Simultaneous Equations

Linear Algebra - Lecture 1
Computational Physics 301

Linear Algebra

- **A vast, and incredibly important area of numerical methods**
- **What is it used for ?**
  - Physics, engineering, economics, modelling, simulation, signal analysis, error correction, prediction, machine learning, quantum computing, games, graphics, …………
- **What isn't it used for ?**
  - Non-linear problems?
    - But we often find an approximation and use linear techniques anyway !

"Linear algebra is what most computers are doing, most of the time…"

Simultaneous Equations

- A big, big class of problems…

$$\begin{aligned} ax_1 + bx_2 &= y_1 \\ cx_1 + dx_2 &= y_2 \end{aligned} \qquad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \qquad A\underline{x} = \underline{y}$$

- Numerical methods are unsurprisingly useful when the number of equations (and variables) N becomes large

- Many methods for solving :
  - Matrix inversion
  - Gaussian elimination
  - Matrix decomposition
  - …

Matrix Inversion

- A solution becomes clear if we use the matrix form :

$$
\begin{aligned}
ax_1 + bx_2 &= y_1 \\
cx_1 + dx_2 &= y_2
\end{aligned}
$$

$$
\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}
$$

$$
A\underline{x} = \underline{y}
$$

$$
\begin{aligned}
A^{-1}A\underline{x} &= A^{-1}\underline{y} \\
\underline{x} &= A^{-1}\underline{y}
\end{aligned}
$$

- If we can invert the matrix, we can find the solution

Matrix Inversion

- One approach would be to use Cramer's rule : $$A^{-1} = \frac{1}{\det A} C^T$$
- Where C is the matrix of co-factors
- $C_{i,j}$ given in terms of the minor $M_{i,j}$

$$M = \begin{pmatrix} & & \\ 4 & 0 & \\ -1 & 9 & \end{pmatrix} \longrightarrow M_{1,3} = \det \begin{pmatrix} 4 & 0 \\ -1 & 9 \end{pmatrix} = (4 \times 9) - (-1 \times 0) = 36$$

$$C_{1,3} = (-1)^{1+3} M_{1,3} = -1^4 \times 36 = 36$$

- In theory this should work for any matrix - right ?

Matrix Inversion

- **What happens when N becomes large ?**
    - How many operations are required, in terms of N ? (ie. what is the algorithmic complexity of this method ?)
- **What about singular matrices ?**
    - When $\det A = 0$ we have a divide by zero ☹️
- **What about near-singular matrices ?**
    - When $\det A$ is very large, or very small, rounding errors become important

Gaussian Elimination

- Some sets of simultaneous equations are easy to solve

$$\begin{aligned} ax_1 &= y_1 \\ bx_2 &= y_2 \\ cx_3 &= y_3 \end{aligned} \;\rightarrow\; \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \;\rightarrow\; \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1/a \\ y_2/b \\ y_3/c \end{pmatrix}$$

Row echelon form — *Reduced row echelon form*

- Gauss-Jordan Elimination is a process that reduces any set of linear equations to this form
  - It can be shown that the reduced row echelon form is unique
  - And therefore independent of the order of operations required to obtain it

Gaussian Elimination

- Usually use the augmented matrix, which includes coefficients of the result :

$$\begin{pmatrix} 1 & 2 & 2 \\ 3 & 1 & 6 \\ 0 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 7 \\ 1 \end{pmatrix} \longrightarrow \left(\begin{array}{ccc|c} 1 & 2 & 2 & 2 \\ 3 & 1 & 6 & 7 \\ 0 & 2 & 2 & 1 \end{array}\right)$$

Augmented matrix

- Then apply simple operations to reduce this to row echelon form
  1. Multiply a row by a constant
  2. Swap two rows
  3. Add two rows in linear combination

Sound familiar? This is just a way to formalist techniques you already use

Gaussian Elimination - Example

Initial matrix :
$$\left(\begin{array}{ccc|c} 1 & 2 & 2 & 2 \\ 3 & 1 & 6 & 7 \\ 0 & 2 & 2 & 1 \end{array}\right)$$

$$R_1 \to R_1 - R_3 \qquad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 3 & 1 & 6 & 7 \\ 0 & 2 & 2 & 1 \end{array}\right)$$

$$R_2 \to R_2 - 3R_1 \qquad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 6 & 4 \\ 0 & 2 & 2 & 1 \end{array}\right)$$

$$R_2 \to R_2 - 3R_3 \qquad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & -5 & 0 & 1 \\ 0 & 2 & 2 & 1 \end{array}\right)$$

$$R_3 \to R_3 + \frac{2}{5}R_2 \qquad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & -5 & 0 & 1 \\ 0 & 0 & 2 & \frac{7}{5} \end{array}\right)$$

For reduced row echelon form :

$$R_2 \to \frac{-1}{5}R_2 \qquad R_3 \to \frac{1}{2}R_3 \qquad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -\frac{1}{5} \\ 0 & 0 & 1 & \frac{7}{10} \end{array}\right)$$

Solution is :

$$x_1 = 1$$

$$x_2 = -\frac{1}{5}$$

$$x_3 = \frac{7}{10}$$

Check it !

Gaussian Elimination

- Another clear example from wikipedia

System of equations	Row operations	Augmented matrix
$\begin{alignedat}{4} 2x &+ y &- z &= 8 \\ -3x &- y &+ 2z &= -11 \\ -2x &+ y &+ 2z &= -3 \end{alignedat}$		$\left[\begin{array}{ccc\|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right]$
$\begin{alignedat}{4} 2x &+ y &- z &= 8 \\ &\tfrac{1}{2}y &+ \tfrac{1}{2}z &= 1 \\ &2y &+ z &= 5 \end{alignedat}$	$L_2 + \frac{3}{2}L_1 \to L_2$ $L_3 + L_1 \to L_3$	$\left[\begin{array}{ccc\|c} 2 & 1 & -1 & 8 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 2 & 1 & 5 \end{array}\right]$
$\begin{alignedat}{4} 2x &+ y &- z &= 8 \\ &\tfrac{1}{2}y &+ \tfrac{1}{2}z &= 1 \\ & &-z &= 1 \end{alignedat}$	$L_3 + -4L_2 \to L_3$	$\left[\begin{array}{ccc\|c} 2 & 1 & -1 & 8 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 1 \end{array}\right]$
The matrix is now in echelon form (also called triangular form)		
$\begin{alignedat}{4} 2x &+ y & &= 7 \\ &\tfrac{1}{2}y & &= \tfrac{3}{2} \\ & &-z &= 1 \end{alignedat}$	$L_2 + \frac{1}{2}L_3 \to L_2$ $L_1 - L_3 \to L_1$	$\left[\begin{array}{ccc\|c} 2 & 1 & 0 & 7 \\ 0 & \frac{1}{2} & 0 & \frac{3}{2} \\ 0 & 0 & -1 & 1 \end{array}\right]$
$\begin{alignedat}{4} 2x &+ y & &= 7 \\ &y & &= 3 \\ & &z &= -1 \end{alignedat}$	$2L_2 \to L_2$ $-L_3 \to L_3$	$\left[\begin{array}{ccc\|c} 2 & 1 & 0 & 7 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$
$\begin{alignedat}{4} x & & &= 2 \\ &y & &= 3 \\ & &z &= -1 \end{alignedat}$	$L_1 - L_2 \to L_1$ $\frac{1}{2}L_1 \to L_1$	$\left[\begin{array}{ccc\|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$

https://en.wikipedia.org/wiki/Gaussian_elimination#Example_of_the_algorithm

LU Decomposition

- LU decomposition is closely related to Gaussian Elimination
  - Preferred when dealing with multiple sets of equations based on the same matrix
  - We decompose the matrix once, then apply it to multiple equations

- It involves writing a general matrix as the product of two triangular matrices

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \longrightarrow A = LU = \begin{pmatrix} 1 & 0 & 0 \\ l_{12} & 1 & 0 \\ l_{13} & l_{23} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{21} & u_{31} \\ 0 & u_{22} & u_{32} \\ 0 & 0 & u_{33} \end{pmatrix}$$

- Various methods to find the matrices L and U, including modified form of Gaussian elimination described on the previous slides

LU Decomposition

- We can apply LU decomposition to solve simultaneous equation problems
- By substituting LU for A, we end up with two simultaneous equations

$$A\underline{x} = \underline{y} \quad \Rightarrow \quad L(U\underline{x}) = L\underline{c} = \underline{y} \quad \Rightarrow \quad L\underline{c} = \underline{y} \qquad U\underline{x} = \underline{c}$$

  - We can first solve for $\underline{c}$ then for $\underline{x}$

- Solving is straightforward, since both L and U are triangular

$$\begin{pmatrix} 1 & 0 & 0 \\ l_{12} & 1 & 0 \\ l_{13} & l_{23} & 1 \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \qquad \begin{pmatrix} u_{11} & u_{21} & u_{31} \\ 0 & u_{22} & u_{32} \\ 0 & 0 & u_{33} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Singular Value Decomposition

- Sometimes we have more unknowns than equations
  - eg. MxN matrix with M<N, or M=N but equations are degenerate
  - There may be no solution, or a space of solutions
- Or we have a square matrix with determinant zero
- Such matrices are called singular, and are not invertible
- SVD is useful in these circumstances
  - Also for near-singular, eg. when $\frac{1}{\det A}$ is close to floating point precision, and rounding error becomes a problem

Singular Values

- For an $m \times n$ matrix A, the "singular values" σ satisfy $A\underline{\hat{v}} = \sigma\underline{\hat{u}}$, $A^T\underline{\hat{u}} = \sigma\underline{\hat{v}}$
- This pair of equations are closely related to the eigenvalue equation
  - In fact, the singular values are the square roots of the eigenvalues of A^TA
- An SVD decomposition is $A = U\Sigma V^T$
  - U, V are orthonormal matrices, ie. composed from a set of orthogonal unit vectors
  - Σ is a matrix with the singular values of A on its diagonal

Singular Value Decomposition

- SVD allows us to calculate the pseudo-inverse matrix : $A^\dagger = V\Sigma^\dagger U^T$
  - Where $\Sigma^\dagger$ is obtained by replacing each non-zero element on the diagonal of Σ with its reciprocal, and transposing the result
- Applying the pseudo-inverse matrix to matrix problem $A\underline{x} = \underline{y}$ gives $\bar{x} = V\Sigma^\dagger U^T \underline{y}$
- $\bar{x}$ is interesting for a variety of reasons
  - If A is non-singular, then $\bar{x} = \underline{x}$, the solution to $A\underline{x} = \underline{y}$
  - $\bar{x}$ is also the solution to a *least squares problem* - if there is no solution to $A\underline{x} = \underline{y}$, then $\bar{x}$ is the closest approximation to one

Summary

- Simultaneous equations are a large class of problems
  - *Numerical methods are extremely useful, especially for large N*
- We looked at several methods for finding solutions
  - Matrix inversion
  - Gaussian elimination
  - LU decomposition
  - SVD decomposition

- **Next** - study the notebook on Matrix Inversion to see an example implementation, and application to a very simple physics problem