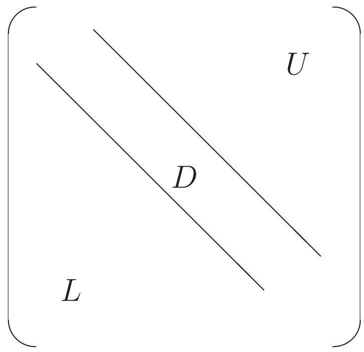
4.1 Jacobi method

A is decomposed into a sum of lower-triangular (L), diagonal (D) and upper-triangular terms (U):

$$A = L + D + U$$

$A =$



4.1.1 Example

A for Backward Euler Method with Dirichlet boundary conditions:

$$\mathbf{A} = \begin{pmatrix} 1+2s & -s & & 0 \\ -s & 1+2s & -s & \\ \ddots & \ddots & \ddots & \\ & -s & 1+2s & -s \\ 0 & & -s & 1+2s \end{pmatrix}$$

$$\Rightarrow \mathbf{D} = \begin{pmatrix} 1+2s & & 0 \\ & 1+2s & \\ & & \ddots & \\ 0 & & & 1+2s \end{pmatrix}, \mathbf{L} = \begin{pmatrix} 0 & & 0 \\ -s & \ddots & \\ & \ddots & \ddots \\ 0 & -s & 0 \end{pmatrix},$$

$$\mathbf{U} = \begin{pmatrix} 0 & -s & 0 \\ \ddots & \ddots & \\ & \ddots & -s \\ 0 & & 0 \end{pmatrix}$$

We want to solve $A\vec{x} = b$

$$\Rightarrow (D + L + U)\vec{x} = \vec{b}$$

$$\text{or} \qquad D\vec{x} = b - (L+U)\vec{x}$$

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

If $\vec{x}^k$ is k^{th} estimate of solution $A\vec{x} = \vec{b}$ then the $(k+1)^{th}$ estimate is:

$$D\vec{x}^{k+1} = b - (L+U)\vec{x}^k \qquad \text{(Jacobi Method)}$$

Since D is diagonal ($D_{ij} = \delta_{ij} A_{ij}$), we can write the vector equation above for $\vec{x}^{k+1}$ for each component $(x_1^{k+1}, ...x_n^{k+1})$.

$$x_i^{k+1} = \underbrace{\frac{1}{A_{ii}}}_{D^{-1}} \left(b_i - \underbrace{\sum_{j \neq i} A_{ij} x_j^k}_{\text{L+U part}} \right), \qquad 1 \leq i \leq n \tag{4.1}$$

4.1.2 Applying the Jacobi method

To start the scheme use an initial guess $\vec{x}^0$, (eg. $\vec{x}^0 = \vec{0}$). The iterations are repeated until $A\vec{x}^k \approx \vec{b}$ or the residual:

$$|\vec{b} - A\vec{x}^k| \quad < \quad \text{error tolerance} \quad (\text{eg.}10^{-5})$$

Jacobi method *converges* to correct solution $x^k \to x$ as $k \to \infty$ *if*:

$$\|D^{-1}(L+U)\| < 1 \Rightarrow \qquad \underbrace{|A_{ii}| > \sum_{j \neq i} |A_{ij}|}_{\text{A is } \textit{strictly diagonally dominant}} \qquad , \quad 1 \leq i \leq n$$

where $\|B\|$ is the row-sum norm defined below:

$$\|B\| \;=\; \max_{1 \leq i \leq n} \sum_{j=1}^{n} |B_{ij}| \;=\; \max_{1 \leq i \leq n} \sum_{j=1, j \neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|} \quad < 1.$$

The degree to which the convergence criteria:

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}|, 1 \leq i \leq n$$

holds is a measure of how fast the estimate $\vec{x}^k$ converges to actual solution $\vec{x}$.

Look for matlab code on this free source website: `http://www.netlib.org/` which implements the Jacobi method and try for yourself.

4.2 Gauss-Seidel Method

- improves convergence of Jacobi method by simple modification
- in Jacobi method the new estimate, x_i^{k+1} is computed using only the current estimate, x_j^k
- Gauss-Seidel method uses all the possible new estimates ($j \leq i-1$) $x_{i-1}^{k+1}, x_{i-2}^{k+1}, ..., x_1^{k+1}, x_0^{k+1}$ when updating the new estimate x_i^{k+1}:

  $$x_i^{k+1} = \frac{1}{A_{ii}} \left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} A_{ij} x_j^k \right), \qquad 1 \leq i \leq n.$$

  This is an improvement over the Jacobi method because it uses the new estimate x_j^{k+1} when it can. In vector form: $\vec{x}^{k+1} = D^{-1}(b - L\vec{x}^{k+1} - U\vec{x}^k)$ or $(D+L)\vec{x}^{k+1} = b - Ux^k$.
- solution converges $x^k \to x$ as $k \to \infty$ if: $\|(D+L)^{-1}U\| \leq 1$.

4.2.1 Example: using Gauss-Seidel method to solve a matrix equation

The matlab code for this example is **GaussSeidel.m**.

Solve $A\vec{x} = \vec{b}$ for $Res = |b - Ax^{k+1}| < 1e^{-3}$ with:

$$A = \begin{pmatrix} 5 & 0 & -2 \\ 3 & 5 & 1 \\ 0 & -3 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 7 \\ 2 \\ -4 \end{pmatrix}, \quad x^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

with initial guess $\vec{x}^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\Rightarrow$ takes 9 iterations. The Jacobi method needs 17 iterations to converge so takes nearly twice as long as the Gauss-Seidel method.

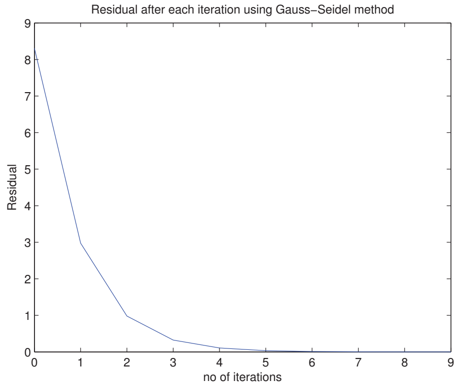


Figure 4.1: Plot of residual using the Gauss-Seidel method after each iteration

Figure 4.1 shows the residual using the Gauss-Seidel method after each iteration, it takes 9 iterations for the residual error to be less than 0.001.

4.3 Relaxation Methods

Relaxation methods generalise Gauss-Seidel method by introducing a relaxation factor, $\alpha > 0$. If α is optimised for the system this can increase the rate of convergence of the solution x^k by modifying the size of the correction:

$$x_i^{k+1} = x_i^k + \frac{\alpha}{A_{ii}} \left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{k+1} - \sum_{j=i}^{n} A_{ij} x_j^k \right), \qquad 1 \leq i \leq n. \tag{4.2}$$

This is called the successive relaxation (SR) method and for:

- $0 < \alpha < 1 \Rightarrow$ under-relaxation
- $\alpha = 1 \Rightarrow$ Gauss-Seidel method
- $\alpha > 1 \Rightarrow$ over-relaxation

We can re-write equation 4.2:

$$x_i^{k+1} = (1-\alpha)x_i^k + \frac{\alpha}{A_{ii}}\left[b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} A_{ij}x_j^k\right], \qquad 1 \le i \le n$$

Solution converges, ie $x^k \to x$ as $k \to \infty$ if: $\|(D+\alpha L)^{-1}[(1-\alpha)D-\alpha U]\| < 1$.