Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

Study of Numerical Solutions of Stiff Differential Equations Using Rk Method and Adaptive Stepsize Control Method

Nazneen Parveen[1], Najmuddin Ahmad*[2]

Department of Mathematics and Statistics, Integral University, Lucknow-226026 (UP), India.

Article History:

Received: 14-01-2025

Revised: 15-023-2025

Accepted: 21-03-2025

Abstract: In this paper, chemical kinetics, electrical circuits, and spring-damping systems, stiff differential equations are specialized initial value issues. Numerical methods are needed for accurate computations because most practical stiff systems lack analytical solutions because to their complexity. This work investigates stiffness phenomena and general-purpose techniques to solve stiff differential equations. We examine numerous standard methods, including the Runge-Kutta method, the Adaptive Stepsize Control for Runge-Kutta method, and the EPISODE ODE Solver package, and list their attributes and computing efficiency. Traditional numerical methods like Euler, explicit Runge-Kutta, and Adams-Moulton require minuscule step sizes for great precision. This may cause substantial round-off errors and solution instability. We analyze the efficiency of the Runge-Kutta and EPISODE algorithms' Adaptive Stepsize Control algorithm to overcome these concerns. Comparative evaluations against accurate solutions demonstrate the effectiveness and dependability of these sophisticated approaches to stiff differential equations.

Keywords: Stiff differential equations, Runge-Kutta method, Adaptive Stepsize Control, EPISODE ODE Solver, Numerical methods, Computational efficiency, Stability analysis, Comparative study
Mathematics Subject Classification: 65L04 , 65L06, 65L50

1. Introduction

A few numerical models bring about frameworks of ordinary differential conditions that, albeit hypothetically well-posed, are for all intents and purposes unsolvable utilizing typical numerical methods due to the severe step size constraints imposed by numerical stiffness. Numerically, these conditions are referred to as stiff conditions. They are characterized by the presence of transient components that are negligible compared to other components of the solution. This characteristic of these conditions restricts the scope of the step size in traditional numerical methods to be on the same scale as the smallest constant in the problem. When faced with several first-order differential conditions, there is a possibility of encountering a stiff arrangement of conditions (Curtiss &Hirschfelder, 1952). Stiffness arises when there are significant variations in the sizes of the independent variable, causing the dependent variables to undergo significant changes (Ahmad, 2016). Fortunately, it is often possible to anticipate the presence of challenging conditions based on the underlying physical problem from which they are generated (Curtiss &Hirschfelder, 1952). By exercising caution, it is possible to manage and minimize errors associated with these conditions (Ahmad & Singh, 2020).

Stiffness plays a crucial role in the numerical approximation of ordinary differential conditions. The outcome depends on the specific differential condition, the initial conditions, and the interval being examined. The solution to stiff differential conditions includes a component in the form of exp(-At), where A is a significant positive constant. Typically, this is a component of the solution known as the transient solution. The dominant component of the solution is referred to as the steady-state solution. The transient component of a stiff condition decreases rapidly to zero as t grows. However, due to the size of the nth derivative of this term being A exp(-At), the derivative does not decrease as quickly (Dahlquist, 1963; Ahmad, 2018). In fact, since the derivative in the error term is evaluated at a value between zero and t, rather than at t itself, the derivative terms may increase as t increases, and do so rather rapidly (Ahmad, 2017). When dealing with conditions of this kind, stability demands require the use of several minute time intervals. This occurs when we encounter a set of interconnected differential conditions that exhibit at least two significantly divergent scales of the independent variable being integrated (Dahlquist, 1963). For instance, we might assume that our reaction is composed of two exponential decay curves, one displaying a rapid decline and the other showing a moderate decay. Except for a few time steps deviating from the initial condition, the slowly decaying curve prevails since the rapid curve has already decayed (Ahmad, 2017). However, because of the variable time step technique required to ensure stability for both components, we must limit ourselves to small time advances, even if the dominant component might allow for much larger time steps (Ahmad, 2020).

Curtiss and Hirschfelder (1952) proposed one of the earliest approaches to address the challenges posed by stiffness, which they discovered during their investigations into energy. They proposed special multistep conditions that could provide satisfactory approximations. The numerical analysts generally ignored the challenge of stiffness until Dahlquist (1963) identified numerical instability as the root cause and offered fundamental definitions and concepts that have greatly aided future research. Dahlquist (1963) proposed that the trapezoidal rule with extrapolation is an appropriate method for solving stiff conditions. Since the publication of Dahlquist's study, there has been a significant degree of activity in the field, with various novel techniques being developed for the numerical solution of stiff conditions (Rizvi et al., 2021). Some commonly used methods for solving stiff conditions include: variable-order methods that utilize backward differentiation multistep equations, which were initially analyzed and implemented by Gear (1969, 1971) and later modified and studied by Hindmarsh (1974) and Byrne and Hindmarsh (1975). Another approach is using methods based on second derivative multistep equations, such as those developed by Wait and Willoughby (1967) and Enright (1974) (Ahmad, 2016; Hasan, 2015). Additionally, there are various techniques that use the trapezoidal rule, such as the methods introduced by Dahlquist (1963) and further examined by Lindberg (1971, 1972). There are also implicit Runge-Kutta methods that are specifically designed for addressing stiff conditions, which rely on the formulae developed by Butcher (1964) and investigated by Ehle (1968). Furthermore, there are several approaches that rely on preliminary numerical transformations to eliminate stiffness, followed by solving the transformed problem using traditional methods, as explored and utilized by Lawson and Ehle (1972). Unfortunately, despite the presence of many proven approaches and several proposed fundamental formulae for stiff conditions, there has been a lack of guidance or assistance to help practitioners in selecting an appropriate solution for the given situation until recently (Ahmad, 2016).

This work focuses on the behaviour of the solution of stiff initial value problems using two alternative methodologies. Initially, the traditional Runge-Kutta procedure is used, followed by the implementation of the Adaptive Step-Size Runge-Kutta method to obtain improved results (Ahmad, 2015). In addition to the Runge-Kutta approach, the EPISODE Tribute solver program is used to obtain a new set of solutions. The solution sets are compared together to evaluate the effectiveness of different approaches. The challenges encountered in addressing stiff initial value problems using the aforementioned approaches have been identified, and a concise description of how to address them is provided (Ahmad, 2017; Rizvi et al., 2021).

2. Computational Details

The review centers around three specific strategies: the fourth request Runge-Kutta procedure for frameworks, the versatile stepsize control for Runge-Kutta, and an all inclusive Tribute bundle called EPISODE. Allow us to look at the underlying worth issue.

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

and a given initial condition, $y(t_0) = a$.

Runge-Kutta Method: The Runge-Kutta strategy, named after the German mathematicians Runge and Kutta, is a kind of mathematical procedure utilized for settling differential conditions. It is alluded to as a solitary step approach since it just depends on the data from the past step that was determined. This approach effectively tackles a differential condition by approximating the exact response by means of matching the principal n terms of the Taylor series extension. We will just look at the fourth request Runge-Kutta strategy, which is a technique for higher request. Fourth-request Runge-Kutta strategies are the most frequently involved and reliable methods for deciding arrangements of conventional differential conditions with higher orders.

$$k_1 = hf(t_n, y_n)$$
$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$
$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$
$$k_4 = hf(t_n + h, y_n + k_3)$$
$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Adaptive Stepsize Control for Runge-Kutta: A compelling Tribute integrator ought to have the capacity to independently change its step size to guarantee versatile command over its encouraging. The essential goal of executing this versatile step size is to accomplish a pre-laid out degree of accuracy in the arrangement while limiting handling assets. To execute versatile step size the board, the calculation should give data on its presentation, specifically a gauge of its truncation blunder.

$$\text{Error estimate } e_n = |\hat{y}_{n+1} - \tilde{y}_{n+1}|$$
$$h_{\text{new}} = h\left(\frac{\epsilon}{e_n}\right)^{\frac{1}{p}}$$

Solver Package EPISODE: In specific fields, particularly in compound applications, one regularly experiences frameworks of normal differential conditions that are numerically all around adapted however very testing to tackle utilizing customary mathematical strategies because of the severe step size restriction forced by mathematical soundness. Over the most recent thirty years, there has been prominent progression in the making of mathematical answers for firm standard differential conditions (Tributes). Subsequently, a few exceptionally viable and reliable answers for standard differential conditions (Tributes) have been made, including DIFSUB, Stuff, LSODE, EPISODE, VODE, LSODPK, and VODPK.

$$y_{n+1} = y_n + \frac{h}{2}[f(t_{n+1}, y_{n+1}) + f(t_n, y_n)]$$

$$\sum_{i=0}^{k} \alpha_i y_{n-i} = h \sum_{i=0}^{k} \beta_i f(t_{n-i}, y_{n-i})$$

The EPISODE program is an assortment of FORTRAN subroutines intended to settle issues with little exertion, even within the sight of potential difficulties consequently. The program consolidates a flexible Adams approach, which is great for nonstiff issues, as well as an adaptable in reverse separation equation (BDF), which is great for firm issues. The two methodologies are of a verifiable multistep nature. While handling testing undertakings, the product broadly depends on the N × N Jacobian grid.

$$J = \frac{\partial \mathbf{f}}{\partial \mathbf{y}} = \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{y}_j}\right)_{i,j=1}^{N}$$

The vector parts of f and y are addressed by f_i and y_j, separately.

A complete examination of the utilization of EPISODE is given in reference [9]. By the by, it is important to give a couple of crucial boundary definitions to delineate the occurrences. Notwithstanding the issue explanation, which is shown as a visual cue 1 and perhaps model 2, the essential information boundary for EPISODE is the strategy banner, MF. The accompanying qualities are incorporated: 10, 11, 12, 13, 20, 21, 22, and 23. The main digit of MF, alluded to as METH, signifies the two central ways to deal with be utilized, in particular understood Adams and BDF. The subsequent number, alluded to as Miter, indicates the iterative arrangement strategy for the implied conditions got from the chosen equation. The boundary Miter might be allotted one of four qualities (0, 1, 2, 3) to address the accompanying implications:

- Utilitarian cycle, otherwise called fixed-point emphasis, is a strategy that doesn't include the use of a Jacobian grid.
- The harmony procedure, otherwise called the summed up Newton technique or semi-fixed Newton emphasis, utilizes a client provided subroutine to compute the Jacobian.
- A harmony approach utilizing inside delivered Jacobian by means of limited contrasts.
- A harmony approach utilizing an inclining guess to the Jacobian, which is inside made. This technique brings about lower capacity and computational expenses, however its viability is decreased.

To utilize the EPISODE bundle, one should conjure the EPSODE driver capability, which from there on summons a few different methodology inside the bundle to address what is going on within reach. The capability f is sent through a subroutine called DIFFUN, which the client is expected to make. Furthermore, a technique for computing the Jacobian, named PEDERV, must be carried out. EPSODE is called iteratively, with each call comparing to one of the client's result focuses. The expected result esteem is determined by doling out a worth of t to the boundary Promote in the EPSODE capability. At the point when the predefined worth of Promote is reached, control is gotten back to the guest program, offering the benefit of y at t = Promote. The EPSODE capability has an extra info named File, which is utilized to show assuming that the ongoing call is the first for the issue. This contention is utilized to decide if certain factors ought to be initialised. It is likewise utilized as a result boundary to show the achievement or disappointment of the bundle in executing the ideal work. The sort of the blunder control done inside not set in stone by two extra info boundaries: EPS and IERROR.

The EPISODE bundle has eight FORTRAN subroutines that should be incorporated with the client's calling project and Subroutines DIFFUN and PEDERV. As recently referenced, the client just calls Subroutine EPSODE, while different subroutines are called inside the bundle. The eight bundle methodology might be compactly summed up as follows:

- EPSODE lays out capacity, starts correspondence with the centre integrator, TSTEP, handles mistake returns, and results blunder messages as the need should arise.
- The INTERP capability computes interjected upsides of y (t) at the result focuses picked by the client, utilizing a multistep history information cluster.
- TSTEP executes one cycle of the joining system and handles the administration of nearby mistake, including deciding the step size and request, for that emphasis.
- COSET appoints coefficients that are utilized by TSTEP for both the essential mix step and mistake control.
- Change alters the set of experiences cluster when the request is diminished.
- PSET initialises the network p by taking away the result of h, β, and J from the character framework I. It then, at that point, registers P to address following direct logarithmic frameworks, where P is the coefficient grid.
- The DEC capability plays out a lower-upper three-sided (LU) disintegration of a grid. The element of the lattice is N × N. SOL is utilized to take care of direct logarithmic issues that have been considered utilizing DEC.

The subroutine EPSODE, which depends on the variable coefficient in reverse separation recipe, might be utilized. The nonstiff choice utilizes an Adams-Bashforth indicator and an Adams-Moulton corrector.

$$y_{n+1} = y_n + h\sum_{i=1}^{k} \beta_i y'_{n+1-i}\&$$

The order may vary from one to seven.

3. **Numerical Implementation and Discussion**: To think about the approaches examined above, we will investigate the accompanying starting worth issue framework. This issue has been intentionally browsed Weight [2] in light of the fact that the exact response is open for correlation.

$$u_1' = 9u_1 + 24u_2 + 5\cos\, t - \frac{1}{3}\sin\, t, \qquad u_1(0) = \frac{4}{3}$$

$$u_2' = -24u_1 - 51u_2 - 95\cos\, t + \frac{1}{3}\sin\, t, \quad u_2(0) = \frac{2}{3}$$

has the unique solution

$$u_1(t) = 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos\, t$$

$$u_2(t) = -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos\, t$$

The presence of the dramatic part e-39t in the arrangement brings about the framework being delegated firm. The issue has been effectively settled by utilizing the Runge-Kutta strategy, Versatile Stepsize Control for Runge-Kutta, and a Tribute Solver bundle called EPISODE. The outcomes are introduced in a plain organization. What's more, we have led a correlation between our discoveries and the exact qualities.

We have assessed the step size upsides of h=0.1, 0.5, 0.01, and 0.001 utilizing the proper step size Runge-Kutta calculation. By utilizing a stage size of h=0.1, we have gotten a devastating result as displayed in Figure 1. This result fundamentally veers off us from the arrangement bend.

Table 1: Runge-Kutta Method for H=0.1

t	Approximated value of u1(t)	Exact value of u1(t)	Approximated value of u2(t)	Exact value of u2(t)
0.000	1.3333333333333	1.3333333333333	0.6666666666667	0.6666666666667
0.100	-2.6451816939060	1.7930625550154	7.8445430187225	-1.0320024225810
0.200	-18.4516909266633	1.4239024050657	38.8765909278864	-0.8746810328574
0.300	-87.4732736758291	1.1315765076212	176.4848024413650	-0.7249985392877
0.400	-394.0775832568734	0.9094085674642	789.3658355982662	-0.6082142119572
0.500	-1760.0500273821960	0.7387878279152	3521.0619840365300	-0.5156576807036
0.600	-7848.7060676766740	0.6057096139135	15698.1842790460000	-0.4404108271537
0.700	-34990.4606015940000	0.4998602647830	69981.5471720920000	-0.3774038285905
0.800	-155983.6252183370000	0.4136714644918	311967.7496820700000	-0.3229535146958
0.900	-695351.1889572670000	0.3416143376540	1390702.9081460000000	-0.2744088049155
1.000	-3099764.2917920000000	0.2796748877511	6199529.0164400000000	-0.2298878009170

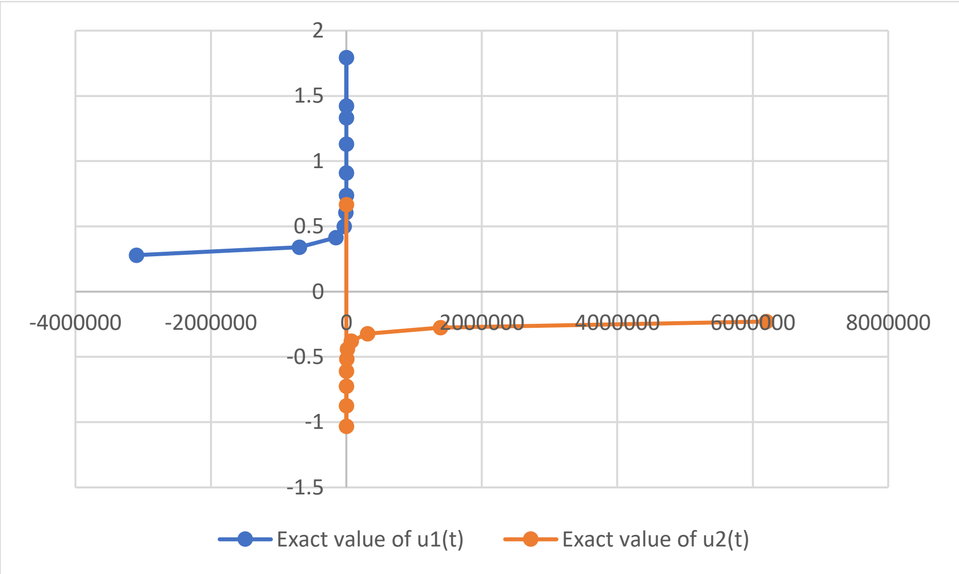

Figure 1: Runge-Kutta Method for H=0.1

To get a more precise gauge contrasted with the past one, we have diminished the step size considerably, specifically to h = 0.05. As an outcome, we have seen an extensive improvement in the outcomes, as displayed in the going with figure. These superior outcomes are exact up to three huge digits.

Table 2: Runge-Kutta Method for H=0.05

t	Approximated value of u1(t)	Exact value of u1(t)	Approximated value of u2(t)	Exact value of u2(t)

0.000	1.3333334000000	1.3333334000000	0.6666667000000	0.6666667000000
0.100	1.7122205164380	1.7930625550154	-0.8703150849322	-1.0320024225810
0.200	1.4140718118310	1.4239024050657	-0.8550150926608	-0.8746810328574
0.300	1.1305256895810	1.1315765076212	-0.7228917836346	-0.7249985392877
0.400	0.9092782234410	0.9094085674642	-0.6079484701226	-0.6082142119572
0.500	0.7387520378960	0.7387878279152	-0.5155814212635	-0.5156576807036
0.600	0.6056845252510	0.6057096139135	-0.4403562134733	-0.4404108271537
0.700	0.4998372925820	0.4998602647830	-0.3773544837041	-0.3774038285905
0.800	0.4136502932950	0.4136714644918	-0.3229082265018	-0.3229535146958
0.900	0.3415951400390	0.3416143376540	-0.2743678955285	-0.2744088049155
1.000	0.2796577590690	0.2796748877511	-0.2298516466851	-0.2298878009170

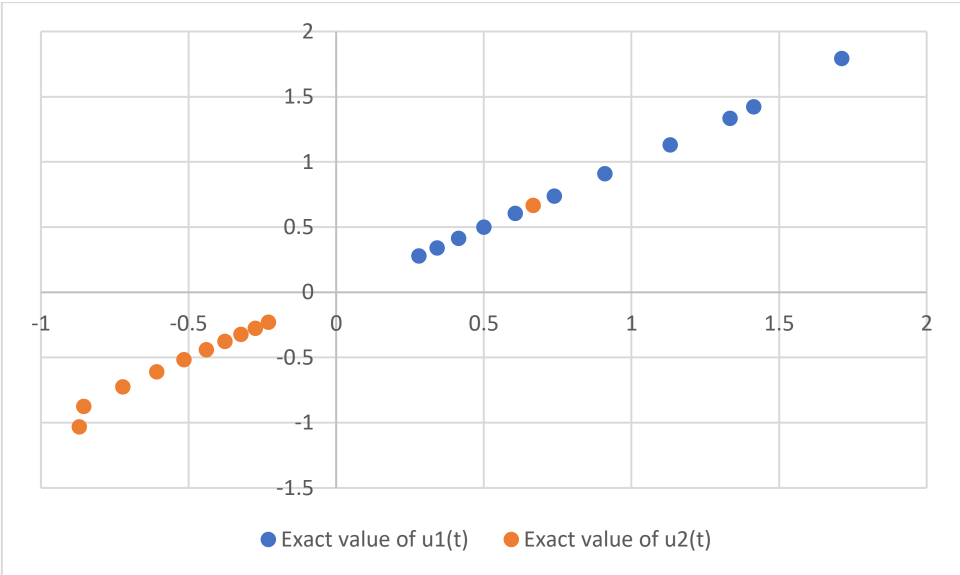

Figure 2: Runge-Kutta Method for H=0.05

To improve our result, we put forth another attempt utilizing a more modest step size (h = 0.01), which is a lot more modest than the past qualities. As a result of this change, the result currently lines up with the exact number, matching it up to 5 huge figures.

Table 3: Runge-Kutta Method for H=0.01

t	Approximated value of u1(t)	Exact value of u1(t)	Approximated value of u2(t)	Exact value of u2(t)
0.000	1.3333330000000	1.3333334000000	0.6666667000000	0.6666667000000
0.100	1.7930413000000	1.7930625550154	-1.0319601339400	-1.0320024225810
0.200	1.4239013000000	1.4239024050657	-0.8746792441463	-0.8746810328574
0.300	1.1315763000000	1.1315765076212	-0.7249984023810	-0.7249985392877
0.400	0.9094084000000	0.9094085674642	-0.6082140611411	-0.6082142119572
0.500	0.7387878000000	0.7387878279152	-0.5156575981525	-0.5156576807036
0.600	0.6057096000000	0.6057096139135	-0.4404107145071	-0.4404108271537

0.700	0.4998603000000	0.4998602647830	-0.3774037966508	-0.3774038285905
0.800	0.4136715000000	0.4136714644918	-0.3229534788065	-0.3229535146958
0.900	0.3416144000000	0.3416143376540	-0.2744087698759	-0.2744088049155
1.000	0.2796749000000	0.2796748877511	-0.2298878000000	-0.2298878009170

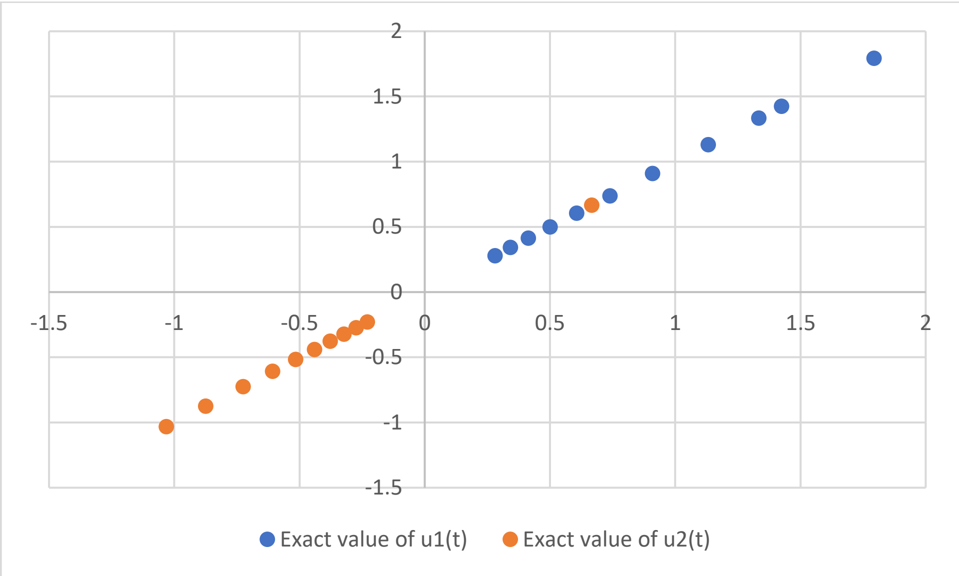

Figure 3: Runge-Kutta Method for H=0.01

By additional diminishing the worth of h to h = 0.001, we have found that our outcome intently matches the exact arrangement of the issue, with a precision of up to 7 decimal spots.

Table 4: Runge-Kutta Method for H=0.001

t	Approximated value of u1(t)	Exact value of u1(t)	Approximated value of u2(t)	Exact value of u2(t)
0.000	1.3333330000000	1.3333334000000	0.6666667000000	0.6666667000000
0.100	1.7930620000000	1.7930625550154	-1.0320022000000	-1.0320024225810
0.200	1.4239023000000	1.4239024050657	-0.8746809000000	-0.8746810328574
0.300	1.1315769000000	1.1315765076212	-0.7249987000000	-0.7249985392877
0.400	0.9094091000000	0.9094085674642	-0.6082144000000	-0.6082142119572
0.500	0.7387882000000	0.7387878279152	-0.5156578000000	-0.5156576807036
0.600	0.6057100000000	0.6057096139135	-0.4404109000000	-0.4404108271537
0.700	0.4998602000000	0.4998602647830	-0.3774038000000	-0.3774038285905
0.800	0.4136715000000	0.4136714644918	-0.3229535000000	-0.3229535146958
0.900	0.3416144000000	0.3416143376540	-0.2744089000000	-0.2744088049155
1.000	0.2796749000000	0.2796748877511	-0.2298878000000	-0.2298878009170

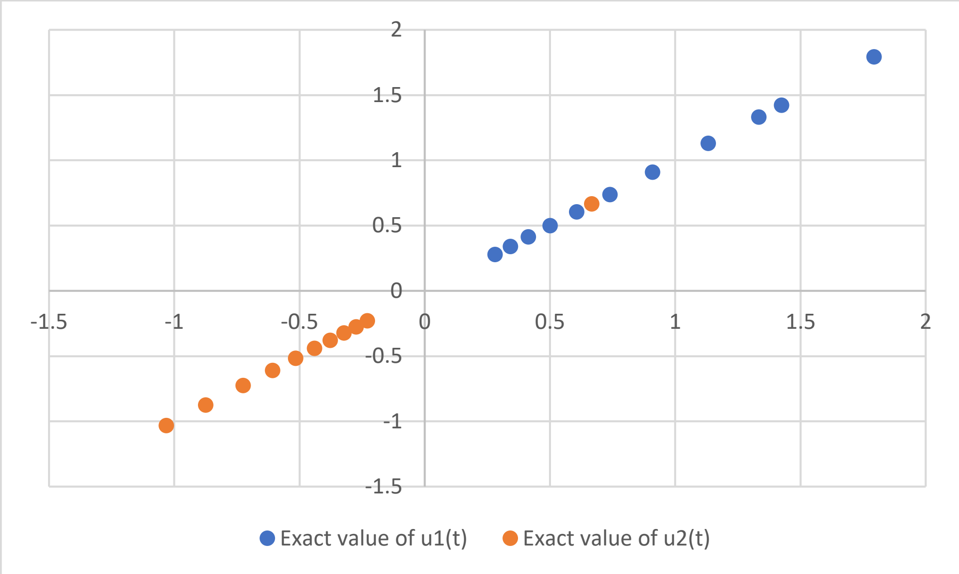

Figure 4: Runge-Kutta Method for H=0.001

While utilizing the decent step size Runge-Kutta procedure, it is important to settle the issue utilizing a pre-decided step size. Be that as it may, this approach conveys the gamble of getting an off base arrangement. Thusly, we can't get the degree of accuracy we look for. In this situation, we have tried different things with many step measures and went on until the last step, which is a tedious cycle. To resolve this issue, we have effectively settled it by utilizing Versatile Step size control for Runge-Kutta. This approach utilizes different step sizes in every cycle to get a foreordained degree of accuracy. The table underneath shows the result of the issue while utilizing Versatile Stepsize control for Runge-Kutta.

Table 5: Adaptive Stepsize Control for Runge-Kutta Method

t	Approximated value of u1(t)	Exact value of u1(t)	Approximated value of u2(t)	Exact value of u2(t)
0.000	1.3333330000000	1.3333334000000	0.6666667000000	0.6666667000000
0.100	1.7930486000000	1.7930625550154	-1.0319745000000	-1.0320024225810
0.200	1.4239021000000	1.4239024050657	-0.8746804000000	-0.8746810328574
0.300	1.1315765000000	1.1315765076212	-0.7249985000000	-0.7249985392877
0.400	0.9094086000000	0.9094085674642	-0.6082142000000	-0.6082142119572
0.500	0.7387878000000	0.7387878279152	-0.5156577000000	-0.5156576807036
0.600	0.6057096000000	0.6057096139135	-0.4404108000000	-0.4404107272201
0.700	0.4998602000000	0.4998602647830	-0.3774038000000	-0.3774038285905
0.800	0.4136715000000	0.4136714644918	-0.3229535000000	-0.3229535146958
0.900	0.3416143000000	0.3416143376540	-0.2744088000000	-0.2744088049155
1.000	0.2796749000000	0.2796748877511	-0.2298878000000	-0.2298878009170

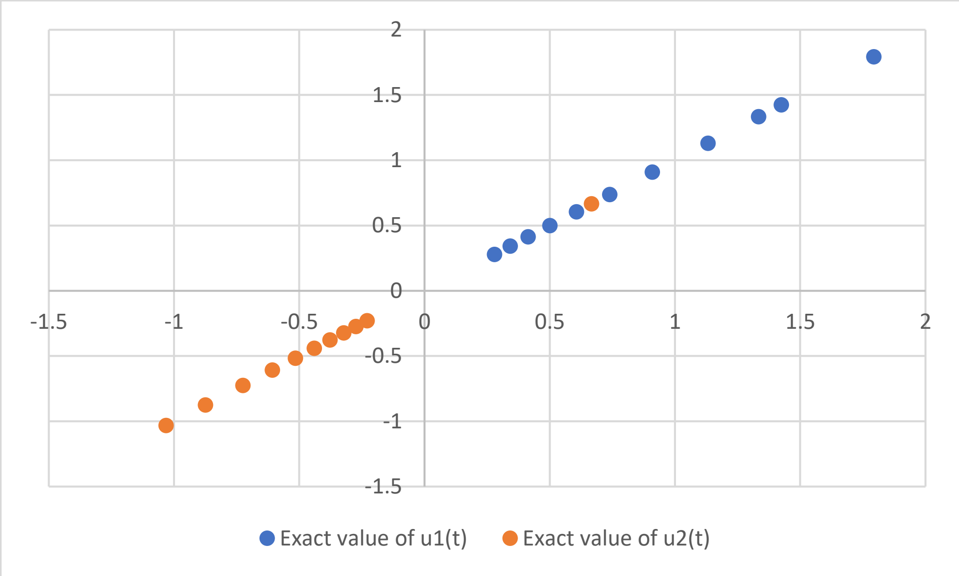

Figure 5: Adaptive Stepsize Control for Runge-Kutta

Because of the severe step size restriction forced by mathematical soundness, traditional mathematical procedures are frequently unfit to address most genuine Solid starting worth issues. Therefore, numerous extremely productive Tribute solvers have been made. To approve our discoveries, we utilized the EPISODE bundle, a nonexclusive solver for common differential conditions (Tributes). The result of our examination is compactly summed up in the table underneath.

Table 6: Episode

t	H	Approximated Value of u1(t)	Exact value of u1(t)	Approximated Value of u2(t)	Exact value of u2(t)
0.0	0.40E-02	1.3333333400000	1.3333334000000	0.6666666700000	0.6666667000000
0.1	0.40E-02	1.7930620000000	1.7930629999985	-1.0320020000000	-1.0320020000000
0.2	0.91E-02	1.4239025000000	1.4239020000000	-0.8746815000000	-0.8746810000000
0.3	0.12E-01	1.1315765000000	1.1315770000000	-0.7249982000000	-0.7249986000000
0.4	0.33E-01	0.9094085000000	0.9094086000000	-0.6082140000000	-0.6082142000000
0.5	0.33E-01	0.7387880000000	0.7387878000000	-0.5156578000000	-0.5156577000000
0.6	0.45E-01	0.6057103000000	0.6057097000000	-0.4404108000000	-0.4404108000000
0.7	0.66E-01	0.4998615000000	0.4998603000000	-0.3774045000000	-0.3774038000000
0.8	0.66E-01	0.4136745000000	0.4136715000000	-0.3229545000000	-0.3229535000000
0.9	0.66E-01	0.3416192000000	0.3416143000000	-0.2744110000000	-0.2744088000000
1.0	0.66E-01	0.2796809000000	0.2796749000000	-0.2298908000000	-0.2298878000000

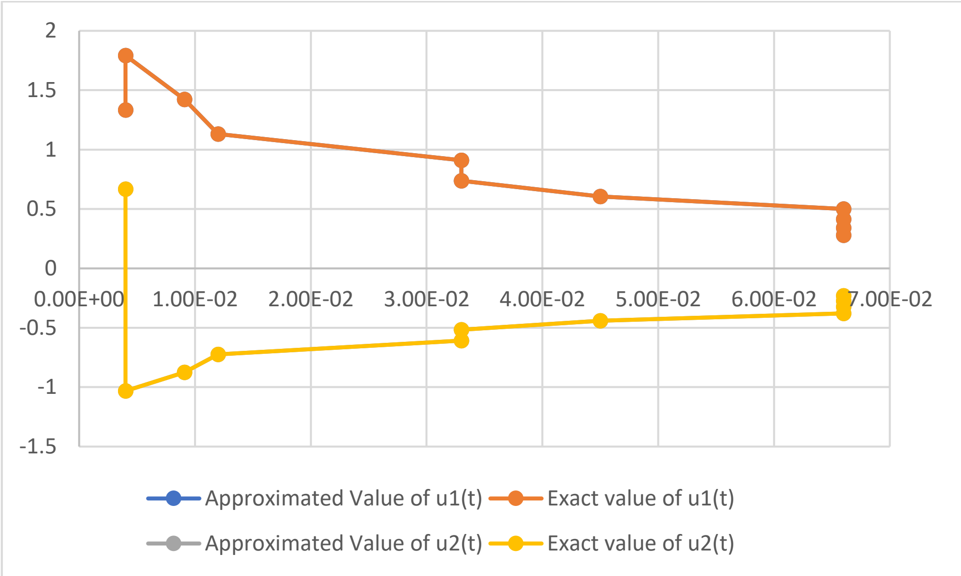

Figure 6: Episode

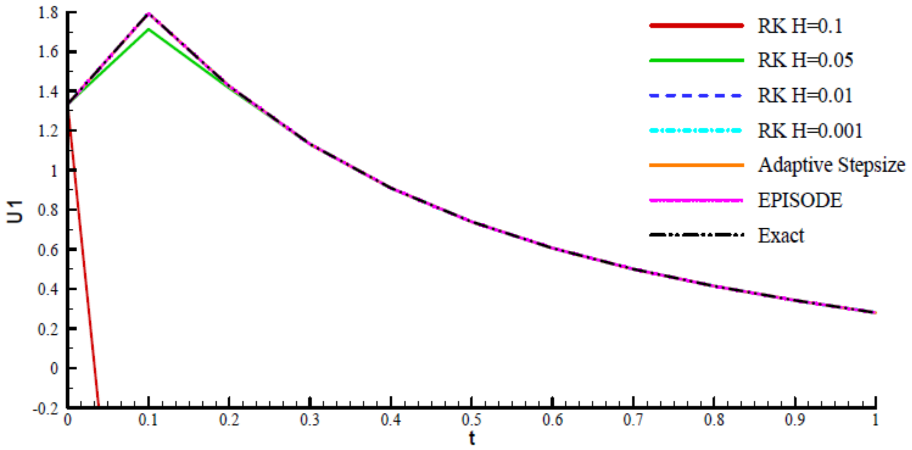

Figure 7: The graph of the solutions for u_1

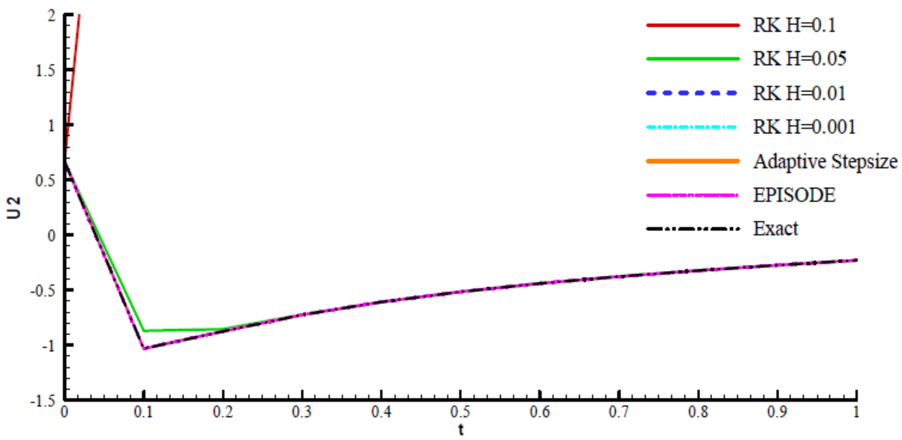

Figure 8: The graph of the solutions for u_2

In view of the information displayed in figures (7) and (8), it tends to be reasoned that all arrangements, aside from the Runge-Kutta procedure with H = 0.1 and the Runge-Kutta technique with H = 0.05, are reliable with the exact arrangement.

4. Conclusion

In spite of the fact that there are a few basic formulae and productive projects for settling such issues, there has been an absence of counsel or help to help clients in choosing a proper technique for handling firm starting worth issues. This work explicitly inspects the Runge-Kutta procedure, Versatile Stepsize Control for Runge-Kutta, and EPISODE, a Tribute Solver bundle. A critical end arrived at all through the discussion is that the decent step size Runge-Kutta strategy is unsatisfactory for tackling solid differential conditions. While utilizing the proper step size Runge-Kutta strategy, we can see that as the step size diminishes, the exactness of the guess increments. The client is as yet stood up to with the issue of deciding the suitable step size to accomplish the necessary degree of accuracy. On the other hand, the Versatile step size Runge-Kutta procedure empowers us to indicate the ideal level of precision by changing the step size as per different upsides of the autonomous variable. EPISODE, then again, is a solver program that offers a more significant level of ease of use and conveys a good estimation, though its precision isn't actually that high of the Versatile Step Size Runge-Kutta method. Notwithstanding, EPISODE is especially useful at tending to genuine issues that include immense and complex frameworks.

Acknowledgment: IU/R&D/2024-MCN0002940 office of research and development Integral University, Lucknow.

References

[1] Atkinson KE. An Introduction to Numerical Analysis. John Wiley & Sons; 1989.
[2] Burden RL, Faires JD. Numerical Analysis. PWS Publishers; 2002.
[3] Byrne GD, Hindmarsh AC. Stiff ODE Solvers: A Review of Current and Coming Attractions. J Comput Phys. 1986;70(1):1-62.
[4] Byrne GD, Hindmarsh AC. A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations. ACM Trans Math Softw. 1975;1:71-96.
[5] Enright WH, Hull TE. Comparing Numerical Methods for the Solution of Stiff Systems of ODEs Arising in Chemistry. In: Numerical Methods. Academic Press; 1976.
[6] Gear CW. The Automatic Integration of Stiff Ordinary Differential Equations. In: Morrell AJH, editor. Information Processing 68. North Holland; 1969. p. 187-193.
[7] Gear CW. Algorithm 407: DIFSUB for Solution of Ordinary Differential Equations. Commun ACM. 1971;14:185-190.
[8] Gerald C, Wheatley OP. Applied Numerical Analysis. Pearson Education Inc; 1999.
[9] Hindmarsh AC, Byrne GD. EPISODE: An Experimental Package for the Integration of Systems of Ordinary Differential Equations. LLL Report UCID-30112; 1975. Available from: www.netlib.org/ode/epsode.f
[10] Hindmarsh AC, Gear CW. Ordinary Differential Equation System Solver. LLL Report UCID-30001, rev. 3; 1974. Available from: www.netlib.org/ode/epsode.f
[11] Lapidus L, Schiesser WE. Numerical Methods for Differential Systems. Academic Press; 1976.
[12] Nejad LA. A Comparison of Stiff ODE Solvers for Astrochemical Kinetics Problems. Astrophys Space Sci. 2005;299:1-29.
[13] Shampine LF, Gear CW. A User's View of Solving Stiff Ordinary Differential Equations. SIAM Rev. 1979;21:1-17.

[14] Charan S, Ahmad N. Study of Numerical Solution of Fourth Order Ordinary Differential Equations by Fifth Order Runge-Kutta Method. Int J Sci Res Sci Eng Technol. 2018;4(3):1-5.
[15] Rizvi STR, Seadawy AR, Younis M, Ahmad N, Zaman S. Optical Dromions for Perturbed Fractional Nonlinear Schrödinger Equation with Conformable Derivatives. Opt Quant Electron. 2021;53(8):477. https://doi.org/10.1007/s11082-021-03047-w
[16] Ahmad N, Deeba KF. The Study of New Approaches in Cubic Spline Interpolation for Automobile Data. J Sci Arts. 2017;40(3):401-406.
[17] Ahmad N, Deeba KF. Study of Numerical Accuracy in Different Spline Interpolation Techniques. Glob J Pure Appl Math. 2020;16(5):687-693.
[18] Ahmad N, Singh B. Numerical Solution of Integral Equation Using Galerkin Method with Hermite, Chebyshev & Orthogonal Polynomials. J Sci Arts. 2020;50(1):35-42.
[19] Singh VP, Ahmad N. Some New Three-Step Iterative Methods for Solving Nonlinear Equation Using Steffensen's and Halley Method. Br J Math Comput Sci. 2016;19(2):1-9.
[20] Charan S, Ahmad N. Numerical Accuracy Between Runge-Kutta Fehlberg Method and Adams-Bashforth Method for First Order Ordinary Differential Equations with Boundary Value. J Math Comput Sci. 2016;6(6):1145-1156.
[21] Charan S, Singh VP, Ahmad N. Study of Numerical Accuracy of Runge-Kutta Second, Third, and Fourth Order Method. Int J Comput Math Sci. 2015;4(6):111-118.
[22] Ahmad N, Hasan A. Comparative Study of a New Iterative Method with That of Newton's Method for Solving Algebraic and Transcendental Equations. Int J Comput Math Sci. 2015;4(3):32-37.
[23] Charan S, Ahmad N. A Comparative Study on Numerical Solution of Ordinary Differential Equations by Different Methods with Initial Value Problem. Int J Recent Sci Res. 2017;8(10):21134-2139.
[24] Dahlquist G. A Special Stability Problem for Linear Multistep Methods. BIT Numer Math. 1963;3(1):27-43. https://doi.org/10.1007/BF01963532
[25] Gear CW. Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall; 1971.
[26] Hairer E, Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer; 1996.
[27] Shampine LF, Reichelt MW. The MATLAB ODE Suite. SIAM J Sci Comput. 1997;18(1):1-22. https://doi.org/10.1137/S1064827594276424
[28] Hindmarsh AC. ODEPACK, a Systematized Collection of ODE Solvers. Sci Comput. 1983;55-64. https://doi.org/10.1007/978-94-009-7362-4_6
[29] Butcher JC. The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. John Wiley & Sons; 1987.
[30] Cash JR. Modified Extended Backward Differentiation Formulae for the Numerical Solution of Stiff Initial Value Problems. J Comput Appl Math. 2000;125(1-2):41-49. https://doi.org/10.1016/S0377-0427(00)00432-8
[31] Enright WH, Pryce JD. Two-Derivative Runge-Kutta Methods for Stiff Differential Equations. SIAM J Numer Anal. 1987;24(2):433-448. https://doi.org/10.1137/0724030
[32] Curtiss CF, Hirschfelder JO. Integration of Stiff Equations. Proc Natl Acad Sci. 1952;38(3):235-243. https://doi.org/10.1073/pnas.38.3.235
[33] Ehle BL. High Order A-Stable Methods for the Numerical Solution of Stiff Systems of Ordinary Differential Equations. BIT Numer Math. 1968;8(4):276-291. https://doi.org/10.1007/BF01930862
[34] Gear CW. The Automatic Integration of Stiff Ordinary Differential Equations. In: Information Processing 69. North Holland; 1969. p. 187-193. https://doi.org/10.1007/978-3-642-46841-5_2

[35] Deuflhard P, Bornemann F. Scientific Computing with Ordinary Differential Equations. Springer; 2002. https://doi.org/10.1007/978-1-4757-4318-3

[36] Hundsdorfer W, Verwer JG. Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer; 2003. https://doi.org/10.1007/978-3-662-09017-6

[37] Shampine LF, Gordon MK. Computer Solution of Ordinary Differential Equations: The Initial Value Problem. W. H. Freeman; 1975.

[38] Lawson JD, Ehle BL. Consistency of Predictor-Corrector Methods for Stiff Equations. BIT Numer Math. 1972;12(2):99-110. https://doi.org/10.1007/BF01934429