Numerical Methods for Model Rocket Altitude Simulation - A Comparative Study of Accuracy and Efficiency

Kenneth J. Karbon

NAR# 72175

Submitted to NARAM 40 Research and Development Competition

August, 1998

CONTACT

Kenneth J. Karbon
230 Hillview Terrace
Fenton, MI 48430

(H) (810) 750-2767
(W) (810) 236-0503

(H) karbon@tir.com
(W) pzd5fy@clc.gmeds.com

NAR# 72175
Huron Valley Rocket Society, NAR Section #463

Contents

INTRODUCTION

Predicting the altitude of model rockets has been discussed extensively for the past 35 years. Two approaches are available to the rocketeer: closed form exact solutions (the well known Fehskens-Malewicki equations) and numerical methods. From the author's literature research in these topics, there appears to be some apprehension in the model rocket community regarding the use of numerical methods. Even with today's high-speed digital computers and the numerous rocket simulation software packages available on the market, some hobbyists still feel that approximate numerical methods are "complex" or "inaccurate."

This paper presents a detailed study that will hopefully waylay those fears. The first part of the paper introduces some numerical methods applicable to solving the equations of model rocket flight. The second part of the research then demonstrates the use of these methods on a sample problem involving an Estes Alpha III model rocket and C6 engine. The methods will be compared amongst themselves as well as with the exact solution. Evaluations of accuracy, computer efficiency, and practicality will be presented. All the work done on this project is theoretical in nature. Experiments and tests were performed on a desktop computer.

Rocketeers with a background in engineering or mathematics should be familiar with many of the concepts presented here. However, this paper hopefully avoids excessive technical jargon and complex derivations such that all enthusiasts with knowledge of model rocket physics can benefit from the information given. The findings from this project may be especially useful to modelers interested in programming their own computer solutions to model rocket design. Rocketeers looking to purchase commercial software can also benefit from this information by using it to evaluate the different features of the software packages before they buy.

BACKGROUND AND PRIOR WORK

The Rocket Equation And Exact Solutions

For one-dimensional model rocket flight, the governing equation of motion is obtained by summing the forces in the vertical direction on the rocket:

$$thrust - weight - aerodynamic\ drag = mass \times acceleration$$

In differential form this equation and its initial conditions are written:

$$T - mg - \frac{1}{2}\rho C_D A \left(\frac{dx}{dt}\right)^2 = m\frac{d^2x}{dt^2}, \quad x(t_0) = 0\,AGL, \quad \frac{dx}{dt}(t_0) = 0 \tag{1}$$

where:

thrust, $T = T(t)$

mass, $m = m(t)$

air density, $\rho = \rho(x)$

drag coefficient, $C_D = C_D(\rho, \frac{dx}{dt})$

cross-sectional area, A

gravitational constant, g

vertical displacement, x

time, t

vertical acceleration, $\frac{d^2x}{dt^2}$

vertical velocity, $\frac{dx}{dt} = v$

Equation (1) is a second order, nonlinear, initial-value, ordinary differential equation.

The Fehskens-Malewicki exact closed form solution [1] to equation (1) has been widely used to predict a rocket's velocity and altitude during the coarse of flight. However, in order to make equation (1) tractable, the "F-M" equations assume constant values for $T, m, \rho,$ and C_D throughout the entire thrust phase or coast phase. For typical model rockets, these assumptions have been shown to be quite valid. However, for high power applications and high-performance competition rockets, these assumptions begin to fall apart and therefore may incorrectly model the true physics of the problem.

Over the years, a few enhancements have been made to the F-M equations that work around the need for the constant parameters listed above. Kuechler and Kelley [2] incorporated variable air density into the equations by choosing altitude instead of time as the independent variable. John Viggiano [3] suggested breaking the boost phase into several intervals to better represent a model rocket engine thrust curve. Larry Curcio [4] implemented a similar multi-interval thrust phase in his *DigiTrak* algorithm found in Chuck Gibke's *wRASP 2.0* freeware available on the Internet.

A problem with the multi-interval thrust phase approach is that during low thrust intervals, a model rocket's mass may exceed its thrust causing a negative to appear under the radical of the F-M equations. Viggiano overcomes this situation by deriving a new low-thrust equation. Curcio on the other hand, adjusts the gravitational constant during these conditions.

Numerical Methods

Numerical methods is a discipline of mathematics and engineering that came of age with the advent of the computer. World War II, the Cold War, and the Space Race demanded fast and accurate solutions to complex physical problems, and the computer was developed as the tool to perform these tasks. Unlike exact, analytical equations, numerical methods solve problems in many small steps or iterations, work well suited for a computer. In fact, practical engineering situations rarely have exact solutions and therefore demand numerical approximations. By today's computational standards, the ordinary differential equation of rocket motion is very easy to solve numerically.

Applied to model rocket altitude simulation, numerical methods do not require any constant values or solution adjustments. Variable thrust, mass, aerodynamic drag, and air density can all be accounted for in a straightforward manner, giving a more faithful representation of the model rocket's physics. Numerical approximations do generate small errors due to their derivations, repeated calculations, and computer round-off. (For this study, "error" is used to mean the difference between the numerical mathematical solution and the exact mathematical solution.) However, with the proper choice of method and parameters, these errors are virtually eliminated for all practical rocket (model, competition, and high-power) altitude predictions.

SAMPLE PROBLEM AND APPROACH

In this paper, various numerical methods will be presented to predict model rocket motion. Each numerical solution will be compared to the exact Fehskens-Malewicki calculations. In order to maintain a one-to-one comparison, the numerical solutions will make the same assumptions as the F-M equations, namely constant T, m, ρ, and C_D. The test case for evaluation is given in Table 1.

Table 1. Estes Alpha III and C6 Engine

Rocket Mass (kg)	0.034
Rocket Diameter (m)	0.0248
Drag Coefficient	0.6
Air Density (kg/m^3)	1.205
Gravitational Constant (m/s^2)	9.80665
Total Engine Mass (kg)	0.0222
Propellant Mass (kg)	0.0108
Average Thrust (N)	5.875
Thrust Duration (s)	1.45
Engine Mass During Thrust (kg)	0.0168 (0.0222-0.5*0.0108)

Rocket data was obtained from the 1997 Estes catalog. Engine data was obtained from the wRASP 2.0 database. The exact solution to this problem is shown in Figure 1 with the key parameters given in Table 2.

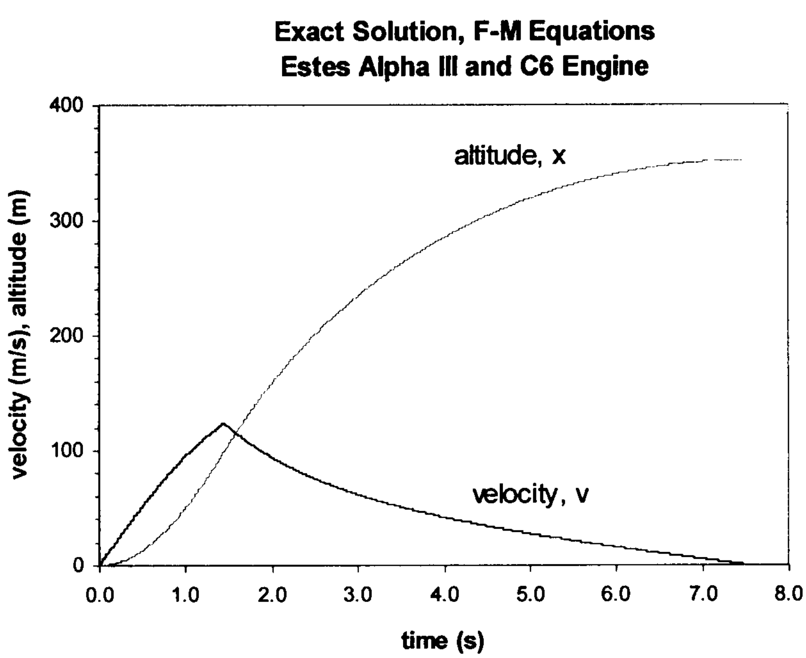

Figure 1.

Table 2. Exact Solution (F-M Equations)

Estes Alpha III and C6 Engine

Burnout Velocity (m/s)	123.4830
Burnout Altitude (m)	99.43599
Coast Time (s)	6.089279
Coast Distance (m)	252.5951
Total Time to Apogee (s)	7.539279
Maximum Altitude (m)	352.0311

Simple numerical routines and the F-M equations were programmed in FORTRAN 77 (see appendix for sample listing). For each method, the numerical solution and exact solution for velocity and altitude were calculated throughout the rocket flight time. The error is defined as:

$$error = numerical\ value - exact\ value$$

To minimize computer round off, the FORTRAN routines were programmed in double precision, and the calculations were performed on a 13 decimal digit UNIX workstation. The results will be presented to 7 decimal digits however, since this is the machine accuracy of a typical 16 bit personal computer application.

PART I. OVERVIEW OF NUMERICAL METHODS TO SOLVE THE MODEL ROCKET EQUATION

This section will give moderately detailed information on how numerical methods are developed and classified. Rocketeers may have seen some of the following terminology in their freeware or commercial software packages. Where appropriate, relevance to model rocket altitude simulation will be highlighted.

To solve numerically, equation (1) is first separated into a *system* of two *first order* equations that must be solved *simultaneously*:

$$x' = v = f(v), \quad x(0) = 0 \qquad \text{(altitude equation, 1a)}$$

$$v' = \frac{1}{m}\left(T - mg - \frac{1}{2}\rho C_D A v^2\right) = g(t, v, x), \quad v(0) = 0 \qquad \text{(velocity equation, 1b)}$$

The objective of a numerical method for solving (1a, 1b) is to transform a calculus problem into an algebra problem. One such approach, *the finite difference method*, makes the transformation as follows [5]:

1. Discretize the solution domain
2. Approximate the exact derivatives in the differential equations with algebraic formulas called *finite difference approximations* (FDAs)
3. Plug the FDAs into the differential equations to create *finite difference equations* (FDEs) which are systematically marched through time to solve for the dependent variables.

1. Solution Domain

The time solution domain must be *discretized* by a one-dimensional set of grid points (Figure 2). The numerical solution for x and v will be obtained at these points. The points are equally spaced by a time step of Δt. As the following sections will point out, the choice of Δt is crucial to the numerical solution.

```
| Δt |
●----●----●-- ... --●----●----●----●------> time, t
0    1    2      n-1   n   n+1  n+2
```

Figure 2. Time Domain

2. Finite Difference Approximations

One way of approximating a derivative is through a Taylor series expansion. The infinite Taylor series can be used to evaluate a function, given its derivatives and its value at some base point. Consider the function $f(t)$, known at time t_0, and with derivatives f', f'', etc. The Taylor series is:

$$f(t) = f(t_0) + \frac{1}{1!} f'(t_0)(t - t_0) + \frac{1}{2!} f''(t_0)(t - t_0)^2 + \ldots \quad (2)$$

or, in a more compact format,

$$f(t) = f_n + f_t|_n \Delta t + \frac{1}{2} f_{tt}|_n \Delta t^2 + \ldots + \frac{1}{m!} f_{(m)}|_n \Delta t^m + \ldots \quad (3)$$

where the subscript n denotes the base point location in the discrete temporal grid.

For example, consider the equally spaced discrete grid in Figure 2. Choose time n as the base point, and write the Taylor series for f_{n+1}:

$$f_{n+1} = f_n + f_t|_n \Delta t + \frac{1}{2} f_{tt}|_n \Delta t^2 + \frac{1}{6} f_{ttt}|_n \Delta t^3 + \ldots \quad (4)$$

Solving for $f_t|_n$ gives:

$$f_t|_n = \frac{f_{n+1} - f_n}{\Delta t} - \frac{1}{2} f_{tt}|_n \Delta t - \frac{1}{6} f_{ttt}|_n \Delta t^2 - \ldots \quad (5)$$

which is an exact *finite difference formula* for the first derivative.

The infinite series is not practical, however, so it must be shortened. The terms that are removed from the series are called the *truncation error*, and the terms that remain create a *finite difference approximation* (FDA). We are generally only concerned with the *order* of the truncation error, which is the rate at which the truncation error approaches zero as Δt goes to zero. The order is determined by the first term truncated and is denoted by $O(\Delta t^m)$. Truncating (5) after the first term gives a *first order accurate* FDA for the first derivative at time n:

$$f_t|_n = \frac{f_{n+1} - f_n}{\Delta t} + O(\Delta t) \quad (6)$$

In a similar fashion, FDAs for higher derivatives and/or order can be generated by writing Taylor series for additional points such as f_{n+2}, f_{n+3}, etc.

3. Finite Difference Equations

Several finite difference equations applicable to one-dimensional initial-value problems like the rocket trajectory equations will be presented in this section. Each FDE will be stated without derivation for the general, nonlinear differential equation:

$$y' = f(t, y), \quad y(t_0) = y_0 \quad (7)$$

where $f(t, y)$ is called the *derivative function.*

General features of the FDEs will also be pointed out along with the pros and cons of using the equations for model rocket altitude analysis.

Explicit Euler Method

If the FDA using base point n developed in equation (6) is plugged into equation (7), we get:

$$\frac{y_{n+1} - y_n}{\Delta t} + O(\Delta t) = f(t_n, y_n) = f_n \tag{8}$$

rearranging gives:

$$y_{n+1} = y_n + \Delta t f_n \quad O(\Delta t) \tag{9}$$

This was the method used by G. Harry Stine in the very first RASP program [6] and is still the workhorse of many programs today. Each new solution (y_{n+1}) is calculated from the previous solution (y_n) as the method marches through time. The equation is termed *explicit* because f_n only depends on y_n and not y_{n+1}. The explicit nature of the method allows straightforward calculation of any variable or nonlinear term in the derivative function, since all values are known at time *n*. For the model rocket velocity equation (1b), any dependency on time, velocity, or altitude can be accounted for in the thrust, mass, drag, and air density terms since they are updated algebraically at each time step. (For example, a C_D vs. velocity curve can be readily incorporated in the algorithm for high Mach number flights.) Explicit numerical methods do not require any "constant" values like the Fehskens-Malewicki equations.

Since the FDA was first order accurate, the resulting FDE is also globally first order accurate. For most engineering applications, first order methods perform poorly because the truncation errors are relatively large, and they decrease only linearly with smaller values of Δt. For the model rocket equations however, the explicit Euler method with a sufficiently small time step still gives good results. This will be discussed more in the following sections.

Implicit Euler Method

If the point $n+1$ is used as the base point in the Taylor series and the resulting FDA is plugged into (7), we get:

$$y_{n+1} = y_n + \Delta t f_{n+1} \quad O(\Delta t) \tag{10}$$

This equation is also globally first order, but it is *implicit*, meaning f_{n+1} depends on the yet unknown solution, y_{n+1}. For the rocket velocity equation (1b), the derivative function $g(t, v, x)$ is nonlinear in v due to the aerodynamic drag term, and thus (10) would also become a nonlinear FDE in v_{n+1}. A FDE of this nature cannot simply march from one time to the next. The nonlinear FDE usually must be solved by *iteration* at each time step which adds considerable effort to the computations.

The advantage of implicit methods like the implicit Euler is that they feature *unconditional numerical stability*. Simply put, an unconditionally stable method will produce a bounded solution to a bounded differential equation for all values of Δt. This means that very large time steps can be taken without the solution oscillating wildly or "blowing up." Larger time steps require fewer calculations and can lead to a faster solution. By contrast, explicit methods have a limit on the maximum allowable step size Δt in order to remain stable. Figure 3 illustrates numerical stability.

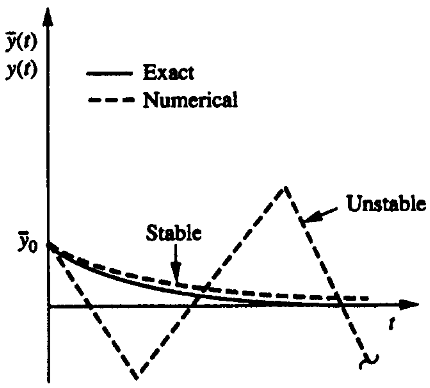

Figure 3. Numerical Stability [5]

One way of determining stability limits is by a *von Neumann stability analysis.* This technique incorporates Fourier series to determine the maximum allowable Δt. For example, the stability limit for the velocity equation using the explicit Euler method works out to be:

$$\Delta t \leq \frac{2m}{A\rho C_D v} \tag{11}$$

Using the parameters of our Alpha III test case during the thrust phase gives:

$$\Delta t \leq \frac{290.912}{v} \tag{12}$$

Since the velocity equation is nonlinear, the stability criterion is not a constant. It varies with velocity, v, as the solution progresses. When v is a maximum (burnout velocity), the limit on Δt is the most stringent. For the Alpha III case, the burnout velocity is estimated to be 123.4830 m/s, giving a time step requirement of:

$$\Delta t \leq 2.356 \text{ s} \tag{13}$$

Despite the appealing large size, using a time step this big is not practical for a model rocket altitude analysis. Considering that the burn times of Estes engines are typically less than 2 seconds, a Δt of 2.356 s would jump right over the thrust phase of the flight in just one time step! This would lead to very poor predictions of burnout velocity and altitude and would compromise the entire solution. Hence, the rocket equations (like most ordinary differential equations) demand small step sizes for the sake of *accuracy*, not *stability*, and therefore the more complicated, nonlinear, implicit equations offer no advantage.

Second Order Adams-Bashforth Method

The family of Adams methods are *multipoint* methods, meaning they use more than one known point to advance the calculations. With more points, more desirable higher order solutions can be achieved. Multipoint methods are typically derived by fitting a polynomial through the selected points and then integrating over them. The *second order* (the truncation errors decrease quadratically with smaller time steps) Adams-Bashforth method is as follows:

$$y_{n+1} = y_n + \frac{1}{2}\Delta t(3f_n - f_{n-1}) \quad O(\Delta t^2) \tag{14}$$

This FDE is explicit, but two previously known points, n and n-1, are needed (and must be stored in computer memory) at each time step. wRASP 2.0 offers this method as a calculation option. Adams-Bashforth methods as accurate as $O(\Delta t^6)$ have been derived.

Fourth Order Runge-Kutta Method

The family of Runge-Kutta methods are *multistep* methods. Intermediate points between n and n+1 are introduced to obtain higher order accuracy. The most popular of these methods is the *fourth order* Runge-Kutta method:

$$y_{n+1} = y_n + \frac{1}{6}(\Delta y_1 + 2\,\Delta y_2 + 2\,\Delta y_3 + \Delta y_4) \quad O(\Delta t^4) \tag{15}$$

where:

$$\Delta y_1 = \Delta t\, f(t_n, y_n) \qquad \Delta y_3 = \Delta t\, f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta y_2}{2})$$

$$\Delta y_2 = \Delta t\, f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta y_1}{2}) \qquad \Delta y_4 = \Delta t\, f(t_n + \Delta t, y_n + \Delta y_3)$$

This FDE is explicit and uses only one known point, but it requires four derivative function evaluations per step. These evaluations are usually the most demanding calculations that the computer must do, and consequently this method requires four times the work as compared to a single step method. However, this is a small price to pay (a programmable hand calculator can readily handle this task) for the high level accuracy. This method is the best choice for most engineering problems. *Rogers Aeroscience* advertises the use of the fourth order Runge-Kutta method in their software.

PART II. NUMERICAL RESULTS AND COMPARISONS FOR THE SAMPLE PROBLEM

This section will apply three of the numerical algorithms mentioned above: the explicit Euler, the second order Adams-Bashforth, and the fourth order Runge-Kutta, to the example Estes problem. The solution will be monitored at each time step and compared to the exact F-M equations. Key parameters such as burnout velocity, time to apogee, and maximum altitude will be highlighted.

Explicit Euler Method

Table 5 gives the velocity and altitude results for the explicit Euler method, the exact solution, and the errors between them for time steps of 0.05 and 0.025 seconds. The first three columns also give time, thrust, and mass, respectively. The numerical solution stops approximately at the rocket's apogee, determined when the velocity first turns negative. (Ideally, the time to apogee should be *interpolated* between the two times that bracket zero velocity. For most model rocket applications however, this level of precision is not needed.) For $\Delta t = 0.05$, the approximate time to apogee is 7.55 s, and the corresponding maximum altitude is 351.0431 m. Compared to the exact solution, the errors are 0.010721 s and -0.988 m, respectively.

Using $\Delta t = 0.025$, the increase in accuracy of the Euler method due to the smaller time step is seen. Since the algorithm is first order (linearly) accurate with respect to time, the 1/2 reduction in Δt should correspond to a 1/2 reduction in the errors. Consider the ratio of the altitude errors at $t = 7.50$ s:

$$\frac{x\,error(\Delta t = 0.025)}{x\,error(\Delta t = 0.05)} = \frac{-0.4769877}{-0.9820129} = 0.4857244 \sim \frac{1}{2}$$

The slight deviation from the theoretical value of 0.5 is due to the finite step size. This calculation is a way of demonstrating the order of a numerical method.

Table 6 shows the results of the explicit Euler method with a time step of 0.001 seconds. This is a common choice for Δt in the commercial software packages.

There has been some criticism of the use of Euler's method in RASP programs. The above results, however, indicate that reasonable results can be obtained with a time step as large as 0.05 seconds. With $\Delta t = 0.001$ s, the time to apogee is accurate to the maximum 3 decimal places of the exact solution, and the numerical error is just -0.0186 m in maximum altitude and 0.0183 m/s in burnout velocity. This level of accuracy is probably acceptable to most rocketeers.

Second Order Adams-Bashforth Method

Table 7 gives the solution for the second order Adams-Bashforth method. At the beginning of the calculations two points are needed, n-1 and n. However, only the one initial condition at $t = 0$ is known. To overcome this situation, the exact solution was used at the first time step to initiate the numerical solution. In practice where no exact values are known, the first calculation is done by another method (like the explicit Euler method) to give the starting points for the Adams-Bashforth routine.

The increased accuracy over the Euler method can be seen. This is due to the higher order behavior of the Adams-Bashforth algorithm. The error at maximum altitude is less than 11 cm with $\Delta t = 0.05$ s. An error ratio at $t = 7.50$ gives:

$$\frac{x\,error(\Delta t = 0.025)}{x\,error(\Delta t = 0.05)} = \frac{0.2805019D-01}{0.1092054D+00} = 0.2568572 \sim \frac{1}{4}$$

The second order accuracy of the method results in a factor of four decrease in the errors for a one half reduction in time step. This behavior is highly desirable. As a rule of thumb, all engineering problems should use a *minimum* second order accurate numerical method.

Table 8 gives the Adams-Bashforth solution for $\Delta t = 0.001$. The last two columns of errors clearly show the marked improvement in accuracy over the Euler method. Predicted values for burnout velocity and maximum altitude are accurate to seven digits. Thus, the numerical solution is *identical* to the exact Fehskens-Malewicki solution on a typical, 16 bit personal computer!

Fourth Order Runge-Kutta Method

Table 9 gives the Runge-Kutta solution. For $\Delta t = 0.05$, the error in burnout velocity is beyond 7 decimal digits, and the maximum altitude error is just -0.0006 m - less than one millimeter! An error ratio at $t = 7.50$ gives:

$$\frac{x\,error(\Delta t = 0.025)}{x\,error(\Delta t = 0.05)} = \frac{-0.3125201D-06}{-0.5033126D-05} = 0.06209264 \sim \frac{1}{16}$$

demonstrating the fourth order of the method. (Recall that $(\frac{1}{2})^4 = \frac{1}{16}$)

Table 10 shows the results for $\Delta t = 0.001$ s. The errors are on order of 10^{-11} m/s in burnout velocity and 10^{-12} m in maximum altitude, significantly less than the other methods and probably well beyond the needs of any model rocketeer. Again, the numerical solution is identical to the exact solution to seven decimal digits.

Figure 4 illustrates graphically the absolute value of the numerical errors of the three methods in logarithmic scale. The superiority of the fourth order Runge-Kutta method is clearly shown.

Some rocketeers have shied away from Runge-Kutta methods, calling them "non-traditional", "exotic", or "complex." These statements are unfounded, considering that this technique is the most popular choice for solving engineering problems involving ordinary differential equations. The formulation is very straightforward, requiring only 10 lines of FORTRAN calculations as shown in the program listing in the Appendix. Though the second order Adams-Bashforth method gives comparable machine-accurate results, the fourth order Runge-Kutta method should still be the first choice for altitude simulation.

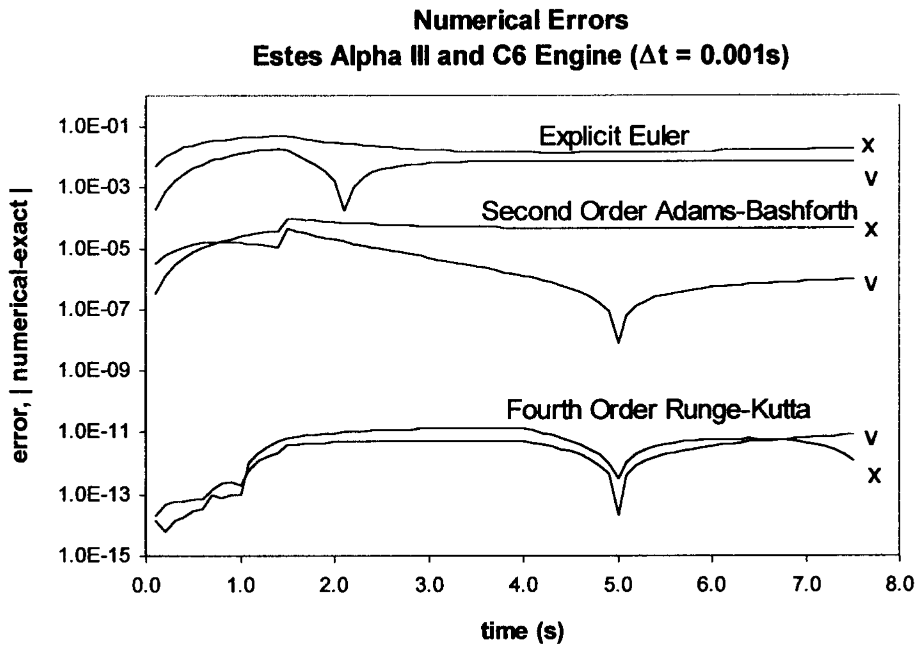

Figure 4.

Additional Research And Observations

Table 3 compares the calculation time required by each method to solve the sample test case. One thousand (1000) loops of each algorithm with $\Delta t = 0.001$ were timed in CPU seconds on a Silicon Graphics UNIX workstation and then averaged. The time required by the Runge-Kutta method reveals the factor of four increase in work relative to the Euler method due to the additional derivative function evaluations. The Adams-Bashforth method requires slightly more time than the Euler method because of the handling of the additional data point at time n-1. All three of these computer times are minuscule and virtually indistinguishable in real time. Computer file handling, input/output, and any graphing performed during the calculations would dominate the overall solution time.

Table 3. Calculation times for the Estes Alpha III and C6 Engine Problem ($\Delta t = 0.001$)

Method	Average Calculation Time (cpu seconds)
Explicit Euler	0.004179
Second order Adams-Bashforth	0.005505
Fourth order Runge-Kutta	0.016087

Since most rocketeers probably do not have access to high-end UNIX workstations, the numerical algorithms were also run on a home computer. Using a 486DX2-66 personal computer, all the methods took only a couple of seconds in real elapsed time, and the velocity and altitude results were identical to the UNIX results to 7 decimal digits. Calculation time is certainly not an issue for any of these methods. In fact, the numerical algorithms can run on a hand-held programmable calculator - ideal for field use.

A key point revealed in this research is that the precision of the numerical method must match the precision of the model rocket data. For example, the thrust duration of the C6 engine is 1.45 seconds, i.e. two decimal points. Therefore, the Δt used in the calculations must also be *at least* two decimal points and *divide evenly* into 1.45 in order to properly capture the thrust phase. For this sample problem, the maximum allowable time step is thus 0.05 s. Time steps that do not meet this criteria will calculate a thrust phase that is either too long or too short, resulting in inaccurate burnout conditions and significant deviations from the F-M equations during the coast phase. Stine's choice of $\Delta t = 0.1$ in *RASP-93* [6], though not ideal for accuracy, is nonetheless consistent with the precision of his thrust data.

CONCLUSION

Numerical methods for solving the differential equation governing model rocket vertical motion have been presented. The first part of this paper introduced the methods with derivations and discussed the benefits, limitations, and applicability to model rocketry for each technique. Key terms such as *truncation error, order, FDA, FDE, explicit, multipoint,* and *multistep* were detailed. The methods were then illustrated with a sample model rocket altitude simulation and the results compared to the exact Fehskens-Malewicki solution.

Each numerical method - explicit Euler, second order Adams-Bashforth, and fourth order Runge-Kutta, produced results that compared favorably with the exact solution, even with time steps as large as 0.05 s. Overall, however, the fourth order Runge-Kutta method with $\Delta t = 0.001$ was demonstrated to be the best numerical solution. The maximum errors were on order of 10^{-11} m/s in velocity and 10^{-11} m in altitude, well beyond the precision of a 16 bit personal computer application and well beyond the needs of even the most demanding model rocketeer. The computer time needed to generate these numerical solutions was negligible as well, requiring only a couple of seconds. Table 4 summarizes the key parameters.

Table 4. Summary of Results (to seven decimal digits)

Estes Alpha III and C6 Engine ($\Delta t = 0.001$ s)

	Exact F-M equations	Explicit Euler	Second Order Adams-Bashforth	Fourth order Runge-Kutta
Burnout Velocity (m/s)	123.4830	123.5013	123.4830	123.4830
Burnout Altitude (m)	99.43599	99.38646	99.43603	99.43599
Coast Time (s)	6.089279	6.089	6.090	6.090
Coast Distance (m)	252.5951	252.6260	252.5951	252.5951
Total Time to Apogee (s)	7.539279	7.539	7.540	7.540
Maximum Altitude (m)	352.0311	352.0125	352.0311	352.0311

In his paper, Viggiano [3] states that numerical solutions will still be "approximate." While this is a true statement, numerical errors in the model rocket equation can be essentially eliminated with the proper choice of method and parameters. On a typical 16 bit personal computer, the numerical solution can in fact be *identical* to the exact solution with minimal effort. In this research paper, the numerical techniques themselves have been shown to be highly accurate. Significant errors could be induced only when the algorithms are *misapplied*, such as with an inappropriate time step or inconsistent model rocket data.

The results presented here indicate that relative to the F-M equations, numerical methods offer an equally effective means of predicting model rocket altitude. It is the author's opinion that numerical methods, namely the fourth order Runge-Kutta, are in fact *superior* to the F-M equations because there are no limitations on the actual *physics* of model rocket flight. Numerical methods do not require any constant values like the F-M equations.

Larry Curcio's *DigiTrak* [7] features additional results such as optimal rocket mass and backtracked drag coefficients. Extracting this data is simply done with the analytical F-M equations. This information should also be possible to obtain from numerical techniques and can be the subject of a future study in model rocket simulation.

REFERENCES

1. *Model Rocket Altitude Prediction Charts*, TR-10, Estes Industries, 1971.
2. Kuechler, Thomas and Kelley, Clifford, "A New Approach to Coast Phase Altitude Calculation", *NAR Technical Review*, Volume 4, 1979.
3. Viggiano, John, "A New Technique for Integrating the Motion Equation for Rocket Altitude Simulation", *NAR Technical Review-Volume 8*, MARSCON 1993.
4. Personal communication with Larry Curcio, October, 1997.
5. Hoffman, Joe D., *Numerical Methods for Engineers and Scientists*, McGraw-Hill, 1992.
6. Stine, G. Harry, *Handbook of Model Rocketry*, Wiley and Sons, 1994.
7. Curcio, Larry, *wRASP 2.0* Help File.

EXPLICIT EULER METHOD
DT= 0.0500 PRINT FREQUENCY= 0.0500

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0500	5.8750	0.0508	0.5292148D+01	0.0000000D+00	0.5290544D+01	0.1322836D+00	0.1603956D-02	-0.1322836D+00
0.1000	5.8750	0.0508	0.1057948D+02	0.2646074D+00	0.1057148D+02	0.5288942D+00	0.8004046D-02	-0.2642868D+00
0.1500	5.8750	0.0508	0.1585239D+02	0.7935815D+00	0.1583326D+02	0.1189112D+01	0.1913060D-01	-0.3955307D+00
0.2000	5.8750	0.0508	0.2110135D+02	0.1586201D+01	0.2106649D+02	0.2111744D+01	0.3485425D-01	-0.5255433D+00
0.2500	5.8750	0.0508	0.2631697D+02	0.2641269D+01	0.2626198D+02	0.3295132D+01	0.5498840D-01	-0.6538638D+00
0.3000	5.8750	0.0508	0.3149008D+02	0.3957117D+01	0.3141079D+02	0.4737164D+01	0.7929277D-01	-0.7800472D+00
.								
.								
.								
1.4000	5.8750	0.0508	0.1216557D+03	0.9084227D+02	0.1207545D+03	0.9332958D+02	0.9012290D+00	-0.2487309D+01
1.4500	5.8750	0.0508	0.1244041D+03	0.9692506D+02	0.1234830D+03	0.9943599D+02	0.9211361D+00	-0.2510937D+01
1.5000	0.0000	0.0454	0.1209375D+03	0.1031453D+03	0.1201395D+03	0.1055259D+03	0.7979278D+00	-0.2380640D+01
1.5500	0.0000	0.0454	0.1176343D+03	0.1091921D+03	0.1169471D+03	0.1114525D+03	0.6872727D+00	-0.2260326D+01
.								
.								
.								
7.3000	0.0000	0.0454	0.1995522D+01	0.3507897D+03	0.2348220D+01	0.3517502D+03	-0.3526981D+00	-0.9605753D+00
7.3500	0.0000	0.0454	0.1504424D+01	0.3508894D+03	0.1857034D+01	0.3518554D+03	-0.3526099D+00	-0.9659289D+00
7.4000	0.0000	0.0454	0.1013656D+01	0.3509647D+03	0.1366198D+01	0.3519360D+03	-0.3525418D+00	-0.9712872D+00
7.4500	0.0000	0.0454	0.5231261D+00	0.3510154D+03	0.8756200D+00	0.3519920D+03	-0.3524939D+00	-0.9766490D+00
7.5000	0.0000	0.0454	0.3274093D-01	0.3510415D+03	0.3852072D+00	0.3520235D+03	-0.3524662D+00	-0.9820129D+00
7.5500	0.0000	0.0454	-0.4575918D+00	0.3510431D+03				

DT= 0.0250 PRINT FREQUENCY= 0.0250

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0250	5.8750	0.0508	0.2646074D+01	0.0000000D+00	0.2645873D+01	0.3307467D-01	0.2005492D-03	-0.3307467D-01
0.0500	5.8750	0.0508	0.5291546D+01	0.6615185D-01	0.5290544D+01	0.1322836D+00	0.1002254D-02	-0.6613180D-01
0.0750	5.8750	0.0508	0.7935214D+01	0.1984405D+00	0.7932811D+01	0.2975818D+00	0.2402928D-02	-0.9914133D-01
0.1000	5.8750	0.0508	0.1057588D+02	0.3968208D+00	0.1057148D+02	0.5288942D+00	0.4398482D-02	-0.1320733D+00
0.1250	5.8750	0.0508	0.1321234D+02	0.6612178D+00	0.1320536D+02	0.8261158D+00	0.6982943D-02	-0.1648981D+00
0.1500	5.8750	0.0508	0.1584341D+02	0.9915262D+00	0.1583326D+02	0.1189112D+01	0.1014849D-01	-0.1975860D+00
.								
.								
.								
1.4250	5.8750	0.0508	0.1225869D+03	0.9512457D+02	0.1221331D+03	0.9636573D+02	0.4538184D+00	-0.1241163D+01
1.4500	5.8750	0.0508	0.1239416D+03	0.9818924D+02	0.1234830D+03	0.9943599D+02	0.4585452D+00	-0.1246751D+01
1.4750	0.0000	0.0454	0.1222193D+03	0.1012878D+03	0.1217917D+03	0.1025018D+03	0.4275663D+00	-0.1214063D+01
1.5000	0.0000	0.0454	0.1205377D+03	0.1043433D+03	0.1201395D+03	0.1055259D+03	0.3981982D+00	-0.1182641D+01
.								
.								
.								
7.4000	0.0000	0.0454	0.1191665D+01	0.3514642D+03	0.1366198D+01	0.3519360D+03	-0.1745325D+00	-0.4717988D+00
7.4250	0.0000	0.0454	0.9463626D+00	0.3514940D+03	0.1120883D+01	0.3519670D+03	-0.1745199D+00	-0.4730956D+00
7.4500	0.0000	0.0454	0.7011102D+00	0.3515176D+03	0.8756200D+00	0.3519920D+03	-0.1745097D+00	-0.4743927D+00
7.4750	0.0000	0.0454	0.4558967D+00	0.3515351D+03	0.6303987D+00	0.3520108D+03	-0.1745020D+00	-0.4756901D+00
7.5000	0.0000	0.0454	0.2107105D+00	0.3515465D+03	0.3852072D+00	0.3520235D+03	-0.1744967D+00	-0.4769877D+00
7.5250	0.0000	0.0454	-0.3446003D-01	0.3515518D+03				

Table 5. Explicit Euler Method

EXPLICIT EULER METHOD
DT= 0.0010 PRINT FREQUENCY= 0.0010

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0010	5.8750	0.0508	0.1058430D+00	0.0000000D+00	0.1058429D+00	0.5292147D-04	0.1283632D-07	-0.5292147D-04
0.0020	5.8750	0.0508	0.2116859D+00	0.1058430D-03	0.2116858D+00	0.2116859D-03	0.6418154D-07	-0.1058429D-03
0.0030	5.8750	0.0508	0.3175287D+00	0.3175288D-03	0.3175285D+00	0.4762930D-03	0.1540354D-06	-0.1587642D-03
0.0040	5.8750	0.0508	0.4233713D+00	0.6350575D-03	0.4233710D+00	0.8467428D-03	0.2823976D-06	-0.2116853D-03
0.0050	5.8750	0.0508	0.5292136D+00	0.1058429D-02	0.5292132D+00	0.1323035D-02	0.4492674D-06	-0.2646062D-03
0.0060	5.8750	0.0508	0.6350556D+00	0.1587642D-02	0.6350550D+00	0.1905169D-02	0.6546440D-06	-0.3175266D-03
0.0070	5.8750	0.0508	0.7408972D+00	0.2222698D-02	0.7408963D+00	0.2593145D-02	0.8985264D-06	-0.3704467D-03
0.0080	5.8750	0.0508	0.8467383D+00	0.2963595D-02	0.8467371D+00	0.3386961D-02	0.1180913D-05	-0.4233662D-03
0.0090	5.8750	0.0508	0.9525788D+00	0.3810333D-02	0.9525772D+00	0.4286619D-02	0.1501804D-05	-0.4762852D-03
0.0100	5.8750	0.0508	0.1058419D+01	0.4762912D-02	0.1058417D+01	0.5292116D-02	0.1861196D-05	-0.5292035D-03
.								
.								
.								
1.4490	5.8750	0.0508	0.1234478D+03	0.9926301D+02	0.1234296D+03	0.9931254D+02	0.1825818D-01	-0.4952873D-01
1.4500	5.8750	0.0508	0.1235013D+03	0.9938646D+02	0.1234830D+03	0.9943599D+02	0.1826537D-01	-0.4953720D-01
1.4510	0.0000	0.0454	0.1234328D+03	0.9950996D+02	0.1234146D+03	0.9955944D+02	0.1821552D-01	-0.4948472D-01
1.4520	0.0000	0.0454	0.1233644D+03	0.9963339D+02	0.1233462D+03	0.9968282D+02	0.1816578D-01	-0.4943232D-01
.								
.								
.								
7.5300	0.0000	0.0454	0.8408599D-01	0.3520121D+03	0.9100031D-01	0.3520307D+03	-0.6914313D-02	-0.1861063D-01
7.5310	0.0000	0.0454	0.7427932D-01	0.3520121D+03	0.8119363D-01	0.3520308D+03	-0.6914312D-02	-0.1861264D-01
7.5320	0.0000	0.0454	0.6447265D-01	0.3520122D+03	0.7138696D-01	0.3520308D+03	-0.6914310D-02	-0.1861465D-01
7.5330	0.0000	0.0454	0.5466598D-01	0.3520123D+03	0.6158029D-01	0.3520309D+03	-0.6914309D-02	-0.1861666D-01
7.5340	0.0000	0.0454	0.4485932D-01	0.3520123D+03	0.5177363D-01	0.3520310D+03	-0.6914308D-02	-0.1861867D-01
7.5350	0.0000	0.0454	0.3505266D-01	0.3520124D+03	0.4196697D-01	0.3520310D+03	-0.6914308D-02	-0.1862069D-01
7.5360	0.0000	0.0454	0.2524601D-01	0.3520124D+03	0.3216031D-01	0.3520310D+03	-0.6914307D-02	-0.1862270D-01
7.5370	0.0000	0.0454	0.1543935D-01	0.3520124D+03	0.2235366D-01	0.3520311D+03	-0.6914307D-02	-0.1862471D-01
7.5380	0.0000	0.0454	0.5632702D-02	0.3520125D+03	0.1254701D-01	0.3520311D+03	-0.6914306D-02	-0.1862672D-01
7.5390	0.0000	0.0454	-0.4173948D-02	0.3520125D+03				

Table 6. Explicit Euler Method

2ND ORDER ADAMS BASHFORTH METHOD
DT= 0.0500 PRINT FREQUENCY= 0.0500

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0500	5.8750	0.0508	exact solution	exact solution	0.5290544D+01	0.1322836D+00		
0.1000	5.8750	0.0508	0.1057548D+02	0.5290744D+00	0.1057148D+02	0.5288942D+00	0.3997656D-02	0.1802484D-03
0.1500	5.8750	0.0508	0.1584120D+02	0.1189972D+01	0.1583326D+02	0.1189112D+01	0.7932925D-02	0.8593085D-03
0.2000	5.8750	0.0508	0.2107826D+02	0.2113674D+01	0.2106649D+02	0.2111744D+01	0.1176356D-01	0.1929819D-02
0.2500	5.8750	0.0508	0.2627743D+02	0.3298514D+01	0.2626198D+02	0.3295132D+01	0.1544932D-01	0.3381360D-02
0.3000	5.8750	0.0508	0.3142974D+02	0.4742364D+01	0.3141079D+02	0.4737164D+01	0.1895265D-01	0.5200220D-02
.								
.								
.								
1.4000	5.8750	0.0508	0.1207806D+03	0.9341925D+02	0.1207545D+03	0.9332958D+02	0.2612761D-01	0.8966952D-01
1.4500	5.8750	0.0508	0.1235072D+03	0.9952934D+02	0.1234830D+03	0.9943599D+02	0.2419103D-01	0.9334214D-01
1.5000	0.0000	0.0454	0.1200192D+03	0.1057729D+03	0.1201395D+03	0.1055259D+03	-0.1203009D+00	0.2469555D+00
1.5500	0.0000	0.0454	0.1168403D+03	0.1116866D+03	0.1169471D+03	0.1114525D+03	-0.1067274D+00	0.2341592D+00
.								
.								
.								
7.3000	0.0000	0.0454	0.2350214D+01	0.3518591D+03	0.2348220D+01	0.3517502D+03	0.1993762D-02	0.1088179D+00
7.3500	0.0000	0.0454	0.1859065D+01	0.3519643D+03	0.1857034D+01	0.3518554D+03	0.2031010D-02	0.1089091D+00
7.4000	0.0000	0.0454	0.1368266D+01	0.3520450D+03	0.1366198D+01	0.3519360D+03	0.2068487D-02	0.1090040D+00
7.4500	0.0000	0.0454	0.8777262D+00	0.3521011D+03	0.8756200D+00	0.3519920D+03	0.2106235D-02	0.1091028D+00
7.5000	0.0000	0.0454	0.3873515D+00	0.3521327D+03	0.3852072D+00	0.3520235D+03	0.2144299D-02	0.1092054D+00
7.5500	0.0000	0.0454	-0.1029502D+00	0.3521398D+03				

DT= 0.0250 PRINT FREQUENCY= 0.0250

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0250	5.8750	0.0508	exact solution	exact solution	0.2645873D+01	0.3307467D-01		
0.0500	5.8750	0.0508	0.5291045D+01	0.1322949D+00	0.5290544D+01	0.1322836D+00	0.5009902D-03	0.1127782D-04
0.0750	5.8750	0.0508	0.7933811D+01	0.2976357D+00	0.7932811D+01	0.2975818D+00	0.1000022D-02	0.5385348D-04
0.1000	5.8750	0.0508	0.1057297D+02	0.5290155D+00	0.1057148D+02	0.5288942D+00	0.1495740D-02	0.1213480D-03
0.1250	5.8750	0.0508	0.1320734D+02	0.8263294D+00	0.1320536D+02	0.8261158D+00	0.1986803D-02	0.2135957D-03
0.1500	5.8750	0.0508	0.1583573D+02	0.1189443D+01	0.1583326D+02	0.1189112D+01	0.2471893D-02	0.3303757D-03
.								
.								
.								
1.4250	5.8750	0.0508	0.1221394D+03	0.9638916D+02	0.1221331D+03	0.9636573D+02	0.6310286D-02	0.2342298D-01
1.4500	5.8750	0.0508	0.1234891D+03	0.9945987D+02	0.1234830D+03	0.9943599D+02	0.6063046D-02	0.2387626D-01
1.4750	0.0000	0.0454	0.1217616D+03	0.1025640D+03	0.1217917D+03	0.1025018D+03	-0.3008940D-01	0.6212349D-01
1.5000	0.0000	0.0454	0.1201112D+03	0.1055864D+03	0.1201395D+03	0.1055259D+03	-0.2835129D-01	0.6050997D-01
.								
.								
.								
7.4250	0.0000	0.0454	0.1121442D+01	0.3519951D+03	0.1120883D+01	0.3519670D+03	0.5595186D-03	0.2800902D-01
7.4500	0.0000	0.0454	0.8761842D+00	0.3520200D+03	0.8756200D+00	0.3519920D+03	0.5642369D-03	0.2802251D-01
7.4750	0.0000	0.0454	0.6309677D+00	0.3520389D+03	0.6303987D+00	0.3520108D+03	0.5689773D-03	0.2803623D-01
7.5000	0.0000	0.0454	0.3857809D+00	0.3520516D+03	0.3852072D+00	0.3520235D+03	0.5737411D-03	0.2805019D-01
7.5250	0.0000	0.0454	0.1406123D+00	0.3520582D+03	0.1400338D+00	0.3520301D+03	0.5785295D-03	0.2806439D-01
7.5500	0.0000	0.0454	-0.1045496D+00	0.3520586D+03				

Table 7. Second Order Adams-Bashforth Method

2ND ORDER ADAMS BASHFORTH METHOD
DT= 0.0010 PRINT FREQUENCY= 0.0010

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0010	5.8750	0.0508	exact solution	exact solution	0.1058429D+00	0.5292147D-04		
0.0020	5.8750	0.0508	0.2116858D+00	0.2116859D-03	0.2116858D+00	0.2116859D-03	0.3209075D-07	0.2888094D-10
0.0030	5.8750	0.0508	0.3175286D+00	0.4762932D-03	0.3175285D+00	0.4762930D-03	0.6418131D-07	0.1380049D-09
0.0040	5.8750	0.0508	0.4233711D+00	0.8467431D-03	0.4233710D+00	0.8467428D-03	0.9627152D-07	0.3112866D-09
0.0050	5.8750	0.0508	0.5292133D+00	0.1323035D-02	0.5292132D+00	0.1323035D-02	0.1283612D-06	0.5487605D-09
0.0060	5.8750	0.0508	0.6350551D+00	0.1905170D-02	0.6350550D+00	0.1905169D-02	0.1604504D-06	0.8504065D-09
0.0070	5.8750	0.0508	0.7408965D+00	0.2593146D-02	0.7408963D+00	0.2593145D-02	0.1925387D-06	0.1216214D-08
0.0080	5.8750	0.0508	0.8467373D+00	0.3386963D-02	0.8467371D+00	0.3386961D-02	0.2246262D-06	0.1646248D-08
0.0090	5.8750	0.0508	0.9525775D+00	0.4286621D-02	0.9525772D+00	0.4286619D-02	0.2567126D-06	0.2140390D-08
0.0100	5.8750	0.0508	0.1058417D+01	0.5292118D-02	0.1058417D+01	0.5292116D-02	0.2887978D-06	0.2698787D-08
.								
.								
.								
1.4490	5.8750	0.0508	0.1234296D+03	0.9931258D+02	0.1234296D+03	0.9931254D+02	0.9734743D-05	0.3901665D-04
1.4500	5.8750	0.0508	0.1234830D+03	0.9943603D+02	0.1234830D+03	0.9943599D+02	0.9718636D-05	0.3904529D-04
1.4510	0.0000	0.0454	0.1234145D+03	0.9955954D+02	0.1234146D+03	0.9955944D+02	-0.4816903D-04	0.9999748D-04
1.4520	0.0000	0.0454	0.1233462D+03	0.9968292D+02	0.1233462D+03	0.9968282D+02	-0.4805513D-04	0.9989330D-04
.								
.								
.								
7.5300	0.0000	0.0454	0.9100129D-01	0.3520307D+03	0.9100031D-01	0.3520307D+03	0.9843700D-06	0.4606806D-04
7.5310	0.0000	0.0454	0.8119461D-01	0.3520308D+03	0.8119363D-01	0.3520308D+03	0.9846776D-06	0.4606905D-04
7.5320	0.0000	0.0454	0.7138794D-01	0.3520309D+03	0.7138696D-01	0.3520308D+03	0.9849853D-06	0.4607003D-04
7.5330	0.0000	0.0454	0.6158127D-01	0.3520309D+03	0.6158029D-01	0.3520309D+03	0.9852930D-06	0.4607101D-04
7.5340	0.0000	0.0454	0.5177461D-01	0.3520310D+03	0.5177363D-01	0.3520310D+03	0.9856009D-06	0.4607199D-04
7.5350	0.0000	0.0454	0.4196795D-01	0.3520310D+03	0.4196697D-01	0.3520310D+03	0.9859088D-06	0.4607298D-04
7.5360	0.0000	0.0454	0.3216130D-01	0.3520311D+03	0.3216031D-01	0.3520310D+03	0.9862167D-06	0.4607396D-04
7.5370	0.0000	0.0454	0.2235465D-01	0.3520311D+03	0.2235366D-01	0.3520311D+03	0.9865248D-06	0.4607495D-04
7.5380	0.0000	0.0454	0.1254800D-01	0.3520311D+03	0.1254701D-01	0.3520311D+03	0.9868329D-06	0.4607593D-04
7.5390	0.0000	0.0454	0.2741346D-02	0.3520311D+03	0.2740359D-02	0.3520311D+03	0.9871411D-06	0.4607692D-04
7.5400	0.0000	0.0454	-0.7065304D-02	0.3520311D+03				

Table 8. Second Order Adams-Bashforth Method

4TH ORDER RUNGE-KUTTA METHOD
DT= 0.0500 PRINT FREQUENCY= 0.0500

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0500	5.8750	0.0508	0.5290544D+01	0.1322836D+00	0.5290544D+01	0.1322836D+00	-0.3649960D-07	-0.3032020D-09
0.1000	5.8750	0.0508	0.1057148D+02	0.5288942D+00	0.1057148D+02	0.5288942D+00	-0.7335368D-07	-0.4213041D-08
0.1500	5.8750	0.0508	0.1583326D+02	0.1189112D+01	0.1583326D+02	0.1189112D+01	-0.1110229D-06	-0.1160894D-07
0.2000	5.8750	0.0508	0.2106649D+02	0.2111744D+01	0.2106649D+02	0.2111744D+01	-0.1499529D-06	-0.2229653D-07
0.2500	5.8750	0.0508	0.2626198D+02	0.3295132D+01	0.2626198D+02	0.3295132D+01	-0.1905637D-06	-0.3601420D-07
0.3000	5.8750	0.0508	0.3141079D+02	0.4737164D+01	0.3141079D+02	0.4737164D+01	-0.2332395D-06	-0.5244252D-07
.								
.								
.								
1.4000	5.8750	0.0508	0.1207545D+03	0.9332958D+02	0.1207545D+03	0.9332958D+02	-0.1712936D-05	-0.4528149D-06
1.4500	5.8750	0.0508	0.1234830D+03	0.9943599D+02	0.1234830D+03	0.9943599D+02	-0.1776166D-05	-0.4635392D-06
1.5000	0.0000	0.0454	0.1201395D+03	0.1055259D+03	0.1201395D+03	0.1055259D+03	-0.1668844D-05	-0.7191408D-06
1.5500	0.0000	0.0454	0.1169471D+03	0.1114525D+03	0.1169471D+03	0.1114525D+03	-0.1572833D-05	-0.9493550D-06
.								
.								
.								
7.3000	0.0000	0.0454	0.2348220D+01	0.3517502D+03	0.2348220D+01	0.3517502D+03	-0.2328214D-06	-0.4986610D-05
7.3500	0.0000	0.0454	0.1857034D+01	0.3518554D+03	0.1857034D+01	0.3518554D+03	-0.2325975D-06	-0.4998255D-05
7.4000	0.0000	0.0454	0.1366198D+01	0.3519360D+03	0.1366198D+01	0.3519360D+03	-0.2324174D-06	-0.5009888D-05
7.4500	0.0000	0.0454	0.8756197D+00	0.3519920D+03	0.8756200D+00	0.3519920D+03	-0.2322811D-06	-0.5021511D-05
7.5000	0.0000	0.0454	0.3852069D+00	0.3520235D+03	0.3852072D+00	0.3520235D+03	-0.2321885D-06	-0.5033126D-05
7.5500	0.0000	0.0454	-0.1051325D+00	0.3520305D+03				

DT= 0.0250 PRINT FREQUENCY= 0.0250

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0250	5.8750	0.0508	0.2645873D+01	0.3307467D-01	0.2645873D+01	0.3307467D-01	-0.1140302D-08	-0.4779649D-11
0.0500	5.8750	0.0508	0.5290544D+01	0.1322836D+00	0.5290544D+01	0.1322836D+00	-0.2283370D-08	-0.6631398D-10
0.0750	5.8750	0.0508	0.7932811D+01	0.2975818D+00	0.7932811D+01	0.2975818D+00	-0.3432829D-08	-0.1842746D-09
0.1000	5.8750	0.0508	0.1057148D+02	0.5288942D+00	0.1057148D+02	0.5288942D+00	-0.4592268D-08	-0.3578378D-09
0.1250	5.8750	0.0508	0.1320536D+02	0.8261158D+00	0.1320536D+02	0.8261158D+00	-0.5765228D-08	-0.5859003D-09
0.1500	5.8750	0.0508	0.1583326D+02	0.1189112D+01	0.1583326D+02	0.1189112D+01	-0.6955190D-08	-0.8670871D-09
.								
.								
.								
1.4250	5.8750	0.0508	0.1221331D+03	0.9636573D+02	0.1221331D+03	0.9636573D+02	-0.1081372D-06	-0.3030306D-07
1.4500	5.8750	0.0508	0.1234830D+03	0.9943599D+02	0.1234830D+03	0.9943599D+02	-0.1100335D-06	-0.3067365D-07
1.4750	0.0000	0.0454	0.1217917D+03	0.1025018D+03	0.1217917D+03	0.1025018D+03	-0.1065883D-06	-0.3876482D-07
1.5000	0.0000	0.0454	0.1201395D+03	0.1055259D+03	0.1201395D+03	0.1055259D+03	-0.1033344D-06	-0.4643715D-07
.								
.								
.								
7.4250	0.0000	0.0454	0.1120883D+01	0.3519670D+03	0.1120883D+01	0.3519670D+03	-0.1430232D-07	-0.3114475D-06
7.4500	0.0000	0.0454	0.8756199D+00	0.3519920D+03	0.8756200D+00	0.3519920D+03	-0.1429844D-07	-0.3118051D-06
7.4750	0.0000	0.0454	0.6303987D+00	0.3520108D+03	0.6303987D+00	0.3520108D+03	-0.1429523D-07	-0.3121626D-06
7.5000	0.0000	0.0454	0.3852072D+00	0.3520235D+03	0.3852072D+00	0.3520235D+03	-0.1429270D-07	-0.3125201D-06
7.5250	0.0000	0.0454	0.1400338D+00	0.3520301D+03	0.1400338D+00	0.3520301D+03	-0.1429083D-07	-0.3128774D-06
7.5500	0.0000	0.0454	-0.1051326D+00	0.3520305D+03				

Table 9. Fourth Order Runge-Kutta Method

4TH ORDER RUNGE-KUTTA METHOD
DT= 0.0010 PRINT FREQUENCY= 0.0010

TIME (s)	T (N)	M (kg)	V NUM (m/s)	X NUM (m)	V EXACT (m/s)	X EXACT (m)	V ERROR (m/s)	X ERROR (m)
0	0	0.0508	0	0	0	0		
0.0010	5.8750	0.0508	0.1058429D+00	0.5292147D-04	0.1058429D+00	0.5292147D-04	-0.1110223D-15	-0.1924899D-14
0.0020	5.8750	0.0508	0.2116858D+00	0.2116859D-03	0.2116858D+00	0.2116859D-03	-0.1942890D-15	-0.2684864D-14
0.0030	5.8750	0.0508	0.3175285D+00	0.4762930D-03	0.3175285D+00	0.4762930D-03	-0.4440892D-15	0.1277190D-13
0.0040	5.8750	0.0508	0.4233710D+00	0.8467428D-03	0.4233710D+00	0.8467428D-03	-0.4996004D-15	0.4959683D-14
0.0050	5.8750	0.0508	0.5292132D+00	0.1323035D-02	0.5292132D+00	0.1323035D-02	-0.6661338D-15	0.9080627D-14
0.0060	5.8750	0.0508	0.6350550D+00	0.1905169D-02	0.6350550D+00	0.1905169D-02	-0.8881784D-15	0.5884399D-14
0.0070	5.8750	0.0508	0.7408963D+00	0.2593145D-02	0.7408963D+00	0.2593145D-02	-0.9992007D-15	-0.1367526D-13
0.0080	5.8750	0.0508	0.8467371D+00	0.3386961D-02	0.8467371D+00	0.3386961D-02	-0.9992007D-15	0.1623311D-13
0.0090	5.8750	0.0508	0.9525772D+00	0.4286619D-02	0.9525772D+00	0.4286619D-02	-0.1221245D-14	-0.2197114D-13
0.0100	5.8750	0.0508	0.1058417D+01	0.5292116D-02	0.1058417D+01	0.5292116D-02	-0.1554312D-14	0.2298769D-13
.								
.								
.								
1.4490	5.8750	0.0508	0.1234296D+03	0.9931254D+02	0.1234296D+03	0.9931254D+02	0.2287948D-11	0.5883294D-11
1.4500	5.8750	0.0508	0.1234830D+03	0.9943599D+02	0.1234830D+03	0.9943599D+02	0.2302158D-11	0.5968559D-11
1.4510	0.0000	0.0454	0.1234146D+03	0.9955944D+02	0.1234146D+03	0.9955944D+02	-0.3637979D-11	0.6082246D-11
1.4520	0.0000	0.0454	0.1233462D+03	0.9968282D+02	0.1233462D+03	0.9968282D+02	-0.3694822D-11	0.6124878D-11
.								
.								
.								
7.5300	0.0000	0.0454	0.9100031D-01	0.3520307D+03	0.9100031D-01	0.3520307D+03	0.8319651D-11	-0.8526513D-12
7.5310	0.0000	0.0454	0.8119363D-01	0.3520308D+03	0.8119363D-01	0.3520308D+03	0.8322912D-11	-0.7958079D-12
7.5320	0.0000	0.0454	0.7138696D-01	0.3520308D+03	0.7138696D-01	0.3520308D+03	0.8326173D-11	-0.7958079D-12
7.5330	0.0000	0.0454	0.6158029D-01	0.3520309D+03	0.6158029D-01	0.3520309D+03	0.8329448D-11	-0.7958079D-12
7.5340	0.0000	0.0454	0.5177363D-01	0.3520310D+03	0.5177363D-01	0.3520310D+03	0.8332723D-11	-0.7958079D-12
7.5350	0.0000	0.0454	0.4196697D-01	0.3520310D+03	0.4196697D-01	0.3520310D+03	0.8336005D-11	-0.7958079D-12
7.5360	0.0000	0.0454	0.3216031D-01	0.3520310D+03	0.3216031D-01	0.3520310D+03	0.8339274D-11	-0.7389644D-12
7.5370	0.0000	0.0454	0.2235366D-01	0.3520311D+03	0.2235366D-01	0.3520311D+03	0.8342545D-11	-0.7389644D-12
7.5380	0.0000	0.0454	0.1254701D-01	0.3520311D+03	0.1254701D-01	0.3520311D+03	0.8345819D-11	-0.7389644D-12
7.5390	0.0000	0.0454	0.2740359D-02	0.3520311D+03	0.2740359D-02	0.3520311D+03	0.8349094D-11	-0.7389644D-12
7.5400	0.0000	0.0454	-0.7066291D-02	0.3520311D+03				

Table 10. Fourth Order Runge-Kutta Method

APPENDIX

SAMPLE FORTRAN SOURCE CODE

```
C-----------------------------------------------------------------------
C  MAIN PROGRAM TO SOLVE THE MODEL ROCKET TRAJECTORY EQUATIONS BY
C  NUMERICAL METHODS AND COMPARISON TO EXACT "FM" EQUATIONS
C
C  DEVELOPED FOR NARAM40 R&D COMPETITION
C
C  4TH ORDER RUNGE-KUTTA METHOD
C
C  KENNETH J. KARBON
C  NAR # 72175
C
C  1998
C-----------------------------------------------------------------------
C
C
      DOUBLE PRECISION GRAV,C,DT,XN,VN,XNP1,VNP1,M,TN,PFREQ
     &,VERROR,XERROR,TC,SC,W,VES,XES,RHO,PI,MROC,D,CD
     &,MENGI,MPROP,TDUR,T,VBO,SBO,MC
      CHARACTER*10 OUTFILE
      COMMON GRAV,C,M,T
C
C--OPEN OUTPUT FILES----------------------------------------------------
      OUTFILE='RK.OUT'
      OPEN(UNIT=9,FILE=OUTFILE,STATUS='UNKNOWN')
C
C--PHYSICAL AND ENVIRONMENTAL CONSTANTS---------------------------------
      RHO=1.205D+00
      PI=4.D+00*DATAN(1.D+00)
      GRAV=9.80665D+00
C
C--ESTES ALPHA III MODEL ROCKET (FROM 1997 CATELOG)---------------------
      MROC=.034D+00
      D=.0248D+00
      CD=.6D+00
      C=.5D+00*RHO*CD*PI/4.D+00*D**2
C
C--ESTES C6 ENGINE (FROM WRASP DATABASE)--------------------------------
      MENGI=.0222D+00
      MPROP=.0108D+00
      TDUR=1.45D+00
      T=5.875D+00
C
      M=MROC+MENGI-.5D+00*MPROP
C
C--TIME STEP CONTROLS---------------------------------------------------
      WRITE(*,53)
53    FORMAT('0',' ENTER DT AND PRINT FREQ')
      READ(*,*) DT,PFREQ
```

```
      MAXSTE=500000
      IFREQ=PFREQ/DT
C
C--INITIAL CONDITIONS---------------------------------------------
      TN=0.D+00
      VN=0.D+00
      XN=0.D+00
C
C--BURNOUT VELOCITY AND DISTANCE EXACT SOLUTION (VIGGIANO)---------
      VBO=DSQRT((T-M*GRAV)/C)*DTANH(TDUR/M*DSQRT(C*(T-M*GRAV)))
      SBO=M/C*DLOG(DCOSH(TDUR/M*DSQRT(C*(T-M*GRAV))))
C
C--COAST DISTANCE AND COAST TIME EXACT SOLUTION (VIGGIANO)---------
      MC=MROC+MENGI-MPROP
      TC=DSQRT(MC/(GRAV*C))*DATAN(VBO*DSQRT(C/(MC*GRAV)))
      SC=MC/(2.D+00*C)*DLOG(1.D+00+(C*VBO**2)/(MC*GRAV))
C
C--WRITE HEADER INFO----------------------------------------------
      WRITE(9,*) '4TH ORDER RUNGE-KUTTA METHOD'
      WRITE(9,*) 'ESTES ALPHA III AND C6 ENGINE'
      WRITE(9,*) 'ALL UNITS ARE SI (M, KG, S)'
      WRITE(9,23) DT,PFREQ
23    FORMAT('DT=',F8.4,1X,'PRINT FREQUENCY=',F8.4)
      WRITE(9,*) 'EXACT BURNOUT VELOCITY AND DISTANCE'
      WRITE(9,63) VBO,SBO
63    FORMAT('VBO=',D14.7,1X,'SBO=',D14.7)
      WRITE(9,*) 'EXACT COAST TIME AND DISTANCE'
      WRITE(9,33) TC,SC
33    FORMAT('TC=',D14.7,1X,'SC=',D14.7)
      WRITE(9,73) TDUR+TC,SBO+SC
73    FORMAT('EXACT TIME TO APOGEE=',D14.7,1X,'EXACT ALTITUDE=',D14.7
      WRITE(9,43)
43    FORMAT(1X,'TIME (s)',2X,'T (N)',4X,'M (kg)',2X,'V NUM (m/s)',7X
     &'X NUM (m)',3X,'V EXACT (m/s)',3X,'X EXACT (m)',3X,
     &'V ERROR (m/s)',3X,'X ERROR (m)')
C
C--BEGIN TIME MARCHING--------------------------------------------
C
      DO 100 I=1,MAXSTE
         TN=TN+DT
         IF ((TN-TDUR).GT.1.D-06) THEN
          T=0.D+00
          M=MROC+MENGI-MPROP
C--EXACT SOLUTION COAST VELOCITY(TN) AND DIST(TN) (ESTES TR-10)----
          W=M*GRAV
          VES=DSQRT(W/C)*DTAN(GRAV*DSQRT(C/W)*(TC-(TN-TDUR)))
          XES=SBO+SC+M/C*DLOG(DCOS(DSQRT(GRAV*C/M)*(TC-(TN-TDUR))))
         ELSE
```

```
C--EXACT SOLUTION THRUST VELOCITY(TN) AND DIST(TN) (VIGGIANO)------
           VES=DSQRT((T-M*GRAV)/C)*DTANH(TN/M*DSQRT(C*(T-M*GRAV)))
           XES=M/C*DLOG(DCOSH(TN/M*DSQRT(C*(T-M*GRAV))))
          ENDIF
C
C
C--NUMERICAL SUBROUTINE AND COMPARSION TO EXACT SOLUTION-----------
          CALL RK(VN,XN,DT,VNP1,XNP1)
          VERROR=VNP1-VES
          XERROR=XNP1-XES
          IF (VNP1.LE.0.D+00) THEN
             GOTO 999
          ENDIF
          IF (MOD(I,IFREQ).EQ.0) THEN
             WRITE(9,10) TN,T,M,VNP1,XNP1,VES,XES,VERROR,XERROR
          ENDIF
          VN=VNP1
          XN=XNP1
100   CONTINUE
C
C--WRITE SOLUTION DATA---------------------------------------------
C
999   WRITE(9,10) TN,T,M,VNP1,XNP1
10    FORMAT(F8.4,1X,F8.4,1X,F8.4,1X,D14.7,1X,D14.7,1X,D14.7,1X,D14.7

     &D14.7,1X,D14.7)
      END
C
C---4TH ORDER RUNGE-KUTTA NUMERICAL SUBROUTINE---------------------
      SUBROUTINE RK(VN,XN,DT,VNP1,XNP1)
      DOUBLE PRECISION VN,XN,DT,VNP1,XNP1,K1,K2,K3,K4,L1,L2,L3,L4
     &,F,G
      K1=DT*F(VN)
      L1=DT*G(VN)
      K2=DT*F(VN+L1/2.D+00)
      L2=DT*G(VN+L1/2.D+00)
      K3=DT*F(VN+L2/2.D+00)
      L3=DT*G(VN+L2/2.D+00)
      K4=DT*F(VN+L3)
      L4=DT*G(VN+L3)
      XNP1=XN+1.D+00/6.D+00*(K1+2.D+00*K2+2.D+00*K3+K4)
      VNP1=VN+1.D+00/6.D+00*(L1+2.D+00*L2+2.D+00*L3+L4)
      RETURN
      END
C
C--VELOCITY EQUATION DERIVATIVE FUNCTION---------------------------
      DOUBLE PRECISION FUNCTION G(V)
      DOUBLE PRECISION M,C,V,GRAV,T
```

```
      COMMON GRAV,C,M,T
      G=(T-M*GRAV-C*DABS(V)*V)/M
      RETURN
      END
C
C--ALTITUDE EQUATION DERIVATIVE FUNCTION---------------------------
      DOUBLE PRECISION FUNCTION F(V)
      DOUBLE PRECISION V
      F=V
      RETURN
      END
```