

Subsections

- [Gauss Elimination](#)
 - [Solving Small Numbers of Equations](#)
 - [Naive Gauss Elimination](#)
 - [Pitfalls of Elimination Methods](#)
 - [Techniques for Improving Solutions](#)
 - [Complex Systems](#)
 - [Nonlinear Systems of Equations](#)
 - [Gauss-Jordan](#)
- [LU Decomposition and Matrix Inversion](#)
 - [LU Decomposition](#)
 - [The Matrix Inverse](#)
 - [Error Analysis and System Condition](#)
- [Special Matrices and Gauss-Seidel](#)
 - [Special Matrices](#)
 - [Gauss-Seidel](#)
 - [Linear Algebraic Equation with Libraries and Packages](#)
- [Engineering Applications: Linear Algebraic Equations](#)

Linear Algebraic and Equations

Matrix notation:

- symmetric matrix
- diagonal matrix
- principal or main diagonal of the matrix
- identity matrix
- upper triangular matrix
- lower triangular matrix
- banded matrix
- tridiagonal matrix
- transpose
- trace

Gauss Elimination

Solving Small Numbers of Equations

Problems when solving sets of linear equations

- Singular
 - no solution
 - infinite solution
- Ill-conditioned : system that are very close to being singular

Cramer's Rule : each unknown in a system of linear algebraic equations may be expressed as a fraction of two determinants with denominator D and with the numerator obtained from D by replacing the column of coefficients of the unknown in question by the constant

b_1, b_2, \dots, b_n . For more than three equations, Cramer's rule becomes impractical because, as the number of equations increases, the determinants are time-consuming to evaluate.

Naive Gauss Elimination

The elimination of unknowns consists of two steps

1. The equations are manipulated to eliminate one of the unknowns from the equations. The result of this elimination step is that we have

- on equation with one unknown.
- 2. This equation can be solved directly and the result back-substituted into one of the original equations to solve for the remaining unknown.

But the above method can't avoid division by zero in computer program. Need more elaborated algorithms !

Pitfalls of Elimination Methods

- Division by zero : partially avoided by the technique of pivoting
- Round-off errors : An error in early steps will tend to propagate-that is, it will cause errors in subsequent steps.
- Ill-conditioned systems : small changes in coefficients result in large changes in the solution. An ill-conditioned system is one with a determinant close to zero. This means that there is no solutions or an infinite number of solutions. However, it is difficult to specify how close to zero the determinant must be to indicate ill-conditioning. This is complicated by the fact that the determinant can be changed by multiplying one or more of the equations by a scale factor without changing the solution. Consequently, the determinant is a relative value that is influenced by the magnitude of the coefficients. One way to partially prevent a scaling effect is to scale the equations so that the maximum element in any row is equal to 1.
- Singular systems : lose one degree of freedom and we would be dealing with the impossible case of $n - 1$ equations with n unknowns. Check the determinant whether it is zero or not !

Techniques for Improving Solutions

- Use more significant figures in the computation
- Use pivoting
 - partial pivoting : make the largest element in the column to be the pivot element
 - avoiding division by zero
 - minimizes round-off error
 - gives a partial remedy for ill-conditioning
- Use scaling : minimize round-off error for those cases where some of the equations in a system have much larger coefficients than others.

However, sometime scaling introduces a round-off error. Thus, it is used as a criterion for pivoting and the original coefficient values are retained for the actual elimination and substitution computation.

Complex Systems

Convert a complex system into two real system and employ the algorithm for the real system.

Nonlinear Systems of Equations

Use a multidimensional version of Newton-Raphson method for nonlinear system. However, there are two major shortcoming.

- Difficult to calculate partial derivatives
- Need excellent initial guesses

Need optimization techniques !

Gauss-Jordan

Difference between Gauss and Gauss-Jordan

- An unknown is eliminated from all other equation rather than just the subsequent ones.
- All rows are normalized by dividing them by their pivot elements.
 - the elimination step results in an identity matrix rather than a triangular matrix
 - No need to employ back substitution to obtain the solution.

LU Decomposition and Matrix Inversion

LU decomposition provides

- well-suited for those situations where many right-hand side vector $\{B\}$ must be evaluated for a single value of $[A]$.
- an efficient means to compute the matrix inverse.

LU Decomposition

Gauss elimination is designed to solve system of linear algebraic equations

$$[A]\{X\} = \{B\} \quad (3.1)$$

Gauss elimination involves two steps: forward elimination and back substitution. Of these, the forward elimination step require more computation times. LU decomposition methods separate the time-consuming elimination of the matrix $[A]$ from the manipulations of the right-hand side $\{B\}$. Thus, once $[A]$ has ben decomposed, multiple right-hand side vectors can be evaluated in an efficient manner.

Equation (3.1) can be rearranged to give

$$[A]\{X\} - \{B\} = 0 \quad (3.2)$$

The above equation can be reduced into upper triangular form with Gauss elimination.

$$[U]\{X\} - \{D\} = 0 \quad (3.3)$$

And assume a lower triangular matrix $[L]$ which satisfies the following equation

$$[L]\{[U]\{X\} - \{D\}\} = [A]\{X\} - \{B\} \quad (3.4)$$

If this equation holds,

$$[L][U] = [A] \quad (3.5)$$

$$[L][D] = \{B\} \quad (3.6)$$

A two-step strategy for obtaining solutions with LU decomposition

1. Decompose $[A]$ into $[L]$ and $[U]$
2. Find a intermediate vector $\{D\}$ with equation (3.6) and solve equation (3.3) for $\{X\}$

LU decomposition can be performed in Gauss elimination. $[U]$ is a direct product of the forward elimination. For example, consider the following 3×3 system

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (3.7)$$

The forward elimination step reduce the original matrix $[A]$ to the form

$$[U] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \quad (3.8)$$

which is in the desired upper triangular format. The matrix $[L]$ is also produced during the step.

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ f_{21} & 1 & 0 \\ f_{31} & f_{32} & 1 \end{bmatrix} \quad (3.9)$$

LU decomposition algorithm:

- The factors generated during the elimination phase are stored in the lower part of the matrix
- Keep track of pivoting
- Scaled values of the elements are used to determine whether pivoting is to be implemented
- The diagonal term is monitored during the pivoting phase to detect near-zero occurrences in order to flag singular systems.

The Matrix Inverse

The inverse can be computed in a column-by column fashion by generating solutions tieh unit vector as the right-hand-side constants. The best way to implement such a calculation is with the LU decomposition algorithm and it is one of the great strengths of LU decomposition.

Error Analysis and System Condition

Determination of ill-conditioned system:

- Scale the matrix and invert the scaled matrix. If there are element of $[A]^{-1}$ that are several orders of magnitude greater than one, then the system is ill-conditioned.
- Multiply the inverse by the original matrix and assess whether the result is close to the identity matrix. If not, it indicate ill-conditioning.
- Invert the inversed matrix and assess whether the result is sufficiently close to the original matrix. If not, it indicate that the system is ill-conditioned.

The indication of ill-conditioning with a single number

- Norm: a real-valued function that provide a measure of the size of multicomponent mathematical entities. For example, consider a vector in three-dimensional Euclidean space

$$[F] = [a \quad b \quad c]$$

The length of this vector-that is, the distance from the coordinate (0, 0, 0) to (a, b, c)

$$\|F\|_e = \sqrt{a^2 + b^2 + c^2}$$

where the nomenclature $\|F\|_e$ indicates that this length is referred to as the Euclidean norm of $[F]$. For matrix case,

$$\|A\|_e = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{i,j}^2}$$

which is given a special name-the Frobenius norm. it provide a single value to quantify the "size" of $[A]$.

- Matrix condition number:

$$\text{Cond}[A] = \|A\| \cdot \|A^{-1}\|$$

Note that for a matrix $[A]$, this number will be greater than or equal to 1. However, the above equation require computation time to obtain $\|A^{-1}\|$.

Special Matrices and Gauss-Seidel

Special Matrices

- Banded matrix : a square matrix that has all elements equal to zero, with exception of a band centered on the main diagonal. The conventional LU decomposition methods are inefficient to solve banded systems.
- Tridiagonal system : Thomas algorithm
- Symmetric matrix : Cholesky decomposition

Gauss-Seidel

The Gauss-Seidel method:

- An alternative to the elimination method
- Iterative or approximate method

Convergence enhancement with relaxation

- Relaxation is a weighted average of the result of the previous and the present iteration:

$$x_i^{\text{new}} = \lambda x_i^{\text{new}} + (1 - \lambda)x_i^{\text{old}}$$

- $\lambda = 1$: the result is unmodified.
- $0 < \lambda < 1$: underrelaxation, make a nonconvergent system converge or hasten convergence by dampening out oscillation.
- $\lambda > 1$: overrelaxation, accelerate the convergence of an already convergent system. It is also called successive or simultaneous overrelaxation, SOR.

Linear Algebraic Equation with Libraries and Packages

- Matlab:
 - cond : matrix condition number
 - norm : matrix or vector norm
 - rank : no. of linearly independent rows or columns
 - det : determinant
 - trace : sum of diagonal elements
 - / : linear equation solution
 - chol : cholesky factorization

- lu : factors from gauss elimination
- inv : matrix inverse
- IMSL: various routines are exist to solve linear system

Engineering Applications: Linear Algebraic Equations

See the textbook

[Next](#) [Up](#) [Previous](#) [Contents](#)

Next: [Optimization](#) **Up:** [Numerical Analysis for Chemical](#) **Previous:** [Roots of Equations](#)

Taechul Lee
2001-11-29