On error estimation in almost Runge-Kutta (ARK) methods

Ochoche Abraham[1], Kayode R. Adeboye[2] and Alimi K. Olumide[3]

Abstract

In Runge-Kutta Theory, it is common knowledge that the bounds for the local truncation errors do not form a suitable basis for monitoring the local truncation error [7]. In a number of literatures including [1], Richardson Extrapolation technique has been shown to be a very reliable means of obtaining error estimates for Runge-Kutta methods. In this paper, we have sought to investigate, by means of rigorous numerical experiments, if this effectiveness of Richardson extrapolation technique extends to ARK methods and our results show that it doesn't.

Mathematics Subject Classification: 65G99

[1] Department of Information and Media Technology, Federal University of Technology Minna, Nigeria, e-mail: abochoche@futminna.edu.ng
[2] Department of Mathematics and Statistics, Federal University of Technology Minna, Nigeria, e-mail: profadeboye@gmail.com
[3] Department of Mathematics and Statistics, Federal University of Technology Minna. Nigeria, e-mail: alimikayum@yahoo.com

1   Introduction

Mathematics is the lingua franca of life; by it the physical world finds expression. Historically, different equations have originated in Chemistry, Physics and Engineering. More recently they have also arisen in models in Medicine, Biology, Anthropology and the likes. In this paper, we shall restrict our attention to just numerical methods for approximating the solution of ordinary differential equations with prime focus on initial value problems (IVP); so called because the conditions on the solution of the differential equation are all specified at the start of the trajectory i.e. they are initial conditions.

Numerical solution of ODE is the most important technique ever developed in continuous time dynamics. Since most ODEs are not soluble analytically, numerical integration is the only way to obtain information about the trajectory. Many different methods have been proposed and used in an attempt to solve accurately, various types of ODE. However, there is a handful of methods known and used universally (e.g. Runge-Kutta ([6], [12]), Adam-Bashforth-Moulton ([3], [8]) and Backward Difference Formulae). All these, discretize the differential system, to produce a difference equation or map.

The methods produce different maps from the same equation, but they have the same aim: that the dynamics of the maps, should correspond closely, to the dynamics of the differential equation. From the Runge-Kutta family of algorithms, come the most well-known and used methods for numerical integration [5].

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

2   Preliminary Notes

Almost Runge-Kutta (ARK) methods are a very special class of general linear methods introduced by Butcher in 1997 [4]. The basic idea of these methods is to retain the multi-stage nature of Runge-Kutta methods, while allowing more than one value to be passed from step to step. Hence, they have a multi-value nature. These methods have advantages over traditional methods, which are to be found in low-cost local error estimation and dense output. These latter features will be a consequence of the higher stage orders that are possible because of the multi-value nature of the new methods. Though this multivalue nature brings its own difficulties in the sense that satisfactory solutions have to be found not only for stepsize changes, but also for the starting steps.

The number of values passed between steps varies among general linear methods; the number of values is three for ARK methods. Of the three input and output values in ARK methods, one approximates the solution value i.e. $y(x_n)$ the second approximates the scaled first derivative ($hy'(x_n)$) while the third approximates the scaled second derivatives ($h^2 y''(x_n)$). To simplify the starting procedure, the second derivative is required to be accurate only to within $O(h^3)$, where h as usual is the stepsize. In other to ensure that this low order does not adversely affect the solution value, the method has inbuilt *annihilation conditions*. As a result of these extra input values, we are able to obtain stage order two, unlike explicit Runge-Kutta methods that are only able to obtain stage order one. The main advantage of the increase in stage order is that error estimates and continuous solutions can be more easily achieved [10].

ARK have the general form:

$$
\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \\ \hline y_1^{[n]} \\ y_2^{[n]} \\ y_3^{[n]} \end{bmatrix} = \left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] \begin{bmatrix} hF(Y_1) \\ hF(Y_2) \\ \vdots \\ hF(Y_s) \\ \hline y_1^{[n-1]} \\ y_2^{[n-1]} \\ y_r^{[n-1]} \end{bmatrix}
\tag{2.1}
$$

with $\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] = \left[\begin{array}{c|ccc} A & e & c-Ae & \frac{1}{2}c^2 - Ac \\ \hline b^T & 1 & b_0 & 0 \\ e_s^T & 0 & 0 & 0 \\ \beta^T & 0 & \beta_0 & 0 \end{array}\right]$

$$\tag{2.2}$$

where s is the number of internal stages.

The use of an ARK method is very similar to that of an RK method, with the main difference being that three pieces of information is now passed between steps. The first two starting values are $y(x_0)$ and $hf(y(x_0))$ respectively and the third starting value is obtained by taking a single Euler step forward and then taking the difference between the derivatives at these two points. Therefore, the starting vector is given by

$$
\left[y(x_0), hf(y(x_0)), hf\big(y(x_0) + hf(y(x_0))\big) - hf(y(x_0)) \right]
\tag{2.3}
$$

This choice of starting method was chosen for its simplicity, but it is adequate, at least for low order methods. The method for computing the three starting approximations can be written in the form of the generalized Runge–Kutta tableau

$$
\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & -1 & 1
\end{array}
$$

where the zero in the first column of the last two rows indicates the fact that the term y_{n-1} is absent from the output approximation. This can be interpreted in the same way as a Runge–Kutta method, but with three output approximations.

The usual way to change the stepsize is to simply scale the vector in the same way one would scale a Nordsieck vector [9]. Setting $r = h_j / h_{j-1}$ means the y vector needs to be scaled by $\begin{bmatrix} 1 & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r^2 \end{bmatrix}$.

For an order p method the three output values are given by the equations below

$$\left.\begin{array}{l} y_1^{[n]} = y(x_n) + O(h^{p+1}), \\ y_2^{[n]} = hy'(x_n) + O(h^{p+2}), \\ y_3^{[n]} = h^2 y''(x_n) + O(h^3). \end{array}\right\} \tag{2.4}$$

In choosing the coefficients of the method, we are careful to ensure that the simple stability properties of Runge-Kutta methods are retained [10].

In this paper we will concentrate on methods where A is strictly lower triangular, and hence the methods are explicit, but most of the theory can be carried over to implicit methods [2].

3   Richardson extrapolation technique

The *deferred approach to the limit*, otherwise known as Richardson extrapolation [11] involves solving a problem twice using step sizes h and $2h$. Under the localizing assumption that no previous errors have been made, we may write

$$y(x_{n+1}) - y_{n+1} = T_{n+1} = \varphi(x_n, y(x_n))h^{p+1} + O(h^{p+2}) \tag{3.1}$$

where p is the order of the method, $\varphi(x_n, y(x_n))h^{p+1}$ is the principal local truncation error. Next, we will compute y_{n+1}^*, a second approximation to $y(x_{n+1})$,

obtained by applying the same method at x_{n-1} with steplenght $2h$. Under the same localizing assumption, it follows that:

$$y(x_{n+1}) - y^*_{n+1} = \varphi(x_{n-1}, y(x_{n-1}))(2h)^{p+1} + O(h^{p+2}) \qquad (3.2)$$

and on expanding $\varphi(x_{n-1}, y(x_{n-1}))$ about (x_n, y_n),

$$y(x_{n+1}) - y^*_{n+1} = \varphi(x_n, y(x_n))(2h)^{p+1} + O(h^{p+2}) \qquad (3.3)$$

On subtracting (3.1) from (3.3), we obtain

$$y(x_{n+1}) - y^*_{n+1} = (2^{p+1} - 1)\varphi(x_n, y(x_n))h^{p+1} + O(h^{p+2})$$

Therefore, the principal local truncation error which is taken as an estimate for the local truncation error may be written as:

$$\varphi(x_n, y(x_n))h^{p+1} = T_{n+1} = \frac{y(x_{n+1}) - y^*_{n+1}}{2^{p+1} - 1} \qquad (3.4)$$

$$\Rightarrow T_{n+1} = \frac{y(x_{n+1}) - y^*_{n+1}}{2^{p+1} - 1} \qquad (3.5)$$

3.1   Numerical experiments

In [1] as well as other literature, Richardson Extrapolation has been shown to be a means of obtaining quick and acceptable estimates of the local truncation errors in computations using any s-stage explicit Runge-Kutta method, without having to obtain the exact solution first. The goal of this paper is to investigate the viability of equation (3.5) for ARK methods by solving the following initial value problems:

1. $y' = \dfrac{y}{4}(y - \dfrac{y}{20});\quad y(0) = 1$.   Exact solution: $y = \dfrac{20e^t}{19 + e^{t/4}}$

2. $y' = y\cos x,\quad y(0) = 1$.   Exact Solution: $y = Ce^{\sin x},\quad C = 1$.

A numerical solver was developed using Java programming language to solve the differential equations above using the following Almost Runge-Kutta methods.

3.2.1 Method 1

ARK45: A Fourth order method with five stages

$$c^T = [\tfrac{1}{4}, \tfrac{1}{2}, \tfrac{3}{4}, 1, 1], \ \varphi = 4$$

$$
\left[
\begin{array}{ccccc|ccc}
0 & 0 & 0 & 0 & 0 & 1 & \dfrac{1}{4} & \dfrac{1}{32} \\[2mm]
\dfrac{2}{5} & 0 & 0 & 0 & 0 & 1 & \dfrac{1}{10} & \dfrac{1}{40} \\[2mm]
\dfrac{3}{140} & \dfrac{75}{112} & 0 & 0 & 0 & 1 & \dfrac{33}{560} & -\dfrac{33}{560} \\[2mm]
\dfrac{543}{245} & -\dfrac{87}{49} & 1 & 0 & 0 & 1 & -\dfrac{108}{245} & \dfrac{41}{190} \\[2mm]
\dfrac{16}{45} & \dfrac{2}{15} & \dfrac{16}{45} & \dfrac{7}{90} & 0 & 1 & \dfrac{7}{90} & 0 \\[2mm]
\hline
\dfrac{16}{45} & \dfrac{2}{15} & \dfrac{16}{45} & \dfrac{7}{90} & 0 & 1 & \dfrac{7}{90} & 0 \\[2mm]
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\[2mm]
-\dfrac{56}{9} & \dfrac{62}{25} & -\dfrac{784}{225} & -\dfrac{196}{225} & \dfrac{24}{5} & 0 & \dfrac{742}{225} & 0
\end{array}
\right]
$$

3.2.2 Method 2

ARK5:　A Fifth order method with five stages

$$
\left[
\begin{array}{ccccc|ccc}
0 & 0 & 0 & 0 & 0 & 1 & 53/150 & 50/801 \\
-150/817 & 0 & 0 & 0 & 0 & 1 & 271/625 & 57/593 \\
1564/749 & -162/209 & 0 & 0 & 0 & 1 & -143/254 & -1/3 \\
1747/250 & 2927/450 & 1327/994 & 0 & 0 & 1 & -10979/794 & -2859/622 \\
220/989 & 64/279 & 369/950 & 4/55 & 0 & 1 & 83/954 & 0 \\
\hline
220/989 & 64/279 & 369/950 & 4/55 & 0 & 1 & 83/954 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
14651/302 & 7424/279 & 0 & -1016/873 & 4 & 0 & -7562/97 & 0
\end{array}
\right]
$$

4   Main Results

In Figure 1 we can see from the solution curves that the curve for the error estimates using Richardson extrapolation is very far from the curve of the numerical approximation using ARK45 with h= 0.1 and h = 0.2.

In Figure 2 we can see from the solution curves again, that the curve for the error estimates using Richardson extrapolation is very far from the curve of the numerical approximation using ARK5 with h= 0.1 and h = 0.2.

In Figure 3 we can see from the solution curves that the estimates from Richardson extrapolation again, has no similarities whatsoever with the solution curves of the numerical approximation using ARK45 with h = 0.1 and h = 0.2.

In Figure 4 just as in Figure 3, we can see from the solution curves, that the estimates from Richardson extrapolation again, has no similarities whatsoever with the solution curves of the numerical approximation using ARK5 with h = 0.1 and h = 0.2.
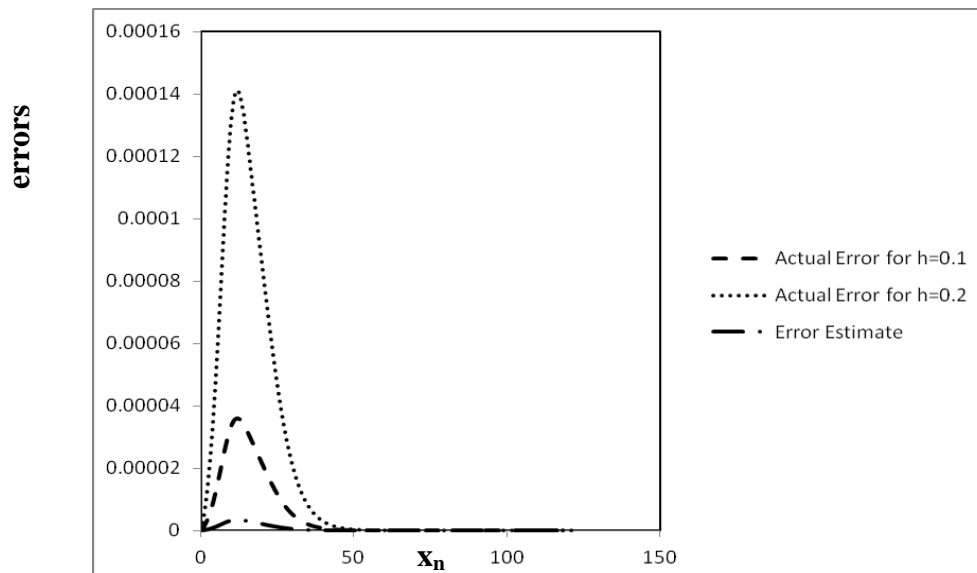
5   Labels of figures and tables

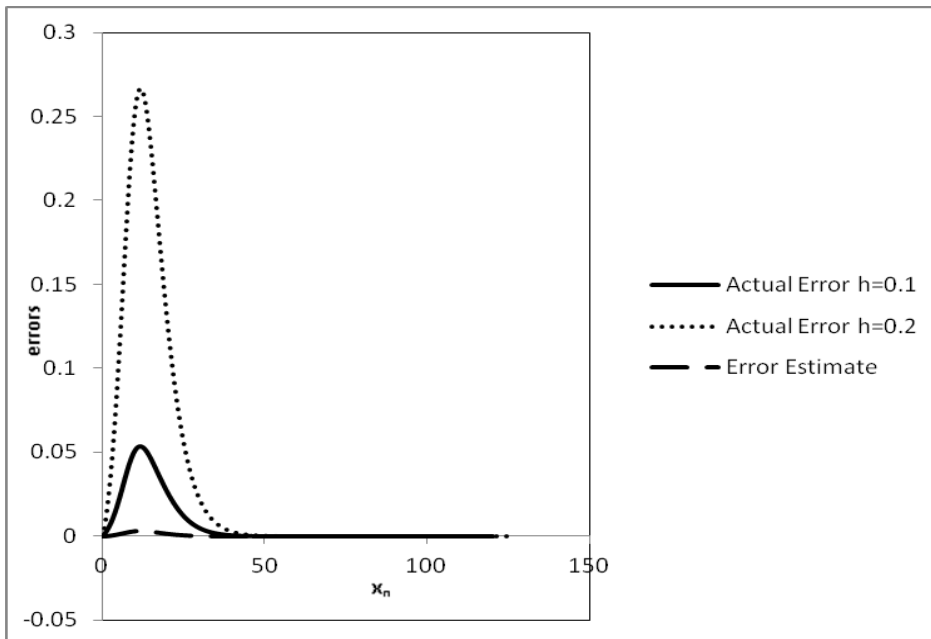Figure 1: Graph of actual errors and estimated errors for problem 1 using ARK45

Figure 2: Graph of actual errors and estimated errors for problem 1 using ARK5
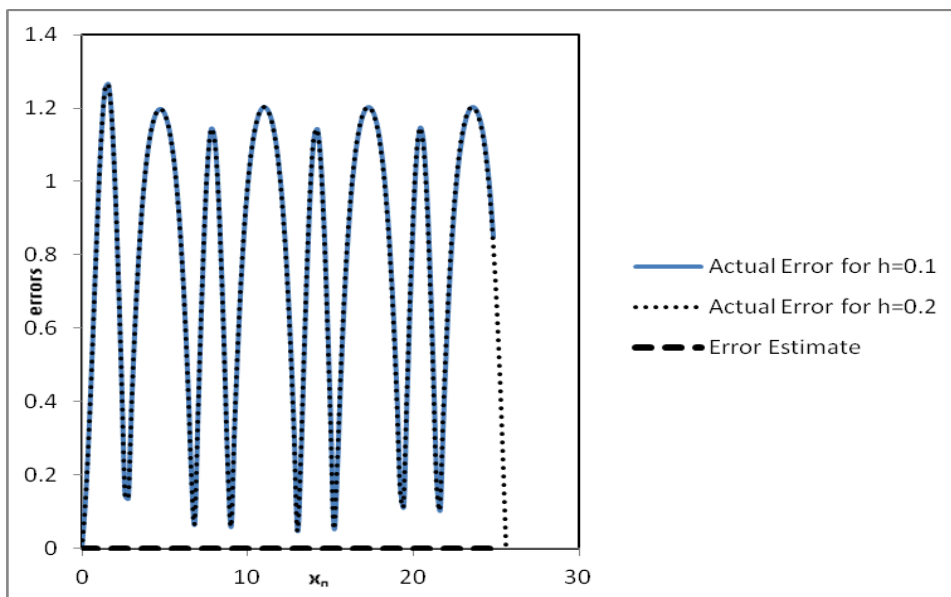

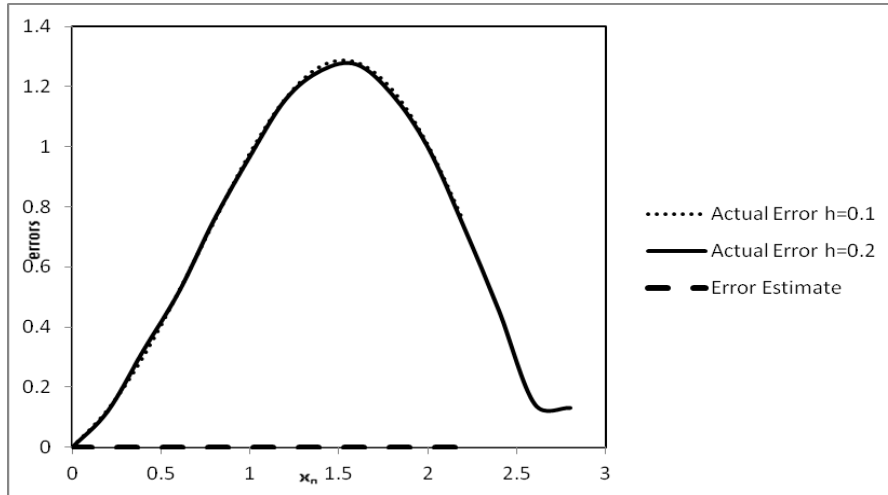Figure 3: Graph of actual errors and estimated errors for problem 2 using ARK45

Figure 4: Graph of actual errors and estimated errors for problem 2 using ARK5

6   Conclusion

From the results obtained in section 5.0, it can be therefore concluded that while Richardson extrapolation technique is a very effective means of obtaining acceptable error estimates for Runge-Kutta methods, it is not the same for Almost Runge-Kutta methods.

References

[1] O. Abraham and G. Bolarin, On Error Estimation in Runge-Kutta Methods, *Leonardo Journal of Sciences*, **18**, (January-June, 2011), 1-10.

[2] O. Abraham, *Development of Some New Classes of Explicit Almost Runge-Kutta (EARK) Methods for Non-Stiff Differential Equations*, Ph.D Thesis, Federal University of Technology, Minna, Nigeria, 2011.

[3] F. Bashforth and J.C. Adams, *An attempt to Test the Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops of Fluid, with an Explanation of the Method of Integration Employed in Constructing the Tables which Give the Theoretical Forms of Such Drops*, Cambridge University Press, Cambridge, 1883.

[4] J.C. Butcher, An Introduction to "Almost Runge–Kutta" Methods, *Applied Numerical Mathematics,* **24**, (1997), 331-342.

[5] J.H.C. Julyan and O. Piro, The Dynamics of Runge-Kutta Methods, *International Journal Bifurcation and Chaos*, **2**, (1992), 1-4.

[6] W. Kutta, Beitrag fur N¨Aherungsweisen Integration Totaler Differentialgleichungen, *Z. Math. Phys*., **46**, (1901), 435-453.

[7] J.D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley and Sons, USA, pp. 149-150, 1991.

[8] F.R. Moulton, *New Methods in Exterior Ballistics*, University of Chicago Press, 1926.

[9] A. Nordseick, On Numerical Integration of Ordinary Differential Equations, *Math. Comp.*, **16**, (1962), 22-49.

[10] N. Rattenbury, *Almost Runge-Kutta Methods for Stiff and Non-Stiff Problems*, PhD Thesis, The University of Auckland, New Zealand, 2005.

[11] L.F. Richardson, The deferred approach to the limit, 1-single lattice, *Trans. Roy. Soc. London*, **226**, (1927), 299-349.

[12] C. Runge, Ueber Die Nuumerische Auflösung von Differentialgleichungen, *Math. Ann*., **46**, (1895), 167-178.