Project 03

APSC 607 Fall 2017

Seth Goodman

November 12, 2017

1 Introduction

This project explores different numerical methods of solving two well-posed initial value problems (IVPs), y', at discrete points within a range $[a\ b]$ for a given initial value $y(a)$ as defined in Equation 1 and described in Burden and Faires (2010). The methods that will be tested are Euler's method (i.e., first order Taylor method), fourth order Runge-Kutta (i.e., fourth order Taylor method), and an implicit trapezoidal method (trapezoidal with Newton Iteration). The behavior and characteristics of these methods will be reviewed and their effectiveness evaluated based on the true solution to the IVP.

$$y' = \frac{dy}{dt} = f(t, y), \qquad a \leq t \leq b, \qquad y(a) = \alpha \tag{1}$$

All computations are performed using MATLAB using the code accompanying this report. Section 2 will present the methods used in MATLAB to explore functions. Section 3 contains the results and related outputs for each function, and **??** includes observations, discussion. All figures and tables found in this report are available in the output subdirectory of the accompanying zip file. Additionally, all code and figures found in the zip file can be accessed via GitHub[1]

[1] `https://github.com/sgoodm/apsc607/tree/master/project_03`

2 Methods

The two unique functions which will be explored in this project, Functions **A** and **B**, are defined by Equations 2 and 3, respectively.

$$y' = -9y \tag{2}$$

$$y' = 20(y - t^2) + 2t \tag{3}$$

The associated initial values for Functions A and B, are defined by Equations 4 and 5.

$$y(0) = e \tag{4}$$

$$y(0) = \frac{1}{3} \tag{5}$$

Both IVPs will be examined over the range $0 \leq t \leq 1$, using a baseline step size of $h = 0.1$.

To validate and compare each method for solving the IVPs, the true solution is required. The true solution is generated using the Symbolic Toolkit in MATLAB as seen in the example below for Function A.

```
syms y(t)
ode = diff(y,t) == -9*y;
cond = y(0) == exp(1);
sol = dsolve(ode, cond);
```

The true values for Functions A and B are compared to the results of integration using the Trapezoidal, Midpoint, and Simpson's Rules for a range of subintervals. The range of subinterval values, **n**, can vary depending on the function and rule tested, in order to ensure that an accuracy within a tolerance of 10^{-4} and 10^{-8} of the true value is achieved by at least one **n** value in the range.

The resulting vector of integral values across varying **n** can compared with the true integral value generated earlier, to produce an error vector. The error vector is used to identify the value of **n** (and thus **h**) at which the integration rule produced results that were accurate within a desired tolerance. No **n** value needs to be specified for the adaptive approach because, as seen in Subsection **??**, this approach iteratively checks (within the function) that a specified tolerance has been reached before producing a single output rather than using a set **n** value to produce a set of output values.

The remainder of this section will introduce the three composite numerical integration approaches (Trapezoidal, Midpoint, and Simpson's) and the adaptive approach using Simpson's Rule.

2.1 Euler's (first order Taylor)

page 266

2.2 Runge-Kutta (fourth order Taylor)

page 288

2.3 Implicit Trapezoidal (Trapezoidal with Newtonian Iteration)

page 248

3 Results

This section will review the results of the different integration rules for Functions A and B. An overview of these results can be seen in Table 1, which shows the the number of subintervals required to reach each tolerance specified, as well as the final error achieved. Figure **??** contains the true plots of both function, for reference.

Function	**Method**	**N:** 10^{-4}	**N:** 10^{-8}	**Minimum Error**
a	trapezoidal	768	65536	4.26887e-09
a	midpoint	1024	98304	8.53747e-09
a	simpsons	42	322	1.30384e-12
a	adaptive	24	278	-
b	trapezoidal	12	1082	1.29289e-09
b	midpoint	22	1522	2.58233e-09
b	simpsons	2	22	2.22044e-16
b	adaptive	2	16	-

Table 1: Results

3.1 Euler's Results

3.2 Runge-Kutta Results

3.3 Implicit Trapezoidal Results

4 Discussion and Conclusions

References

R. Burden and J Faires. *Numerical Analysis*. Brooks/Cole, 9th edition, 2010.