Full Length Research Paper

Implementation of a new 4th order runge kutta formula for solving initial value problems (I.V.Ps)

G. U. Agbeboh¹, U .S. U Aashikpelokhai¹ and I. Aigbedion²*

¹Mathematics Department, Ambrose Alli University, Ekpoma, Nigeria.
²Physics Department, Ambrose Alli University, Ekpoma, Nigeria.

Accepted 8 March, 2007

A new method for solving singular initial value problems in ordinary differential equations is developed and implemented using the test problem in (1). Results generated through a FORTRAN program were found to be highly accurate and consistent with minima errors in the solution of some selected singular ivps. A comparison of the results generated from the formula was carried out with other Runge Kutta formulae and were found to compare favourably well.

Keywords: Discontinuity, Singularities, Runge- Kutta, Initial Value Problem (ivp).

INTRODUCTION

The initial value problem represented by

\[ y' = f(x, y), \ y(x_o) = y_o, \quad a \leq x \leq b \quad (1) \]

has solution function \( y' \in [a, b] \rightarrow \mathbb{R} \) or the gradient function \( f(x, y) \) may have points of discontinuities at some points in differential systems which are referred to as points of singularities. The solutions to these kinds of problems are often very difficult using analytical methods. Where they exist, they merely estimate the actual results. This contention led us to derive the new method which users will find handy.

Existing integrators and motivation

Many methods exist for the solution of IVPs in differential equations. According to Butcher (1987), it is a know fact that not all such methods have the capacity to find solution to these IVPs. This led us to search and developed some one-step methods which we believed can provide solution to singular problems. Before designing our formulae, we considered many methods and we were motivated by the striking proposal made by Evans and Sangui (1986), Aashikpelokhai (1991), Fatunla (1980, 1988) to study Runge–Kutta method of order 4.

In real life, problems with singularities abound in physical phenomena such as simulation, control theory, economics analysis and production processes, oil spillage, chemical kinetics, electrical network, tunnel switching, petroleum exploration, population problems, the states of the national economy nuclear reactor control as indicated in the work of Ascher and Mattheij (1988), Edsberg (1988), Fatunla (1980, 1987a and 1988), Kaps (1984), Lee and Preiser (1978), Norelli (1985), Parker (1982) and Robertso (1976) are problem whose differential equations are stiff, singular or oscillatory.

According to Bergamini (1963) our civilization will scarcely exist without the physical laws and intellectual techniques developed as bye-products of mathematical research. Man in his attempt to make the best use of his environment often finds obstacles to contend with. These obstacles come in different ways such as problems of controlling human disasters that arise as a result of break down of either man made laws, or of cosmic laws which can cause discontinuity.

Other areas of discontinuity in real life are Currency fluctuation, gas leakage flowing in an annulus pipe. To solve this problem we may need more sophisticated methods of complex integral formulae such as Cauchy integral formula, Taylor series expansion, Laurent series expansion, residue theory and other analytic approaches. The results from these analytic methods are usually either under estimated or over estimated hence the need for numerical approximation that gives accurate results.
According to Fatunla (1988) the Conventional numerical integrators are, in general, formulated on the basis of polynomial interpolation, with the tacit assumption that the IPV in (1) satisfies the hypothesis of the existence and uniqueness theorem.

Consequently such algorithms perform poorly when they are applied to IVPs that violate the hypothesis of the said theorem. Therefore, the conventional numerical integrators are thus inefficient, as they merely track the solution which explodes in the neighborhood of this singularity, as in the case of the IVP in (2) which violates the hypothesis. Here it’s Lipschitz constant \( L \) or \( f \) is unbounded near \( x = \frac{\pi}{4} \). Fatunla (1988) further said that the problem is compounded by the fact that there maybe no clue as to the location of these singularities, particularly if \( f(x,y) \) is non-linear. Such problems in (3) and their like can be handled by the one-step scheme, by ensuring that the point of discontinuity is a mesh point. The subroutine that evaluates the derivatives contains a switch that alerts the integrators when a discontinuity is to be over stepped. By applying these three steps, we can achieve a lot.

(i) detecting a discontinuity, (ii) locating the point of discontinuity, (iii) Restarting the integration beyond the point of discontinuity.

Carver (1977) in the code FORSIM provided options to locate discontinuity; Gear and Osterby (1984) investigated the problem for non stiff problems, while Hindmarch (1980) incorporated a root finder in the code LSODAR.

Gear (1980d) proposed a suitable Runge-Kutta Step to generate enough information for a four-step multi-step scheme to continue accurately beyond discontinuity. Amongst other existing algorithms designed for singular/discontinuous IVPs are:


b) Perturbed polynomial techniques Lambert and Shaw (1966), Shaw (1967).


The main focus of this work is to implement the integrator to generate improved results for the IVP in (1). Base on this formula, we developed an algorithm (AGU) which will be used to solve problem like:

\[
\begin{align*}
(1) & \quad y' = 1 + y^2, \quad y(0) = 1 \quad \forall \quad 0 \leq x \leq \frac{\pi}{4}, \\
(2) & \quad y' = -y, \quad y(0) = 2, \\
(3) & \quad y' = y, \quad y(0) = 1 \\
(4) & \quad -10(y - 1)^2, \quad y(0) = 2, \quad 0 \leq x \leq 1
\end{align*}
\]

Results from our methods were compared with other results given by other methods and their differences in error were examined and found to be of high degree of accuracy.

The occurrence of the reaction,

\[
\phi \left( x_n, y_n \right), \phi \left( x_{n+1}, y_{n+1} \right) < 0
\]

implies the existence of singularity in the interval \( x_n < x < x_{n+1} \).

Gear and Osterby (1984) proposed an efficient method (using local error estimators) to detect and locate the point of discontinuity without using the singularity function. They made provision for passing the discontinuity and then restart the integration process. Gear (1980d), Ellison (1981) and Enright et al., (1986) provided Runge-Kutta like formulas that enable an efficient restart of the multi-step algorithm at discontinuities.

This research is concerned with constructing and implementing a modified Runge-Kutta Method of order 4 for solving the type of problems in (1.1). We are interested in Runge-Kutta Integrator of order 4 because of its age, wide spread and portability. It is from a class of the well known infinitely many Runge-Kutta integrators. However, Runge-Kutta Method has its own shortcomings. One of such, which we will address in this research, is that it cannot handle stiff problems effectively. Often they give spurious results. This situation led Luck el ta (1975), Fatunla (1982), Niekerk (1987) and Aashikpelokhai (1991) to derive a different rational integrators for this purpose. We choose a particular rational integrator with \( K=13 \) from Aashikpelokhai (1991). We are attracted to examine this rational integrators with \( K=13 \) because of the good results given so far by the cases \( 1 \leq k \leq 11 \). Other research students are currently working on the case \( k=12 \) and 14. This research is, in part designed to compare the performance of our modified Runge-Kutta method of order 4 with the rational integrator \( k=13 \). At this juncture, we will give the definition of some relevant terms.

However early studies were on the explicit RKM which up till now is not exhausted but has now extended to implicit methods which are now recognized as appropriate for stiff differential equations. Other recent contributions are the work of Butcher (1963), Gill (1951), Merson (1957), Sarafyan (1965), Shintani (1966a) Ralston (1962b, 1965), King (1966), Lawson (1966, 1967a), Conte and Reaves (1956) Blum (1962), Fyfe (1966) and many others who made various contribution in minimizing the error, absolute interval of stability and storage.
reduction.

Butcher (1965) established the following relationship between the stages and the order of explicit Runge-Kutta process:

\[ P(s) = s, 1 \leq s < 4 \]
\[ P(s) = s-1, 5 \leq s \leq 7, \text{ and} \]
\[ P(s) = s-2, 5 \geq 8. \]

Furthermore, Butcher (1963, 1976b), Wanner et al. (1963) and Hairer and Wanner (1981) also use the concept of rooted trees (in graph theoretic sense) to establish the order conditions for all classes of R-K Process as follows.

\[ C(p) : \sum_{j=1}^{s} C_{j}^{k-1} = \frac{C_{k}}{k}, i = 1(1)s, k \leq p \quad (4) \]
\[ D(p) : \sum_{j=1}^{s} C_{j}^{k-1} a_{lj} = \frac{b_{l}(i-C_{j}^{k})}{k}, j = 1(1)s, k \leq p \quad (5) \]
\[ B(p) : \sum_{j=1}^{s} b_{j}C_{j}^{k-1} = \frac{1}{k}, j = 1(1)p \quad (6) \]

However, according to Butcher (1987) a number of different approaches have been used in the analysis of Runge-Kutta methods. This is where our approach and method of analysis of order 4 R-K becomes very relevant. A retrospective outline of different approaches used by various researchers’ shows;

(1) Euler (1768) as show in (1)
(2) Runge and Kutta (1895, 1905) respectively used

\[ y_{n+1} = y_{n} + h\phi(x_{n}, y_{n}, j) \quad (7) \]

With \[ \phi_{r}(x_{n}, y_{n}, h) = \sum_{j=0}^{p-1} \frac{h^{r}}{(r+1)!} f^{(r)}(x_{n}, j) \quad (8) \]

where \( f^{(r)}(x, y(x)), r = 0(1)p - 1 \) denotes the r-th derivative of \( f(x, y(x)) \) and \( \tau \) in (7) represents Taylor series expansion. (8) Fehlberg (1964) (p,q) used pair approach as given by

\[ y_{n+1} = y_{n} + h_{n} \sum_{i=1}^{s} b_{i}y_{i}, \text{ order } q=p+1. \text{ Fatunla (1986), Heun (1900) and a host of other early researcher followed Runge-Kutta foot step to evolve their methods.} \]

Many popular R-K Codes have on several occasions been developed to cope with some particular test equations, c.f Fatunla (1988). Enright and Hull (1976) felt that an embedded pair of formulas of orders 4 and 5 due to Fehlberg (1964) is much more efficient than the classical four – stage-fourth-order R-K formula with doubling or Richardson Extrapolation. Deuflhard (1983) disagreed with this view and argued that it is hard to construct even one good formula of high order, harder to construct a pair. Kaps (1984) asserted that the two procedures are very much the same, and that for other kinds of one-step methods, stability is crucial, and that it may be difficult to construct an embedding pair with both having good stability. Several other good codes have since been developed. In this paper we have developed a FORTRAN code for the new 4th order R-K formula.

We decided to work within this limitation because we quite agreed with what Lambert (1977,1995) said when he acknowledged that in several areas of numerical analysis there was then a feeling that what was needed was greater insight into the working of existing methods, rather than develop new ones.


The Runge-Kutta methods provide a suitable way of numerical solution to ordinary differential equations. Different approaches have been used by many authors in the past. Authors like Butcher (1963) following from the work of Gill (1951) and Merson (1957) developed a method of analysis of a general explicit Runge-Kutta tree Algorithm. Here, we will adopt a new fourth order method developed through Geometric approach, implement it with some tested ivps and compare results with other existing methods. The aim is to reduce the computational rigours involved in the use of classical Runge-Kutta fourth order methods. It is our belief that the new formula will create more interest in the use of RKF methods.

**Derivation of the method**

The new 4th order is carved out of the existing forth order classical Runge Kutta method. The classical Runge Kutta method is based on Arithmetic mean for \( k_{n}, i = 1,2,3,4 \) also called a One-Sixth Runge – Kutta method because it averages out to six components. On the other hand our new fourth order RKF is of Geometric mean for \( k_{n}, i = 1,2,3,4 \) and can be called a One-third Kutta formula because it averages out to three components of the Runge Kutta formula given by

\[ y_{n+1} = y_{n} + h\phi(x_{n}, y_{n}, h) . \quad (9) \]

Where \[ \phi(x_{n}, y_{n}, h) = \sum_{j=1}^{k} C_{j}k_{j} , \quad (9a) \]
\[ k_{j} = f(x_{n}, y_{n}) \quad (9b) \]
\[ k_j = f(x_n + ah, y_n + h \sum_{i=1}^{j-1} b_i k_i) \quad \text{for } j = 2, 3, 4 \ldots R \]  

(10)

and

\[ a_i = \sum_{j=1}^{i-1} b_j \quad \forall \ j = 2, 3, 4 \ldots R. \]  

(10a)

To construct our new formula from the classical 4th order RKM, we recall the process of Arithmetic mean for arbitrary numbers \( a, b, c \) with their common difference of progression being \( b-a, c-b \). Therefore if \( b-a = c-b \), we have

\[ 2b = a + c \Rightarrow b = \frac{a + c}{2} \]  

which represents an arithmetic average giving rise to

\[ y_{n+1} - y_n = \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \Rightarrow y_{n+1} = y_n + \frac{h}{3} \left( k_2 + k_3 + k_4 \right) \]  

(11)

In the same way if the three arbitrary numbers \( a, b, c \) are manipulated in geometric progression, such that \( b \) is called the geometric mean of \( a \) and \( c \) with their common ratio given as \( b/a \) or \( c/b \).

Then,

\[ b/a = c/b \Rightarrow b^2 = ac \Rightarrow b = \sqrt{ac} \quad \text{or} \quad c = \sqrt{ab} \]  

or

\[ a^* = \sqrt{bc} \]  

such that

\[ \alpha_j = \sum_{i=1}^{j} \sqrt{\frac{k_i}{k_j}}, \quad \text{where } j \in 1 \]  

(11a)

so that

\[ \alpha_1 = \sqrt{k_1/k_2}, \quad \alpha_2 = \sqrt{k_2/k_3}, \quad \text{and } \alpha_3 = \sqrt{k_3/k_4}. \]

Rearranging for \( k_i \) as in (9) we have by analogy, our formula

\[ y_{n+1} = y_n + h \left( \frac{1}{3} \sqrt{k_2/k_1} + \frac{1}{3} \sqrt{k_3/k_2} + \frac{1}{3} \sqrt{k_4/k_3} \right). \]  

(12a)

\[ k_1 = f(y_n) \]
\[ k_2 = f(y_n + ha) \]
\[ k_3 = f(y_n + h(a_2 + ax_2)) \]
\[ k_4 = f(y_n + h(a_3 + ax_3)) \]  

(12b)

To evaluate the RHS of (12a), we apply a binomial expansion technique with fractional index

\[ (1 + x)^{1/2} = 1 + \frac{1}{2} x^2 - \frac{1}{8} x^3 + \frac{1}{16} x^4 + \ldots \]  

(13)

We have by setting

\[ \frac{1}{\sqrt{k_2}} = f(1 + x)^{1/2}, x = \frac{k_2 - 1}{f^2} \]

By substituting

\[ x = \frac{k_2 - 1}{f^2} \quad (i = 1, 2, 3) \]  

(13a)

in (12) we obtain in ascending powers of \( h \) the expansion of \( k_i \) as follows. Or we linearize along the \( y \) function by setting \( a_1 = b_{21}, a_2 = b_{32}, a_3 = b_{41}, a_4 = b_{22}, a_5 = b_{32} \) so that

\[ k_i = f(x, y) = f_n \]  

(14a)

\[ k_i = f(x + \frac{h}{2} a_i, y + \frac{h}{2} k_i) \Rightarrow k_i = \sum_{j=1}^{i} \frac{1}{2} \left( \frac{h}{2} a_i \right)^j \frac{\partial^j f(x, y)}{\partial y^j} \]  

(14b)

\[ k_i = k_i + ha k_i f_{,y} + \frac{h^3}{6} a_i k_i f_{,yy} + \frac{h^3}{24} a_i^2 k_i f_{,yyy} + (Oh^4) \]  

(14c)

\[ k_i = \left[ k_i + ha k_i f_{,y} + \frac{h^3}{6} a_i k_i f_{,yy} + \frac{h^3}{24} a_i^2 k_i f_{,yyy} + \frac{h^3}{6} a_i^3 k_i f_{,yyyy} \right] + (Oh^4) \]

(15)

Similarly we obtain the derivatives of \( k_i \) where \( i = 1, 2, 3, 4 \).

\[ \frac{d}{dx} f(x, y) = \frac{d}{dx} f_n + \frac{1}{2} \frac{d}{dx} f_{,y} = \frac{d}{dx} \left( \frac{d}{dx} f_n \right) \]

(18)

\[ \frac{d}{dx} f_{,y} = \frac{d}{dx} f_{,y} + \frac{1}{2} \frac{d}{dx} f_{,yy} = \frac{d}{dx} \left( \frac{d}{dx} f_{,y} \right) \]

(19)

\[ \frac{d}{dx} f_{,yy} = \frac{d}{dx} f_{,yy} + \frac{1}{2} \frac{d}{dx} f_{,yyy} = \frac{d}{dx} \left( \frac{d}{dx} f_{,yy} \right) \]

(20)
\( k_i = k + h(a_i + a_i)k f_i + h' \left[ a_i + a_i \right] k f_i^2 + \frac{h^2}{2} \left[ a_i + a_i \right]^2 k f_i^3 + h' a_i a_i k f_i^2 + \frac{h^2}{2} \left[ a_i + a_i \right] k f_i^3 + \frac{h^2}{6} \left[ a_i + a_i \right]^3 k f_i^3 \) 

\( k_i = k + 2h(a_i + a_i)k f_i + h' \left[ 2a_i + 2a_i \right] k f_i + \frac{h^2}{8} \left[ a_i + a_i \right]^2 k f_i^3 + h' a_i a_i k f_i^2 + \frac{h^2}{2} \left[ a_i + a_i \right] k f_i^3 + \frac{h^2}{6} \left[ a_i + a_i \right]^3 k f_i^3 \) 

Using a binomial expansion strategy given by 

\[(1 + x)^2 = 1 + \frac{1}{2} x^2 - \frac{1}{8} x^3 + ... \]

and with the help of the reduce formula manipulation package,

we write 

\[ \sqrt{k_i k_j} = f (1 + x)^2 \] 

\( \forall i = 1, 2, 3, 4 \)

Such that 

\[ (k_i k_j)^{1/2} = f (1 + x)^{1/2} \Rightarrow \frac{k_i k_j}{f^2} - 1 = x \] 

(18a)

Evaluate Equation (11) with x as given in (11a) and by using equation (9a) we obtain

\[ \sqrt{k_i k_j} = 1 + \frac{1}{2} \left[ \frac{k_i k_j}{f^2} - 1 \right]^2 + \frac{1}{8} \left[ \frac{k_i k_j}{f^2} - 1 \right]^3 + ... \] 

\[ \sqrt{k_i k_j} = 1 + \frac{1}{2} \left[ \frac{k_i k_j}{f^2} - 1 \right]^2 + \frac{1}{8} \left[ \frac{k_i k_j}{f^2} - 1 \right]^3 + \frac{1}{16} \left[ \frac{k_i k_j}{f^2} - 1 \right]^4 + ... \] 

Substituting for \( k_i^r \) \( \forall i = 1, 2, 3, ... n \) in their respective powers, we have by using the functional derivative with respective to y only, that

\[ \sqrt{k_i k_j} = 1 + \frac{1}{2} \left[ \frac{k_i k_j}{f^2} - 1 \right]^2 + \frac{1}{8} \left[ \frac{k_i k_j}{f^2} - 1 \right]^3 + \frac{1}{16} \left[ \frac{k_i k_j}{f^2} - 1 \right]^4 + ... \]
Substituting in 

\[ y_{\nu+1} = y_{\nu} + \frac{h}{3} \left( \sqrt{k_1 k_2} + \sqrt{k_2 k_3} + \sqrt{k_3 k_4} \right) \]

summarizing, we have,

\[ y_{\nu+1} - y_{\nu} = h \left[ \frac{2a_i^3 + 2a_i a_j + 4a_i a_j}{6} \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \right] k_i f_m \]

\[ + \frac{h^3}{3} \left\{ \frac{2a_i^3}{3} + 2a_i a_j + 4a_i a_j \right\} \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \]

\[ + \frac{2a_i^3}{36} \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \]

\[ + \frac{h^4}{24} \left\{ -2a_i^3 + 2a_i a_j + a_i a_j \right\} \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \]

\[ + \frac{h^5}{32} \left\{ 2a_i^3 + 2a_i a_j + a_i a_j \right\} \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \]

\[ + \frac{h^6}{36} \left\{ 2a_i^3 + 2a_i a_j + a_i a_j \right\} \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \left( a_i + a_j + a_k + a_l + a_m + a_n \right) \]

By Taylor Series expansion of one variable we have

\[ y_{\nu+1} = \sum_{j=0}^{\infty} \frac{h^j}{j!} f(y_{\nu+j}) \]

So that

\[ y(x) = y_{\nu} + \frac{h}{2} y'(x) + \frac{h^2}{24} y''(x) + \frac{h^3}{144} y'''(x) + \frac{h^4}{512} y^{(iv)}(x) + \frac{h^5}{3600} y^{(v)}(x) + \frac{h^6}{245760} y^{(vi)}(x) + \frac{h^7}{12441600} y^{(vii)}(x) + O(h^8) \]

We then find the partial of y^n for n= i, ii, iii, iv and substitute in Given rise to

\[ y_{\nu+1} - y_{\nu} = h^2 \left[ f(x) + \frac{h^2}{6} f''(x) + \frac{h^4}{24} f''''(x) + \frac{h^6}{5040} f''''''(x) + \frac{h^8}{362880} f'''''''(x) + O(h^9) \right] \]

Hence by k for f the Taylor Series in terms of y-functional derivatives yield

\[ y_{\nu+1} - y_{\nu} = h^2 k f_m + \frac{h^4}{6} k f_{m+1} + \frac{h^6}{24} k f_{m+2} + \frac{h^8}{5040} k f_{m+3} + \frac{h^{10}}{362880} k f_{m+4} + O(h^{11}) \]

Comparing equation (24) with (27) we have the following six equations in six unknowns

\[ 2a_i^3 + 2a_i a_j + a_i a_j + a_k + a_l + a_m + a_n = 3 \]

\[ 2a_i^3 + 2a_i a_j + a_i a_j + a_k + a_l + a_m + a_n = 2 \]

\[ -2a_i^3 + 2a_i a_j + a_i a_j + a_k + a_l + a_m + a_n = -2 \]

\[ 2a_i^3 + 2a_i a_j + a_i a_j + a_k + a_l + a_m + a_n = 3/2 \]

\[ -2a_i^3 + 2a_i a_j + a_i a_j + a_k + a_l + a_m + a_n = 4 \]

\[ -2a_i^3 + 2a_i a_j + a_i a_j + a_k + a_l + a_m + a_n = 4 \]
\[
\left[2a_1^2 + 4a_1a_2 + 8a_2a_3 + 4a_3a_4 + (a_1 + a_2 + a_3 + a_4) + 4a_4(a_3 + a_4) + a_2(a_3 + a_4)\right] \\
+ 2(a_2 + a_3)^2 + (a_3 + a_4)^2 - a_1(a_2 + a_3)^2 - a_4(a_3 + a_4) - 8a_2a_3(a_3 + a_4) \\
- 4a_1a_1(a_1 + a_2 + a_3) \\
- [(a_1 + a_2 + a_3)(a_1 + a_2 + a_3) + (a_1 + a_2 + a_3) + (a_1 + a_2 + a_3) + 4a_3^2] \\
= 2
\]

(28f)

Solving the above equations we obtain the values of the parameters \(a_1, a_2, a_3, a_4, a_5, \) and \(a_6\) as following.

\[a_1 = 1/2, a_2 = -1/16, a_3 = 9/16, a_4 = -1/8, a_5 = 5/24, a_6 = 11/12\]

Substituting these values of \(a_i\) in equation (10b) we have the required formula given by

\[y_{n+1} = y_n + \frac{h}{3} \left( k_1 + 2k_2 + k_3 \right)\]

(29)

with

\[k_1 = f(y_n)\]

(29a)

\[k_2 = f(y_n + \frac{h}{2} k_1)\]

(29b)

\[k_3 = f(y_n + \frac{h}{6} (-k_1 + 9k_2))\]

(29c)

\[k_4 = f(y_n + \frac{h}{24} (-3k_1 + 5k_2 + 22k_3))\]

(29d)

**Theorem**

The new fourth order Runge Kutta has a local truncation error \(e_t\) that can be estimated by integrating between two points \(x_n\) and \(x_{n+1}\) using two different step sizes \(h_1\) and \(h_2\) to evaluate \(y_{n+1}\).

**Proof**

Let the local truncation error i.e. be

\[e_t = kh^{n+1}\]

(30)

With \(k = \text{constant}\), and corresponding solution \(y_{n+1}, \alpha_1\) and \(y_{n+1}, \alpha_2\)

Then, if the true solution is \(y^*_{n+1}\), using Richardson extrapolation technique, we have

\[y^*_{n+1} - y_{n+1}, \alpha_1 = kh_1^{p+1} \frac{y_{n+1} - y_n}{h_1}\]

(31)

\[y^*_{n+1} - y_{n+1}, \alpha_2 = kh_2^{p+1} \frac{y_{n+1} - y_n}{h_2}\]

(32)

Where \(\alpha_1\) and \(\alpha_2\) are solution points

Dividing (31) by (32) and solving for \(y^*_{n+1}\), we have

\[y^*_{n+1} - y_{n+1}, \alpha_1 = kh_1^{p+1} \frac{(y_{n+1} - y_n)/h_1}{h_2}\]

\[y^*_{n+1} - y_{n+1}, \alpha_2 = kh_2^{p+1} \frac{(y_{n+1} - y_n)/h_2}{h_2}\]

So that \(y^{s}_{n+1} - y_{n+1}, \alpha_1 = (y^* - y_{n+1}, \alpha_2) (h_1/p/h_2) = y^{s}_{n+1} (h_1/p/h_2) - (y_{n+1}, \alpha_2) h_1/p/h_2\)

\[y^{s}_{n+1} = y_{n+1}, \alpha_1 - y_{n+1}, \alpha_2 + (h_1/p/h_2)\]

\[y^{s}_{n+1} = y_{n+1}, \alpha_1 - 2^p (y_{n+1}, \alpha_2)\]

(32)

It can be seen that an estimate of the i.t.e for \(y_{n+1}, \alpha_1\) assuming that \(h_{n+1}=h_1\) is

\[e_t = kh^{n+1}_1 = 2^p \frac{(y_{n+1}, \alpha_2 - y_{n+1}, \alpha_1)}{2^p - 1}\]

(33)

So that for our new RKF, \(P = 4\)

We have \(et = kh^{5}_1 = \frac{16}{15} (y_{n+1}, \alpha_2 - y_{n+1}, \alpha_1)\)

We therefore conclude that the error bound for the integrator is within manageable limit. We will consider in the next publication the consistency and convergence level of the integrator. Also we will implement the integrator by using it to solve existing initial value problem and show that it can be implemented by use of an appropriate computer program.

**Implementation and results**

The new formula was implemented by use of FORTRAN program we developed to solve some selected i.v.p.s whose results are as shown below side by side with the results of two other fourth Runge Kutta Formulae. The three formulae are here displayed with their results as shown below.
Table 1(a). \( y' = 1 + y^2, \ y(0) = 1, \quad 0.1 \leq x \leq 1.0 \)

<table>
<thead>
<tr>
<th>( T_{sol} )</th>
<th>Exact-Sol</th>
<th>Error</th>
<th>Exact-Sol</th>
<th>Error</th>
<th>Exact-Sol</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000D+00</td>
<td>2.27800</td>
<td>1.2780D+00</td>
<td>0.122D+01</td>
<td>0.205D-02</td>
<td>0.122D+01</td>
<td>0.663D-05</td>
</tr>
<tr>
<td>1.1054D+00</td>
<td>1.1054D+00</td>
<td>3.2885D-12</td>
<td>0.150D+01</td>
<td>0.656D-02</td>
<td>0.151D+01</td>
<td>0.277D-04</td>
</tr>
<tr>
<td>1.2230D+00</td>
<td>1.2230D+00</td>
<td>3.9968D-15</td>
<td>0.188D+01</td>
<td>0.170D-01</td>
<td>0.190D+01</td>
<td>0.996D-04</td>
</tr>
<tr>
<td>1.3561D+00</td>
<td>1.3561D+00</td>
<td>7.6941D-09</td>
<td>0.242D+01</td>
<td>0.440D-01</td>
<td>0.246D+01</td>
<td>0.383D-03</td>
</tr>
<tr>
<td>1.5085D+00</td>
<td>1.5085D+00</td>
<td>1.0565D-07</td>
<td>0.328D+01</td>
<td>0.126D+00</td>
<td>0.341D+01</td>
<td>0.182D-02</td>
</tr>
<tr>
<td>1.6858D+00</td>
<td>1.6858D+00</td>
<td>9.2532D-07</td>
<td>0.487D+01</td>
<td>0.461D+00</td>
<td>0.532D+01</td>
<td>0.136D-01</td>
</tr>
<tr>
<td>1.8958D+00</td>
<td>1.8958D+00</td>
<td>5.9195D-06</td>
<td>0.865D+01</td>
<td>0.303D+01</td>
<td>0.114D+02</td>
<td>0.298D+00</td>
</tr>
<tr>
<td>2.1497D+00</td>
<td>2.1497D+00</td>
<td>3.0092D-05</td>
<td>0.236D+02</td>
<td>-0.921D+02</td>
<td>0.123D+03</td>
<td>-0.192D+03</td>
</tr>
<tr>
<td>2.4650D+00</td>
<td>2.4650D+00</td>
<td>1.2864D-04</td>
<td>0.160D+03</td>
<td>-0.168D+03</td>
<td>0.258D+12</td>
<td>-2.58D+12</td>
</tr>
<tr>
<td>2.8689D+00</td>
<td>2.8689D+00</td>
<td>4.8193D-04</td>
<td>0.542D+09</td>
<td>-0.542D+09</td>
<td>0.746+123</td>
<td>-0.746+123</td>
</tr>
<tr>
<td>3.4082D+00</td>
<td>3.4066D+00</td>
<td>1.6356D-03</td>
<td>0.310D+70</td>
<td>-0.310D+70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1694D+00</td>
<td>4.1642D+00</td>
<td>5.1869D-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1(b). \( y' = -y, \ y(0) = 1, \quad 0 \leq x \leq 1.0 \)

<table>
<thead>
<tr>
<th>( T_{sol} )</th>
<th>Exact-Sol</th>
<th>Error</th>
<th>Exact-Sol</th>
<th>Error</th>
<th>Exact-Sol</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9048D+00</td>
<td>0.8948D+00</td>
<td>0.1001D-01</td>
<td>0.905D+00</td>
<td>-0.820D-07</td>
<td>0.905E+00</td>
<td>-0.191E-06</td>
</tr>
<tr>
<td>0.8187D+00</td>
<td>0.8101D+00</td>
<td>0.8597D-02</td>
<td>0.819D+00</td>
<td>-0.148D-06</td>
<td>0.819E+00</td>
<td>-0.347E-06</td>
</tr>
<tr>
<td>0.7408D+00</td>
<td>0.7334D+00</td>
<td>0.7406D-02</td>
<td>0.741D+00</td>
<td>-0.201D-06</td>
<td>0.741E+00</td>
<td>-0.472E-06</td>
</tr>
<tr>
<td>0.6703D+00</td>
<td>0.6639D+00</td>
<td>0.6397D-02</td>
<td>0.670D+00</td>
<td>-0.243D-06</td>
<td>0.670E+00</td>
<td>-0.569E-06</td>
</tr>
<tr>
<td>0.6065D+00</td>
<td>0.6010D+00</td>
<td>0.5540D-02</td>
<td>0.607D+00</td>
<td>-0.275D-06</td>
<td>0.607E+00</td>
<td>-0.644E-06</td>
</tr>
<tr>
<td>0.5488D+00</td>
<td>0.5440D+00</td>
<td>0.4811D-02</td>
<td>0.549D+00</td>
<td>-0.298D-06</td>
<td>0.549E+00</td>
<td>-0.699E-06</td>
</tr>
<tr>
<td>0.4966D+00</td>
<td>0.4924D+00</td>
<td>0.4189D-02</td>
<td>0.497D+00</td>
<td>-0.315D-06</td>
<td>0.497E+00</td>
<td>-0.738E-06</td>
</tr>
<tr>
<td>0.4493D+00</td>
<td>0.4457D+00</td>
<td>0.3656D-02</td>
<td>0.449D+00</td>
<td>-0.326D-06</td>
<td>0.449E+00</td>
<td>-0.764E-06</td>
</tr>
<tr>
<td>0.4066D+00</td>
<td>0.4034D+00</td>
<td>0.3198D-02</td>
<td>0.407D+00</td>
<td>-0.331D-06</td>
<td>0.407E+00</td>
<td>-0.778E-06</td>
</tr>
<tr>
<td>0.3679D+00</td>
<td>0.3651D+00</td>
<td>0.2805D-02</td>
<td>0.368D+00</td>
<td>-0.333D-06</td>
<td>0.368E+00</td>
<td>-0.782E-06</td>
</tr>
</tbody>
</table>

Formula A
\[
k_1 = f(x_n, y_n), \quad k_2 = f(x_n, y_n + \frac{1}{2} h k_1), \quad k_3 = f(x_n + \frac{1}{2} h, y_n + h k_2), \quad k_4 = f(x_n + h, y_n + h k_3), \quad y_{n+1} = y_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \tag{34}
\]

Formula B
\[
k_1 = f(y_n), \quad k_2 = f(y_n + \frac{h}{2} k_1), \quad k_3 = f(y_n + \frac{h}{2} k_2), \quad k_4 = f(y_n + h k_3)
\]

\[ C \ k_1 = f(y_n), \ k_2 = f(y_n + \frac{h}{2} k_1), \ k_3 = f(y_n + \frac{h}{16} [-k_1 + 9k_2]) \]

\[ k_4 = f(y_n + \frac{h}{24} [-3k_1 + 5k_2 + 22k_3]) \]

\[ y_{n+1} = y_n + \frac{h}{3} (\sqrt{k_1 k_2 + k_2 k_3 + k_3 k_4}) \tag{36} \]

And in Tables 1a, b and c.

Conclusion

After a successful derivation of the algorithm, we implemented the new methods to solve some set of standard singular ivps and found that the results generated have high accuracy and have minimal errors. From the above results, it will be observed that our new formula is
Table 1(c). $y' = -10(y - 1)^2, y(0) = 2, \quad 0 \leq x \leq 1.0$

<table>
<thead>
<tr>
<th>Tsol</th>
<th>Exact-Sol</th>
<th>Error</th>
<th>Exact-Sol</th>
<th>Error</th>
<th>Exact-Sol</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5000D+00</td>
<td>1.0910D+00</td>
<td>4.0901D-01</td>
<td>0.190D+01</td>
<td>0.104D+00</td>
<td>0.187E+01</td>
<td>0.125E+00</td>
</tr>
<tr>
<td>1.3333D+00</td>
<td>1.3333D+00</td>
<td>4.6503D-05</td>
<td>0.181D+01</td>
<td>-3.14D+00</td>
<td>0.177E+01</td>
<td>-2.67E+00</td>
</tr>
<tr>
<td>1.2500D+00</td>
<td>1.2500D+00</td>
<td>5.8936D-07</td>
<td>0.175D+01</td>
<td>-4.14D+00</td>
<td>0.168E+01</td>
<td>-3.42E+00</td>
</tr>
<tr>
<td>1.2000D+00</td>
<td>1.2000D+00</td>
<td>1.1308D-05</td>
<td>0.169D+01</td>
<td>-4.41D+00</td>
<td>0.160E+01</td>
<td>-3.47E+00</td>
</tr>
<tr>
<td>1.1667D+00</td>
<td>1.1665D+00</td>
<td>1.3662D-04</td>
<td>0.164D+01</td>
<td>-4.44D+00</td>
<td>0.153E+01</td>
<td>-3.31E+00</td>
</tr>
<tr>
<td>1.1429D+00</td>
<td>1.1419D+00</td>
<td>9.6932D-04</td>
<td>0.160D+01</td>
<td>-4.37D+00</td>
<td>0.147E+01</td>
<td>-3.08E+00</td>
</tr>
<tr>
<td>1.1250D+00</td>
<td>1.1204D+00</td>
<td>4.6283D-03</td>
<td>0.157D+01</td>
<td>-4.25D+00</td>
<td>0.143E+01</td>
<td>-2.84E+00</td>
</tr>
<tr>
<td>1.1111D+00</td>
<td>1.0952D+00</td>
<td>1.5941D-02</td>
<td>0.154D+01</td>
<td>-4.12D+00</td>
<td>0.138E+01</td>
<td>-2.60E+00</td>
</tr>
<tr>
<td>1.1000D+00</td>
<td>1.0588D+00</td>
<td>4.1239D-02</td>
<td>0.151D+01</td>
<td>-3.98D+00</td>
<td>0.135E+01</td>
<td>-2.38E+00</td>
</tr>
<tr>
<td>1.0909D+00</td>
<td>1.0076D+00</td>
<td>8.3280D-02</td>
<td>0.148D+01</td>
<td>-3.84D+00</td>
<td>0.132E+01</td>
<td>-2.17E+00</td>
</tr>
</tbody>
</table>

The above results show that method can cope very well in solving initial value problems in ordinary differential equations.

A-stable, and shows high degree of consistency when used to solve singular initial value problems. Therefore, we summarize this work by simply saying that our new 4th order Runge-Kutta Formula have value, it is reliable and I therefore, recommend it for use in the computation of O.D.E problems. The various computations displaced in the tables above are enough proof of the performances of our new methods.

REFERENCES


Euler L (1768), Opera Omnia, Series Prima, 11, Leipsig and Beline.


