

Giving the linear system of equations

$$Ax = b.$$

The matrix A can be written as

$$A = D + U + L.$$

Where $D, U,$ and L denote the diagonal, strictly upper triangular and strictly lower triangular parts of A.

Using the successive over relaxation technique, the solution of the partial differential equation (PDE) is obtained using

$$x^{(k)} = (D + wL)^{-1}\{wb - [wU + (w - 1)D]x^{(k-1)}\}.$$

Where $x^{(k)}$ represents the k th iterate.

The SOR rate of convergence strongly depends on the choice of the relaxation factor w. Extensive work has been done on finding a good estimate of this factor in the [0, 2] interval.

Recent studies have shown that for the case where

- ✧ $w = 1$, SOR simplifies to Gauss-Seidel method.
- ✧ $w \le 0$ or $w \ge 2$, SOR fails to converge.
- ✧ $w \succ 1$, SOR used to speed up convergence of a slowly converging process.
- ✧ $w \prec 1$, SOR helps to establish convergence of diverging iterative process.

Demonstrate of Algorithm

Original Model: the Splitting-Based Iterative Methods.

- ✧ Split the system matrix A in two parts. $A = A_1 + A_2$.

$$Ax = b \Rightarrow (A_1 + A_2)x = b \Rightarrow A_1 x = b - A_2 x$$

- ✧ Introduce a fixed point iteration of type.

$$A_1 x^{k+1} = b - A_2 x^k \Rightarrow x^{k+1} = A^{-1}(b - A_2 x^k)$$

- ✧ This is the core idea of the so-called Gauss-Seidel and SOR.

Forward substitution:

- ✧ A is a lower-triangular system with form given below .The diagonal elements are nonzero.

$$a_{1,1}x_1 = b_1$$
$$a_{2,1}x_1 + a_{2,2}x_2 = b_2$$
$$a_{3,1}x_1 + a_{3,2}x_2 + a_{3,3}x_3 = b_3$$
$$\ldots$$
$$a_{n,1}x_1 + a_{n,2}x_2 + a_{n,3}x_3 + \ldots + a_{n,n}x_n = b_n$$

✧ First compute

$$x_1 = \frac{b_1}{a_{1,1}}$$

✧ And then use the rule

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{i,j}}{a_{i,i}} (i = 1,2,\ldots n)$$

Basic Idea: **Gauss-Seidel**

✧ Spilt the system matrix.

$$A = (\underbrace{D + L}_{A_1} + \underbrace{U}_{A_2}) \Rightarrow (D + L)x = b - Ux$$

✧ Introduce a fixed point iteration of type.

$$(D + L)x^{k+1} = b - Ux^k \Rightarrow x^{k+1} = (D + L)^{-1}(b - Ux^k)$$

✧ Use forward substitution and get the point notation.

$$\tilde{x}_i^{(k+1)} = \frac{[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}]}{a_{ii}}$$

Improved idea: **SOR**

✧ Add the relaxation factor w.

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega\tilde{x}_i^{(k+1)}$$

✧ So we can get the SOR iteration.

$$x_i^{(k+1)} = (1 - w)x_i^k + \frac{w}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}] \; (i = 1,2,\ldots,n)$$

Symmetric Successive Over Relaxation

Description of Algorithm

Symmetric successive over relaxation combines two SOR sweeps together in such a way that resulting iteration matrix is similar to a symmetric matrix. So SSOR is a forward SOR sweep followed by a backward SOR sweep.

Giving the linear system of equations

$$Ax = b.$$

The matrix A can be written as

$$A = D + U + L.$$

Where $D, U,$ and L denote the diagonal, strictly upper triangular and strictly lower triangular parts of A.

Using the successive over relaxation technique, the solution of the partial differential equation (PDE) is obtained using

$$x^{(k)} = (D + wL)^{-1}\{wb - [wU + (w - 1)D]x^{(k-1)}\}$$

Demonstrate of Algorithm

Forward SOR Sweep: we obtain $x^{(k+\frac{1}{2})}$ as follows,

$$x^{(k+\frac{1}{2})} = (D + wL)^{-1}(-wU + (1 - w)D)x^{(k)} + w(D + wL)^{-1}b$$

Or

$$x^{(k+\frac{1}{2})} = B_2 x^{(k)} + C_2$$

Where

$$B_2 = (D + wL)^{-1}(-wU + (1 - w)D)$$
$$B_2 = (D + wL)^{-1}(D + wL - wL - wU - wD) = 1 - w(D + wL)^{-1}A$$

And

$$C_2 = w(D + wL)^{-1}b$$

Backward SOR Sweep: we compute $x^{(k+1)}$ from $x^{(k+\frac{1}{2})}$ as follows,

$$x^{(k+1)} = B_1 x^{(k+\frac{1}{2})} + C_1$$

Where

$$B_1 = (D + wU)^{-1}(-wL + (1 - w)D)$$
$$B_1 = (D + wU)^{-1}(D + wU - wL - wU - wD) = 1 - w(D + wU)^{-1}A$$

And

$$C_1 = w(D + wU)^{-1}b$$

Then, we delete $x^{(k+\frac{1}{2})}$ and obtain the SSOR as follows:

$$x^{(k+1)} = B_1\left(B_2 x^{(k)} + C_2\right) + C_1 = B_1 B_2 x^{(k)} + B_1 C_2 + C_1$$

Where

$$B_1 C_2 + C_1 = (D + wU)^{-1}(-wL + (1 - w)D)w(D + wL)^{-1}b + w(D + wU)^{-1}b$$
$$= w(D + wU)^{-1}\left((-wL + (1 - w)D)(D + wL)^{-1} + 1\right)b$$
$$= w(D + wU)^{-1}(-wL - wD + D + D + wL)(D + wL)^{-1}b$$
$$= w(2 - w)(D + wU)^{-1}D(D + wL)^{-1}b$$

So we can obtain SSOR matrix

$$x^{(k+1)} = B_1 B_2 x^{(k)} + w(2 - w)(D + wU)^{-1}D(D + wL)^{-1}b$$

Where

$$B_1 = (D + wU)^{-1}(-wL + (1 - w)D);\ \text{Backward SOR Sweep}$$
$$B_2 = (D + wL)^{-1}(-wU + (1 - w)D);\ \text{Forward SOR Sweep}$$

So we can obtain the iterative methods of SSOR in component form.

Forward SOR Sweep: (i=0; i < n; i++)

$$x_i^{(k+\frac{1}{2})} = (1 - w)x_i^{(k)} + \frac{w}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+\frac{1}{2})} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})$$

Backward SOR Sweep: (i= n; i > 0; i--)

$$x_i^{(k+1)} = (1 - w)x_i^{(k+\frac{1}{2})} + \frac{w}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+\frac{1}{2})} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k+1)})$$