وزارة التعليم العالي والبحث العلمي

جامعة ديالى

كلية التربية المقداد

قسم الرياضيات

Find an approximate solution using the Simpson method

الى مجلس قسم الرياضيات / كلية التربية المقداد / جامعة ديالى وهو جزء من متطلبات نيل شهادة البكالوريوس – تربية في الرياضيات بحث تقدم به الطالبان

مصطفى فصيل احمد

عبد الله محمد غازي

بأشراف م. م. ايناس حسن عبد

2022

1443

بسم الله الرحمن الرحيم

﴿ فَإِذَا عَزَمْتَ فَتَوَكَّلْ عَلَى اللَّهِ ﴾

صدق الله العظيم

سورة آل عمران اية (159)

((اقرار المشرف))

اشهد بأن اعداد هذا المشروع الموسوم

Find an approximate solution using the Simpson method

والمعد من قبل الطلبة

مصطفى فصيل احمد وعبد الله محمد غازي

قد تم بأشرافي في قسم الرياضيات / كلية التربية المقداد / جامعة ديالى

وهو جزء من متطلبات نيل شهادة البكالوريوس / الرياضيات

التوقيع :

اسم المشرف : م. م . **ايناس حسن عبد** الكاظم

المرتبة العلمية : مدرس مساعد

التاريخ :

Dedication and Acknowledgments

We would like to present our thanks fist to our Almighty Allah, SWT, for giving us power, strength, and patience.

Secondly, Our supervisor, **Assistant teacher. Inas Hasan Abed**, for his guidance and for giving us the golden notes . Without his help, this research would not be completed.

Lastly, many thanks for our friends, relatives and classmates for helping us .

Contents

ABSTRACT

In this research, attention was paid to the trapezoidal or Simpson method in numerical analysis as one of the methods of determination in finding the integration of the function curve, where the pros and cons of the method were discussed, and we found that this method is more accurate and faster and is used in applications of image processors in the field of medicine. In addition, it was solved Some examples to illustrate this method

Chapter one

1.1 Introduction

Numerical analysis and mathematical modelling have become essential in many areas of modern life. Sophisticated numerical analysis software is being embedded in popular software packages, e.g. spreadsheet programs, allowing many people to perform modelling even when they are unaware of the mathematics involved in the process. This requires creating reliable, efficient, and accurate numerical analysis software; and it requires designing problem solving environments (PSE) in which it is relatively easy to model a given situation. The PSE for a given problem area is usually based on excellent theoretical mathematical models, made available to the user through a convenient graphical user interface. Such software tools are well-advanced in some areas, e.g. computer aided design of structures, while other areas are still grappling with the more basic problems of creating accurate mathematical models and accompanying tools for their solution, e.g. atmospheric modelling.

Computer aided design (CAD) and computer aided manufacturing (CAM) are important areas within engineering, and some quite sophisticated PSEs have been developed for CAD/CAM. A wide variety of numerical analysis is involved in the mathematical models that must be solved. The models are based on the basic Newtonian laws of mechanics; there are a variety of possible models, and research continues on designing such models. An important CAD topic is that of modelling the dynamics of moving mechanical systems. The mathematical model involves systems of both ordinary differential equations and algebraic equations (generally nonlinear. The numerical analysis of these mixed systems, called differential-algebraic systems, is quite difficult but important to being able to model moving mechanical systems. Building simulators for cars, planes, and other vehicles requires solving differential-algebraic systems in real-time.

1.2 Numerical algorithms

Numerical algorithms are almost as old as human civilization. The Rhind Papyrus (˜1650 BC) of ancient Egypt describes a root finding method for solving a simple equation;. Archimedes of Syracuse (287- 212 BC) created much new mathematics, including the "method of exhaustion" for calculating lengths, areas, and volumes of geometric figures .

When used as a method to find approximations, it is in much the spirit of modern numerical integration; and it was an important precursor to the development of the calculus by Isaac Newton and Gottfried Leibnitz. A major impetus to developing numerical procedures was the invention of the calculus by Newton and Leibnitz, as this led to accurate mathematical models for physical reality, first in the physical sciences and eventually in the other sciences, engineering, medicine, and business. These mathematical models cannot usually be solved explicitly, and numerical methods to obtain approximate solutions are needed. Another important aspect of the development of numerical methods was the creation of logarithms by Napier (1614) and others, giving a much simpler manner of carrying out the arithmetic operations of multiplication, division, and exponentiation. Newton created a number of numerical methods for solving a variety of problems, and his name is attached today to generalizations of his original ideas. Of special note is his work on root finding and polynomial interpolation. Following Newton, many of the giants of mathematics of the 18th and 19th centuries made major contributions to the numerical solution of mathematical problems. Foremost among these are Leonhard Euler (1707-1783), Joseph-Louis Lagrange (1736-1813), and Karl Friedrich Gauss (1777-1855). Up to the late 1800's, it appears that most mathematicians were quite

Most mathematical modelsused in the natural sciences and engineering are based on ordinary differential equations, partial differential equations, and integral equations. The numerical methods for these equations are primarily of two types. The first type approximates the unknown function in the equation by a simpler function, often a polynomial or piecewise polynomial function, choosing it to satisfy the original equation approximately. Among the best known of such methods is the finite element method for solving partial differential equations; see [8]. The second type of numerical method approximates the derivatives or integrals in the equation of interest, generally solving approximately for the solution function at a discrete set of points. Most initial value problems for ordinary differential equations and partial differential equations are solved in this way, and the numerical procedures are often called finite difference methods, primarily for historical reasons. Most numerical methods for solving differential and integral equations involve both approximation theory and the solution of quite large linear and nonlinear systems. For an introduction to the numerical analysis of differential equations.

Virtually all numerical computation is carried out on digital computers, and their structure and properties affect the structure of numerical algorithms, especially when solving large linear systems. First and foremost, the computer arithmetic must be understood. Historically, computer arithmetic varied greatly between different computer manufacturers, and this was a source of many problems when attempting to write software which could be easily ported between different computers. This has been lessoned significantly with the development of the IEEE (Institute for Electrical and Electronic Engineering) standard for computer floating-point arithmetic. All small computers have adopted this standard, and most larger computer manufacturers have done so as well. For a discussion of the standard and of computer floating-point arithmetic in general,.

For large scale problems, especially in numerical linear algebra, it is important to know how the elements of an array A or a vector x are stored in memory. Knowing this can lead to much faster transfer of numbers from the memory into the arithmetic registers of the computer, thus leading to faster programs. A somewhat related topic is that of pipelining. This is a widely used technique whereby the execution of computer operations are overlapped, leading to faster execution. Machines with the same basic clock speed can have very different program execution times due to differences in pipelining and differences in the way memory is accessed

Most present-day computers are sequential in their operation, but parallel computers are being used ever more widely. Some parallel computers have independent processors that all access the same computer memory (shared memory parallel computers), whereas other parallel computers have separate memory for each processor (distributed memory parallel computers). Another form of parallelism is the use of pipelining of vector arithmetic operations. Some parallel machines are a combination of some or all of these patterns of memory storage and pipelining. With all parallel machines, the form of a numerical algorithm must be changed in order to make best use of the parallelism. For examples of this in numerical linear algebra.

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; these problems occur throughout the natural sciences, social sciences, engineering, medicine, and business. During the past half-century, the growth in power and availability of digital computers has led to an increasing use of realistic mathematical models in science and engineering, and numerical analysis of increasing sophistication has been needed to solve these more detailed mathematical models of the world. The formal academic area of numerical analysis varies from quite theoretical mathematical studies (e.g. see [5]) to computer science issues. With the growth in importance of using computers to carry out numerical procedures in solving mathematical models of the world, an area known as scientific computing or computational science has taken shape during the 1980s and 1990s. This area looks at the use of numerical analysis from a computer science perspective; see [20], [16]. It is concerned with using the most powerful tools of numerical analysis, computer graphics, symbolic mathematical computations, and graphical user interfaces to make it easier for a user to set up, solve, and interpret complicated mathematical models of the real world.

With the growth in importance of using computers to carry out numerical procedures in solving mathematical models of the world, an area known as scientific computing or computational science has taken shape during the 1980s and 1990s. This area looks at the use of numerical analysis from a computer science perspective. It is concerned with using the most powerful tools of numerical analysis, computer graphics, symbolic mathematical computations, and graphical user interfaces to make it easier for a user to set up, solve, and interpret complicated mathematical models of the real world

Numerical analysis is the study of algorithms that use numerical approximation (as opposed to symbolic manipulations) for the problems of mathematical analysis (as distinguished from discrete mathematics). Numerical analysis finds application in all fields of engineering and the physical sciences, and in the 21st century also the life and social sciences, medicine, business and even the arts. Current growth in computing power has enabled the use of more complex numerical analysis, providing detailed and realistic mathematical models in science and engineering. Examples of numerical analysis include: ordinary differential equations as found in celestial mechanics (predicting the motions of planets, stars and galaxies), numerical linear algebra in data analysis and stochastic differential equations and Markov chains for simulating living cells in medicine and biology.

Before modern computers, numerical methods often relied on hand interpolation formulas, using data from large printed tables. Since the mid 20th century, computers calculate the required functions instead, but many of the same formulas continue to be used in software algorithms.

The numerical point of view goes back to the earliest mathematical writings. A tablet from the Yale Babylonian Collection (YBC 7289), gives a sexagesimal numerical approximation of the square root of 2, the length of the diagonal in a unit square.

1.3 Advanced numerical methods

Numerical analysis continues this long tradition: rather than giving exact symbolic answers translated into digits and applicable only to real-world measurements, approximate solutions within specified error bounds are used.

The overall goal of the field of numerical analysis is the design and analysis of techniques to give approximate but accurate solutions to hard problems, the variety of which is suggested by the following:

- Advanced numerical methods are essential in making numerical weather prediction feasible.
- Computing the trajectory of a spacecraft requires the accurate numerical solution of a system of ordinary differential equations.
- Car companies can improve the crash safety of their vehicles by using computer simulations of car crashes. Such simulations essentially consist of solving partial differential equations numerically.
- Hedge funds (private investment funds) use tools from all fields of numerical analysis to attempt to calculate the value of stocks and derivatives more precisely than other market participants.
- Airlines use sophisticated optimization algorithms to decide ticket prices, airplane and crew assignments and fuel needs. Historically, such algorithms were developed within the overlapping field of operations research.

- Insurance companies use numerical programs for actuarial analysis.

The rest of this section outlines several important themes of numerical analysis.

History.

The field of numerical analysis predates the invention of modern computers by many centuries. Linear interpolation was already in use more than 2000 years ago. Many great mathematicians of the past were preoccupied by numerical analysis as is obvious from the names of important algorithms like Newton's method, Lagrange interpolation polynomial, Gaussian elimination, or Euler's method.

To facilitate computations by hand, large books were produced with formulas and tables of data such as interpolation points and function coefficients. Using these tables, often calculated out to 16 decimal places or more for some functions, one could look up values to plug into the formulas given and achieve very good numerical estimates of some functions. The canonical work in the field is the NIST publication edited by Abramowitz and Stegun, a 1000-plus page book of a very large number of commonly used formulas and functions and their values at many points. The function values are no longer very useful when a computer is available, but the large listing of formulas can still be very handy.

The mechanical calculator was also developed as a tool for hand computation. These calculators evolved into electronic computers in the 1940s, and it was then found that these computers were also useful for administrative purposes. But the invention of the computer also influenced the field of numerical analysis,[5] since now longer and more complicated calculations could be done.

1.4 Numerical stability and well-posed problems

Numerical stability is a notion in numerical analysis. An algorithm is called 'numerically stable' if an error, whatever its cause, does not grow to be much larger during the calculation. This happens if the problem is 'well-conditioned', meaning that the solution changes by only a small amount if the problem data are changed by a small amount. To the contrary, if a problem is 'ill-conditioned', then any small error in the data will grow to be a large error.

Both the original problem and the algorithm used to solve that problem can be 'well-conditioned' or 'ill-conditioned', and any combination is possible.

Much effort has been put in the development of methods for solving systems of linear equations. Standard direct methods, i.e., methods that use some matrix decomposition are Gaussian elimination, LU decomposition, Cholesky decomposition for symmetric (or hermitian) and positive-definite matrix, and QR decomposition for non-square matrices. Iterative methods such as the Jacobi method, Gauss–Seidel method, successive over-relaxation and conjugate gradient method[12] are usually preferred for large systems. General iterative methods can be developed using a matrix splitting.

Root-finding algorithms are used to solve nonlinear equations (they are so named since a root of a function is an argument for which the function yields zero). If the function is differentiable and the derivative is known, then Newton's method is a popular choice. Linearization is another technique for solving nonlinear equations.

Numerical integration, in some instances also known as numerical quadrature, asks for the value of a definite integral. Popular methods use one of the Newton–Cotes formulas (like the midpoint rule or Simpson's rule) or Gaussian quadrature. These methods rely on a "divide and conquer" strategy, whereby an integral on a relatively large set is broken down into integrals on smaller sets. In higher dimensions, where these methods become prohibitively expensive in terms of computational effort, one may use Monte Carlo or quasi-Monte Carlo methods (see Monte Carlo integration), or, in modestly large dimensions, the method of sparse grids.

Since the late twentieth century, most algorithms are implemented in a variety of programming languages. The Netlib repository contains various collections of software routines for numerical problems, mostly in Fortran and C. Commercial products implementing many different numerical algorithms include the IMSL and NAG libraries; a free-software alternative is the GNU Scientific Library.

Over the years the Royal Statistical Society published numerous algorithms in its *Applied Statistics* (code for these "AS" functions is here); ACM similarly, in its *Transactions on Mathematical Software* ("TOMS" code is here). The Naval Surface Warfare Center several times published its *Library of Mathematics Subroutines* .

There are several popular numerical computing applications such as MATLAB, TK Solver, S-PLUS, and IDL as well as free and open source alternatives such as FreeMat, Scilab, GNU Octave (similar to Matlab), and IT++ (a C++ library). There are also programming languages such as R (similar to S-PLUS), Julia, and Python with libraries such as NumPy, SciPy and SymPy. Performance varies widely: while vector and matrix operations are usually fast, scalar loops may vary in speed by more than an order of magnitude.

Many computer algebra systems such as Mathematica also benefit from the availability of arbitrary-precision arithmetic which can provide more accurate results.

Also, any spreadsheet software can be used to solve simple problems relating to numerical analysis. Excel, for example, has hundreds of available functions, including for matrices, which may be used in conjunction with its built in "solver".

Over the years, we have been taught on how to solve equations using various algebraic methods. These methods include the substitution method and the elimination method. Other algebraic methods that can be executed include the quadratic formula and factorization. In Linear Algebra, we learned that solving systems of linear equations can be implemented by using row reduction as an algorithm. However, when these methods are not successful, we use the concept of numerical methods. Numerical methods are used to approximate solutions of equations when exact solutions cannot be determined via algebraic methods. They construct successive approximations that converge to the exact solution of an equation or system of equations. In Math 3351, we focused on solving nonlinear equations involving only a single variable. We used methods such as Newton’s method, the Secant method, and the Bisection method. We also examined numerical methods such as the Runge-Kutta methods, that are used to solve initial-value problems for ordinary differential equations. However these problems only focused on solving nonlinear equations with only one variable, rather than nonlinear equations with several variables. The goal of this paper is to examine three different numerical methods that are used to solve systems of nonlinear equations in several variables. The first method we will look at is Newton’s method. This will be followed by Broyden’s method, which is sometimes called a Quasi-Newton method; it is derived from Newton’s method. Lastly, we will study the Finite Difference method that is used to solve boundary value problems of nonlinear ordinary differential equations. For each method, a breakdown of each numerical procedure will be provided. In addition, there will be some discussion of the convergence of the numerical methods, as well as the advantages and disadvantages of each method. After a discussion of each of the three methods, we will use the computer program Matlab to solve an example of a nonlinear ordinary differential equation using both the Finite Diffference method and Newton’s method .

Chapter Two

2.1 Simpson’s Rule

2.2 Recommended articles

2.1 Simpson's Rule

With the midpoint rule, we estimated areas of regions under curves by using rectangles. In a sense, we approximated the curve with piecewise constant functions. With the trapezoidal rule, we approximated the curve by using piecewise linear functions. What if we were, instead, to approximate a curve using piecewise quadratic functions? With Simpson's rule, we do just this. We partition the interval into an even number of subintervals, each of equal width. Over the first pair of subintervals we approximate.

$\int_{x_0}^{x^2} f(x)dx \; with \; \int_{x_0}^{x^2} p(x)dx, where\; p(x) = Ax^2 + Bx + C$ is the quadratic function passing through $\left(x_0 , f(x_0)\right) , \left(x_1, f(x_1)\right) , and \left(x_2 , f\ (x_2)\right)$ ($Figure$ 2.5.4). Over the next pair of subintervals we approximate. $\int_{x_2}^{x_4} f(x)dx$ with the integral of another quadratic function passing through $\left(x_2 , f\ (x_2)\right) \; , \left(x_3 , f\ (x_3)\right) , and \left(x_4 , f\ (x_4)\right).$ This process is continued with each successive pair of subintervals.

Figure 2.5.4 : With Simpson's rule, we approximate a definite integral by integrating a piecewise quadratic function.

To understand the formula that we obtain for Simpson's rule, we begin by deriving a formula for this approximation over the first two subintervals. As we go through the derivation, we need to keep in mind the following relationships:

$$f\ (x_0) = p(x_0) = Ax_0^2 + Bx_0 + C \quad (2.5.6)$$

$$f\ (x_1) = p(x_1) = Ax_1^2 + Bx_1 + C \quad (2.5.7)$$

$$f\ (x_2) = p(x_2) = Ax_2^2 + Bx_2 + C \quad (2.5.8)$$

$x_2 - x_0 = 2\Delta\, x \; , where\; \Delta\, x\; is\; the\; length\; of\; a\; , subinterval.$

$x_2 + x_0 = 2\Delta\, x_1\ ,\ since\ x_1 = \frac{(\,x_2 + x_0)}{2}$

Thus ,

$$\int_{x_0}^{x_2} f(x)dx \ \approx\ \int_{x_0}^{x_2} p(x)dx$$

$$= \int_{x_0}^{x_2} (\,Ax^2 + Bx + C)\,dx$$

$$\left(\frac{A}{3}X^3 + \frac{B}{2}X^2 + Cx\right)\Big|_{x_0}^{x_2}$$

Find the antiderivative.

$= \frac{A}{3}(\,x_2^3 -\ x_0^3) +\ \frac{B}{2}(\,x_2^2 -$
$x_0^2) + C(\,x_2 -\ x_0)$ Evaluate the antiderivative.

$$= \frac{A}{3}\ (\,x_2 -\ x_0)\ (x_2^2 +\ x_2x_0 + x_0^2) + \frac{B}{2}\ (\,x_2 -\ x_0)(\,x_2 +\ x_0) + C\ (\,x_2 -\ x_0)$$

$$= \frac{x_2 - x_0}{6}(2A(X_2^2\ x_2x_0 + X_0^2\ ) + 3B\ (\ x_2 +\ x_0)6C\ )$$

Factor out $\frac{x_2 - x_0}{6}$

$= \frac{\Delta X}{3}(Ax_2^2 + BX_2 + C) + (Ax_0^2 + BX_0 + C)A(x_2^2 + 2x_2x_0 +$
$x_0^2) + 2B(x_2 +\ x_0) + 4C\ )$ Rearrange the terms. Note: $\Delta x = \frac{x_2 - x_0}{2}$

$= \frac{\Delta X}{3}(F(X_2) + F(X_0) + A(X_2 + X_0)^2 + 2B(\ x_2 +\ x_0) + 4C)$
Factor and substitute:$f(x_2) = Ax_2^2 + BX_2 + C\ and\ f(x_0) = Ax_0^2 + Bx_0 + C.$

$= \frac{\Delta X}{3}(F(X_2) + F(X_0) + A(2X_1)^2 + 2B(2X_1) + 4C\ )$ Substitute $x_2 +\ x_0 = 2X_1$. Note : $X_1 = \frac{x_2 + x_0}{2}\ ,the\ midpoint.$

$= \frac{\Delta X}{3}\left(F(X_2) + 4F(X_1 + F(X_0)\right)$ Expand and , substitute $f(x_1) = AX_1^2 + BX_1 + C$.

If we approximate $\int_{X_2}^{X_4} F(X)dX$ using the same method, we see that we have

$$\int_{X_0}^{X_4} F(X)dX \approx \frac{\Delta X}{3}\left(f(x_4) + 4f(x_3) + f(x_2)\right).$$

Combining these two approximations, we get

$$\int_{X_0}^{X_4} F(X)dX$$
$$= \frac{\Delta X}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3)f(x_4)\right).$$

The pattern continues as we add pairs of subintervals to our approximation. The general rule may be stated as follows.

Simpson’s Rule

Assume that f(x) is continuous over [a,b]. Let n be a positive even integer and $\Delta x =$ $\frac{b-a}{2}$ $\text{Let } [a,b] \text{ be divided into n subintervals, each of length } \Delta x\text{, with en}$ $\{x0, x1, x2, \ldots, xn\}. P = \{x_0, x_1, x_2, \ldots, x_n\}.$ Set.

$\text{Sn} = \frac{\Delta x}{3}(f(x0) + 4f(x1) + 2f(x2) + 4f(x3) + 2f(x4) + \cdots + 2f(xn - 2) + 4f(xn - 1) + f(xn)). (2.5.9)$

Then ,

$$\lim_{n \to +\infty} \text{Sn} = \int_a^b f(x)dx.$$

Just as the trapezoidal rule is the average of the left-hand and right-hand rules for estimating definite integrals, Simpson's rule may be obtained from the midpoint and trapezoidal rules by using a weighted average. It can be shown that S2n= $\left(\frac{2}{3}\right) M_n + \left(\frac{1}{3}\right) T_n$

It is also possible to put a bound on the error when using Simpson's rule to approximate a definite integral. The bound in the error is given by the following rule:

Rule: Error Bound for Simpson's Rule

Let f(x)be a continuous function over [a, b] having a fourth derivative, $f^{(4)}$

$\left|f^{(4)}(x)\right|$ over [a , b] , then the upper bound for the error in using Sn to estimate

$$\int_a^b f(x)dx \text{ is given by}$$

$Error\ in\ S_n \le \frac{M(b-a)^5}{180n^4}$

- **Trapezoidal rule**

Tn=$\frac{\Delta x}{2}$ (f(x0)+2f(x1)+2f(x2)+⋯+2f(xn−1)+f(xn))

- **Simpson's rule**

$$\text{Sn} = \frac{\Delta x}{3}(\text{f(x0)} + 4\text{f(x1)} + 2\text{f(x2)} + 4\text{f(x3)} + 2\text{f(x4)} + 4\text{f(x5)} + \cdots + 2\text{f(xn} - 2) + 4\text{f(xn} - 1) + \text{f(xn))}$$

- **Error bound for midpoint rule**

Error in $T_n \le \frac{M(b-a)^3}{24n^2}$

- **Error bound for trapezoidal rule**

Error in $T_n \le \frac{M(b-a)^3}{12n^2}$

- **Error bound for Simpson’s rule**

Error in $S_n \le \frac{M(b-a)^5}{180n^4}$

EX\\ Find the value of the integral using simpson's ruie when n=6

$$\int_0^1 e^{\sqrt{x}}\,dx$$

$$h = \frac{b-a}{n} = \frac{1-0}{6} = \frac{1}{6}$$

$$\int_0^1 e^{\sqrt{x}}\,dx = \frac{\frac{1}{6}}{3}\Big\{ f(0) + 4f\left(0+\frac{1}{6}\right) + 2f\left(0+\frac{2}{6}\right) + 4f\left(0+\frac{3}{6}\right) + 2f\left(0+\frac{4}{6}\right) + 4f\left(0+\frac{5}{6}\right) + f(1)\Big\}$$

$$\int_0^1 e^{\sqrt{x}}\,dx \cong \frac{1}{18}\left\{ e^{\sqrt{0}} + 4e^{\sqrt{\frac{1}{6}}} + 2e^{\sqrt{\frac{1}{3}}} + 4e^{\sqrt{\frac{1}{2}}} + 2e^{\sqrt{\frac{2}{3}}} + 4e^{\sqrt{\frac{5}{6}}} + e^{\sqrt{1}} \right\}$$

$$= \frac{1}{18}(1 + 4 + 1.5042 + 2 \times 1.7813 + 4 \times 2.0281 + 2 \times 2.2626 +$$

$$4 \times 2.4915 + 2.7183 = 1.9945$$

EX\\ Find the value of the integral using simpson's ruie when n=6

1- $\int_0^1 e^{x^2}\,dx$ 2- $\int_1^2 \frac{e^x}{x}\,dx$

1- $\int_0^1 e^{x^2}\,dx$

$h = \frac{b-a}{n} = \frac{1-0}{6} = \frac{1}{6}$

$$\int_0^1 e^{x^2}\,dx \cong \frac{\frac{1}{6}}{3}\left[f(0) + 4f\left(0+\frac{1}{6}\right) + 2f\left(0+\frac{2}{6}\right) + 4f\left(0+\frac{3}{6}\right) + 2f\left(0+\frac{4}{6}\right) + 4f\left(0+\frac{5}{6}\right) + f(1)\right]$$

$\int_0^1 e^{x^2}\,dx \cong \frac{1}{18}\{e^0 + 4e^{(0.1667)^2} + 2e^{(0.3333)^2} + 4e^{(0.5)^2} + 2e^{(0.6667)^2} + 2e^{(0.6667)^2} + 4e^{(0.1667)^2} + e^1$

$= \frac{1}{18}\{\, 1 + 4 * 1.0282 + 2 * 1.1175 + 4 * 1.2840 + 2 * 1.5597 + 4 * 2.0025 + 2.7183$

$$\frac{1}{8}\{1 + 4.1128 + 2.235 + 5.136 + 3.1194 + 8.01 + 2.718$$

$$= 1.4629$$

2- $\int_1^2 \frac{e^x}{x}\,dx$

$$h = \frac{2-1}{6} = \frac{1}{6}$$

$$\int_1^2 \frac{e^x}{x}\,dx$$

$$\cong \frac{\frac{1}{6}}{3}\left[f(1) + 4f\left(1+\frac{1}{6}\right) + 2f\left(1+\frac{2}{6}\right) + 4f\left(1+\frac{3}{6}\right) + 2f\left(1+\frac{4}{6}\right) + 4f\left(1+\frac{5}{6}\right) + f(2)\right]$$

$$\int_1^2 \frac{e^x}{x}\,dx \cong \frac{1}{18}\left[\frac{e^1}{1} + 4 * \frac{e^{1.1667}}{1.1667} + 2 * \frac{e^{1.8333}}{1.8333} + \frac{e^2}{2}\right]$$

$$\frac{1}{18}[2.7183 + 4 * 2.7525 + 2 * 2.8452 + 4 * 2.9878 + 2 * 3.1767 + 4 * 3.4116 + 3.6945]$$

$$\frac{1}{18}[2.7183 + 11.01 + 5.6904 + 11.9512 + 6.3534 + 13.6464 + 3.6945 = 3.0591]$$

2.2 Recommended articles

1. 5.3: Riemann SumsA fundamental calculus technique is to first answer a given problem with an approximation, then refine that approximation to make it better, then use ...
2. 1.11: Numerical IntegrationIn this section we turn to the problem of how to find (approximate) numerical values for integrals, without having to evaluate them algebraically. To ...
3. 5.5: Numerical IntegrationThe Fundamental Theorem of Calculus gives a concrete technique for finding the exact value of a definite integral. That technique is based on computin...
4. Simpson's RuleThe Trapezoidal and Midpoint estimates provided better accuracy than the Left and Right endpoint estimates. It turns out that a certain combination of...
5. 7.6: Numerical IntegrationThe antiderivatives of many functions either cannot be expressed or cannot be expressed easily in closed form (that is, in terms of known functions). ...

الخاتمة

لقد وصلنا لنهاية هذا البحث ، وفي النهاية لا يسعني سوى ان اشكركم على حسن متابعتكم لهذا البحث ، وانا قد عرضت بهذا البحث رأيي المتواضع ببركة الله تعالى وكرمه وتوفقيه ، وقد اكرمني الله بأن ادلو بدلوي تجاه هذا الموضوع :

(Find an approximate solution using the Simpson method)

ولعل الله تعالى قد وفقني بهذا البحث وفي هذا الموضوع ، ولعل قلمي وفق في تقديم ما يدور بخلدي ، وفي نهاية الامر فأنني بشر اصيب واخطأ ، واني اتوجه الى الله بالدعاء على توفيقي في تقديم هذا البحث وعلى حسن قراءتكم ومتابعتكم لهذا البحث ، ونشكر لكم سعة صدركم ونرجوا ان ينال البحث اعجابكم ، والحمد لله الذي هداني الى هذا .

REFERNCES

1- R. Aiken (editor). Stiff Computation, Oxford University Press, Oxford 1985 .

2- U. Ascher and R. Russell, eds. Numerical Boundary Value ODEs, Birkha user, Boston, MA, 1985.

3- U. Ascher, R. Mattheij, and R. Russell. Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Prentice-Hall، Englewood Cliffs, New Jersey, 1988.

4- K. Atkinson. An Introduction to Numerical Analysis, 2nd ed., John Wiley، New York, 1989.

5- K. Atkinson and W. Han. Elementary Numerical Analysis, 3rd ed., John Wiley, New York, 2004.

6- A. Aziz. Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations, Academic Press, New York, 1975.

7- L. Shampine, I. Gladwell, and S. Thompson. Solving ODEs with MATLAB، Cambridge University Press, 2003.

8- J.C. Butcher. "General linear methods", Acta Numerical , Cambridge University Press,2006.

9- C.W.Gear. NumericalInitial aValue Problemsin Ordinary Differential Equations، Prentice Hall, Englewood Cliffs, NJ, 1971.

10- E. Isaacson and H. Keller. Analysis of Numerical Methods, John Wiley, New York, 1966