Chap 2: Solving sets of Equations

- Metrics and vectors
- Elimination methods
- Inverse of matrix
- Ill-conditioned systems
- Iterative methods

Matrices and vectors

- **Matrix notation**
- **Operations on matrices**
- **Linear system in matrix form**
- **Inner product, outer product**
- **Unit vector, zero vector**
- **Diagonal matrix**
- **Identity matrix**
- **Transposition matrix**
- **Permutation matrix**

Matrices and vectors

- **Transposition matrix**
  - **Two rows of an identity matrix are interchanged**

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad A = \begin{bmatrix} 9 & 6 & 2 & 13 \\ 4 & 2 & 8 & 1 \\ 0 & 7 & 1 & 9 \\ 3 & 2 & 6 & 8 \end{bmatrix}$$

$$P_1 A = \begin{bmatrix} 9 & 6 & 2 & 13 \\ 3 & 2 & 6 & 8 \\ 0 & 7 & 1 & 9 \\ 4 & 2 & 8 & 11 \end{bmatrix} \qquad AP_1 = \begin{bmatrix} 9 & 13 & 2 & 6 \\ 4 & 8 & 8 & 2 \\ 0 & 9 & 1 & 7 \\ 3 & 11 & 6 & 2 \end{bmatrix}$$

Row interchanged   Column interchanged

Matrices and vectors

- **Permutation matrix**
  - **Multiplication of several transposition matrices**
- **Symmetric matrix: $a_{ij}=a_{ji}$**
- **Transpose of A matrix**
  - **Writing the rows as columns**

If A is symmetric, then $A = A^T$

For any matrix, $(A^T)^T = A$ and $(AB)^T = B^T A^T$

- **Trace of a square matrix**

$\mathrm{tr}(A) =$ sum of diagonal elements

$\mathrm{tr}(A) = \mathrm{tr}(A^T)$

Matrices and vectors

- Lower triangular matrix: $a_{ij}=0$, for j>i
- Upper triangular matrix: $a_{ij}=0$, for i>j
- Tridiagonal matrix
  - Has nonzero elements only on the diagonal and in the position adjacent to the diagonal
  - Can be stored as a matrix of nx3

Matrices and vectors

- Determinant of a square matrix: det(A)
  - A 3x3 matrix: spaghetti rule
  - General rule:
    - To expand in terms of the minors of some row or column
  - Triangularize a matrix first before computing the determinant
    - Determinant of a triangular matrix
      - Product of the diagonal elements

Matrices and vectors

- **Characteristic polynomial of a matrix**

λ is an eigenvalue of the natrix A if there is a nonzero vector x such that

$$Ax = \lambda x$$

$$(A - \lambda I)x = 0.$$

The characteristic polynomial:

$$P_A(\lambda) = \det(A - \lambda I)$$

Eigenvalues are the roots of

$$P_A(\lambda) = 0.$$

Engenvector corresponds to an eigenvalue λ: is the nonzero v such that

$$Av = \lambda v.$$

Trace of a matrix A = sum of eigenvalues of A.

If a matrix is triangular, its eigenvalues are equal to the diagonal elements.

Elimination methods

Solving a linear system Ax=b

- **If A is a upper-triangular matrix, the system can be solved by back substitution**

$$\begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -10x_2 + 19x_3 &= 77 \\ -72x_3 &= -216 \end{aligned}$$

Elimination methods

Solving a linear system Ax=b

- **Reduce the coefficient matrix to a upper-triangular matrix**
  - **Elimination method based on elementary row operations**

$$\begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -3x_1 - x_2 + 4x_3 &= 8 \\ x_1 - x_2 + 3x_3 &= 13 \end{aligned} \qquad \begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -10x_2 + 19x_3 &= 77 \\ -x_2 + 11x_3 &= 37 \end{aligned} \qquad \begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -10x_2 + 19x_3 &= 77 \\ -72x_3 &= -216 \end{aligned}$$

Elimination methods

Solving a linear system Ax=b

- **Elementary row operations**
  - Multiply any row of the augmented coefficient matrix by a constant
  - Add the multiple of one row to a multiple of any other row
  - Interchange the order of any two rows if necessary
    - Need to guard against zero multipliers by row interchange

Yield an equivalent linear system, why?

But may have effect on the accuracy of the computed solution!

Elimination methods

Matrix form: work on augmented coefficient matrix A|b

$$\begin{bmatrix} 4 & -2 & 1 \\ -3 & -1 & 4 \\ 1 & -1 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15 \\ 8 \\ 13 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -2 & 1 & 15 \\ -3 & -1 & 4 & 8 \\ 1 & -1 & 2 & 13 \end{bmatrix} \underset{(-1)R_1+4R_3}{\overset{3R_1+4R_2}{\Rightarrow\Rightarrow}} \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -10 & 19 & 77 \\ 0 & -2 & 11 & 37 \end{bmatrix}$$

$$\underset{2R_2-10R_3}{\Rightarrow\Rightarrow} \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -10 & 19 & 77 \\ 0 & 0 & -72 & -216 \end{bmatrix}$$

Elimination methods

Solving a linear system Ax=b

- During triangulation, if a zero is encountered on the diagonal, we cannot use that row to eliminate coefficients below that zero element
  - Need to do row interchange
- If there is a zero on the diagonal after triangulation, the back-substitution fails and there is no solution!

Gaussian elimination

- **Avoids the large coefficients resulting from elimination by subtracting a_{ij}/a_{jj} times the first equation from the i_{th} equation**
  - **Increase precision**
- **Apply *pivoting* to avoid zero multiplier and increase precision**
  - **Complete pivoting**
    - **May require row and column interchange**
    - **Not frequently used**
  - **Partial pivoting**
    - **Require only row interchange**
    - ***Order vector* can be used to keep track of the order of rows when a row interchange is done**

Gaussian elimination

- ***pivoting*** **increases precision. Why?**

In reduction :

$$\text{row}_\text{j} - \frac{a_{ji}}{a_{ii}} \text{row}_\text{i}$$

Elements in row_i has propogated errors.

$$\frac{a_{ji}}{a_{ii}}[a_{ik} + \varepsilon_{ik}] \text{ has less error when } a_{ii} \text{ is large.}$$

- ***pivoting*** **avoids zero diagonal element**

In back substitution :

$$x_i = \frac{1}{u_{ii}} \left(c_i - [u_{i,i+1,}, \ldots, u_{i,n,}] \begin{bmatrix} x_{i+1} + \varepsilon_{i+1} \\ \vdots \\ x_n + \varepsilon_n \end{bmatrix} \right)$$

Gaussian elimination

$$\begin{bmatrix} 4 & -2 & 1 \\ -3 & -1 & 4 \\ 1 & -1 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15 \\ 8 \\ 13 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -2 & 1 & 15 \\ -3 & -1 & 4 & 8 \\ 1 & -1 & 2 & 13 \end{bmatrix} \overset{R_2-(-3/4)R_1}{\underset{R_3-(1/4)R_1}{\Rightarrow\Rightarrow}} \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -2.5 & 4.75 & 19.25 \\ 0 & -0.5 & 2.75 & 9.25 \end{bmatrix}$$

$$\underset{R_3-(-0.5/-2.5)R_2}{\Rightarrow\Rightarrow} \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -2.5 & 4.75 & 19.25 \\ 0 & 0.0 & 1.80 & 5.40 \end{bmatrix}$$

Gaussian elimination
LU decomposition

- **The multipliers can be stored in place of zero**
  - **Form a lower-triangular matrix, called *L*.**

$$\begin{bmatrix} 4 & -2 & 1 & 15 \\ (-0.75) & -2.5 & 4.75 & 19.25 \\ (0.25) & (0.20) & 1.80 & 5.40 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -0.75 & 1 & 0 \\ 0.25 & 0.20 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 4 & -2 & 1 \\ 0 & -2.5 & 4.75 \\ 0 & 0 & 1.80 \end{bmatrix}$$

We can easily varify that

$A = LU$ (when no pivoting is done) WHY??

Gaussian elimination LU decomposition

- **If row interchange is performed,**

  ***A′*=*LU*, where**

  ***A′* is a permutation of the rows of A due to row interchange from pivoting**

Gaussian elimination LU decomposition

- Det(A) = det(LU) = det(L) det(U)

  = det(U)   (∵ det(L)=1)

  = the product of diagonal elements
- Solving Ax=b is equivalent to solving

  LUx=b <-> Ly=b and Ux=y

  - Forward substitution, followed by backward substitution.
  - Useful when solving a number of Ax=b, where A is not changed, i.e.,

    $Ax=b_i$, $i=1,2,..m$

Gaussian elimination

- **Gaussian elimination does the following**
  - It solves the system of linear equation
  - It computes the determinant of a matrix very efficiently

    $$det(A) = (-1)^m u_{11} \cdots u_{nn},$$

    where m the number of row interchange
  - It can provide us with the LU decomposition of the coefficient matrix, in the sense that L*U may give us a permutation of the rows of the original matrix

Gaussian elimination LU decomposition

Row interchange can be expensive.

Order vector: keeps track the order of rows. When a row interchanges is indicated, we only change the corresponding elements in the order vector

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

$$\begin{bmatrix} 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{bmatrix} \overset{\text{row 1}\leftrightarrow\text{row 4}}{\Rightarrow\Rightarrow} \begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

$$\overset{G.E.}{\Rightarrow\Rightarrow} \begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & 1.6667 & 5 & 3.6667 & -4 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix} \overset{\text{row 2}\leftrightarrow\text{row 3}}{\Rightarrow\Rightarrow} \begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 1.6667 & 5 & 3.6667 & -4 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

Gaussian elimination LU decomposition

$$\overset{G.E.}{\Rightarrow\Rightarrow}\begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 0 & 6.8182 & 5.6364 & -9.0001 \\ 0 & 0 & 2.1818 & 3.3636 & -5.9999 \end{bmatrix} \overset{G.E.}{\Rightarrow\Rightarrow}\begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 0 & 6.8182 & 5.6364 & -9.0001 \\ 0 & 0 & 0 & 1.5600 & -3.1199 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.66667 & 1 & 0 & 0 \\ 0.33333 & -0.45454 & 1 & 0 \\ 0.0 & -0.54545 & 0.32 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 6 & 1 & -6 & -5 \\ 0 & -3.6667 & 4 & 4.3333 \\ 0 & 0 & 6.8182 & 5.6364 \\ 0 & 0 & 0 & 1.5600 \end{bmatrix}$$

$$LU = A' = \begin{bmatrix} 6 & 1 & -6 & -5 \\ 4 & -3 & 0 & 1 \\ 2 & 2 & 3 & 2 \\ 0 & 2 & 0 & 1 \end{bmatrix} \qquad \det(A) = (-1)^2 \cdot 6 \cdot -3.6667 \cdot 6.8182 \cdot 1.5600 = -234.0028$$

Gaussian elimination Operational count

- **For augmented matrix [A b]:**

To reduce the elements below the diagonal in column 1:

$$\text{Divisions} = n-1,$$

$$\text{multiplications} = n(n-1)$$

$$\text{Substractions} = n(n-1)$$

For column i:

$$\text{Divisions} = n-i,$$

$$\text{multiplications} = (n-i+1)(n-i)$$

$$\text{Substractions} = (n-i+1)(n-i)$$

Total:

Divisions

$$= \sum_{i=1}^{n-1} n-i = n^2/2 - n/2,$$

multiplications

$$= \sum_{i=1}^{n-1} (n-i+1)(n-i) = n^3/3 - n/3$$

Substractions

$$= \sum_{i=1}^{n-1} (n-i+1)(n-i) = n^3/3 - n/3$$

$$\text{Total}: 2n^3/3 + n^2/2 - 7n/6.$$

Gaussian elimination Gauss-Jordan scheme

- Variants to the Gaussian elimination
  - Back-substitution can be performed by eliminating elements above the diagonal after the triangulation, using elementary row operation upward from the last row
    - The diagonal elements may all be made one as the first step
- Gauss-Jordan scheme
  - The elements above and below diagonal are made zero at the same time
  - Diagonal elements are made ones at the same time, resulting in the identity matrix
  - The column of right-hand side is the solution

Gaussian elimination
Gauss-Jordan method

$$\begin{bmatrix} 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 0 & 1.0001 \\ 0 & 0 & 1 & 0 & 0.3333 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Gaussian elimination
Gauss-Jordan method

$$\begin{bmatrix} 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{bmatrix} \overset{\text{row 1} \leftrightarrow \text{row 4}}{\underset{\substack{\text{divide first row by 6} \\ \text{Reduce the 1st column}}}{\Rightarrow\Rightarrow}} \begin{bmatrix} 1 & 0.1667 & -1 & -0.8333 & 1 \\ 0 & 1.6667 & 5 & 3.3667 & -4 \\ 0 & -3.6667 & 4 & 4.3334 & -11 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

$$\overset{\text{row 2} \leftrightarrow \text{row 3}}{\underset{\substack{\text{Divide new 2nd row by -3.6667} \\ \text{Reduce 2nd column below and} \\ \text{above the diagonal}}}{\Rightarrow\Rightarrow}} \begin{bmatrix} 1 & 0 & -0.8182 & -0.6364 & 0.5 \\ 0 & 1 & -1.0909 & -1.1818 & 3 \\ 0 & 0 & 6.8182 & 5.6364 & -9 \\ 0 & 0 & 2.1818 & 3.3636 & -6 \end{bmatrix} \overset{\substack{\text{Divide 3rd row by 6.8182} \\ \text{and reduce other elements} \\ \text{in 3rd column}}}{\Rightarrow\Rightarrow} \begin{bmatrix} 1 & 1 & 0 & 0.04 & -0.58 \\ 0 & 1 & 0 & -0.280 & 1.56 \\ 0 & 0 & 1 & 0 & -1.32 \\ 0 & 0 & 0 & 1.5599 & -3.12 \end{bmatrix}$$

$$\overset{\substack{\text{Divide 4th row by 1.5599} \\ \text{and reduce other elements} \\ \text{in 4th column}}}{\Rightarrow\Rightarrow} \begin{bmatrix} 1 & 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 0 & 1.0001 \\ 0 & 0 & 1 & 0 & 0.3333 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Gaussian elimination Gauss-Jordan method

- The solution computed by Gauss-Jordan method differs slightly from that obtained with G.E.
  - Round-off errors have been entered in a different way
- Gauss-Jordan method requires almost 50% more operations than G.E.
  - (n^2-n)/2 divisions
  - (n^3-n)/2 multiplications
  - (n^3-n)/2 subtractions
  - Total: n^3 + n^2 - 2n ~ O(n^3), compared to 2n^3/3 + n^2/2 - 7n/6 ~ O(2n^3/3) for G.E.

Gaussian elimination
Scaled partial pivoting

- Partial pivoting without scaling
  - When some rows have coefficients that are very large in comparison to those in other rows, partial pivoting may not give a correct solution
    - Quantities of variables maybe in widely different units
- Scaled partial pivoting
  - Scale each row by its coefficient of largest magnitude first then solve it using G.E.
  - A better way
    - Use original equations, eliminating the round-off that may occur in the scaling
    - Use scaling vector whose elements are elements in each row of largest magnitude

Gaussian elimination Scaled partial pivoting

$$\begin{bmatrix} 3 & 2 & 100 \\ -1 & -3 & 100 \\ 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 105 \\ 102 \\ 2 \end{bmatrix}$$

with correct solution

$x = [1.00, 1.00, 1.00]^T$.

Triangulate without pivoting

$$\begin{bmatrix} 3 & 2 & 100 & 105 \\ 0 & 3.67 & 133 & 135 \\ 0 & 0 & -82.4 & -82.6 \end{bmatrix}$$

$x = [0.939, 1.09, 1.00]^T$.

Scaled partial pivoting:

Scale each row first with the largest magnitude

$$\begin{bmatrix} 0.03 & 0.01 & 1.00 & 1.05 \\ -0.01 & 0.03 & 1.00 & 1.02 \\ 0.5 & 1.00 & -0.50 & 1.00 \end{bmatrix}$$

Now we need interchange row 1 with row 3.

Gaussian elimination Scaled partial pivoting

Using scaling vector

$$\begin{bmatrix} 3 & 2 & 100 \\ -1 & -3 & 100 \\ 1 & 2 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 105 \\ 102 \\ 2 \end{bmatrix}$$

Scaling vector $S = [100, 100, 2]^T$.

Divide each element in first column by corresponding element in S to get

$$R = [0.0300, -0.0300, 0.500]^T$$

and row 1 and 3 should be intercganged.

Interchange elements of S to get

$$S = [2, 100, 100]^T$$

Reducing the1st column :

$$\begin{bmatrix} 1 & 2 & -1 & 2 \\ 0 & 5 & 99 & 104 \\ 0 & -4 & 103 & 99 \end{bmatrix}$$

We divide each element in 2nd column by corresponding element in S to get $R = [1, 0.0500, -0.0400]^T$ and no row intercgang is needed.

Reducing the1st column :

$$\begin{bmatrix} 1 & 2 & -1 & 2 \\ 0 & 5 & 99 & 104 \\ 0 & 0 & 182 & 182 \end{bmatrix}$$

Solution $x = [1.00, 1.00, 1.00]$

Gaussian elimination Using order vector

$$\begin{bmatrix} 4 & -3 & 0 & -7 \\ 2 & 2 & 3 & -2 \\ 6 & 1 & -6 & 6 \end{bmatrix}$$

Order vector $O = [1, 2, 3]^T$.

For column 1: A(3,1) should be the pivot; exchange elements in O to get $O = [3, 2, 1]^T$. In reducing column 1, we use row 3 as pivot row to get

$$\begin{bmatrix} (0.6667) & -3.667 & 4 & -11 \\ (0.3333) & 1.667 & 5 & -4 \\ 6 & 1 & -6 & 6 \end{bmatrix}$$

For column 2:

A(1,2) should be next pivot.

Exchange elements in O to get $O = [3, 1, 2]^T$; row 1 as the next pivot row.

Reducing column 2:

$$\begin{bmatrix} (0.6667) & -3.667 & 4 & -11 \\ (0.3333) & (-0.4545) & 6.8182 & -9 \\ 6 & 1 & -6 & 6 \end{bmatrix}$$

Back - substitution in the order given by the final order vector: first 2, then 1, then 3

Gaussian elimination
Multiple right-hand sides

- When all the right-hand sides are known, we can augment A with those right-hand sides and do the G.E.
- When the right-hand sides are not known in advance
  - Suppose we have solved Ax=b by G.E. – we have A=LU. For a new right-hand side b

    LUx=b <-> Ly=b and Ux=y

    - Forward substitution, followed by backward substitution

Gaussian elimination Tridiagonal systems

- Only those elements on the diagonal and adjacent to the diagonal are nonzero.
- In the Gaussian elimination, only the nonzero elements are used.
  - There is no need to store the zeros
  - Coefficient matrix can be compressed into an array of 3 columns
  - Arithmetic count is reduced significantly

$$\begin{bmatrix} -4 & 2 & 0 & 0 \\ 6 & -3 & 1 & 0 \\ 0 & 7 & -2 & 5 \\ 0 & 0 & 8 & 1 \end{bmatrix}$$

Inverse of matrices

- Division by a matrix is not defined
  - The equivalent is to find the inverse of a matrix
    - If A*B=I, B is said to be the inverse of A (and A is the inverse of B)
- Finding the inverse of a matrix
  - Using determinant
    - Not efficient
  - Use Gauss-Jordan form on [A, I]
    - More expensive than using GE
  - Use Gaussian elimination
    - Apply GE on [A,I]
    - Apply A=LU to solving AX=I

Pathological systems

- **Questions can be asked**
  - **Does every square matrix have an inverse?**
  - **Is there unique solution to the set of equation?**
- **So far we know that if there is zero on the diagonal after elimination, then no unique solution can be found for that system**

$$A=\begin{bmatrix} 1 & -2 & 3 \\ 2 & 4 & -1 \\ -1 & -14 & 11 \end{bmatrix}$$

LU is

$$\begin{bmatrix} 2.0 & 4.0 & -1.0 \\ (-0.5) & -12.0 & 10.5 \\ (0.5) & (0.333) & 0 \end{bmatrix}$$

It means that we can not solve the system and can not find the inverse of A. In this case, we said the matrix A is singular.

When A is singular, A does not have an inverse and $Ax=b$ may have no solution or infinitely many solutions, depending on b.

Pathological systems

- Can we see if a matrix singular A without trying to triangulate it?
  - A singular matrix has a determinant 0
    - The matrix A on last slide has a zero on U, so det(A)=0
  - The rank of the matrix is less than n, the number of rows
  - A singular matrix has rows that are linearly dependent vectors
    - Ex. For matrix A: -3 row1 + 2 row2 + row3=[0,0,0]
  - A singular matrix has columns that are linearly dependent vectors
    - Ex. For matrix A: -10 col1 + 7 col2 + 8 col3 =[0,0,0]
  - Ax=b has no unique solution (no sol. or inf. many sol)

Redundant and inconsistent systems

- **Even though a coefficient matrix A is singular, Ax=b may have no solution or infinitely many solutions for some b.**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 2 & 4 & -1 \\ -1 & -14 & 11 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 7 \\ 1 \end{bmatrix}$$

Apply GE to $[A, b]$, we have

$$\begin{bmatrix} 2.0 & 4.0 & -1.0 & 7.0 \\ (-0.5) & -12.0 & 10.5 & 4.5 \\ (0.5) & (0.333) & 0 & 0 \end{bmatrix}$$

We find that x_3 can be any value. Setting $x_3 = 0$, we have $[17/4, -3/8, 0]$.

Setting $x_3 = 1$, we have $[3, 1/2, 1]$.

We actually have infinitely many solutions!!

Let $x_3 = c$, we can represent x_1 and x_2 as functiona of c, so the solution space will be of dimension 1.

The system is redundant! i.e., any one equation can be a linear combination of the other two.

Redundant and inconsistent systems

- Redundant system (for b=[5, 7, 1])
  - The system has an infinitely many solution
  - Dimension of the solution space of AX=b is 1
    - Any one equation is a linear combination of other equations, so there is a free variable x_3

$$-3[1,-2,3,5]+2[2,4,-1,7]=-1[-1,-14,11,1]$$

- Inconsistent system (for b=[5, 7, 2])
  - No solution satisfies the equations.
    - U(3, 3)=0, but b'(3)=-0.3333

$$\begin{bmatrix} 2.0 & 4.0 & -1.0 & 7.0 \\ (-0.5) & -12.0 & 10.5 & 5.5 \\ (0.5) & (0.333) & 0 & -0.3333 \end{bmatrix}$$

Singular vs. nonsingular matrices

- Singular matrix A
  - It has no inverse
  - Its determinant is zero
  - There is no unique solution to the system Ax=b
  - Gaussian elimination cannot avoid a zero on the diagonal
  - The rank is less than n
  - Rows/columns are linearly dependent

- Nonsingular matrix A
  - It has an inverse
  - Its determinant is nonzero
  - There is a unique solution to the system Ax=b
  - Gaussian elimination does not encounter a zero on the diagonal
  - The rank equals n
  - Rows/columns are linearly independent

Ill-conditioned systems

- Nearly singular coefficient matrix A
  - It is nonsingular, but its U matrix has near-zero elements on diagonal, and its determinant is close to 0
- A system whose coefficient matrix is nearly singular is called ill-conditioned systems
  - Solutions to Ax=b is sensitive to the changes in the elements of b and/or A
    - That is, for small changes in the input (i.e., elements of b or A), we get large changes in the solution
  - This phenomenon shows up even more pointedly in large systems
    - Even the 2x2 system shows the effect of near singularity!

Ill-conditioned systems

$$A = \begin{bmatrix} 3.02 & -1.05 & 2.53 \\ 4.33 & 0.56 & -1.78 \\ -0.83 & -0.54 & 1.47 \end{bmatrix}$$

$$LU = \begin{bmatrix} 4.33 & 0.56 & -1.78 \\ (0.6975) & -1.4406 & 3.7715 \\ (-0.1917) & (0.3003) & -0.0039 \end{bmatrix}$$

See very small element in $U[3,3]$.

$$A^{-1} = \begin{bmatrix} 5.6611 & -7.2732 & -18.5503 \\ 200.5046 & -268.2570 & -669.9143 \\ 76.8511 & -102.6500 & -255.8846 \end{bmatrix}$$

has elements very large in comparison to A.

U has close to 0 diagonal and inv(A) has very large elements compared to A, so A is nonsingular but is almost singular!

Consider $b = [-1.61, 7.23, -3.38]^T$,

Solution to $Ax = b$ is

$x = [1.0000, 2.0000, -1.0000]^T$.

Let's make a small change in just the 1st element of b:

$b_1 = [-1.60, 7.23, -3.38]^T$,

Solution to $Ax = b_1$ is

$x_1 = [1.0566, 4.0051, -0.2315]^T$.

What a difference!

Ill-conditioned systems

Let's consider another small change to b

$$b = [-1.61, 7.22, -3.38]^T,$$

Solutions $x = [1.07271, 4.6826, 0.0265]^T$,

which differs much from the true solution!

When $A[1,1]$ is changed from 3.02 to 3.00,

Solution to $Ax = b$ is

$$x = [1.1277, 6.5221, 0.7333]^T.$$

The system is also sensitive to the error in coeeficient matrix.

Note that we cannot test for the accuracy of the computed solution merely by substituting it into the equation to see if the right-hand sides are reproduced.

$$Ax = b$$

$$Ax_1 = b_1 \approx b$$

$$Ax_2 = b_2 \approx b$$

Ill-conditioned systems

- **Even the systems of two equation suffer from this problem….**

$$\begin{bmatrix} 1.01 & 0.99 \\ 0.99 & 1.01 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2.00 \\ 2.00 \end{bmatrix}$$

$\text{Solution}: x = [1.00, 1.00]^T.$

$\text{For } b_1 = [2.02, 1.98]^T, \text{ solution is}$

$x_1 = [2, 0]^T.$

$\text{For } b_2 = [1.98, 2.02]^T, \text{ solution is}$

$x_2 = [0, 2]^T !!$

$\text{For } b_{\smile} = [2.00, 2.00]^T, \text{ solution is}$

$x_2 = [1, 1]^T !!$

Effect of precision

- **What we can do for ill-conditioned systems?**
  - **Do the calculations in higher precision.**

$$[A,b]=\begin{bmatrix} 3.02 & -1.05 & 2.53 & -1.61 \\ 4.33 & 0.56 & -1.78 & 7.23 \\ -0.83 & -0.54 & 1.47 & -3.38 \end{bmatrix}$$

Using Maple with 10 digits of precision, we get

$x=[1.000000037, 2.000001339, -0.9999994882]$

which is pretty close to the exact solution

$x=[1, 2, -1]$.

With precision of 20, we get a more accurate solution but it is still not exact.

Use only three digits：

$$\begin{bmatrix} 1 & 0 & -.073 & 0 \\ 0 & 1 & -2.62 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Maple says that the matrix is singular!

Actually the system is inconsistent!

Use only four digits：

$$\begin{bmatrix} 1 & 0 & 0 & 0.9824 \\ 0 & 1 & 0 & 1.346 \\ 0 & 0 & 1 & -1.250 \end{bmatrix}$$

We get a poor solution!

Condition numbers and norms

- The degree of ill-conditioning of a matrix is measured by its condition number, which is defined in terms of its norms
  - Norm is a measure of the magnitude of the matrix
    - To measure a single number a, we use its distance from zero: |a|
- For any norm, the following properties are essential:

$$1. \|A\| \geq 0 \text{ and } \|A\| = 0 \text{ iff } A = 0.$$

$$2. \|kA\| = |k| \, \|A\|.$$

$$3. \|A+B\| \leq \|A\| + \|B\|. \text{ (Triangle inequality)}$$

$$4. \|AB\| \leq \|A\| \, \|B\|.$$

Condition numbers and norms
Vector norms

- Norms of a vector
  - Euclidean norm or 2 norm
    - Length of the vector

$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}$$

  - 1-norm
    - Sum of the absolute values of the elements

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

  - Maximum norm (or infinite norm)
    - The maximum value of the elements

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

Condition numbers and norms
Matrix norms

- **Norms of a matrix**
  - **1-norm or maximum column sum**

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

  - **infinite norm or maximum row sum**

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

  - **2-norm (or spectral norm) is not readily computed**
    - **Suppose r is the largest eigenvalue of** $A^T A$

$$\|A\|_2 = r^{1/2}$$

Note that:

$$\|A\|_2 \le \|A\|_1$$

$$\|A\|_2 \le \|A\|_\infty$$

Errors in the solution

- **How large is the errors in the solution?**
  - **Substituting solution to the equation…**
    - **Not good for ill-conditioned systems – can not tell how large the error is**
  - **Use norms**

在這裡鍵入方程式。

Let $\bar{x}$ be the computed solution.

Define the residual $r = b - A\bar{x}$

Let e be the error in $\bar{x}$, $e = x - \bar{x}$.

Because $Ax = b$, we have

$$r = b - A\bar{x} = Ax - A\bar{x}$$

$$= A(x - \bar{x}) = Ae$$

Hence

$$e = A^{-1}r.$$

Taking norms, we have

$$\|e\| = \|A^{-1}r\| \le \|A^{-1}\|\|r\|.$$

Errors in the solution

$\|e\| = \|A^{-1}r\| \le \|A^{-1}\|\|r\|.$

Since $r = Ae$, we have $\|r\| \le \|A\|\|e\|$.

So

$$\frac{\|r\|}{\|A\|} \le \|e\| \le \|A^{-1}\|\|r\|.$$

Applying the same reasoning to $Ax = b$ and $x = A^{-1}b$, we have

$$\frac{\|b\|}{\|A\|} \le \|x\| \le \|A^{-1}\|\|b\|.$$

All together, we have

$$\frac{1}{\|A\|\|A^{-1}\|}\frac{\|r\|}{\|b\|} \le \frac{\|e\|}{\|x\|} \le \|A\|\|A^{-1}\|\frac{\|r\|}{\|b\|}.$$

Condition number

Condition number of a matrix

- Condition number of A=$\|A\|\left\|A^{-1}\right\|$
  - The product of the norm of A and the norm of its inverse
    - A small number means good conditioning
    - A large number means ill-conditioning
  - The relative error in the computed solution can be as **great** as the relative residual multiplied by the condition number

$$\frac{1}{\text{Condition no.}}\frac{\|r\|}{\|b\|} \le \frac{\|e\|}{\|x\|} \le \text{Condition no.}\frac{\|r\|}{\|b\|}.$$

Condition number of a matrix

- **Condition number of A=$\|A\|\|A^{-1}\|$**
  - **The relative error in the computed solution can be as small as the relative residual divided by the condition number.**

$$\frac{1}{\text{Condition no.}}\frac{\|r\|}{\|b\|}\le\frac{\|e\|}{\|x\|}\le\text{Condition no.}\ \frac{\|r\|}{\|b\|}$$

  - **When the condition number is large, the residual gives little information about the accuracy of the solution (condition # dominates)**
  - **When the condition number is nearly unity, the relative residual is a good measure of the relative error of the computed solution**

Condition number of a matrix

- Condition number of A: $\|A\|\|A^{-1}\|$
  - Errors in the coefficients
    - We have already seen that an ill-conditioned system is extremely sensitive to small changes in the coefficients
    - The condition number relates the changes in the solution to such errors in the coefficients
  - Errors in the right-handed side vector
    - We have already seen that an ill-conditioned system is extremely sensitive to small changes in the right-handed side vector
    - The condition number relates the changes in the solution to such errors in the right-handed side vector

Condition number of a matrix

- **Errors in the coefficient matrix**

Let A be the true coefficient matrix, E be the errors. Let $\overline{A} = A + E$.

Let $\overline{x}$ be the computed solution to

$$\overline{A}x = (A+E)x = b,$$

and x be the solution of $Ax = b$,

we have

$$\begin{aligned} x = A^{-1}b = A^{-1}(\overline{A}\overline{x}) &= A^{-1}(A + \overline{A} - A)\overline{x} \\ &= \left[I + A^{-1}(\overline{A} - A)\right]\overline{x} \\ &= \overline{x} + A^{-1}(\overline{A} - A)\overline{x} \\ &= \overline{x} + A^{-1}E\overline{x}. \end{aligned}$$

So $x - \overline{x} = A^{-1}E\overline{x}$, and we have

$$\begin{aligned} \|x - \overline{x}\| = \|A^{-1}E\overline{x}\| &\le \|A^{-1}\|\|E\|\|\overline{x}\| \\ &= \|A^{-1}\|\|A\|\frac{\|E\|}{\|A\|}\|\overline{x}\|, \end{aligned}$$

and

$$\frac{\|x - \overline{x}\|}{\|\overline{x}\|} \le \|A^{-1}\|\|A\|\frac{\|E\|}{\|A\|} = \text{Condition no.} \frac{\|E\|}{\|A\|}$$

Relative error of the solution relative to the computed solution can be as large as the relative error in A multiplied by the condition number.

Condition number of a matrix

- **Errors in right-handed side vector**

Let $\bar{b}$ be the perturbed vector of b.

If x and $\bar{x}$ satisfy $Ax = b$ and $A\bar{x} = \bar{b}$.

$$\|x-\bar{x}\| = \|A^{-1}b - A^{-1}\bar{b}\| = \|A^{-1}(b-\bar{b})\|$$

$$\leq \|A^{-1}\|\|b-\bar{b}\| = \|A^{-1}\|\|Ax\|\frac{\|b-\bar{b}\|}{\|b\|}$$

$$\leq \|A^{-1}\|\|A\|\|x\|\frac{\|b-\bar{b}\|}{\|b\|}$$

So

$$\frac{\|x-\bar{x}\|}{\|x\|} \leq \|A^{-1}\|\|A\|\frac{\|b-\bar{b}\|}{\|b\|}$$

$$= \text{condition no.}\frac{\|b-\bar{b}\|}{\|b\|}$$

Relative error of the solution relative to the true solution can be as large as the relative error in b multiplied by the condition number.

Iterative improvement Residual correction method

- **Computed solution of Ax=b is an approximate solution $\bar{x}$, we can apply iterative improvement to correct $\bar{x}$**

Let $\bar{x}$ be the computed solution of $Ax = b$.

Define $e = x - \bar{x}$ and $r = b - A\bar{x}$.

Since

$$Ae = r,$$

we can solve this equation for e, and apply the computed solution $\bar{e}$ as a correction to $\bar{x}$, i.e., $x = \bar{x} + \bar{e}$.

Repeat this correction until desired accuracy is achieved.

Note:

1. A can be decomposed to LU, and apply LU to solve
   $Ax = b$ and $Ae = r$.
2. The computation of r should be done in higher precision to avoid cancellation error.

Iterative improvement Residual correction method

Solution of $Ae = r$ is also subject to round-off error as solution of $Ax = b.$

Even so, unless the system is so ill-conditioned that $\bar{e}$ is not a reasonable approximate to $e,$ we will get an improved estimate of x from $\bar{x} + \bar{e}.$

Note:

The computation of r must be as accurate as possible. So use double-precision arithmetic.

Iterative improvement
Residual correction method

$$A = \begin{bmatrix} 4.23 & -1.06 & 2.11 \\ -2.53 & 6.77 & 0.98 \\ 1.85 & -2.11 & -2.32 \end{bmatrix},$$

$$b = \begin{bmatrix} 5.28 & 5.22 & -2.58 \end{bmatrix}^T$$

Exact solution:

$$x = [1.000, 1.000, 1.000]^T.$$

Using 3-digit precision

Computed solution:

$$\bar{x} = [0.991, 0.997, 1.000]^T.$$

Compute $r = b - A\bar{x}$

using double precision:

$$A\bar{x} = [5.24511, 5.22246, -2.59032]^T.$$

$$r = [0.0349, -0.00246, 0.0103]^T.$$

We solve $Ae = r$ and get

$$\bar{e} = [0.00822, 0.00300, -0.00000757]^T.$$

Finally,

$$\bar{x} + \bar{e} = [0.999, 1.000, 1.000]^T$$

gives almost exactly the correct solution.

Pivoting and precision

- **Pivoting can**
  - **Avoids zero diagonal elements for nonsingular matrix**
  - **Reduces the errors due to round off**
    - **Only if the problem is mildly ill-conditioned**

$$\begin{cases} \varepsilon\, x + By = C \\ Dx + Ey = F \end{cases}$$

with ε a very small number.

Without pivoting, we get

$$\varepsilon\, x + By = C$$

$$(E - DB/\varepsilon)y = F - CD/\varepsilon$$

Solving for y, we have

$$y = \frac{F - CD/\varepsilon}{E - DB/\varepsilon} \approx \frac{CD}{DB} \quad \text{if } \varepsilon \text{ is very small.}$$

$$x = \frac{C - B(C/B)}{\varepsilon} = \frac{C - C}{\varepsilon} = 0\,!!$$

showing that $x = 0$ for any value of C and F if ε is small enough.

Pivoting and precision Example

Suppose $F = D + E$ and $C = \varepsilon + B$.

With pivoting, we reduce

$$\begin{cases} Dx + Ey = D + E \\ \varepsilon\, x + By = \varepsilon + B \end{cases}$$

to

$$\begin{cases} Dx + Ey = D + E \\ (B - (\varepsilon / D)E)y \\ = \varepsilon + B - (\varepsilon / D)(D + E) \\ = \dfrac{\varepsilon D + BD - \varepsilon D - \varepsilon E}{D} = \dfrac{BD - \varepsilon E}{D} \end{cases}$$

So that

$$y = \frac{(BD - \varepsilon E)/D}{(BD - \varepsilon E)/D} = 1,$$

$$x = \frac{D + E - E}{D} = 1$$

Pivoting and precision Example

- For severely ill-conditioned problems, pivoting alone cannot save the accuracy
  - The best way to remedy is to increase precision of computation

Iterative methods

- **Direct methods**
  - Gaussian elimination
  - Gaussian-Jordan
  - Decomposition methods
    - LU, QR, SVD

- **Iterative methods**
  - Jacobi method
  - Gauss-Seidel method
  - Good for large sparse coefficient matrix
  - Will converge for any starting values if the coefficient matrix is **diagonally dominant**, i.e.,

$$\text{For each } i = 1,2,\ldots,n, \left|a_{ii}\right| > \sum_{\substack{j=1 \\ j\neq i}}^{n} \left|a_{ij}\right|$$

Iterative methods
Jacobi method

$$\begin{cases} 6x_1 - 2x_2 + x_3 = 11 \\ x_1 + 2x_2 - 5x_3 = -1 \\ -2x_1 + 7x_2 + 2x_3 = 5 \end{cases}$$

Solution : $x_1 = 2,\ x_2 = x_3 = 1.$

Rewrite the system:

$$\begin{cases} 6x_1 - 2x_2 + x_3 = 11 \\ -2x_1 + 7x_2 + 2x_3 = 5 \\ x_1 + 2x_2 - 5x_3 = -1 \end{cases}$$

Rearrange the equations :

$$x_1 = 1.8333 + 0.3333x_2 - 0.1667x_3$$

$$x_2 = 0.7143 + 0.2857x_1 - 0.2857x_3$$

$$x_3 = 0.2000 + 0.2000x_1 + 0.4000x_2.$$

Starting from some initial approximation, we iterate based on

$$x_1^{(n+1)} = 1.8333 + 0.3333x_2^{(n)} - 0.1667x_3^{(n)}$$

$$x_2^{(n+1)} = 0.7143 + 0.2857x_1^{(n)} - 0.2857x_3^{(n)}$$

$$x_3^{(n+1)} = 0.2000 + 0.2000x_1^{(n)} + 0.4000x_2^{(n)}$$

Starting with $x^{(0)} = (0, 0, 0)$, at 9th iteration we get $x^{(9)} = (2.00, 1.00, 1.00)$.

Iterative methods
Jacobi method

General form for rearrangement :

$$x_i = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j ,\ \ i = 1,2,\ldots,n.$$

Matrix representation :

Let $A = L + D + U$.

$$Ax = (L + D + U)x = b$$

$$Dx = -(L + U)x + b$$

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

Iterative form :

$$\begin{aligned} x^{(n+1)} &= -D^{-1}(L + U)x^{(n)} + D^{-1}b \\ &= b' - Bx^{(n)} \end{aligned}$$

$$x^{(n+1)} = G(x^{(n)}) = b' - Bx^{(n)}$$

Iterative methods Jacobi method

$$Ax = b,$$

$$\begin{bmatrix} 6 & -2 & 1 \\ -2 & 7 & 2 \\ 1 & 2 & -5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \\ -1 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 2 & 0 \end{bmatrix}, D = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & -5 \end{bmatrix},$$

$$U = \begin{bmatrix} 0 & -2 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

Iterative form:

$$\begin{aligned} x^{(n+1)} &= -D^{-1}(L+U)x^{(n)} + D^{-1}b \\ &= G(x^{(n)}) = b' - Bx^{(n)} \end{aligned}$$

$$b' = D^{-1}b = \begin{bmatrix} 1.8333 \\ 0.7143 \\ 0.2000 \end{bmatrix}$$

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

$$= \begin{bmatrix} 0 & -0.3333 & 0.1667 \\ -0.2857 & 0 & 0.2857 \\ -0.2000 & -0.4000 & 0 \end{bmatrix}$$

Iterative methods
Gauss-Seidel method

- **Similar to Jacobi method, but use always the most recent approximations of the other variables**
- **The rate of convergence is more rapid than for the Jacobi method**

Starting from some initial approximation, we iterate based on

$$x_1^{(k+1)} = 1.8333 + 0.3333x_2^{(k)} - 0.1667x_3^{(k)}$$

$$x_2^{(k+1)} = 0.7143 + \boxed{0.2857x_1^{(k+1)}} - 0.2857x_3^{(k)}$$

$$x_3^{(k+1)} = 0.2000 + \boxed{0.2000x_1^{(k+1)}} + \boxed{0.4000x_2^{(k+1)}}$$

Starting with $x^{(0)} = (0, 0, 0)$, at 6th iteration we get $\boxed{x^{(6)}} = (2.00, 1.00, 1.00)$.

Iterative methods
Gauss-Seidel method

- **Matrix representation**

Matrix representation :

Let $A = L + D + U$.

$Ax = (L + D + U)x = b$

$(L + D)x = -Ux + b$

$x = -(L + D)^{-1}Ux + (L + D)^{-1}b$

Iterative form :

$x^{(k+1)} = -(L + D)^{-1}Ux^{(k)} + (L + D)^{-1}b$

- **The eigenvalues of $D^{-1}(L+U)$ and $(D+L)^{-1}(L+U)$ indicate how fast the iterations will converge**
- **Without diagonal dominance, neither Jacobi nor Gauss-Seidel is sure to converge!**

Iterative methods Convergence issues

- If the coefficient matrix is diagonally dominant, Jacobi and Gauss-Seidel converge for any initial values
- Without diagonal dominance, neither Jacobi nor Gauss-Seidel is sure to converge
  - There are some instances where the coefficient matrix does not have diagonal dominance but still both Jacobi and Gauss-Seidel do converge
  - It can be shown that, if the coefficient matrix A is symmetric and positive definite, the Gauss-Seidel method will converge from any starting values
- When both methods converge, Gauss-Seidel converges faster.

Iterative methods Convergence issues

- Most coefficient matrices are neither diagonally dominant nor symmetric and positive definite, it is suggested to solve as many equations for the variable having the largest coefficient!

Iterative methods Convergence proof

Assume that A is diagonally dominant.

Let $x = (x_1, x_2, .., x_n)$ be the exact solution to $Ax = b$. Then

$$x_i = \frac{1}{a_{ii}} \left\{ b_i - \sum_{j \neq i} a_{ij} x_j \right\},$$

for $i = 1, 2, .., n.$

Let the error of the i-th component of $x^{(k)}$ be

$\varepsilon_i^k = x_i - x_i^{(k)}$, for $i = 1, 2, .., n.$

Then the error of

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

satisfies

$$\begin{aligned} \varepsilon_i^{k+1} &= x_i - x_i^{(k+1)} \\ &= \frac{-1}{a_{ii}} \left\{ \sum_{j \neq i} a_{ij} \left(x_j - x_j^{(k)} \right) \right\} \\ &= \frac{-1}{a_{ii}} \left\{ \sum_{j \neq i} a_{ij} \varepsilon_j^k \right\} \end{aligned}$$

Iterative methods
Convergence issues

So, if $\left|\varepsilon_i\right|_{\max}^{k}$ denotes the largest $\left|\varepsilon_j^k\right|$ for $j \neq i$, then

$$\left|\varepsilon_i^{k+1}\right| = \frac{1}{\left|a_{ii}\right|}\left|\sum_{j\neq i} a_{ij}\varepsilon_j^k\right| \leq \frac{\sum_{j\neq i}\left|a_{ij}\right|\left|\varepsilon_j^k\right|}{\left|a_{ii}\right|}$$

$$\leq \frac{\sum_{j\neq i}\left|a_{ij}\right|}{\left|a_{ii}\right|}\left|\varepsilon_i\right|_{\max}^{k} \leq \delta\left|\varepsilon_i\right|_{\max}^{k}$$

where

$$\delta = \max_{i=1,2,..,n}\left\{\frac{\sum_{j\neq i}\left|a_{ij}\right|}{\left|a_{ii}\right|}\right\}.$$

This implies that $\left|\varepsilon_i^{k+1}\right|$ is smaller than $\left|\varepsilon_i\right|_{\max}^{k}$ by a factor of at least δ. The convergence will therefore be ensured if $\delta < 1$,

i.e., if A is diagonally dominant.

Iterative methods
Accelerating convergence

- **Convergence in the Gauss-Seidel method can be speeded if we do overrelaxation**

Gauss-Seidel iteration:

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

Equivalently,

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

Overrelaxation can be applied to Gauss-Seidel method if we add to some multiple of the second term :

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

w: the overrelaxation factor,

$1 \le w < 2.$

Iterative methods
Accelerating convergence

$$\begin{bmatrix} -4 & 1 & 1 & 1 \\ 1 & -4 & 1 & 1 \\ 1 & 1 & -4 & 1 \\ 1 & 1 & 1 & -4 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

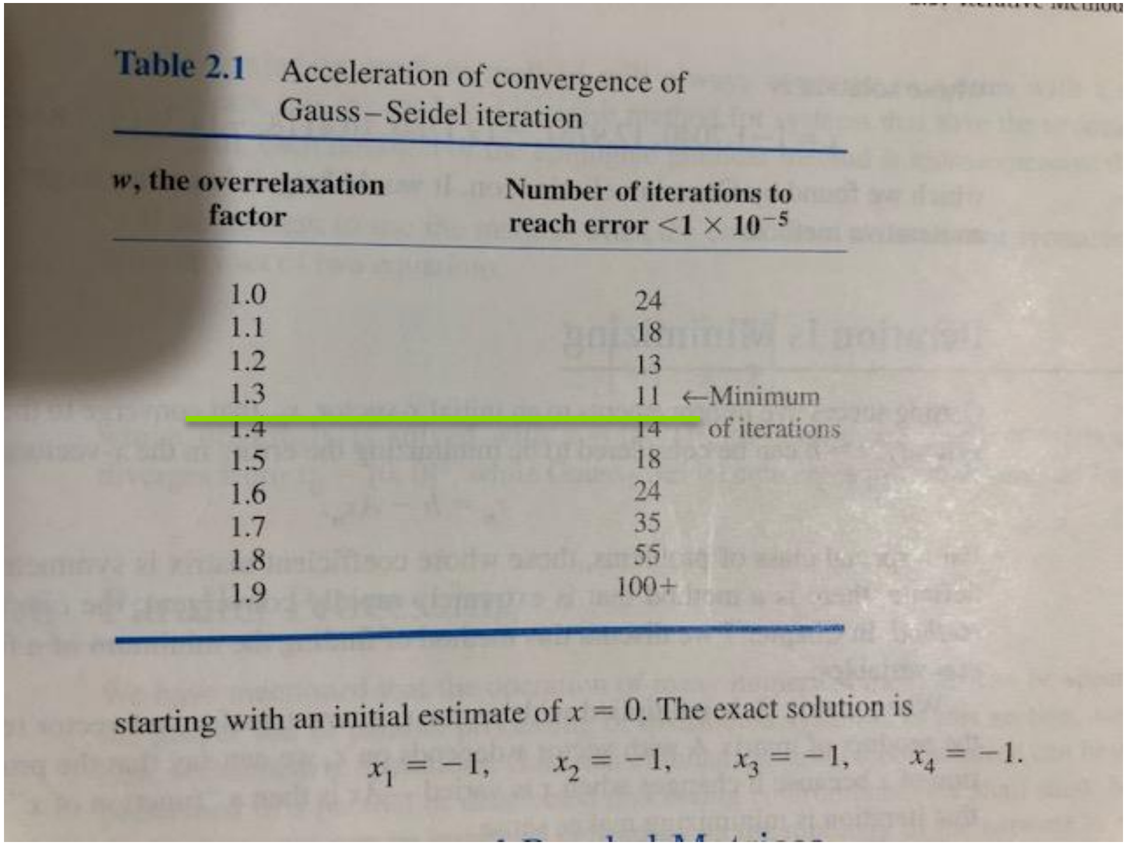

Table 2.1 Acceleration of convergence of Gauss–Seidel iteration

w, the overrelaxation factor	Number of iterations to reach error $<1 \times 10^{-5}$	
1.0	24	
1.1	18	
1.2	13	
1.3	11	←Minimum
1.4	14	of iterations
1.5	18	
1.6	24	
1.7	35	
1.8	55	
1.9	100+	

starting with an initial estimate of $x = 0$. The exact solution is

$$x_1 = -1, \quad x_2 = -1, \quad x_3 = -1, \quad x_4 = -1.$$

Iterative methods
Iteration is minimizing

- Getting successive improvements that converges to the solution to Ax=b can be considered to be minimizing the residual error r(x)=b-Ax
- If A is symmetric and positive definite, conjugate gradient method gives extremely rapidly convergence
  - It always converges in n tries with system of n equations
  - Each iteration of the conjugate gradient is more expensive than Jacobi or Gauss-Seidel

Iterative methods
Iteration is minimizing

$$\begin{bmatrix} 4 & -3 & -1 \\ -3 & 5 & 2 \\ -1 & 2 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \\ -3 \end{bmatrix}$$

Solution :

$[3.9167, 3.5833, -2.0833]^T$

The coefficient matrix is symmetric and positive definite.

Start with $x_0 = [0,0,0]^T$,

Gauss-Seidel converges in 20 iterations, Jacobi fails to converge.

If conjugate gradient is applied with x_0, it converges in three tries:

$x_0 = [0,0,0]^T$,

$x_1 = [2.4520, 0.7006, -1.0508]^T$

$x_2 = [4.0670, 3.4771, -1.6197]^T$

$x_3 = [3.9167, 3.5833, -2.0833]^T$