Training and Workshop on Python Programming and Numerical Methods

An Introduction to Numerical Linear Algebra and Differential Equations

RHUDAINA MOHAMMAD

Numerical Analysis & Scientific Computing Group
Institute of Mathematics, UP Diliman
rzmohammad@up.edu.ph

29-31 January 2025 · Visayas State University, Baybay City, Leyte

– BREAK –

15:00 – 15:30

LINEAR SPARSE SYSTEMS AND ITERATIVE METHODS

Linear Sparse Systems

- A linear system is said to be sparse if the matrix $A \in \mathbb{R}^{n \times n}$ has a number of nonzero entries of order of n (and not n^2).
  - For example, matrix

$$A = \begin{bmatrix} * & * & * & * & * \\ * & * & & & \\ * & & * & & \\ * & & & * & \\ * & & & & * \end{bmatrix}$$

    has $3n - 2$ nonzero entries. After the first step of LU decomposition, the matrix becomes dense with $n(n-1)+1$ nonzero entries.
  - However, reordering rows and columns of matrix

$$A = \begin{bmatrix} * & & & & * \\ & * & & & * \\ & & * & & * \\ & & & * & * \\ * & * & * & * & * \end{bmatrix}$$

    minimizes the number of nonzero entries in its decomposition.

Sparse Systems

Storage formats for sparse matrices can be divided into two groups:

- for efficient modification, such as coordinate (COO) format, which are typically used to contruct matrices
  - COO stores a list of (row, column, value) tuples.
- for efficient access and matrix operations, such as Compressed Sparse Row/Column (CSR/CSC) Format
  - It is similar to COO, but compresses the row/column indices, hence the name
  - CSR stores a sparse $m \times n$ matrix in row form using three 1D arrays: (1) matrix entries, (2) corresponding column indices, and (3) total number of nonzeros above each row

Iterative Methods for Linear Systems

Solve a system of n linear equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= y_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_2 &= y_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= y_n
\end{aligned}
$$

Given a matrix $A = [a_{ij}] \in \mathbb{C}^{n\times n}$ and $y \in \mathbb{C}^n$, find $x \in \mathbb{C}^n$ such that

$$Ax = y$$

- take an initial guess $x_0 \in \mathbb{C}^n$
- construct a sequence of iterates $\{\mathbf{x}_k\} \subset \mathbb{C}^n$ such that

$$x_k \to x, \quad \text{as } k \to \infty$$

Iterative Methods for Linear Systems

- Transform $Ax = y$ into an equivalent fixed-point form.
- Decompose

$$A = D + A_L + A_U$$

into diagonal matrix $D = \operatorname{diag}(a_{11}, \ldots, a_{nn})$ and proper lower and upper triangular matrices

$$A_L = \begin{bmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix} \qquad A_U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

respectively.

Iterative Methods for Linear Systems

- Jacobi Method (Simultaneous Displacements). Suppose D is nonsingular, then linear system $Ax = y$ is transformed into an equivalent form

$$x = -D^{-1}(A_L + A_U)x + D^{-1}y$$

This is solved by successive approximations

$$x_{k+1} = -D^{-1}(A_L + A_U)x_k + D^{-1}y, \qquad k = 0, 1, \ldots$$

with arbitrarily chosen initial x_0

Written in components,

$$x_{(k+1),i} = -\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_{k,j} + \frac{y_i}{a_{ii}}, \qquad i = 1, \ldots, n$$

Iterative Methods for Linear Systems

- Gauss-Seidel Method (Successive Displacements). The linear system $Ax = y$ is transformed into an equivalent form

$$x = -(D + A_L)^{-1} A_U x + (D + A_L)^{-1} y,$$

which is solved by successive approximations

$$x_{k+1} = -(D + A_L)^{-1} A_U x_k + (D + A_L)^{-1} y, \qquad k = 0, 1, \ldots$$

with arbitrarily chosen initial x_0

In actual computations, we solve linear system

$$(D + A_L) x_{k+1} = -A_U x_k + y$$

Written in components,

$$x_{(k+1),i} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_{(k+1),j} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_{k,j} + \frac{y_i}{a_{ii}}, \qquad i = 1, \ldots, n$$

Try this!

Write a code that implements Gauss-Seidel Method. Generate a ”noisy” tridiagonal matrix A of your choice, and a random vector y. Use your code, to solve linear system $Ax = y$, and plot your solution.

– END OF DAY 2 –

THANK YOU SO MU–