Almost Runge-Kutta Methods Of Orders Up To Five

ABRAHAM OCHOCHE
Department of Information Technology,
Federal University of Technology,
Minna, Niger State P.M.B. 65,
Nigeria
abochoche@gmail.com

PETER NDAJAH
Graduate School of Science and Technology,
Niigata University,
8050 Ikarashi 2-no-cho, Nishi-ku
Niigata, 950-2181, Japan
ndajah@telecom0.eng.niigata-u.ac.jp

Abstract: In this paper, we have sought to investigate the viability of a type of general linear methods called Almost Runge-Kutta (ARK) methods, as a means of obtaining acceptable numerical approximations of the solution of problems in continuous mathematics. We have outlined the derivation and implementation of this class of methods up to order five. Extensive numerical experiments were carried out and the results clearly show that ARK methods are indeed a viable alternative to existing traditional methods.

Key–Words: Almost, Order, Alternative, Euler, Runge – Kutta, General Linear Methods.

1 Introduction

Ordinary differential equations (ODEs) can be used to model many different types of physical behaviors, form chemical reactions to the notion of planets around the sun. They are at the very heart of our understanding of the physical world today[see, 1–15]. Unfortunately, as the systems being modeled get more complex, so do the solutions. This can mean that we are left in the position of being unable to solve the problem analytically. This is where the need for numerical methods become obvious. Even though they do not give exactly solutions, they do enable us to find good approximations to the solution at any point where it is required.

2 General Linear Methods

The name “General linear methods” applies to a large family of numerical methods for ordinary differential equations. They were introduced in [9] to provide a unifying theory of the basic questions of consistency, stability and convergence. Later they were used as a framework for studying accuracy questions and later the phenomena associated with nonlinear convergence. They combine the multi-stage nature of Runge-Kutta methods with the notion of passing more than one piece of information between steps that is used in linear multi-step methods. Their discovery opened the possibility of obtaining essentially new methods that were neither Runge-Kutta nor linear multi-step methods would exist which are practical and have advantages over the traditional methods.

3 Almost Runge-Kutta Methods

Almost Runge-Kutta (ARK) methods are a very special class of general linear methods introduced by Butcher [10]. 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. This multi-value nature brings its own difficulties.

The number of values passed between steps varies among general linear methods; the number of val-
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 second, derivatives respectively \( (h^2y''(x_n)) \). 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 order to ensure that this low order does not adversely affect the solution value, the method has an adjunction of the form of Runge-Kutta methods as well as pass through very little information between steps. The in-}

\[ \begin{bmatrix}
  Y_1 \\
  Y_2 \\
  \vdots \\
  Y_s \\
  y[n]_1 \\
  y[n]_2 \\
  y[n]_3 \\
\end{bmatrix}
= \begin{bmatrix}
  0 & 0 & 0 & \ldots & 0 & 0 \\
  a_{21} & 0 & 0 & \ldots & 0 & 0 \\
  a_{31} & a_{32} & 0 & \ldots & 0 & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  a_{s-1,1} & a_{s-1,2} & a_{s-1,3} & \ldots & 0 & 0 \\
  b_1 & b_2 & b_3 & \ldots & b_{s-1} & 0 \\
  0 & 0 & 0 & \ldots & 0 & 1 \\
  \beta_1 & \beta_2 & \beta_3 & \ldots & \beta_{s-1} & \beta_s \\
\end{bmatrix}
\begin{bmatrix}
  e \\
  c - Ae \\
  \frac{e^2}{2} - Ac \\
\end{bmatrix}
\begin{bmatrix}
  hF(Y_1) \\
  hF(Y_2) \\
  \vdots \\
  hF(Y_s) \\
  y[n+1]_1 \\
  y[n+1]_2 \\
  y[n+1]_3 \\
\end{bmatrix}
\] (1)

As in traditional Runge-Kutta theory, \( b \) is a vector of length \( s \) representing the weights and \( c \) is a vector of length \( s \) representing the positions at which the function \( f \) is evaluated. The vector \( e \) is a vector of ones, and is also of length \( s \).

\[ y(x_0 + hc_i) = u_{i1}y_0 + u_{i2}hy_0' + u_{i3}h^2y_0'' \]
\[ +h \sum_{j=1}^{i-1} a_{ij} y'(x_0 + hc_j) \quad (4) \]

By carrying out the Taylor series expansion on both sides of equation (4) and equate the coefficients of \( y_0, y'_0 \) and \( y''_0 \), we find

\[ u_{41} = 1, \quad i = 1, 2, \ldots, s \]
\[ u_{42} = c_i - \sum_j a_{ij} \]
\[ u_{43} = \frac{c^2}{2} - \sum_j a_{ij} c_j \quad (5) \]

We must note some very important features of ARK methods:

1. The first row of the \( B \) matrix is the same as the last row of the \( A \) matrix, we denote this as \( b^T \) and the first row of the \( V \) matrix is the same as the last row of the \( U \) matrix, because we wish the final internal stage to give us the same quantity that is to be exported as the first outgoing approximation. It also means that \( c_4 = 1 \).

2. We also wish the second outgoing approximations to be \( h \) times the derivative of the final stage. This means that the second row of the \( B \) and \( V \) matrices consists of zeros, with the exception of a 1 in the \((2, s)\) position of \( B \) that is \( B = e_s^T \). The last row of \( B \) is \( \beta_s^T \).

There are three types of conditions to consider when deriving Almost Runge-Kutta methods:

1. Order conditions.
2. Annihilation conditions.

Detailed explanation of how these conditions are implemented, can be found in [11], [12].

**Theorem 3.1:** An ARK method of order \( p \) with \( p \) stages has RK stability if and only if

\[ \beta^T (I + \beta_s A) = \beta_s e_s^T \quad (6) \]

\[ (1 + \frac{1}{2} \beta_s c_1) b^T A s - 2 c = \frac{1}{s!} \quad (7) \]

\[ c_1 = -2 \exp_s(-\beta_s) \frac{\beta_s \exp_{s-1}(-\beta_s)}{\beta_s} \quad (8) \]

**Proof:** A proof of this theorem can be found in a number of literature including [11], [13].

### 3.3 Methods with \( s = p \)

This section, considers the derivation procedure for methods which have the same number of stages as the order of the method.

Methods with this property are considered so as to minimize computational costs, and because, it is not possible to satisfy all the order conditions for \( s < p \). Also, the stability function is always a polynomial of degree \( s \) but the stability function is an approximation to \( \exp(z) \) with an error of \( o\left(z^{p+1}\right) \) as \( |z| \to 0 \). This means that the stability function must have degree at least \( p \) and therefore \( s \geq p \).

#### 3.3.1 Derivation of Methods With \( s = p = 4 \)

The general form of a fourth order, four stage ARK method is

\[
\begin{bmatrix}
A \\
U \\
B \\
V
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
1 & c_1 & c_2 - a_{21} & \frac{1}{2} c_2^2 - a_{21} c_1 \\
a_{31} & a_{32} & 0 & 0 \\
b_1 & b_2 & b_3 & 0 \\
0 & 0 & 0 & 1 \\
\beta_1 & \beta_2 & \beta_3 & \beta_4
\end{bmatrix}
\]

(9)

The order conditions for a fourth order method are:

\[ b_0 + b^T e = 1 \quad (10) \]
\[ b^T c = \frac{1}{2} \quad (11) \]
\[ b^T e^2 = \frac{1}{3} \quad (12) \]
\[ b^T e^3 = \frac{1}{4} \quad (13) \]

\[ b^T A c = \frac{1}{6} \quad (14) \]
\[ b^T A c^2 = \frac{1}{12} \quad (15) \]
\[ \beta^T e + \beta_0 = 0 \quad (16) \]
\[ \beta^T (I + \beta_4 A) = \beta_4 e_s^T \quad (17) \]
\[ c_1 = \frac{2 \exp_4(-\beta_s)}{\beta_4 \exp_3(-\beta_1)} \quad (18) \]
and (15). Provided $c_1 \neq c_2$,
\[
a_{32} = \frac{1 - 2c_1}{12b_3c_2(c_2 - c_1)}
\]  
(23)

From conditions (14) and (19),
\[
a_{21} = \frac{1}{24b_3a_{32}c_1(1 + \frac{1}{3}\beta_4 c_1)}
\]  
(24)
\[
a_{31} = \frac{1}{b_3} - b_3a_{32}c_2 - b_2a_{21}c_1
\]  
(25)

Finally, $\beta^T$ can be found from equation (17).

Based on the complete classification of fourth order methods with four stages given in [14], we present two methods based on Case 4 (ark4a) and Case 5 (ark4b).

\[
e^T = [1, \frac{1}{2}, 0, 1] , \quad b_3 = \frac{3}{4} , e^T = [1, \frac{1}{2}, 1, 1] , \quad \beta_4 = 2
\]

3.3.2 Derivation of Methods with $s = p = 5$

A fifth order, five stage ARK method takes the form
\[
\begin{bmatrix}
A \\
B \\
V
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & c_1 \\
1 & c_2 - a_{21} \\
\frac{1}{2}c_2^2 - a_{21}c_1 \\
b_1 & b_2 & b_3 & b_4 & 0 \\
b_1 & b_2 & b_3 & b_4 & 0 \\
b_1 & b_2 & b_3 & b_4 & b_0
\end{bmatrix}
\]  
(27)

with $c = [c_1, c_2, c_3c_4, 1]^T$.

The order conditions for a fifth order five stage method are:
\[
b_0 + b^T e = 1
\]  
(28)
\[
b^T c = \frac{1}{2}
\]  
(29)
\[
b^T c^2 = \frac{1}{3}
\]  
(30)
\[
b^T c^3 = \frac{1}{4}
\]  
(31)
\[
b^T c^4 = \frac{1}{5}
\]  
(32)
\[
c_4 = 1
\]  
(33)
\[
b^T A = b^T (I - C)
\]  
(34)
\[
b^T (I - C)Ac = \frac{1}{24}
\]  
(35)
\[
b^T (I - C)Ac^2 = \frac{1}{60}
\]  
(36)
\[
\beta^T e + \beta_0 = 0 \quad (37)
\]
\[
\beta^T (I + \beta_5 A) = \beta_5 e_4^T \quad (38)
\]
\[
c_1 = \frac{2 \exp(-\beta)}{\beta_5 \exp(-\beta_5)} \quad (39)
\]
and
\[
(1 + \frac{1}{2} \beta_5 c_1) b^T A^3 c = \frac{1}{5!} \quad (40)
\]

3.3.3 Some Derived Methods

Not much is known about the best choice of parameters for order five methods; however we will attempt to present some methods

ARK5a
\[
c^T = \left[ \frac{52}{165}, \frac{1}{2}, 1, 1, 1 \right], \beta_5 = 3
\]
\[
\begin{bmatrix}
A & U \\
B & V
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & \frac{52}{165} & 1352 \\
-0.26015 & 0 & 0 & 0 & 0 & 1 & -0.104639 & 27725 \\
0.9768753 & -1.708 & 0 & 0 & 0 & 1 & -3.335551 & -1286 \\
-4.131219 & 0.5061 & 0 & 0 & 0 & 1 & 0.664368 & 22003 \\
-2.00174 & 1.980 & 0 & 0 & 0 & 1 & 1.788909 & 39830 \\
-1.94133 & -0.54 & 0 & 0 & 0 & 1 & -0.796 & 0 \\
-7.817477 & 0.43 & 0 & 0 & 0 & 1 & -0.332 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}
\] (41)

and ARK5a
\[
c^T = \left[ \frac{52}{165}, \frac{1}{2}, 1, 1, 1 \right], \beta_5 = 3
\]
\[
\begin{bmatrix}
A & U \\
B & V
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & \frac{52}{165} & 1352 \\
0.36095 & -0.0715 & 0 & 0 & 0 & 1 & -0.104639 & 27725 \\
0.48914 & -0.1792 & 0 & 0 & 0 & 1 & -3.335551 & -1286 \\
-1.19722 & 0.5061 & 0 & 0 & 0 & 1 & 0.664368 & 22003 \\
0.9882675 & 0.521 & 0 & 0 & 0 & 1 & 1.788909 & 39830 \\
0.9882675 & 0.43 & 0 & 0 & 0 & 1 & 0 & 0 \\
-2.293901 & 0.43 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}
\] (42)

respectively.

3.4 Methods with s = p+1

From traditional Runge-Kutta theory, we know that we can achieve enhanced performance if there are more stages than are required for a particular order; the same applies to ARK methods. The focus is on the case where there exists one more stage than is required to attain the required order.

3.4.1 Methods with s = 5, p = 4

A fourth order method with five stages takes the form

\[
\begin{bmatrix}
A & U \\
B & V
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & e & c - Ae & z^2 - Ae \\
1 & 0 & 0 & 0 & 0 & b_0 & 0 \\
0 & 1 & 0 & 0 & 0 & b_0 & 0 \\
0 & 0 & 1 & 0 & 0 & b_0 & 0 \\
0 & 0 & 0 & 1 & 0 & b_0 & 0 \\
0 & 0 & 0 & 0 & 1 & b_0 & 0 \\
0 & 0 & 0 & 0 & 0 & b_0 & 0 \\
0 & 0 & 0 & 0 & 0 & b_0 & 0 \\
0 & 0 & 0 & 0 & 0 & b_0 & 0 \\
0 & 0 & 0 & 0 & 0 & b_0 & 0
\end{bmatrix}
\] (43)

with stability function
\[
R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{3!} + \frac{z^4}{4!} + K z^5
\]
3.4.2 Order Conditions

The following conditions must be satisfied by an ARK method with five stages to have order four

\[ b_0 = 1 - b^T e \]  
\[ b^T c = \frac{1}{2} \]  
\[ b^T c^2 = \frac{1}{3} \]  
\[ b^T c^3 = \frac{1}{4} \]  
\[ c_4 = 1 \]  
\[ b^T Ac = \frac{1}{6} \]  
\[ b^T Ac^2 = \frac{1}{12} \]  
\[ \beta_0 = -\beta^T e \]  
\[ \beta^T c = 1 \]  
\[ \beta^T Ac = \frac{\beta_0 (\theta + 1) - \phi}{\theta \beta_5} \]

\[ \beta_5 e_5^T (I + \theta A) = \beta^T (I + \phi A + \beta_5 \theta A^2) \]  
\[ K \left( \frac{1}{2} \beta_5 c_1 \theta \alpha_4 - \alpha_5 \right) = (1 + \frac{1}{2} \beta_5 c_1)(1 + \alpha_1 + \frac{\alpha_2}{2} + \frac{\alpha_3}{3!} + \frac{\alpha_4}{4!}) \]

where the values of \( \alpha_i \) are determined by expanding

\[ \frac{1 + (\phi - \beta_5)z}{1 + \phi z + \beta_5 \theta z^2} = \sum_{i=0}^{\infty} \alpha_i z^i \]

\[ b^T A^2 c - \frac{1}{24} \theta (b^T A^3 c - K) \]

and

\[ \beta_5 (b^T A^2 c^2 - K) = (\beta_5 - \phi)(b^T A^3 c - K) \]

3.4.3 Some Examples

Two methods are presented here: ark45a and ark45b respectively. We have chosen \( L = \frac{1}{5} \) and \( K = \frac{1}{120} \) for both methods, ensuring zero error for both the bushy tree and the tall tree.

\[ e^T = \left[ \frac{1}{4}, \frac{1}{4}, \frac{3}{4}, 1, 1 \right], \quad \phi = 4, \quad \mu = 8, \quad L = \frac{1}{5}, \quad K = \frac{1}{120} \]

\[ \begin{bmatrix} A & U \\ B & V \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & \frac{1}{2} & 1 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 1 & \frac{1}{2} & 1 \end{bmatrix} \]  

\[ c^T = \left[ \frac{1}{4}, \frac{1}{4}, \frac{3}{4}, 1, 1 \right], \quad \phi = 4, \quad \mu = 8, \quad L = \frac{1}{5}, \quad K = \frac{1}{120} \]

\[ \begin{bmatrix} A & U \\ B & V \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & \frac{1}{2} & 1 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 1 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 1 & \frac{1}{2} & 1 \end{bmatrix} \]  

4 Implementation of ARK Methods

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 \( h f(y(x_0)) \) 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
\[ [y(x_0), hf(y(x_0)), hf(y(x_0) + hf(y(x_0))) - hf(y(x_0))] \]
(61)

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

\[
\begin{array}{cccc}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & -1 & 1 \\
\end{array}
\]

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.

To change the stepsize we simply scale the vector in the same way we would scale a Nordsieck vector \([15]\). Setting \( r = h_j / h_{j-1} \) means the \( y \) vector needs to be scaled by \([1, r, r^2]^T\).

Once the starting vector is obtained from equation (61), the next thing is to calculate the internal stages;

\[ Y_i^{[n]} = \sum_{j=1}^{s} (hF(Y_i^{[n]}a_{ij}) + Uy_i^{[n-1]} \]
(62)
as well as the output approximations;

\[ y_i^{[n]} = \sum_{j=1}^{s} (hF(Y_i^{[n]}B_{ij}) + V y_i^{[n-1]} \]
(63)

5 Numerical Experiments

The problems that have been chosen are part of the DETest set of problems [16]; a set of standard test problems were suggested for testing ODE solvers. Solutions have all been done with variable stepsize.

A: Logistics Curve: Single Equations

\[ y' = \frac{y}{4}(1 - \frac{y}{20}); \quad y(0) = 1 \]

Exact Solution: \( y_E(t) = \frac{20e^{\frac{t}{4}}}{19 + e^{\frac{t}{4}}} \)

B: The Radioactive Decay Chain: Moderate Systems

\[
\begin{bmatrix}
y_1' \\
y_2' \\
\vdots \\
y_9' \\
y_{10}'
\end{bmatrix} = \begin{bmatrix}
-1 & 0 & \cdots & 0 \\
1 & -2 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & 2 & \cdots & \cdots & \cdots \\
0 & \cdots & 9 & 0 & \cdots & \cdots \\
\end{bmatrix} \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_9 \\
y_{10}
\end{bmatrix}
\]
(64)

with \( y(0) = [1, 0, 0, \cdots , 0]^T \).

C: A Non-linear Chemical Reaction Problem

\[ \begin{align*}
y_1' &= -y_1 , \quad y_1(0) = 1 , \\
y_2' &= y_1 - y_2^2 , \quad y_2(0) = 0 , \\
y_3' &= y_2^2 , \quad y_3(0) = 0 .
\end{align*} \]
(65)

D: A Problem Derived From Van der Pol’s Equation

\[ \begin{align*}
y_1' &= y_2 , \\
y_2' &= (1 - y_1^2)y_2 - y_1 , \quad y_1(0) = 2 , \\
y_2' &= (1 - y_1^2)y_2 , \quad y_2(0) = 0 .
\end{align*} \]
(66)

6 Discussions of Results

Figures 1 to 12 (see, next pages) show the results when the ARK methods we have derived in this paper, are used to solve problems A – D. From the errors given by these methods, they may be considered as very good approximations of the exact results and we can say that they performed very well.

7 Conclusion

In this paper an effort has been made to present this unique class of methods and show that they are quite viable and reliable for solving not just single ODEs but large system as well. For a more detailed analysis of Ark methods see [17].
Figure 1: Solving Problem A using ark4a and ark4b

Figure 2: Solving Problem A using ark45a and ark45b

Figure 3: Solving Problem A using ARK5a and ARK5b

Figure 4: Solving Problem B using ARK4a and ARK4b

Figure 5: Solving Problem B using ARK45a and ARK45b

Figure 6: Solving Problem B using ARK5a and ARK5b
Figure 7: Solving Problem C using ARK4a and ARK4b

Figure 8: Solving Problem C using ARK45a and ARK45b

Figure 9: Solving Problem C using ARK5a and ARK5b

Figure 10: Solving Problem D using ARK4a and ARK4b

Figure 11: Solving Problem D using ARK45a and ARK45b

Figure 12: Solving Problem D using ARK5a and ARK5b
Special Acknowledgement

The authors wish to thank the ISAT fund of New Zealand for funding the recent visit of Abraham Ochoche, to the Department of Mathematics, The University of Auckland, Auckland, New Zealand where he had a one month stimulating exposure under Prof. J. C. Butcher, who facilitated the visit. Also, to acknowledge the immense contributions and cooperation received from Dr. Nicolette Rattenbury, of the Manchester Metropolitan University, UK and Dr. Helmut Podhaisky of the Institute of Mathematics, Martin-Luther-University, Halle-Wittenberg, Germany. We also want to acknowledge Dr. Allison Heard of the Department of Mathematics, University of Auckland, New Zealand for her immense contributions that led to the success of the Auckland visit.

References: