More on Linear Systems
Iterative Solutions of Linear Systems

CS3010

Numerical Methods

Dr. Amar Raheja

Section 8.4

Lecture 7

Basic Iterative Methods

- The iterative-method strategy produces a sequence of approximate solution vectors $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$ for system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.
- The numerical procedure is designed so that, in principle, the sequence of vectors converges to the actual solution.
- The process can be stopped when sufficient precision has been attained.
- Select a nonsingular matrix $\boldsymbol{Q}$, and having chosen an arbitrary starting vector $\boldsymbol{x}^{(0)}$, generate vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$. recursively from the equation

$$\boldsymbol{Q}\boldsymbol{x}^{(\boldsymbol{k})} = (\boldsymbol{Q} - \boldsymbol{A})\boldsymbol{x}^{(\boldsymbol{k}-\mathbf{1})} + \boldsymbol{b} \qquad (k = 1,2,\ldots)$$

Iterative Solutions

- To see that this is sensible, suppose that the sequence $\boldsymbol{x}^{(\boldsymbol{k})}$ does converge, to a vector $\boldsymbol{x}^*$, say. Then by taking the limit as $k \to \infty$ in previous equation, we get

$$\boldsymbol{Q}\boldsymbol{x}^* = (\boldsymbol{Q} - \boldsymbol{A})\boldsymbol{x}^* + \boldsymbol{b}$$

If non-singular matrix Q is chosen as coefficient matrix A, then we get

$$\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$$

Starting with the given system of linear equations

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad\qquad (1 \le i \le n)$$

Jacobi Method

- One assumes that all diagonal elements are nonzero.
- If this is not the case, rearrange the equations so that it is.
- Starting with a guess for $x^{(0)}$, solve the *i*^*th*^ equation for the *i*^*th*^ unknown term, we obtain an equation that describes the **Jacobi method**:

$$x^{(k)} = \left[- \sum_{j=1, j\neq i}^{n} \left(a_{ij}/a_{ii}\right) x_j^{(k-1)} + \left(b_i/a_{ii}\right) \right] \qquad (1 \leq i \leq n)$$

Notice that we need diagonal elements to be non-zero because of they are being used for division in the formulation

Jacobi Method Example (1 of 2)

- Use a few iterations of Jacobi to get to solution of this linear system

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$$

The 3 iterative equations are now given as

$$x_1^{(k)} = \frac{1}{2} x_2^{(k-1)} + \frac{1}{2}$$

$$x_2^{(k)} = \frac{1}{3} x_1^{(k-1)} + \frac{1}{3} x_3^{(k-1)} + \frac{8}{3}$$

$$x_3^{(k)} = \frac{1}{2} x_2^{(k-1)} - \frac{5}{2}$$

Consider starting solution as $x^{(0)} = [0\ 0\ 0]^T$

Jacobi Method Example (2 of 2)

- We get the following values for the x column vector as we do more iterations

$$x^{(0)} = [0\ 0\ 0]^T$$
$$x^{(1)} = [0.5000 \quad 2.6667 \quad -2.5000]^T$$
$$x^{(2)} = [1.8333 \quad 2.0000 \quad -1.1667]^T$$

$$\dots$$

$$\dots$$

$$x^{(21)} = [2.0000 \quad 3.0000 \quad -1.0000]^T$$

Gauss-Seidel Method

- In the Jacobi method, the equations are solved in order, but the components $x_j^{(k-1)}$ and the corresponding new values $x_j^{(k)}$ can be used immediately in their place.
- Doing this, we get the iterative solution for Gauss-Seidel Method:

$$x^{(k)} = \left[- \sum_{j=1, j<i}^{n} \left(a_{ij}/a_{ii} \right) x_j^{(k)} - \sum_{j=1, j>i}^{n} \left(a_{ij}/a_{ii} \right) x_j^{(k-1)} + \left(b_i/a_{ii} \right) \right] \quad (1 \leq i \leq n)$$

Gauss-Seidel Method Example (1 of 2)

- Use a few iterations of Gauss-Seidel to get to solution of this linear system

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$$

The 3 iterative equations are now given as

$$x_1^{(k)} = \frac{1}{2} x_2^{(k-1)} + \frac{1}{2}$$

$$x_2^{(k)} = \frac{1}{3} x_1^{(k)} + \frac{1}{3} x_3^{(k-1)} + \frac{8}{3}$$

$$x_3^{(k)} = \frac{1}{2} x_2^{(k)} - \frac{5}{2}$$

Gauss-Seidel Method Example (2 of 2)

- We get the following values for the x column vector as we do more iterations

$$x^{(0)} = [0\ 0\ 0]^T$$
$$x^{(1)} = [0.5000 \quad 2.8333 \quad -1.0833]^T$$
$$x^{(2)} = [1.9167 \quad 2.9444 \quad -1.0278]^T$$

$$\dots$$

$$\dots$$

$$x^{(9)} = [2.0000 \quad 3.0000 \quad -1.0000]^T$$

- As seen, the convergence of the Gauss-Seidel method is approximately twice as fast as that of the Jacobi method.

Gauss-Seidel with SOR

- An acceleration of the Gauss-Seidel method is possible by the introduction of a relaxation factor ω, resulting in the **successive overrelaxation (SOR) method**:

$$x_i^{(k)} = \omega\left[-\sum_{j=1,j<i}^{n}\left(a_{ij}/a_{ii}\right)x_j^{(k)} - \sum_{j=1,j>i}^{n}\left(a_{ij}/a_{ii}\right)x_j^{(k-1)} + \left(b_i/a_{ii}\right)\right] + (1-\omega)x_i^{(k-1)} \qquad (1 \le i \le n)$$

- The SOR method with ω = 1 reduces to the Gauss-Seidel method.

Gauss-Seidel with SOR Example

- Repeat the preceding example using the SOR iteration with ω = 1.1

$$x_1^{(k)} = \omega\left[\frac{1}{2}x_2^{(k-1)} + \frac{1}{2}\right] + (1-\omega)x_1^{(k-1)}$$
$$x_2^{(k)} = \omega\left[\frac{1}{3}x_1^{(k)} + \frac{1}{3}x_3^{(k-1)} + \frac{8}{3}\right] + (1-\omega)x_2^{(k-1)}$$
$$x_3^{(k)} = \omega\left[\frac{1}{2}x_2^{(k)} - \frac{5}{2}\right] + (1-\omega)x_3^{(k-1)}$$

As seen below, the convergence of the SOR method is faster than that of the Gauss-Seidel method.

$$x^{(0)} = [0\ 0\ 0]^T$$
$$x^{(1)} = [0.5500 \quad 3.1350 \quad -1.0257]^T$$
$$x^{(2)} = [2.2193 \quad 3.0574 \quad -0.9658]^T$$

…

…

$$x^{(7)} = [2.0000 \quad 3.0000 \quad -1.0000]^T$$

Convergence of Jacobi and Guass-Seidel

- To converge with any starting x vector, the coefficient matrix A should satisfy the condition of being strictly diagonally dominant i.e.

$$|a_{ii}| > \left(\sum_{j=1, j\neq i}^{n} |a_{ij}|\right) \text{ for all } i, \text{and}$$

$$|a_{ii}| > \left(\sum_{j=1, j\neq i}^{n} |a_{ij}|\right) \text{ for at least one } i$$

Some linear systems will still converge even if this condition is not satisfied, but if this is satisfied, it will surely converge.

Example

$$\begin{bmatrix} 7 & 1 & -1 & 2 \\ 1 & 8 & 0 & -2 \\ -1 & 0 & 4 & -1 \\ 2 & -2 & -1 & 6 \end{bmatrix}$$

Does this matrix meet the convergence criteria? Check both conditions

Convergence Check for Error: Use Norms

- What is a vector norm?
- The simplest vector norms are

$$\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n| \qquad \textbf{(}\boldsymbol{\ell_1}\textbf{-vector norm)}$$

$$\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \qquad \textbf{(Euclidean/}\boldsymbol{\ell_2}\textbf{-vector norm)}$$

$$\|\boldsymbol{x}\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\} \qquad \textbf{(}\boldsymbol{\ell_\infty}\textbf{-vector norm)}$$

- Here, x_i denotes the *i*th component of the vector. Any norm can be thought of as assigning a *length* to each vector.
- For example, if we know that $\| x - y \|_\infty < 10^{-8}$, then we know that each component of x differs from the corresponding component of y by at most 10^{-8} and that the converse is also true

Norms Example

- Determine the l_2 and l_∞ norms of the vector x=[−1 1 −2]^t.

$$||\boldsymbol{x}||_2 = \sqrt{(-1)^2 + (1)^2 + (-2)^2} = \sqrt{6}.$$

$$||\boldsymbol{x}||_\infty = \max\{|-1|, |1|, |-2|\} = 2.$$

Convergence Check for Error: Use Norms

- Calculate the error as defined by the following formula that mirrors approx. relative error but using norms of vectors. The vector $x^{(k)}$ is the solution at the current iteration and $x^{(k-1)}$ at the previous iteration, then the approx. relative error can be determined by the following

$$Error = \frac{\left\| x^{(k)} - x^{(k-1)} \right\|_2}{\left\| x^{(k)} \right\|_2} < \varepsilon$$

Calculate the errors for these iterations:

$$x^{(0)} = [0\ 0\ 0]^T$$
$$x^{(1)} = [0.5000 \quad 2.8333 \quad -1.0833]^T$$
$$x^{(2)} = [1.9167 \quad 2.9444 \quad -1.0278]^T$$