وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كلية التربية المقداد
قسم الرياضيات

إيجاد الحل التقريبي بأستخدام طريقة شبه المنحرف

الى مجلس قسم الرياضيات – كلية التربية المقداد – جامعة ديالى
وهو جزء من متطلبات نيل شهادة البكالوريوس
بحث تتقدم به الطالبتان

اسيل حسين جبار
نور نوري عبد الوهاب

بأشراف
م.م ايناس حسن

2022

1443

(اقرار المشرف)

اشهد بأن اعداد هذ المشروع الموسوم

إيجاد الحل التقريبي بأستخدام طريقة شبه المنحرف

والمعد من قبل الطلبة

اسيل حسين جبار - نور نوري عبد الوهاب

قد تم بأشرافي في قسم الرياضيات / كلية التربية المقداد / جامعة ديالي وهي جزء من متطلبات نيل شهادة البكالوريوس / الرياضيات

التوقع :

اسم المشـرف : م.م ايناس حسن

المرتبة العلمية : مدرس

التاريـــــــخ :

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

((يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۚ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ))

صدق الله العظيم

المجادلة اية (11)

الاهداء

مرَّت قاطرة البحث بكثير من العوائق، ومع ذلك حاولت أن أتخطَّاها بثبات بفضل من الله ومنِّه

إلى صاحب السيرة العطرة، والفكر المُستنير؛

ومن كان له الفضل الأوَّل في بلوغي التعليم العالي

(والدي الحبيب)، أطال الله في عُمره

وإلى من وضعتني على طريق الحياة، وجعلتني رابط الجأش،

وراعتني حتى صرت كبيرًا

(أمي الغالية)، طيَّب الله ثراها

والى مشرفتنا بالبحث الاستاذة (ايناس حسن)

إلى جميع أساتذتي الكرام؛ ممن لم يتوانوا في مد يد العون لي

أُهدي إليكم بحثي...

الشكر وتقدير

وانطلاقًا من مبدأ أنه لا يشكر الله من لا يشكر الناس، فإننا نتوجه بالشكر الجزيل للأستاذة (ايناس حسن) الذي رافقنا في مسيرتنا لإنجاز هذا البحث وكانت لها بصمات واضحة من خلال توجيهاتها وانتقاداتها البناءة والدعم الأكاديمي،

كما نشكر عائلاتنا التي

صبرت وتحملت معنا ورفدتنا بالكثير من الدعم على جميع الأصعدة، ونشكر الأصدقاء والأحباب وكل من قدم لما الدعم المادي أو المعنوي

(ملخص البحث)

في هذا البحث ثم الاهتمام بطريقه (شبه المنحرف او (سمبون)) في التحليل العددي با اعتبارها طريقة من طرق التقريب في ايجاد الله تكامل منحني الدالة حيث ثم مناقشه ايجابيات وسلبيات الطريقة وقد وجدنا ان هذه الطريقة تكون إدق واسرع وتستخدم في تطبيقات المعالجات الصورية في مجال الطب ،

بالاضافة الى تم حل بعض الأمثلة لتوضيح هذه الطريقة .

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. Building simulators for cars, planes, and other vehicles requires solving differential-algebraic systems in real-time.

((Chapter one))

1.1 Numerical algorithms

1.2 Advanced numerical methods

1.3 Numerical stability and well-posed problems

Chapter one

1.1 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.

1.2 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.

1.3 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))

Trapezoidal Rule

2.1 Trapezoidal Rule

2.2 Absolute and Relative Error

2.3 Numerical implementation

2.4 Error analysis

2.5 Trapezoidal Rule Formula

2.6 Periodic and peak functions

2.7 Trapezoid Laws

2.1 Trapezoidal Rule

We can also approximate the value of a definite integral by using trapezoids rather than rectangles. In Figure 2.1 , the area beneath the curve is approximated by trapezoids rather than by rectangles.

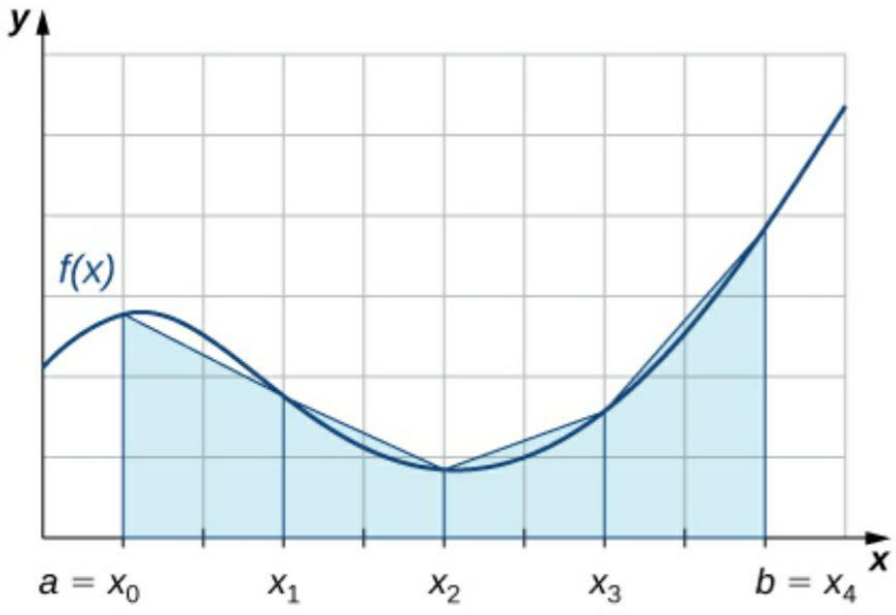


Figure: 2.1 Trapezoids may be used to approximate the area under a curve, hence approximating the definite integral.

for estimating definite integrals uses trapezoids rather than rectangles to approximate the area under a curve. To gain insight into the final form of the rule, consider the trapezoids shown in Figure 1 . We assume that the length of each subinterval is given by Δ_x. First, recall that the area of a trapezoid with a height of h and bases of length b1 and b2 is given by

Area=1/2h(b1+b2)

We see that the first trapezoid has a height Δ_x and parallel bases of length f(x0) and f(x_1). Thus, the area of the first trapezoid in

Figure 2.1 is

$$\frac{1}{2}\Delta x\big(f(x_0)+f(x_1)\big)$$

The areas of the remaining three trapezoids are

$$\frac{1}{2}\Delta x\big(f(x_1)+f(x_2)\big),\ \frac{1}{2}\Delta x\big(f(x_2)+f(x_3)\big),\ \text{and}$$
$$\frac{1}{2}\Delta x\big(f(x_3)+f(x_4)\big).$$

After taking out a common factor of 1/2Δ_x and combining like terms,

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\Delta x\big(f(x_0)+f(x_1)\big)+\frac{1}{2}\Delta x\big(f(x_1)+\ f(x_2)\big)+\frac{1}{2}\Delta x\big(f(x_2)+f(x_3)\big)+\frac{1}{2}\Delta x\big(f(x_3)+f(x_4)\big)$$

we have

$$\int_a^b f(x)\,dx \approx \frac{\Delta x}{2}\big(f(x_0)+2\,f(x_1)+2\,f(x_2)+2\,f(x_3)+f(x_4)\big)$$

Generalizing, we formally state the following rule.

The Trapezoidal Rule

Assume that f(x) is continuous over [a,b]. Let n be a positive integer and $\Delta_x = b-a\ /n$.

Let [a,b] be divided into nn subintervals, each of length Δx, with endpoints at P={x0,x1,x2…,xn}.

Set

$$T_n = \frac{\Delta x}{2}\left(f(x_0) + 2\,f(x_1) + 2\,f(x_2) + \cdots + 2\,f(x_{n-1}) + f(x_n)\right)$$

$$\text{Then, } \lim_{n\to+\infty} T_n = \int_a^b f(x)\,dx$$

Before continuing, let's make a few observations about the trapezoidal rule. First of all, it is useful to note that

$$T_n = \frac{1}{2}(L_n + R_n) \text{ where } L_n = \sum_{i=1}^{n} f(x_{i-1})\Delta x \text{ and}$$

$$R_n = \sum_{i=1}^{n} f(x_i)\Delta x$$

That is, L_n and R_n approximate the integral using the left-hand and right-hand endpoints of each subinterval, respectively. In addition, a careful examination of Figure 2 leads us to make the following observations about using the trapezoidal rules and midpoint rules to estimate the definite integral of a nonnegative function. The trapezoidal rule tends to overestimate the value of a definite integral systematically over intervals where the function is concave up and to underestimate the value of a definite integral systematically over intervals where the function is concave down. On the other hand, the midpoint rule tends to average out these errors somewhat by partially overestimating and partially underestimating the value of the definite integral over these same types of intervals. This leads us to hypothesize that, in general, the midpoint rule tends to be more accurate than the trapezoidal rule.

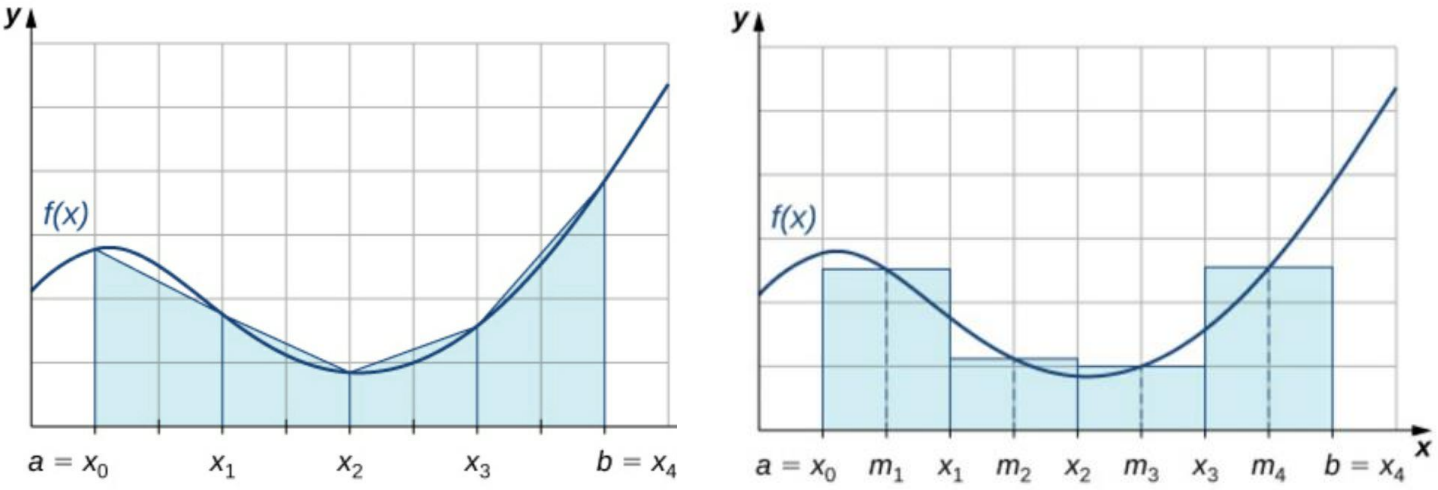
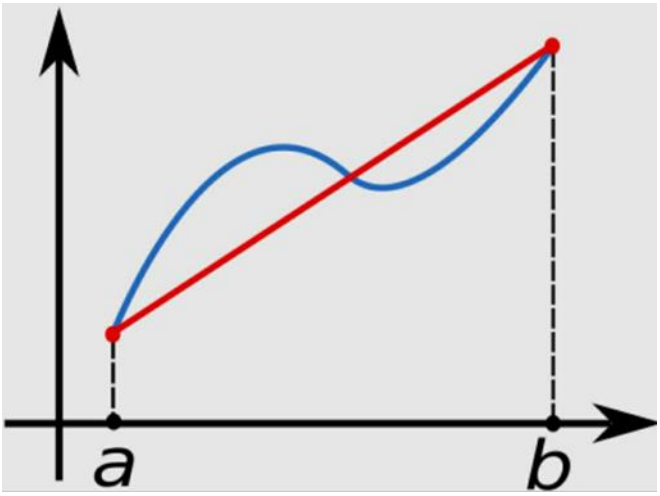


Figure 2 :The trapezoidal rule tends to be less accurate than the midpoint rule.

In mathematics, and more specifically in numerical analysis, the trapezoidal rule (also known as the trapezoid rule or trapezium rule; see Trapezoid for more information on terminology) is a technique for approximating the definite integral.



The function f(x) (in blue) is approximated by a linear function (in red).

$$\int_a^b f(x)\,dx$$

The trapezoidal rule works by approximating the region under the graph of the function f(x) as a trapezoid and calculating its area.

It follows that

$$\int_a^b f(x)\,dx \approx (b-a)\cdot\tfrac{1}{2}(f(a)+f(b))$$

The trapezoidal rule may be viewed as the result obtained by averaging the left and right Riemann sums, and is sometimes defined this way. The integral can be even better approximated by partitioning the integration interval, applying the trapezoidal rule to each subinterval, and summing the results. In practice, this "chained" (or "composite") trapezoidal rule is usually what is meant by "integrating with the trapezoidal rule". Let $\{x_k\}$ be a partition of [a,b] such that

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$$

be the length of the k-th subinterval that is,

$$\Delta x_k = x_k - x_{k-1}$$

then

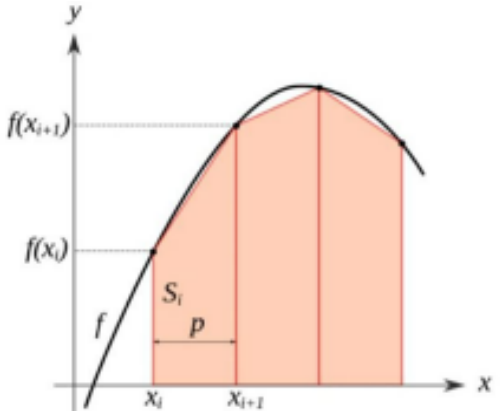
$$\int_a^b f(x)\,dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k$$

When the partition has a regular spacing, as is often the case, that is, when all the Δx_k have the same value Δ_x, the formula can be simplified for calculation efficiency by factoring Δ_x out

$$\int_a^b f(x)\,dx \approx \frac{\Delta x}{2} \left(f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + \cdots + 2f(x_{N-1}) + f(x_N)\right)$$

The approximation becomes more accurate as the resolution of the partition increases (that is, for larger N , all Δx_k decrease).

As discussed below, it is also possible to place error bounds on the accuracy of the value of a definite integral estimated using a trapezoidal rule.



Example 2.1.1 : Using the Trapezoidal Rule

Use the trapezoidal rule to estimate $\int_0^1 x^2\,dx$ using four subintervals.

Solution:

The endpoints of the subintervals consist of elements of the set P={0,14,12,34,1} and Δ_x=1−04=14. Thus,

$$\int_0^1 x^2 dx \approx \frac{1}{2}\cdot\frac{1}{4}\left(f(0)+2f\left(\tfrac{1}{4}\right)+2f\left(\tfrac{1}{2}\right)+2f\left(\tfrac{3}{4}\right)+f(1)\right)$$

$$= \tfrac{1}{8}\left(0+2\cdot\tfrac{1}{16}+2\cdot\tfrac{1}{4}+2\cdot\tfrac{9}{16}+1\right)$$

$$= \frac{11}{32}$$

2.2 Absolute and Relative Error

An important aspect of using these numerical approximation rules consists of calculating the error in using them for estimating the value of a definite integral. We first need to define absolute error and relative error.

Definition: absolute and relative error

If BB is our estimate of some quantity having an actual value of A , then the absolute error is given by |A−B|.

The relative error is the error as a percentage of the actual value and is given by

$$\left|\frac{A-B}{A}\right| \cdot 100\%.$$

Example 2.2.1 : Calculating Error in the Midpoint Rule

Calculate the absolute and relative error in the estimate of using the $\int_0^1 X^2\,dx$ midpoint rule, found in Example 2 .

Solution:

The calculated value is $\int_0^1 X^2\,dx = 13$

and our estimate from the example is M_4=21/64. Thus, the absolute error is given by |1/3 − 21/64| = 1/192 ≈ 0.0052.

The relative error is

$$\frac{1/96}{1/3} = 0.03125 \approx 3.1\%$$

Example 2.2.2 : Calculating Error in the Trapezoidal Rule

Calculate the absolute and relative error in the estimate $\int_0^1 X^2\ dx$ of using the trapezoidal rule .

Solution: The calculated value is $\int_0^1 x2\ \ dx$ and our estimate from the example is M_4=21/64. Thus, the absolute error is given by |1/3−21/64 | = 11/92 ≈ 0.0052.

The relative error is

$$\frac{1/192}{1/3} = \frac{1}{64} \approx 0.015625 \approx 1.6\%$$

2.3 Numerical implementation

2.3.1 Non-uniform grid

When the grid spacing is non-uniform, one can use the formula

$$\int_a^b f(x)\,dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k$$

2.3.2 Uniform grid

For a domain discretized into N equally spaced panels, considerable simplification may occur. Let

$$\Delta x_k = \Delta x = \frac{b-a}{N}$$

the approximation to the integral becomes

$$\begin{aligned}\int_a^b f(x)\,dx &\approx \frac{\Delta x}{2}\sum_{k=1}^{N}\left(f(x_{k-1})+f(x_k)\right)\\ &= \frac{\Delta x}{2}\left(f(x_0)+2f(x_1)+2f(x_2)+2f(x_3)+\cdots+2f(x_{N-1})+f(x_N)\right)\\ &= \Delta x\left(\sum_{k=1}^{N-1} f(x_k)+\frac{f(x_N)+f(x_0)}{2}\right).\end{aligned}$$

2.4 Error analysis

The error of the composite trapezoidal rule is the difference between the value of the integral and the numerical result:

$$\mathrm{E} = \int_a^b f(x)\,dx - \frac{b-a}{N}\left[\frac{f(a)+f(b)}{2}+\sum_{k=1}^{N-1} f\left(a+k\frac{b-a}{N}\right)\right]$$

There exists a number ξ between a and b, such that

$$\mathrm{E} = -\frac{(b-a)^3}{12N^2} f''(\xi)$$

It follows that if the integrand is concave up (and thus has a positive second derivative), then the error is negative and the trapezoidal rule overestimates the true value. This can also be seen from the geometric picture: the trapezoids include all of the area under the curve and extend over it. Similarly, a concave-down function yields an underestimate because area is unaccounted for under the curve, but none is counted above. If the interval of the integral being approximated includes an inflection point, the error is harder to identify.

An asymptotic error estimate for N → ∞ is given by

$$\mathrm{E} = -\frac{(b-a)^2}{12N^2}\left[f'(b) - f'(a)\right] + O(N^{-3})$$

Further terms in this error estimate are given by the Euler–Maclaurin summation formula.

2.5 Trapezoidal Rule Formula

We apply the trapezoidal rule formula to solve a definite integral by calculating the area under a curve by dividing the total area into little trapezoids rather than rectangles. This rule is used for approximating the definite integrals where it uses the linear approximations of the functions. The trapezoidal rule takes the average of the left and the right sum.

Let y = f(x) be continuous on [a, b]. We divide the interval [a, b] into n equal subintervals, each of width, h = (b - a)/n,

such that $a = x_0 < x_1 < x_2 < \cdots < x_n = b$

Area $= \frac{h}{2}[y_0 + 2(y_1 + y_2 + y_3 + \ldots . + y_{n-1}) + y_n]$

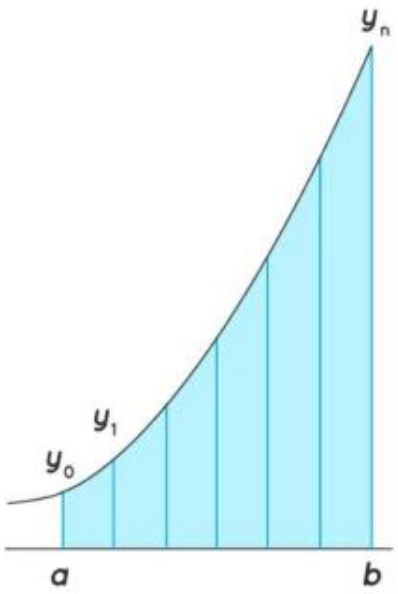
where,

- $y_0, y_1, y_2 \ldots .$ are the values of function at x = 1, 2, 3….. respectively.

Trapezoidal Rule Formula

$$\text{Area} = \int_a^b y\,dx \approx \frac{1}{2}h[y_0 + 2(y_1 + y_2 + \ldots + y_{n-1}) + y_n]$$

$$\text{where } h = \frac{b - a}{n}$$



Example 2.1.2 :

Use the Trapezoidal Rule with n=6 to approximate

$$\int_0^{\pi} \sin^2 x dx.$$

Solution

Here

$$f(x) = \sin^2 x, \quad a = 0, \quad b = \pi.$$

The width of each subinterval is

$$\Delta x = \frac{b-a}{n} = \frac{\pi}{6},$$

so the grid points have the coordinates

$$x_i = \frac{i\pi}{6}$$

Calculat the values of the function f(x) at the points x_i

$$f(x_0) = f(0) = \sin^2 0 = 0^2 = 0;$$
$$f(x_1) = f\left(\frac{\pi}{6}\right) = \sin^2 \frac{\pi}{6} = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$$
$$f(x_2) = f\left(\frac{2\pi}{6}\right) = \sin^2 \frac{\pi}{3} = \left(\frac{\sqrt{3}}{2}\right)^2 = \frac{3}{4}$$
$$f(x_3) = f\left(\frac{3\pi}{6}\right) = \sin^2 \frac{\pi}{2} = 1^2 = 1$$
$$f(x_4) = f\left(\frac{4\pi}{6}\right) = \sin^2 \frac{2\pi}{3} = \left(\frac{\sqrt{3}}{2}\right)^2 = \frac{3}{4}$$
$$f(x_5) = f\left(\frac{5\pi}{6}\right) = \sin^2 \frac{5\pi}{6} = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$$
$$f(x_6) = f(\pi) = \sin^2 \pi = 0^2 = 0.$$

The Trapezoidal Rule formula is written in the form

$$\int_0^{\pi} \sin^2 x dx \approx T_6 = \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + \quad \cdots + 2f(x_5) + f(x_6)] = \frac{\pi}{12}\left[0 + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{4} + 0\right] =$$

$$\frac{\pi}{12}\left[\frac{1}{2} + \frac{3}{2} + 2 + \frac{3}{2} + \frac{1}{2}\right] = \frac{\pi}{12} \cdot \frac{12}{2} = \frac{\pi}{2}$$

We can also determine the exact value of the integral:

$$\int_0^{\pi} \sin^2 x dx = \frac{1}{2}\int_0^{\pi} (1 - \cos 2x) dx = \frac{1}{2}\left[x - \frac{\sin 2x}{2}\right]_0^{\pi} = \frac{1}{2}[(\pi - 0) - 0] = \frac{\pi}{2}$$

So, in this particular example, the trapezoidal approximation T_6 coincides with the exact value of the integral.

2.6 Periodic and peak functions

The trapezoidal rule converges rapidly for periodic functions. This is an easy consequence of the Euler-Maclaurin summation formula, which says that if f is p times continuously differentiable with period T

$$\sum_{k=0}^{N-1} f(kh)h = \int_0^T f(x)\,dx + \sum_{k=1}^{\lfloor p/2 \rfloor} \frac{B_{2k}}{(2k)!}(f^{(2k-1)}(T) - f^{(2k-1)}(0)) - (-1)^p h^p \int_0^T \tilde{B}_p(x/T) f^{(p)}(x)\,\mathrm{d}x$$

where h :=T/N and $\dot{B}_p$ is the periodic extension of theplaystyle p th Bernoulli polynomial.Due to the periodicity, the derivatives at the endpoint cancel and we see that the error is O(h^p) .

A similar effect is available for peak-like functions, such as Gaussian, Exponentially modified Gaussian and other functions with derivatives at integration limits that can be neglected. The evaluation of the full integral of a Gaussian function by trapezoidal rule with 1% accuracy can be made using just 4 points. Simpson's rule requires 1.8 times more points to achieve the same accuracy.

Although some effort has been made to extend the Euler-Maclaurin summation formula to higher dimensions, the most straightforward proof of the rapid convergence of the trapezoidal rule in higher dimensions is to reduce the problem to that of convergence of Fourier series. This line of reasoning shows that if f is periodic on a n -dimensional space with p continuous derivatives, the speed of convergence is O($h^{p/d}$) . For very large dimension, the shows that Monte-Carlo integration is most likely a better choice, but for 2 and 3 dimensions, equispaced sampling is efficient. This is exploited in computational solid state physics where equispaced sampling over primitive cells in the reciprocal lattice is known as Monkhorst-Pack integration.

2.7 Trapezoid Laws

$$\text{where } h = \frac{b - a}{n}$$

$$\text{Area} = \int_a^b y\,dx \approx \frac{1}{2} h [y_0 + 2(y_1 + y_2 + \ldots + y_{n-1}) + y_n]$$

Example 2.2.3: Find the integral of the function in the trapezoidal way

$\int_0^1 (x^3 + 1)$ n =1

$$h = \frac{b - a}{n}$$

h = 1-0 /1 = 1

$\int_0^1 (x^3+1)\,dx$ =1/2 [1+2] = 3/2 = 1.5

Example 2.2.4 : Use the trapezoid formula to find the integral

$$\int_0^1 X^2 \, dx \qquad n = 4$$

$$h = \frac{b - a}{n}$$

h = 1-0 / 4 = 1 / 4 = 0.25

a = 0 b = 1

X	0	0.25	0.5	0.75	1
F(X)	0	0.0625	0.25	0.5625	1

$$\int_0^1 f(x)\,dx = h/2 \; [\, f_0 + 2f_1 + 2f_2 + 2f_3 + f_4 \,]$$

$$\int_0^1 X^2 \, dx = 0.25/2 \; [\, 0 + 2(0.0625) + 2(0.25) + 2(0.5625) + 1$$

$$= 0.125 \; [2.75] = 0.34375$$

Example 2.2.5 : Use the trapezoid formula to find the integral

$\int_0^1 \sqrt{1-x^2}\ dx$ n = 5

Solution

h = $x_1 - x_0$ / n = 1-0 / 5 = 1/5 = 0.2

x	0	0.2	0.4	0.6	0.8	1
y	1	0.9797	0.9165	0.8	0.6	0

$$\int_{x0}^{x1} f(x)\ dx = \frac{h}{2}\ [\ y0 + 2y1 + 2y2 + 2y3 + 2y4 + y5]$$

$$\int_0^1 \sqrt{1+x^2}\ dx$$

$$= \frac{0.2}{2}\ [\ 1 + 2(0.9797) + 2(0.9165) + 2(0.8) + 2(0.6) + 0]$$

= 0.1 [1+1.9594 + 1.833 + 1.6 + 1.2]

=0.75924

الخاتمة

لقد وصلنا لنهاية هذا البحث ، وفي النهاية لا يسعني سوى أن أشكركم على حسن متابعتكم لهذا البحث ، وأنا قد عرضت بهذا البحث رائي المتواضع ببركة الله تعالى وكرمه وتوفيقة ، وقد أكرمني الله بأن أدلوا بدلوي تجاه هذا الموضوع

(إيجاد الحل التقريبي باستخدام طريقة شبه المنحرف)

ولعل الله تعالى قد وفقني في هذا البحث في هذا الموضوع ، ولعل قلمي وفق في تقديم ما يدور بخلدي ، وفي نهاية الأمر فإنني بشر أصيب وأخطئ ، وإنني أتوجه إلى الله بالدعاء على توفيقي في تقديم هذا البحث وعلى حسن قراءتكم ومتابعتكم لهذا البحث ، ونشكر لكم سعة صدركم ونرجو أن ينال البحث إعجابكم ، والحمد لله الذي هدانا إلى هذا

(التوصيات)

1. **نوصي باستخدام طريقة اخرى عندما تكون متعدده الحدود من الدرجة الرابعة فما فوق**
2. **البحث عن التطبيقات التي تلائم هذه الطريقة لكي تكون فائدة علميه للقارئ**
3. **اضافة الى ما تقدم الأهتمام بجانب التحليل العددي لما له من أهمية في حل المسائل التطبيقة**

REFERENCES

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.NumericalInitialValueProblemsinOrdinaryDifferential Prentice Hall , Englewood Cliffs , NJ , 1971 .

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