Linear Systems of Equations

Solution of Linear Systems. Solving linear systems may very well be the foremost assignment of numerical analysis. Much of applied numerical mathematics reduces to a set of equations, or linear system:

$$Ax = b \qquad (1)$$

with the matrix A and vector b given, and the vector x to be determined. An extraordinary collection of algorithms has been developed for achieving this has been developed. These can be classified into two categories: direct methods and iterative methods. The variety of the algorithms indicated that the apparently elementary character of the problem is deceptive. We will first consider direct methods: The most common method is Gaussian elimination (and back-substitution) or LU decomposition (forward and back-substitution).

Gaussian Elimination.

We can write the linear system (1) as

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i \qquad i=1,\ldots,n \qquad (2)$$

A is a square n x n matrix with coefficients:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad (3)$$

In order for this system to have a unique solution, it must be nonsingular, i.e. its inverse must exist. Then the solution is given by

$$x = A^{-1} \cdot b \qquad (4)$$

Note $I = A^{-1} \cdot A$, the identity matrix of order n, which has 1's on the main diagonal and zero's everywhere else. In general, computing the inverse of a matrix is not easy.

In Gaussian elimination we transform the system (1) into an **upper triangular system**, by carrying out elementary operations on the equations. This gives

$$U\,x = c. \qquad (5)$$

This system can be easily solved by backward substitution. The only non-zero elements in U occur on and above the main diagonal. Hence we first find, x_n, then $x_{n-1}, \ldots, x_1$.

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{n,n} \end{bmatrix} \qquad (6)$$

We can rewrite the system (1) in long format as:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n &= b_2 \\ &\vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n &= b_n \end{aligned}$$

We now eliminate x_1 from the 2nd to the nth equation, by subtracting $a_{i,1} / a_{1,1}$ times the 1st equation from the ith equation. This ratio is called the multiplier. Equations 2 to n now only involve x_2 to x_n. The same procedure is carried out to eliminate x_2 from equations 3 to n. The same operations are carried out on the right-hand side to modify the b_i terms. Finally, the last equation will only involve x_n. This system is straightforward to solve.

Example. Consider the simple example of 3 equations

$$3x_1 - x_2 + 2x_3 = 12$$
$$x_1 + 2x_2 + 3x_3 = 11$$
$$2x_1 - 2x_2 - x_3 = 2$$

Gaussian Elimination Algorithm. To solve a system of linear equations:

1. Augment the n x n coefficient matrix with the vector of right-hand sides to form an *n x (n+1)* matrix. The RHS is the (n+1)st column.
2. Interchange rows (if necessary, or required) to make $a_{1,1}$ the largest magnitude of any coefficient in the first column. (Optional)
3. Create zeros in the 2nd through to the nth rows in the first column by subtracting $a_{j,1}/a_{1,1}$ times the first row from the j-th row, j = 2 to n.
4. Repeat steps (2) and (3) for the i = 2nd through to (n-1)st rows, putting the largest-magnitude coefficient on the diagonal by interchanging rows (considering only rows i to n), and then subtracting $a_{j,i}/a_{i,i}$ times the i-th row from the j-th row, (for j = i+1 to n) so as to create zeros in all positions of the i-th column below the diagonal. At the end of this step, the system is *upper-triangular*.
5. Solve for x_n from the nth equation by $x_n = a_{n,n+1} / a_{n,n}$. Note the RHS $b_n = a_{n,n+1}$
6. Solve for $x_{n-1}, x_{n-2}, \ldots, x_1$ from the (n-1)st through the first equation in turn, by

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{i,j} x_j}{a_{i.i}} \qquad \text{i.e. backward-substitution} \qquad (7)$$

In terms of pseudocode the algorithm is basically as follows: for each pivot element $a_{i,i}$, eliminate the remaining column j, by subtracting from row_j the multiplier*row_i:

```
for i = 1 to n-1
    pivot = a(i,i)
    for j = i + 1 to n        // for row j, compute multiplier
        m(j,i) = a(j,i)/a(i,i)
        for k = i+1 to n      // elements of row j
            a(j,k) = a(j,k) - m(j,i)*a(i,k)
        end
    end
end
```

LU Decomposition.

Another variation of Gaussian elimination is the LU decomposition of the matrix A. Here L is a lower triangular matrix, and U is an upper triangular matrix. Then the system of equations can be written as

$$LUx = b, \quad \text{or} \quad Ly = b, \quad \text{where} \quad Ux = y, \qquad (8)$$

So first solve the system $Ly = b$, for y by forward substitution, and then solve the system

$Ux = y$, for x by backward substitution. The elements of the lower triangular matrix L are in fact the multipliers $a_{j,i}/a_{i,i}$ obtained in the Gaussian elimination routine, and U is the upper-triangular matrix obtained by the Gaussian elimination algorithm. Thus

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{2,1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m_{n.1} & m_{n,2} & \cdots & 1 \end{bmatrix} \tag{9}$$

Exercise. Verify the LU decomposition for the 3 equations example given above.

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix} \tag{10}$$