Direct Methods for Solving Linear Systems and Least Squares Problems.

Basics.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & & \vdots \\ \vdots & & \vdots \\ a_{m1} & & a_{mn} \end{pmatrix}$$ A is an m by n matrix

A^T: Transpose of A in which the $(i,j)^{th}$ entry is a_{ji}

If A is complex, we often use the Hermitian transpose (A^*) in which the $(i,j)^{th}$ entry is the complex conjugate of a_{ji}

A is Symmetric if $A^T = A$

A is Hermitian if $A^* = A$

(In both cases, the matrix must be square)

Vector $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$.

Matrix vector Product.

$$Ab = \begin{pmatrix} a_{11}b_1 + a_{12}b_2 + \cdots + a_{1n}b_n \\ \vdots \\ a_{m1}b_1 + \cdots\cdots + a_{mn}b_n \end{pmatrix}.$$

The i^{th} entry in Ab is the dot product of i^{th} row of A with b

$$Ab = \begin{pmatrix} \langle a_{1,:}, b \rangle \\ \vdots \\ \langle a_{m,:}, b \rangle \end{pmatrix}$$

The product Ab can also be thought of as a linear combination of columns of A, where each column of A is scaled by corresponding entry in b.

Gaussian Elimination.

A: $n \times n$ nonsingular matrix

b: n dimensional vector.

$Ax = b$ has a unique solution.

$$x_1 + 2x_2 + 3x_3 = 1$$
$$4x_1 + 5x_2 + 6x_3 = 0$$
$$7x_1 + 8x_2 = 2$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}.$$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 0 \\ 7 & 8 & 0 & 2 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 0 & -3 & -6 & -4 \\ 0 & -6 & -21 & -5 \end{array}\right)$$

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 0 & -3 & -6 & -4 \\ 0 & 0 & -9 & 3 \end{array}\right) \qquad \begin{aligned} x_1 + 2x_2 + 3x_3 &= 1 \\ -3x_2 - 6x_3 &= -4 \\ -9x_3 &= 3. \end{aligned}$$

$x_3 = -1/3$, $x_2 = 2$, $x_1 = -2$.

Gaussian Elimination as Matrix factorization.

The first set of eliminations can be written as

$$\begin{pmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -7 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -21 \end{pmatrix}$$

$L_1 \times A$.

L_1 is easy to invert -- Simply negate off diagonal entries.

$$\begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 0 & 1 \end{pmatrix}.$$

The second elimination step can be written as

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -21 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & -9 \end{pmatrix}$$

$L_2 \times L_1 \times A = U$

U

Inverse of L_2 is easy to compute as well.

$$(L_2 L_1)^{-1} = L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{pmatrix}$$

$L_2 L_1 A = U$

$$A = (L_2 L_1)^{-1} U$$
$$= L_1^{-1} L_2^{-1} U = LU$$

To solve the same system with multiple RHSs.

$$Ax = LUx = b$$

$$Ly = b$$

$$Ux = y.$$

Computational complexity.

$$\sum_{i=1}^{n} (n-i)^2 = \sum_{i=0}^{n-1} i^2 \approx O(n^3)$$

Pivoting.

What happens if at an intermediate step, the diagonal entry of the pivot row is zero?

Interchange it with one of the other rows.

This is called pivoting.

We should be able to find atleast one such row (with a non-zero at corresponding column) otherwise the matrix must be singular!

Finite precision causes problems.

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

Exact solution is nearly $x_1 = x_2 = 1$

$$\left(\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \longrightarrow \left(\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & -10^{+20} & -10^{20} \end{array}\right)$$

In finite precision, we get $x_2 = 1$ and $x_1 = 0$!

Fix (Swap rows).

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 10^{-20} & 1 & 1 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right)$$

$x_2 = 1 \quad x_1 = 1$

Use Partial Pivoting.

Search for the largest entry in the column in absolute value and use that as the pivot element.

This ensures that multipliers for all other rows are less than 1.