Numerical Methods Cheat Sheet

Compiled with LaTeX

Root Finding Methods

- **Bisection Method:** Error bound after n steps:

$$|x_n - x^*| \leq \frac{b-a}{2^n}$$

- **Newton's Method:**

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

  Requires $f'(x)$ and has quadratic convergence.

- **Secant Method:** Approximates $f'(x)$:

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

Polynomial Interpolation

- **Lagrange Interpolation:**

$$P(x) = \sum_{j=0}^{n} f(x_j)\ell_j(x), \quad \ell_j(x) = \prod_{i \neq j} \frac{x - x_i}{x_j - x_i}$$

- **Error:**

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

Least Squares Approximation

- Model: $A\beta = y$, where A contains basis functions.
- Solve using Normal Equations:

$$A^T A\beta = A^T y$$

Numerical Integration

- **Trapezoid Rule:**

$$\int_a^b f(x)dx \approx \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right]$$

- **Simpson's Rule:**

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[f(a) + 4\sum f(x_{\text{odd}}) + 2\sum f(x_{\text{even}}) + f(b)\right]$$

- **Error (Trapezoid):**

$$E_T = -\frac{(b-a)^3}{12n^2} f''(\xi)$$

Essential Python Functions

- `numpy`:
  - `np.linspace, np.polyfit, np.polyval`
  - `np.linalg.solve`
- `matplotlib.pyplot`:
  - `plt.plot, plt.scatter, plt.grid, plt.legend`
- `scipy`:
  - `optimize.newton, integrate.quad`

Notebook Index

- **Root Finding:** `1-RootFinding-Corr.ipynb`
  - `newton` (cell 29, 40), `bisection` (cell 23), `secant` (cell 26)
- **Polynomial Approximation:** `2-PolynomialApproximation-Corr.ipynb`
  - `interpolation` (cell 2, 16), `lagrange` (cell 16), `least squares` (cell 39, 40)
- **Numerical Integration:** `3-NumericalIntegration-Corr.ipynb`
  - `composite trapezoid` (cell 15), `simpson` (cell 25)