COMPARISON

**Non-Linear Equations:**

Bisection: Always converges, but is slow.

Newton-Raphson: More efficient than Bisection, but a derivate is required.

False-Position: Always converges, but can get stuck.

Secant Method: More efficient than Newton Method, but if the initial values are not close to the root, then it might not converge.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Bisection** | **Regular Falsi** | **Newton** | **Secant** |
| **Type** | Bracket | Bracket | Open | Open |
| **Num of Initial Guess** | 2 | 2 | 1 | 2 |
| **Rate of Convergence** | Slow but Steady | Slow | Faster | Faster |
| **Accuracy** | Great | Good | Good | Good |
| **Convergence** | Linear | Linear | Quadratic | Super Linear |
| **Method of Approach** | MidPoint | Interpolation | Taylor Series | Interpolation |
| **Prog Effect** | Easy | Tedious | Easy | Tedious |

**Interpolation:**

The Lagrange interpolation formula involves very considerable computation and its use can be quite risky. It is much more efficient to use the divided differences method for interpolation.

Lagrange's form is more efficient when you have to interpolate several data sets on the same data points.

Newton's form is more efficient when you have to interpolate data incrementally.

With Newton interpolation, you get the coefficients reasonably fast (quadratic time), the evaluation is much more stable (roughly because there is usually a single dominant term for a given x), the evaluation can be done quickly and straightforwardly.

Maximum errors in these interpolating polynomials have been calculated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No. | Function | Interval | Maximum percentage error in Newton’s interpolating  polynomial | Maximum percentage error in Lagrange’s interpolating  polynomial |
| 1 | cos x | [ -3, 3 ] | 0.064992690 | 2.294741840 |
| 2 | cos x | [-1, 1 ] | 0.000059420 | 0.049786950 |
| 3 | cos x | [ -2, 2 ] | 0.041294390 | 12.05671096 |
| 4 | sin x | [-1, 1 ] | 0.000200020 | 0.024267050 |
| 5 | sin x | [-2, 2 ] | 0.000600110 | 0.040910650 |
| 6 | sec x | [-1, 1 ] | 0.011954750 | 0.015129500 |
| 7 | cos-1x | [-1, 1 ] | 99.39362100 | 73.26376200 |
| 8 | sin-1x | [-1, 1 ] | 6.639571000 | 6.626192000 |
| 9 | ex | [0, 2 ] | 0.000022710 | 7.618216000 |
| 10 | √x | [0, 2 ] | 59.37311000 | 59.37330000 |
| 11 | x | [ 0, 2 ] | 0.000800000 | 28.09092000 |
| 12 | log x | [0.02, 2 ] | 1.547646000 | 111.1734000 |
| 13 | ( 1-x )-1/2 | [-1.5, 0.95] | 9.423458530 | 9.418445080 |
| 14 | (1+10x2)-1 | [-1, 1 ] | 588.8133427 | 588.8680960 |
|  | Average | | 765.3107 | 898.9139 |

Table 1. Maximum percentage error in Newton’s and Lagrange’s interpolating polynomials

Average of the maximum percentage error for the function in Newton’s interpolating polynomial is 765.3107 where as it is 898.9139 in Lagrange’s interpolating polynomial. It is clear that Newton’s interpolating polynomial is approximately 1.174574 times better than Lagrange’s interpolating polynomial.

**Numerical Integration:**

* Midpoint rule is one of the least accurate methods and can lead to very wrong result.
* Trapezoid rule is similar to midpoint rule, but instead of taking rectangles, we use trapezoid. In other words, we approximate by inscribing polygonal chain in the graph of the function, taking separate segment for each subinterval.
* Simpson’s rule is the most accurate method and the fastest convergent

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | n | The exact value | Simpson’s rule | Trapezoid rule | Midpoint rule |
| *sin*(*x*) | 0 | 0.77 | 1 | 0.28208933000000003 | 0.178675 | 0.26801206700000002 | 0.28918053500000002 |
|  | 0 | 0.77 | 4 | 0.28208933000000003 | 0.28209099999999998 | 0.28121769400000002 | 0.28252535000000001 |
|  | 0 | 0.77 | 10 | 0.28208933000000003 | 0.28208899999999998 | 0.28194994099999998 | 0.28215902999999998 |
| *x*3 | 0 | 2.5 | 1 | 9.765625 | 13.020799999999999 | 19.53125 | 4.8828125 |
|  | 0 | 2.5 | 4 | 9.765625 | 9.7656299999999998 | 10.37597656 | 9.4604492189999991 |
|  | 0 | 2.5 | 10 | 9.765625 | 9.7656299999999998 | 9.86328125 | 9.716796875 |
| *ex* | 0 | 2 | 1 | 6.3890560990000003 | 6.4207278040000002 | 8.3890560989999994 | 5.4365636569999998 |
|  | 0 | 2 | 4 | 6.3890560990000003 | 6.3891937250000002 | 6.5216101100000001 | 6.3229855329999998 |
|  | 0 | 2 | 10 | 6.3890560990000003 | 6.3890596439999996 | 6.4103387679999999 | 6.3784200819999999 |
| *log*(*x*) | 1 | 3 | 1 | 1.2958368659999999 | 1.2904003369999999 | 1.0986122890000001 | 1.2969442799999999 |
|  | 1 | 3 | 4 | 1.295836867 | 1.2957983500000001 | 1.2821045820000001 | 1.3026452340000001 |
|  | 1 | 3 | 10 | 1.2958368680000001 | 1.2958358109999999 | 1.2936188740000001 | 1.2969442799999999 |
| *x* | 0 | 2 | 1 | 2 | 2 | 2 | 2 |
|  | 0 | 2 | 4 | 2 | 2 | 2 | 2 |
|  | 0 | 2 | 10 | 2 | 2 | 2 | 2 |
| *x*3− *x* | 0 | 2 | 1 | 2 | 4 | 6 | 0 |
|  | 0 | 2 | 4 | 2 | 2 | 2.25 | 1.875 |
|  | 0 | 2 | 10 | 2 | 2 | 2.04 | 1.9799924799999999 |

Table I: Comparison results for selected methods

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | n | Error for Simpson’s rule | Error for trapezoid rule | Error for midpoint rule |
| *sin*(*x*) | 0 | 0.77 | 1 | 0.3666013537144141 | 4.9903565900870923E-2 | 2.5138153565741946E-2 |
|  | 0 | 0.77 | 4 | 5.9187314800328924E-6 | 3.0899296609271398E-3 | 1.5456808262322172E-3 |
|  | 0 | 0.77 | 10 | 1.1712211858265657E-6 | 4.9413208506619789E-4 | 2.4708439425509267E-4 |
| *x*3 | 0 | 2.5 | 1 | 0.33332991999999995 | 1 | 0.5 |
|  | 0 | 2.5 | 4 | 5.1199999998061685E-7 | 6.25E-2 | 3.125E-2 |
|  | 0 | 2.5 | 10 | 5.1199999998061685E-7 | 0.01 | 5.0000000000000001E-3 |
| *ex* | 0 | 2 | 1 | 4.9571806194195298E-3 | 0.31303528549466747 | 0.1490818717605904 |
|  | 0 | 2 | 4 | 2.1540908057286479E-5 | 2.0747041271432749E-2 | 1.0341209174399794E-2 |
|  | 0 | 2 | 10 | 5.5486599967827115E-7 | 3.3311132255614338E-3 | 1.6647242702880168E-3 |
| *log*(*x*) | 1 | 3 | 1 | 4.195380717004561E-3 | 0.1521986155624622 | 8.5459355281236604E-4 |
|  | 1 | 3 | 4 | 2.9723648848697756E-5 | 1.0597232838259647E-2 | 5.2540306157380376E-3 |
|  | 1 | 3 | 10 | 8.1568909350790468E-7 | 1.7116305723136754E-3 | 8.5459200808910485E-4 |
| *x* | 0 | 2 | 1 | 0 | 0 | 0 |
|  | 0 | 2 | 4 | 0 | 0 | 0 |
|  | 0 | 2 | 10 | 0 | 0 | 0 |
| *x*3− *x* | 0 | 2 | 1 | 1 | 2 | 1 |
|  | 0 | 2 | 4 | 0 | 0.125 | 6.25E-2 |
|  | 0 | 2 | 10 | 0 | 2.0000000000000018E-2 | 1.0003759765624953E-2 |

Table II: The results of the comparison of the error value.

**Ordinary Differential Equations:**

The 4 R-K Method gives the best approximate result as the error is very small as compared to Heun’s, Midpoint or Modified Euler. Heun’s method is slightly better than Midpoint Formula which is slightly better in approximation than Modified Euler.

As far as computation is concerned, the Midpoint formula is the easiest to compute, followed by Modified Euler and Heun’s Method. The 4-RK method requires most computation.