Necessary Revisit to Euler Methods

[1]S Chatterjee, [2]S Guha Mallick, [3]S Pal, [4]B N Biswas
[1]Kanailal Vidyamandir (Fr. Section), Chandernagore, Hooghly, West Bengal, India
[2,3,4]Sir J.C. Bose School of Engg., SKF Group of Institutions, Mankundu, Hooghly, West Bengal, India

Abstract
In digital signal processing the consideration is fidelity, speed and costs which demand simplicity of system, whereas fidelity / accuracy need a complicated system involving greater costs. Again balance has to be made against the requirements and allowable tolerance in the performance. In this sense computer oriented mathematics is different from numerical mathematics. In most of the text emphasis has been given on the dependence of the accuracy of the solution on the step size of the independent variable rather than on the stability of the method and the effect of truncation and round off errors. However nothing is discussed on the error due to nesting of forward Euler method in backward Euler method as well as trapezoidal method. It is shown in this paper that there is a maximum value of step size up to which the numerical algorithms are stable. This tutorial paper attempt to illustrate these issues by taking a simple practical example that has exact analytical solutions, so that pitfalls of the numerical algorithm can be vividly illustrated. Moreover, the discussion of linear systems offers considerable insight into the solution of nonlinear equations. To appreciate the nature of approximation involved in numerical solution, knowledge of the form of exact solution is important and as such a linear system has been chosen.

Keywords
Euler Method, Round off error, Truncation

I. Introduction

Euler algorithm introduced by L. Euler in 1728, is one of the simplest numerical methods of solving the difficult differential equations which are not amenable to analytical solutions [1]. It is the simplest and the most analyzed numerical integration; it has become the stepping-stone of numerical methods for solving Initial value Problems in Ordinary Differential Equations. "A downside however is that it can sometimes have a tendency to be unstable unless you take stupidly small steps in the algorithm, in cases like this there are some other methods that work better." Differential equations are one of the most important mathematical tools used in modeling problems in physical sciences. Historically, Differential Equations (DE) have originated in chemistry, physics and engineering. More recently, they have also arisen in medicine, biology, anthropology, and the like. Ordinary differential equations arise frequently in the study of physical systems [2-5]. This is why the ability to numerically approximate these methods is so important [6-7]. Numerical solution of Ordinary Differential Equations (ODEs) is the most important technique ever developed in continuous time dynamics and so numerical integration is the only way to obtain information about the system.

But these numerical methods can only be applied to finding specific areas because they deal with actual numbers rather than variables. This means that they can be used to evaluate definite integrals but cannot help at all in finding an indefinite integral. Numerical methods are employed to calculate the values taken by the dependent variables over a range of values of the independent variables.

With the advent of computers, numerical methods are now an increasingly attractive and efficient way to obtain approximate solutions to differential equations that had hitherto proved difficult, even impossible to solve analytically [8-10].

Here it is important to note that it is necessary to discretize the independent variable since computers do not understand continuous variables. Further it is very interesting to note that Euler Algorithm is ideally suited for computer solution as it discretizes the differential equation. Euler gave this idea long before the concept of programming was introduced by Ada Lovelace, the daughter of Lord Byron, in 1842.

Humans can handle indeterminate quantity, which a computer cannot. Because of the computers are sometimes referred as "Supersonic Morons" where as a mathematician is designated as a 'Deliberate Genius". Numerical techniques thus aim at solving this mismatch between humans and computers.

II. Round-off and Truncation

Obviously the speed and accuracy with which a computer can achieve, depends on the number of decimal digits involved in this process, i.e. on the complexity of the machine. This is a very important in the practical use of computers, like computer aided design, digital signal processing etc. for example; consider design of a bridge over a river. In such process you will need a number of steel rods or plates (aside other things) with different lengths, breadths and thickness. Say, your calculation needs a particular steel plate of 10.56267 meters. But the accuracy with which you can select the length is up to the five places of decimal digit. Then you have got to take either 10.562 or 10.563 meters. Then the question is which one will you choose? What will be basis of such choice-obviously, safety, longevity accuracy, computation time and cost? A judicious balance is to be made. In the language of mathematics in one case i.e., 10.562, we commit an error called "truncation error" and in the case i.e., 10.563, it is called 'round-off' error. In digital signal processing the consideration is fidelity, speed and costs. Here when one requires speed, it demands simplicity of system, whereas fidelity / accuracy need a complicated system involving greater costs. Again balance has to be made against the requirements and allowable tolerance in the performance. In this sense computer oriented mathematics is different from numerical mathematics. In most of the text emphasis has been given on the dependence of the accuracy of the solution on the step size of the independent variable rather than on the stability of the method and the effect of truncation and round off errors.

This tutorial paper attempt to illustrate these issues by taking a simple practical example that has exact analytical solutions, so that pitfalls of the numerical algorithm can be vividly illustrated. Moreover, the discussion of linear systems offers considerable insight into the solution of nonlinear equations. To appreciate the nature of approximation involved in numerical solution, knowledge of the form of exact solution is important which is possible for a linear system. This paper does not contain much new materials. However, a simplistic approach has been presented in order to whet the students' appetite for subsequent in-depth study of the subject.

III. Numerical Experiment with Charging of a Capacitor Through a Resistance

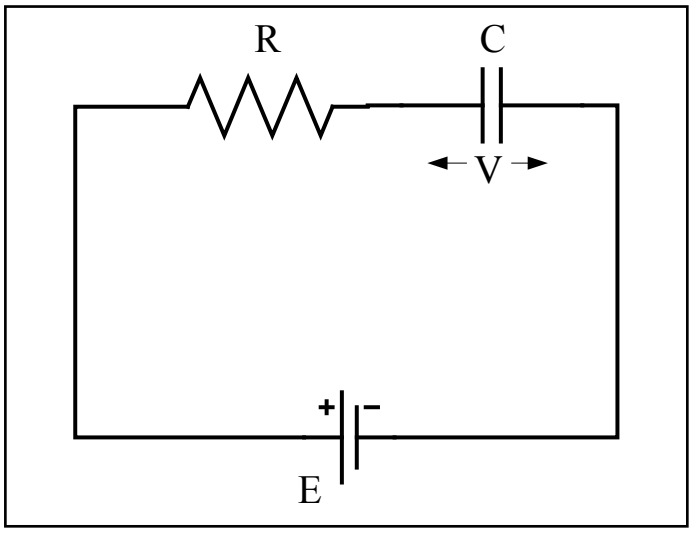


Fig. 1: Charging of Capacitor

In order to illustrate the objectives of the paper, we consider a simple practical problem of charging a capacitor through a resistance as shown in the fig. 1. It is required to plot the variation of voltage $V(t)$ across the capacitance with time. It can be done in two ways by hardware experiment or by numerical experiment, based on computer simulation program. The behavior of the system is governed by

$$E = RC\frac{dV}{dt} + V(t) \quad , \tag{1}$$

with the initial condition $V(0) = 0$.

Referring to (1) and substituting

$$x = \frac{E}{RC} - \frac{V}{RC} \tag{2}$$

i.e., $x(0) = \frac{E}{RC}$, since at $t = 0$, $V = 0$

thus,

$$\frac{dx}{dt} = -\frac{x}{RC} \quad . \tag{3}$$

A. Forward Euler Algorithm

It is possibly the simplest algorithm to implement and is sometimes known as Explicit Euler Method. We write (3) for a finite time interval Δt

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = -\frac{x(t)}{RC} \tag{4}$$

Discretization of (3) is necessary since computers do not understand continuous time. That is why we divide the time internal $0 \le t \le t$, into $'n'$ equal parts. Here $'n'$ is a natural number.

Thus $\Delta t = \frac{t}{n}$ (5)

From (4) one finds

$$x(t+\Delta t) = x(t)\left(1 - \frac{\Delta t}{RC}\right)$$

Therefore,

$$x(\Delta t) = x(0)\left(1 - \frac{\Delta t}{RC}\right) = \frac{E}{RC}\left(1 - \frac{\Delta t}{RC}\right)$$

$$x(2\Delta t) = \frac{E}{RC}\left(1 - \frac{\Delta t}{RC}\right)^2$$

i.e. $x(n\Delta t) = \frac{E}{RC}\left(1 - \frac{\Delta t}{RC}\right)^n$

Thus

$$V(n\Delta t) = E\left[1 - \left(1 - \frac{\Delta t}{RC}\right)^n\right] \tag{6}$$

From the physical condition of the problem, it is at once seen that

$$\lim_{n\Delta t \to \infty} V(n\Delta t) = E \tag{7}$$

Therefore, using (6) and (7) it is found for numerical stability

$$\left|1 - \frac{\Delta t}{RC}\right| < 1 \tag{8}$$

i.e., $0 < \Delta t < 2RC$ (8a)

This is commonly known as the criterion for numerical stability. This limits the maximum step size depending circuit parameters (R & C). This dictates the dependence of minimum time of computation with values of the circuit parameters, i.e. system parameters.

B. Backward Euler Algorithm

This algorithm is sometimes called Implicit Euler Method and it is more difficult to implement for nonlinear system. To illustrate let us consider the system defined by the following differential equation

$$\frac{dy}{dt} = \sin(y)$$

with $y(0) = 1$

Thus $\frac{y(t+\Delta t) - y(t)}{\Delta t} = \sin\left[y(t+\Delta t)\right]$

That is, $y(t+\Delta t) - \sin\left[y(t+\Delta t)\right]\Delta t = y(t)$ (9)

Thus to know the value of $y(t+\Delta t)$ one has to solve the equation (9) with a periodic non-linearity. Thus an extra work to be done but it is a more accurate method. To prove this we have chosen the linear deferential equation (1).

Referring to (3) we write

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = -\frac{x(t+\Delta t)}{RC} \tag{10}$$

Hence

$$x(t+\Delta t) = x(t)\left(\frac{1}{1 + \frac{\Delta t}{RC}}\right)$$

Proceeding as in the case forward Euler algorithm one finds

$$x(n\Delta t) = \frac{E}{RC}\left(\frac{1}{1+\frac{\Delta t}{RC}}\right)^n$$

i.e. $$V(n\Delta t) = E\left[1-\frac{1}{\left(1+\frac{\Delta t}{RC}\right)^n}\right] \quad (11)$$

From equation (11) it is clear that for the stability of this numerical method, it is necessary that $\Delta t > 0$. This is putting no restriction on the choice of the step size (Δt). But there is a little 'but'. We have to check the accuracy of the solution. To check the validity of these algorithms we implement the definition of the derivative that $n \to \infty$ and $\Delta t \to 0$ .Using $n\Delta t \to t$ and using starling approximation

$$\lim_{\rho \to 0} (1-\rho)^{\frac{1}{\rho}} = e^{-1} \quad (12)$$

and

$$\lim_{\rho \to 0} \frac{1}{(1+\rho)^{\frac{1}{\rho}}} = e^{-1} \quad (13)$$

One finds from the above calculations

$$V(t) = E\left[1-\exp\left(-\frac{t}{CR}\right)\right] \quad (14)$$

which one easily verify by using standard method of solving (1).

C. Forward Euler Algorithm Plus Backward Euler Algorithm

Adding (4) and (10) one finds

$$\frac{x(t+\Delta t)-x(t)}{\Delta t} = -\frac{1}{RC}\frac{x(t)+x(t+\Delta t)}{2}$$

That is,

$$x(t+\Delta t) = x(t) - \left[\frac{1}{RC}\frac{x(t)+x(t+\Delta t)}{2}\right]\Delta t$$

Note that the second term indicates the area of a trapezoid. As such it is commonly identified as a trapezoidal method. Anyway following the steps as outlined in the previous sections it is easily shown

$$x(t+\Delta t) = x(t)\frac{1-\frac{\Delta t}{2RC}}{1+\frac{\Delta t}{2RC}} \quad (15)$$

Following the methods as outlined in sections 2.1 and 2.2, it is easily shown that

$$V(n\Delta t) = E\left[1-\left(\frac{1-\frac{\Delta t}{2RC}}{1+\frac{\Delta t}{2RC}}\right)^n\right] \quad (16)$$

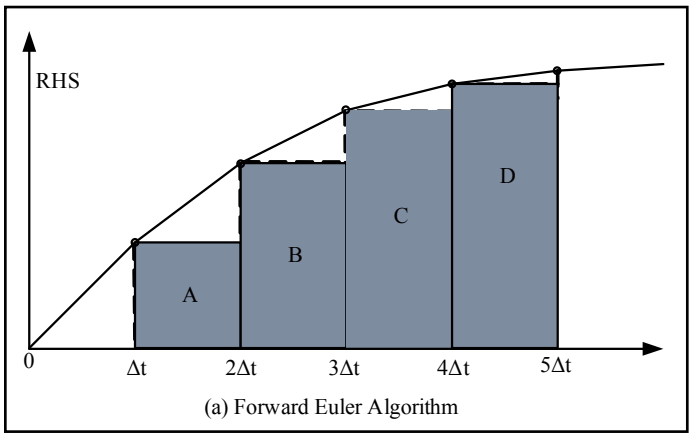
In implementing the condition of differentiation it can be easily shown that (16) also leads to (14). For stability of the numerical solution it is necessary that

$$0 < \Delta t < 4RC \quad (17)$$

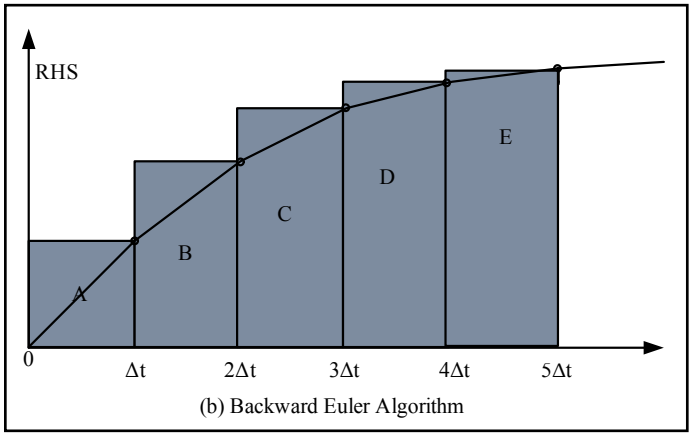
But for large values of RC, numerical instability creeps in.

IV. Geometrical Interpretation

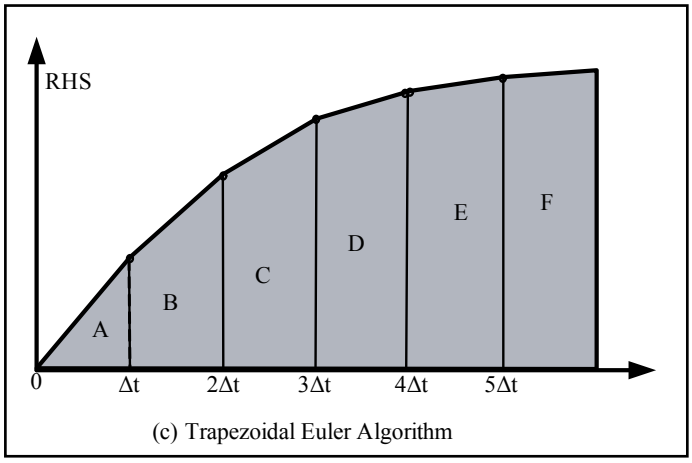
Remember that at $t = 0$, $x = \frac{E}{CR}$ and when $t \to 0$, $x \to 0$ Therefore, $x(t)$ versus t, looks like as shown in fig. 2. Geometrically the solution curve is approximated to by a series of straight lines. That horizontal projection of each of these lines is of length Δt and their slopes are calculated from the original differential equation. The addition of these areas reads to the value of hence $V(n\Delta t)$. It is to be noted that we have illustrated three basic numerical methods in order to make it vivid that the choice of appropriate numerical algorithm is important. If the practical algorithm indicates uncertainties of 1.0 percent then it is pointless to try to achieve 6 digits of accuracy with that particular algorithm.



(a) Forward Euler Algorithm



(b) Backward Euler Algorithm



(c) Trapezoidal Euler Algorithm

Fig. 2: Geometrical Representation of Euler Method (a) Forward (b) Backward and (c) Trapezoidal

V. Stability Zone

Considering the most general case when the capacitance is placed by a complex one due to the complex nature of the dielectric constant: we write

$$\frac{1}{RC} = s = \sigma + j\omega \tag{18}$$

Referring to (8), we find for numerical stability

$$\left|(1 - s\Delta t)\right| < 1$$

$$(1+\sigma\Delta t)^2 + (\omega\Delta t)^2 < 1 \tag{19}$$

The stability boundary is shown in fig. 3(a). This is for the forward Euler Algorithm.

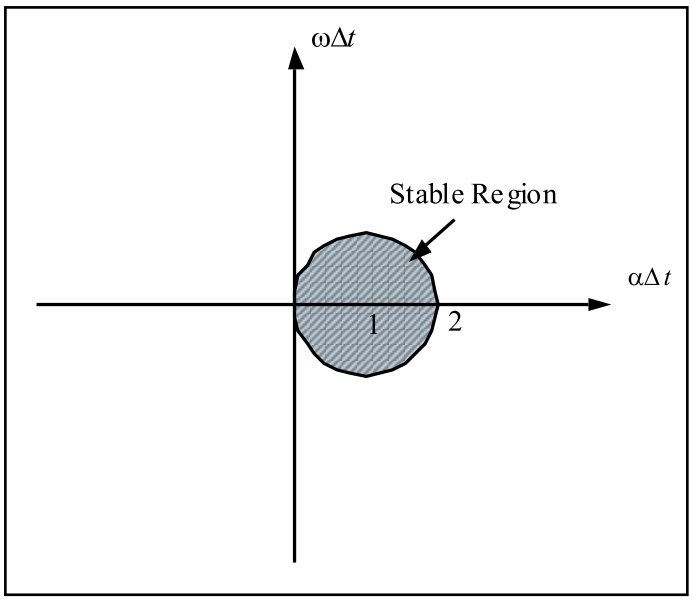


Fig. 3(a): Stability Region for Forward Euler Algorithm

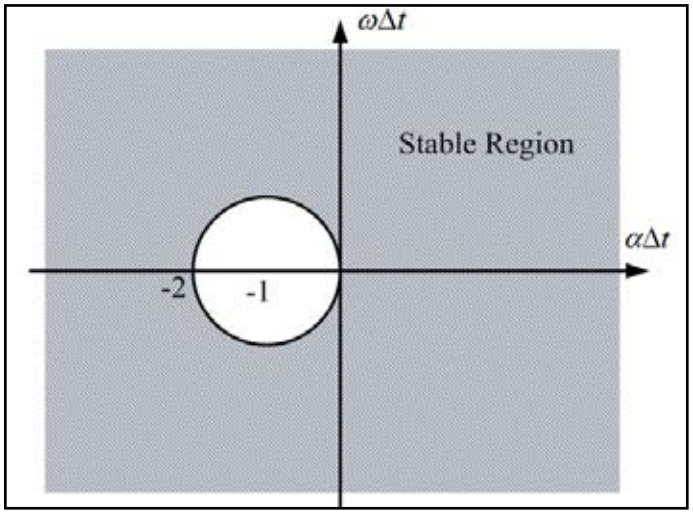


Fig. 3(b): Stability Region for Backward Euler Algorithm

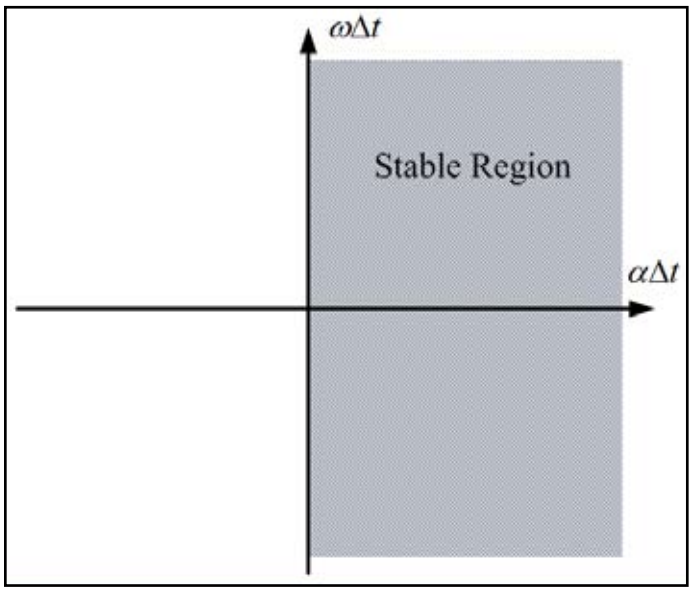


Fig. 3(c): Stability Region for Trapezoidal Euler Algorithm

Referring to (11) one can easily conclude that for stability of solution it is necessary

$$\left|1 + s\Delta t\right| > 1 \tag{20}$$

As the boundary condition of the problem suggests that the charge across the condenser must decay to zero in the steady state $(n \to \infty)$ and so also $x(n\Delta t)$ is also zero in the steady state. Noting that

$$s = \sigma \pm j\omega \tag{21}$$

One find from (13)

$$(1+\sigma\Delta t)^2 + (\omega\Delta t)^2 > 1 \tag{22}$$

This is shown in fig. 3(b). This is for Backward Euler Algorithm.

Backward Euler Algorithm is absolutely stable in the exterior of the unit circle coated

$$\sigma\Delta t = -1 + j0$$

For a trapezoidal the corresponding equation for the stability is given by

$$Z = \frac{(1-\frac{s}{2}.\Delta t)}{(1+\frac{s}{2}\Delta t)} = \frac{\left[(1-\frac{\sigma\Delta t}{2})-\frac{j\omega\Delta t}{2}\right]}{\left[(1+\frac{\sigma\Delta t}{2})+j\frac{\omega\Delta t}{2}\right]}$$

$$|Z| = \sqrt{\frac{(1-\frac{\sigma\Delta t}{2})^2+(\frac{\omega\Delta t}{2})^2}{(1+(\frac{\sigma\Delta t}{2})^2+(\frac{\omega\Delta t}{2})^2}}$$

(23)

When $|Z|<1$, $\sigma\Delta t\rangle 0$; When $|Z|=1$, $\sigma\Delta t=0$ and when $|Z|\rangle 1$, $\sigma\Delta t\langle 0$. Thus it is absolutely stable on the right half-plane in fig. 3(c).

VI. Simulation: Result and Discussion

Numerical methods are employed to calculate the values taken by the dependent variables over a range of values of the independent variables. Fig. 4 shows the simulation result regarding solution of equation using Euler method (Forward, Backward and Trapezoidal) with round and floor command in MathCAD. The variation of amount of error with normalized time constant is shown in fig. 5 in Euler method (Forward, Backward and Trapezoidal) with floor command in MathCAD.

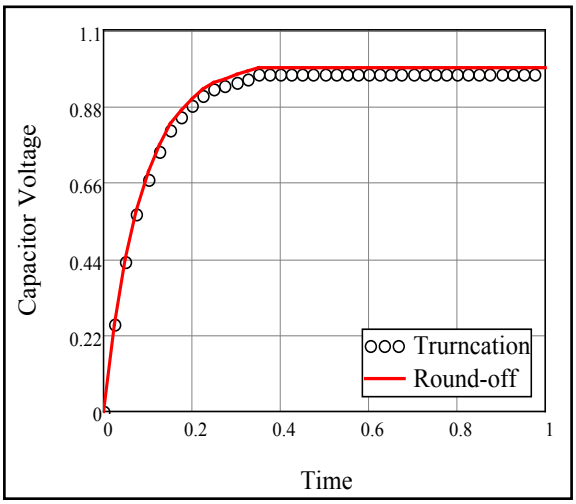


Fig. 4(a): Forward Euler Method

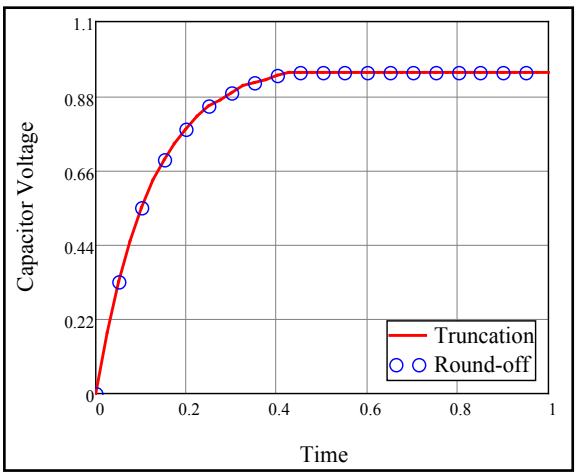


Fig. 4(b): Backward Euler Method

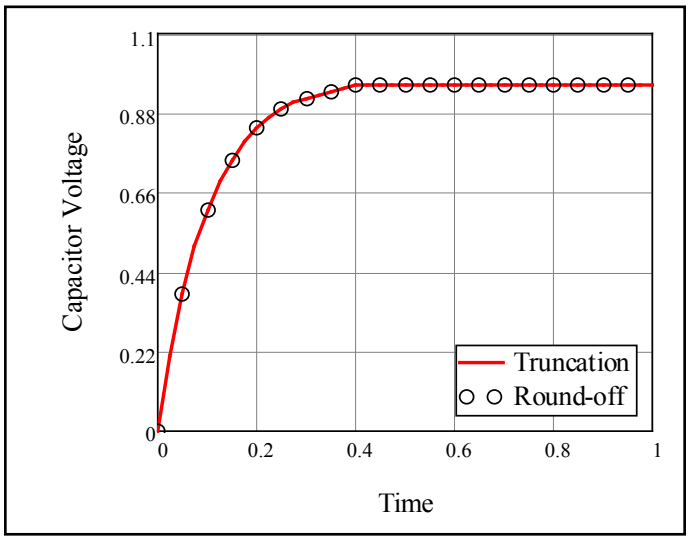


Fig. 4(c): Trapezoidal Euler Method

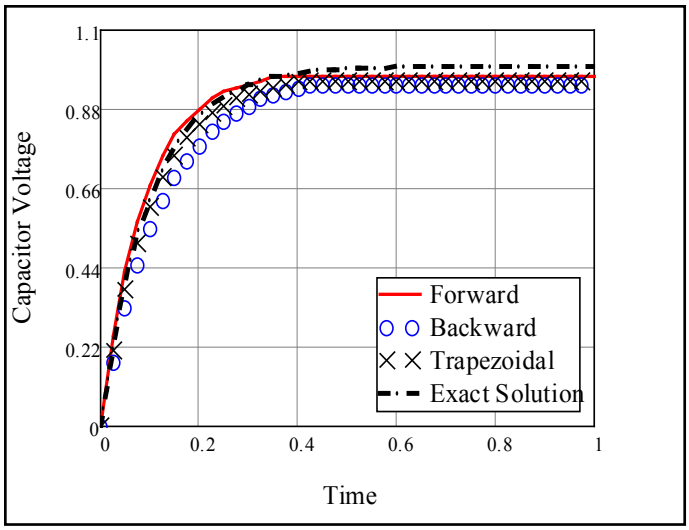


Fig. 4(d): Euler Method (Floor, N = 40)

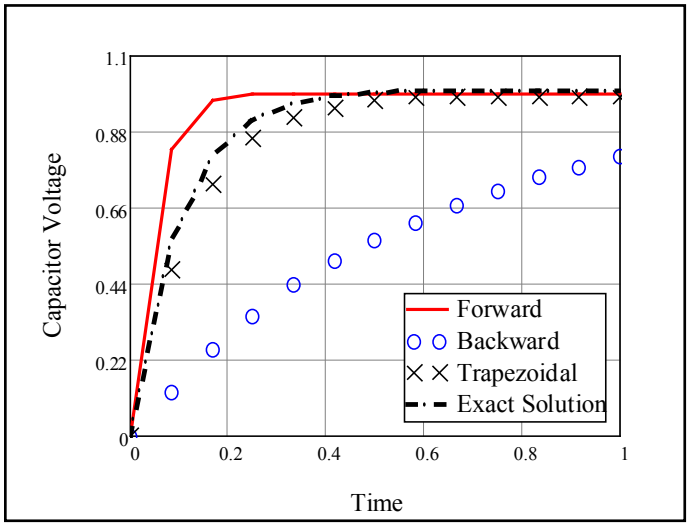


Fig. 4(e): Euler Mrthod (Floor) (N = 12)

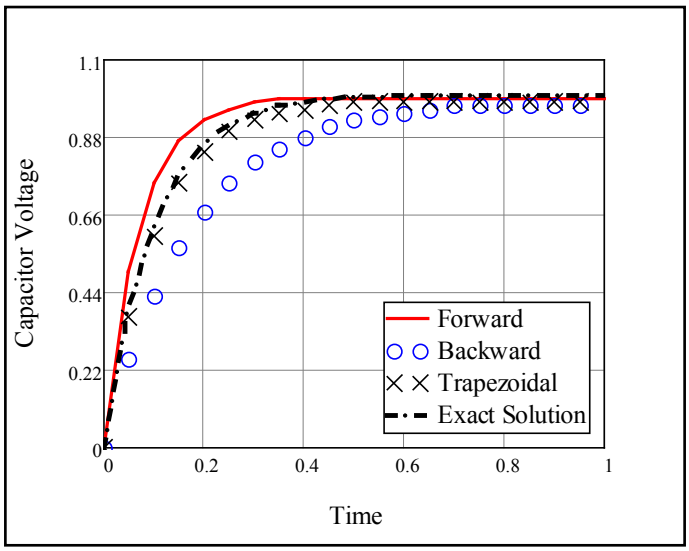


Fig. 4(f): Euler Method (Floor, N= 20)

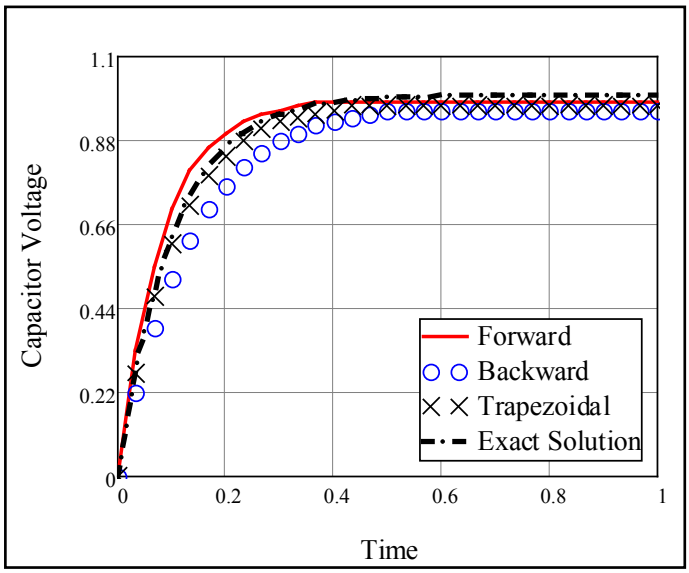


Fig. 4(g): Euler Method (Floor, N = 30)

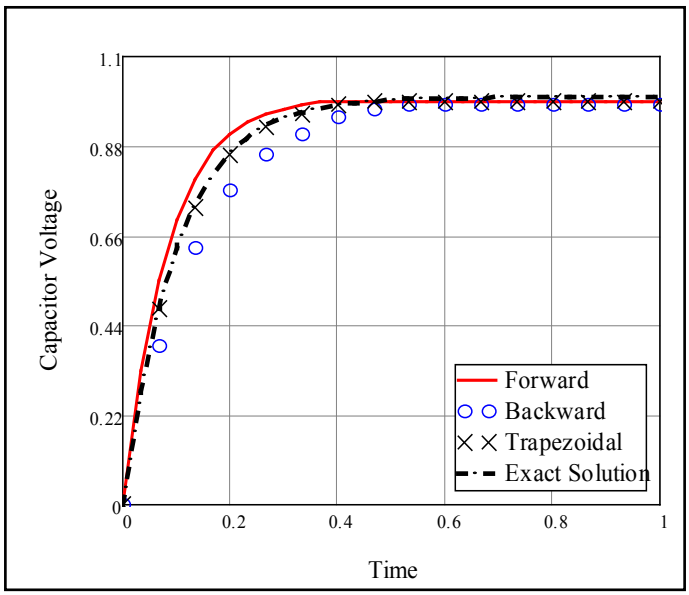


Fig. 4(h): Euler Method (Round, N = 30)

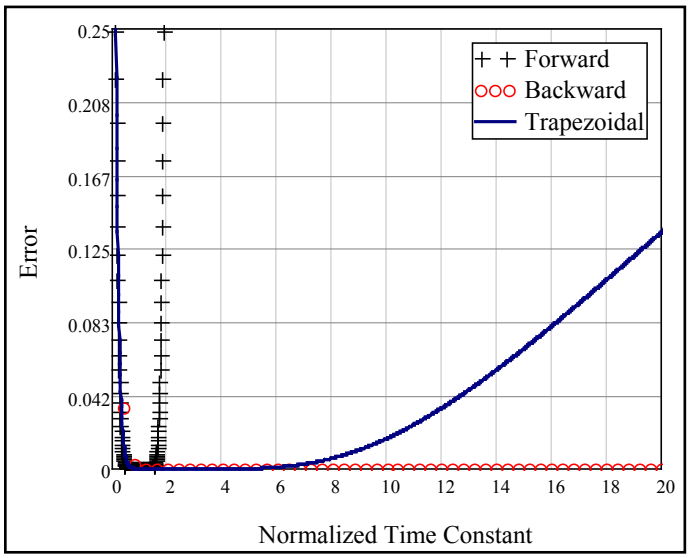


Fig. 5: Variation of Error with Normalized Time Constant Euler

VII. Limitation in Backward and Trapezoidal Euler Method

It is to be noted that the fig. 4, does not truly reflect the advantages or disadvantages of the backward Euler method or Trapezoidal method because in order to make these methods we have to approximate the "y_{n+1}" inside the function term $f(y_{n+1})$ by using forward Euler method. The effect is illustrated in fig. 6.

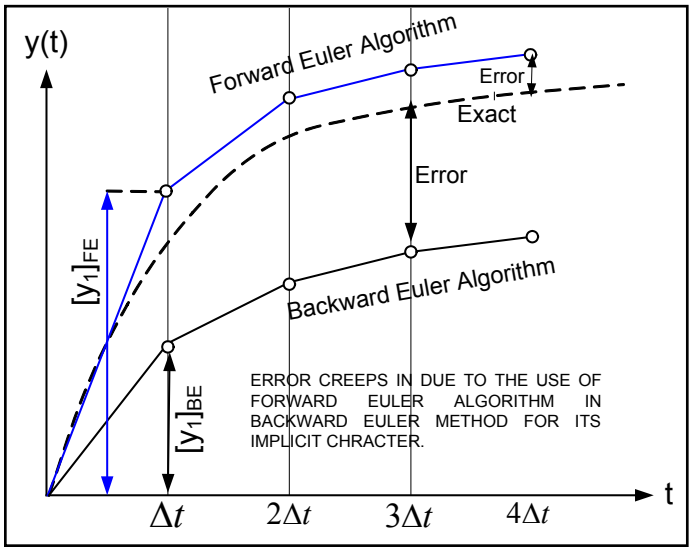


Fig. 6: Error in Backward Euler Method due to the use of Forward Euler Algorithm

This can be illustrated as follows. Consider the first order ODE with following initial conditions, viz., at t = 0, y = y_0.

$$\frac{dy}{dt} = f(y)$$

In the case of backward Euler algorithm we can write

$$y_1 = y_0 + f\left[y_0 + f(y_0)\Delta t\right]\Delta t$$

For the method to follow through

$$y_0 = y_0 + f\left[y_0 + f(y_0)\Delta t\right]\Delta t$$

That is

$$f\left[y_0 + f(y_0)\Delta t\right] = 0 \qquad (23)$$

From (23) we can easily find the maximum value Δt for the algorithm will fall through. For example, in our case, $f(y) = 10(1-y)$. Therefore,

$f(y_0) = 10$

$(1 - 0 - 10\Delta t) = 0$

Therefore,

$$\Delta t = 0.1 \qquad (24)$$

It is clear from the fig. 6 and fig. 7(a) that if the time step is increased beyond a certain value depending on the nature of the f(y) the computation procedure falls through.

Now in the case of trapezoidal method we can write

$$y_1 = y_0 + \left[f(y_0) + f\left[y_0 + f(y_0)\Delta t\right]\right]\Delta t/2 \qquad (25)$$

For the method to follow through

$$y_0 = y_0 + \left[f(y_0) + f\left[y_0 + f(y_0)\Delta t\right]\right]\Delta t/2 \qquad (26)$$

So that $f(y_0) + f\left[y_0 + f(y_0)\Delta t\right] = 0$ (27)

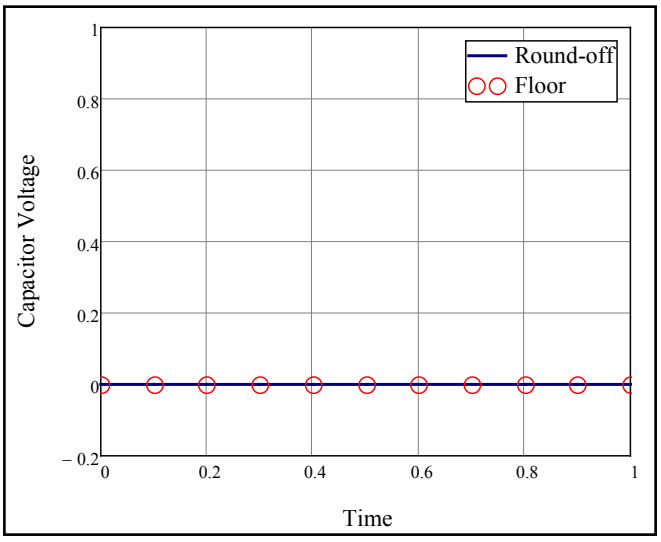


Fig. 7(a): Backward Euler algorithm for Δt =0.1

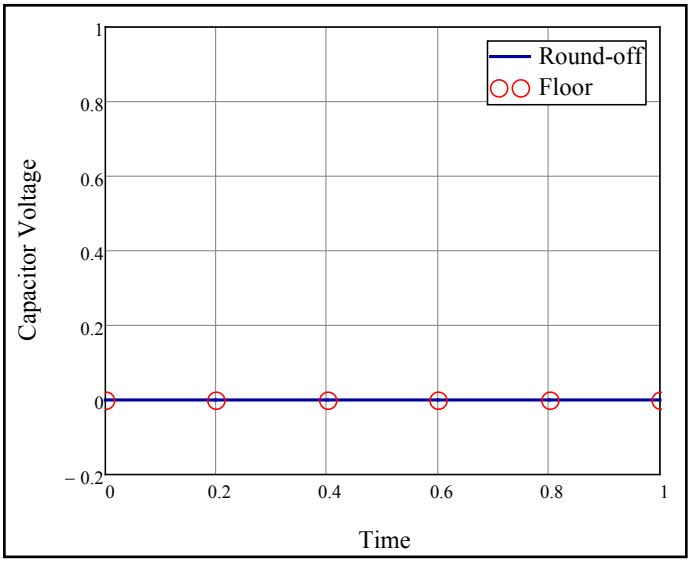


Fig. 7(b): Trapezoidal Euler Algorithm for Δt =0.2

Considering the same example we get the maximum value of $\Delta t = 0.2$ after which the trapezoidal algorithm (nothing but an average of forward and backward Euler methods) will fall though (fig. 7(b)). For the particular value of Δt (0.1 for backward and 0.2 for trapezoidal) the effect of round-off or truncation error is nil as because for this particular value of Δt the system becomes unstable. Simulation results confirm the theoretical results.

VIII. Conclusion

However the result, as depicted in fig. 5 and fig. 6, does reflect the performances of the three methods. It is further to be noted that here in our representation of the computation results we do not mean the usual meaning of the words "truncation" and "round-off" errors. We have illustrated the effect of repeating the computation cycle with finite number of digits. Here we tried to observe the effect of the meaning of these terms as outlined in section II. We have truncated or rounded the digits during each cycle of the calculation process. The paper does not incorporate much new results. However, Section VII, presents a new insight into the algorithm error implement of backward Euler and trapezoidal methods. Methods of numerical computation have been revisited for the benefit of the student community. Enough scope has been left to whet students' appetite for a subsequent in-depth study on this subject. We have emphasized the need of choosing the time step in relation to the system parameters and not independently.

IX. Acknowledgement

The authors are thankful to the SKFGI Management and in particular to Mr Bijoy Guha Mallick, Chairman for providing all the assistances for carrying out this tutorial work. This work has been carried out at the Sir J C Bose Creativity Centre of SKFGI.

References

[1] Euler H.,"Institutiones calculi integralis", Volumen Primum (1768), Opera Omnia, Vol. XI, B. G. Teubneri Lipsiae et Berolini MCMXIII.

[2] Runge C.,"Ueber die numerische Auflösung von differenti algleichungen", Math Ann 46, 1895.

[3] Hull T. E. et al,"Comparing numerical methods for ordinary differential equations", SIAM J Numer Anal 9, 1972.

[4] Butcher J. C.,"General linera methods: A survey", Appl Numer Math 1, 1985.

[5] Fatunla S. O.,"Numerical methods for initial value problems in ordinary differential equations", Academy Press Inc. (London) Ltd.

[6] Kendall E. A.,"An introduction to numerical analysis (second edition)", John Wiley and Sons, 1989.

[7] Lambert J. D.,"Numerical methods for ordinary differential equations", John Wiley and Sons, USA, 1991.

[8] Julyan E. H. C., Piro O.,"The dynamics of Runge-Kutta methods", Int'l J Bifur and Chaos 2, 1992.

[9] Butcher, J. C., Wanner, G.,"Ruge-Kutta methods: some historical notes", Appl Numer Math 22, 1996.

[10] Abraham O.,"A fifth-order six stage explicit Runge-Kutta method for the solution of initial value problems", M. Tech Dissertation, Federal University of Technology, Minna, Nigeria, 2004.

[11] Abraham O.,"Improving the modified Euler method", Leonardo J. Sci, 10, pp. 1-8, 2007.

[12] Rattenbury N.,"Almost Runge-Kutta methods for stiff and non-stiff problems, Ph.D dissertation", The University of Auckland, New Zealand, 2005.