NUMERICAL Algorithms assignment 1

Applied in c++

Ellis Rourke

S5057468

question 1: Centered difference

Algorithm

RESULTs

Conclusion

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

question 2: Radar coordinates

Algorithm

Given the radar coordinates and time of the aircraft, we can firstly calculate the cartesian coordinates for each radar “ping” we have received. For a given point the coordinates can be calculated using the following

After finding these values for each of our points, we can use centered finite differences to calculate the velocity and acceleration, the first and second order derivative respectively; and represent these values as vectors. After computing each component of the vector (i, j and magnitude) we can combine and present these the represent the final vector for each point.

RESULTs

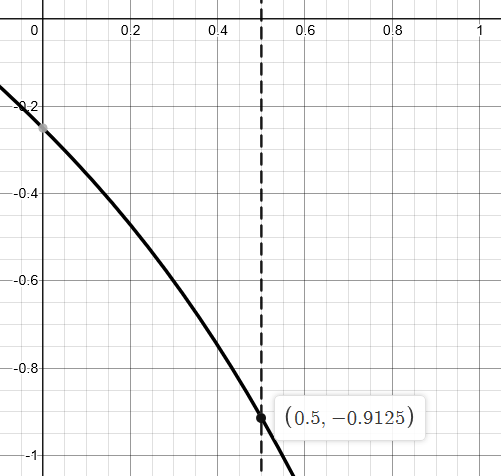
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T | X | Y | Velocity | | Acceleration | |
| 202 | 4037.2 | 3540.9 |  | 128.635 |  | 19.0689 |
| 204 | 4252.52 | 3581.85 |  | 121.122 |  | 12.1335 |
| 206 | 4509.92 | 3647 |  | 125.268 |  | 13.0394 |
| 208 | 4726.44 | 3744.55 |  | 114.038 |  | 5.26665 |

question 3: richardson extrapolation

Algorithm

Richardson Extrapolation and Central Difference are two different methods for approximating the derivate of a function at some point. Both methods base their calculations on some value h, where h is the size of each segment under the curve. Intuitively, as h approaches 0 the approximation will be closer to the exact value.  
we are given the function and can mathematically solve the derivative with respect to x which results in the function

Therefore, at the given value of the true value of the derivative is -0.9125



RESULTs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| h | Richardson value | Richardson error | Central difference value | Central difference error |
| 2 | -0.917969 | 0.60% | 1.2625 | 38.36% |
| 1 | -0.913867 | 0.15% | -1 | 9.59% |
| 0.5 | -0.912842 | 0.04% | -0.934375 | 2.40% |
| 0.25 | -0.912585 | 0.01% | -0.917969 | 0.60% |
| 0.1 | -0.912514 | 0.001% | -0.913375 | 0.10% |

Conclusion

As you can see from the above results, it is clear that both methods converge close to the real value of the derivative as the h value decreases, however the Richardson extrapolation method clearly converges on true much faster than the comparative central dereference calculations.

Although Richardson extrapolation runs many more iterations for each h value while central difference approximation is a single calculation.  
Even with quite a small h value of 0.1, Richardson extrapolation far outperforms the central difference method by over 100%. In theory, as h approaches 0, both methods would eventually converge on the exact value with an infinity small value of h.

question 4: romberg integration

Algorithm

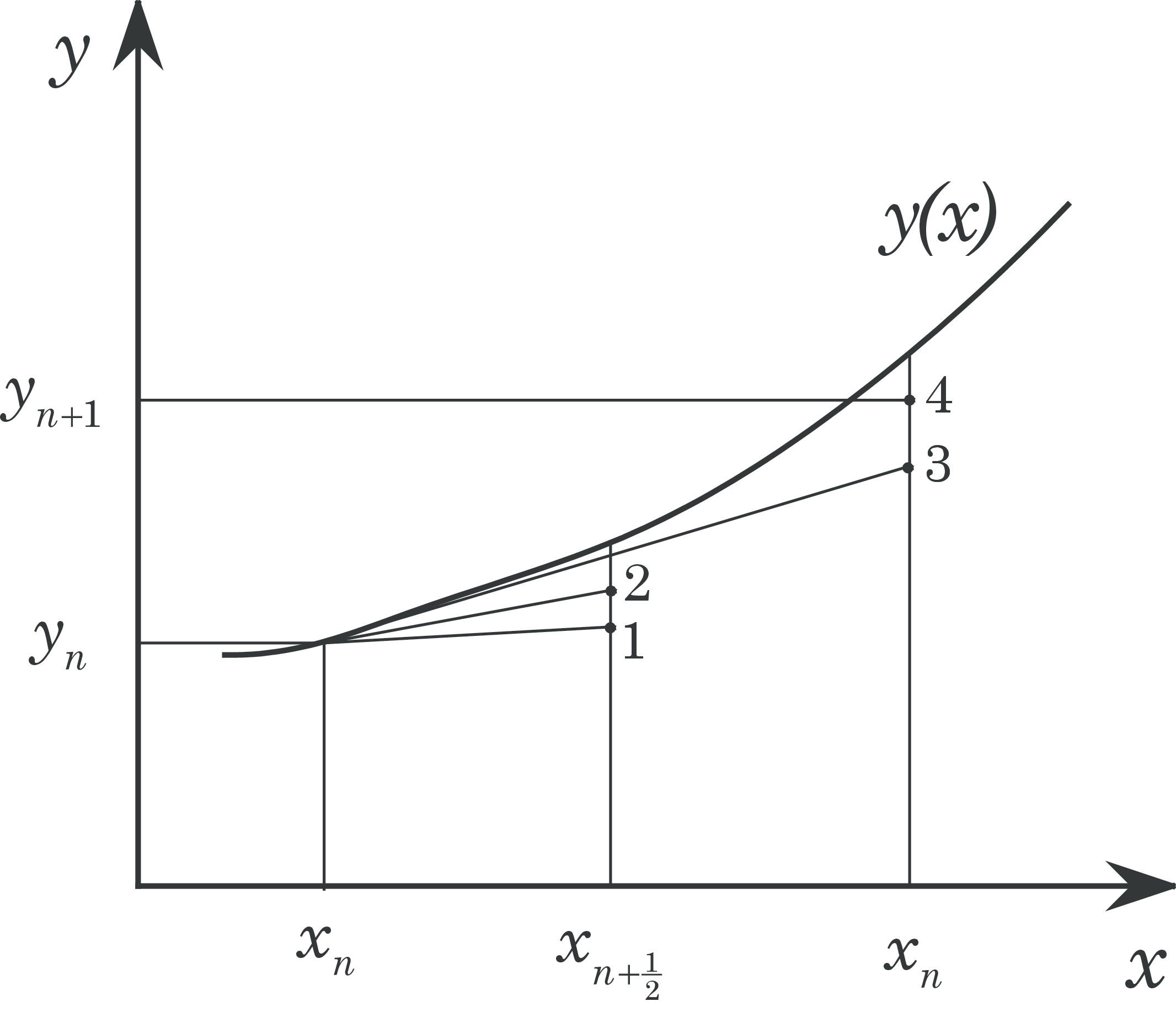
RESULTs

Conclusion

question 5: boat mast deflection

Algorithm

The question requires us to calculate the deflection of a boat mast given the parameters for wind, mast length and moment of inertia etc. We are given the differential equation to model this deflection.  
We can use an estimation of the differential equation to solve for our parameters. Since the equation is an ODE (Ordinary Differential Equation) we can use a Runge-Kutta method.

The RK4 method iterative steps through the ODE and computes next values by using a weighted average of 4 different points on the curve.

is the slope at the beginning of the interval

is the slope at the midpoint of the interval

is the slope of the midpoint again (but using k2 as the start)

is the slope at the end of the interval

RESULTs

|  |  |
| --- | --- |
| z | y |
| 0 | 0 |
| 1 | 0.00431994 |
| 2 | 0.00863988 |
| 3 | 0.0129598 |
| 4 | 0.0172798 |
| 5 | 0.0215997 |
| 6 | 0.0259196 |
| 7 | 0.0302396 |
| 8 | 0.0345595 |
| 9 | 0.0388794 |
| 10 | 0.0431994 |

question 6: function roots

Algorithm

We are tasked with finding the roots of the function by implementing 3 different estimation methods; Bisection, Newton-Raphson, and Secant. These algorithms work by converging on the root given we know two surrounding values.   
These methods will only work on continuous functions and aim to find points of the function that = 0, also known as the roots.   
  
The Bisection method works by choosing x values of the function, and iteratively “Bisecting” the values by a given interval and selecting the sub section where the function changes sign, as this must contain the root. This bisection method is fairly slow comparatively but can be used to find a rough estimate of the root of a function.

The Newton Raphson method is a faster method that bisection and quickly finds an approximation of the root by using a tangent line of the function and simple linear approximation. The algorithm quickly converges on the true value.

The Secant method is similar to the bisection method but instead of dividing each section using the midpoint, it divides the subsections by using the secant line formed by connecting the endpoints.

RESULTs

Given the start points we can implement the above algorithms to approximate our function. Each method is implemented in C++ and results are presented in comparison to the true value calculated with a graphing calculator.

|  |  |
| --- | --- |
| Approximation Method | Approximation |
| Bisection | 0.567144 |
| Newton Raphson | 0.567143 |
| Secant | 0.567143 |

As you can see, for 4 significant figures, all of the approximations are able to give an approximation correct for the required accuracy, when providing more the 4 significant figures, the algorithms may diverge from the true value.

