# Numerical and Statistical Analysis Methods

|  |  |  |  |
| --- | --- | --- | --- |
| Main Topic | Methods | Formulae | Comments |
| Polynomial Root-Finding | Muller's method | Iterative quadratic approximation formula | Suitable for complex roots |
|  | Birge-Vieta method | Iterative scheme to refine roots for polynomials | Used for algebraic equations |
|  | Graeffe’s root squaring method | Transformation method to isolate roots | Finds all roots simultaneously |
| Simultaneous Equations | Gauss-Jacobi method | x\_i^(k+1) = (1/a\_ii) \* (b\_i - Σ a\_ij \* x\_j^(k)) | Requires diagonal dominance; simple iteration |
|  | Gauss-Seidel method | Similar to Jacobi but updates variables immediately | Faster convergence than Jacobi method |
| Interpolation | Newton’s Forward Interpolation | f(x) = f\_0 + pΔf\_0 + (p(p-1)/2!)Δ²f\_0 + ... | Used when values are near the beginning |
|  | Newton’s Backward Interpolation | Similar to forward interpolation formula | Used when values are near the end |
|  | Stirling’s Interpolation | Combination of forward and backward differences | Used for values near the middle of the table |
|  | Bessel’s Interpolation | Improved accuracy for mid-range interpolation | Preferred for central interpolation |
|  | Lagrange Interpolation | f(x) = Σ ((x-x\_j)/(x\_i-x\_j))f(x\_i) | Useful for unequal intervals |
| Numerical Differentiation | Forward Difference Formula | f'(x) ≈ (f(x+h) - f(x))/h | First-order approximation |
|  | Central Difference Formula | f'(x) ≈ (f(x+h) - f(x-h))/(2h) | More accurate than forward difference |
| Numerical Integration | Trapezoidal Rule | ∫f(x)dx ≈ (h/2) [f(a) + 2Σf(x\_i) + f(b)] | Simple and effective for uniformly spaced data |
|  | Simpson’s 1/3 Rule | ∫f(x)dx ≈ (h/3) [f(a) + 4Σf(odd) + 2Σf(even) + f(b)] | High accuracy for even sub-intervals |
|  | Simpson’s 3/8 Rule | Similar to 1/3 rule but h = 3 | Used when intervals are not divisible by 2 but divisible by 3 |
|  | Romberg Integration | Recursive trapezoidal refinement | Increases accuracy using extrapolation |
| Numerical ODE Solutions | Taylor Series Method | y(x) = y\_0 + hy' + (h²/2)y'' + ... | Good for initial value approximations |
|  | Euler’s Method | y\_(n+1) = y\_n + hf(x\_n, y\_n) | Basic, but less accurate over large intervals |
|  | Improved Euler’s Method (Heun’s) | y\_(n+1) = y\_n + (h/2)[f(x\_n, y\_n) + f(x\_(n+1), y\_pred)] | More accurate than basic Euler |
|  | Runge-Kutta (2nd Order) | y\_(n+1) = y\_n + hK\_2 | Higher accuracy, requires function evaluations |
|  | Runge-Kutta (4th Order) | y\_(n+1) = y\_n + (h/6)(K\_1 + 2K\_2 + 2K\_3 + K\_4) | Standard for solving ODEs due to good accuracy |
|  | Milne's Predictor-Corrector Method | Combination of predictor and corrector steps | Needs multiple initial values |
| Statistical Distributions & Hypothesis | Probability Distributions | Different statistical distribution formulas | Understanding mean, variance, and distribution properties |
|  | Hypothesis Testing | Test statistics such as Z-test, T-test, Chi-square | Used for making statistical decisions |
| Non-Parametric & Time Series Analysis | Non-parametric Methods | Statistical methods without assuming a fixed distribution | Used for median-based and rank-based tests |
|  | Time Series Analysis | Moving averages, Exponential Smoothing | Analyzing data over time for trends and seasonal effects |

## Notes: Key Nuances for Method Selection

1. \*\*Interpolation:\*\*  
 - Use \*\*Stirling’s\*\* or \*\*Bessel’s\*\* for mid-range interpolation:  
 - Stirling’s is preferred if data points are evenly spaced with minimal variation.  
 - Bessel’s is better for more irregular data.  
 - Use \*\*Newton’s Forward\*\* when interpolating near the start, and \*\*Backward\*\* near the end.  
  
2. \*\*Integration:\*\*  
 - \*\*Simpson’s 1/3 Rule\*\*: Requires an even number of intervals.  
 - \*\*Simpson’s 3/8 Rule\*\*: Can handle odd numbers of intervals divisible by 3.  
 - \*\*Trapezoidal Rule\*\*: Use for simpler problems or as a basis for refining (e.g., Romberg).  
  
3. \*\*ODE Solvers:\*\*  
 - \*\*Euler’s Method\*\*: Use for a quick approximation, but expect higher error.  
 - \*\*Runge-Kutta Methods\*\*: Use when higher accuracy is needed, especially the 4th order method.  
 - \*\*Milne’s Predictor-Corrector\*\*: Requires multiple initial values but improves accuracy iteratively.  
  
4. \*\*Root-Finding:\*\*  
 - Muller's handles complex roots efficiently.  
 - Graeffe’s method finds all roots simultaneously for polynomials.