Home Assignments in Numerical Methods

Vasily Arzhanov

HA08 (12p)

Preamble. A finite-difference approximation of a partial differential equation often results in a system of linear algebraic equations, which can in general be written as

$$\mathbf{Ax} = \mathbf{b} \text{ or } \sum_{j=1}^{n} a_{ij} x_j = b_i .$$

The matrix here is characterized by having many zero elements (sparse) and high order n especially in two- and three-dimensional problems. An iterative method often suggests an effective and cheap alternative to so-called direct methods that require $O(n^3)$ arithmetic operations and additional computer memory. Perhaps, the simplest family of iterative methods is so-called one-step (first order) stationary algorithms with the generic form

$$\mathbf{x}^{(k+1)} = \mathbf{Bx}^{(k)} + \mathbf{f}$$

Here, $\mathbf{x}^{(0)}$ is an arbitrary starting vector (guess), and $\mathbf{B}$ is referred to as the iteration matrix. The necessary and sufficient condition for convergence is

$$\rho(\mathbf{B}) < 1 .$$

A systematic way to construct such iterative methods begins with splitting the original matrix $\mathbf{A}$ into a preconditioner $\mathbf{P}$ and the rest $\mathbf{N}$:

$$\mathbf{A} = \mathbf{P} - \mathbf{N} \text{ and } \mathbf{N} = \mathbf{P} - \mathbf{A} .$$

Next, the original system of equations rewrites as follows

$$\mathbf{Px} = \mathbf{Nx} + \mathbf{b} .$$

It is then logical to distribute iterative indices as

$$\mathbf{Px}^{(k+1)} = \mathbf{Nx}^{(k)} + \mathbf{b}$$

Thus, we arrive at

$$\mathbf{x}^{(k+1)} = \mathbf{P}^{-1}\mathbf{Nx}^{(k)} + \mathbf{P}^{-1}\mathbf{b} = \left(\mathbf{I} - \mathbf{P}^{-1}\mathbf{A}\right)\mathbf{x}^{(k)} + \mathbf{f}$$

Clearly, the preconditioner should have two properties: (1) $\mathbf{P}$ must be easily invertible; (2) $\mathbf{P}$ must be as close (in some sense) to $\mathbf{A}$ as possible. Selecting the preconditioner can be based on studying the algebraic structure of the original matrix $\mathbf{A}$. One general way of doing so is to consider $\mathbf{A}$ as consisting of the diagonal, $\mathbf{D}$, lower triangular, $\mathbf{L}$, and upper triangular, $\mathbf{U}$, matrices

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U} .$$

Considered in this course, the finite-difference approximation of the neutron diffusion equation, leads to matrices $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{U}$ all having non-negative elements. The original equation can thus be rewritten as

$$\mathbf{Dx} = \mathbf{Lx} + \mathbf{Ux} + \mathbf{b} .$$

The Jacobi method is based on the following distribution of iteration indices

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

Equivalently, it is defined by

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

The iteration matrix becomes

$$\mathbf{B} = \mathbf{B}_J \equiv \mathbf{D}^{-1}\left(\mathbf{L}+\mathbf{U}\right)$$

Alternatively, in the coordinate form, the Jacobi method runs as

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

The Gauss-Seidel method is based on the following distribution of iteration indices

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

Equivalently, it is defined by

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

The iteration matrix becomes

$$\mathbf{B} = \mathbf{B}_{GS} \equiv \left(\mathbf{D}-\mathbf{L}\right)^{-1}\mathbf{U}$$

Alternatively, in the coordinate form, the Gauss-Seidel method runs as

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

The Successive Over-Relaxation method, SOR, involves an accelerating parameter, ω, and is based on the following distribution of iteration indices

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

Skipping some simple algebra, the iteration matrix becomes

$$\mathbf{B} = \mathbf{B}_{\omega} \equiv \left(\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{L}\right)^{-1}\left[\left(1-\omega\right)\mathbf{I} + \omega\mathbf{D}^{-1}\mathbf{U}\right]$$

Alternatively, in the coordinate form, the SOR method runs as

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

Exercise 1 (1p). Find by hand the first two iterations of the Jacobi method for the following linear system, using $\mathbf{x}^{(0)} = \mathbf{0}$:

$$\begin{cases} 3x_1 & - & x_2 & + & x_3 & = & 1 \\ 3x_1 & + & 6x_2 & + & 2x_3 & = & 0 \\ 3x_1 & + & 3x_2 & + & 7x_3 & = & 4 \end{cases}$$

Exercise 2 (1p). Repeat Exercise 1 using the Gauss-Seidel method.

Exercise 3 (2p). Find by hand the first two iterations of the SOR method with $\omega = 1.1$ for the following linear system, using $\mathbf{x}^{(0)} = \mathbf{0}$:

$$\begin{cases} 3x_1 & - & x_2 & + & x_3 & = & 1 \\ 3x_1 & + & 6x_2 & + & 2x_3 & = & 0 \\ 3x_1 & + & 3x_2 & + & 7x_3 & = & 4 \end{cases}$$

Exercise 4 (2p). The linear system

$$\begin{cases} 2x_1 & - & x_2 & + & x_3 & = & -1 \\ 2x_1 & + & 2x_2 & + & 2x_3 & = & 4 \\ -x_1 & - & x_2 & + & 2x_3 & = & -5 \end{cases}$$

has the solution $(1, 2, -1)^T$.

a) Show that $\rho\left(\mathbf{B}_J\right) = \frac{\sqrt{5}}{2}$.

b) Show that $\rho\left(\mathbf{B}_{GS}\right) = \frac{1}{2}$.

c) What can we say about the convergence?

Exercise 5 (2p). The linear system

$$\begin{cases} x_1 + 2x_2 - 2x_3 = 7 \\ x_1 + x_2 + x_3 = 2 \\ 2x_1 + 2x_2 + x_3 = 5 \end{cases}$$

has the solution $(1, 2, -1)^T$.

a) Show that $\rho(\mathbf{B}_J) = 0$.

b) Show that $\rho(\mathbf{B}_{GS}) = 2$.

c) What can we say about the convergence?

Exercise 6 (2p). Using the Jacobi, Gauss-Seidel and SOR ($\omega = 1.1$) iterative methods, write and execute a computer program to solve the following linear system to four decimal places (rounded) of accuracy:

$$\begin{bmatrix} 7 & 1 & -1 & 2 \\ 1 & 8 & 0 & -2 \\ -1 & 0 & 4 & -1 \\ 2 & -2 & -1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ -5 \\ 4 \\ -3 \end{bmatrix}$$

Compare the number of iterations needed in each case. *Hint*: The exact solution is

$$\mathbf{x} = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}^T$$

Exercise 7 (2p). Solve the following linear system to four decimal places of accuracy using the SOR iterative method with values of $\omega_j = 1 + 0.1j$ where $j = 1, 2, \ldots, 9$.

$$\begin{bmatrix} 7 & 3 & -1 & 2 \\ 3 & 8 & 1 & -4 \\ -1 & 1 & 4 & -1 \\ 2 & -4 & -1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -3 \\ 1 \end{bmatrix}$$

Plot the number of iterations for convergence versus the value of ω. Which value of ω results in the fastest convergence? *Hint*: The exact solution is

$$\mathbf{x} = \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix}^T$$