JMSBI, Volume 1, Issue 1,
October. 2016, pages 53-61
http://www.ibii-us.org/Journals/JMSBI/
ISBN 2472-9264 (Online), 2472-9256 (Print)
DOI https://doi.org/10.5281/zenodo.376751

An Adaptive Time Step Scheme based on Taylor's Remainder Term

Henry Foust

Department of Mathematics, University of Saint Thomas, Houston, TX, USA
fousth@stthom.edu

Received: June 06, 2016; revised: October 04, 2016; published: October 07, 2016

Abstract

An adaptive time step method was developed based on the Taylor series remainder term associated with Euler's method, which is utilized to solve initial value problems involving ordinary differential equations. The accuracy and stability of the developed method was determined for three test cases where one of the test cases was stiff. It is also show that the accuracy of the developed method compares well with Runge Kutta 2. In future research, this method will be applied to explicit and implicit versions of Runge Kutta 2 to include Calahan's method, which is a variation of the Rosenbrock's scheme.

Keywords: Runge Kutta; Adaptive Time Step; Error Control; Numerical Analysis

1. INTRODUCTION

The purpose of this paper is to present an adaptive time step method applied to Euler's schemes; this adaptive time step is based on the remainder term associated with the Taylor series representation of the RK scheme. Give the following initial value problem

$$\frac{dy}{dx} = f(x) \tag{1}$$

With $y(0) = y_0$

A possible numerical integration scheme is based on the idea that an RK method of order p has the following representation

$$y_{n+1} = y_n + \frac{y\prime h}{1!} + \frac{y\prime\prime h^2}{2!} + \cdots + \frac{y^p h^p}{p!} + R_p \tag{2}$$

Where R_p represents the remainder term and accounts for the terms truncated from the Taylor Series representation for y_{n+1}. The remainder term is also the local truncation error and represented as E_n with the following form

$$E_n = \frac{y^{p+1} h^{p+1}}{(p+1)!} \tag{3}$$

This work develops a method to estimate E_n and y^{p+1} and determine an adaptive time step, h_n, based on Equation 3. The developed h_n is then utilized within RK 1 (Euler's) scheme, which will be shown to be essentially as accurate as an RK-2 scheme.

The intent of this work is to lay a foundation for future research that will apply these ideas to a form of implicit RK scheme known as Calahan's method, which was expressly developed to address issues of stiffness. A stiff dynamic system is one where there are "fast" and "slow" dynamics as seen by eigenvalues that may differ by orders of magnitude [5].
Formally, stiffness can be defined as

$$S = \frac{max|Re\,(\lambda_i)|}{min|Re\,(\lambda_i)|} \tag{4}$$

Where the numerator is the value of the largest eigenvalue and the denominator is the value of the smallest eigenvalue. When S is greater than 100, the system is considered stiff [5].

Given in Figure 1, and as Equation 9, is a system that exhibits features of stiffness.

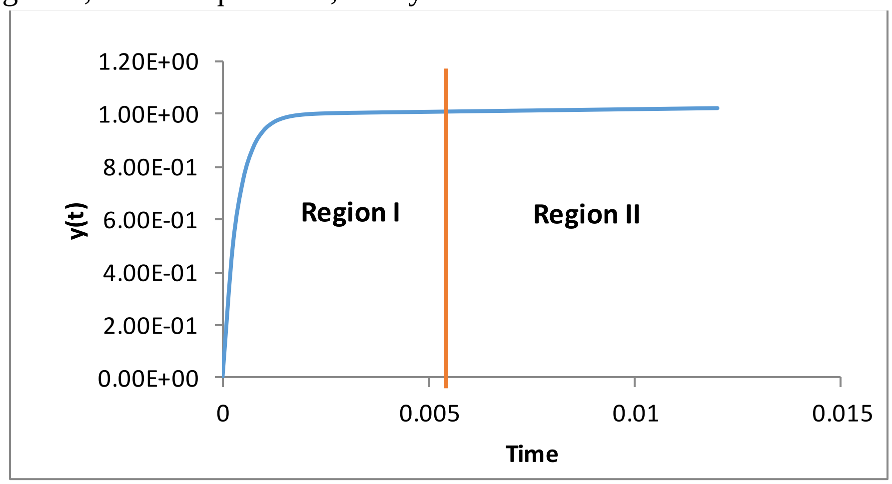

Fig. 1 Stiff Dynamic System

This figure has been divided into two regions given as Regions I and II. In Region I, and especially in the interval $t \in [0, .002]$, the slope of the tangent to the curve is very large and numerical integration schemes have difficulty integrating systems with large Jacobians, which introduce large errors unless a very small time step, h, is utilized.

For the proposed adaptive time step (see Figure 2), it can be seen that the h(i) correctly addresses this issue. For Region I, h(i) is small and for Region II, h(i) is large.

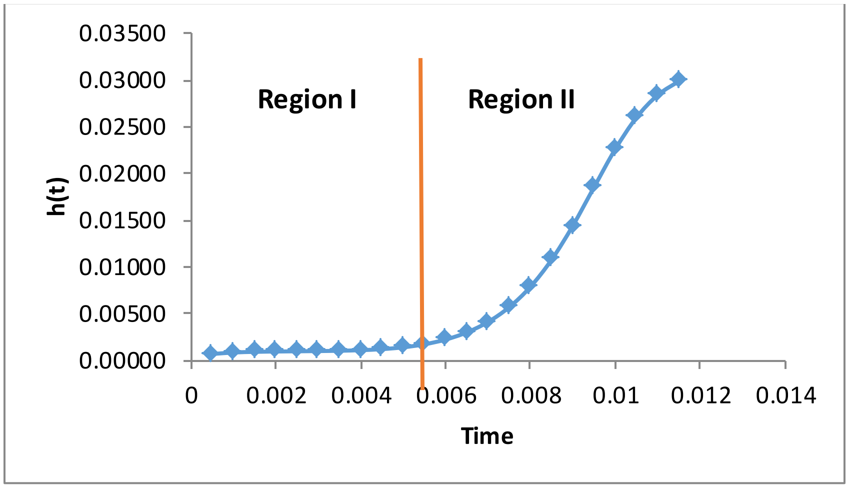

Fig. 2 Adaptive Time Step for System 1.

2 BACKGROUND

In Chapra and Canale [3], which presents a well-accepted method for adaptive time steps, the following sequence is utilized:

1. The local truncation error is estimated by two different methods (or results from a difference of using the same method with time steps of h and h/2)
2. The new step size is estimated as follows

$$h_{new} = h_{present} \left| \frac{\Delta_{new}}{\Delta_{present}} \right|^{\alpha} \tag{5}$$

Where h_{new} is the next step size, $h_{present}$ is the current step size, Δ_{new} is the desired accuracy, and $\Delta_{present}$ is the current local truncation error. There are additional equations to determine Δ_{new} and a scale factor given as y_{scale}. Other resources in this area include [1, 2, 4, 5, 6].

There are essentially two main methods that utilize some form of the above procedure: 1) Runge Kutta Fehlberg and 2) Dormand-Price [7]. More often than not, the Dormand-Price scheme is utilized and is the basis to the Matlab® differential equation solver known as ODE45.

The main difference between these two schemes and what is proposed is the following. The two methods mentioned above insure a certain level of local truncation error at each iteration whereas the proposed methods do not explicitly insure this. As a trade-off in favor of the proposed method, Runge-Kutta-Fehlberg and Dormand-Price require much more computational effort and may not have a global error convergence comparable with the proposed method. These considerations are the subject of a future paper.

In this paper, a similar approach is utilized where the first step is the same, but the second step is based on the remainder term associated with the Taylor series representation for the Euler's scheme and is given as Equation 7.

3 METHOD

3.1 RK-1 Scheme

Euler's method or Runge Kutta 1 is given as

$$y_{n+1} = y_n + f(y_n)h \tag{6}$$

And the adaptive time step associated with this scheme, and determined from the remainder term, is

$$\phi h_n = \sqrt{\frac{2|E_n|}{\left|{}^{df_n}/_{dy_n}\right|}} \tag{7}$$

Where ϕ is a tuning factor and may depend on the type of dynamic system to be solved.

3.2 Estimating E_n

One means to estimate the local truncation error, E_n, is to determine y_{n+1} using time step h and then recalculate y_{n+1} using time step ${}^{h}/_{2}$. The difference between $y_{n+1}(h)$ and $y_{n+1}\left(\frac{h}{2}\right)$ is a measure of the local truncation error [2, 4].

In Figures 3, the local truncation error (actual) and local truncation error (estimated) compare favorably.

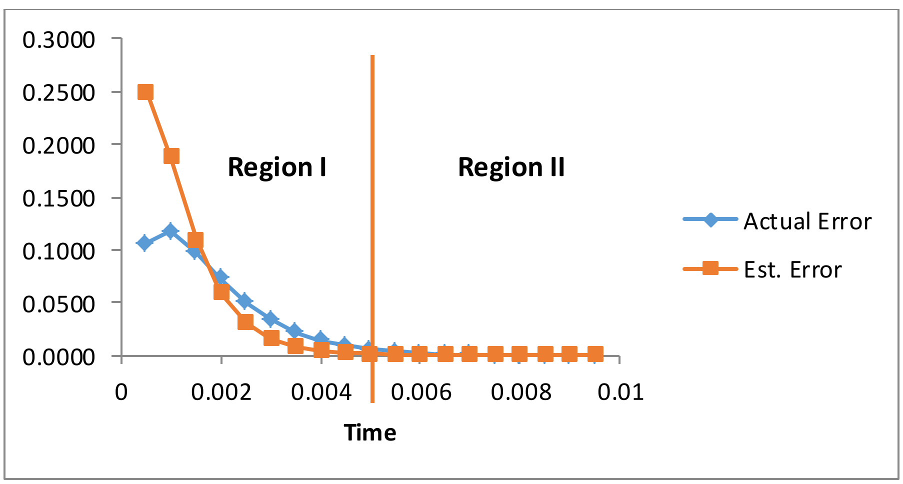

Fig. 3 E(i) versus Time for System 1

3.3 Estimating y^{p+1}

A central difference scheme was not utilized, but would improve the method, and is given as

1st Order

$$\frac{\Delta f}{\Delta t} = \frac{f(n+1)-f(n-1)}{2h} = \frac{f(n)-f(n-1)}{2h} \tag{8}$$

Instead, a forward difference was utilized. Equation 8 could be applied to the two iterations conducted at "h/2" to determine the local truncation error at each "n."

4 RESULTS

4.1 Dynamic Systems Considered

The three dynamics systems (Systems 1 to 3) considered below include one stiff system (System 1) and two non-stiff systems. For the systems where a solution is available (Systems 1 to 3), accuracy was determined for the four methods of numerical integration utilizing mean square error (MSE); see Table 1. The comparison was made for the half time step fixed step and adaptive step.

System	t(0)	t(f)	h(1), h(i)	ϕ	MSE
One, Adaptive	0	0.006	1.00E-06	1.618034	0.03
One, Fixed	0	0.006	3.00E-04	n/a	0.15
Two, Adaptive	0	.4	0.1	1.618034	0.002
Two, Fixed	0	2	0.1	n/a	0.43
Three, Adaptive	0	1	0.1	1.618034	0.1
Three, Fixed	0	2	0.1	n/a	2

*Both full step methods utilized 10 intervals for all systems.

Table 1 Parameter Conditions

4.2 Dynamic System I

$$\frac{dy}{dt} = -1000y + 3000 - 2000e^{-t} \tag{9}$$

With y(0) = 0 and analytic solution

$$y = 3 - .998e^{-1000t} - 2.002e^{-t} \tag{10}$$

Numerical solutions are given as Figures 4 and 5. From the mean square error given in Figures 4 and 5, it’s easy to see that the adaptive time step method provides more accurate answers.

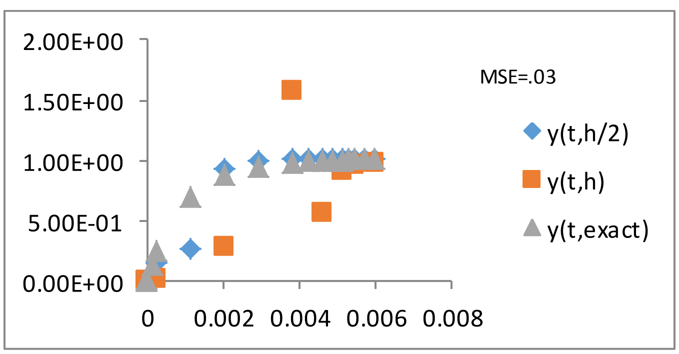

Fig. 4 Time vs. y(t) for adaptive time step, System 1

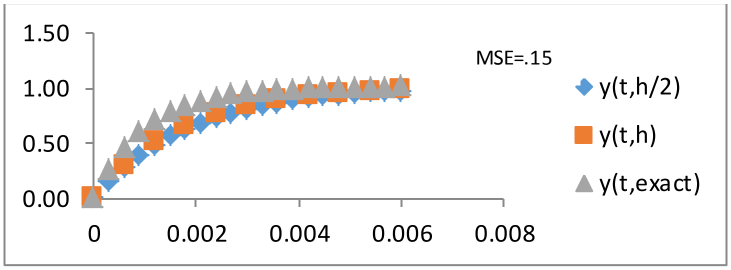

Fig. 5 Time vs. y(t) for fixed time step, System 1

4.3 Dynamic System II

Integrate both analytically and numerically

$$\frac{dy}{dt} = \sin(t) \tag{11}$$

For y(0)=0 with analytic solution

$$y(t) = 1 - \cos(t) \tag{12}$$

Numerical solutions are given as Figures 6 and 7. Again from the mean square error given in Figures 6 and 7, it's easy to see that the adaptive time step methods provide more accurate answers.

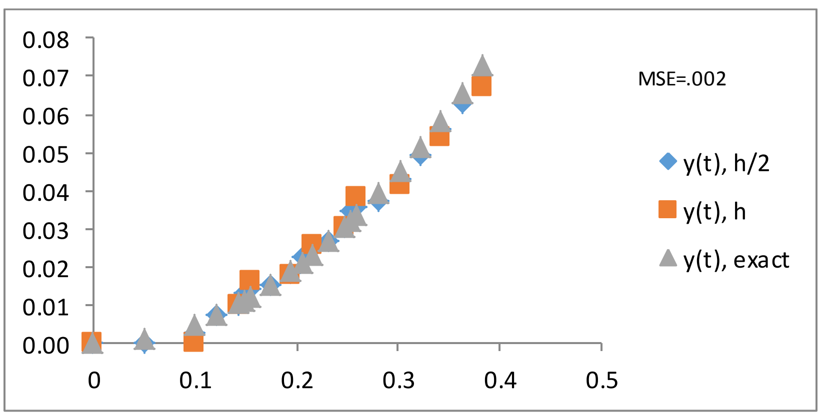

Fig. 6 Time vs. y(t) adaptive time step, System 2

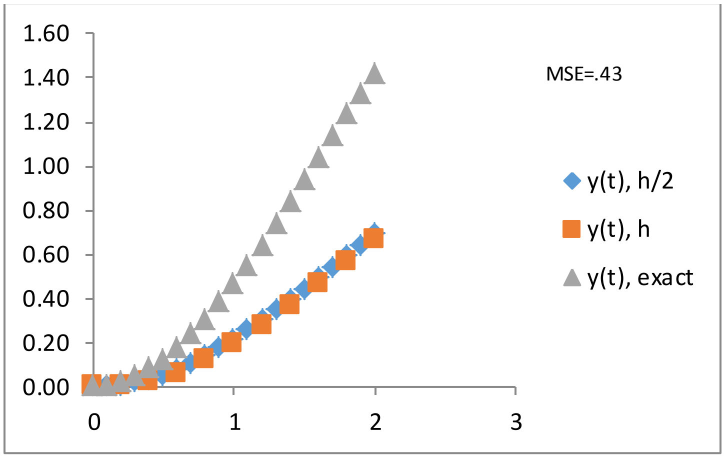

Fig. 7 Time vs. y(t) fixed time step, System 2

4.4 Dynamic System III

Integrate both analytically and numerically

$$\frac{dy}{dt} = 10 - y(t), y(0) = 1; \tag{13}$$

With analytic solution

$$y(t) = y_0 e^{-t} + 10(1 - e^{-t}) \tag{14}$$

Numerical solutions are given as Figures 8 and 9. Further, the mean square error for the adaptive time step method is much better than the fixed time step method.

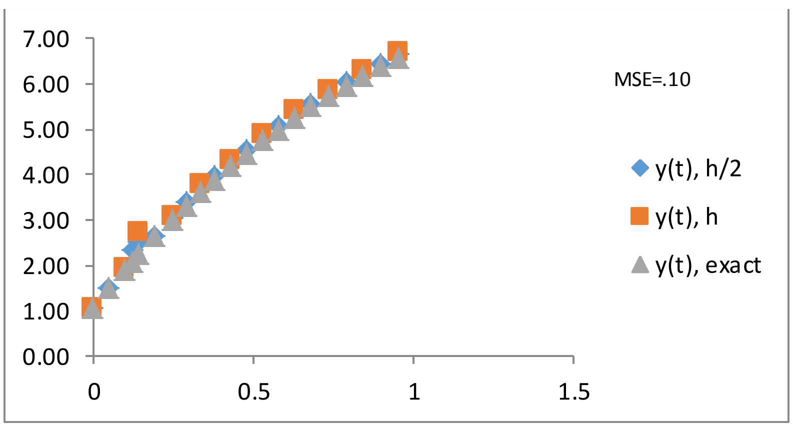

Fig. 8 Time vs. y(t), adaptive time step, System 3

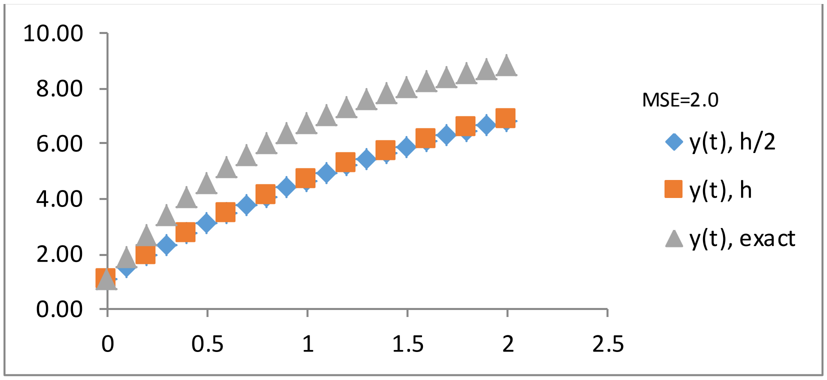

Fig. 9 Time vs. y(t), fixed time step, System 3

4.5 Comparison with RK-2

It is natural to wonder if adding in the Taylor remainder term associated with a Euler's scheme is essentially a RK-2 method. As seen in Figures 10 and 11 for Systems 1 and 3, the accuracy of the Adaptive Euler's scheme and RK-2 are comparable. Additionally, the MSE associated with each pair in Figures 10 and 11 are comparable.

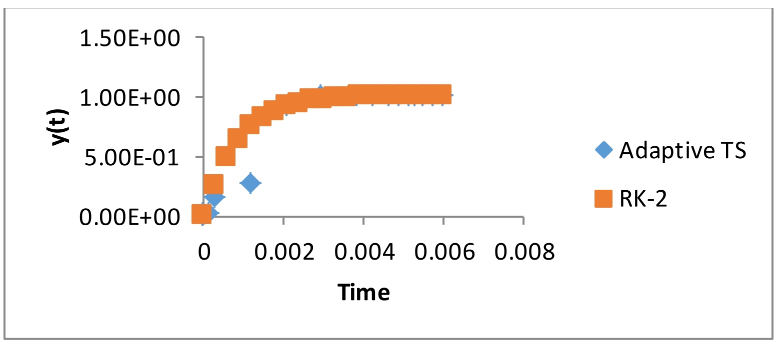

Fig. 10 Comparison between Adaptive Time Step Scheme and RK-2 for System I

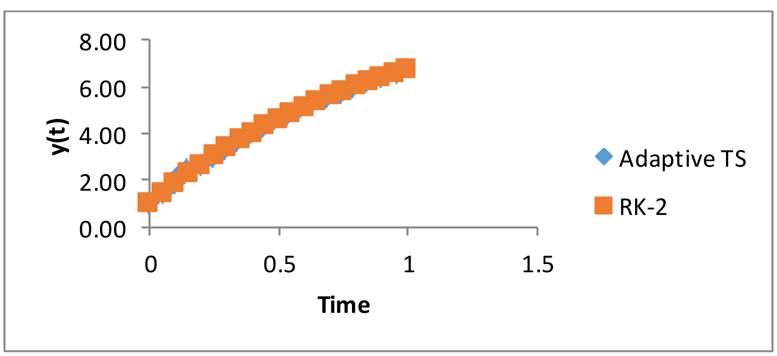

Fig. 11 Comparison between Adaptive Time Step Scheme and RK-2 for System III

5 DISCUSSION/CONCLUSIONS

An adaptive time step method was developed for Euler's scheme and is based on the remainder term associated with a Taylor series representation. The developed scheme was compared against a fixed time step Euler's scheme for three dynamics systems to include one that is stiff and showed higher accuracy. In fact, the accuracy of the developed method is comparable with RK-2 (see Figures 10 and 11). It is noted that the method's accuracy and numerical stability is dependent on the tuning factor and initial step sized used.

Future work will look at applying the developed scheme to RK-2, RK-4 and an implicit RK method known as Calahan's method. Additionally, the effects of initial step size and tuning factor on accuracy and stability of the scheme will be explored as well as looking at global error convergence when compared against other adaptive step schemes and computational effort.

ACKNOWLEDGEMENTS

The author would like to thank Brent Rankin and Fred Schauer at the Air Force Research Laboratory, Wright Pat-terson Air Force Base for the opportunity to do summer research through a U.S. Air Force Summer Faculty Research Fellowship where some of these ideas were first developed.

REFERENCES

[1] Aiken and Lapidus (1974). AIChE Journal, 20, pp. 368.

[2] J.C. Butcher (2008). Numerical Methods for Ordinary Differential Equations; John Wiley and Sons, New York, NY.

[3] Chapra and Canale (2010). Numerical Methods for Engineers; McGraw-Hill, Boston, MA.

[4] C.W. Gear (1971). Numerical Initial Value Problems in Ordinary Differential Equations; Prentice-Hall.

[5] Prokopakis and Selder (1981). "Adaptive Semi-Implicit Runge Kutta Method for Solution of Stiff Ordinary Differential Equations"; Ind. Eng. Chem. Fundam., 20, pp. 255 to 266.

[6] Seinfeld, Lapidus, and Hwang (1970). "Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations"; Ind. Eng. Chem. Fundam., 9, 2, pp. 266 to 275.

[7] Golberg, Oleg (2007). Accessed from https://dspace.mit.edu/bitstream/handle/1721.1/55903/18-034Spring-2007/NR/rdonlyres/Mathematics/18-034Spring-2007/Projects/euler.pdf on 10/4/2016.