Chapter 6

RELAXATION METHODS

6.1 Stationary Methods

6.2 Convergence

We seek to solve $Ax = b$ where $A$ large and sparse. $x^{(0)}, x^{(1)}, x^{(2)}, \ldots \to x$. These methods are very economical with storage. They are the only way to solve general discretized PDEs in 3 space variables.

6.1 Algorithms

Saad, Section 4.1.

Example

\[
\begin{array}{ccc}
0 & 0 & 0 \\
\hline
1 & 1 & 1 \\
0 & \ldots & 0 \\
1 & 1 & 1 \\
0 & \ldots & 0 \\
1 & 1 & 1 \\
0 & \ldots & 0 \\
1 & 1 & 1 \\
0 & \ldots & 0 \\
1 & 1 & 1 \\
\end{array}
\]

Away from a boundary

\[
\frac{1}{h^2} \begin{pmatrix}
- u_{i+3} \\
- u_{i-1} + 4u_i - u_{i+1} \\
- u_{i-3}
\end{pmatrix} = 0
\]
\textit{Note.} This is not a matrix but a two dimensional arrangement of an expression. Matrices are enclosed in brackets.

\[ u_i = \frac{1}{4} \left( \begin{array}{c} u_{i+3} \\ + u_{i-1} + u_{i+1} \\ + u_{i-3} \end{array} \right) \]

Jacobi’s method \( u_i^{\text{new}} = \frac{1}{4} \left( \begin{array}{c} u_{i+3} \\ + u_{i-1} + u_{i+1} \\ + u_{i-3} \end{array} \right)^{\text{old}} \)

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

This is also called the method of simultaneous displacements. (Note \( (0.375 + 0.0625) + 0.0625 = 0.5 \) for round-to-even but \( 0.501 \) for rounding.)

Gauss-Seidel method \( u_i^{\text{new}} = \frac{1}{4} \left( u_{i-1}^{\text{new}} \right) + \frac{1}{4} \left( u_{i+3}^{\text{new}} + u_{i+1}^{\text{old}} \right) \)

Assuming we sweep from the lower left to the upper right (↗),

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

This is also called the method of successive displacements. Gauss-Seidel requires only half as much memory because \( u_i^{\text{new}} \) can overwrite \( u_i^{\text{old}} \). For this problem Gauss-Seidel converges twice as fast as Jacobi. We call these \textit{relaxation} methods because at each point we relax one variable so as to satisfy an equation.
I recommend this method to you for imitation. You will hardly ever again eliminate directly, at least not when you have more than 2 unknowns. The indirect procedure can be done while half asleep, or while thinking about other things.

Gauss, December 26, 1823

In general for \( \sum_j a_{ij} x_j = b_i \), solve for \( x_i \) in \( i \)th equation:

\[
x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j \right).
\]

Iterate

\[
x^{(k+1)}_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x^{(k)}_j \right). \quad \text{Jacobi}
\]

This can be expressed in matrix notation if we write \( A = L + D + U \) where \( L \) is strictly lower triangular, \( D \) is diagonal, and \( U \) is strictly upper triangular:

\[
Ax = b, \quad (L + D + U)x = b, \quad x = D^{-1}(b - (L + U)x),
\]

\[
x^{(k+1)} = D^{-1}(b - (L + U)x^{(k)}). \quad \text{Jacobi}
\]

Gauss-Seidel is

\[
x^{(k+1)}_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j < i} a_{ij} x^{(k+1)}_j - \sum_{j > i} a_{ij} x^{(k)}_j \right)
\]
or

\[
x^{(k+1)} = D^{-1}(b - Lx^{(k+1)} - Ux^{(k)}). \quad \text{Gauss-Seidel}
\]

**residual correction paradigm** Suppose we have \( C \approx A^{-1} \). We can get \( x^{(1)} = C \cdot b \). We have

\[
\text{remainder} = A^{-1} \cdot \text{residual}.
\]

Therefore

\[
x = x^{(1)} + A^{-1}(b - Ax^{(1)}).
\]

How about \( x^{(2)} = x^{(1)} + C \cdot (b - Ax^{(1)}) \)? Consider as an example \( 3x = 1, C = 0.3 \). The iteration is

\[
x^{(k+1)} = x^{(k)} + 0.3(1 - 3x^{(k)}),
\]

and the iterates are \( x^{(0)} = 0, x^{(1)} = 0.3, x^{(2)} = 0.33, x^{(3)} = 0.333, \ldots \). In this framework

\[
\begin{align*}
\text{Jacobi} : & \quad x^{(k+1)} = x^{(k)} + D^{-1}(b - Ax^{(k)}) \\
\text{Gauss-Seidel} : & \quad x^{(k+1)} = x^{(k)} + (L + D)^{-1}(b - Ax^{(k)})
\end{align*}
\]
Viewed this way, we see that the correction is some approximation to $A^{-1}$ times the residual. Also iterative improvement fits in this framework.

For further insight consider the simple example

$$-\frac{d^2}{dx^2} u = 0, \quad 0 < x < 1, \quad u(0) = 0, \quad \frac{d}{dx} u(1) = 0$$

![Diagram of grid](image)

$h=1/J$

A numerical solution $u_j \approx u(jh)$ is obtained by solving

$$-\frac{1}{h} \left( \frac{u_{j+1} - u_j}{h} - \frac{u_j - u_{j-1}}{h} \right) = 0, \quad 1 \leq j \leq J - 1, \quad u_0 = 0, \quad \frac{u_J - u_{J-1}}{h} = 0.$$  

The value of $u_0$ is obviously known:

$$u_0^\text{new} = u_0^\text{old} = 0$$

Solving $j$th equation for $j$th unknown:

Jacobi

$$u_j^\text{new} = \frac{1}{2} (u_{j-1}^\text{old} + u_{j+1}^\text{old})$$

$$u_J^\text{new} = u_{J-1}^\text{old}$$

Gauss-Seidel

$$u_j^\text{new} = \frac{1}{2} (u_{j-1}^\text{new} + u_{j+1}^\text{new})$$

$$u_J^\text{new} = u_{J-1}^\text{new}$$

overdo it by 50 %

$$u_j^\text{new} = u_j^\text{old} + 1.5\left(\frac{1}{2}u_{j-1}^\text{new} + \frac{1}{2}u_{j+1}^\text{old} - u_j^\text{old}\right)$$

$$u_J^\text{new} = u_J^\text{old} + 1.5(u_{J-1}^\text{new} - u_J^\text{old})$$

Note: $1.5 \to 1$ gives Gauss-Seidel.
successive over-relaxation (SOR)  Gauss-Seidel:

\[
\begin{align*}
  x^{(k+1)} &= D^{-1}(b - Lx^{(k+1)} - Ux^{(k)}) \\
  x^{(k+1)} &= x^{(k)} + D^{-1} \left(b - Lx^{(k+1)} - (D + U)x^{(k)}\right)
\end{align*}
\]

\[
\begin{align*}
  x_i^{(k+1)} &= x_i^{(k)} + \frac{1}{\omega} a_{ii} \left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right)
\end{align*}
\]

Introduce relaxation parameter \( \omega \) to hasten convergence. \( \omega > 1 \Longleftrightarrow \) over-relaxation

\[
\begin{align*}
  x^{(k+1)} &= x^{(k)} + \omega D^{-1}(b - Lx^{(k+1)} - (D + U)x^{(k)}) \\
  (I + \omega D^{-1}L)x^{(k+1)} &= (I + \omega D^{-1}L)x^{(k)} + \omega D^{-1}(b - Ax^{(k)}) \\
  x^{(k+1)} &= x^{(k)} + \left(L + \frac{1}{\omega} D\right)^{-1}(b - Ax^{(k)})
\end{align*}
\]

Other methods

- symmetric SOR—SSOR: reverse the ordering every iteration

- block Jacobi, block SOR: collectively relax a block of unknowns, e.g., line Gauss-Seidel relaxes unknowns on an entire line of grid points making their residuals all vanish.

**Review questions**

1. What is another name for the method of simultaneous displacements? successive displacements?

2. What is the Jacobi method for \((L + D + U)x = b\) where \(L\) is strictly lower triangular, \(D\) is diagonal, and \(U\) is strictly upper triangular? Gauss-Seidel? SOR? Write each method as a matrix iteration which mirrors its implementation.

3. Apply Jacobi, Gauss-Seidel, and SOR to specific examples.

4. What does SOR abbreviate? What do we call the parameter that it uses?
5. Given the computational form of a stationary iterative method, rewrite it in the form of iterative refinement in which some matrix \( M \) is approximating the original matrix \( A \).

6. What is SSOR?

7. What is the form of a stationary iterative method?

**Exercises**

1. The following algorithm is supposed to do one iteration of SOR for \( Ax = b \). Diagonal elements of \( A \) are stored in \( d \) and the \( i \)th component of \( b - Ax \) is computed by the function \( \text{residual}(i, x) \).

   for \( i = 1, 2, \ldots, n \) do
   \[
   \Delta x_i = \text{residual}(i, x)/d_i;
   \]
   \[
   x = x + \omega \ast \Delta x;
   \]

   This is not SOR. Rewrite the algorithm correctly (assuming \( A \) and \( b \) are inaccessible except through \( \text{residual} \)).

2. Consider the system of equations

   \[
   5x + 2y + 2z = 10, \\
   x + 2y + z = 10, \\
   x + 5y + 10z = 10.
   \]

   Suppose we are solving this with SOR with relaxation parameter \( \omega = 1.5 \) and have already computed the iterates

<table>
<thead>
<tr>
<th>( k )</th>
<th>( x_k )</th>
<th>( y_k )</th>
<th>( z_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Determine \( y_1 \).

**6.2 Convergence**

*Saad, Sections 4.2.0, 4.2.1.*
6.2.1 Conditions for convergence

Jacobi \[ x^{(k+1)} = (I - D^{-1}A)x^{(k)} + D^{-1}b, \]
SOR \[ x^{(k+1)} = (I - (L + \frac{1}{\omega}D)^{-1}A)x^{(k)} + (L + \frac{1}{\omega}D)^{-1}b. \]

Both are stationary iterative methods of the form

\[ x^{(k+1)} = Bx^{(k)} + c \]

(stationary because \( B \) and \( c \) do not change) where

\[ B_J = -D^{-1}(L + U), \]
\[ B_\omega = (D + \omega L)^{-1}((1 - \omega)D - \omega U). \]

In general

\[ x^{(k)} \to x \implies x = Bx + c. \]

This latter condition is satisfied for \( B_J \) and \( B_\omega \). (Why?) Hence successive errors satisfy

\[ x^{(k+1)} - x = B(x^{(k)} - x), \]

so

\[ x^{(k)} - x = B^k(x^{(0)} - x). \]

What is \( B^k \)? If \( B \) is diagonalizable

\[ B = X\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)X^{-1} \]

and

\[ B^k = X\text{diag}(\lambda_1^k, \lambda_2^k, \ldots, \lambda_n^k)X^{-1}. \]

\[ \lim_{k \to \infty} B^k = 0 \iff \lim_{k \to \infty} X^{-1}B^kX = 0 \]
\[ \iff \lim_{k \to \infty} \lambda_i^k = 0 \quad \text{for } i = 1, 2, \ldots, n \]
\[ \iff |\lambda_i| < 1 \quad \text{for } i = 1, 2, \ldots, n \]
\[ \iff \rho(B) < 1. \]

THEOREM A stationary iterative method \( x^{(k+1)} = Bx^{(k)} + c \) converges for an arbitrary initial approximation \( x^{(0)} \) if and only if \( \rho(B) < 1. \)

Proof. Recall \( x^{(k)} - x = B^k(x^{(0)} - x). \)

\( (\Leftarrow) \rho(B) < 1 \implies B^k \to 0 \text{ as } k \to \infty \implies x^{(k)} \to x \text{ as } k \to \infty \)

\( (\Rightarrow) \) Assume \( \rho(B) \geq 1. \) There exists \( v \neq 0 \) and \( |\lambda| \geq 1 \) such that \( Bv = \lambda v. \) Let \( x^{(0)} = x + v. \) Then \( x^{(k)} - x = B^kv = \lambda^kv \not\to 0 \text{ as } k \to \infty. \) \( \square \)
Jacobi and Gauss-Seidel converge if \( A \) is strictly diagonally dominant. If \( n = 3 \)

\[
B_J = \begin{bmatrix}
0 & -a_{12} & -a_{13} \\
-a_{21} & a_{11} & a_{11} \\
a_{22} & 0 & -a_{23} \\
a_{31} & -a_{32} & a_{33} \\
a_{33} & 0 & -a_{33}
\end{bmatrix}.
\]

\[\|B_J\|_\infty = \max \left\{ \frac{|a_{12}| + |a_{13}|}{|a_{11}|}, \frac{|a_{21}| + |a_{22}|}{|a_{22}|}, \frac{|a_{31}| + |a_{32}|}{|a_{33}|} \right\} < 1 \]

\[\Rightarrow \rho(B_J) < 1\]

For SOR, \( \rho(B_\omega) \geq |\omega - 1| \) regardless of \( A \). Convergence implies \( 0 < \omega < 2 \). Thus only these values of \( \omega \) are of interest. Conversely, if \( A \) is symmetric positive definite, \( 0 < \omega < 2 \) implies convergence.

### 6.2.2 Rate of convergence

Convergence is not enough; we need a reasonable rate of convergence. We want to know the number of iterations \( k \) required to reduce error by a factor \( \varepsilon \):

\[
\varepsilon = \max_{x^{(0)} \neq x} \frac{\|B^k(x^{(0)} - x)\|}{\|x^{(0)} - x\|} = \|B^k\|.
\]

It can be shown \( \lim_{k \to \infty} \|B^k\|^{1/k} = \rho(B) \) where \( \|B^k\|^{1/k} \) represents the average error reduction per iteration. Therefore

\[\varepsilon^{1/k} \sim \rho(B) \text{ as } k \to \infty,\]

\[k \sim \frac{\log \varepsilon}{\log \rho(B)} \text{ as } k \to \infty,\]

or

\[k(\varepsilon) \sim \frac{\log \varepsilon}{\log \rho(B)} \text{ as } \varepsilon \to 0.\]

E.g., for \( \varepsilon = 10^{-d} \) and \( \rho(B) = 0.99 \) the number of iterations = 229\(d\).

*Example* Five-point difference operator on an \( N + 1 \) by \( N + 1 \) grid:
\[
\rho(B_J) = \cos \frac{\pi}{N} = 1 - \frac{\pi^2}{2N^2} + O\left(\frac{1}{N^4}\right) \\
\rho(B_{GS}) = \cos\frac{\pi}{N} = 1 - \frac{\pi^2}{N^2} + O\left(\frac{1}{N^4}\right)
\]
\[
\log \rho(B_J) = -\frac{\pi^2}{2N^2} + O\left(\frac{1}{N^4}\right) \\
\log \rho(B_{GS}) = -\frac{\pi^2}{N^2} + O\left(\frac{1}{N^4}\right)
\]
\[
k(\varepsilon) \approx \frac{2\pi^2}{N^2} \log \frac{1}{\varepsilon} \\
k(\varepsilon) \approx \frac{N^2}{2\pi^2} \log \frac{1}{\varepsilon}
\]

Let \(n = (N - 1)^2\). The cost per iteration is \(2n\) “multiplications.” The cost for Gauss-Seidel is \(2n \cdot \frac{n}{\pi^2} \log \frac{1}{\varepsilon}\). If we want to reduce the iteration error \(x^{(k)} - x\) to the level of discretization error \(O\left(\frac{1}{N^2}\right)\), then \(\varepsilon = O\left(\frac{1}{N^2}\right)\) and cost = \(O(n^2 \log n)\).

For SOR we want to choose \(\omega\) to minimize \(\rho(B_\omega)\).

“SOR theory”

For the example one should choose

\[
\bar{\omega} = \frac{2}{1 + \sin \frac{\pi}{N}},
\]

for which

\[
\rho(B_\bar{\omega}) = \frac{1 - \sin \frac{\pi}{N}}{1 + \sin \frac{\pi}{N}} = 1 - \frac{\pi}{N} + O\left(\frac{1}{N^2}\right) = 1 - \frac{2\pi}{N} + O\left(\frac{1}{N^2}\right).
\]

The number of iterations to reduce error by \(\varepsilon\) is

\[
k(\varepsilon) \sim \frac{\log \varepsilon}{\log \rho(B_\bar{\omega})} \sim \frac{N}{2\pi} \log \frac{1}{\varepsilon},
\]

and the computation is

\[
u_i^{\text{new}} = u_i^{\text{old}} + \frac{\omega}{4} \left( \begin{array}{c}
-u_{i-1} \\
-u_{i-N+1}
\end{array} \right) - \left( \begin{array}{c}
-\frac{u_{i-n-1}}{+4u_i - u_{i+1}} \\
-u_{i-N+1}
\end{array} \right)
\]

\[
= (1 - \omega)u_i^{\text{old}} + \frac{\omega}{4} \left( \begin{array}{c}
u_{i-1} \\
+u_{i-N+1}
\end{array} \right) + \left( \begin{array}{c}
u_{i+N-1} \\
+u_{i+1}
\end{array} \right),
\]
which is 3 “multiplications.” The cost of reducing the error by $\varepsilon$ is

$$\frac{3}{2\pi} n^{3/2} \log \frac{1}{\varepsilon} \text{ “multiplications.”}$$

**Review questions**

1. What is a necessary and sufficient condition for convergence of a stationary iterative method $x^{(k+1)} = Bx^{(k)} + c$ for an arbitrary initial approximation?

2. What condition on the relaxation parameter $\omega$ of SOR is necessary for convergence? What condition is sufficient for convergence in the case of a symmetric positive definite matrix?

3. Asymptotically how many iterations are required for a stationary iterative method $x^{(k+1)} = Bx^{(k)} + c$ to reduce the error by a factor $\varepsilon$?

**Exercises**

1. For the problem

$$\begin{bmatrix} 2 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 5 \end{bmatrix}$$

   determine the Jacobi iteration matrix and its spectral radius. Use the latter to estimate how many Jacobi iterations would be needed to reduce by a factor of 1000 the error in some approximate solution.

2. Estimate how many Jacobi iterations would be needed to reduce the error by a factor of $10^{-6}$ for $Ax = b$ where

   $$A = \begin{bmatrix} 4 & 1 & 1 \\ 1 & 4 & \cdots \\ \cdots & \cdots & 1 \\ 1 & 1 & 4 \end{bmatrix}$$

3. For SOR applied to $\sum_{j=1}^{n} a_{ij}x_j = b_i$ write down the equation which defines $x_i^{(k+1)}$ in terms of previously computed values.

4. Show that for any 2 by 2 system, the Jacobi method converges if and only if Gauss-Seidel converges.

5. Prove that SOR with $0 < \omega \leq 1$ converges for strictly diagonally dominant matrices.
6. Show that for any \( \omega \) between 1 and 2 there exists a strictly diagonally dominant matrix (depending on \( \omega \)) such that SOR does not converge for that matrix.

7. (a) Let \( A \) be the \( (N - 1)^2 \times (N - 1)^2 \) matrix defined as follows: For any vector 
\[
u = [u_{1,1}; u_{1,2}; u_{2,1}; \ldots; u_{1,N-1}; u_{N-2}; \ldots; u_{N-1,1}; u_{2,N-1}; u_{3,N-1}; \ldots; u_{N-1,2}; \ldots; u_{N-1,N-1}]^T
\]
define
\[(Au)_{i,j} = 4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}
\]
where 
\[u_{0,j} = u_{N,j} = u_{i,0} = u_{i,N} = 0.\]
The eigenvectors of \( A \) are \( v^{k,l}, 1 \leq k, l \leq N - 1 \), where
\[v_{i,j}^{k,l} = \sin\left(\frac{k\pi i}{N}\right) \sin\left(\frac{l\pi j}{N}\right).
\]
Determine the eigenvalues of \( A \).
(b) Express \( B_J \) as a polynomial in \( A \), and thus determine the eigenvalues and the spectral radius of \( B_J \).
(c) Write \( A \) as a block tridiagonal matrix and separately define each of the blocks.