Iterative Methods for Solving Linear Systems

Michael D'Antonio Shimon Epstein Felix Lee Benjamin Levine
Griffin Rickle

Fall 2014

In this presentation we will introduce and compare three related iterative methods for solving linear systems. We will first describe the Jacobi method and then derive refinements called the Gauss-Seidel and Successive Over-relaxation (SOR) methods.

1 Jacobi

1.1 The method

For a linear system $A\mathbf{x} = \mathbf{b}$, $x = (x_1, \ldots, x_n)$, the Jacobi method has the explicit analytical form for each x_i given by the equation:

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

The derivation of this analytical formula relies on a simple decomposition of the system $A\mathbf{x} = \mathbf{b}$ into a diagonal component D, and a remainder term R. For the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

we decompose it into $A = D + R$, where

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}, \; R = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}$$

The solution $\mathbf{x}$ is then obtained iteratively via the formula

$$\mathbf{x}^{(k+1)} = D^{-1}(\mathbf{b} - R\mathbf{x}^{(k)}).$$

In fact, we can further decompose our matrix R into strictly lower and upper triangular portions L and U, and our iterative formula then becomes

$$\begin{aligned} D\mathbf{x}^{(k+1)} + (L+U)\mathbf{x}^{(k)} &= \mathbf{b} \\ \mathbf{x}^{(k+1)} &= D^{-1}\left[(-L-U)\mathbf{x}^{(k)} + \mathbf{b}\right]. \end{aligned}$$

Some basic algebra applied to this form indeed yields the explicit formula given by (1).

1.2 Example

Suppose we are given the linear system $A\mathbf{x} = \mathbf{b}$ defined by:

$$A = \begin{bmatrix} 4 & -1 & -1 \\ -2 & 6 & 1 \\ -1 & 1 & 7 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 3 \\ 9 \\ -6 \end{bmatrix}.$$

Decomposing this system as above and arbitrarily choosing $\mathbf{x}^{(0)} = (0, 0, 0)$ as our starting point, we obtain our first value using the formula $D\mathbf{x}^{(k+1)} + (L + U)\mathbf{x}^{(k)} = \mathbf{b}$:

$$\begin{bmatrix} 4 & & \\ & 6 & \\ & & 7 \end{bmatrix} \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & -1 & -1 \\ -2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ -6 \end{bmatrix}$$

As we've chosen for our initial point $\mathbf{x}^{(0)} = (0, 0, 0)$, our system simply becomes

$$\begin{array}{rcrcrcr} 4x_1^{(1)} & + & 0 & + & 0 & = & 3 \\ 0 & + & 6x_2^{(1)} & + & 0 & = & 9 \\ 0 & + & 0 & + & 7x_3^{(1)} & = & -6 \end{array}$$

yielding our next term,

$$\mathbf{x}^{(1)} = \begin{bmatrix} 3/4 \\ 3/2 \\ -6/7 \end{bmatrix}.$$

Continuing this process, one obtains within five iterations that

$$\mathbf{x}^{(5)} = \begin{bmatrix} .999 \\ 1.997 \\ -1.000 \end{bmatrix}$$

and the process indeed converges to the true solution,

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}.$$

2 Gauss-Seidel

2.1 The method

The Gauss-Seidel method is a refinement on the Jacobi method, given by the formula:

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

Whereas in the case of Jacobi we had the formula $D\mathbf{x}^{(k+1)} + (L + U)\mathbf{x}^{(k)} = \mathbf{b}$, with the Gauss-Seidel method we apply the $\mathbf{x}^{(k+1)}$ term to the lower triangular portion of our remainder term as well, and the expression thus becomes

$$\begin{aligned} (L + D)\mathbf{x}^{(k+1)} + U\mathbf{x}^{(k)} &= \mathbf{b} \\ \mathbf{x}^{(k+1)} &= (L + D)^{-1} \left[-U\mathbf{x}^{(k)} + \mathbf{b} \right] \end{aligned}$$

2.2 Example revisited

Revisiting our example from the Jacobi method, we're given the system:

$$A = \begin{bmatrix} 4 & -1 & -1 \\ -2 & 6 & 1 \\ -1 & 1 & 7 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 3 \\ 9 \\ -6 \end{bmatrix}.$$

Using the same initial value $\mathbf{x}^{(0)} = (0, 0, 0)$, we find that our first term yielded by the Gauss-Seidel method is the solution to the lower triangular system $(L + D)\mathbf{x}^{(k+1)} = \mathbf{b}$:

$$\begin{array}{rcrcrcr} 4x_1^{(1)} & + & 0 & + & 0 & = & 3 \\ -2x_1^{(1)} & + & 6x_2^{(1)} & + & 0 & = & 9 \\ -x_1^{(1)} & + & x_2^{(1)} & + & 7x_3^{(1)} & = & -6 \end{array}$$

By solving these iterative systems using simple forward substitution, one will find that the system converges to the true solution of $\mathbf{x} = (1, 2, -1)$ to within three decimal places after only four iterations.

3 Error Analysis of Jacobi and Gauss-Seidel

As the methods illustrated here chiefly deal with linear mappings $F : \mathbb{R}^n \to \mathbb{R}^n$ as opposed to the single variable processes $f : \mathbb{R} \to \mathbb{R}$ primarily covered in the Suli text, it is necessary to introduce a somewhat novel approach in defining our error term. Therefore, our first task will be to derive an analog to the familiar single-variable iterative systems that take the form $x_{n+1} = g(x_n)$.

In both Jacobi and Gauss-Seidel we're given a process in the form $M\mathbf{x}^{(k+1)} = N\mathbf{x}^{(k)} + \mathbf{b}$. To formulate an n-dimensional analog of the above, we isolate our $\mathbf{x}^{(k+1)}$ term by multiplying both sides by M^{-1}, so that

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

which can be rewritten for convenience as

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \beta$$

where $B = M^{-1}N$ and $\beta = M^{-1}\mathbf{b}$. As in the single variable case, the convergence of an iterative method relies on Brouwer's fixed point theorem, namely that our solution $\mathbf{x}$ is a fixed point of the mapping B—and thus, we can assert that $\mathbf{x} = B\mathbf{x} + \beta$. From this form, we obtain our successive error term:

$$\mathbf{e}^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k+1)} = B(\mathbf{x} - \mathbf{x}^{(k)}).$$

We note here that the expression $B(\mathbf{x} - \mathbf{x}^{(k)}) = B\mathbf{e}^{(k)}$, and thus a relationship emerges, namely

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

From here we take advantage of the fact that our induced norm $\|\cdot\|$ is sub-multiplicative: for a matrix $A \in \mathbb{R}^{n\times n}$ and vector $\mathbf{v} \in \mathbb{R}^n$, we are guaranteed that

$$\|A\mathbf{v}\| \leq \|A\|\|\mathbf{v}\|$$

Thus, we have

$$\|\mathbf{e}^{(k)}\| = \|B^k\mathbf{e}^{(0)}\| \leq \|B\|^k\|\mathbf{e}\|$$

And therefore $\|\mathbf{e}^{(k)}\| \to 0$ as $k \to \infty$ for $\|B\| < 1$.

In the case of the Jacobi and Gauss-Seidel method, our iteration matrix B is given by $D^{-1}(-L - U)$

and $(L+D)^{-1}(-U)$, respectively, and thus our rate of convergence is going to be related to the system we're given. In general, for $B \in \mathbb{R}^{n\times n}$ with a full set of n distinct eigenvectors, $\|B\| = |\lambda_{max}|$. For a given system, the iterative method yielding the highest rate of convergence will generally be the one whose iteration matrix B has eigenvalues that are as small in magnitude as possible; however, no generalizations can be made from the iteration matrix of an iterative method alone.

4 Successive Over-relaxation (SOR)

4.1 The method

Successive over-relaxation is a variant of the Gauss-Seidel method that introduces a relaxation factor $\omega > 1$, from which we obtain the formula:

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

This formula is obtained by applying a relaxation factor ω to our successive error term from Gauss-Seidel. Using some simple algebra on our explicit form (for A, D, L, and U as defined previously):

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

we can obtain a general formula for this successive Gauss-Seidel error term:

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = D^{-1}\left[\mathbf{b} - L\mathbf{x}^{(k+1)} - D\mathbf{x}^{(k)} - U\mathbf{x}^{(k)}\right]$$

To obtain the SOR method, a relaxation factor is introduced into this equation:

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

By solving the above for $\mathbf{x}^{(k+1)}$, we obtain the analytical expression:

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

The formula for each element then follows from this expression.

In general, the principles of SOR can be applied to any iterative method for solving linear systems by incorporating a technique called Richardson extrapolation, although further details are outside the scope of this paper.

4.2 Error Analysis

As with the Jacobi and Gauss-Seidel methods, there is little that can be said of the SOR method's rate of convergence other than it tends to be faster than that of Gauss-Seidel's (which in turn tends to be faster than Jacobi's). However, the choice of ω depends on properties of our iteration matrix, and it is generally not possible to find a value of ω in advance that will maximize the rate of convergence for a given system.

It has been proven in an oft-cited paper by the Russian mathematician Alexander Ostrowski that for a matrix A that is symmetric and positive-definite, SOR systems converge for $0 < \omega < 2$—however, as SOR is meant to be an improvement on the convergence rate of other methods whose convergence is often also guaranteed, this result is of somewhat limited practical usefulness. Values of $\omega > 1$ are used to speed up convergence of a slow-converging process, while values of $\omega < 1$ can be used to establish

convergence in a process that diverges. Some methods adaptively set the relaxation parameter ω to achieve superlinear convergence using a similar mechanism to MATLAB's globally adaptive quadrature used in numerical integration, but these methods tend to work only in special cases and fail to converge in others.

We can generally state that fixing $\omega = 1$ yields a Gauss-Seidel iterative mechanism, and values $1 < \omega < 2$ can guarantee for sufficiently defined asymmetric systems (see C.G. Broyden 1963); however, as the convergence of a linear system depends on both the eigenvalues of the iteration matrix as well as the choice of ω, no single definitive statement can be made on the rate of convergence of the successive over-relaxation method.