ORIGINAL RESEARCH PAPER

Volume-6 | Issue-9 | September-2017 | ISSN No 2277 - 8179 | IF : 4.176 | IC Value : 78.46

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

A STUDY ON NUMERICAL ANALYSIS USING BOUNDARY VALUE PROBLEMS

Mathematics

Lavenya K M.Phil Research Scholar Department of Mathematics PRIST University Manamai Campus (Mahabalipuram) Manamai-603 127

M.Karthigeyan M.Sc.,M.Phil.,B.Ed., Research Supervisor Assistant Professor Department of Mathematics PRIST University Manamai Campus (Mahabalipuram) Manamai-603 127

ABSTRACT

The original motivation behind this thesis was to construct embedded Runge-Kutta methods for use in computing numerical solutions to hyperbolic conservation laws. The methods would use a high-order linearly stable Runge-Kutta scheme in smooth regions of the spatial domain and, in the vicinity of shocks or other discontinuities, switch to a lower-order scheme possessing a \nonlinear stability" property which would help prevent spurious oscillations and overshoots. The derivation of such a method turned out to be more challenging and interesting than was originally thought and, as such, this thesis has more to do with the construction of these embedded methods than it does with the original motivational example.

This chapter begins with an introduction to Runge-Kutta methods and linear stability. It then touches briey on the topics related to the solution of hyperbolic conservation laws, including nonlinear stability and strong-stability-preserving Runge-Kutta schemes. Finally, the chapter concludes with a discussion of linearly stable Runge-Kutta methods with embedded strong-stability-preserving Runge-Kutta schemes.

One of the earliest mathematical writings is a Babylonian tablet form the Yale Babylonian collection (YBC 7289), which gives a hexadecimal approximation of the length of the diagonal in a unit square. Being able to compute the sides of a triangle (and hence, being able to compute square roots) is extremely important, for instance, in astronomy, carpentry and construction.

Then, in the Euler's method is reviewed. The derivation of Euler's method is stated and using some basic definitions, error. After completing this, we will view the Runge Kutta Method of order four (RK4), we present some necessary definitions of the stability, and examined the absolute stability of Fourth Order Runge-Kutta method. Then, we give the general form of system for first order equations. The aim of this study is to compute the approximated solutions of system of differential equations. Then, it follows the stability theory for systems. Finally, some examples are given, and each numerical method is solved manually. After that, the results of numerical methods are analyzed. Finally, we concluded by giving the advantages and disadvantages. And the Runge kutta method is wide-used in solving ordinary differential equations, and it is more accurate than the Eul er method. Then, the error between Euler Explicit Method and Runge Kutta Method compared with the Exact Method.

KEYWORDS

Introduction

Euler's Method

Many differential equations in engineering are so intricate. It is inapplicable to have solution. Numerical methods provide ease for solving the differential equations. The simplest method to solve initial value problem is **Euler method** which have one step process for each equation before move on next step. This method is not an adequate method to get the certain approximation. Differential equation can be solved simply even though it is rather rough and least accurate. It is restricted to utilize because each successive step during the process accumulates large errors. It has slow of convergence which means a method of order 1. So the error is O(h). In contrast, the remainder term and error analysis in Euler method provide convenience to state the difference between the approximate and exact solutions.

Definition

Assume that initial value problem and

$$\frac{\partial y}{\partial x} = y'(x) = f(x, y(x))$$

y(x0)=b

is applied to numerical method on the specified interval: $[x_0, b]$. We can create nodes with equi-spaced subinterval for simplicity as

x0<x1<… <xN<b

If nodes is taken to be evenly spaced, numerical methods will generate an approximatesolution y_n. It is written simply which is called **mesh point** as follows

xn=x0+nh, n=0,1,…,N

where

$$h = \frac{(b - x_0)}{N}$$

and *h* is defined as **step-size** or (**step of integration**), a positive real number *N*. For each *n*, the numerical approximation y_n at a mesh points x_n can be smoothly obtained. The initial condition is known as

y(x0)=y0

Assume that we have already calculated y_n up to some n. This represent

yn+1=yn+hf(xn,yn) (2.2.1)

It is known as Euler method with the initial condition. To attain Euler's method, consider a forward difference approximation to the derivative

$$y'(x) \approx \frac{1}{h}[y(x+h) - y(x)]$$

Equalize this to the initial value problem at x_n. We obtain

$$\frac{1}{h}[y(x_n + h) - y(x_n)] = f(x_n, y(x_n))$$

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n))$$

Euler's method can be represented by considering the approximated values

$$y_{n+1} = y_n + hf(x_n, y(x_n)) \quad 0 \le n \le N-1$$

The other way to derive the Euler's method is to integrate the differential equation

$$\frac{\partial y}{\partial x} = y'(x) = f(x, y(x))$$

between two consecutive mesh points x_n and x_{n+1}. We conclude that

$$\int_{x_n}^{x_{n+1}} y'(x)dx = \int_{x_n}^{x_{n+1}} f(x, y(x))dx$$

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y(x))dx$$

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))dx \quad (2.2.2)$$

Here, we cannot integrate *f(x,y(x))* without knowing *y(x)*. Hence we must approximate the integral. Apply the numerical integration rule

$$\int_{x_n}^{x_{n+1}} g(x)dx \approx hg(x)$$

which is known as the **rectangle rule** with *g(x) = f (x, y(x))*. This means the simplest approach is to approximate the area under function *f (x, y(x))* by the rectangle with base *[xn, xn+1]* and height *f (x, y(x))*. We can define the rectangle rule

$$\int_{x_n}^{x_{n+1}} g(x)dx \approx h[(1-\Theta)g(x_n) + \Theta g(x_{n+1})] \quad (2.2.3)$$

with [0,1]Θ∈. Then, substitute (2.2.3) into (2.2.2) by considering *g(xn)=f(xn,y(xn))* to obtain

$$y(x_{n+1}) = y(x_n) + h[(1-\Theta)g(x_n) + \Theta g(x_{n+1})]$$
$$y(x_{n+1}) \approx y(x_n) + h[(1-\Theta)f(x_n, y(x_n)) + \Theta f(x_{n+1}, y(x_{n+1}))]$$
$$y(x_0) = y_0$$

Then, supply the initial conditions to the one parameter method that is mentioned above. This gives us the following method where Θ [0, 1]

$$y_{n+1} = y_n + h[(1-\Theta)f(x_n, y_n) + \Theta f(x_{n+1}, y_{n+1})]$$

This definition motivates **Θ−method** by considering approximate values. The Θ−method referred Euler's method as Θ = 0 which $_{yn+1}$ must be found merely left hand side. This definition that give $_{yn+1}$ directly is called **explicit methods**. For Θ = 1 we recover

the **Implicit (backward) Euler Method.**

$$y_{n+1} = y_n + hf(x_{n+1}, y(x_{n+1})) \qquad n=0,1,\ldots,N-1 \qquad (2.2.4)$$

In order to identify $_{yn+1}$, (2.2.4) need the solution of an implicit equation. Euler's method is also referred as Explicit Euler Method in order to pick out the difference. This scheme gives a result for the value of 12Θ=which is denominated **Trapezium Rule Method.** It is shown as

$$y_{n+1} = y_n + \frac{1}{2}h[f(x_n, y_n) + f(x_{n+1}, y_{n+1})], \qquad n=0,1,\ldots,N-1$$

The other way to obtain numerical method is using multistep method. In general form of multistep method, it is easier to achieve any numerical methods. Consider the given form of

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h\sum_{j=-1}^{p} b_j f(x_{n-j}, y_{n-j})$$

to obtain Euler method. As *p = 0*, we get that

$$y_{n+1} = a_0 y_n + h[b_{-1} f(x_{n+1}, y_{n+1}) + b_0 f(x_n, y_n)],$$

If we give values instead of a_0, b_{-1}, b_0 kind of $a_0 = 1, b_{-1} = 0, b_0 = 1$

gives Euler method formulae. In contrast, if the values are taken as

$$a_0 = 1, b_{-1} = 1, b_0 = 0$$

we get Implicit Euler method

$$y_{n+1} = y_n + hf(x_{n+1}, y(x_{n+1}))$$

Which *yn+1* will occur in both sides.

Runge Kutta Method

The derivation of Runge-Kutta schemes involving higher derivatives is now on the increase. Traditionally, given an initial value problem (IVP), classical explicit Runge- Kutta methods are derived with the intention of performing multiple evaluations of *f(y)* in each internal stage for a given accuracy. Recently, Akanbi derived multi-derivative explicit Runge-Kutta method involving up to second derivative. Goeken and Johnson also derived explicit Runge-Kutta schemes of stages up to four with the first derivative of *f(y)*. However, the new scheme is derived with the notion of incorporating higher order derivatives of up to the second derivative. The cost of internal stage evaluations is reduced greatly and there is an appreciable improvement on the attainable order of accuracy of the method.

The Runge-Kutta method is popular method for solving initial value problem. It is most accurate and stable method. It arise when Leonhard Euler have made improvements on Euler method to produce Improved Euler method. Then, Runge is realized this method which is similar method with the second order RungeKutta method. A few years later in 1989 Runge acquired Fourth Order of RungeKutta method and afterwards, it is developed by Heun(1900) and Kutta(1901). Fourth Order Runge-Kutta method intend to increase accuracy to get better approximated solution. This means that the aim of this method is to achieve higher accuracy and to find explicit method of higher order. In this section, we discuss the formulation of method, concept of convergence, stability, consistency for RK4 method. In spite of the fact that Runge-Kutta methods are all explicit, implicit Runge-Kutta method is also observed. It has the same idea of Euler method. Euler method is the first order accurate; in addition it require only a single evaluation of f (x_n, y_n) to obtain yn+1 from yn. In contrast, Runge-Kutta method has higher accuracy. It re-evaluates the function f at two consecutive points (x_n, y_n) and (x_{n+1}, y_{n+1}). It requires four evaluations per step. Due to this, Runge-Kutta method is quite accurate, and it has faster rates of convergence.

Definition

Analyze the initial value problem to find the solution of ordinary differential equations'

x)=f(x,y(x)) (3.2.1)

y(x0)=y0

The general form Runge-Kutta method is

yn+1=yn+hF(xn,yn;h), n=0,1,...,N-1 (3.2.2)

where *F(.,.;.)* is called an increment function on the interval [xn,xn+1]. It can be defined in general form as

F(x,y;h)=b1k1+b2k2+…+bnkn. (3.2.3)

where bn's are constant and k_n's are

k1=f(xn,yn)
k2=f(xn+c2h,yn+a21k1h) (3.2.4)
k3=f(xn+c3h, yn+a31k1h+a32k2h)

Ks=f(xn+csh, yn+as,1k1h+as,2k2h+…+as,s-1ks-1h)

where c's and a's are constants. Here, each of function k's are represent slope of the solution which are approximated to *y(x)*. These coefficients $_{aij,cj,bj}$ must be satisfied the system of Runge-Kutta method. These are usually arranged in Butcher tableau or (*partitioned tableau*)

c	A
	bT

The coefficient bT is a vector of quadrature weights, and aij (*i,j= 1,2,...,s*) indicates the matrix A.

System of Ordinary Differential Equations

More application problems include a system of several equation. The solution of a system of ordinary differential equation is required in engineering and science that has more complicated situations. The initial value problem of m differential equation's system can be put into a form as

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0$$

We can write as follows

$$y_1'(x) = f_1(x, y_1(x), y_2(x), \ldots, y_m(x)), \qquad y_1(x_0) = y_{1,0}$$

$$y_2'(x) = f_2(x, y_1(x), y_2(x), \ldots, y_m(x)), \qquad y_2(x_0) = y_{2,0}$$

$$y_m'(x) = f_m(x, y_1(x), y_2(x), \ldots, y_m(x)), \qquad y_m(x_0) = y_{m,0}$$

with the given some interval $x0 \le x \le b$. The general form of system can be represented the solution and the differential equation by using the column vector. Indicate,

$$\mathbf{y}(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_m(x) \end{bmatrix}, \mathbf{y}_0 = \begin{bmatrix} y_{1,0} \\ y_{2,0} \\ \vdots \\ y_{m,0} \end{bmatrix}, \mathbf{f}(x, \mathbf{y}) = \begin{bmatrix} f_1(x, y_1, y_2, \ldots, y_m) \\ f_2(x, y_1, y_2, \ldots, y_m) \\ \vdots \\ f_m(x, y_1, y_2, \ldots, y_m) \end{bmatrix}$$

with *y=[y1,y2,…,ym]T.*

Example: *The initial value problem*

$$x' = x + 2y; \; x(0) = 6 \qquad (4.1.1a)$$

and

$$y' = 3x + 2y; \; y(0) = 4 \qquad (4.1.1b)$$

then find the exact value.

Solution:

$$\frac{1}{2}(x' - x) = y$$

substitute the above value in (4.1.1b), then we will get,

$$\frac{1}{2}(x'' - x') = 3x + (x' - x)$$

$$x'' - 3x' - 4x = 0$$

Then by Auxiliary equation, we get,

$$m^2 - 3m - 4 = 0$$

$$m = -1, +1$$

$$x(t) = c_1 e^{-t} + c_2 e^{4t} \qquad (4.1.1c)$$

$$y(t) = -c_1 e^{-t} + \frac{3}{2} c_2 e^{4t} \qquad (4.1.1d)$$

Substitute x(0)=6 and y(0)=4 in (4.1.1c) and (4.1.1d) respectively, then we obtain,

$$c_1 = 2 \; and \; c_2 = 4$$

Then the general solution is,

$$x(t) = 2e^{-t} + 4e^{4t}$$

$$y(t) = -2e^{-t} + 6e^{4t}$$

Numerical Experiments on Simple Systems

Numerical experiments is mentioned to prove which numerical methods converge faster to analytic solutions. Numerical experiments are investigated over an example. In comparing different numerical methods, numerical experiments must be done using the same of ordinary differential equations. For comparison bases, Hull and Enright (1976) pointed out some assumptions that must be undertaken. Assumptions include assuming that the method is modelled to integrate between initial values specified. Another assumption is assuming that local error is observed by keeping the absolute error under the specified error tolerance. Differential equation is inspected two varied numerical methods.

The given differential equations are analyzed for Explicit Euler method, Explicit Runge-Kutta method. Analytical solution of ODE is calculated as well. Each one of all is examined at varied step sizes. Step size is started with 0.1 and continued with halved. Afterward, the absolute error is identified. Initially, exact solution of DE is computed and then approximated solution is calculated for each numerical method at different step size for h=0.1 and h=0.05. Afterwards, absolute error are computed by taking difference analytical solution to approximated solutions and presented in the same table at each step size. Ultimately, errors at each step size of method are compared with other numerical methods. All numbers in the table.

Numerical Experiments on Stiff System(Numerical Experiments on Explicit Euler method)

Consider the following example of stiff system;

$$\frac{dx}{dt} = f(t, x, y) \; and \; \frac{dy}{dt} = g(t, x, y)$$

$$\frac{dx}{dt} = (x + 2y) \; and \; \frac{dy}{dt} = (3x + 2y)$$

$$dx = (x + 2y)dt \; and \; dy = (3x + 2y)dt$$

where

$$dt = t_{k+1} - t_k \; and \; h = \frac{b - a}{n}$$

$$and \; t_{k+1} = h + t_k$$

$$dx = x_{k+1} - x_k \; and \; dy = y_{k+1} - y_k$$

$$x_{k+1} - x_k = (x + 2y)h \; and \; y_{k+1} - y_k = (3x + 2y)h$$

$$x_{k+1} = x_k + h(x_k + 2y_k) \; and \; y_{k+1} = y_k + h\left(3x_k + 2y_k\right)$$

where initial conditions are $x_0 = x(0) = 6 \; and \; y_0 = y(0) = 4$ It can be shown in general form as

$$z(0) = [6, 4]^T$$

Explicit Euler method and Explicit Fourth Order Runge Kutta method of numerical methods is considered to solve system of ordinary differential equations. Initially, Explicit Euler method and Explicit Fourth Order Runge Kutta method is computed for the given system at different step size. Then analytical solution and approximated solution is evaluated with step sizes 0.1 and 0.05. Afterwards, the absolute errors which is obtain by differ approximated solution from analytical solution, are calculated at specified step size. Error tables with exact solution which is related with exact solution is shown in tables (5.2.1a) and (5.2.1b). In this table, we can see how the approximated solutions behave when the step size is reduced by half. We can compare the exact solution with approximate solution and identify the safe step size ,which the approximated solution tends to analytical solution, through this table. It can be determined the proximity of the approximated solution to exact solution since the step size is changed.

In order to compute the approximated solution, step size is started with 0.1 for Explicit Euler method. As can be seen from the table in step size 0.1 and 0.05 the resulted of approximated solution is not closeness to exact solution. So, these step sizes do not lie in the region of absolute stability. The difference between exact solution and approximated solution must be excessively small. The computed approximated solutions have much difference from analytical solutions as it seen in the table. Nevertheless, the step size will lie inside the region of absolute stability since $h < 0.05$. The approximated solution is closer to exact solution. This shows that the small step size provides the better approximation. Application of Explicit Euler method is not occasionally preferred to use in stiff systems. The reason is that cannot give the accurate approximated solution in fixed error tolerance well. In order to achieve certain approximation, the step size must be taken very small. This cause more iterations and high computation.

In this thesis, we will mention and discuss about numerical methods for solving systems of ordinar differential equations. Some necessary conditions and definitions are given to examine the numerical methods. After that, by considering these conceptions and definitions, Euler methods and Runge Kutta method of order 4 are developed and, derivations and stabilities of both methods are discussed. Afterwards, the system of ODE is given and, we discussed the efficiency of two methods at the different step size for systems using Tables of approximated solutions with exact solutions. At the end, we gave information about the advantages and disadvantages of Euler's method and Runge Kutta method. Consequently, we see that in the Euler's method excessively small step size converge to analytical solution. Therefore, large number of computation is needed. In contrast, Runge Kutta method gives better results and it converge faster to analytical solution and has less iteration to get accuracy solution.

REFERENCES

[1]. J.C. Butcher and G. Hojjati, Second derivative methods with RK stability, Numerical Algorithms, (2005), 415-429.
[2]. Raltson, Runge-Kutta with Minimum Error Bound, Maths Computation., 16, 431 - 438.
[3]. D.J. Okunbor, Explicit Runge-Kutta scheme for stiff system of ODEs, M.sc Thesis, University of Benin, Benin City, (Unpublished), 1985.
[4]. Hong Yuan Fu, A class of A-stable or A(α) stable explicit scheme, Proceedings of BAILJ Conference Trinity College, Dublin, (1982), 231-241.
[5]. Atkinson, K. E., and Han, W., and Steward, D.,(1961), Numerical Solutions of Ordinary Differential Equations, John Wiley & Sons, Inc., Hoboken, New Jersey, Canada
[6]. http://www.math.uiowa.edu/~atkinson/m171.dir/sec_6-3.pdf
[7]. http://people.maths.ox.ac.uk/suli/nsodes.pdf
[8]. http://plato.asu.edu/MAT275/odes_matlab/Chapter12.pdf
[9]. http://www.siam.org/books/ot98/sample/OT98Chapter7.