ORDER-2 EXTRAPOLATED IMPLICIT RUNGE-KUTTA METHODS WITH SMOOTHING FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

AMIRA BINTI ISMAIL

DISSERTATION SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE (APPLIED MATHEMATICS)
(MASTER BY RESEARCH)

FACULTY OF SCIENCE AND MATHEMATICS
UNIVERSITI PENDIDIKAN SULTAN IDRIS

2015

ABSTRACT

The aim of this research is to study the efficiency of the order-2 extrapolated implicit Runge-Kutta methods. The order-2 methods being considered are the implicit midpoint (IMR) and implicit trapezoidal (ITR) rules. These methods are applied with the polynomial and rational extrapolations actively and passively with smoothing to improve the accuracy. In order to reduce the round-off errors, a technique known as compensated summation with simplified Newton is implemented in all the numerical codes. The results are given based on numerical experiments that are carried out using MATLAB software. The findings showed that passive polynomial extrapolation by the IMR and ITR are more efficient than with active and passive rational extrapolation. The smoothing technique with extrapolation by the IMR and ITR gives better behavior than without the smoothing technique. It is therefore concluded that in solving chemistry, linear and nonlinear chemical and logistic curve problems, passive polynomial extrapolation with smoothing (PPXS) by the IMR gives better efficiency than the passive and active polynomial extrapolation by the 2-stage Radau IIA method (R2PX) and (R2AX). However, for higher dimensional nonlinear problems, R2AX and R2PX can be as efficient as the PPXS. The implication of this study is that, PPXS that has a cheaper implementation cost can be a very efficient method in solving linear and nonlinear stiff problems when compared with other higher order methods. Therefore, it is recommended to apply IMR with smoothing and extrapolation in the future research for comparison involving lower order methods.

KAEDAH TERSIRAT RUNGE-KUTTA PERINGKAT-2 YANG DITENTULUAR DENGAN TEKNIK PELICINAN BAGI MENYELESAIKAN PEMBEZAAN PERSAMAAN BIASA

ABSTRAK

Tujuan kajian ini dijalankan adalah untuk menentukan kecekapan peringkat-2 tersirat Runge-Kutta yang ditentuluar. Kaedah peringkat-2 yang dikaji adalah petua titik tengah tersirat (IMR) dan petua trapezoidal tersirat (ITR). Bagi meningkatkan ketepatan, kaedah ini digunakan dengan tentuluar jenis polinomial dan nisbah secara pasif dan aktif berserta dengan teknik pelicinan. Untuk mengurangkan ralat pembundaran, satu teknik yang dikenali sebagai lebihan penjumlahan menggunakan kaedah Newton yang telah dipermudahkan telah dilaksanakan dalam semua kod. Dapatan kajian yang diperolehi adalah berdasarkan ujikaji secara berangka yang dijalankan menggunakan perisian MATLAB. Hasil daripada dapatan kajian menunjukkan bahawa, pasif polinomial dengan penentuluaran bagi IMR and ITR memberikan keputusan yang paling cekap berbanding dengan tentuluar jenis nisbah secara aktif dan pasif. Teknik pelicinan dengan tentuluar bagi IMR dan ITR juga memberikan keputusan yang lebih baik berbanding dengan tentuluar tanpa teknik pelicinan. Kesimpulannya, dalam penyelesaian masalah kimia linear dan bukan linear serta masalah lengkung logistik, tentuluar pasif polinomial dengan pelicinan (PPXS) bagi IMR memberikan kecekapan yang paling tinggi berbanding dengan tahap dua Radau IIA yang menggunakan tentuluar pasif (R2PX) dan aktif (R2AX). Namun begitu, bagi penyelesaian masalah berdimensi tinggi, R2PX dan R2AX boleh memberikan kecekapan yang hampir sama dengan PPXS. Implikasi daripada kajian ini adalah PPXS yang mempunyai kos pelaksanaan yang lebih mudah boleh memberikan kecekapan yang sangat tinggi bagi penyelesaikan masalah kaku linear dan bukan linear jika dibandingkan dengan kaedah peringkat tinggi yang lain. Oleh yang demikian, adalah dicadangkan penggunaan IMR dengan teknik pelicinan dan tentuluar digunakan sebagai bandingan dengan kaedah peringkat rendah yang lain dalam kajian akan datang.

CONTENTS

LIST OF TABLES

Tables No. **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Numerical ODE

Ordinary differential equations (ODE) frequently occur as mathematical models in many branches of science, engineering and economy. Unfortunately it is seldom that these equations have solutions that can be expressed in closed form, so it is common to seek approximate solutions by means of numerical methods. Nowadays this can usually be achieved very inexpensively to high accuracy and with a reliable bound on the error between the analytical solution and its numerical approximation.

1.1.1 Ordinary Differential Equations

Physical, biological or chemical processes often use differential equations to solve daily problems. Ordinary differential equation, abbreviated ODE is a branch of differential equations. It involve derivatives of unknown solutions. All derivatives in the differential equations are with respect to a single independent variable. If the derivatives of differential equations are respect to more than one independent variables, then it is called partial differential equations.

Consider an ODE in the form of

$$y' = f(x,y), \quad y(x_0) = y_0, \quad f : [x_0, x_n] \times \mathbb{R}^N \to \mathbb{R}^N. \tag{1.1}$$

In this equation, x is the *independent* or time variable and y is the *dependent* variable. Function f is used to determine the unknown function y satisfying ODE. x_0 is the initial time, y_0 is the initial value, $\mathbb{R}^N$ is a set of real number and N is a set of positive integers. If the value of x_0 and y_0 are given, then equation (1.1) is known as initial value problems.

Although ordinary differential equation can be solved analytically like separable variable method, factorization method, substitution method and many more, but it is difficult to solve nonlinear equations especially higher order equations analytically. In this case numerical method is preferable. Hence, numerical method can be used to get the approximation of the solution when the exact solution is unknown.

Some famous numerical methods are Runge-Kutta methods, linear multistep methods and general linear methods. Runge-Kutta method is a one step methods due to Runge, Heun and Kutta (Butcher, 2008).

Linear multistep method on the other hand is an extended of Euler method by allowing the approximation solution at a point to depend on the solution values and derivative values at several previous step values. It was introduced by Bashforth and Adam (1883) for solving practical problems of capillary action.

General linear methods was introduced by Butcher (1966). It is used to obtain a general formulation of methods that possess the multivalue attributes of linear multistep methods as well as the multistage attributes of Runge-Kutta methods. It is also known as multistage-multivalue methods. Multivalue method is a method that collect input in vectors forms at the beginning of step and a similar collection is passed on as an output from the current step and as input into the following step. Multistage method is a computation in forming the output quantities.

In this research, only Runge-Kutta method will be focused. A brief introduction to Runge-Kutta methods is given in Subsection 1.1.2.

Certain applications in science and engineering involve completing physical phenomena with having widely different time scales especially problems involving combustion, energy conservation, temperature and density conditions (Faou et al., 2004) and (Kadoura et al., 2014). Stiffness is a special problem that can arise in the solution of ODE. A stiff system is one involving rapidly changing components together with slowly changing ones. A stiff problem is often referred as a problem with "widely differing time constant" or as a system "with a large Lipschitz constant" (Hall and Watt, 1976). If the derivatives $\frac{\delta f}{\delta y}$ are continuous and bounded, the Lipschitz constant is defined as

$$L = ||\frac{\delta f}{\delta y}|| > \rho\left(\frac{\delta f}{\delta y}\right),$$

where λ is defined as eigenvalues of $\frac{\delta f}{\delta y}$ and ρ is defined as

$$\rho = |\lambda_i|, i = 1, 2, ..n.$$

To understand stiffness, consider the Dahlquist test equation (Wanner, 2006) as given in equation (1.2),

$$y' = qy, \quad y(x_0) = y_0, \tag{1.2}$$

where q should be focused because it is a negative real part of a complex number. When $Re(q)$ is small and negative, it corresponds to a slow decaying component, whereas when $Re(q)$ is large and negative it corresponds to a rapidly decaying components. A problem is said to be stiff if the presence of the large negative $Re(q)$'s results in using a much smaller stepsize as requested by stability (Gorgey, 2012) and (Butcher, 2008).

For any numerical methods there are errors that can either destroy the solutions or contributes to the solutions. The good errors can be divided into two types. These errors are local and global errors. Local error is error committed by the method in a single step when the values at the beginning of that step are assumed to be exact. Global error is defined as accumulation of local errors after n steps. The accumulation does not mean the summation of local error at each n steps but it is bounded by the sum of the bounds on the local errors. Local error, l_n can be written as

$$l_n = u_n(x_n) - y_n, \tag{1.3}$$

where, u_n is the solution curve and y_n is exact solution curve. The global error, ε_n is given by

$$\varepsilon_n = y(x_n) - y_n,$$

where $y(x_n)$ is solution curve at n steps. With equation (1.3), ε_n becomes

$$\varepsilon_n = y(x_n) - u_n(x_n) + l_n. \tag{1.4}$$

where ε_n is the actual error after n steps. Hence, there are two components of global error, one is due the to local error at the present step and the other is due to the local error at the previous steps.

The bad errors are the roundoff errors. These errors if it is not well taken can destroy the numerical solutions. Detailed about roundoff errors is given in Chapter 3.

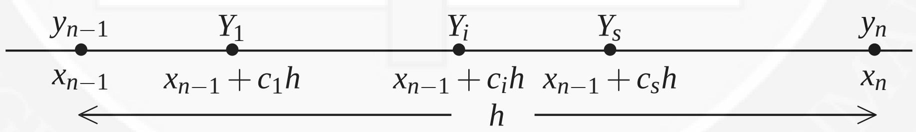
1.1.2 Introduction to Runge-Kutta Methods

In general Runge-Kutta (RK) method is defined as

$$Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f\left(x_{n-1} + c_j h, Y_j\right) \quad i = 1,2,\ldots,s, \tag{1.5a}$$

$$y_n = y_{n-1} + h \sum_{j=1}^{s} b_j f(x_{n-1} + c_j h, Y_j) \quad j = 1,2,\ldots,s. \tag{1.5b}$$

The general formula is a one step method that can be illustrated schematically with the following diagram:



where Y_i represent the internal stage values and y_n represent the update of y at the n^{th} step. Assumed that the row-sum condition holds:

$$c_i = \sum_{j=1}^{s} a_{ij}, i = 1,2,\ldots,s. \tag{1.6}$$

General formula (1.5a) and (1.5b) can be displayed by a partitioned tableau known as

Butcher tableau of the form

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array},$$

where the vector c is the vector of abscissae that indicates the positions within the steps of the stage values, the matrix A are the coefficients used to find the internal stages, s using linear combinations of the stage derivatives and the vector b represents the quadrature of weight indication of the approximation to get the solution that depends on the derivatives of the internal stages.

Runge-Kutta methods can be divided into two main types according to the style of the matrix A (Butcher, 2008). If matrix A is strictly lower triangular, it is called explicit and else the methods is called implicit. Implicit method can be divided into four categories. The categories are fully implicit if matrix A is not lower triangular, semi-implicit if matrix A is lower triangular with at least one non-zero diagonal element, diagonal implicit if matrix A is a lower triangular with all elements are equal and non-zero and the last is singly implicit if matrix A is a non-singular matrix with single eigenvalue.

There are many types of explicit and implicit methods. Some explicit methods are Euler method, explicit midpoint rule method, explicit trapezoidal method and higher order explicit methods. Example of implicit methods are the implicit Euler method, implicit midpoint rule, implicit trapezoidal rule and other higher order implicit methods. This research is only focusing on implicit midpoint and implicit trapeziodal rules. Table 1.1 shows the family of implicit Runge-Kutta methods of order-2 (Butcher, 2008).

The defining equations for the IMR and ITR are given in equation (1.7), equation (1.8) and equation (1.9), equation (1.10) and equation (1.11) respectively.

Table 1.1

Butcher Tableau for IMR and ITR

Implicit Midpoint Rule	Implicit Trapezoidal Rule
$\begin{array}{c\|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$	$\begin{array}{c\|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$

$$Y = y_{n-1} + \frac{h}{2} f\left(x_{n-1} + \tfrac{h}{2}, Y\right), \tag{1.7}$$

$$y_n = y_{n-1} + hf(x_{n-1} + \tfrac{h}{2}, Y). \tag{1.8}$$

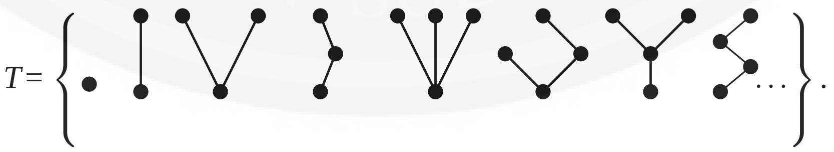
$$Y_1 = y_{n-1}, \tag{1.9}$$

$$Y_2 = y_{n-1} + \frac{h}{2} f\left(x_{n-1}, Y_1\right) + \frac{h}{2} f\left(x_{n-1} + h, Y_2\right), \tag{1.10}$$

$$y_n = Y_2. \tag{1.11}$$

In the following subsections, the theory of Runge-Kutta methods such as the elementary weights, elementary differentials and conditions for order are given for the set of rooted trees.

Let T denote the set of rooted trees as given in Butcher (2008)

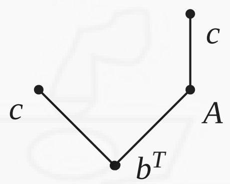


The rooted trees defined above are given up to order-4. Based on this set of rooted trees, the construction of the elementary weights, elementary equations and conditions for order are defined in the next subsections.

Elementary Weights

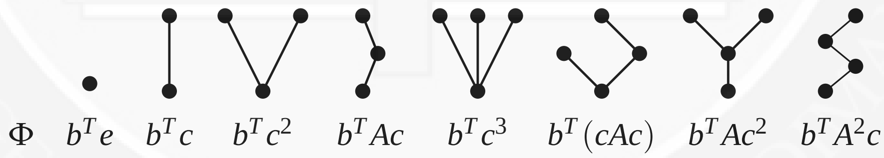
A tree can be labeled using the coefficients A, b^T and c. For each tree t defined an expression in term of the coefficient for the method by associating with each edge of the tree as the coefficient matrix A. Each vertex of the tree as the component wise product of all upward growing vectors, with the convention that an empty product is the vector e with each component equal to 1. Finally, the root of the tree is the operation on forming an inner-product with the vector b^T. Corresponding to each t is a real number called the elementary weights.

For example, consider a tree with four vertices:



$$\Phi(t) = b^T cAc$$

where b^T is the inner product and cAc show the component wise product.The summary of the elementary weights for 8 trees is given below as defined by Butcher (2008).



Elementary Differentials

To derive the elementary differentials consider the numerical solution of an autonomous differential equation system given by

$$y' = f(y(x)).$$

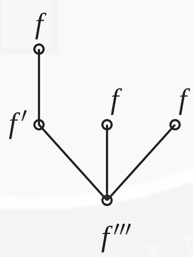
The higher differentiation can be calculated using repeated differentiation and chain rule. For example up to third derivatives is given as

$$y'' = f'(y(x))y' = f'(y(x))(f(y(x))),$$
$$y''' = f''(y(x))(f(y(x)), f(y(x))) + f'(y(x))(f'(y(x))(f(y(x)))).$$

For compactness of notation, it is easier to write $\mathbf{f} = f(y(x)), \mathbf{f}' = f'(y(x)), \mathbf{f}'' = f''(y(x))$, and so on. Therefore, the elementary differentials up to the fourth derivatives is given below:

$$y' = \mathbf{f},$$
$$y'' = \mathbf{f}'\mathbf{f},$$
$$y''' = \mathbf{f}''(\mathbf{f},\mathbf{f}) + \mathbf{f}'\mathbf{f}'\mathbf{f},$$
$$y^{(4)} = \mathbf{f}'''(\mathbf{f},\mathbf{f},\mathbf{f}) + \mathbf{3f}''(\mathbf{f},\mathbf{f}'\mathbf{f}) + \mathbf{f}'\mathbf{f}''(\mathbf{f},\mathbf{f}) + \mathbf{f}'\mathbf{f}'\mathbf{f}'\mathbf{f}.$$

Elementary differentials are related to the rooted-trees. Trees can be constructed from the derivatives. Consider the tree with 4 vertices.



The elementary differential for the tree with 4 vertices is given by

$$F(t)(y(x)) = \mathbf{f}'''(\mathbf{f}'\mathbf{f},\mathbf{f},\mathbf{f}).$$

Comparison of successive term in Taylor series expansions of computed solution with the exact solutions is used to derived the order conditions of the Runge-Kutta methods. The order conditions are used to investigate the error in carrying out a single step of a Runge-Kutta method. The exact solution at x_n, $y(x_n) = y(x_0 + h)$ up to order p is represented by Taylor series expansions.

The formal Taylor expansion of the solution at $x_0 + h$ defined in Butcher (2008) is given by

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{\alpha(t) h^{r(t)}}{r(t)!} F(t) y_0, \qquad (1.12)$$

where $t \in T$, $r(t)$ is the order of t, $\alpha(t)$ is the number of ways of labeling the tree with an ordered set and $F(t)y_0$ is the elementary differential. $\alpha(t)$ is defined by

$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)}. \qquad (1.13)$$

For all $t \in T$, using the known formula for $\alpha(t)$ given in equation (1.13), equation (1.12) which is the Taylor series expansion of the exact solution is given by

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t) y_0. \qquad (1.14)$$

where $\gamma(t)$ is known as the density of t.

Now to derive the order condition, one need to compare the Taylor series expansion of the exact solution with the Taylor series expansion of the numerical approximations. As given in (Butcher, 2008), the formal Taylor expansion of the numerical approximation to the solution at $x_0 + h$ is

$$y_1 = y_0 + \sum_{t \in T} \frac{\beta(t) h^{r(t)}}{r(t)!} \Phi(t) F(t) y_0, \qquad (1.15)$$

where $\beta(t)$ is the number of ways labeling the tree with an unordered set. Using the known formula for $\beta(t)$

$$\beta(t) = \frac{r(t)!}{\sigma(t)},$$

equation (1.15) becomes

$$y_1 = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)} \gamma(t) \Phi(t) F(t)(y_0). \tag{1.16}$$

Comparing the exact and the numerical solution, equation (1.14) with equation (1.16) gives

$$\Phi(t) = \frac{1}{\gamma(t)},$$

for all trees such that $r(t) \le p$ which is known as the **order conditions**.

The computation of the order conditions up to order-4 is given in Table 1.2.

Table 1.2

Computation of order conditions

t								
$r(t)$	1	2	3	3	4	4	4	4
σ (t)	1	1	2	1	6	1	2	1
γ (t)	1	2	3	6	4	8	12	24
α (t)	1	1	1	1	1	3	1	1
β (t)	1	2	3	6	4	24	12	24
$F(t)$	f	$f'f$	$f''(f,f)$	$f'f'f$	$f^{(3)}(f,f,f)$	$f''(f,f'f)$	$f'f''(f,f)$	$f'f'f'f$

where

$r(t)$	Order of t, it is convenient with number of vertices
$\sigma(t)$	Symmetry of t, it is convenient with order of automorphism group
$\gamma(t)$	Density of t
$\alpha(t)$	Number of ways of labeling with an ordered set
$\beta(t)$	Number of ways labeling with an unordered set
$F(t)y_0$	Elementary differential

The higher the order the more number of order conditions need to be derived. This can become unmanageable. For this reason, simplifying assumptions are introduced to simplify the order conditions. They are due to Butcher (1963).

In matrix form, the order condition can be simplified as

$$\begin{aligned} B(p): &\quad b^T c^{k-1} = \frac{1}{k}, \quad k = 1,\dots,p, \\ C(q): &\quad Ac^{k-1} = \frac{c^k}{k}, \quad k = 1,\dots,q, \\ D(r): &\quad b^T C^{k-1} A = \frac{1}{k}[b^T - b^T C^k], \quad k = 1,\dots,r. \end{aligned} \tag{1.17}$$

where C=diag($c_1,\dots,c_s$). The $B(p)$ conditions refer to the bushy trees such as  and . The minimum of p and q when $B(p)$ and $C(q)$ hold is called the stage order.

1.1.3 Stability of Runge-Kutta Methods

Stability concerns the behavior of solutions near an equilibrium point in the long term. Stability of RK method gives

$$R(z) = 1 - zb^T(I - zA)^{-1}e,$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{ss} \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix},$$

and e is a vector of unit 1.

Definition 1 *A method whose stability region contains the whole left half-plane is called* **A-stable**.

In another sentence, implicit methods have stability function that are rational functions and there is a possibility that $|R(z)| \leq 1$ holds for $z \in \mathbb{C}^-$ (Butcher, 2008).

Definition 2 *A method is said to be L-stable if it is A-stable and if, in addition,*

$$\lim_{z \to \infty} R(z) = 0.$$

For implicit methods to be L-stable, the degree of the numerator must be less than the degree of the denominator (Butcher, 2008).

Although stability of Runge-Kutta methods often related to stiffness, one special type of Runge-Kutta methods that is considered is the one that satisfies the symmetric properties. This type of Runge-Kutta method which also known as symmetric Runge-Kutta methods are discussed in Section 2.1. Symmetric methods plays an important role when applying extrapolation therefore it is important to study about them.

1.2 Problem Statement

Iterated methods or usually called numerical methods such as Runge-Kutta methods, general linear methods and multistep methods are always applied in solving linear and nonlinear problems. Higher order methods such as Radau IIA of order-5 or 3-stage Gauss method of order-6 are always preferable since these methods are of higher order and therefore will give greater accuracy than lower order methods. However, instead of using higher order methods, another way to get greater accuracy is by applying extrapolation. Lower order methods are always preferable since they are cheaper to implement than higher order methods.

Lower order of Runge-Kutta methods are divided into two types, implicit and explicit. Implicit method have higher stability than the explicit ones. It need fewer stages for the same order when compared to the explicit method (Butcher, 2008). Implicit RK methods also play important role in solving stiff problems (Butcher, 1975).

Implicit midpoint (IMR) and implicit trapezoidal (ITR) rules are chosen because they are symmetric and therefore special especially when applying extrapolation. For example, if extrapolation technique is applied with Euler method, the order of the Euler method will give 2. When extrapolation is applied with IMR, the order of the method will increase up to order 4 (Richardson, 1911).

Extrapolation is a process of the deferred approached to the limit (Richardson and Gaunt, 1972). There are two ways of applying extrapolation such as active and passive extrapolation. Active extrapolation is applied when the extrapolated values are used to propagate the next step. Passive extrapolation can be applied when the extrapolated values are not used in any subsequent computation. There are also two types of applying extrapolation which are polynomial and rational extrapolation. Deuflhard (1985) investigated that polynomial extrapolation gives more efficiency than the rational ones.

In addition to these, extrapolation with smoothing had shown to improve the behaviour of the numerical solutions (Gragg, 1964). Smoothing technique can dampen the oscillatory behaviour of the numerical solutions. Only one effect from dampened oscillatory, the numerical solutions by the method is improved.

1.3 Research Objectives

The objective of this research are

1. To study the efficiency of the symmetric Runge-Kutta methods especially for implicit midpoint and implicit trapezoidal rules in solving stiff and nonstiff linear and non-linear problems numerically.
2. To apply active and passive extrapolation with and without smoothing technique of the implicit midpoint and implicit trapezoidal rules numerically.
3. To compare the efficiency of the extrapolated implicit midpoint and implicit trapezoidal rules with smoothing with the 2-stage Radau IIA method.
4. To compare the efficiency of the polynomial and rational extrapolation in solving stiff and non-stiff linear and nonlinear problems.

1.4 Significance of Research

At the end of these research, it is hoped that:

1. The readers know that the efficiency of the symmetric Runge-Kutta methods especially for the implicit midpoint and implicit trapezoidal rules in solving stiff linear and nonlinear problems can be computed numerically and analyzed theoritically for the Prothero-Robinson problem.
2. Active and passive modes with polynomial and rational extrapolation can be applied