Convergence Analysis of Iterative Methods

We want to answer two questions:

1. When will the Jacobi or Gauss-Seidel Methods work? That is, under what conditions will they produce a sequence of approximations $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)}, \ldots$ which converge to the true solution?
2. When they do work, how fast will the convergence to the true solution be? That is, what will the *rate of convergence* be?

In general, an iterative method that finds the solution to $A\boldsymbol{x} = \boldsymbol{b}$ takes the form

$$M\boldsymbol{x}^{(k+1)} = N\boldsymbol{x}^{(k)} + \boldsymbol{b},$$

so that

$$\boldsymbol{x}^{(k+1)} = M^{-1}N\boldsymbol{x}^{(k)} + M^{-1}\boldsymbol{b},$$

which we can rewrite as

$$\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + \widetilde{\boldsymbol{b}},$$

where $B = M^{-1}N$ and $\widetilde{\boldsymbol{b}} = M^{-1}\boldsymbol{b}$.

If the current approximation $\boldsymbol{x}^{(k)}$ is, in fact, the exact solution $\boldsymbol{x}$, then the iterative method should certainly find that the next iteration $\mathbf{x}^{(k+1)}$ is also the exact solution $\mathbf{x}$. That is, it should be that $\boldsymbol{x} = B\boldsymbol{x} + \widetilde{\boldsymbol{b}}$, so that

$$\boldsymbol{x} = B\boldsymbol{x} + \widetilde{\boldsymbol{b}} \quad \Rightarrow \quad M\boldsymbol{x} = N\boldsymbol{x} + \boldsymbol{b} \quad \Rightarrow \quad (M - N)\boldsymbol{x} = \boldsymbol{b}.$$

Of course since the original problem we are trying to solve is $A\boldsymbol{x} = \boldsymbol{b}$, it must be that M and N are chosen so that $A = M - N$. On the other hand, it turns out that choosing $A = M - N$ does not necessarily guarantee that the iterative method will find a sequence of vectors $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)}, \ldots$ that converges to the true solution $\boldsymbol{x}$.

Whether or not a particular method will work will depend on the matrix $B = M^{-1}N$. In fact, it turns out that in general the matrix B *completely* determines the convergence (or not) of an iterative method. In particular, the initial guess generally has no effect on whether or not a particular method is convergent or on the rate of convergence, although if the initial guess is far away from the true solution, more iterations will be required to get an acceptable approximation for the true solution than if the initial guess were closer to the true solution.

To understand the convergence properties of an iterative method $\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + \widetilde{\boldsymbol{b}}$, we subtract the equation $\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + \widetilde{\boldsymbol{b}}$ from the equation $\boldsymbol{x} = B\boldsymbol{x} + \widetilde{\boldsymbol{b}}$, which gives us $\boldsymbol{x} - \boldsymbol{x}^{(k+1)} = B(\boldsymbol{x} - \boldsymbol{x}^{(k)})$. That is, where the current error is $\boldsymbol{e}^{(k)} = \boldsymbol{x} - \boldsymbol{x}^{(k)}$, we have

$$\boldsymbol{e}^{(k)} = B\boldsymbol{e}^{(k-1)} = B(B\boldsymbol{e}^{(k-2)}) = B^2\boldsymbol{e}^{(k-2)} = \cdots = B^k\boldsymbol{e}^{(0)}.$$

To be clear, the superscript of matrix B is the power of B, while the superscript of vector $\boldsymbol{e}$ (inside parentheses) is the iteration number to which this particular error corresponds. We want $\boldsymbol{e}^{(k)} \to \boldsymbol{0}$ as $k \to \infty$. Since $\|\boldsymbol{e}^{(k)}\| = \|B^k\boldsymbol{e}_0\| \le \|B\|^k \|\boldsymbol{e}_0\|$, then we will have $\|\boldsymbol{e}^{(k)}\| \to 0$

(which is the same as $\boldsymbol{e}^{(k)} \to \boldsymbol{0}$) if $\|B\| < 1$. Just like the norm of a vector, the norm $\|B\|$ of matrix B tells us the "size" of the matrix, or more precisely, how much bigger or smaller $B\boldsymbol{v}$ will be compare to $\boldsymbol{v}$. As we show below, there is a particular B matrix for each of the Jacobi and Gauss-Seidel Methods. For each method, the smaller $\|B\|$ is, the faster the method will converge, or if $\|B\| \geq 1$, neither method will normally converge.

One condition sometimes encountered in practice that will guarantee that $\|B\| < 1$ is that matrix A is *strictly diagonally dominant*. A matrix is strictly diagonally dominant if for each of its rows, the absolute value of the *diagonal* element is larger than the sum of the absolute values of the *off*-diagonal elements. That is,

$$\text{where} \quad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad \text{we have} \quad \begin{array}{lll} |a_{11}| & > & |a_{12}|+|a_{13}|+\cdots+|a_{1n}| \\ |a_{22}| & > & |a_{21}|+|a_{23}|+\cdots+|a_{2n}| \\ \vdots & & \vdots \\ |a_{nn}| & > & |a_{n1}|+|a_{n2}|+\cdots+|a_{n\,n-1}| \end{array}.$$

Example 3 Matrix A_1 is diagonally dominant and matrix A_2 is not:

$$A_1 = \begin{bmatrix} 4 & 1 & 2 \\ 2 & -5 & -2 \\ -1 & 2 & 7 \end{bmatrix} \begin{array}{l} |4| > |1|+|2| \\ |-5| > |2|+|\text{-}2| \\ |7| > |-1|+|2| \end{array} \qquad A_2 = \begin{bmatrix} 4 & 2 & 2 \\ 4 & -5 & -2 \\ -1 & 2 & 7 \end{bmatrix} \begin{array}{l} |4| = |2|+|2| \\ |-5| < |4|+|\text{-}2| \\ |7| > |-1|+|2| \end{array}$$

Analysis of Jacobi and Gauss-Seidel Methods for 2 x 2 Systems

We will continue our discussion with only the 2 x 2 case, as the Java applet to be used in doing the homework exercises deals with this case. As discussed earlier, the Jacobi and Gauss-Seidel Methods are both of the form $\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + \tilde{\boldsymbol{b}}$, where their B matrices for the 2 x 2 case are

Method	B
Jacobi	$D^{-1}(-L-U) = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} 0 & -a_{12} \\ -a_{21} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 \end{bmatrix}$
Gauss-Seidel	$(L+D)^{-1}(-U) = \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} 0 & -a_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} \\ 0 & \frac{a_{12}\,a_{21}}{a_{11}\,a_{22}} \end{bmatrix}$

Notice for both methods that the diagonal elements of A must be non-zero, $a_{11} = 0$ and $a_{22} = 0$. It turns out that if n x n matrix B has a full set of n distinct eigenvectors (which is always the case for 2 x 2 systems using Jacobi or Gauss-Seidel Methods), then $\| B \| = |\lambda_{\max}|$, where $\lambda_{\max}$ is the eigenvalue of matrix B with largest magnitude. Consequently, in the 2 x 2 case, the Jacobi and Gauss-Seidel Methods are guaranteed to converge if all of the eigenvalues of the matrix B corresponding to that method are of magnitude < 1. This also includes the case that B has complex eigenvalues. We note that for n x n systems, things are more complicated.

We have now answered the first question posed at the beginning of this section. Because $\| \boldsymbol{e}^{(k)} \| \leq \| B \|^k \| \boldsymbol{e}_0 \|$, the second question is also answered. For example, if $\| B \| = 0.5$, then the error $\boldsymbol{e}^{(k)} = \boldsymbol{x} - \boldsymbol{x}^{(k)}$ will be cut approximately in half by each additional iteration. The size $\| B \|$ is directly proportional to the size of the eigenvalues of B. Consequently, a major goal in designing an iterative method is that the corresponding B matrix has eigenvalues that are as small as possible.

The eigenvalues and corresponding eigenvectors for the Jacobi and Gauss-Seidel Methods are

Method	Eigenvalues	Eigenvectors
Jacobi	$\lambda_1 = \sqrt{\frac{a_{12}\, a_{21}}{a_{11}\, a_{22}}}, \quad \lambda_2 = -\sqrt{\frac{a_{12}\, a_{21}}{a_{11}\, a_{22}}}$	$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ -\sqrt{\frac{a_{11}\, a_{21}}{a_{12}\, a_{22}}} \end{bmatrix}, \quad \boldsymbol{v}_2 = \begin{bmatrix} 1 \\ \sqrt{\frac{a_{11}\, a_{21}}{a_{12}\, a_{22}}} \end{bmatrix}$
Gauss-Seidel	$\lambda_1 = 0, \quad \lambda_2 = -\frac{a_{12}\, a_{21}}{a_{11}\, a_{22}}$	$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{v}_2 = \begin{bmatrix} 1 \\ -\frac{a_{21}}{a_{22}} \end{bmatrix}$

Notice for both methods that we will have $\| B \| = |\lambda_{\max}| < 1$ if $\left| \frac{a_{12}\, a_{21}}{a_{11}\, a_{22}} \right| < 1$. Also notice that the non-zero eigenvalue for the Gauss-Seidel Method is the square of the two eigenvalues for the Jacobi Method. As a result, when dealing with 2 x 2 systems, $\| B_{GS} \| = \| B_{Jacobi} \|^2$, which means that the rate of convergence of the Gauss-Seidel Method is the square of the rate of convergence of the Jacobi Method. For example, if each iteration of the Jacobi Method causes the error to be halved, each iteration of the Gauss-Seidel Method will cause the error to be quartered, since $\frac{1}{4} = \left(\frac{1}{2}\right)^2$. Another way to look at this is that, for 2 x 2 systems, approximately twice as many iterations of the Jacobi Method iterations are needed to achieve the same level of accuracy (to the true solution $\boldsymbol{x}$) as the Gauss-Seidel Method. We note that for general n x n systems, things are more complicated than (but similar to) the 2 x 2 case that we have been discussing.

SOR Method

A third iterative method, called the Successive Overrelaxation (SOR) Method, is an improvement on the Gauss-Seidel Method. The basic idea is this: in finding $\boldsymbol{x}^{(k+1)}$ given $\boldsymbol{x}^{(k)}$, we move a certain amount in a particular direction from $\boldsymbol{x}^{(k)}$ to $\boldsymbol{x}^{(k+1)}$. This direction is simply the vector $\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$, since $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})$. Assuming that the direction from $\boldsymbol{x}^{(k)}$ to $\boldsymbol{x}^{(k+1)}$ is taking us closer, but not all the way, to the true solution $\boldsymbol{x}$, the basic idea of the SOR Method then is to move in the same direction $\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$, but farther along that direction.

To derive the SOR Method from the Gauss-Seidel Method, notice that we could also write the Gauss-Seidel equation as

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

so that

$$\boldsymbol{x}^{(k+1)} = D^{-1}[\boldsymbol{b} - L\boldsymbol{x}^{(k+1)} - U\boldsymbol{x}^{(k)}].$$

We can subtract $\boldsymbol{x}^{(k)}$ from both sides to get

$$\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} = D^{-1}[\boldsymbol{b} - L\boldsymbol{x}^{(k+1)} - D\boldsymbol{x}^{(k)} - U\boldsymbol{x}^{(k)}].$$

One can think of this as the *Gauss-Seidel correction* $(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})_{GS}$. As suggested above, it turns out that convergence $\boldsymbol{x}^{(k)} \to \boldsymbol{x}$ is often faster if we go beyond the Gauss-Seidel correction. The idea of the SOR Method is to iterate

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})_{GS}$$

where

$$(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})_{GS} = D^{-1}[\boldsymbol{b} - L\boldsymbol{x}^{(k+1)} - D\boldsymbol{x}^{(k)} - U\boldsymbol{x}^{(k)}],$$

and where generally $1 < \omega < 2$. Notice that if $\omega = 1$, then we simply have the Gauss-Seidel Method. Written out in detail, the SOR Method is

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega D^{-1}[\boldsymbol{b} - L\boldsymbol{x}^{(k+1)} - D\boldsymbol{x}^{(k)} - U\boldsymbol{x}^{(k)}].$$

We can multiply both sides by matrix D and divide both sides by ω, and rewrite this as

$$\tfrac{1}{\omega} D\boldsymbol{x}^{(k+1)} = \tfrac{1}{\omega} D\boldsymbol{x}^{(k)} + [\boldsymbol{b} - L\boldsymbol{x}^{(k+1)} - D\boldsymbol{x}^{(k)} - U\boldsymbol{x}^{(k)}],$$

then collect the $\boldsymbol{x}^{k+1}$ terms on the left hand side to get

$$\begin{aligned} & (L + \tfrac{1}{\omega} D)\boldsymbol{x}^{(k+1)} &=& \tfrac{1}{\omega} D\boldsymbol{x}^{(k)} + [\boldsymbol{b} - D\boldsymbol{x}^{(k)} - U\boldsymbol{x}^{(k)}] \\ \Rightarrow \quad & (L + \tfrac{1}{\omega} D)\boldsymbol{x}^{(k+1)} &=& (\tfrac{1}{\omega} D - D - U)\boldsymbol{x}^{(k)} + \boldsymbol{b} \\ \Rightarrow \quad & \boldsymbol{x}^{(k+1)} &=& (L + \tfrac{1}{\omega} D)^{-1}[(\tfrac{1}{\omega} D - D - U)\boldsymbol{x}^{(k)} + \boldsymbol{b}] \end{aligned}$$

Notice that the SOR Method is also of the form $\boldsymbol{x} = B\boldsymbol{x} + \tilde{\boldsymbol{b}}$, so that the convergence analysis just done for the Jacobi and Gauss-Seidel Methods can also be done for the SOR Method. The B matrix which determines the convergence of the SOR Method is $(L + \frac{1}{\omega} D)^{-1}(\frac{1}{\omega} D - D - U)$. From this point of view, the idea is to choose a value of ω which minimizes

$\|(L+\frac{1}{\omega}D)^{-1}(\frac{1}{\omega}D-D-U)\|$. As we did earlier for the Jacobi and Gauss-Seidel Methods, it is possible to find the eigenvalues and eigenvectors for the *B* matrix when using the SOR Method for 2 x 2 systems. However, because it is significantly more complicated, we do not do the derivations here. It is left as an exercise for the ambitious student (or the challenging instructor).

Algorithms

In practice, we are usually using a computer to do the iterations, so we need implementable algorithms in order to use the above methods for *n* x *n* systems. We conclude by giving these algorithms for finding element $x_i^{(k+1)}$, given $x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}$. We note that although these particular algorithms are not quite optimally efficient, writing the algorithms this way makes more obvious the slight (but important) differences between the three methods.

Method	Algorithm for performing iteration $k+1$: For i = 1 to n do:
Jacobi	$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)} \right)$
Gauss-Seidel	$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)} \right)$
SOR	$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)} \right)$