Numerical Methods Formulas
Gürkan Hoca | Unicourse

Preliminaries

- **True Error (E_t):** $E_t =$ True Value $-$ Approximate Value
- **Approximate Error (E_a):** $E_a =$ Current Approx. $-$ Previous Approx.
- **Absolute Error:** | Error |
- **Relative Error:** $\epsilon = \frac{|\text{Error}|}{|\text{Reference Value}|} \times 100\%$

(Ref. can be True or Approx. value)

Solution of Nonlinear Equations $f(x) = 0$

- **Fixed point iteration Method:** $x_{i+1} = g(x_i)$, where $x = g(x)$ is equivalent to $f(x) = 0$. Converges if $|g'(x)| < 1$ near root.
- **Bisection Method:** Given $[a, b]$ with $f(a)f(b) < 0$, $x_m = \frac{a+b}{2}$. New interval $[a, x_m]$ or $[x_m, b]$.
- **False Position Method (Regula Falsi):** Given $[a, b]$ with $f(a)f(b) < 0$, $x_r = b - \frac{f(b)(a-b)}{f(a)-f(b)}$. Update a or b.
- **Newton-Raphson Method:** $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
- **Convergence of N-R:** $M = 1 : Quadratic, M > 1 : Linear$
- **Accelerated Newton Raphson Method:** For root of multiplicity m: $x_{i+1} = x_i - m\frac{f(x_i)}{f'(x_i)}$
- **Convergence of Accel. N-R:** $Quadratic$
- **Secant Method:** $x_{i+1} = x_i - \frac{f(x_i)(x_{i-1}-x_i)}{f(x_{i-1})-f(x_i)}$

Analytical Solution of Linear Systems

- **Upper and Lower Triangular Linear Systems:** Solve by back-substitution (Upper) or forward-substitution (Lower).
- **Gaussian Elimination and Pivoting:** Use elementary row operations to achieve row echelon form (Upper Triangular). Pivoting swaps rows to avoid division by small/zero elements.
- **LU Decomposition and Permutation matrices:** $A = LU$ or $PA = LU$, where L is lower triangular (unit diagonal often), U is upper triangular, P is permutation matrix.
- **Solution of linear systems with LU decomposition:** Solve $A\mathbf{x} = \mathbf{b}$ by first solving $L\mathbf{y} = P\mathbf{b}$ (forward sub.), then $U\mathbf{x} = \mathbf{y}$ (back sub.).

Iterative Solution Methods for Linear Systems

- **Diagonally Dominant:** $|a_{diagonal}| > \sum_{j \neq i} |a_{others}|$ in each row.
- **Gauss Jacobi Method:** Use same values for each iter.
- **Gauss Seidel Method:** Use most recent values.

Iterative Solution Method for Nonlinear Systems: Newton's Method

- **Newton's Method for Systems:** Iterate $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J(\mathbf{x}^{(k)})^{-1}\mathbf{f}(\mathbf{x}^{(k)})$, where Jacobian Matrix: $J_{ij} = \frac{\partial f_i}{\partial x_j}$.

Interpolation and Polynomial Approximation

- **Lagrange Polynomials:** $P_n(x) = \sum_{i=0}^{n} y_i L_{n,i}(x)$, where $L_{n,i}(x) = \prod_{j=0,j\neq i}^{n} \frac{x-x_j}{x_i-x_j}$
- **Newton's Divided Difference table:** $P_n(x) = f[x_0] + f[x_0,x_1](x-x_0) + \cdots + f[x_0,\ldots,x_n]\prod_{i=0}^{n-1}(x-x_i)$.

Differences: $f[x_i,\ldots,x_j] = \frac{f[x_{i+1},\ldots,x_j]-f[x_i,\ldots,x_{j-1}]}{x_j-x_i}$

Curve Fitting

- **Least-squares Line:** $y = a_0 + a_1 x$. Solve normal equations: $\begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix}$
- **Least-squares Parabola:** $y = a_0 + a_1 x + a_2 x^2$. Solve corresponding normal equations (3x3 system).
- **Data Linearization:** Transform model to linear form, e.g., $y = ae^{bx} \to \ln y = \ln a + bx$; $y = ax^b \to \log y = \log a + b\log x$.
- **Bezier Curve Fitting:** Parametric curve $\mathbf{P}(t) = \sum_{i=0}^{n}\mathbf{P}_i B_{i,n}(t)$, where $\mathbf{P}_i$ are control points and $B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$ are Bernstein polynomials.
- **Interpolation by Spline Functions:** Piecewise polynomials (e.g., cubic splines) joined at knots with continuity conditions on function and derivatives.

Numerical Differentiation

- **Approximating the Derivative:** Use finite differences based on Taylor series.
- **Forward Difference:** $f'(x_i) \approx \frac{f(x_{i+1})-f(x_i)}{h}$
- **Backward Difference:** $f'(x_i) \approx \frac{f(x_i)-f(x_{i-1})}{h}$
- **Central Difference:** $f'(x_i) \approx \frac{f(x_{i+1})-f(x_{i-1})}{2h}$ (More accurate)
- **Second Derivative (Central):** $f''(x_i) \approx \frac{f(x_{i+1})-2f(x_i)+f(x_{i-1})}{h^2}$

Numerical Integration (Quadrature)

- **Trapezoidal Method:** $\int_a^b f(x)dx \approx \frac{h}{2}[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)]$, $h = \frac{b-a}{n}$
- **Mid-Point Method:** $\int_a^b f(x)dx \approx h\sum_{i=0}^{n-1} f(x_i + h/2)$, $h = \frac{b-a}{n}$
- **Simpson's 1/3 Method (n even):** $\int_a^b f(x)dx \approx \frac{h}{3}[f(x_0) + 4\sum_{i=1,\text{odd}}^{n-1} f(x_i) + 2\sum_{i=2,\text{even}}^{n-2} f(x_i) + f(x_n)]$
- **Simpson's 3/8 Method (n multiple of 3):** $\int_a^b f(x)dx \approx \frac{3h}{8}[f(x_0) + 3\sum_{i=1,4,7..}^{n-2}(f(x_i) + f(x_{i+1})) + 2\sum_{i=3,6,9..}^{n-3} f(x_i) + f(x_n)]$

Numerical Solution of ODEs

- **Euler's method:** For $y'(x) = f(x,y)$, $y(x_0) = y_0$: $y_{i+1} = y_i + hf(x_i, y_i)$, where $h = x_{i+1} - x_i$.

Eigenvalues and Eigenvectors

- **Analytical solution:** Solve characteristic equation $\det(A-\lambda I) = 0$ for eigenvalues λ. Solve $(A-\lambda I)\mathbf{v} = \mathbf{0}$ for eigenvectors $\mathbf{v}$.
- **Power Method:** Finds dominant eigenvalue (λ_1) and eigenvector. Iterate $\mathbf{y}^{(k+1)} = A\mathbf{x}^{(k)}$, $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)}/\|\mathbf{y}^{(k+1)}\|_\infty$. Eigenvalue estimate $\lambda_1 \approx \|\mathbf{y}^{(k+1)}\|_\infty/\|\mathbf{x}^{(k)}\|_\infty$ or Rayleigh quotient.
- **Inverse Power Method:** Finds eigenvalue closest to shift σ. Iterate $(A-\sigma I)\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)}$, $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)}/\|\mathbf{y}^{(k+1)}\|_\infty$. Eigenvalue $\lambda \approx \sigma + 1/$(eigenvalue estimate from Power Method applied to $(A-\sigma I)^{-1}$).