Numerical Simulation of Rocket Launch & Re-Entry Using Adaptive Stepping

-Pranshu Pant & Akanksh Shetty

1. Objective

The main objective of this project is to numerically simulate a rocket launch and re-entry using adaptive time stepping. For this project, we are using SpaceX's Falcon 9 rocket as a benchmark for replication.

A reusable rocket launch mission can be broken down into the following critical stages:

- Launch – The static launch of the rocket from the earth's surface
- Stage Separation – The rocket separates into two stages with the upper stage going into orbit and the lower stage returning back to earth. This project primarily focuses on the stage that re-enters the earth's atmosphere
- Propulsive Re-entry – Re-lighting of the engines during re-entry. This is used while the rocket is descending, in order to reduce the velocity sufficiently for safely deploying parachutes.
- Deploy Parachute – Deploying parachutes in order to achieve a soft landing back to earth

Additionally, the benefits and drawbacks of using adaptive time stepping will be discussed. The Runge-Kutta Fehlberg method will be used to numerically model this problem.

2. Motivation

"If one can figure out how to effectively reuse rockets just like airplanes, the cost of access to space will be reduced by as much as a factor of a hundred." – Elon Musk

A single rocket costs approximately $60 million to build and until recently, it could only be used once. This has frequently been compared to using a commercial airliner to fly from one destination to another just once and then sending it straight to the scrapyard. If such a business model was used by airlines, the ticket prices would have gone up to as much as $300,000.
The fuel for the rocket missions cost only around $200,000 (~0.33% of the total cost of the rocket). Majority of the cost goes into building the rocket. Hence, reusability is a key issue. It is the breakthrough needed to revolutionize access to space.
Therefore, this problem is interesting to study and model. In this project we use the adaptive RK45 Fehlberg method to model a multistage rocket launch and reentry.

[Equation 1] and [Equation 2] are the differential equations that govern the problem. [Equation 3] is the position-velocity equation that is used to reduce the order of the system of equations.

$$\dot{x}_1 = \frac{T}{M(t)} - g - \frac{\rho_0 e^{kx_2}}{2 \times M(t)} \times C_d A x_1^2$$

Equation 1: Equation for Ascent

$$\dot{x}_1 = \frac{T'}{M(t)} - g + \frac{\rho_0 e^{kx_2}}{2 \times M(t)} \times C_d A x_1^2$$

Equation 2: Equation for Descent

$$x_1 = \dot{x}_2$$

Equation 3: Position Velocity Equation

In [Equation 1] and [Equation 2], M(t) represents the mass of the rocket which is a function of time. It is very important to note that this is a variable mass problem. This is because the majority of the mass of the rocket is accounted for by the amount of propellent, which varies as the mission progresses.

The g term in [Equations 1] and [Equation 2] is the value of gravity which is calculated as shown in [Equation 4]. In this equation , g_0 = 9.81 m/s^2, h is the altitude in meters and R is the radius of earth in meters.

$$g = g_0 \times \frac{1 - 2h}{R}$$

Equation 4: Calculating gravity based on altitude

T is thrust which is calculated as shown in [Equation 5]. This value is different for ascent and descent. This is because a Falcon 9 rocket uses 9 engines during launch and only 3 engines during re-entry.

$$Thrust\ (T) = No, of\ Engines \times Mass\ Flow\ Rate \times Exhaust\ Velocity$$

Equation 5: Equation to calculate thrust

The $\rho_0 e^{kx_2}$ term in [Equation 1 and 2], accounts for the density which varies with the current altitude of the rocket. The density data obtained from various sources online was discrete. The data from [1] was used for this problem. In order to avoid potential sources of error, the $\rho_0 e^{kx_2}$ term was linearized to obtain equation 6. Regression was then used on the above equation and in order to obtain an exponential fit. In the original form, the value of the constants are as follows: $\rho_0 = 1.315091$ and $K = -1.23393 \times 10^{-4}$. R^2 value of 0.9876 was obtained and hence this suggests that the constants above give a reasonable fit to the discrete data. The plot of the fit is shown in [Figure 1] below.

$$ln\ (y) = ln\ (\rho_0) + (k \times x_2)$$

Equation 6: Linearized from of the equation

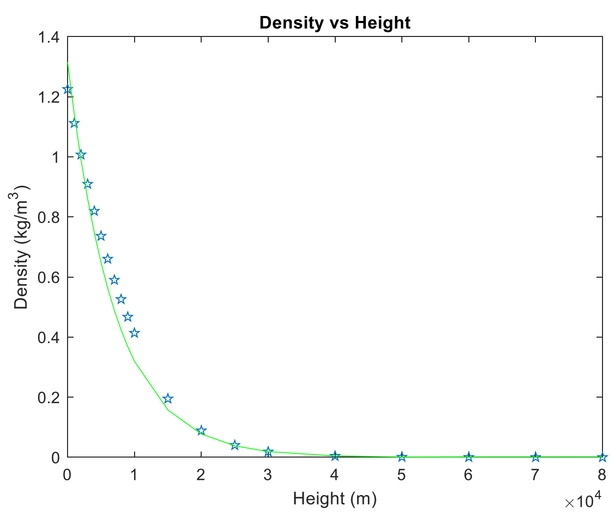

Figure 1:Curve fit to discrete density data

The C_d in [Equation 1] and [Equation 2] is the coefficient of drag. The A stands for area. Naturally, both of these values depend on the flight regime as follows:

Before the parachute is deployed:	**After the Parachute is deployed:**
C_d = 0.74	C_d = 1.75
Area = 43.008 m^2	Area = 10000 m^2

3. Method

German mathematicians Carl Runge and Wilhelm Kutta developed the Runge-Kutta methods to approximate ordinary differential equations. The Runge-Kutta-Fehlberg method was developed by Erwin Fehlberg, another German mathematician in 1969. The Runge-Kutta-Fehlberg method compares two different function approximations at each step. The error in the solution can be estimated by comparing the lower order and higher order method. Thus, using this error, the step size is determined. Two methods of varying the step size have been discussed later in this section.
In the RK45-Fehlberg method, 6 values of K are calculated at each step. [Equations 7-12] show how the different values for K are computed. h in these equations is the step size.

$$K_1 = h \times f(t, x)$$

Equation 7: Equation used to calculate K1

$$K_2 = h \times f\left(t + \frac{1}{4}h, x + \frac{1}{4}K_1\right)$$

Equation 8: Equation used to calculate K2

$$K_3 = h \times f\left(t + \frac{3}{8}h, x + \frac{3}{32}K_1 + \frac{9}{32}K_2\right)$$

Equation 9: Equation used to calculate K3

$$K_4 = h \times f\left(t + \frac{12}{13}h, x + \frac{1932}{2197}K_1 - \frac{7200}{2197}K_2 + \frac{7296}{2197}K_3\right)$$

Equation 10: Equation used to calculate K4

$$K_5 = h \times f\left(t + h, x + \frac{439}{216}K_1 - 8K_2 + \frac{3680}{513}K_3 - \frac{845}{4104}K_4\right)$$

Equation 11: Equation used to calculate K5

$$K_6 = h \times f\left(t + \frac{1}{2}h, x - \frac{8}{27}K_1 + 2K_2 - \frac{3544}{2565}K_3 + \frac{1859}{4104}K_4 - \frac{11}{40}K_5\right)$$

Equation 12: Equation used to calculate K6

Once these values of Ks are obtained, the 4th order function evaluation is calculated as shown in [Equation 13].

$$x(t + h) = x(t) + \frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5$$

Equation 13: 4th order function evaluation

The 5th order function evaluation is calculated as shown in [Equation 14].

$$\underline{x}(t + h) = x(t) + \frac{16}{135}K_1 + \frac{6656}{12825}K_3 + \frac{28561}{56430}K_4 - \frac{9}{50}K_5 + \frac{2}{55}K_6$$

Equation 14: 5th order function evaluation

The higher order function evaluation is assumed to be closer to the actual answer. Hence, the error can be calculated as shown in [Equation 15].

$$x(t + h) - \underline{x}(t + h)$$

Equation 15: Formula to calculate error for current timestep

Once the error has been calculated, this is then used to adapt the time step. For this project, two different methods were used to adapt time step.

The first method is the half-double method. In this method, the step size is halved if the error is greater than the tolerance. If the error is lower than the tolerance, the value of the step size is doubled.

The second method of adapting the time step was the power method [2].

[Equation 16] and [Equation 17] represent the power method. Here, h_0 and h_1 in these equations are the new time step and old time step respectively. The Δ_0 is the value for the desired error and Δ_1 is the value of the current error.

Now, since for an RK45 method the order of the leading order error term is proportional to fifth power of the step size, The ratio of the desired error Δ_0 and the current error $\Delta_1$must also be proportional to the fifth power of the ratio of the desired timestep h_0 and the current time step h_1 [Equation 17]. On rearranging, this equation [Equation 17] we get the power method [Equation 16].

$$h_0 = h_1 \left|\frac{\Delta_0}{\Delta_1}\right|^{0.2}$$

Equation 16: Power method equation

$$\left|\frac{\Delta_0}{\Delta_1}\right| = \left(\frac{h_0}{h_1}\right)^5$$

Equation 17: Rearranged form of the power method equation

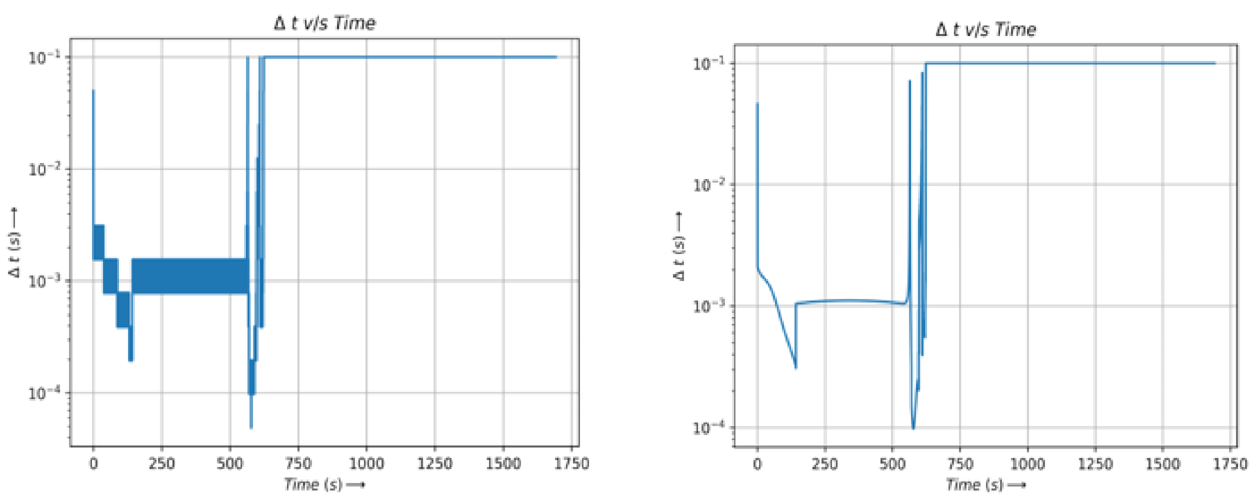

Figure 2:Δt vs Time comparison for half-double method (left) and power method (right)

On comparing the two plots it can be seen that the plot for the half-double method fluctuates much more as compared to the power method [Figure 2]. These fluctuations can be detrimental to the solution and can even result in the solution blowing up. Hence, for the final simulations we utilized the power method for time adaption.

4. Results

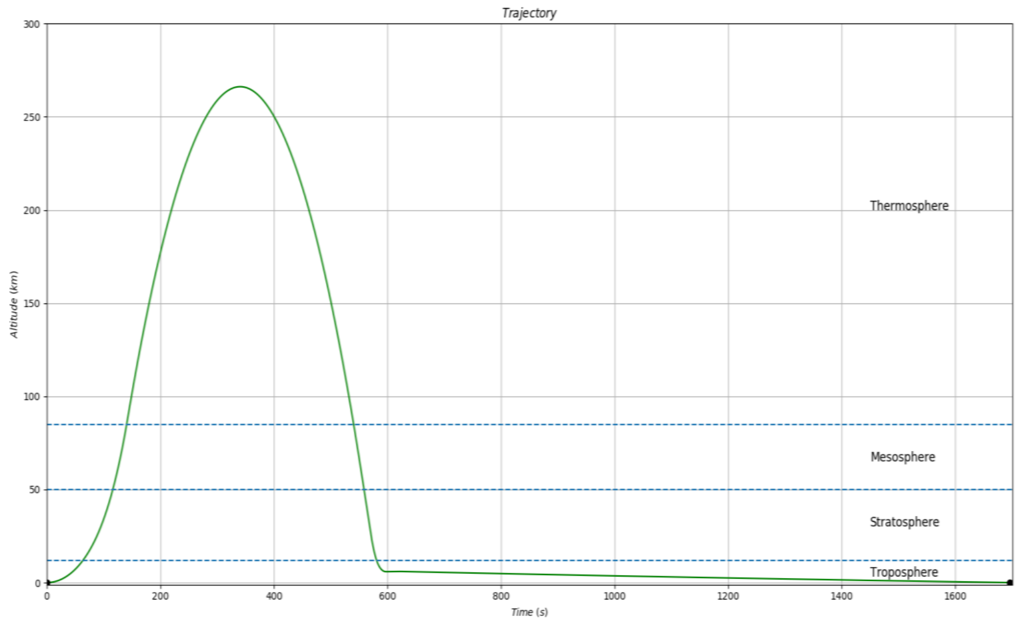

Figure 3:Trajectory of rocket (Altitude vs Time)

[Figure 3] shows the trajectory of the rocket with time. The rocket follows a trajectory that is mostly parabolic except in a few notable regions.
Firstly, during the ascent of the rocket, the altitude increases almost exponentially. Such a trajectory is observed since the mass of the rocket decreases with time. It ejects the propellants in its exhaust and accelerates on account of the conservation of momentum. Furthermore, the drag resistance decreases as the rocket moves into the upper layers of the atmosphere which further speeds up the rocket.

At exactly 162s after launch the 9 Merlin Engines (Falcon-9's engine designation) cut off and stage separation takes place. The second stage goes into orbit and the first stage of the rocket essentially follows an upward parabolic trajectory, attaining a peak altitude and then falling back towards the earth's atmosphere. The rocket attains an apogee of 265 km which corresponds to a low earth orbit (LEO) and is in agreement with the flight path information provided by SpaceX and other sources [6,7,8].

As the first stage falls back to earth it gains velocity due to the Earth's gravity and acquires a peak velocity of around 2000 m/s [Figure 4]. Re-entering the earth's dense atmosphere at such high speeds can be extremely strenuous for the rocket, so at an altitude of 70 km a re-entry burn is initiated which slows down the rocket significantly by re-lighting three of the nine Merlin engines.

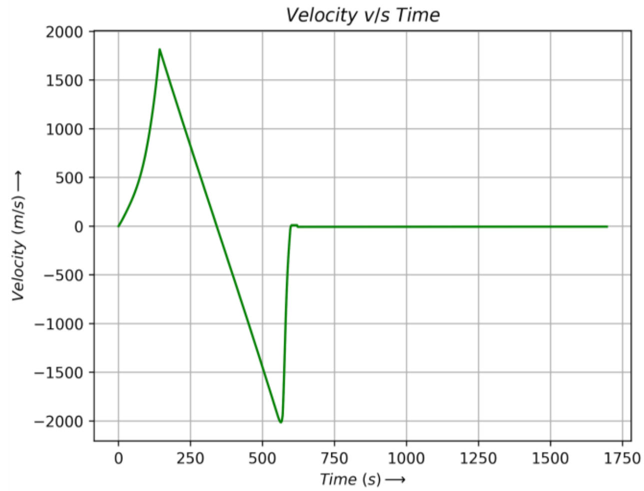

Figure 4: Velocity vs Time graph

When, the speed of the rocket finally goes down below 150 km/h the engines shut down and the parachutes and deployed. The value of 150 km/h was chosen as it is usually considered a safe deployment velocity according to NASA and several rocketry forums [4, 5]. The area of the parachute was determined by the following relation [Equation 18] [3].

$$A = \frac{2gM}{\rho\, C_d V^2}$$

Equation 18: Area of parachute

Also to check the check the sensibility of the values obtained, the area obtained by this formula (A $\approx 10{,}000\ m^2$) was compared to the area of the parachute for the Orion spacecraft ($A \approx 5000\ m^{2)}$). Both values were roughly in the same ballpark after accounting for the difference in mass.

Finally, the rocket with the help of the parachute performs a soft landing with a landing velocity of around 4m/s which is considered a safe landing speed [4].

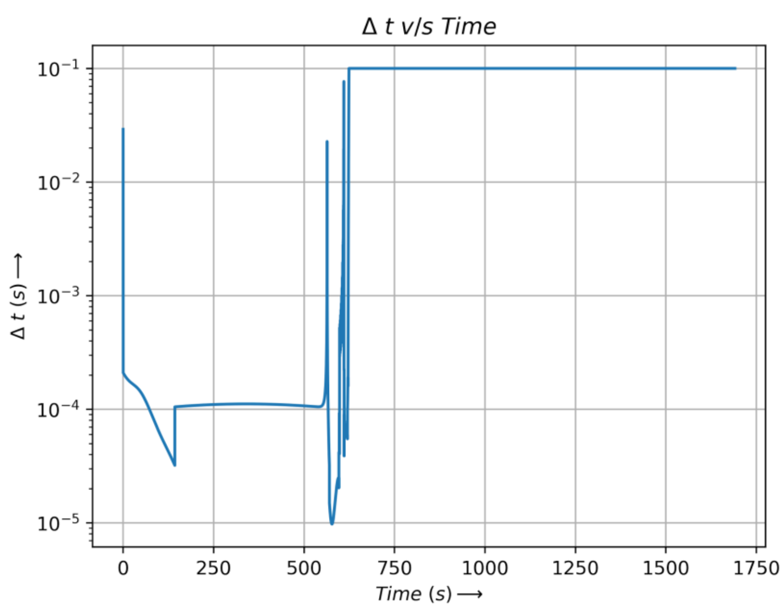

Figure 5:Δt vs time plot for power method

In the $\Delta t\ v/s\ time$ plot the time adaptation can be clearly observed. Also, the time adaptation takes place only during the critical phases of the mission. Major dips in Δt values are observed during stage separation (t = 162s), re-entry burn (t = 520s) and just after opening the parachute (t = 570s) [Figure 5]. Also, the desired increase in Δt value can be observed during non critical phases such as the gradual parachute descent stage (t > 600s).

For, the power method, a desired tolerance (Δ_0) of 10^{-3} was chosen as it provided us with solution times which were manageable for carrying out parametric studies and also didn't affect the solution fidelity significantly. Finally, an upper bound of 10^{-1} and a lower bound of 10^{-6} was set on the step size to ensure stability and manageable solution times.

5. Discussion

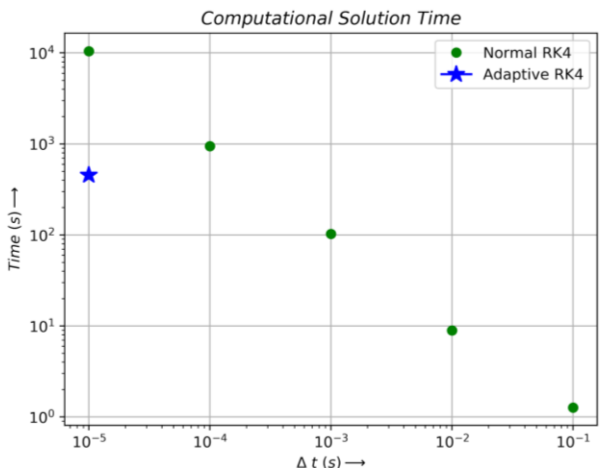

Figure 6:Total Computational time for each time step

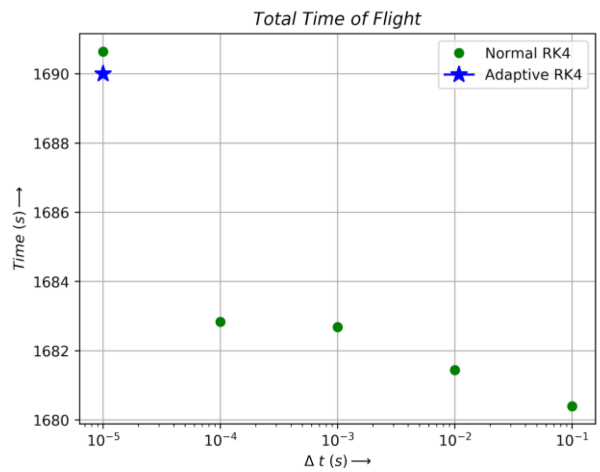

Figure 7:Total time of flight v/s respective Δt for two different methods

[Figure 6] shows the total computational time with different timesteps for the Normal RK4 method compared with minimum time step value encountered during the adaptive RK45 Fehlberg evaluation. On analyzing the plot, it is evident that the computational time scales exponentially with the reduction step size for the normal RK4 method. Furthermore, for a time step of 10^{-5}, the adaptive method takes much less time (more than an order of magnitude less), when compared to the normal RK4 method.

[Figure 7] shows that the solutions evaluated by the Normal RK4 and Adaptive RK4 methods are in close agreement (Relative Error = 0.031%) for a minimum time step of 10^{-5}. Thus, this establishes that the adaptive method achieves a faster solution but not at the cost of solution fidelity.
So, the adaptive method produces a solution which is highly accurate and requires much less computational cost to evaluate. Thus, the RK45 Fehlberg method is a much superior method as compared to the normal RK4.

6. Conclusions

- Adaptive Runge-Kutta-Fehlberg was successfully used for simulating the trajectory of a rocket launch and propulsive re-entry.
- Two different schemes for updating time-step namely, Half-Double and Power method were compared.
- The solution fidelity was largely unaffected by the use of adaptive methods as compared to a normal RK4 method.
- It was observed that for the normal RK4 method, the computational time increased exponentially with decreasing time step. Thus, the adaptive RK45 Fehlberg was more found to be a much more computationally efficient method.
- Hence, adaptive methods can mitigate the trade-off between computational time and solution fidelity by reducing the time step only in the required regions while maintaining a larger time step elsewhere.

7. References

[1] https://www.engineeringtoolbox.com/standard-atmosphere-d_604.html

[2] Chapter 16; section 16.2 of Numerical Recipes book by William H. Press et.al

[3] https://apogeerockets.com/education/downloads/Newsletter149.pdf

[4] https://www.nasa.gov/sites/default/files/atoms/files/orion_parachutes.pdf

[5] https://www.rocketryforum.com/threads/deployment-velocity.23011/

[6] https://www.spacex.com/falcon9

[7] https://www.spaceflightinsider.com/hangar/falcon-9/

[8] https://en.wikipedia.org/wiki/Falcon_9

8. Appendix

Falcon 9 Specifications

No. Engines	9	units
Length	41.2	m
Mass Flow Rate (Per Engine)	270	kg/s
Mass Flow Rate (Total) x 9	2430	kg/s
Exhaust Velocity	2770	m/s
Thrust	6731100	N
Burn Time	162	s
Re-Entry Burn Time (3 Engines)	50	s
Empty Mass	25,600	kg
Propellant Mass	395,700	kg
Re-entry Propellant Requirement	40500	kg
Propellant for Ascent	355200	kg
Total Weight Second Stage	96570	kg
Area	43.008	m^2
Area with Parachute	10000	m^2
Parachute opening velocity	150	km/h