An Algorithm for Integration, Differentiation and Finding Root Numerically

Nizhum Rahman¹, Aminul Islam² and B. K. Datta³

Received: 12 December 2013 Accepted: 5 January 2014 Published: 15 January 2014

Abstract

Numerical analysis concerns the development of algorithms for solving various types of problems of mathematics; it is a vast-ranging field having deep interaction with computer science, mathematics, engineering, and the sciences. Numerical analysis mainly consists of Numerical Integration, Numerical Differentiation and finding Roots numerically. In this paper we develop an algorithm combination of Numerical Integration (Trapezoidal rule, Simpson’s ??/?? rule, Simpson’s ??/?? rule and Weddle’s rule.), Numerical Differentiation (Euler, modified Euler and Runge-Kutta second and fourth order) and finding Roots (Bisection method and False position method) numerically.

Index terms— computer science, mathematics, engineering, and the sciences.

1 Introduction

Numerical analysis is the area of mathematics and computer science that creates, analyzes, and implements algorithms for solving numerically the problems of continuous mathematics. Such problems originate generally from real-world applications of algebra, geometry, and calculus, and they involve variables which vary continuously. The formal academic area of numerical analysis varies from highly theoretical mathematical studies to computer science issues involving the effects of computer hardware and software on the implementation of specific algorithms [5].

An algorithm is a procedure or formula for solving a problem. The word derives from the name of the mathematician, Mohammed ibn-Musa al-Khwarizmi.

Given a set of data of points (?? 0 , ?? 0 ), (?? 1 , ?? 1 ), ? ? . . , (?? ?? , ?? ?? ) of a function ?? = ??(??) where ??(??) is not known explicitly, it is required to compute the value of the definite integral ?? = ? ?? ???? ?? ?? (1) We derive a general formula for numerical integration using Newton’s forward difference formula. Let the interval [??, ??] be divided into n equal subintervals such that?? = ?? 0 < ?? 1 < ? ? ? ? ? ? . < ?? ?? = ??

Clearly, ?? ?? = ?? 0 + ??. Hence the integral becomes ?? = ? ?? ???? ?? ?? 0

Approximating ?? by Newton’s forward difference formula, we obtain. [4] ?? = ? ?? 0 + ???? 0 + ?

Since ?? = ?? 0 + ???, ??? = ? ?? and hence the above integral becomes?? = ? ?? 0 + ???? 0 + ?

From this general formula, we can obtain different integration formula by putting?? = 1,2,3 ? ? etc.

Programming language C is very flexible and powerful. It originally designed in the early 1970s [3]. It allows us to maximum control with minimum command. It is recognized worldwide and used in a multitude of applications especially in Numerical Analysis. Along with other numerous benefits, we have used programming language C in this paper.

Author ? ? ? ?: Department of Mathematics, Pabna University of Science & Technology (PUST). e-mails: nl.nizhum@gmail.com, pust.pulok@gmail.com, bimaldu@gmail.com, rajpust09@gmail.com Author ?: Department
I. PROCEDURE

The outline of this paper is as follows: Section 2 contains the brief description of the existing methods with methodology. In Section 3, we develop an algorithm, using the programming language C, which gives us the solution of a problem simultaneously regarding four popular existing numerical integration methods namely Trapezoidal rule, Simpson’s 1/3 rule, Simpson’s 3/8 rule and Weddle’s rule or the solution of an ordinary differential equation simultaneously regarding four popular existing methods namely Euler, Runge-Kutta second and fourth order or Newton’s method of finding roots numerically.

Conclusions are given at the end at Section 4.

II. Existing Methods

We give a brief description of the existing methods of Numerical Integration like Trapezoidal rule, Simpson’s 1/3 rule, Simpson’s 3/8 rule and Weddle’s rule, methods of numerical differential equations like Euler, modified Euler, Runge-Kutta second and fourth order and numerical methods namely Bisection method and False position method in this section with their methodology.

\[ \int_a^b f(x) \, dx \approx \frac{b-a}{2} \left[ f(a) + f(b) \right] \]

This rule is known as Trapezoidal rule.

\[ \int_a^b f(x) \, dx \approx \frac{b-a}{3} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \]

This rule is known as Simpson’s 1/3 rule.

\[ \int_a^b f(x) \, dx \approx \frac{b-a}{8} \left[ 3f(a) + 3f\left(\frac{a+b}{2}\right) + f(b) \right] \]

This rule is known as Simpson’s 3/8 rule.

\[ \int_a^b f(x) \, dx \approx \frac{b-a}{6} \left[ f(a) + 3f(b) + 3f\left(\frac{a+b}{2}\right) + 2f\left(\frac{a+3b}{4}\right) \right] \]

This rule is known as Weddle’s rule.

Euler Method

In mathematics and computational science, the Euler method is a first-order numerical procedure for solving ODEs with a given initial value. It is the most basic explicit method for numerical ODEs [1].

i. Procedure

We consider the differential equation:

\[ \frac{dy}{dx} = f(x, y) \]

with the initial condition \( y(x_0) = y_0 \)
Suppose that we wish to solve the equation (3) with (4) for the value of $y$ at $x = x_0 + n \Delta x$, where $n = 1, 2, \ldots$. Integrating (3) with $x_0$ to $x_1$ and $y_0$ to $y_1$, we get $y_1 = y_0 + \int_{x_0}^{x_1} \frac{dy}{dx} \, dx$ (5) Assuming that $\frac{dy}{dx} = \frac{dy}{dx}(x_0, y_0)$ in $x_0 \leq x \leq x_1$, this gives Euler's formula $y_1 = y_0 + \int_{x_0}^{x_1} \frac{dy}{dx}(x_0, y_0) \, dx$ (6)

Similarly for the range $x_1 \leq x \leq x_2$, we have $y_2 = y_1 + \int_{x_1}^{x_2} \frac{dy}{dx} \, dx$

Substituting $\frac{dy}{dx}(x_1, y_1)$ for $\frac{dy}{dx}(x, y)$ where $x_1 \leq x \leq x_2$, we have $y_2 = y_1 + \int_{x_1}^{x_2} \frac{dy}{dx}(x_1, y_1) \, dx$ (7)

We thus obtain the iterative formula $y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} \frac{dy}{dx} \, dx$ for $n = 0, 1, 2, \ldots$

9 Substituting

10 And hence equation (16) becomes

This is the Runge-Kutta second order formula. h) Runge-Kutta method (Fourth order) i. Procedure

We mention the fourth order formulae defined by $y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} \frac{dy}{dx} \, dx$ for $n = 0, 1, 2, \ldots$ Where $\frac{dy}{dx} = \frac{dy}{dx}(x_n, y_n)$ in $x_n \leq x \leq x_{n+1}$, we obtain $y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} \frac{dy}{dx}(x_n, y_n) \, dx$ (17)

Where $\frac{dy}{dx} = \frac{dy}{dx}(x_0, y_0)$ in $x_0 \leq x \leq x_1$, we have $y_1 = y_0 + \int_{x_0}^{x_1} \frac{dy}{dx} \, dx$

Substituting $\frac{dy}{dx}(x_1, y_1)$ for $\frac{dy}{dx}(x, y)$ where $x_1 \leq x \leq x_2$, we have $y_2 = y_1 + \int_{x_1}^{x_2} \frac{dy}{dx}(x_1, y_1) \, dx$ (18)

Where the parameters have to be determined by expanding both sides of (17) by Taylor's series and securing agreement of terms up to and including those containing $\Delta x^4$. The choice of the parameters is, again arbitrary and we can have several fourth order Runge-Kutta formulae. If for example we set $\frac{dy}{dx} = \frac{dy}{dx}(x_0, y_0)$ in $x_0 \leq x \leq x_1$, we obtain $y_1 = y_0 + \int_{x_0}^{x_1} \frac{dy}{dx} \, dx$ (19)

Where $\frac{dy}{dx} = \frac{dy}{dx}(x_0, y_0)$ in $x_0 \leq x \leq x_1$, we have $y_1 = y_0 + \int_{x_0}^{x_1} \frac{dy}{dx} \, dx$

Substituting $\frac{dy}{dx}(x_1, y_1)$ for $\frac{dy}{dx}(x, y)$ where $x_1 \leq x \leq x_2$, we have $y_2 = y_1 + \int_{x_1}^{x_2} \frac{dy}{dx}(x_1, y_1) \, dx$ (20)

Where the bisection method is a rootfinding method which repeatedly bisects an interval and then selects a subinterval in which a root must lie for further processing. It is a very simple and robust method, but it is also relatively slow. Because of this, it is often used to obtain a rough approximation to a solution which is then used as a starting point for more rapidly converging methods. The method is also called the binary search method or the dichotomy method [4].
11 i.) Procedure

The method is applicable when we wish to solve the equation $f(x) = 0$ for interval $[a, b]$ and $f(a)f(b)$ the real variable, $x$ where $f$ is a continuous function defined on an and have opposite signs, so the method is applicable to this smaller interval.

12 j.) False Position Method

In problems involving arithmetic or algebra, the false position method or regula falsi is used to refer to basic trial and error methods of solving problems by substituting test values for the unknown quantities.

13 i.) Procedure

The poor convergence of the bisection method as well as its poor adaptability to higher dimensions (i.e., systems of two or more non-linear equations) motivate the use of better techniques. One such method is the Method of False Position. Here, we start with an initial interval $[a, b]$ function changes sign only once in this interval. Now we find an $x$ is given by of the problem. In other words, finding $3)$ $3) ( ( ) ( ) ( 1 2 1 2 1 3 x f x f x x x x ? ? ? = ? Now, $x$ is a static procedure in the case of the bisection method since for a given $1$ and $2$, it gives identical $3$, no matter what the function we wish to solve. On the other hand, the false position method uses the information about the function to arrive at $3$.  

14 Algorithm

INPUT: Type your choice C.

15 If C==1 {

INPUT: function $f(x)$, limits ?? 0 and ?? ?? , number of division ??, direct result ??.

Step-1: Compute $? = ?? ? ?? 0$ 

Step-2: Set $?? = 0$ 

Step-3: While ?? ? ??, repeat Step-4 

Step-4: Set $?? = ??(?? 0 + ??)$ 

Step-5: Set $?? = 1$ 

Step-6: While ?? < ??, repeat Step-7 

Step-7: Set $????_1 = ?????_1 + 2??(??)$ If ??2 = 0 Set $????_2 = ??0 = 2??(??)$ Else Set $????_2 = ??0 = 4??(??)$ 

Step-8: Set $????_1 = (??/2) * (??0 + ??) + ?????_1$ Else $????_1 = (??/3) * (??0 + ??) + ?????_1$ 

Step-9: Set $????_3 = (??/8) * (??0 + ??) + ?????_1$ If ??6 = 0 Set $????_4 = ?????_4 + 2??(??)$ Else if ??6 = 0 Set $????_4 = ?????_4 + 6??(??)$ Else Set $????_4 = ?????_4 + ??(??)$ Else Set $????_4 = ???? ?? 4 + 5??(??)$ Else 

The false position method differs from the bisection method only in the choice it makes for subdividing the interval at each iteration. It converges faster to the root because it is an algorithm which uses appropriate weighting of the initial end points $1$ and $2$ using the information about the function, or the data$????_4 = ?????_4 + ??(??)$; $????_4 = ??(??) + ??$ 

Step-10: Set $????_1 = ????_1 + 2??(??)$ If ??2 = 0 Set $????_2 = ??0 = 2??(??)$ Else Set $????_2 = ??0 = 4??(??)$ 

Step-11: Set $????_1 = (??/2) * (??0 + ??) + ?????_1$ Else $????_1 = (??/3) * (??0 + ??) + ?????_1$ 

Step-12: Set $????_3 = (??/8) * (??0 + ??) + ?????_1$ If ??6 = 0 Set $????_4 = ?????_4 + 2??(??)$ Else if ??6 = 0 Set $????_4 = ?????_4 + 6??(??)$ Else Set $????_4 = ?????_4 + ??(??)$ Else Set $????_4 = ???? ?? 4 + 5??(??)$ Else 

16 ??

Step-13: Set ?? = 0 

Step-14: While ?? ? ??, repeat Step-4 

Step-15: Set ?? = ??() ??(??) + ?????_4 

Step-16: Set ?? = 1 

Step-17: While ?? < ??, repeat Step-7 

Step-18: Set $????_1 = ?????_1 + 2??(??)$ If ??2 = 0 Set $????_2 = ??0 = 2??(??)$ Else Set $????_2 = ??0 = 4??(??)$ 

Step-19: Set $????_1 = (??/2) * (??0 + ??) + ?????_1$ Else $????_1 = (??/3) * (??0 + ??) + ?????_1$ 

Step-20: Set $????_3 = (??/8) * (??0 + ??) + ?????_1$ If ??6 = 0 Set $????_4 = ?????_4 + 2??(??)$ Else if ??6 = 0 Set $????_4 = ?????_4 + 6??(??)$ Else Set $????_4 = ?????_4 + ??(??)$ Else Set $????_4 = ???? ?? 4 + 5??(??)$ Else 

The false position method differs from the bisection method only in the choice it makes for subdividing the interval at each iteration. It converges faster to the root because it is an algorithm which uses appropriate weighting of the initial end points 1 and 2 using the information about the function, or the data$????_4 = ?????_4 + ??(??)$; $????_4 = ??(??) + ??$ 

Step-21: Set $????_1 = ????_1 + 2??(??)$ If ??2 = 0 Set $????_2 = ??0 = 2??(??)$ Else Set $????_2 = ??0 = 4??(??)$ 

Step-22: Set $????_1 = (??/2) * (??0 + ??) + ?????_1$ Else $????_1 = (??/3) * (??0 + ??) + ?????_1$ 

Step-23: Set $????_3 = (??/8) * (??0 + ??) + ?????_1$ If ??6 = 0 Set $????_4 = ?????_4 + 2??(??)$ Else if ??6 = 0 Set $????_4 = ?????_4 + 6??(??)$ Else Set $????_4 = ?????_4 + ??(??)$ Else Set $????_4 = ???? ?? 4 + 5??(??)$ Else 

17 ??

OUTPUT: $????_1, ?, ???, ?????_2, ??, ???$, and $????_4$ with message which sum is most accurate.

STOP. } 2 Else If C==2 { INPUT: function $f(x)$, initial condition ($?, ??$), interval ?, value of $??$, direct result ??.

Step-1: Set $n= (?) ? ?? 0 )/? , ?? 00 = ?? 0 , ?? 00 = ?? 0 , ?? 02 = ?? 04 = ?? 0 , ?? 1 = ?? 00 + 3?? ??(??) (?? 0 , ?? 00 ) 

Step-2: Set ?? = 1 

Step-3: While ?? ? ??, repeat Step-4 to step-7 

Step-4: Set ?? = 1
Step-5: While ?? ? ?? repeat step-10

Step-6: Set ?? 1 = ?? 0 + ?, ?? 10 = ?? 00 + 1 ?Δ ??"??"(?? 0 , ?? 0 ) + Δ ??"??"(?? 1 , ?? 1 ), ?? 1

Step-7: Set ?? 1?? = ?? 0?? + Δ ??"??"(?? 0 , ?? 0?? ), ?? 11 = ?Δ ??"??"(?? 0 , ?? 04 ), ?? 22 =

?Δ ??"??"(?? 0 + IV.

18 Conclusion

In this paper, we develop an algorithm incorporated with Numerical Integration (Trapezoidal rule, Simpson’s 1/3 rule, Simpson’s 3/8 rule and Weddle’s rule.), Numerical Differentiation (Euler, modified Euler and Runge-Kutta second and fourth order) and finding Roots (Bisection method and False position method) numerically. We observed that the result obtained according to our procedure is completely identical with the hand calculation and save our time and labour. Moreover, Weddle’s rule gives the best solution, the Runge-Kutta fourth order gives the best solution and the False position method gives the best solution in Numerical Integration, Numerical Differentiation and finding Roots numerically respectively.

Figure 1:
Figure 2:

Figure 3:
Else ?? 1 ?? 4 Output: ?? 1?? , ?? 10 , ?? 1 ?? 2 and ?? 1 ?? 4 with message which method gives best solution.

} [1] Else If C==3 { INPUT: function f end points , , b a initial approximations , , 0 tolerance TOL, maximum number of iterations N direct result r .

Step-1: Set
Step-3: Set
Step-5: Set
Step-8: OUTPU (failure)
Step-9: Set
Step-10: While N i ? do steps 11-14
Step-11: Set
Step-13: . 1

Step-14: If


