

# NUMERICAL ANALYSIS FORMULAS

## ABSOLUTE ERROR

Absolute Error:  $|\text{True} - \text{Approx}|$

Relative Error:  $\left| \frac{\text{Error}}{\text{True}} \right|$

Percentage Error:  $\left( \frac{\text{Error}}{\text{True}} \right) \times 100$

Machine Epsilon: Smallest  $\epsilon$  such that  $1 + \epsilon \neq 1$

Condition Number:  $\kappa = |f'(x)|^*$

Bisection Method:  $c = \frac{a+b}{2}$

## TRAPEZOIDAL RULE

Trapezoidal Rule:  $\int \approx \frac{h}{2}(y_0 + y_n + 2\sum y_i)$

Simpson's 1/3:  $\int \approx \frac{h}{3}(y_0 + y_n + 4\sum_{\text{odd}} + 2\sum_{\text{even}})$

Simpson's 3/8:  $\int \approx \frac{3h}{8}(y_0 + y_n + 3\sum + 2\sum)$

Gaussian Quadrature:  $\sum w_i f(x_i)$

Romberg Integration: Extrapolation of trapezoidal

Euler Method:  $y_{n+1} = y_n + hf(x_n, y_n)$

## CONVERGENCE (BISECTION)

Convergence (Bisection): Error  $\leq \frac{b-a}{2^n}$

Fixed Point Iteration:  $x_{n+1} = g(x_n)$

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

Secant Method:

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

Regula Falsi: Linear interpolation

Order of Convergence:  $p$

## IMPROVED EULER

Improved Euler: Avg slope method

Runge-Kutta (RK4): Weighted slopes ( $k_1, k_2, k_3, k_4$ )

Finite Difference: Replace derivatives with differences

FTCS Scheme: Forward time, centered space

Crank-Nicolson: Average of implicit & explicit

LU Decomposition:  $A = LU$

## GAUSSIAN ELIMINATION

Gaussian Elimination: Row reduction

Jacobi Iteration:  $x_i = \frac{b_i - \sum a_{ij}x_j}{a_{ii}}$

Gauss-Seidel: Updated values used

SOR Method: Relaxation factor  $\omega$

Conjugate Gradient: Solve  $Ax=b$  efficiently

Power Method:  $\lambda \approx$  dominant eigenvalue

## INTERPOLATION

Interpolation:  $f(x) \approx P(x)$

Lagrange Polynomial:  $\sum_i^n y_i L_i(x)$

Newton Divided Diff:  $f[x_0, \dots, x_n]$

Forward Difference:  $\Delta y = y_{n+1} - y_n$

Backward Difference:  $\nabla y = y_n - y_{n-1}$

Central Difference:  $\delta y = \frac{y_{n+1} - y_{n-1}}{2}$

## INVERSE POWER

Inverse Power: Smallest eigenvalue

QR Algorithm:  $A = QR$

FFT:  $O(n \log n)$

Stability: Small error growth

Stiff Equation: Requires stable methods