Contents

Topic 1: Numerical Integration

Numerical Methods:

Trapezoid Method

- Uses a first-order polynomial approximation to improve accuracy.
- Calculates the area under a curve using linear functions instead of step functions.
- Trapezoid method involves evaluating the function at two points and averaging the result.
- Accuracy improves by reducing the step size or increasing the number of panels.
- Function reuse between iterations speeds up the computation.

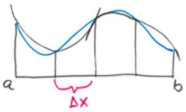
Function Approximation by Polynomials:

- **Zero-order Polynomial**: Provides a coarse estimate of an integral unless the step size is very small. This requires evaluating the integrand many times.
- **First-order Polynomial (Trapezoid Rule)**: Represents the integrand as a straight line. While it poorly captures curvature, it can approximate slopes.
- **Improving Accuracy**: Two ways to improve accuracy:
  1. Use a higher-order polynomial (p-type improvement).
  2. Use a smaller step size or more steps (n-type improvement).

Simpson's Rule (2nd Order Polynomial)

- Is higher order – approximates area under a parabola.
- Given the end points a and b and the midpoint c, Simpson's Rule can be expressed as:

$$S_k = \frac{1}{3}\Delta x_k \left[f_a + f_b + 4 \sum_{i=1,3,5\ldots}^{n-1} f(a + i\Delta x_k) + 2 \sum_{i=2,4,6\ldots}^{n-2} f(a + i\Delta x_k) \right]$$



- Odd steps = weight 4
- Even = weight 2 (because they are shared as endpoints for neighbouring parabolas.
- Even steps – requires dividing the interval into even portions

For the Trapezoid Rule the weights are (1, 2,… 2, 1) and for Simpson's Rule (1, 4, 2, 4,… 2, 4, 1) and the points come from dividing the interval some number of times

- **Accuracy** – proportional to the fourth derivative of the integrand (can excetly integrate a 3rd order polynomial)

We can compute simpson's rule results from trapezoid rule: $S_k = T_K + \frac{T_K - T_{K-1}}{3}$

Ways to improve accuracy

1. **Higher-order Polynomials**: Using higher-order polynomials can improve accuracy but makes the code more complex.
2. **Smaller Step Size**: Increasing accuracy by reducing the step size comes at the expense of increased computation time.
3. **Adaptive Algorithms**: These algorithms adjust step sizes based on the difficulty of the function's behaviour, using smaller steps in tricky areas.

Applications:

- Integrate raw data (must be evenly spaced).
- Uneven – can still be done. each set of 2 or 3 points must be integrated separately, adding complexity

Topic 2: Gaussian Quadrature

A method of integration using weights and nodes. Is more accurate compared to other methods, especially for higher degree polynomials.

- Higher accuracy with less points.
- No need for evenly spaced points
- Applications like FEM, **computational mechanics**, **electromagnetism**, and **biomechanics** to solve complex integrals that appear in numerical methods.

Legendre Polynomials

- Points used are derived from the roots of Legendre polynomials
- Roots determine the sampling points for the quadrature.

Interval: defined over [-1,1], we use change of variables for other forms.

Gaussian Quadrature

- The Gaussian quadrature algorithm is defined for the interval from -1 to 1 and sampling points and weights are symmetrical about 0
- *n* is the number of points being used

$$\int_{-1}^{1} f(x)dx \approx w_0 f(x_0) + \sum_{i=1}^{k} w_i[f(x_i) + f(-x_i)]$$

for odd n, $k = \frac{n-1}{2}$

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{k} w_i[f(x_i) + f(-x_i)]$$

for even n, $k = \frac{n}{2}$

The interval of integration

- Specific to the case of mapping an integrand to the -1 to 1 interval and using *x* to symbolize the variable on the original interval *a* to *b* and ξ to be the dummy variable that fits the interval -1 to 1

$$-1 \le \xi \le 1 \leftrightarrow a \le x \le b$$

$$\frac{\xi - (-1)}{1 - (-1)} = \frac{x - a}{b - a}$$

$$\therefore x = \left(\frac{b-a}{2}\right)\xi + \left(\frac{b+a}{2}\right)$$

$$dx = \left(\frac{b-a}{2}\right)d\xi$$

$$\int_{a}^{b} f(x)dx = \int_{-1}^{1} f\left[\left(\frac{b-a}{2}\right)\xi + \left(\frac{b+a}{2}\right)\right]\left(\frac{b-a}{2}\right)d\xi$$

Topic 3: Roots of Non-Linear Algebraic Equations:

There are different ways we can find the roots of non-linear equations.

1. Graphical
2. Stepping
3. Bisection

Graphical Plot the function and visually inspect the root locations.

- Quick
- Reduced accuracy
- not automatic
- cannot be integrated into larger algorithms.

Stepping Method change a range of values for changes in sign of the function – this will indicate where the root is.

- Accuracy will depend on the step size – if multiple roots lie in the same step, they can be missed.
- Not always reliable

Bisection Method given an interval where the function changes sign, the interval is successively halved until you find the root.

Steps:

$$: x_m = \frac{x_0 + x_1}{2}.$$

- Compute midpoint
- If $f(x0)\cdot f(xm)<0$, the root lies between x_0 and x_m, otherwise, it's between x_m and x_1
- Continue refining until $x_1 - x_0$ is less than a tolerance.
- Slow convergence

$$n \geq \frac{log\left(\frac{x1 - x0}{\varepsilon}\right)}{\log(2)}$$

- Number of iterations:

Newton's Method based on Taylor expansion of the function near an initial guess.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Iteratively update the guess using until the change is within a given tolerance.
- Requires derivative calculation and can fail if derivative is 0 or small.

Secant Method derivative-free

- Approximating derivative with finite differences.
- Slower than newton
- Does not always converge

$$f'(x) \approx \frac{f(x + dx) - f(x)}{dx}$$

- Start with an initial guess x0 and correction dx

$$x_1 = x_0 - \frac{f(x_0)}{\frac{f(x_0 + dx) - f(x_0)}{dx}}$$

$$x_1 = x_0 - dx\frac{f(x_0)}{f(x_0 + dx) - f(x_0)}$$

- Iterate

$$x_n = x_{n-1} - (x_{n-1} - x_{n-2})\frac{f(x_{n-1})}{f(x_{n-1}) - f(x_{n-2})}$$

until $|x_n - x_{n-1}| < \varepsilon$

MATLAB fzero Function

- **Overview**: A built-in MATLAB function that combines the bisection, secant, and inverse quadratic interpolation methods.
- **Disadvantage**: May return discontinuous points or fail for certain root types (e.g., where the function touches but doesn't cross the x-axis).