UNSW Mathematics Society presents

MATH2089 Revision Seminar

Numerical Methods & Statistics

Numerical Methods

T2, 2020

Presented by Janzen Choi
Table of Contents I

1 Part I: Linear Systems
   - LU factorisation
   - Cholesky factorisation
   - Sparsity of matrices
   - Norm and condition number
   - Sensitivity of a linear system

2 Part II: Least Squares & Polynomial Interpolation
   - Least squares
     - Normal equations
     - QR factorisation
   - Polynomial interpolation
     - Lagrange polynomials
     - Chebyshev points

3 Part III: Nonlinear Equations
   - Bisection
   - Fixed point iteration
# Table of Contents II

- Newton’s method
- Secant method

## Part IV: Numerical Differentiation and Integration

- Taylor approximation
- Finite difference methods
  - Forward difference approximation
  - Central difference approximation
- Quadrature rules
  - Trapezoidal rule
  - Simpson’s rule
  - Gauss-Legendre rule
- Change of variables

## Part V: Ordinary Differential Equations

- First order IVP
  - Existence and uniqueness
  - Euler’s method
Table of Contents III

- System of first order ODEs
  - 2 and 4-stage Runge-Kutta methods

Part VI: Partial Differential Equations
- Finite difference methods of PDEs
Part I: Linear Systems
Matrix factorisation – LU factorisation

Given an \( n \times n \) matrix \( A \), if the leading principal sub-matrices \( A_k \) are non-singular for all \( k = 1, \ldots, n \), then there exist \( n \times n \) matrices \( L \) and \( U \) where \( L \) is unit lower triangular and \( U \) is upper triangular, such that

\[
A = LU.
\]

- If the factorisation exists, it is also unique.
- Obtain the \( LU \) factorisation by applying row operations of the form

\[
R_i \leftarrow R_i - L_{ij} R_j.
\]

Cost of factorisation

The cost of the \( LU \) factorisation is

\[
\frac{2n^3}{3} + O(n^2) \text{ flops.}
\]
Effects of the permutation matrix

If $A$ is non-singular, then there exist $n \times n$ matrices $L, U$ and $P$ with $P$ being a permutation matrix such that

$$PA = LU.$$  

- $PA$ reorders the rows of $A$ but does not change the solution to the linear system.
- $AP$ reorders the columns of $A$ and affects the solution to linear system.
Solving linear systems using $LU$ factorisation

We note that

$$Ax = b \implies (PA)x = Pb \implies LUx = Pb.$$ 

1. **Forward substitution:** Solve $Ly = Pb = z$ for $y$. Then

$$y_1 = z_1, \quad y_i = z_i - \sum_{j=1}^{i-1} L_{ij}y_j, \quad i = 2, \ldots, n.$$ 

2. **Back substitution:** Solve $Ux = y$ for $x$. Then

$$x_n = \frac{y_n}{U_{nn}}, \quad x_i = \frac{1}{U_{ii}} \left( y_i - \sum_{j=i+1}^{n} U_{ij}x_j \right), \quad i = n-1, \ldots, 1.$$
Solving linear systems using $LU$ factorisation

Cost of solution

- **LU** factorisation: $\frac{2n^3}{3} + \mathcal{O}(n^2)$ flops.
- Forward substitution: $n^2 + \mathcal{O}(n)$ flops.
- Back substitution: $n^2 + \mathcal{O}(n)$ flops.
- **Total cost:** $\frac{2n^3}{3} + \mathcal{O}(n^2)$ flops.
Given an $n \times n$ matrix $A$, if the Cholesky factorisation exists, then it is of the form

$$A = R^T R$$

where $R$ is an $n \times n$ upper triangular matrix, with $R_{ii} > 0$ for all $i = 1, \ldots, n$.

- A Cholesky factorisation is unique when it exists.
- The matrix $A$ is positive definite and symmetric.
- All eigenvalues of $A$ are positive.

**Cost of factorisation**

The cost of the Cholesky factorisation is

$$\frac{n^3}{3} + O(n^2) \text{ flops}.$$
Solving linear systems using Cholesky factorisation

We note that

\[ Ax = b \implies (R^T R) x = b. \]

1. **Forward substitution**: Solve \( R^T y = b \) for \( y \).

\[
y_1 = \frac{b_1}{R_{11}}, \quad y_i = \frac{1}{R_{ii}} \left( b_i - \sum_{j=1}^{i-1} R_{ji} y_j \right), \quad i = 2, \ldots, n.
\]

2. **Back substitution**: Solve \( R x = y \) for \( x \).

\[
x_n = \frac{y_n}{R_{nn}}, \quad x_i = \frac{1}{R_{ii}} \left( y_i - \sum_{j=i+1}^{n} R_{ij} x_j \right), \quad i = n - 1, \ldots, 1.
\]
Solving linear systems using Cholesky factorisation

**Cost of solution**

- **Cholesky factorisation:** \( \frac{n^3}{3} + \mathcal{O}(n^2) \) flops.
- **Forward substitution:** \( n^2 + \mathcal{O}(n) \) flops.
- **Back substitution:** \( n^2 + \mathcal{O}(n) \) flops.
- **Total cost:** \( \frac{n^3}{3} + \mathcal{O}(n^2) \) flops.
The **sparsity** of a matrix $A$ is given by

$$\text{Sparsity} = \left( \frac{\text{non-zero elements of } A}{\text{total number of elements in } A} \right) \%.$$
Example: (MATH2089, 2010 Q3h)

You are given that using a row-ordering of the variables \( c_{i,j}^{\ell} \) produces the coefficient matrix \( A \) whose non-zero entries are illustrated below.

Calculate the sparsity of \( A \).
Recall that the sparsity of a matrix is given by

$$\text{Sparsity} = \left( \frac{\text{non-zero elements of } A}{\text{total number of elements in } A} \right) \%.$$ 

The number of nonzero entries in a spy plot is given by the variable \(nz\). The dimension of the matrix is given by the grid size, which in this case is \(9 \times 4 = 36\). Hence the sparsity is

$$\text{Sparsity} = \left( \frac{154}{36 \times 36} \right) \% \approx 11.9\%.$$
Vector norm

Properties of vector norms

A vector norm $\| \cdot \|$ is an operation on the vector with the following properties:

- $\| x \| \geq 0$ with $\| x \| = 0$ only if $x = 0$.
- **Triangle inequality**: $\| x + y \| \leq \| x \| + \| y \|$.
- For any constant $\alpha$, $\| \alpha x \| = |\alpha| \| x \|$.

Vector $p$-norms

Vector $p$ norms are special types of norms on $n \times 1$ vectors. By definition, for $p \geq 1$, the $p$-norm of an $n \times 1$ vector is given by

$$\| x \|_p = \left( \sum_{i=1}^{n} x_i^p \right)^{1/p}$$
Vector $p$ norms

Examples of $p$-norms

- Vector 1-norm:
  \[ \| \mathbf{x} \|_1 = \sum_{i=1}^{n} |x_i|. \]

- Vector 2-norm (Euclidean norm):
  \[ \| \mathbf{x} \|_2 = \left( \sum_{i=1}^{n} x_i^2 \right)^{1/2} = \sqrt{\mathbf{x}^T \mathbf{x}}. \]

- Vector $\infty$-norm (max norm):
  \[ \| \mathbf{x} \|_{\infty} = \max_{1 \leq i \leq n} |x_i|. \]
Matrix norms

Properties of vector norms

A matrix norm $\| \cdot \|$ is an operation on a matrix with the following properties:

- $\| A \| \geq 0$ with $\| A \| = 0$ only if $A = 0$.
- **Triangle inequality**: $\| A + B \| \leq \| A \| + \| B \|$.
- For any constant $\alpha$, $\| \alpha A \| = |\alpha| \| A \|$.

Consistent matrix norms

A matrix norm is said to be consistent if

$$\| AB \| \leq \| A \| \| B \|.$$
Matrix $p$-norms

For $p \geq 1$, the $p$-norm of an $m \times n$ matrix is given by

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$
Examples of matrix $p$-norms

- Matrix 1-norm (maximum column sum):
  \[
  \|A\|_1 = \max_{1 \leq j \leq n} \left( \sum_{1 \leq i \leq m} |a_{ij}| \right).
  \]

- Matrix $\infty$-norm (maximum row sum):
  \[
  \|A\|_\infty = \max_{1 \leq j \leq m} \left( \sum_{1 \leq i \leq n} |a_{ij}| \right).
  \]

- Matrix 2-norm (square root of the largest eigenvalue of $A^T A$):
  \[
  \|A\|_2 = \sqrt{\max_{1 \leq j \leq n} \lambda_j (A^T A)}
  \]
A square matrix $A$ is *non-singular* if and only if

- $\det(A) \neq 0$ (invertible).
- All eigenvalues of $A$ are non-zero.

**Condition number**

For a non-singular matrix $A$, the *condition number* is defined as

$$\kappa(A) = \|A\| \|A^{-1}\|.$$
Condition number

Properties of condition numbers

- $\kappa(A) \geq 1$ for consistent matrix norms.
- $\kappa(\alpha I) = 1$ for all $\alpha \neq 0$.
- For a real symmetric matrix, the 2-norm condition number is
  \[ \kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\max_{1 \leq i \leq n} |\lambda_i(A)|}{\min_{1 \leq i \leq n} |\lambda_i(A)|}. \]
- $A$ is said to be *ill-conditioned* if $\kappa(A) > \frac{1}{\varepsilon} \approx 10^{16}$. 

Presented by: Janzen Choi
Example: (MATH2089, S1 2010, Q1c)

The coefficient matrix $A$ and the right-hand-side vector $b$ are known to 8 significant figures, and

$$
\|A\| = 1.9 \times 10^1, \quad \|A^{-1}\| = 2.2 \times 10^3.
$$

What is the condition number $\kappa(A)$?

By definition, for non-singular matrices,

$$
\kappa(A) = \|A\| \|A^{-1}\|.
$$

Hence,

$$
\kappa(A) = \|A\| \|A^{-1}\| = \left( 1.9 \times 10^1 \right) \times \left( 2.2 \times 10^3 \right) = 4.18 \times 10^4.
$$
### Example

Let $A = \begin{bmatrix} 2 & -1 & 2 \\ -1 & 1 & -1 \\ 2 & -1 & 3 \end{bmatrix}$ and $A^{-1} = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix}$.

Compute the condition numbers $\kappa_\infty(A)$ and $\kappa_1(A)$.

The condition number $\kappa_\infty(A)$ is simply just

$$\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty.$$ 

The sum of the magnitude of the rows of $A$ are simply

$|2| + |−1| + |2| = 5$, $|−1| + |1| + |−1| = 3$, and $|2| + |−1| + |3| = 6$. Hence $\|A\|_\infty$ is just 6. We repeat the same process for $A^{-1}$ and get $\|A^{-1}\|_\infty = 4$. So

$$\kappa_\infty(A) = 6 \times 4 = 24.$$ 

Repeat the process for the columns to get $\kappa_1(A)$. 

---

Presented by: Janzen Choi
**Sensitivity of a linear system**

- Let \( \bar{x} \) be an *approximation* to \( x \). Then the absolute error of \( x \) is \( \|x - \bar{x}\| \) and the relative error is
  \[
  \rho_x = \frac{\|x - \bar{x}\|}{\|x\|}.
  \]

- Let \( \bar{A} \) be an approximation to \( A \). Then the absolute error is \( \|A - \bar{A}\| \) and the relative error is
  \[
  \rho_A = \frac{\|A - \bar{A}\|}{\|A\|}.
  \]

(**Theorem**) **Sensitivity of a linear system**

The sensitivity of a linear system \( A\mathbf{x} = \mathbf{b} \) to the error in input data \( A \) and \( \mathbf{b} \) is given by

\[
\rho_x \approx \kappa(A) \times (\rho_A + \rho_b).
\]
Sensitivity of a linear system

Properties of the errors

- If $A$ or $b$ are known exactly, then the errors $\rho_A$ and $\rho_x$ are 0. That is, there is no error in precision.
- If $x$ is known to $k$ significant figures, then

$$\rho_x \leq 0.5 \times 10^{-k}.$$
Example: (MATH2089, 2009 Q1c)

The following MATLAB code generates the given output for a pre-defined real square array \( A \).

```matlab
chk1 = norm(A - A', 1)
chk1 = 1.4052e-015
ev = eig(A);
evlim = [min(ev) max(ev)]
evlim = 4.5107e-002 9.1213e+004
```

- Is \( A \) symmetric?
- Is \( A \) positive definite?
- Calculate the 2-norm condition number \( \kappa_2(A) \) of \( A \).
- When solving the linear system \( Ax = b \), the elements of \( A \) and \( b \) are known to 6 significant decimal digits. Estimate the relative error in the computed solution \( \bar{x} \).
- Given the Cholesky factorisation \( A = R^T R \), explain how to solve the linear system \( Ax = b \).
Example: (MATH2089, 2009 Q1c)

The following MATLAB code generates the given output for a pre-defined real square array $A$.

```matlab
cchk1 = norm(A - A', 1)
cchk1 = 1.4052e-015
```

Is $A$ symmetric?

From the MATLAB command $cchk1 = \text{norm}(A - A', 1)$, this implies that $\| A - A^\top \|_1 \approx 1.4 \times 10^{-15} \approx 7\varepsilon$ where $\varepsilon = 2.2 \times 10^{-16}$. Hence the value is small enough such that

$$\| A - A^\top \|_1 \approx 0 \implies A = A^\top.$$  

Thus $A$ is symmetric with rounding error.
Example: (MATH2089, 2009 Q1c)

The following MATLAB code generates the given output for a pre-defined real square array $A$.

```matlab
ev = eig(A);
evlim = [min(ev) max(ev)]
evlim = 4.5107e-002 9.1213e+004
```

- Is $A$ positive definite?

Because $A$ is symmetric, then the following statements are equivalent.

- $A$ is positive definite.
- All of the eigenvalues in $A$ are positive.

From our MATLAB command, we see that the minimum eigenvalue, given by the command `min(ev)`, is $4.5 \times 10^{-2} > 0$. Hence all of the eigenvalues are positive and thus, $A$ is positive definite.
Example: (MATH2089, 2009 Q1c)

The following MATLAB code generates the given output for a pre-defined real square array $A$.

```matlab
ev = eig(A);
evlim = [min(ev) max(ev)]
evlim = 4.5107e-002 9.1213e+004
```

- Calculate the 2-norm condition number $\kappa_2(A)$ of $A$.

For a real symmetric matrix, the 2-norm condition number is

$$
\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{|\lambda_{\text{max}}(A)|}{|\lambda_{\text{min}}(A)|} = \frac{9.12 \times 10^4}{4.51 \times 10^{-2}} = 2.02 \times 10^6.
$$
Example: (MATH2089, 2009 Q1c)

When solving the linear system $A\mathbf{x} = \mathbf{b}$, the elements of $A$ and $\mathbf{b}$ are known to 6 significant decimal digits. Estimate the relative error in the computed solution $\bar{\mathbf{x}}$.

Since $A$ and $\mathbf{b}$ are known to 6 significant decimal digits, then we have

$$\rho_A \leq 0.5 \times 10^{-6}, \quad \rho_b \leq 0.5 \times 10^{-6}.$$  

Then we have

$$\rho_x \approx \kappa_2(A) [\rho_A + \rho_b] \leq (2 \times 10^6) \left[ 0.5 \times 10^{-6} + 0.5 \times 10^{-6} \right] = 2.$$  

Hence the relative error of $\mathbf{x}$ is 2.
Example: (MATH2089, 2009 Q1c)

- Given the Cholesky factorisation \( A = R^T R \), explain how to solve the linear system \( Ax = b \).

Apply forward substitution and back substitution. Let \( A = R^T R \) so that

\[
Ax = b \implies R^T Rx = b \implies R^T y = b,
\]

where \( Rx = y \). Solve \( R^T y = b \) by forward substitution to get \( Rx = y \) and solve \( Rx = y \) by back substitution to get \( x \).
Part II: Least Squares & Polynomial Interpolation
Least squares

- Given a set of data points, determine the line or curve of best fit.

Methods to finding least squares

For a given $m \times n$ matrix $A$ with $m > n$, we can apply two methods to finding least square solutions.

1. **Normal equation**: $A^T A u = A^T y$.
   - $O(mn^2)$ flops.

2. **QR factorisation** and **back substitution**.
   - $O(mn^2)$ flops.
Method 1: Normal equations

Assumptions.

- $A$ is an $m \times n$ matrix (with $m > n$) and $A$ has full rank.

\[ Au = y \implies (A^T A) u = A^T y. \]

- Define a new matrix $B$ to be the matrix $B = A^T A$. $B$ is symmetric and positive definite.

- Solve $B u = A^T y$ by applying either Cholesky or LU factorisation with forward and backward substitutions.

Cost of method and issue

- Dominated by computing $B$: $O(mn^2)$ flops.

- Issue: Condition number is squared!

\[ \kappa_2(B) = \kappa_2(A^T A) = [\kappa_2(A)]^2 \]
Method 2: QR factorisation

We can try and write $A$ as a product of two matrices

$$A = QR,$$

where $Q$ is an orthogonal matrix and $R$ is an upper triangular matrix.

- $Q = \begin{bmatrix} Y & Z \end{bmatrix}$, where $Y$ is an $m \times n$ matrix and $Z$ is an $m \times (m - n)$ matrix.

Cost of method

- Cost: $\mathcal{O}(mn^2)$ flops.
**Polynomial interpolation**

- **Idea**: We want to make a new polynomial (interpolation) that passes through the data points.

Given data points \((x_0, y_0)\) and \((x_1, y_1)\), we solve the simultaneous equation

\[
p(x) = a_0 + a_1 x \implies \begin{cases} y_0 = a_0 + a_1 x_0 \\ y_1 = a_0 + a_1 x_1 \end{cases}
\]

Given \(n\) number of data points, an interpolating polynomial will have degree \((n - 1)\).
Interpolating polynomials in Lagrange form

Given \((n + 1)\) data points \((x_j, y_j)\), construct Lagrange polynomials of degree \(n\) of the form

\[
\ell_i(x) = \prod_{\substack{j=0 \atop j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \quad \text{for } i = 0, \ldots, n.
\]

- Note that \(\ell_i(x_j) = 1\) for \(i = j\) and \(\ell_i(x_j) = 0\) if \(i \neq j\).
- The interpolating polynomial is then

\[
p(x) = \sum_{i=0}^{n} y_i \ell_i(x).
\]
For a function $f$, the following data are known:

\[
    f(0) = 12.6, \quad f(1) = 6.7, \quad f(2) = 4.3, \quad f(3) = 2.7.
\]

- What is the degree of the interpolating polynomial $P$ for these data?
- Assume that we want to find $P$ in the form

\[
    P(x) = a_0 + a_1 x + \cdots.
\]

- Write down the system of linear equations you need to solve to obtain $a_0, a_1, \ldots$.
- Use MATLAB to set up and solve this linear system.

- Write down the Lagrange polynomials $\ell_j(x)$ for $j = 0, 1, 2, 3$.
- Write down the interpolating polynomial $P$ using the Lagrange polynomials.
For a function $f$, the following data are known:

$$f(0) = 12.6, \quad f(1) = 6.7, \quad f(2) = 4.3, \quad f(3) = 2.7.$$

What is the degree of the interpolating polynomial $P$ for these data?

As there are 4 data values and a polynomial of degree $n$ has $n + 1$ coefficients, then the degree of the interpolating polynomial is $n = 4 - 1 = 3$. 
Assume that we want to find $P$ in the form

$$P(x) = a_0 + a_1 x + \cdots.$$ 

Write down the system of linear equations you need to solve to obtain $a_0, a_1, \ldots$.

Use MATLAB to set up and solve this linear system.

As the interpolating polynomial is of degree 3, then

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3.$$ 

We obtain the following system of linear equations

$$f(0) = 12.6 \implies P(0) = a_0 = 12.6$$
$$f(1) = 6.7 \implies a_0 + a_1 + a_2 + a_3 = 6.7$$
$$f(2) = 4.3 \implies P(2) = a_0 + 2a_1 + 4a_2 + 8a_3 = 4.3$$
$$f(3) = 2.7 \implies a_0 + 3a_1 + 9a_2 + 27a_3 = 2.7.$$
Assume that we want to find $P$ in the form

$$P(x) = a_0 + a_1 x + \cdots.$$ 

Write down the system of linear equations you need to solve to obtain $a_0, a_1, \ldots$.

Use MATLAB to set up and solve this linear system.

Use the backslash command to solve for $a$ to obtain the following solution

$$P(x) = 12.6 - 8.5x + 3.1x^2 - 0.45x^3.$$
Write down the Lagrange polynomials $\ell_j(x)$ for $j = 0, 1, 2, 3$.

Recall that the Lagrange polynomials are the degree $n$ polynomials

$$\ell_j(x) = \prod_{\substack{k=0 \atop k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$ 

For the given data $x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 3$, this gives

$$\ell_0(x) = \frac{(x - 1)(x - 2)(x - 3)}{(0 - 1)(0 - 2)(0 - 3)} = -\frac{1}{6} (x - 1)(x - 2)(x - 3)$$

$$\ell_1(x) = \frac{(x - 0)(x - 2)(x - 3)}{(1 - 0)(1 - 2)(1 - 3)} = \frac{1}{2} x(x - 2)(x - 3)$$

$$\ell_2(x) = \frac{(x - 0)(x - 1)(x - 3)}{(2 - 0)(2 - 1)(2 - 3)} = -\frac{1}{2} x(x - 1)(x - 3)$$

$$\ell_3(x) = \frac{(x - 0)(x - 1)(x - 2)}{(3 - 0)(3 - 1)(3 - 2)} = \frac{1}{6} x(x - 1)(x - 2)$$
For a function $f$, the following data are known:

$$f(0) = 12.6, \quad f(1) = 6.7, \quad f(2) = 4.3, \quad f(3) = 2.7.$$ 

Write down the interpolating polynomial $P$ using the Lagrange polynomials.

The interpolating polynomial is simply

$$P(x) = \sum_{j=0}^{3} f_j \ell_j(x)$$ 

$$= 12.6 \times \ell_0(x) + 6.7 \times \ell_1(x) + 4.3 \times \ell_2(x) + 2.7 \times \ell_3(x)$$ 

$$= -\frac{12.6}{6}(x - 1)(x - 2)(x - 3) + \frac{6.7}{2}x(x - 2)(x - 3) +$$ 

$$-\frac{4.3}{2}x(x - 1)(x - 3) + \frac{2.7}{6}x(x - 1)(x - 2).$$
(Theorem) Interpolating polynomial error

If \( f \) is \((n + 1)\) times continuously differentiable on the interval \([a, b]\), then the error in approximating \( f(x) \) by \( p(x) \) is

\[
f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{j=0}^{n} (x - x_j)
\]

for some unknown \( \xi \in [a, b] \) depending on \( x \).
Chebyshev points

- Choose $x_j$ to minimise

$$
\max_{x \in [-1,1]} \prod_{j=0}^{n} (x - x_j).
$$

- On $[-1,1]$, set $t_j = \cos \left( \left( \frac{2n + 1 - 2j}{2n + 2} \right) \pi \right)$ for $j = 0, \ldots, n$.

- On $[a, b]$, set $x_j = \frac{a + b}{2} + \left( \frac{b - a}{2} \right) t_j$ for $j = 0, \ldots, n$.

- Chebyshev nodes are the zeros of the Chebyshev polynomial $T_{n+1}(x)$.

- Interpolation error is minimised by choosing Chebyshev nodes!
Part III: Nonlinear Equations
Nonlinear equation in standard form

\[ f(x) = 0, \quad x \in \mathbb{R}. \]

- We aim to solve for \( x \) (ie find the zeros of \( f \))
- If necessary, rearrange the equation to have the equation in standard form.
Notation

Continuity and differentiability

- If $f \in C^n([a, b])$, then $f$ is continuous on $[a, b]$ and $n$ times differentiable on the interval $(a, b)$.

Results of continuity

- (Intermediate Value Theorem) If $f \in C([a, b])$ and $f(a)f(b) < 0$, then there exists at least one zero of $f$ in the interval on $(a, b)$.

- (Strictly monotone) If $f'(x) > 0$ OR $f'(x) < 0$ for all $x \in (a, b)$, then $f$ is strictly monotone on the interval $[a, b]$.

- (Uniqueness of root) If $f$ is continuous AND strictly monotone on the interval $[a, b]$ and $f(a)f(b) < 0$, then $f$ has a unique root on $(a, b)$.
Iterative methods for solving equations

Iterations

- Have an initial guess or starting point $x_1$.
- Generate sequences of iterates $x_k$ for $k = 2, 3, \ldots$ based on approximations of the problem.
- Determine whether the sequence generated converges to the true solution $x^*$. 

Order of convergence

- Largest $\nu$ such that
  \[
  \lim_{k \to \infty} \frac{e_{k+1}}{e_k^\nu} = \beta,
  \]
  
  where $\beta$ is the asymptotic constant and $e_k = |x_k - x^*|$. 

Presented by: Janzen Choi
Iterative method: Bisection

- Suppose that $f(a)f(b) < 0$ and we have $f \in C([a, b])$.
- Take the midpoint $x_{\text{mid}} = \frac{a + b}{2}$ as $x_1$.
- Choose a new interval depending on the result of $f(x_{\text{mid}})$.
  - If $f(a)f(x_{\text{mid}}) < 0$, then choose new interval to be $[a, x_{\text{mid}}]$.
  - If $f(x_{\text{mid}})f(b) > 0$, then choose $[x_{\text{mid}}, b]$.
- Iterate using the same process.

Order of convergence

- Linear convergence with asymptotic constant $\beta = \frac{1}{2}$. 

Presented by: Janzen Choi
Iterative method: Fixed point iteration

Given a starting point $x_1$, compute

$$x_{k+1} = g(x_k), \quad \text{for } k = 1, 2, \ldots$$

Order of convergence

- If $g \in C^1([a, b])$ and there exists a $K \in (0, 1)$ such that $|g'(x)| \leq K$ for all $x \in (a, b)$, then the fixed point iteration converges linearly with asymptotic constant $\beta \leq K$, for any $x_1 \in [a, b]$. 
Iterative method: Newton’s method

Newton’s approximation

- Approximate $f(x)$ by its tangent at the point $(x_k, f(x_k))$ to form
  \[
  f(x) \approx f(x_k) + (x - x_k)f'(x_k).
  \]
- Choose $x = x_{k+1}$ and set $f(x) = 0$ to form
  \[
  x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad \text{assuming } f'(x_k) \neq 0.
  \]

Order of convergence

- If $f \in C^2([a, b])$ and $x_1$ is sufficiently close to a simple root $x^* \in (a, b)$, then Newton’s method converges quadratically to $x^*$. 
Example: (MATH2089, S1 2009)

To find the root of a real number, computers typically implement Newton’s method. Let $a > 1$ and consider finding the cube root of $a$, that is $a^{1/3}$.

Show that Newton’s method can be written as

$$x_{k+1} = \frac{1}{3} \left( 2x_k + \frac{a}{x_k^2} \right).$$
We want to find \( x = a^{1/3} \implies x^3 - a = 0 \). So set \( f(x) = x^3 - a \).

We then have

\[
f(x_k) = x_k^3 - a, \quad f'(x_k) = 3x_k^2.
\]

By Newton’s method, we obtain the result

\[
x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^3 - a}{3x_k^2} = x_k - \frac{1}{3}x_k + \frac{a}{3x_k^2} = \frac{1}{3} \left[ (3x_k - x_k) + \frac{a}{x_k^2} \right] = \frac{1}{3} \left( 2x_k + \frac{a}{x_k^2} \right).
\]
Example: (MATH2089, S1 2013, Q1d)

Consider the function \( f(x) = e^x \sin(x) - 100 \).

- You are given that \( f(x) \) has a simple zero at \( x^* \approx 6.443 \). If you use a starting value \( x_1 \) near \( x^* \), what is the expected order of convergence for Newton’s method?

The function behaves well since \( e^x \) and \( \sin(x) \) are continuous functions. So it passes the theorem! Hence the expected order of convergence is 2.
Iterative method: Secant method

- Approximate $f(x)$ by a line through the point $(x_k, f(x_k))$ and $(x_{k-1}, f(x_{k-1}))$ to form

$$f(x) \approx f(x_k) + (x - x_k) \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

- Choose $x = x_{k+1}$ and set $f(x) = 0$ to form

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}.$$

Order of convergence

- If $f \in C^2([a, b])$ and $x_1, x_2$ are sufficiently close to $x^*$, then the secant method converges superlinearly with order $\nu = \frac{1 + \sqrt{5}}{2}$. 

Presented by: Janzen Choi
Part IV: Numerical Differentiation and Integration
A **Taylor series** is an infinite polynomial series that approximates non-polynomial functions by taking higher order derivatives centred around a point $x_0$.

### Examples of well-known Taylor series

- $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots$
- $\ln(1 + x) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}x^k}{k} = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots$ for $|x| < 1.$
(Theorem)

Let $f \in C^{n+1}([a, b])$. In other words, let $f$ be continuous on $[a, b]$ and $n + 1$ times differentiable on $(a, b)$. Then

$$f(x + h) = \sum_{k=0}^{n} \frac{f^{(k)}(x)}{k!} h^k + \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1}$$

$$= f(x) + f'(x)h + \frac{f''(x)}{2!} h^2 + \cdots + \frac{f^{(n)}(x)}{n!} h^n + O(h^{n+1})$$

for some unknown $\xi \in (a, b)$.
Finite difference methods

Forward difference approximation

Let \( f \in C^2([a, b]) \). That is, let \( f \) be twice differentiable in the interval \([a, b]\). Then

\[
f'(x) = \frac{f(x + h) - f(x)}{h} + \mathcal{O}(h).
\]

- The roundoff error is

\[
\mathcal{O}\left(\frac{\varepsilon}{h}\right) = \varepsilon \left| \frac{f(x + h) - f(x)}{h} \right|.
\]

- The truncation error is \( \mathcal{O}(h) \) and the total error is \( \mathcal{O}\left(\frac{\varepsilon}{h}\right) + \mathcal{O}(h) \).
Central difference approximation

Let $f \in C^4([a, b])$. That is, let $f$ be four times differentiable on the interval $[a, b]$. Then

$$f''(x) = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2} + O(h^2).$$

- The roundoff error is $O\left(\frac{\varepsilon}{h^2}\right)$ and the truncation error is $O(h^2)$.

- The total error is $O\left(\frac{\varepsilon}{h^2}\right) + O(h^2)$. 
Quadrature rules

- We are approximating integrals using weighted sums of functions values. That is,

\[ Q_N(f) = \sum_{j=0 \text{ or } 1}^{N} w_j f(x_j), \]

where \( \sum_j w_j = b - a. \)

- Quadrature error:

\[ E_N = I(f) - Q_N(f) = \int_{a}^{b} f(x) \, dx - Q_N(f) \]

- We want \( E_N \to 0 \) as \( N \to \infty \) for convergence.
Quadrature rules

We look at three quadrature rules:

- Trapezoidal rule
- Simpson’s rule
- Gauss-Legendre rule
Quadrature rule — Trapezoidal rule

\[ Q_N(f) = h \left( \frac{f_0}{2} + f_1 + f_2 + \cdots + f_{N-1} + \frac{f_N}{2} \right). \]

- Approximate an integral using a bunch of trapeziums and sum up the area under \( f(x) \) using the area of each trapezium.
- The height \( h \) is fixed: \( h = \frac{b-a}{N} \).
- Function values are \( f_j = f(x_j) \) for all \( j = 0, \ldots, N \).
- Weights are \( w_0 = w_N = \frac{h}{2} \) and \( w_j = h \) for all \( h = 1, \ldots, N - 1 \).
Quadrature rule – Trapezoidal rule

(Theorem) Error of trapezoidal rule

Let $f \in C^2([a, b])$. Then

$$E_N(f) = -\frac{b-a}{12}h^2f''(\xi),$$

for some unknown $\xi \in [a, b]$.

- Rate of convergence: $E_N(f) = \mathcal{O}(h^2)$ or $E_N(f) = \mathcal{O}(N^{-2})$. 

Presented by: Janzen Choi
**Quadrature rule — Simpson’s rule**

\[ Q_N(f) = \frac{h}{3} \left( f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{N-2} + 4f_{N-1} + f_N \right). \]

- Approximate an integral using a bunch of parabolas and sum up the area under \( f(x) \) using the area of each parabola through integration.

- The height is fixed: \( h = \frac{b - a}{N} \), with \( N \) being even.

- Function values are \( f_j = f(x_j) \) for all \( j = 0, \ldots, N \).

- Weights are \( w_0 = w_N = \frac{h}{3} \) and \( w_j = \begin{cases} \frac{4h}{3} & \text{for odd } j \\ \frac{2h}{3} & \text{for even } j \end{cases} \).
Quadrature rule — Simpson’s rule

(Theorem) Error of Simpson’s rule

Let \( f \in C^4([a, b]) \). Then

\[
E_N(f) = -\frac{b - a}{180} h^4 f^{(4)}(\xi),
\]

for some unknown \( \xi \in [a, b] \).

- Rate of convergence: \( E_N(f) = \mathcal{O}(h^4) \) or \( E_N(f) = \mathcal{O}(N^{-4}) \).
Quadrature rule — Gauss-Legendre rule

\[ \int_{-1}^{1} f(x) \, dx \approx Q_N(f) = \sum_{j=1}^{N} w_j f(x_j). \]

- Nodes \( x_j \) are the zeros of the Legendre polynomial of degree \( N \) on \([-1, 1]\).
- Weights \( w_j \) are given in terms of the Legendre polynomials.
Quadrature rule – Gauss-Legendre rule

(Theorem) Error of Gauss-Legendre rule

Let $f \in C^{2N}([-1, 1])$. Then

$$E_N(f) = -\frac{e_N}{(2N)!} f^{(2N)}(\xi),$$

where $\xi \in [-1, 1]$ and $e_N$ is some number that depends on $N$. 
Quadrature properties

- Quadratures assume integrand \( f \) is sufficiently smooth on \([a, b]\).
  - Assume that \( f \in C^2([a, b]) \) for trapezoidal rule.
  - Assume that \( f \in C^4([a, b]) \) for Simpson’s rule.
  - Assume that \( f \in C^{2N}([a, b]) \) for Gauss-Legendre rule.
Change of variables

Transform integral

\[
\int_{a}^{b} f(x) \, dx \rightarrow \frac{b - a}{2} \int_{-1}^{1} f \left( \frac{a + b}{2} + \frac{b - a}{2} y \right) \, dy
\]

by substituting

\[
x = \frac{a + b}{2} + \frac{b - a}{2} y.
\]
Tips for estimating difficult integrals

- Unbounded derivatives: apply a change of variables.
- Discontinuity on derivative: split the integral and remove the discontinuous derivative.
- High oscillatory: requires a special analytic method.
- Narrow spike: either underestimate or overestimate the spikes.
Part V: Ordinary Differential Equations
First order ODE

Ordinary differential equations are equations that involve its derivative. A first order differential equation is one such equation where the highest order of derivative is 1. A first order initial value problem is simply a first order ODE with initial conditions.

- First order ODE: \( \frac{dy}{dx} = y \).
- First order IVP: \( \frac{dy}{dx} = y \) with \( y(0) = 1 \).
Existence and uniqueness of solutions

(Theorem)

If \( f(t, y) \) and \( \frac{\partial f(t, y)}{\partial y} \) are continuous and bounded for all \( t \in [t_0, t_{\text{max}}] \) and \( y \in \mathbb{R} \), then the IVP has a unique solution in the time interval \( [t_0, t_{\text{max}}] \).
Euler’s method

Solve a first order IVP \( y' = f(t, y), \quad t \in [t_0, t_{\text{max}}], \quad y(t_0) = y_0 \)
by

\[
y_{n+1} = y_n + h \cdot f(t_n, y_n), \quad n = 0, 1, \ldots, N - 1.
\]
System of first order ODEs

- Often times, we may have a system of many equations involving derivatives.
- We can write it in the form

\[
\frac{dx}{dt} = f(t, x).
\]
Example: (MATH2089, 2010 Q2b)

Consider the initial value problem (IVP)

\[ y''' + 2y' - (\pi^2 + 1)y = \pi(\pi^2 + 1)e^{-t}\sin(\pi t) \]
\[ y(0) = 1, \quad y'(0) = -1, \quad y''(0) = 1 - \pi^2. \]

Reformulate the IVP as a system of first-order differential equations

\[ x'(t) = f(t, x) \]

with the appropriate initial condition.
Begin by observing that the degree of derivative is 3. Hence, we use

\[
x = \begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} = \begin{bmatrix}
y \\
y' \\
y''
\end{bmatrix}.
\]

Then we see that

\[
x' = \begin{bmatrix}
x_1' \\
x_2' \\
x_3'
\end{bmatrix} = \begin{bmatrix}
y' \\
y'' \\
y'''
\end{bmatrix} = \begin{bmatrix}
x_2 \\
x_3 \\
y'''
\end{bmatrix},
\]

where \( y''' \) is just

\[
y''' = \pi(\pi^2 + 1)e^{-t}\sin(\pi t) - 2x_2 + (\pi^2 + 1)x_1.
\]
So we obtain

\[ \mathbf{x}' = \mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} x_2 \\ x_3 \\ \pi(\pi^2 + 1)e^{-t}\sin(\pi t) - 2x_2 + (\pi^2 + 1)x_1 \end{bmatrix}. \]

To find the appropriate initial conditions, take \( t = 0 \) to get

\[ \mathbf{x}(0) = \begin{bmatrix} y(0) \\ y'(0) \\ y''(0) \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 - \pi^2 \end{bmatrix}. \]
2-stage Runge Kutta method

\[ k_1 = f(t_n, y_n), \quad k_2 = f\left(t_n + \frac{2}{3} h, y_n + \frac{2}{3} hk_1\right) \]

\[ y_{n+1} = y_n + \frac{h}{4} [k_1 + 3k_2] \]

4-stage Runge Kutta method

\[ k_1 = f(t_n, y_n), \quad k_2 = f\left(t_n + \frac{1}{2} h, y_n + \frac{1}{2} hk_1\right) \]

\[ k_3 = f\left(t_n + \frac{1}{2} h, y_n + \frac{1}{2} hk_2\right), \quad k_4 = f\left(t_n + h, y_n + hk_3\right) \]

\[ y_{n+1} = y_n + \frac{h}{6} [k_1 + 2k_2 + 2k_3 + k_4]. \]
Example

Find the solution of the initial value problem

\[ y' = 3y + 3t, \quad y(0) = 1, \ t = 0.2 \]

Using Euler’s method with \( h = 0.2 \).

Using the fourth-order Runge Kutta method with \( h = 0.2 \).
Using Euler's method:

- We observe that, here, \( f(t_n, y_n) = 3y + 3t \).
- Next, using our initial value of \( y(t_0) = y_0 \implies y(0) = 1 \), we obtain \( t_0 = 0 \) and \( y_0 = 1 \).
- Next, we want to find \( y_{0.2} \) given \( t = 0.2 \).

\[
y_{0.2} = y_0 + h [f(t_0, y_0)] \\
= 1 + 0.2 [3y_0 + 3t_0] \\
= 1 + 0.2 [3 + 0] \\
= 1.6.
\]
Using the fourth-order Runge Kutta method:

- Observe that
  \[ y_{0.2} = y_0 + \frac{h}{6} [k_1 + 2k_2 + 2k_3 + k_4]. \]

- Calculate the values of \( k_1, k_2, k_3 \) and \( k_4 \) respectively.

  \[ k_1 = f(t_0, y_0) = 3. \]

  \[ k_2 = f \left( t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_1 \right) = 4.2 \]

  \[ k_3 = f \left( t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_2 \right) = 4.56 \]

  \[ k_4 = f \left( t_0 + h, y_0 + hk_3 \right) = 6.336. \]

Then,

\[ y_{0.2} = 1 + \frac{0.2}{6} \left[ 3 + 2 \times 4.2 + 2 \times 4.56 + 6.336 \right] = 1.8952. \]
Other useful methods for solving IVPs

**Taylor method of order 2**

\[
y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left. \frac{\partial f(t, y)}{\partial t} \right|_{t=t_n, y=t_n}
\]

**Implicit Euler’s method**

\[
y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})
\]

**Trapezoidal method**

\[
y_{n+1} = y_n + \frac{h}{2} \left[ f(t_n, y_n) + f(t_{n+1}, y_{n+1}) \right]
\]

**Heun’s method**

\[
y_{n+1} = y_n + \frac{h}{2} \left[ f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n)) \right]
\]
Part VI: Partial Differential Equations
Partial Differential Equations

- Partial Differential Equations (PDEs) are functions of *more than one variable* defined by equations involving their *partial derivatives*.

- Order of the PDE is the order of the highest derivative present!
Finite difference methods

- Treat them no differently to functions of one variable.
- The only difference is changing the variable in the derivative!

\[
\frac{\partial u(x, t)}{\partial x} = \frac{u(x + h, t) - u(x - h, t)}{2h} + O(h^2).
\]
Types of PDEs

We will restrict ourselves to PDEs involving only two variables. A second order quasi-linear PDE is of the form

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} = F \left( x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right).$$

- Elliptic if $B^2 - 4AC < 0$.
- Parabolic if $B^2 - 4AC = 0$.
- Hyperbolic if $B^2 - 4AC > 0$. 
Elliptic PDEs

- Divide $x$ interval $[0, L_x]$ into $m + 1$ equal length subintervals such that
  \[ h_x = \frac{L_x}{m + 1}. \]

- Divide $y$ interval $[0, L_y]$ into $n + 1$ equal length subintervals such that
  \[ h_y = \frac{L_y}{n + 1}. \]

- Central difference approximation of $O(h^2)$ at the grid points $x_i$
  \[
  \frac{\partial^2 u}{\partial x^2}(x_i, y_j) \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2},
  \]
  \[
  \frac{\partial^2 u}{\partial y^2}(x_i, y_j) \approx \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h_y^2}.
  \]
Parabolic PDEs

Method 1 (Explicit method)

- **Forward** difference approximation to the time derivative
  \[
  \frac{\partial u}{\partial t}(x_i, t_\ell) \approx \frac{u_{i}^{\ell+1} - u_{i}^{\ell}}{h_t}.
  \]

- **Central** difference approximation to the space derivative
  \[
  \frac{\partial^2 u}{\partial x^2}(x_i, t_\ell) \approx \frac{u_{i-1}^{\ell} - 2u_{i}^{\ell} + u_{i+1}^{\ell}}{h_x^2}.
  \]

- Substitute into PDE and multiply by \( h_t \) to obtain
  \[
  u_{i}^{\ell+1} = su_{i-1}^{\ell} + (1 - 2s)u_{i}^{\ell} + su_{i+1}^{\ell}, \quad s = \frac{Dh_t}{h_x^2}.
  \]
Parabolic PDEs

Method 2 (Implicit method)

- **Backward** difference approximation to the time derivative
  \[
  \frac{\partial u}{\partial t}(x_i, t_{\ell+1}) \approx \frac{u_i^{\ell+1} - u_i^\ell}{h_t}.
  \]

- **Central** difference approximation to the space derivative
  \[
  \frac{\partial^2 u}{\partial x^2}(x_i, t_{\ell+1}) \approx \frac{u_{i-1}^{\ell+1} - 2u_i^{\ell+1} + u_{i+1}^{\ell+1}}{h_x^2}.
  \]

- Substitute into PDE and multiply by $h_t$ to obtain
  \[
  -su_{i-1}^{\ell+1} + (1 - 2s)u_i^{\ell+1} - su_{i+1}^{\ell+1} = u_i^\ell, \quad s = \frac{Dh_t}{h_x^2}.
  \]
Method 3 (Crank-Nicolson method)

Take the average between the explicit and implicit method to obtain

\[ u_i^{\ell+1} = u_i^\ell + \frac{s}{2} \left[ (u_{i-1}^\ell - 2u_i^\ell + u_{i+1}^\ell) + (u_{i-1}^{\ell+1} - 2u_i^{\ell+1} + u_{i+1}^{\ell+1}) \right] \]

Stability of methods

- Explicit method: Stable if and only if \( s \leq \frac{1}{2} \).
Hyperbolic PDEs

Method (Explicit method)

- Central difference approximation to the time derivative
  \[
  \frac{\partial^2 u}{\partial t^2}(x_i, t_\ell) \approx \frac{u_{i}^{\ell-1} - 2u_{i}^{\ell} + u_{i}^{\ell+1}}{h_t^2}.
  \]

- Central difference approximation to the space derivative
  \[
  \frac{\partial^2 u}{\partial x^2}(x_i, t_\ell) \approx \frac{u_{i-1}^{\ell} - 2u_{i}^{\ell} + u_{i+1}^{\ell}}{h_x^2}.
  \]

- Substitute into PDE and multiply through $h_t^2$ to obtain
  \[
  u_{i}^{\ell+1} = ru_{i-1}^{\ell} + 2(1 - r)u_{i}^{\ell} + ru_{i+1}^{\ell} - u_{i}^{\ell-1}, \quad r = \frac{c^2 h_t^2}{h_x^2}.
  \]
A final problem

The heat conduction equation which models the temperature in an insulated rod with ends held at constant temperatures can be written in the dimensionless form as

\[
\frac{\partial \Theta(x, t)}{\partial t} = \frac{\partial^2 \Theta(x, t)}{\partial x^2}
\]

Write a finite difference approximation of this equation using the Forward-Time, Central-Space scheme and rearrange it to be solved by an explicit method.

By applying the FTCS scheme, we get

\[
\frac{\partial \Theta}{\partial t}(x_i, t_\ell) \approx \frac{\Theta_{i+1}^\ell - \Theta_i^\ell}{h_t} \frac{\partial^2 \Theta(x_i, t_\ell)}{\partial x^2} \approx \frac{\Theta_{i-1}^\ell - 2\Theta_i^\ell + \Theta_{i+1}^\ell}{h_x^2}.
\]
By rearranging to allow explicit solving, we get

\[
\Theta_{i+1}^\ell = \left( \frac{\Delta t}{\Delta x^2} \right) \Theta_{i-1}^\ell + \left( 1 - \frac{2\Delta t}{\Delta x^2} \right) \Theta_i^\ell + \left( \frac{\Delta t}{\Delta x^2} \right) \Theta_{i+1}^\ell.
\]