System of Linear Equations

A general set of n equations with n unknowns has the form

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\
&\vdots \\
&a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n
\end{aligned}
$$

a_{ij} are the coefficients (known) of the equations, b_i are the right hand sides (known), and x_j are the unknowns. The set of equations can be written more compactly as

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad \text{for } i=1,2\ldots n$$

The a_{ij}'s represent a $n \times n$ (square) matrix A, and b_i and x_j are vectors of length n. The system of linear equations can be written even more compactly in matrix notation as

$$Ax = b$$

The solution of this matrix equation is equally simple to write, but can be difficult to compute,

$$x = A^{-1}b$$

where A^{-1} is the inverse of the matrix A. Determining the inverse of a matrix is notoriously difficult and it is not the preferred method for obtaining the solution to a set of equations. Thus, the notation in the equation above is more schematic than utilitarian.

MATLAB Methods

- x=inv(A)*b Avoid this approach if possible
- x=A\b % Recommended approach
- The second approach is more efficient and more accurate for larger problems.
- The backslash operator "\" solves the system of equations using Gaussian elimination method.
- It also solves overdetermined/underdetermined systems of linear equations.
- Overdetermined: more equations than unknowns (least-square solution)
- Underdetermined: fewer equations than unknowns (minimum norm solution)

Example

Solve this linear system using MATLAB

$$
\begin{aligned}
x_1 + 2x_2 + x_3 &= 2 \\
3x_1 + x_2 + 2x_3 &= 1 \\
-2x_2 + 4x_3 &= 1
\end{aligned}
$$

Examples

- A=[1 2 3;4 5 6;7 8 0;2 5 8] 4 equations and 3 unknowns
- b=[366 804 351 514]’
- x=A\b
- A=A’ 3 equations and 4 unknowns
- b=b(1:3)
- x=A\b

Gaussian Elimination

- Gaussian Elimination
  - Among the oldest and most widely used solutions.
  - Repeatedly apply row operations to make system upper triangular.
  - Solve upper triangular system by back substitution.
- Elementary row operations.
  - Exchange row p and row q.
  - Add a multiple α of row p to row q.

Elementary Row Operation

$$\begin{array}{rcrcrcr} 0x_0 & + & 1x_1 & + & 1x_2 & = & 4 \\ 2x_0 & + & 4x_1 & - & 2x_2 & = & 2 \\ 0x_0 & + & 3x_1 & + & 15x_2 & = & 36 \end{array}$$

$$\Downarrow$$

$$\begin{array}{rcrcrcr} 2x_0 & + & 4x_1 & - & 2x_2 & = & 2 \\ 0x_0 & + & 1x_1 & + & 1x_2 & = & 4 \\ 0x_0 & + & 3x_1 & + & 15x_2 & = & 36 \end{array}$$

$$\Downarrow$$

$$\begin{array}{rcrcrcr} 2x_0 & + & 4x_1 & - & 2x_2 & = & 2 \\ 0x_0 & + & 1x_1 & + & 1x_2 & = & 4 \\ 0x_0 & + & 0x_1 & + & 12x_2 & = & 24 \end{array}$$

Upper Triangle System

$$
\begin{array}{rcrcrcr}
2\,x_0 & + & 4\,x_1 & - & 2\,x_2 & = & 2 \\
0\,x_0 & + & 1\,x_1 & + & 1\,x_2 & = & 4 \\
0\,x_0 & + & 0\,x_1 & + & 12\,x_2 & = & 24
\end{array}
$$

Back Substitution

Equation 2: $x_2 = 24/12 = 2.$

Equation 1: $x_1 = 4 - x_2 = 2.$

Equation 0: $x_0 = (2 - 4x_1 + 2x_2) / 2 = -1.$

Gaussian Elimination

Step 1: write the coefficients as an $n \times n+1$ array and reduce the elements of the first column to a 1 in the first row and 0's in the remaining rows:

(i) divide row 1 by a_{11}

(ii) multiply row 1 by a_{21} and subtract from row 2

(iii) perform similar operation for row 3, etc;

It is important that the calculations be carried out with as much precision as possible.

Step 2: make the coefficient of the second column, second row 1 and the elements below the second row in the second column 0 using operations similar to those in Step 1.

Step 3: follow the same steps for each successive column to place 1's on the main diagonal of the coefficient matrix and 0's below. The operations for the last column will leave a simple equation of the form $x_n = \hat{b}_n$, where $\hat{b}$ represents the terms on the right-hand sides of the modified equations.

$$\begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a'_{33} & a'_{34} \\ 0 & 0 & 0 & a'_{44} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix} \tag{2.2.1}$$

Here the primes signify that the a's and b's do not have their original numerical values, but have been modified by all the row operations in the elimination to this point. The procedure up to this point is termed *Gaussian elimination*.

Backsubstitution

But how do we solve for the x's? The last x (x_4 in this example) is already isolated, namely

$$x_4 = b'_4 / a'_{44} \tag{2.2.2}$$

With the last x known we can move to the penultimate x,

$$x_3 = \frac{1}{a'_{33}} [b'_3 - x_4 a'_{34}] \tag{2.2.3}$$

and then proceed with the x before that one. The typical step is

$$x_i = \frac{1}{a'_{ii}} \left[b'_i - \sum_{j=i+1}^{N} a'_{ij} x_j \right] \tag{2.2.4}$$

The procedure defined by equation (2.2.4) is called *backsubstitution*. The combination of Gaussian elimination and backsubstitution yields a solution to the set of equations.

Example

Solve this linear system by hand using Gaussian elimination

$$
\begin{aligned}
x_1 + 2x_2 + x_3 &= 2 \\
3x_1 + x_2 + 2x_3 &= 1 \\
-2x_2 + 4x_3 &= 1
\end{aligned}
$$

Pivoting

- Pivot element

  The element in the diagonal of a matrix by which other elements are divided is called the pivot element.
- In the presence of any round-off error, Gauss-Jordan elimination with no pivoting is numerically unstable.
- Partial pivoting is the interchanging of rows and full pivoting is the interchanging of both rows and columns in order to place a particularly "good" element in the diagonal position prior to a particular operation.
- Pivoting information is needed for calculating determinant.

Stability

- Algorithm fl(x) for computing f(x) is numerically stable if fl(x) ≈ f(x+ε) for some small perturbation ε.

$a = 10^{-17}$

$$\begin{aligned} a\,x_0 &+ 1\,x_1 = 1 \\ 1\,x_0 &+ 2\,x_1 = 3 \end{aligned}$$

Algorithm	x_0	x_1
no pivoting	0.0	1.0
partial pivoting	1.0	1.0
exact	$\frac{1}{1-2a} \approx 1$	$\frac{1-3a}{1-2a} \approx 1$

Theorem. Partial pivoting improves numerical stability.