Methods to solve Linear systems ($A\bar{x} = \bar{b}$)

Chandhini G

Department of Mathematical and Computational Sciences
National Institute of Technology Karnataka

March 8, 2021

System of linear equations - solution methods

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - *LU* factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

Matrix operations I

- Our interest: Linear system $A\bar{x} = \bar{b}$ - with n equations and n unknowns - A is a sqaure matrix.
- Following operations are termed as Elementary operations on a matrix.
  1. Interchanging of two rows or columns of a matrix $(R_i \leftrightarrow R_j, \;\; C_i \leftrightarrow C_j)$.
  2. Multiply all elements of a row/column by a non-zero constant K $(R_i \rightarrow KR_i, \;\; C_i \rightarrow KC_i)$.
  3. To K_1 times elements of a row/column, we add K_2 times elements of another row/column $(R_i \rightarrow K_1R_i + K_2R_j, \;\; C_i \rightarrow K_1C_i + K_2C_j)$.

Matrix operations II

- Echelon form: A matrix is in echelon form, if the number of zeroes preceding the first non-zero entry in each row increases row by row until only zero rows remain.

  Eg:- $\begin{bmatrix} 2 & 4 & 1 & 3 \\ 0 & 6 & 1 & -2 \\ 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 5 & -1 & -6 \\ 0 & -1 & 4 & 3 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

  Note: Given matrix A can be converted into equivalent echelon form by elementary transformations.

- Rank of a matrix A and augmented matrix $[A, \bar{b}]$.

  Definition: Let A be a non-zero matrix of order $m \times n$. A positive integer r is said to the rank of A, if the following conditions are satisfied: (i) A has at least one non-zero minor of order r. (ii) Every minor of A whose order $\geq \; r$ is equal to zero.

Matrix operations III

Properties: (i) $Rank(A) \leq \min(m,n)$, (ii) For a square matrix A of order n, $Rank(A) \leq n$, (iii) $Rank(A) = n \Leftrightarrow det(A) \neq 0$, (iv) A and A^T have same rank, (v) $Rank(A)$ is not altered under elementary transformations.
How to find rank: Convert A into echelon form (E) by elementary transformations. Then $Rank(A) = Rank(E) =$ number of non-zero rows in $E = r$.

- Consistency of a $A\bar{x} = \bar{b}$: A linear system is said to be consistent, if it has a solution.
  1. If $Rank(A) = Rank([A|b]) = n$, then solution is unique.
  2. If $Rank(A) = Rank([A|b]) < n$,then there are infinitely many solutions for the system.
  3. If $Rank(A) \neq Rank([A|b])$, then system is not consistent.

Outline

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - LU factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

How to solve a system of linear equations?

1. Trial and Error (Guess and check by substitution): $\begin{cases} x+y=1 \\ x-y=1 \end{cases}$
2. Reducing the number of equations (In fact, elimination!!!):
$$\begin{cases} x+\ \ y+\ \ z=1 \\ x-2y+\ \ z=2 \\ x+\ \ y-2z=0 \end{cases} \implies \begin{cases} 3x+3z=4 \\ 3x-3z=2 \end{cases} \implies \begin{cases} 6x=6 \end{cases}.$$
(Is that the only way to reduce the number of equations???)
Find x, then evaluate z and obtain y from one of the original equations.
3. What happens, if the system is of size 100×100? How easy a random elimination process???

Example - Gauss Elimination I

Go-to A systematic way of doing this elimination process is called *Gaussian Elimination* by "Johann Carl Friedrich Gauss (1810)"

$$\begin{cases} 2u + v + w = 5 \\ 4u - 6v = -2 \\ -2u + 7v + 2w = 9 \end{cases}$$

$Eqn\ 2 \to Eqn\ 2\ -\ (2) \times\ Eqn1$
$Eqn\ 3 \to Eqn\ 3\ -\ (-1) \times\ Eqn1$

$$\begin{cases} 2u + v + w = 5 \\ -8v - 2w = -12 \\ 8v + 3w = 14 \end{cases}$$

$Eqn\ 3 \to Eqn\ 3\ -\ (-1) \times\ Eqn2$

$$\begin{cases} 2u + v + w = 5 \\ -8v - 2w = -12 \\ (1)w = 2 \end{cases}$$

- The numbers 2, -8 and 1 are known as pivot elements of the elimination (First leading entries in row 1, 2 and 3 in the last system after elimination).
- These pivot elements and the coefficients of the corresponding variables in subsequent equations decides the multipliers 2, -1 and -1 used in above elimination. How?
- From the last system (after completing elimination): $w = 2 \implies v = (-12 + 2 \times w)/(-8) = 1 \implies u = (5 - w - v)/2 = 1$
- Two processes: Forward elimination and Backward substitution.

Example - Gauss Elimination II

In matrix form - used for programming

Include RHS of the equations as extra column

$$\begin{pmatrix} 2 & 1 & 1 & 5 \\ 4 & -6 & 0 & -2 \\ -2 & 7 & 2 & 9 \end{pmatrix} \xrightarrow{R_2 \to R_2 - 2\times R_1, \quad R_3 \to R_3 - (-1)\times R_1}$$

$$\begin{pmatrix} 2 & 1 & 1 & 5 \\ & -8 & -2 & -12 \\ & 8 & 3 & 14 \end{pmatrix} \xrightarrow{R_3 \to R_3 - (-1)\times R_2}$$

$$\begin{pmatrix} 2 & 1 & 1 & 5 \\ & -8 & -2 & -12 \\ & & 1 & 2 \end{pmatrix}$$

Gauss Elimination - Algorithm I

Plan is:

Imagine a 100×100 linear system to be solved. Then you need to do the elimination very systematically so that you/your computer remember the steps as much as possible. Basic steps involved are:

STEP 1 Input augmented matrix of size $n \times (n+1) \quad ([A/\bar{b}])$

STEP 2 First pivot is $a_{11} \neq 0$. Evaluate the multipliers $l_{j1} = \frac{a_{j1}}{a_{11}}, \quad j = 2, \ldots, n$ i.e., a_{11} is the pivot for the first row. Calculate new rows using $a'_{jk} = a_{jk} - l_{j1}a_{1k}, \; k = 1, \ldots, n+1$. This leads to

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \vdots & & \ddots & \vdots & \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ 0 & a'_{22} & \ldots & a'_{2n} & b'_2 \\ 0 & a'_{32} & \ldots & b'_{3n} & b'_3 \\ \vdots & & \ddots & \vdots & . \\ 0 & a'_{n2} & \ldots & a'_{nn} & b'_n \end{pmatrix}$$

Gauss Elimination - Algorithm II

Plan is:

STEP 3 IF $a'_{22} \neq 0$ THEN pivot element for Row 2 is a'_{22} with multipliers $l_{j2} = \frac{a'_{j2}}{a'_{22}},\ j = 3, \ldots, n$. Elimination is done on $(n-1) \times n$ matrix,

$$\begin{pmatrix} a'_{22} & \ldots & a'_{2n} & b'_2 \\ a'_{32} & \ldots & a'_{3n} & b'_3 \\ \vdots & & \ddots & \vdots \\ a'_{n2} & \ldots & a'_{nn} & b'_n \end{pmatrix} \text{ i.e., } a'_{jk} = a_{jk} - l_{j2}a_{2k},\ k = 2, \ldots, n+1$$

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ 0 & a'_{22} & \ldots & a'_{2n} & b'_2 \\ 0 & a'_{32} & \ldots & a'_{3n} & b'_3 \\ \vdots & & \ddots & \vdots & \\ 0 & a'_{n2} & \ldots & a'_{nn} & b'_n \end{pmatrix} \Longrightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & b_1 \\ 0 & a'_{22} & a'_{23} & \ldots & a'_{2n} & b'_2 \\ 0 & 0 & a''_{33} & \ldots & a''_{3n} & b''_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & a''_{n3} & \ldots & a''_{nn} & b''_n \end{pmatrix}$$

STEP 4 STOP elimination process once Row n is reached.

- Gauss Elimination Method:

$$A\bar{x} = \bar{b} \xrightarrow{\text{Elem. oper.}} U\bar{x} = \bar{b}^* \xrightarrow{\text{Back subst}} (x_n \implies x_{n-1} \implies \cdots x_1)$$

i.e.,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\xrightarrow{\text{Elem. oper.}}$$

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^* \\ b_2^* \\ \vdots \\ b_n^* \end{bmatrix}$$

$$\xrightarrow{\text{Back subst}} (x_n \implies x_{n-1} \implies \cdots x_1)$$

- $U\bar{x} = \bar{b}^*$ is an equivalent system to $A\bar{x} = \bar{b}$ obtained through elementary operations. Then evaluate the solution starting from last equation for x_n and so on.
- Example: Consider

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 34 \\ 27 \\ -38 \end{bmatrix}$$

$$\xrightarrow{R_2 \to R_2 - 2\times R_1,\quad R_3 \to R_3 - R_1/2,\quad R_4 \to R_4 + R_1} \text{(Pivot element} = 6)$$

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ 21 \\ -26 \end{bmatrix}$$

$\xrightarrow{R_3 \to R_3 - 3\times R_2,\quad R_4 \to R_4 + R_2/2}$ (Pivot element = -4)

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -21 \end{bmatrix}$$

$\xrightarrow{R_4 \to R_4 - 2\times R_3}$ (Pivot element = 2)

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -3 \end{bmatrix}$$

$\xrightarrow{\text{Back subst on}}$

$$x_4 = 1,\ 2x_3 - 5x_4 = -9,\ -4x_2 + 2x_3 + 2x_4 = 10,\ 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12$$

Solution: $\bar{x} = (1,\ -3,\ -2,\ 1)^T$

Failure of GE - Temporary or Permanent I

Example 1:

$$\begin{cases} 3x + 4y + 7z = 6 \\ 6x + 8y + 3z = 7 \quad \text{pivot} = 3, \ \text{multipliers } l_{21} = 2, \ l_{31} = \frac{1}{3} \\ x + 2y + z = 2 \end{cases}$$

$$\begin{cases} 3x + 4y + 7z = 6 \\ -11z = -5 \quad \text{pivot} = 0. \ \div \text{ by zero in multiplier } (l_{32}) \text{ calculations!} \\ \frac{2}{3}y - \frac{4}{3}z = 0 \end{cases}$$

Solution: Row exchange - Exchange the pivot row with a row below that!

Row 2 $\leftrightarrow$ *Row* 3

$$\begin{cases} 3x + 4y + 7z = 6 \\ \frac{2}{3}y - \frac{4}{3}z = 0 \quad \text{Backward substitution: } z = \frac{5}{11}, \ y = \frac{10}{11}, \ x = \frac{-3}{11} \\ -11z = -5 \end{cases}$$

Temporary failure can be corrected by row exchanges.

Failure of GE - Temporary or Permanent II

Example 2:

$$\begin{cases} 3x + 4y + 7z = 6 \\ 5x + 8y + 9z = 10 \\ \ \ x + 2y + \ \ z = 2 \end{cases} \text{pivot} = 3, \ \text{multipliers } l_{21} = \frac{5}{3}, \ l_{31} = \frac{1}{3}$$

$$\begin{cases} 3x + \ 4y + \ 7z = 6 \\ \qquad \frac{4}{3}y - \frac{8}{3}z = 0 \\ \qquad \frac{2}{3}y - \frac{4}{3}z = 0 \end{cases} \text{pivot} = \frac{4}{3}, \ \text{multipliers } l_{32} = \frac{1}{2}$$

$$\begin{cases} 3x + \ 4y + \ 7z = 6 \\ \qquad \frac{4}{3}y - \frac{8}{3}z = 0 \end{cases} \text{Two eqns. Can't simplify further through elimination}$$

In this example, infinitely many solution exist. What if $b_2 = 9$ in place of 10?

Permanent failure of Gauss elimination!

What can be concluded? I

- Gauss elimination (including forward elimination and backward substitution) helps to find the unique solution of $A\bar{x} = \bar{b}$, if exist. (Include row exchange at every step in above GE algorithm to ensure pivot elements are non-zero.)
- Singular case: Gauss elimination break down! - Permanently. Two situations can happen.
  1. System has "infinitely many solutions", if there are no pivot element (pivot element is a non-zero element) in the last column of the transformed (with row operations) augmented matrix$[A/\bar{b}]$.
  2. System has "no solution", if last column of the transformed $[A/\bar{b}]$ has a pivot element.

What can be concluded? II

Elemetary row operations:

1. Add a non-zero scalar multiple of one row to another: $Row_i \to Row_i + l_{ji}Row_j, \quad i \neq j$
2. Multiply a row by a non-zero scalar factor: $Row_i \to cRow_i, \quad c \neq 0$
3. Interchange a pair of rows: $Row_i \leftrightarrow Row_j$

Remarks:

1. Two matrices are row equivalent to each other, if each can be obtained from the other by applying a sequence of permitted row operations.
2. Let two linear systems be represented by their augmented matrices. If these two augmented matrices are row equivalent to each other, then the solutions of the two systems are identical.

Pivoting I

- Consider

$$\begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \implies \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1-10^{20} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2-10^{20} \end{bmatrix}$$

$$\implies x_2 = \tfrac{2-10^{20}}{1-10^{20}} \approx 1, \quad x_1 = (1-x_2)10^{20} \approx 0.$$

- But actual solution is $(x_1,\ x_2) = (1,\ 1)$.

- Another example:

$$\begin{bmatrix} 1 & 10^{20} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10^{20} \\ 2 \end{bmatrix} \implies \begin{bmatrix} 1 & 10^{20} \\ 1 & 1-10^{20} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10^{20} \\ 2-10^{20} \end{bmatrix}$$

$$\implies x_2 = \tfrac{2-10^{20}}{1-10^{20}} \approx 1, \quad x_1 = 10^{20} - 10^{20}x_2 \approx 0$$

- Again, solution is wrong!

Pivoting II

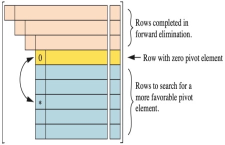
- Solution to above issue is: Pivoting.
  1. Partial pivoting:
     1st stage of elimination: Search for largest (in magnitute) element in the first column and interchanging the corresponding row having largest first element with first row and proceed for elimination.
     2nd stage of elimination: Search for largest (in magnitute) element in the second column (starting from 2nd row) and interchanging the corresponding row having largest second element with second row and proceed for elimination.

     $$\vdots$$

     $$\vdots$$

     kth stage of elimination: Search for largest (in magnitute) element in the k^{th} column (starting from k^{th} row) and interchanging the corresponding row having largest k^{th} element with k^{th} row and proceed for elimination.
  2. Above process is continued till $n-1^{th}$ row.

Pivoting III



To minimize the effect of roundoff, always choose the row that puts the largest pivot element on the diagonal, i.e., find i_p such that $|a_{i_p,i}| = max(|a_{k,i}|)$ for $k = i, \cdots, n$.

- Example: Solve the following after applying partial pivoting.

$$\begin{aligned} 2x_1 + 3x_2 - 6x_3 &= 1 \\ x_1 - 6x_2 + 8x_3 &= 2 \\ 3x_1 - 2x_2 + x_3 &= 3 \end{aligned}$$

Pivoting IV

$$\left[A/\bar{b}\right] = \left[\begin{array}{ccc|c} 2 & 3 & -6 & 1 \\ 1 & -8 & 1 & 2 \\ 3 & -2 & 1 & 3 \end{array}\right] \xrightarrow{R_1 \leftrightarrow R_3} \left[\begin{array}{ccc|c} 3 & -2 & 1 & 3 \\ 1 & -8 & 1 & 2 \\ 2 & 3 & -6 & 1 \end{array}\right]$$

$$\xrightarrow{R_2 \to R_2 - R_1/3, \quad R_3 \to R_3 - 2R_1/3} \left[\begin{array}{ccc|c} 3 & -2 & 1 & 3 \\ 0 & -16/3 & 23/3 & 1 \\ 0 & 13/3 & -20/3 & -1 \end{array}\right]$$

$$\xrightarrow{R_3 \to R_3 + 13R_2/16} \left[\begin{array}{ccc|c} 3 & -2 & 1 & 3 \\ 0 & -16/3 & 23/3 & 1 \\ 0 & 0 & -7/16 & -3/16 \end{array}\right]$$

- Back substitution:

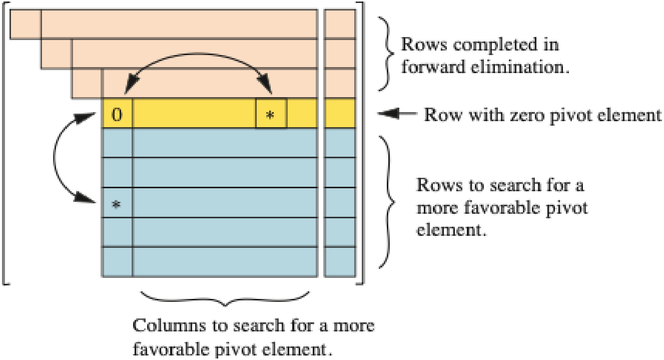
$$x_3 = \frac{(-3/16)}{(-7/16)} = \frac{3}{7}$$

$$\frac{16x_2}{3} + \frac{23x_3}{3} = 1 \implies x_2 = \frac{3}{7}$$

$$3x_1 - 2x_2 + x_3 = 3 \implies x_1 = \frac{8}{7}$$

Pivoting V

- **Complete pivoting:**
  Exchange between rows as well as columns! Compare the computation expense w.r.t partial pivoting!



- Column exchange requires changing the order of the unknown variables x_i's.

Pivoting VI

- For the better numerical stability, complete pivoting identify the largest possible pivot element by searching all rows below pivot row and columns right to the pivot column.
- Full pivoting is less susceptible to the roundoff erros, but at a cost of more complex algorithm due to the increased work associated with searching and data movement.
- Example: Solve the following after applying complete pivoting.

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

Pivoting VII

$$\left[A/\bar{b}\right] = \left[\begin{array}{ccc|c} 1 & -2 & 1 & 1 \\ 0 & 2 & 2 & 4 \\ -2 & 4 & 2 & 2 \end{array}\right] \xrightarrow{R_1 \leftrightarrow R_3} \left[\begin{array}{ccc|c} -2 & 4 & 2 & 2 \\ 0 & 2 & 2 & 4 \\ 1 & -2 & 1 & 1 \end{array}\right] \xrightarrow{C_1 \leftrightarrow C_2} \left[\begin{array}{ccc|c} 4 & -2 & 2 & 2 \\ 2 & 0 & 2 & 4 \\ -2 & 1 & 1 & 1 \end{array}\right]$$

$$\xrightarrow{R_2 \to R_2 - \frac{1}{2} R_1, \quad R_3 \to R_3 - (-\frac{1}{2}) R_1} \left[\begin{array}{ccc|c} 4 & -2 & 2 & 2 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 2 & 2 \end{array}\right]$$

- Complete back substitution and obtain the solution
- Note: Gauss elimination algorithm is modified (computationally) to suit a system with tridiagonal matrix(what is a tridiagonal matrix? What is that algorithm?)

Outline

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - LU factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

Elementary Matrices E_{ij}

1. What does identity matrix I does on A, if multiplied?
2. Can a row operation on A be represented using a matrix-matrix multiplication?
3. Elementary matrix E_{ij} is obtained by replacing ij^{th} element (zero element!) of I by $-l_{ij}$.
4. Pre-multiplying E_{ij} with A does the subtraction of $l_{ij}(Row_j)$ from Row_i according to GE.
5. For eg: If $m = 3$, $E_{31} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -l_{31} & 0 & 1 \end{pmatrix}$

Permutation matrices P_{ij}

1. Row exchange: P_{ij} derived from I, by exchanging the ith and jth rows.
2. Pre-multiplying P_{ij} with A exchanges the i-th and j-th rows of A.
3. For $m = 3$, $P_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
4. Product of two permutation matrices is again a permutation matrix.
5. What about inverse of a permutation matrix P?

Outline

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - *LU* factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

Without row exchanges I

Consider the first GE example

- Step 1 elimination: $Row_2 \to Row_2 - 2Row_1, \quad Row_3 \to Row_3 - (-1)Row_1$
- Elementary matrices: $E_{21} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad E_{31} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$
- Step 2 elimination: $Row_3 \to Row_3 - (-1)Row_2$
- Elementary matrices: $E_{32} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$
- What does $E_{32}E_{31}E_{21}A$ becomes?
- Can I change the order of these multiplications by elementary matrices? Is $E_{31}E_{21} = E_{31}E_{21}$? Is $E_{32}E_{31} = E_{31}E_{32}$?

Without row exchanges II

- $E_{32}E_{31}E_{21}A = U$. It is equivalent to elimination process (GE) on A (Not taken $\bar{b}$ into account!)
- What is the format of the product $E = E_{32}E_{31}E_{21}$? No structure or specific structure?
- How do we get back A from U?
- For eg: How to revert $newRow_2 \rightarrow Row_2 - 2Row_1$?
  This way: $Row_2 \rightarrow newRow_2 + 2Row_1$? What is the corresponding elementary matrix, say F_{21}?
- Are E_{21} and F_{21} connected?

Without row exchanges III

$$\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\0&1&1\end{pmatrix}}_{E_{32}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\1&0&1\end{pmatrix}}_{E_{31}}\underbrace{\begin{pmatrix}1&0&0\\-2&1&0\\0&0&1\end{pmatrix}}_{E_{21}}\underbrace{\begin{pmatrix}2&1&1\\4&-6&0\\-2&7&2\end{pmatrix}}_{A}=\underbrace{\begin{pmatrix}2&1&1\\0&-8&-2\\0&0&1\end{pmatrix}}_{U}$$

$$\underbrace{\begin{pmatrix}1&0&0\\2&1&0\\0&0&1\end{pmatrix}}_{F_{21}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\-1&0&1\end{pmatrix}}_{F_{31}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\0&-1&1\end{pmatrix}}_{F_{32}}\underbrace{\begin{pmatrix}2&1&1\\0&-8&-2\\0&0&1\end{pmatrix}}_{U}=\underbrace{\begin{pmatrix}2&1&1\\4&-6&0\\-2&7&2\end{pmatrix}}_{A}$$

- i.e., $F_{21}F_{31}F_{32} = (E_{32}E_{31}E_{21})^{-1}$ and $F_{21} = E_{21}^{-1}, F_{31} = E_{31}^{-1}, F_{32} = E_{32}^{-1}$

  Triangular factorization: $\boxed{A = LU}$ $\boxed{L = F_{21}F_{31}F_{32}}$
- Diagonals of L are ones and diagonals of U are the pivots.
- In this example: $L = F_{21}F_{31}F_{32} = \begin{pmatrix}1&0&0\\l_{21}&1&0\\l_{31}&l_{32}&1\end{pmatrix}$

Without row exchanges IV

- All multiples used in elimination are the elements of L!!!
- How to solve for $\bar{x}$? we have not applied row operations on RHS $\bar{b}$!

$$\boxed{A\bar{x} = \bar{b} \implies LU\bar{x} = \bar{b}}$$

Solve two triangular systems in the order

$$\boxed{L\bar{y} = \bar{b}; \quad U\bar{x} = \bar{y}}$$

- If A remains same, but $\bar{b}$ changes (in any mathematical model), GE provides both L and U. Only solution need to be found for every changing RHS vector! (Order of n^2 operations!)
- We can also write as $A = LDU$, where D is the diagonal matrix with pivots on the diagonals, L and U are lower and upper triangular matrices with unit diagonal entries. (Is U in the factorization same as that in $A = LU$?)
- $A = LU$ and $A = LDU$ factorizations are unique!

With row exchanges I

For example,

$$A = \begin{pmatrix} 3 & 4 & 7 \\ 6 & 8 & 3 \\ 1 & 2 & 1 \end{pmatrix} \to \begin{pmatrix} 3 & 4 & 7 \\ 0 & 0 & -11 \\ 0 & \frac{2}{3} & -\frac{4}{3} \end{pmatrix} \to \begin{pmatrix} 3 & 4 & 7 \\ 0 & \frac{2}{3} & -\frac{4}{3} \\ 0 & 0 & -11 \end{pmatrix}$$

- Step 1: Multipliers - $l_{21} = 2;\ l_{31} = \frac{1}{3}$
- Step 2: Row exchange? Answer: Pre-multiply by appropriate permutation matrix. $P_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
- Then what happens to $A = LU$ factorization???
- In this example: What we get is $P_{23}A = LU$, where $L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}$

  (See where the multipliers are placed!!!)

With row exchanges II

What if more row exchanges are done on A at different stages of eliminatation?

- Consider $A = \begin{pmatrix} 0 & 3 & 2 \\ 0 & 0 & 4 \\ 2 & 4 & 9 \end{pmatrix}$
- $Row_1 \leftrightarrow Row_3$: $P_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$
- Elimination: $l_{21} = l_{31} = 0$ (No elimination at step 1 after row exchange)
- $Row_2 \leftrightarrow Row_3$: $P_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
- Elimination: $l_{32} = 0$ (No elimination at step 2 after row exchange)
- $U = \begin{pmatrix} 2 & 4 & 9 \\ 0 & 3 & 2 \\ 0 & 0 & 4 \end{pmatrix}$

With row exchanges III

- What we get $P_{23}P_{13}A = LU$ ($L = I$ in this example)

For a nonsingular matrix A, there is a permutation matrix P that reorders the rows of A to avoid zeros in the pivot positions.Then $A\bar{x} = \bar{b}$ has a unique solution. With the rows reordered in advance, PA can be factored into LU

- In practice, we cannot reorder in advance. Still it is possible to obtain correct $P,\ L,\ U$ matrices so that $PA = LU$. How?

- $A\bar{x} = \bar{b} \Rightarrow LU\bar{x} = \bar{b}$, where $L = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}$ and

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{bmatrix}$$

- After factorization, Solve $L\bar{y} = \bar{b}$ using forward substitution and $U\bar{x} = \bar{y}$ using back substitution.

- Iterative methods for linear systems are especially useful, when the system is large and sparse. (Sparse matrix is a matrix in which most of the elements are zero. By contrast, if most of the elements are nonzero, then the matrix is considered dense).
- Starting with an initial guess vector $\bar{x}^{(0)} = (x_1^{(0)},\ x_2^{(0)}, \cdots, x_n^{(0)})^T$, iterative methods generate a sequence of vectors $\{\bar{x}^{(k)}\}$ that converges to actual solution $\bar{x} = (x_1,\ x_2, \cdots, x_n)^T$.
- i.e., These methods, in general, converts $A\bar{x} = \bar{b}$ into an equivalent system of the form $\bar{x} = G\bar{x} + \bar{c}$, for some fixed matrix G and vector $\bar{c}$. Then the iterative methods take the form

$$\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}, \quad k = 0, 1, \cdots\cdots$$

Linear equations:

- Suppose $A\bar{x} = \bar{b}$ is written as,

$$
\begin{aligned}
a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\
a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= b_2 \\
\vdots \qquad\qquad \vdots \quad &\quad \vdots \\
a_{n-1,1}x_1 + a_{n-1,2}x_2 + \cdots + a_{n-1,n}x_n &= b_{n-1} \\
a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n &= b_n
\end{aligned}
$$

Rewriting the linear equations:

- Write each of the above equations in the form:

$$
\begin{aligned}
x_1 &= [b_1 - (a_{1,2}x_2 + \cdots + a_{1,n}x_n)]/a_{1,1} \\
x_2 &= [b_2 - (a_{2,1}x_1 + a_{2,3}x_3 \cdots + a_{2,n}x_n)]/a_{2,2} \\
\vdots \quad & \quad \vdots \qquad\qquad\qquad\qquad \vdots \\
x_{n-1} &= [b_{n-1} - (a_{n-1,1}x_1 + \cdots + a_{n-1,n-2}x_{n-2} + a_{n-1,n}x_n)]/a_{n-1,n-1} \\
x_n &= [b_n - (a_{n,1}x_1 + \cdots + a_{n,n-2}x_{n-2} + a_{n,n-1}x_{n-1})]/a_{n,n}
\end{aligned}
$$

- In general, i^{th} equation in the system is written as,

$$
x_i = \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j \right] \Big/ a_{ii}, \quad i = 1, 2, \ldots, n. \tag{1}
$$

Outline

Jacobi method I

$$x_i^{(k+1)} = \left[b_i - \sum_{\substack{j=0 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right] \bigg/ a_{ii}, \quad i = 1, 2, \ldots, n. \tag{2}$$

Gauss-Siedel method I

$$x_i^{(k+1)} = \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right] \Big/ a_{ii}, \quad i = 1, 2, \ldots, n. \quad (3)$$

- To compute $(k+1)^{th}$ iterates, Jacobi method uses only previous iterative values (k^{th}), while Gauss-Siedel method uses latest iterates.

  i.e., While computing $x_i^{(k+1)}$, Gauss-Siedel method substitutes $x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$ and $x_{i+1}^{(k)}, x_{i+2}^{(k)}, \cdots, x_n^{(k)}$. But Jacobi substitutes all values available at k^{th} iteration $(x_1^{(k)}, x_2^{(k)}, \cdots, x_{i-1}^{(k)}, x_{i+1}^{(k)}, \cdots, x_n^{(k)})$.
- Example:

  $9x - y + 2z = 9; \quad x + 10y - 2z = 15; \quad -2x + 2y + 13z = 17$ with $(x^{(0)}, y^{(0)}, z^{(0)}) = (1, 1, 1)$.
- Stopping criteria:

  $\max(|x^{(k+1)} - x^{(k)}|, |y^{(k+1)} - y^{(k)}|, |z^{(k+1)} - z^{(k)}|) < 10^{-4}$

- Jacobi iterates:

k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	1	1	1
1	0.888888888888889	1.600000000000000	1.307692307692308
2	0.887179487179487	1.672649572649573	1.198290598290598
3	0.919563152896486	1.650940170940171	1.186850756081525
4	0.919693184308569	1.645413835926657	1.195172766454818
5	0.917229811446336	1.647065234860107	1.196042976674141
6	0.917219920167981	1.647485614190195	1.195409934859420
7	0.917407304941262	1.647359994955086	1.195343739381198
8	0.917408057354743	1.647328017382113	1.195391893844027

- Gauss-Siedel iterates:

k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	1	1	1
1	0.888888888888889	1.611111111111111	1.196581196581197
2	0.913105413105413	1.648005698005698	1.194630725399956
3	0.917638249689532	1.647162320111038	1.195457835319768
4	0.917360738830167	1.647355493180937	1.195385422407574
5	0.917398294262866	1.647337255055228	1.195394006031944

Outline

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - LU factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

- Residual vector: Suppose $\bar{x}^* \in \mathbb{R}^n$ is an approximate solution of the linear system defined by $A\bar{x} = \bar{b}$. The residual vector $\bar{r}^*$ with respect to $A\bar{x} = \bar{b}$ is $\bar{r}^* = \bar{b} - A\bar{x}^*$.
- Note: In Jacobi or Gauss-Siedel methods, the objective is to generate a sequence of approximations that will cause residual vectors $\bar{r}^*$ to converge to zero vector $(0, 0, \dots, 0)^T$.

An acceleration of Gauss-Siedel method is possible by the introduction of a relaxation factor ω resulting in successive over/under relaxation.

- Derivation: Rewriting Gauss-Siedel iteration as foloows:

$$\begin{aligned} x_i^{(k+1)} &= \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right], \quad i = 1, 2, \ldots, n. \\ &= \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right] + x_i^{(k)}, \quad i = 1, 2, \ldots, n \\ &= \frac{r_i^{(k+1)}}{a_{ii}} + x_i^{(k)}, \quad i = 1, 2, \ldots, n. \end{aligned}$$

i.e.,

$$x_i^{(k+1)} = x_i^{(k)} + \frac{r_i^{(k+1)}}{a_{ii}}, \quad i = 1, 2, \ldots, n. \qquad (4)$$

i.e., At each Gauss-Siedel iteration, solution is improved by adding the normalized residual value to previous iterative value. Successive over/under relaxation is done through a small modification on above formula by multiplying the residual value $\frac{r_i^{(k+1)}}{a_{ii}}$ by a factor $\omega > 0$. i.e.,

$$x_i^{(k+1)} = x_i^{(k)} + \omega \frac{r_i^{(k+1)}}{a_{ii}}, \quad i = 1, 2, \ldots, n. \tag{5}$$

- If $0 < \omega < 1$, method is called “under relaxation" which is useful (sometimes) in getting converged solutions when Gauss-Siedel iteration diverges.
- If $\omega > 1$ method is called “over relaxation" which is useful in accelerating the convergence of Gauss-Siedel method.

- Use the following formula for calculation purposes:

$$
\begin{aligned}
x_i^{(k+1)} &= x_i^{(k)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right], \quad i = 1, 2, \ldots, n \\
x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right], \\
&\qquad i = 1, 2, \ldots, n.
\end{aligned}
$$

- Comaparison between Gauss-Siedel and SOR ($\omega = 1.25$).
  $4x_1 + 3x_2 = 24;\ 3x_1 + 4x_2 - x_3 = 30;\ -x_2 + 4x_3 = -24;$
  $\bar{x}^{(0)} = (1,\ 1,\ 1)$
- SOR:
$$\begin{aligned} x_1^{(k+1)} &= (1-\omega)x_1^{(k)} + \frac{\omega}{4}(24 - 3x_2^{(k)}) \\ x_2^{(k+1)} &= (1-\omega)x_2^{(k)} + \frac{\omega}{4}(30 - 3x_1^{(k+1)} + x_3^{(k)}) \\ x_3^{(k+1)} &= (1-\omega)x_3^{(k)} + \frac{\omega}{4}(-24 + x_2^{(k+1)}) \end{aligned}$$
- Gauss-Siedel:
$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}(24 - 3x_2^{(k)}) \\ x_2^{(k+1)} &= \frac{1}{4}(30 - 3x_1^{(k+1)} + x_3^{(k)}) \\ x_3^{(k+1)} &= \frac{1}{4}(-24 + x_2^{(k+1)}) \end{aligned}$$
- Calculations are independent for Gauss-Siedel and SOR

Table: Gauss-Siedel iteration (4 more iterations to get 10^{-04} accuracy)

k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	1.000000000000000	1.000000000000000	1.000000000000000
1	5.250000000000000	3.812500000000000	-5.046875000000000
2	3.140625000000000	3.882812500000000	-5.029296875000000
3	3.087890625000000	3.926757812500000	-5.018310546875000
4	3.054931640625000	3.954223632812500	-5.011444091796875
5	3.034332275390625	3.971389770507812	-5.007152557373047
6	3.021457672119141	3.982118606567383	-5.004470348358154
7	3.013411045074463	3.988824129104614	-5.002793967723846
8	3.008381903171539	3.993015080690384	-5.001746229827404
9	3.005238689482212	3.995634425431490	-5.001091393642128
10	3.003274180926383	3.997271515894681	-5.000682121026330
11	3.002046363078989	3.998294697434176	-5.000426325641456
12	3.001278976924368	3.998934185896360	-5.000266453525910
13	3.000799360577730	3.999333866185225	-5.000166533453694

Table: SOR iteration: $\omega = 1.25$

k	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$
0	1.000000000000000	1.000000000000000	1.000000000000000
1	6.312500000000000	3.519531250000000	-6.650146484375000
2	2.622314453125000	3.958526611328125	-4.600423812866211
3	3.133302688598633	4.010264635086060	-5.096686348319054
4	2.957051232457161	4.007483826950192	-4.973489716998301
5	3.003721104119904	4.002924971588072	-5.005713517129152
6	2.996327563106206	4.000926192588054	-4.998282185533945
7	3.000049803672148	4.000258577930990	-5.000348648013079
8	2.999745132271660	4.000065341508485	-4.999892418775328
9	3.000002459267881	4.000014978191950	-5.000022214621183
10	2.999985343128076	4.000003054200322	-4.999993491907104

Outline

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - *LU* factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

- Split the coefficient matrix A in the following form

$$
\begin{aligned}
A &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \\
&= \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & -a_{nn} \end{bmatrix} \\
&\quad - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \cdots & -a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}
\end{aligned}
$$

i.e., $A = D - L - U$

- Jacobi method: Write the Jacobi iteration as

$$a_{ii}x_i^{(k+1)} = \left[b_i - \sum_{\substack{j=0\\ j\neq i}}^{n} a_{ij}x_j^{(k)}\right], \qquad i = 1, 2, \ldots, n.$$

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix}$$

$$= \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} - \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}$$

$$\text{i.e.,} \quad D\bar{x}^{(k+1)} = \bar{b} + (L+U)\bar{x}^{(k)}, \quad k = 0, 1, 2, \ldots$$

$$\text{i.e.,} \quad \bar{x}^{(k+1)} = D^{-1}\bar{b} + D^{-1}(L+U)\bar{x}^{(k)}, \quad k = 0, 1, 2, \ldots$$

$$\text{i.e.,} \quad \bar{x}^{(k+1)} = \bar{c} + G\bar{x}^{(k)}, \quad k = 0, 1, 2, \ldots$$

where $G = D^{-1}(L+U), \quad \bar{c} = D^{-1}\bar{b}$.

- Gauss-Siedel method: Write the Gauss-Siedel iteration as

$$a_{ii}x_i^{(k+1)} = \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right], \qquad i = 1, 2, \ldots, n.$$

$$a_{ii}x_i^{(k+1)} + \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} = \left[b_i - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right], \qquad i = 1, 2, \ldots, n.$$

In vector form

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix}$$

$$= \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} - \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}$$

$$\text{i.e.,} \quad (D-L)\bar{x}^{(k+1)} = \bar{b} + U\bar{x}^{(k)}, \quad k = 0, 1, 2, \ldots$$

$$\text{i.e.,} \quad \bar{x}^{(k+1)} = (D-L)^{-1}\bar{b} + (D-L)^{-1}U\bar{x}^{(k)}, \quad k = 0, 1, 2, \ldots$$

$$\text{i.e.,} \quad \bar{x}^{(k+1)} = \bar{c} + G\bar{x}^{(k)}, \quad k = 0, 1, 2, \ldots$$

where $G = (D-L)^{-1}U, \quad \bar{c} = (D-L)^{-1}\bar{b}$.

- Successive over relaxation method (SOR): From above formula,

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

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

Combining equations for each x_i and writing in matrix-vector notation:

$$\begin{aligned}
D\bar{x}^{(k+1)} - \omega L\bar{x}^{(k+1)} &= (1-\omega)D\bar{x}^{(k)} + [\bar{b} + U\bar{x}^{(k)}] \\
(D-\omega L)\bar{x}^{(k+1)} &= [(1-\omega)D + \omega U]\bar{x}^{(k)} + \omega\bar{b} \\
\bar{x}^{(k+1)} &= (D-\omega L)^{-1}[(1-\omega)D + \omega U]\bar{x}^{(k)} \\
&\qquad + \omega(D-\omega L)^{-1}\bar{b} \\
\bar{x}^{(k+1)} &= G\bar{x}^{(k)} + \bar{c}
\end{aligned}$$

where $G = (D-\omega L)^{-1}[(1-\omega)D + \omega U]$ and $\bar{c} = \omega(D-\omega L)^{-1}\bar{b}$.

Outline

1. Recap: Some linear algebra!
2. Direct methods
   - Gauss Elimination
   - LU factoraization, Cholesky factorization etc.
   - Triangular Factorization
3. Iterative methods
   - Jacobi and Gauss-Siedel methods
   - Successive over relaxation method (SOR)
4. Convergence of iterative methods
   - In the form $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$
   - Jacobi, Gauss-Siedel and SOR
5. Newton's method for system of nonlinear equations

- Norm of a vector $\bar{x} \in \mathbb{R}^n$: We call $\|.\| : \mathbb{R}^n \to [0, \infty)$, norm of $\bar{x}$, if the following conditions are satisfied.
  1. $\|\bar{x}\| \geq 0$. $\|\bar{x}\| = 0$, if and only if, $\bar{x} = \bar{0}$.
  2. $\|\alpha\bar{x}\| = |\alpha|\|\bar{x}\|$
  3. $\|\bar{x} + \bar{y}\| \leq \|\bar{x}\| + \|\bar{y}\|$

  Eg: $\|\bar{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$, $\|\bar{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$,
  $$\|\bar{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$$
- Norm of a matrix $A \in \mathbb{R}^n \times \mathbb{R}^n$: We call $\|.\| : \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$, a matrix norm, if the following conditions are satisfied.
  1. $\|A\| \geq 0$. $\|A\| = 0$, if and only if, A is a matrix with all entries zero.
  2. $\|\alpha A\| = |\alpha|\|A\|$
  3. $\|A + B\| \leq \|A\| + \|B\|$
  4. $\|AB\| \leq \|A\|\|B\|$

- Theorem: For a vector norm $\|.\|$ in $\mathbb{R}^n$, $\|A\| = \max_{\|x\|=1} \|Ax\|$ is a matrix norm.

  Examples: $\|A\|_2 = \max_{\|\bar{x}\|_2=1} \|A\bar{x}\|, \quad \|A\|_\infty = \max_{\|\bar{x}\|_\infty=1} \|A\bar{x}\| = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$

- Eigen values and eigen vectors:
  - If A is a square matrix of order n, the characteristic polynomial of A is defined by $p(\lambda) = det(A - \lambda I)$. It is a polynomial of degree n.
  - If $det(A - \lambda I) = 0$, then the homogeneous linear system $(A - \lambda I)\bar{x} = 0$ has infinitely many non-trivial (non-zero) solutions $(\bar{x} \neq 0)$. i.e., For each root of λ of $det(A - \lambda I) = 0$, there exists at least a vector $\bar{x}$ so that $(A - \lambda I)\bar{x} = 0$. Then λ is called eigen value and $\bar{x}$ is called corresponding eigen vector of A.
  - Finding eigen value and vector:
    Step 1: Solve $det(A - \lambda I) = 0$ to get $\lambda_1, \; \lambda_2, \ldots, \lambda_n$.
    Step 2: Get each eigen vector by solving $(A - \lambda_i I)\bar{x}_i = 0$.

- Definition: The Spectral radius $\rho(A)$ of a matrix A is defined by $\rho(A) = \max_{1 \le i \le n} |\lambda_i|$
- Theorem: The following statements are equivalent for a matrix A:
  1. A is a convergent matrix.
  2. $\lim_{n\to\infty} \|A^n\| = 0$, for some natural norm.
  3. $\lim_{n\to\infty} \|A^n\| = 0$, for all natural norm.
  4. $\rho(A) < 1$.
- Note: $\rho(A) \le \|A\|$, for any matrix norm $\|.\|$. Eigen values and norms are important in understanding the convergence of iterative methods discussed above.
- Conditions for convergence: If $\bar{x}^{(k+1)}$ and $\bar{x}$ are $(k+1)^{th}$ iterate value using $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}$ and $\bar{x}$ is the actual solution of $A\bar{x} = \bar{b}$. Then, $\bar{x} = G\bar{x} + \bar{c}$ is also satisfied. Hence,

$$\bar{e}^{(k+1)} = \bar{x}^{(k+1)} - \bar{x} = G(\bar{x}^{(k)} - \bar{x}) = G\bar{e}^{(k)} \tag{6}$$

Above relation implies convergence depends on iteration matrix G of the iterative method.

- Theorem (Necessary and sufficient condition): For any $\bar{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\bar{x}^{(k)}\}$ defined by $\bar{x}^{(k+1)} = G\bar{x}^{(k)} + \bar{c}, \quad k \geq 1$ converges to a unique solution of $\bar{x} = G\bar{x} + \bar{c}$, if and only if, $\rho(G) < 1$.
- Definition (Diagonal dominance): A square matrix A is said to be diagonally dominant, if $|a_{ii}| \geq \sum_{\substack{j=0 \\ j\neq i}}^{n} |a_{ij}|$. A is strictly diagonally dominant if $|a_{ii}| > \sum_{\substack{j=0 \\ j\neq i}}^{n} |a_{ij}|$.
- Theorem (Sufficient condition): If A is strictly diagonally dominant, both the Jacobi and Gauss-Siedel methods give sequences $\{\bar{x}^{(k)}\}$ that converge to the solution of $A\bar{x} = \bar{b}$ for any starting vector $\bar{x}^{(0)}$.
- Theorem: If $a_{ii} \neq 0, \quad i = 1, 2, \ldots, n$, then $\rho(G) \geq |\omega - 1|$ for SOR iteration matrix G.
  Note: This implies that SOR method can converge only of $0 < \omega < 2$.
- If A is a tridiagonal matrix with $\lambda_j > 0, \ \forall j$, then $\rho(G_\omega) = [\rho(G_J)]^2 < 1$ and optimal value for ω is $\frac{2}{1+\sqrt{1-[\rho(G_J)]^2}}$

- Verification of many of the above properties are computationally very expensive, except possibly checking diagonal dominance. Hence check for diagonal dominance to ensure convergence of Jacobi and Gauss-Siedel convergence.

- Example - 1: Consider $\begin{bmatrix} 1 & -a \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, where a is a real constant. For what values of a, do Jacobi and Gauss-Siedel methods converge for any $\bar{x}^{(0)}$?

  $G_J = D^{-1}(L+U) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \left\{ \begin{bmatrix} 0 & 0 \\ a & 0 \end{bmatrix} + \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix} \right\} = \begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$

  $G_g = (D-L)^{-1}U = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ a & 0 \end{bmatrix} \right\}^{-1} \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & a \\ 0 & a^2 \end{bmatrix}$

  $|G_J - \lambda I = 0 \implies \lambda = a, -a$ and $|G_g - \lambda I = 0 \implies \lambda = 0, a^2$. Jacobi method to converges, if $\rho(G_J) < 1 \implies |a| < 1$

  Gauss-Siedel converges, if $\rho(G_g) < 1 \implies |a^2| < 1 \implies |a| < 1$

- Example - 2: Find the optimal value of ω so that SOR method converges for the following system:

$$\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ 30 \\ -24 \end{bmatrix}$$

$G_J = D^{-1}(L+U)$

$$= \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} 0 & -3 & 0 \\ -3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -3/4 & 0 \\ -3/4 & 0 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}$$

$det(G_J - \lambda I) = 0 \implies \lambda = 0, +\sqrt{0.625}, -\sqrt{0.625}$

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho(G_J)^2}} \approx 1.24$$

- Example - 3: Importance of diagonal dominance

$$\begin{bmatrix} 1 & 17 & -2 \\ 2 & 2 & 18 \\ 30 & -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 48 \\ 30 \\ 48 \end{bmatrix}$$

Take $\bar{x}^{(0)} = (0, 0, 0)$.
Jacobi iterations:
$(0, 0, 0), \quad (48, 15, 16), \quad (-175, -177, -454), \quad (2149, 4276, 1648)$
Gauss-Siedel iterations: $(0, 0, 0), \quad (48, -33, -486), \quad (-363, 4752, 6814)$
After doing partial pivoting - Step 1: $R_1 \leftrightarrow R_3$, Step 2: $R_2 \leftrightarrow R_3$

$$\begin{bmatrix} 30 & -2 & 3 \\ 1 & 17 & -2 \\ 2 & 2 & 18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 48 \\ 48 \\ 30 \end{bmatrix}$$

Jacobi iterations:
$(0.0000, 0.0000, 0.0000), (1.6000, 2.8235, 1.6667)$
$(1.6216, 2.9255, 1.1752), (1.6775, 2.8664, 1.1614)$
$(1.6749, 2.8615, 1.1618), (1.6746, 2.8617, 1.1626)$
$(1.6745, 2.8618, 1.1626), (1.6745, 2.8618, 1.1626)$
Gauss-Siedel iterations:
$(0.0000, 0.0000, 0.0000), (1.6000, 2.7294, 1.1856)$
$(1.6634, 2.8652, 1.1635), (1.6747, 2.8619, 1.1626)$
$(1.6745, 2.8618, 1.1626), (1.6745, 2.8618, 1.1626)$

- Always check for diagonal dominance before proceeding to solve using above iterative schemes.

- **Nonlinear system - Pair of equations with two variables:**

$$f_1(x,y)=0, \quad f_2(x,y)=0$$

Let (α,β) is actual solution of above simultaneous nonlinear equations. Let (x_0,y_0) is the initial guess. Then (using Taylor series expansion as follows and neglecting second and higher order terms),

$$\begin{aligned}
f_1(\alpha,\beta)=0 &= f_1(x_0+(\alpha-x_0), y_0+(\beta-y_0)) && (7)\\
&= f_1(x_0+h_x, y_0+h_y) && (8)\\
0 &\approx f_1(x_0,y_0)+h_x\frac{\partial f_1}{\partial x}+h_y\frac{\partial f_1}{\partial y} && (9)\\
f_2(\alpha,\beta)=0 &= f_2(x_0+(\alpha-x_0), y_0+(\beta-y_0)) && (10)\\
&= f_2(x_0+h_x, y_0+h_y) && (11)\\
0 &\approx f_2(x_0,y_0)+h_x\frac{\partial f_2}{\partial x}+h_y\frac{\partial f_2}{\partial y} && (12)
\end{aligned}$$

Solving (9) and (12) for h_x and h_y,

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}_{(x_0,y_0)} \begin{bmatrix} h_x \\ h_y \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}_{(x_0,y_0)}$$

- The matrix $J = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}$ is called Jacobian matrix of f_1 and f_2.
- Next iteration: $x_1 = x_0 + h_x; \quad y_1 = y_0 + h_y$
- Generalising $x_{k+1} = x_k + h_x^k; \quad y_{k+1} = y_k + h_y^k, \quad k = 0, 1, \ldots$
- (h_x, h_y) needs to be calculated at each iteration by solving a linear system given below.

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}_{(x_k,y_k)} \begin{bmatrix} h_x^k \\ h_y^k \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}_{(x_k,y_k)}$$

- Example: $f_1(x,y) = x^2 - y^2 - 3; \quad f_2(x,y) = x^2 + y^2 - 13$. Using appropriate (x_0, y_0) calculate

$$J_k = \begin{bmatrix} 2x_k & -2y_k \\ 2x_k & 2y_k \end{bmatrix} \text{and} \begin{bmatrix} h_x^k \\ h_y^k \end{bmatrix} = J_k^{-1} \begin{bmatrix} x_k^2 - y_k^2 - 3 \\ x_k^2 + y_k^2 - 13 \end{bmatrix}$$

  and

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} h_x^k \\ h_y^k \end{bmatrix}, \quad k = 0, 1, \dots$$

- For nonlinear system of n equations and n unknowns, above method can be generalised: $\bar{x}^{k+1} = \bar{x}^k - J_k^{-1} \bar{f}_k$
  with $\bar{x} = [x_1, x_2, \dots, x_n]^T$, $\bar{f} = [f_1(\bar{x}), f_2(\bar{x}), \dots, f_n(\bar{x})]^T$ and

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

References I

[1] K E Atkinson, *An Introduction to Numerical Analysis*, 2nd Edn, John-Wiley and Sons.

[2] R L Burden, J Douglas Faires, *Numerical Analysis*, 9th Edn, Brooks/Cole.

[3] B S Grewal, *Higher Engineering Mathematics*, 42nd Edn, Khanna Publishers.

[4] M K Jain, S R K Iyengar, R K Jain, *Numerical methods for scientific and engineering computations*, New age international (P) Ltd.

[5] S S Shastry, *Introductory Methods of Numerical Analysis*, 4th End, Prentice-Hall of India Pvt Ltd.

Thank You