Linear Algebra

Numerical linear algebra

- System of linear equations
  - ODE integration
  - Cubic spline interpolation
  - Diffusion PDE
  - Multivariate root finding
  - Curve fitting

System of linear eqns

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2N}x_N &= b_2 \\
\vdots \quad &\quad \vdots \\
a_{M1}x_1 + a_{M2}x_2 + a_{M3}x_3 + \cdots + a_{MN}x_N &= b_M.
\end{aligned}
$$

$\mathbf{Ax} = \mathbf{b}$

swap rows? ✅
swap columns? ➞ swap variables

- If M > N, more eqns than unknowns. *Overdetermined*. No consistent solutions (unless some eqns repeat or are linear combinations of others). The only option is to find an approximate solution using method of least squares.

- If M < N, there is no unique solution. *Underdetermined*.

- We restrict to M=N case. Square matrix. $\mathbf{x} = \mathbf{A}^{-1}\, \mathbf{b}$ if $\mathbf{A}$ is not singular (zero $\boldsymbol{\Delta}$)

- However, there are methods to solve the equations without finding the inverse. **Direct** and iterative methods. We'll see Gaussian elimination and LU decomposition.

Gaussian elimination

- The main & general technique to solve a linear system $\mathbf{Ax} = \mathbf{b}$
    - No need to compute inverse
    - For special matrices (eg., tridiagonal) faster techniques are available
    - Order of calculations: N^2 (column vector) ; N^3 (general matrix)
- Forward elimination → Row Echelon matrix→ Back substitution
- Round off errors can cause damage. *(No truncation here)*
    - Pivoting helps in such cases. Pivoting algorithms rearrange rows when they spot small diagonal elements.

Pivoting

- If any of the a_{ii} are zero or very small compared to non-diagonal elements, division by zero can occur.

- Pivoting is to reorder the equations, and keep row containing the largest element of column-1, as the 1st row (partial pivoting).

$$\left(\begin{array}{ccc|c} 0 & 3 & -4 & 10 \\ 1 & 5 & -1 & 12 \\ 3 & 7 & -3 & 20 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 3 & 7 & -3 & 20 \\ 1 & 5 & -1 & 12 \\ 0 & 3 & -4 & 10 \end{array}\right)$$

Caveats

- Singular matrix can not be solved this way
- One equation is a linear combination of the other
- However, singular matrices are not always easy to spot. Eg., [1 2 3] [4 5 6] [7 8 9]
- Sometimes roundoff can push a matrix to being singular. Eg., [1+ϵ 1] [2 2]
- Condition Number measures how close to singular a matrix is
  $norm(\mathbf{A}) * norm(\mathbf{A}^{-1})$
- If you suspect you are dealing with an ill conditioned matrix, calculate the absolute error after you get the solution **x**, $|\mathbf{Ax} - \mathbf{b}| / |\mathbf{b}|$, to check if **x** is an accurate solution.

Inverse of a matrix

- Can be represented as a gaussian elimination problem.
  - Setting $\mathbf{b}=\mathbf{I}$ will give $\mathbf{A}^{-1}$ (equation now is $\mathbf{A}\mathbf{x} = \mathbf{I}$)
- LU decomposition is also used

Determinants

- Easy to calculate determinant after gaussian elimination

- $\boldsymbol{\Delta} = (-1)^{np} \prod^{n}_{i=1} \mathbf{A}_{ii}$; p = number of row swaps

Tridiagonal

- Very usual in physical systems
- N calculations

Write a general code, for N=Nmax (say 100) to solve a tridiagonal matrix.

Apply your code to the following matrix (we'll see this matrix again in PDE).

$$\begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} 0.04 \\ 0.04 \\ 0.04 \\ 0.04 \end{pmatrix}$$

LU decomposition

- **A = LU**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} =$$

$$\begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & 1 & u_{23} & \dots & u_{2n} \\ 0 & 0 & 1 & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} .$$

How to find L and U?

Alternate between rows and columns of A to get:-

$$l_{ik} = a_{ik} - \sum_{j=1}^{k-1} l_{ij} u_{jk}, \quad i = k, k+1, \ldots, n,$$

$$u_{kj} = \frac{a_{kj} - \sum_{i=1}^{k-1} l_{ki} u_{ij}}{l_{kk}}, \quad j = k+1, k+2, \ldots, n.$$

An LU decomposition assignment (submit on or before 9 Oct, Friday)

Using the algorithm above, and obtain the L and U matrices for

$$A = \begin{pmatrix} 2 & 3 & -4 \\ 1 & 5 & -1 \\ 3 & 7 & -3 \end{pmatrix}$$

LU decomposition

$LUx = b$

$L\,z = b$

where $Ux = z$

- $\mathbf{A = LU}$
- $\mathbf{Ax = b} \Rightarrow \mathbf{Lz = b}$ & $\mathbf{Ux = z}$
- Have only triangular matrices. So only need to do back substitution.
- If your problem has different **b**s, this is the best method.

Use forward substitution to find z

$$z_i = \frac{b_i - l_{i1}z_1 - l_{i2}z_2 \cdots - l_{i,i-1}z_{i-1}}{l_{ii}}$$

$$= \frac{b_i - \sum_{k=1}^{i-1} l_{ik}z_k}{l_{ii}}, \qquad i = 2, \ldots, N,$$

Then do backward substitution to find y

$$x_{n-i} = z_{n-i} - \sum_{k=1}^{i-1} u_{n-i,k}\, x_k, \qquad i = 1, n-1.$$

Those interested to develop the code for forward and back substitution in LU decomposition, using the above algorithm, are encouraged to do that.

*Use these in spline algorithm