Iterative Methods

Partial Differential Equations

J Robert Buchanan

Department of Mathematics

Fall 2025

Objectives

In this lesson we will:

- present and use the Jacobi method for approximating the solution to a linear system of equations,
- present and use the Gauss-Seidel method for approximating the solution to a linear system of equations,
- present and use the Successive Over-Relaxation (SOR) method for approximating the solution to a linear system of equations,

Background

If A is an $n \times n$ matrix and $\mathbf{u}$ and $\mathbf{b}$ are column vectors with n components, then let $\mathbf{u}^{(0)}$ be an approximation to $\mathbf{u}$ where $\mathbf{u}$ is the solution of

$$A\mathbf{u} = \mathbf{b}.$$

Each of the three iterative methods generates a sequence of vectors $\{\mathbf{u}^{(k)}\}_{k=0}^{\infty}$ which converges to $\mathbf{u}$.

Jacobi Method

Consider a linear system of equations expressed in matrix/vector form as $A\mathbf{u} = \mathbf{b}$. Suppose the diagonal entries of matrix A are all nonzero. The ith equation of this system can be expressed as

$$a_{i1}u_1 + a_{i2}u_2 + \cdots + a_{ii}u_i + \cdots + a_{in}u_n = b_i.$$

Solving this equation for u_i yields

$$u_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}u_j\right).$$

Thus if $\mathbf{u}^{(k)} = (u_1^{(k)}, u_2^{(k)}, \ldots, u_n^{(k)})^T$ for some $k = 0, 1, \ldots$ is an approximation to the solution of $A\mathbf{u} = \mathbf{b}$ then the next approximation is $\mathbf{u}^{(k+1)}$ where

$$u_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}u_j^{(k)}\right),$$

for $i = 1, 2, \ldots, n$.

Jacobi Method

If matrix $A = D + L + U$ where D is a diagonal matrix who entries are the diagonal entries of A, L is a strictly lower triangular matrix (zeros on the diagonal) whose entries are the entries of A strictly below the diagonal, and U is a strictly upper triangular matrix whose entries are the entries of A strictly above the diagonal, then

$$A\mathbf{u} = (D + L + U)\mathbf{u} = \mathbf{b}.$$

If the diagonal entries of A are all nonzero then D is invertible and the Jacobi method may be written in matrix/vector form as

$$\mathbf{u}^{(k+1)} = -D^{-1}(L + U)\mathbf{u}^{(k)} + D^{-1}\mathbf{b} = T_J\mathbf{u}^{(k)} + D^{-1}\mathbf{b}$$

where $T_J = -D^{-1}(L + U)$ is called the **Jacobi iteration matrix**.

Stopping Condition

A stopping condition should indicate that further iteration is unnecessary (either because the solution to the linear system has been found or because the results of further iteration will not appreciably change). Many choices of stopping condition are available and a generally effective choice is that the iteration should cease when the relative change in successive approximations falls below a specified threshold. Thus if the threshold is given by $\epsilon > 0$, calculation of further iterates of the sequence $\{\mathbf{u}^{(k)}\}$ can stop when

$$\frac{\|\mathbf{u}^{(k)} - \mathbf{u}^{(k+1)}\|_2}{\|\mathbf{u}^{(k+1)}\|_2} < \epsilon.$$

The subscript of 2 indicates the l_2-norm is used to measure the vectors.

Example

Use a maximum of five iterations of the Jacobi method to approximate a solution to the following system of linear equations. Let $\epsilon = 10^{-4}$ and $\mathbf{u}^{(0)} = (1, 1, 1, 1)^T$.

$$\begin{aligned} 4u_1 + u_2 - u_3 + u_4 &= 6 \\ u_1 + 4u_2 - u_3 - u_4 &= 6 \\ -u_1 - u_2 + 4u_3 + u_4 &= 3 \\ u_1 - u_2 + u_3 + 3u_4 &= -2 \end{aligned}$$

For the purposes of comparison the exact solution is $\mathbf{u} = \frac{1}{37}(80, 38, 73, -63)^T$.

Solution

k	$u_1^{(k)}$	$u_2^{(k)}$	$u_3^{(k)}$	$u_4^{(k)}$
0	1.0000	1.0000	1.0000	1.0000
1	1.2500	1.7500	1.0000	-1.0000
2	1.5625	1.1875	1.7500	-0.8333
3	1.8490	1.3385	1.6458	-1.3750
4	1.9206	1.1055	1.8906	-1.3854
5	2.0426	1.1462	1.8529	-1.5686

The l_2-norm error in $\mathbf{u}^{(5)}$ is

$$\|\mathbf{u}^{(5)} - \mathbf{u}\|_2 \approx 0.2468$$

The stopping condition was not achieved in the first five iterations, thus emphasizing the need for practical implementations of the Jacobi method (and other iterative methods) to specify a maximum number of iterations to compute.

Gauss-Seidel Method

The Gauss-Seidel method modifies the Jacobi method by using the already calculated entries of $\mathbf{u}^{(k+1)}$ in place of the corresponding entries of $\mathbf{u}^{(k)}$.

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

Gauss-Seidel Method

If $A = D + L + U$ then

$$\begin{aligned} A\mathbf{u} &= \mathbf{b} \\ (D + L + U)\mathbf{u} &= \mathbf{b} \\ \mathbf{u} &= D^{-1}(\mathbf{b} - L\mathbf{u} - U\mathbf{u}). \end{aligned}$$

If $a_{ii} \neq 0$ for all $i = 1, 2, \ldots, n$ then the inverse of diagonal matrix D exists. The Gauss-Seidel formula can then be expressed as

$$\mathbf{u}^{(k+1)} = D^{-1}(\mathbf{b} - L\mathbf{u}^{(k+1)} - U\mathbf{u}^{(k)}).$$

Collecting the $k+1$st iterates on the left-hand side of the equation enables the Gauss-Seidel iteration formula to be written as

$$\mathbf{u}^{(k+1)} = -(D + L)^{-1}U\mathbf{u}^{(k)} + (D + L)^{-1}\mathbf{b} = T_{GS}\mathbf{u}^{(k)} + (D + L)^{-1}\mathbf{b}$$

where $T_{GS} = -(D + L)^{-1}U$ is called the **Gauss-Seidel iteration matrix.**

Example

Use the Gauss-Seidel method to approximate a solution to the following system of linear equations. Let $\epsilon = 10^{-4}$ and $\mathbf{u}^{(0)} = (1, 1, 1, 1)^T$.

$$\begin{aligned} 4u_1 + u_2 - u_3 + u_4 &= 6 \\ u_1 + 4u_2 - u_3 - u_4 &= 6 \\ -u_1 - u_2 + 4u_3 + u_4 &= 3 \\ u_1 - u_2 + u_3 + 3u_4 &= -2 \end{aligned}$$

For the purposes of comparison the exact solution is $\mathbf{u} = \frac{1}{37}(80, 38, 73, -63)^T$.

Solution

k	$u_1^{(k)}$	$u_2^{(k)}$	$u_3^{(k)}$	$u_4^{(k)}$
0	1.0000	1.0000	1.0000	1.0000
1	1.2500	1.6875	1.2344	-0.9323
2	1.6198	1.1706	1.6807	-1.3766
3	1.9717	1.0831	1.8578	-1.5821
4	2.0892	1.0466	1.9295	-1.6574
5	2.1351	1.0343	1.9567	-1.6858

The error in the last approximation to the solution is

$$\|\mathbf{u}^{(5)} - \mathbf{u}\|_2 \approx 0.0002$$

which is clearly an improvement over the Jacobi method's result after five iterations.

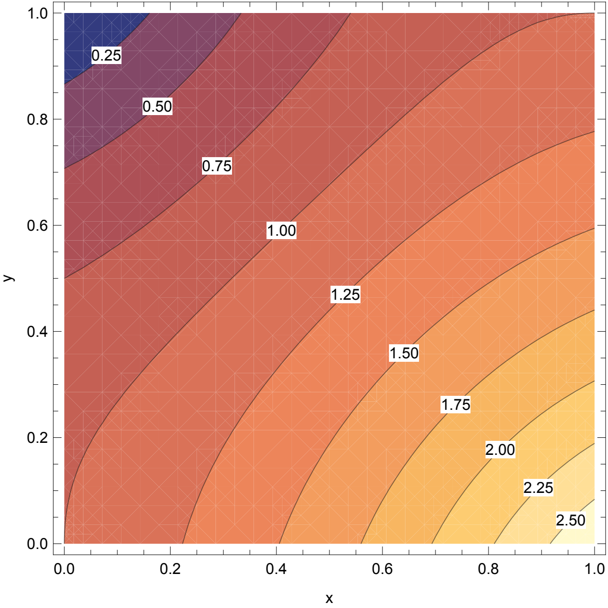
Example

Consider the following boundary value problem:

$$
\begin{aligned}
u_{xx} + u_{yy} &= 0 \text{ for } 0 < x < 1 \text{ and } 0 < y < 1 \\
u(x,0) &= e^x \text{ for } 0 < x < 1 \\
u(x,1) &= \sin(\pi x/2) \text{ for } 0 < x < 1 \\
u(0,y) &= 1 - y^2 \text{ for } 0 < y < 1 \\
u(1,y) &= e^{1-y} \text{ for } 0 < y < 1.
\end{aligned}
$$

If $h = 1/10$, then the linear system $A\mathbf{u} = \mathbf{b}$ consists of a 81×81 matrix A and a column vector $\mathbf{b}$ with 81 elements. Setting a numerical tolerance of $\epsilon = 10^{-6}$ and using the Gauss-Seidel method requires on the order of 100 iterations from a starting approximation of $\mathbf{u}^{(0)} = (1, 1, \ldots, 1)^T$.

Solution


0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2.25
2.50
1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x

Successive Over-Relaxation Method

Since $\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$ can be thought of as the step to take from approximation $\mathbf{u}^{(k)}$ to the better approximation $\mathbf{u}^{(k+1)}$, a natural improvement to the method is to take a longer step in the same direction.

If $\omega > 1$ then $\omega(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)})$ is a vector in the same direction as $\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$ but with greater magnitude. By taking this longer corrective step an algorithm can "accelerate" toward the exact solution.

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \omega(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}).$$

When $\omega > 1$ this iterative method is called an **over-relaxation method**. When $0 < \omega < 1$ this iterative method is called an **under-relaxation method**.

The **successive over-relaxation method** (abbreviated as SOR) is an accelerated modification of the Gauss-Seidel method.

Matrix/Vector Form

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

where $T_{SOR} = (D + \omega L)^{-1}\left[(1-\omega)D - \omega U\right]$ is called the **SOR iteration matrix.**

The ith component of $\mathbf{u}^{(k+1)}$ can be written as

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

Note that if $\omega = 1$ then the successive over-relaxation method is just the Gauss-Seidel method.

Example

Compare the results of the Gauss-Seidel and SOR methods with $\omega = 1.21539$ for approximating the solution to the following linear system of equations. For both methods use $\epsilon = 10^{-4}$ in the stopping criterion.

$$\begin{bmatrix} -4 & 1 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

Gauss-Seidel Results

The initial approximation to the solution will be $(0, 0, 0, 0, 0)^T$. The first ten iterations of the Gauss-Seidel method yield a close approximation of the solution to the equation. The exact solution is $\mathbf{x} = (-129/260, -64/65, -75/52, -116/65, -441/260)^T$.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$x_5^{(k)}$
0	0.0000	0.0000	0.0000	0.0000	0.0000
1	−0.2500	−0.5625	−0.8906	−1.2227	−1.5557
2	−0.3906	−0.8203	−1.2607	−1.7041	−1.6760
3	−0.4551	−0.9290	−1.4083	−1.7711	−1.6928
4	−0.4822	−0.9726	−1.4359	−1.7822	−1.6955
5	−0.4932	−0.9823	−1.4411	−1.7842	−1.6960
6	−0.4956	−0.9842	−1.4421	−1.7845	−1.6961
7	−0.4960	−0.9845	−1.4423	−1.7846	−1.6961
8	−0.4961	−0.9846	−1.4423	−1.7846	−1.6962
9	−0.4961	−0.9846	−1.4423	−1.7846	−1.6962
10	−0.4962	−0.9846	−1.4423	−1.7846	−1.6962

SOR Results

The SOR method has similar performance though to the four decimal places reported in the table, the 9th and 10th iterates are the same.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$x_5^{(k)}$
0	0.0000	0.0000	0.0000	0.0000	0.0000
1	−0.3038	−0.7000	−1.1242	−1.5570	−1.9923
2	−0.4511	−0.9356	−1.4268	−1.9189	−1.6732
3	−0.4910	−0.9889	−1.4878	−1.7625	−1.6944
4	−0.4986	−0.9982	−1.4299	−1.7851	−1.6967
5	−0.4998	−0.9790	−1.4434	−1.7850	−1.6962
6	−0.4937	−0.9854	−1.4424	−1.7846	−1.6961
7	−0.4969	−0.9847	−1.4423	−1.7846	−1.6962
8	−0.4960	−0.9845	−1.4423	−1.7846	−1.6962
9	−0.4962	−0.9846	−1.4423	−1.7846	−1.6962
10	−0.4962	−0.9846	−1.4423	−1.7846	−1.6962

Remarks

- There is no known formula for the optimal value of ω to accelerate the SOR method for a general $n \times n$ matrix A.
- For certain banded matrices (for example tridiagonal matrices) the optimal value of ω can be determined *a priori*.
- For example, if $\omega \approx 1.52786$ then the SOR method approximates the solution to a boundary value problem earlier in the previous section in 25 iterations as opposed to nearly 100 iterations required for the Gauss-Seidel method.

Example

Consider the boundary value problem:

$$\Delta u = e^{x-y} \text{ on } R$$
$$u(x,y) = x\,y \text{ on } \partial R.$$

Let the domain of the boundary value problem be $R = \{(x,y) \,|\, 0 < x < 4,\ 0 < y < 3\}$.

Use the successive over-relaxation method with $\omega = 1.81621$, the nine-point approximation to the Laplacian operator, and $h = 1/10$ to approximate the solution to the boundary value problem. Use $\epsilon = 10^{-6}$ in the stopping criterion.

Solution (1 of 2)

The rectangle over which the solution is sought has dimensions of width 4 and height 3. With Dirichlet boundary conditions and choosing $h = 1/10$ there are

$$\left(\frac{4}{1/10} - 1\right)\left(\frac{3}{1/10} - 1\right) = 1131$$

interior points at which the solution to Poisson's equation must be approximated. The linear system to be solved is of the form $A\,\mathbf{u} = \mathbf{b}$ where A is a banded tridiagonal matrix of size 1131×1131. Despite the size of the linear system, the SOR technique solves it in 92 iterations from a starting approximation of $\mathbf{u}^{(0)} = \mathbf{0}$.

Solution (2 of 2)