RBFTA Revista Brasileira de Física Tecnológica Aplicada

Revista Brasileira de Física Tecnológica Aplicada
ISSN: 2358-0089

https://periodicos.utfpr.edu.br/rbftat

Development of a Model Rocket Trajectory Simulation Tool with Python

Domingos Sávio Pinheiro do Nascimento Júnior
saviojunior123@hotmail.com
orcid.org/0000-0002-4167-4971
Universidade Federal do Ceará, Ceará, Brazil.

Jasson Fernandez Gurgel
jassongurgel@alu.ufc.br
orcid.org/0000-0003-2439-2750
Universidade Federal do Ceará, Ceará, Brazil.

Marcello Reis
mcarv.reis@gmail.com
orcid.org/0000-0002-1132-9034
Universidade Federal do Ceará, Ceará, Brazil.

Claus Franz Wehmann
claus.wehmann@ufc.br
orcid.org/0000-0001-8756-9387
Universidade Federal do Ceará, Ceará, Brazil.

ABSTRACT

In any space mission, the design of the trajectory that the spacecraft will travel across is one of the essential and most important steps. For the launch of smaller rockets, this is also a key analysis to be done during the development of the project, ensuring that the rocket achieves its flight objectives and enabling the definition of a safety radius based on its landing site. Therefore, the present work proposes the development of a tool to simulate the trajectory of small rockets that is satisfactory for use in academic projects, acting as a free alternative to similar software available on the market. For this task, the Python programming language is used due to its open-source characteristic and shorter execution times compared to similar languages during the execution of numerical computation methods, which are necessary for modeling the rocket's flight dynamics. The results found with the use of the tool for simulating the trajectory of a model rocket are sufficient for the purpose of this work, reaching similar findings to those of simulations carried out in OpenRocket (a JAVA rocket trajectory simulation software widely used in academia). Moreover, a Monte Carlo simulation is performed to estimate the impact point dispersion radius.

KEYWORDS: trajectory; simulation; rocket dynamics.

INTRODUCTION

The trajectory simulation is a way of analyzing how the main variables that constitute the rocket's flight dynamics are performing prior to the launch. That way, if there is anything abnormal or that does not meet the mission goals, the simulation is reevaluated and the input data, such as the rocket's body design and atmospheric model, are adjusted in order to address the issue.

In the case of passive control rockets, the focus of this paper, which do not have attitude control mechanisms during flight, simulation is essential, since after taking off, it is not possible to make changes in the direction of the vehicle, which shows the significance of a well-executed simulation to avoid unforeseen events.

Rockets of this type are the most developed in the amateur and academic fields and are becoming increasingly more popular in universities to be used in research and competitions. Despite that, there are only a few tools that are able to simulate a model rocket trajectory, and they have a different focus, which leads to insufficient results. Also, more robust simulations are available in some software, but they are not available to everyone. The main tool used by students and enthusiasts is *OpenRocket*, but it still lacks some important features that competitions ask for, like tridimensional trajectory plots and stochastic simulations. This causes many teams to resort to manufacturing their own codes to meet these needs.

Therefore, this paper aims to address those issues with softwares to be used by students that have the following premises:

- Similar simulation results compared to *OpenRocket*;
- Complete personalization of the input data by the user;
- Trajectory visualization on *Google Earth*;
- Estimation of the impact point dispersion with stochastic simulations.

STATE OF ART

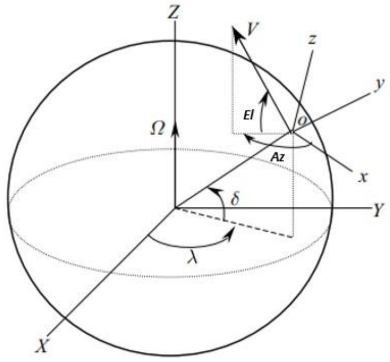
REFERENCE FRAMES

To simulate a rocket's trajectory, it is necessary to express the reference frames upon which the flight dynamics equations will be integrated. In this case, two main frames are considered, an inertial and a moving one.

The inertial frame of reference (IF) is stationary, that is, its origin is not moving or rotating. Here, the origin is located at the Earth's center and the x and y-axis are within the plane of the equator. The z-axis points in the same direction as Earth's angular velocity vector, the x-axis points to the Greenwich meridian and the y-axis completes the right-handed system. In this frame, the spherical coordinates are defined by two angles, latitude (δ) and longitude (λ). This frame of reference is also known as the equatorial rotating geocentric system, which rotates with respect to the celestial sphere. However, since this work deals with short-haul flights, this movement can be disregarded.

The moving frame of reference, or the local frame (LF), is rotating with respect to the inertial frame. Since it is centered at the launch site on the Earth's surface, the frame rotates with the angular velocity of the Earth's rotational motion (Ω). The z-axis points to the zenith, the x-axis points south, and the y-axis points east. In this case, the azimuth (Az) and elevation (El) angles, regarding the direction and inclination of the rocket during launch, define the spherical coordinates.

Figure 1 – Reference Frames



Source: Adapted from Tewari (2007).

FLIGHT PHASES

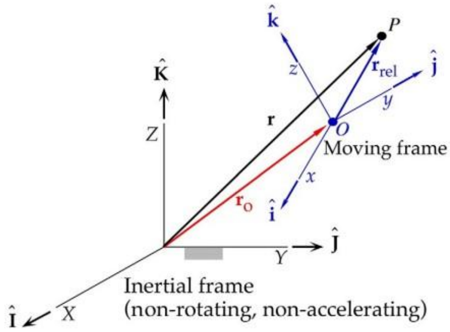
The flight of a small rocket can be divided into four phases (NISKANEN et al., 2009), namely:

- Launch: The rocket engine is ignited on a launch rail;
- Propelled Flight: The rocket continues to accelerate due to the burning of fuel;
- Free Flight: The fuel is completely burned and the rocket starts to decelerate.
- Recovery: After reaching its apogee, the vehicle accelerates down to the ground and its recovery system is activated for landing.

FLIGHT DYNAMICS

The flight dynamics of a rocket comprises the equations of translational and rotational motion of the vehicle, which are deduced based on the inertial and local reference systems.

Figure 2 – Inertial and Moving Frames



Source: Curtis (2013).

To find the trajectory of the rocket, its position and velocity at each time-step must be found. Those variables are calculated with the integration of the translational equation of motion.

The rockets' acceleration relative to the inertial frame can be expressed as:

$$\vec{a} = m(\vec{a}_0 + \dot{\vec{\Omega}} \times \vec{r}_{rel} + \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_{rel}) + 2\vec{\Omega} \times \vec{v}_{rel} + \vec{a}_{rel}) \tag{1}$$

where each term is described by:

- $\vec{a}$ refers to the vehicle's acceleration relative to the IF;
- $\vec{a}_0$ refers to the LF acceleration relative to the IF;
- $\dot{\vec{\Omega}}$ refers to the LF angular acceleration relative to the IF;
- $\vec{\Omega} \times (\vec{\Omega} \times \vec{r}_{rel})$ refers to the centripetal acceleration;
- $2\vec{\Omega} \times \vec{v}_{rel}$ refers to the Coriolis acceleration;
- $\vec{a}_{rel}$ refers to the vehicle's acceleration relative to the LF.

The forces acting on the rocket are considered to be the thrust (T), the drag (D) and the weight (W), which are calculated as it follows:

$$\vec{T} = T_c cos(\alpha) \frac{\vec{v_t}}{||\vec{v_t}||} \tag{2}$$

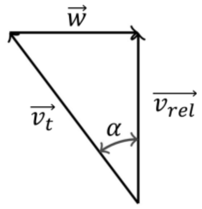
$$\vec{D} = -\frac{1}{2}\rho || \vec{v_t} || S C_D \vec{v_t} \tag{3}$$

$$\vec{W} = -\frac{\mu_{\oplus} m}{(r_o + z)^2} \hat{k} \tag{4}$$

where T_c is the thrust curve absolute value, z is the rocket's altitude, S is the reference section area of the vehicle's body, ρ is the atmospheric density, C_D is the drag coefficient, and $\mu_{\oplus}$ is the geocentric gravitational constant, given by the product of the gravitational constant and Earth's mass.

In addition, $\vec{v_t}$ is the total or resultant velocity of the rocket, which is influenced by the wind velocity, $\vec{w}$. The wind also produces an angle of attack, α, between the rocket's velocity and the total velocity.

Figure 3 – Rocket's velocity components.



Source: Adapted from Pinto et al. (2015).

Applying Newton's second law of motion, the translational motion equation can be found.

$$\sum F = m\vec{a} = m(\vec{a}_0 + \vec{\Omega} \times \vec{r}_{rel} + \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_{rel}) + 2\vec{\Omega} \times \vec{v}_{rel} + \vec{a}_{rel}) \quad (5)$$

Since the trajectory is simulated considering how an observer in the LF would see the flight, then equation (2) can be expressed as:

$$m\vec{a}_{rel} = -m\begin{pmatrix} \vec{a}_0 + \dot{\vec{\Omega}} \times \vec{r}_{rel} + \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_{rel}) \\ +2\vec{\Omega} \times \vec{v}_{rel} \end{pmatrix} + T + D + W \quad (6)$$

Considering that the model rocket will have its movement restricted by the guiding rods, during this short period of time equation becomes:

$$\begin{aligned} m\vec{a}_{rel} &= -m\,\vec{a}_0 - m\,\dot{\vec{\Omega}} \times r - 2m\,\vec{\Omega} \times \vec{v}_{rel} - m\,\vec{\Omega} \times (\vec{\Omega} \times \vec{r}) \\ &-\tfrac{1}{2}\rho||\,\vec{v}\,||SC_D\,\vec{v} + (T_p + ||\,\vec{W}\,||sin(El))\,\hat{t} \end{aligned} \quad (7)$$

where $\hat{t}$ is the unit vector defined by the guiding rod's orientation.

$$\hat{t} = (cosElcosAz, cosElsenAz, senEl) \quad (8)$$

After leaving the guiding rod, the wind starts to influence the rocket's flight and creates an angle of attack between the rocket's velocity and the resultant velocity, as shown in **Figure 3**. Under stable conditions, the rocket enters a damped oscillatory motion compose by a restoring moment (M_s) and a damping moment (M_a), which makes the rocket oscillate relative to the resultant velocity (FEODOSIEV; SINIAREV, 2014).

It is important to notice that only the pitch moment is considered through the calculation of the angle of attack. The yaw and roll moments are disregarded in this work.

$$M_s = N(x_{cp} - x_{cm}) = \tfrac{1}{2}\rho v_r^2 S C_{N_\alpha} \alpha (x_{cp} - x_{cm}) \quad (9)$$

$$M_a = \frac{1}{2}\rho VS\dot{\alpha}\sum_{i=1}^{n} C_{N_{\alpha_i}}(x_{cp_i} - x_{cm})^2 + \dot{\alpha}\dot{m}(x_{nozzle} - x_{cm})^2 \qquad (10)$$

Knowing the moments given by equations (9) and (10), the oscillatory movement equation is given by (PINTO et al., 2015):

$$I\ddot{\alpha} + (\frac{1}{2}\rho VS\sum_{i=1}^{n} C_{N_{\alpha_i}}(x_{cp_i} - x_{cm})^2 + \dot{m}(x_{nozzle} - x_{cm})^2 + \dot{I})\dot{\alpha} + \frac{1}{2}\rho V^2 SC_{N_\alpha}(x_{cp} - x_{cm})\alpha = 0 \qquad (11)$$

The components x_{cm}, x_{cp}, x_{nozzle} are, respectively, the position of the center of mass, the center of pressure and the rocket's nozzle relative to the tip of the nosecone. N corresponds to the normal air pressure force acting on the rocket and C_{N_α} is the normal force coefficient derivative, that accounts for the shape of each structure of the rocket (BARROWMAN, 1970). The term n corresponds to the total number of external components of the vehicle.

ATMOSPHERE

The atmospheric properties, such as air density, temperature and pressure, are calculated based on the International Standard Atmosphere (ISA) (NASA, 2021a).

The presence of air generates a drag force that opposes the rocket's movement in its flight direction, as stated in **Equation 3**. Note that since the rockets fly at high speeds, the drag is proportional to the velocity squared (MARION; THORNTON, 2004).

The horizontal wind model is utilized in this work, which considers that the air is moving parallel to the launch site local horizon. Its speed and intensity are given as input data.

ROCKET DESIGN

In order to calculate the forces and moments acting on the rocket's body, its mass, center of mass, center of pressure, moment of inertia and drag coefficient must be defined.

The mass is given by the sum of the rocket's structural mass and the propellant mass, which varies with time.

The center of mass is also given by the sum of the center of mass for the vehicle's structure and of its fuel.

The center of pressure is calculated using Barrowman's method (BARROWMAN, 1970a). All external components have static centers of pressure, except for the fins, which will depend on the Mach number (NISKANEN et al., 2009).

The moment of inertia is given by the sum of transversal inertial moments for each component.

The drag coefficient is calculated based on empirical equations found by Purdue (2008), which are dependent on the Mach Number (M), the angle of attack (α) and the initial drag coefficient (C_{D_0}). **Equations 12–14** demonstrate the calculation for the coefficient for subsonic, transonic, and supersonic flow regimes, respectively.

$$C_D = 0.2083M^3 + 0.0445M^2 - 0.1494M + 0.18 + C_{D_0} \\ +\alpha 10^{-1}(0.0034M^2 - 0.0003M + 0.4283) \tag{12}$$

$$C_D = 0.5293M^{-1.2374} + C_{D_0} \\ +\alpha 10^{-1}(0.0034M^2 - 0.0003M + 0.4283) \tag{13}$$

$$C_D = 0.355M^{-0.5162} + C_{D_0} \\ +\alpha 10^{-1}(0.0034M^2 - 0.0003M + 0.4283) \tag{14}$$

The value of C_{D_0} is given by sum of the initial drag coefficients for each external component.

IMPACT DISPERSION

The model rocket's impact point is calculated in a determinist way by the simulator. Therefore, to consider uncertainties, a Monte Carlo simulation is performed by adding noise to input data and creating a new sample of probable impact points. By using the multivariate normal distribution (UNICAMP, 2020), dispersion ellipses are created to indicate the probability density of impact regions.

METHODOLOGY

The execution of the rocket trajectory simulation is done from scripts in Python that collect and process user-supplied input data, integrate the equations of motion and, finally, demonstrate the results.

The tool is composed by five main scripts. The first one collects the input data and stores it into dictionaries. The second one calculates variables related to the rocket's structure, such as center of mass and pressure. The third one accommodates the atmosphere and wind models. The fourth one utilizes information from all codes to integrate the equations of motion. Lastly, the fifth script handles important functions used across all codes and displays the results.

ROCKET COMPONENTS

In order to compare the results found in this work to the ones simulated with *OpenRocket*, a control model rocket was developed. **Tables 1, 2 and 3** provide the properties and dimensions of the nosecone, body tube, fins, and parachutes.

Table 1 - Nosecone and Body Tube properties

	Nosecone	Body Tube
Shape	Cone	Cylindrical
Mass (kg)	1	6
External Radius (cm)	8	8
Thickness (cm)	0.3	0.3
Length (cm)	50	200

Source: Own Authorship.

Table 2 - Fins properties

	Fins
Shape	Trapeizodal
Number	4
Mass (kg)	0.1
Thickness (cm)	0.3
Height (cm)	15
Root Chord (cm)	35
Tip Chord (cm)	10
Sweep Length (cm)	20

Source: Own Authorship.

Table 3 - Parachutes properties

	Main Parachute	Pilot Parachute
Radius (cm)	100	30
Mass (kg)	0.500	0.125
Drag Coefficient	0.8	0.8

Source: Own Authorship.

Regarding the motor, its dimensions and its thrust curve also have to be given as input data.

Figure 4 – Thrust curve data

```
Time, Thrust,
0.0, 1499.926,
0.7, 1468.427,
1.1, 1450.428,
1.3, 1441.429,
4.9, 1279.437,
5.2, 1265.937,
5.4, 1256.938,
8.3, 1126.444,
8.5, 1117.445,
10.0, 1049.948,
10.001, 0.0,
```

Source: Own Authorship.

Table 4 - Motor properties

	Fins
Diameter (cm)	127
Length (cm)	154.6
Empty Mass (kg)	9.379
Propellent Mass (kg)	10
Average Thrust (N)	1275
Burn Time (s)	10

Source: Own Authorship.

LAUNCH SITE AND IMPACT POINT

The wind is oriented in the same direction of launch and it has a constant velocity of 5 m/s.

The model rocket is launched at latitude -5.92° and longitude -36.16°. The guiding rod has 6 meters of length and it is oriented at an azimuth of 45° and an elevation of 85°.

The impact point dispersion is simulated by applying the Monte Carlo process with a maximum range of 1° in the azimuth and elevation angles. The final randomized sample is composed by 1000 points of impact based on a ballistic flight without the wind influence.

INTEGRATION METHOD

To integrate the equations of motion, the Velocity Verlet method is used.

$$\begin{aligned} v(t+\tfrac{1}{2}\Delta t) &= v_t + \tfrac{1}{2}a(t)\Delta t \\ x(t+\Delta t) &= x_t + v(t+\tfrac{1}{2}\Delta t)\Delta t \\ a(t+\Delta t) &= f(x(t+\Delta t)) \\ v(t+\Delta t) &= v(t+\tfrac{1}{2}\Delta t) + \tfrac{1}{2}a(t+\Delta t)\Delta t \end{aligned} \qquad (15)$$

RESULTS

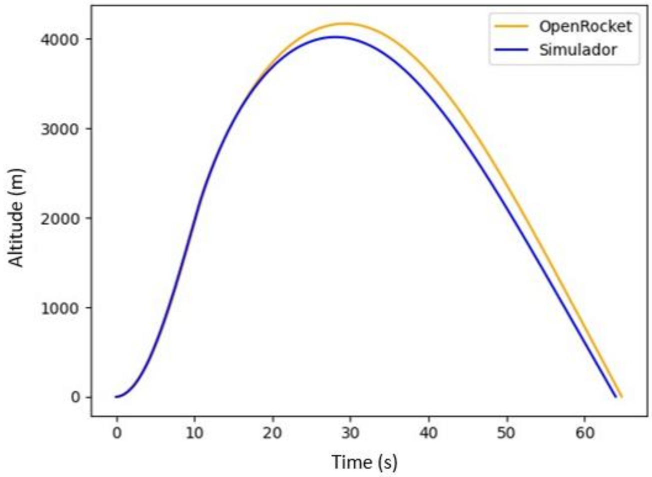
To validate the simulator, three simulations are executed and the results are compared with *OpenRocket*.

The tool developed in this work will be referred as simulator in the results section.

BALLISTIC FLIGHT WITHOUT WIND

This simulation is performed with disregard to the wind influence and without the use of the recovery system.

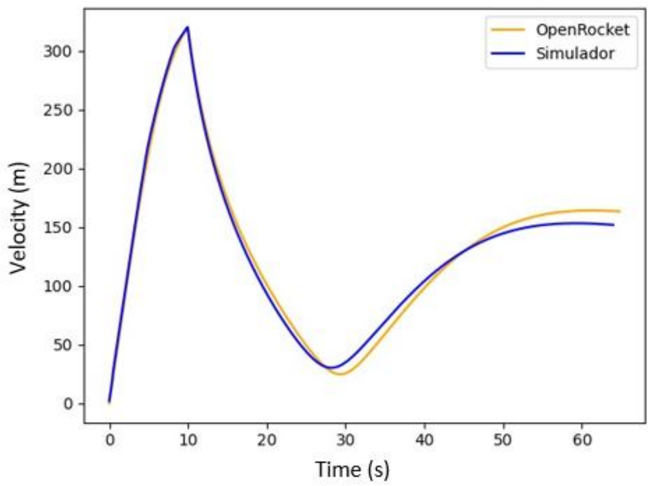
Figure 5 – Altitude comparison



Source: Own Authorship.

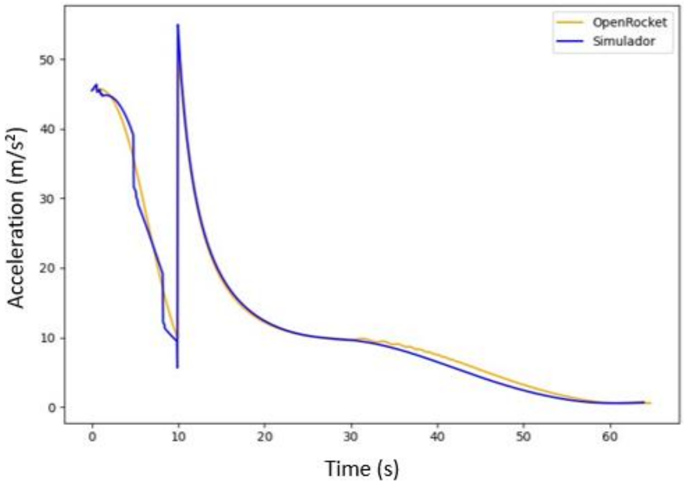
The altitude curves are very similar between both plots, with a variation of 3.6% at the apogee.

Figure 6 – Velocity comparison



Source: Own Authorship.

Figure 7 – Acceleration comparison



Source: Own Authorship.

The velocity and acceleration curve profiles are also very similar with a maximum variation of 7% in both cases.

Figure 8 – Trajectory comparison

Source: Own Authorship.

The flight path is similar, but it is clear that the impact point is more distant to the launch site in the simulator result, as shown in **Table 5**.

Table 5- Location of impact point

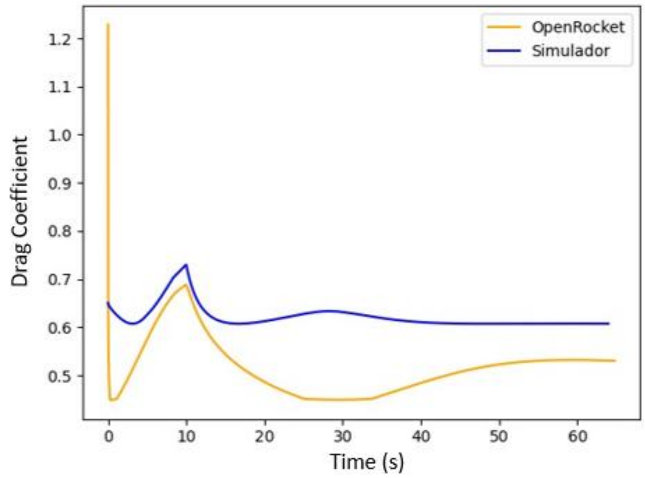
	Simulator	OpenRocket
East Direction (m)	1184.2	1029.2
North Direction (m)	1287.8	1046.9

Source: Own Authorship.

The differences found between the results occur largely due to the drag coefficient model applied. In this work an empirical model is used, whereas *OpenRocket* uses a model following Barrowman's method (BARROWMAN, 1967).

That influences the equations of motion and leads to small divergences. The drag coefficient for this simulation is shown in **Figure 9**.

Figure 9 – Drag coefficient comparison

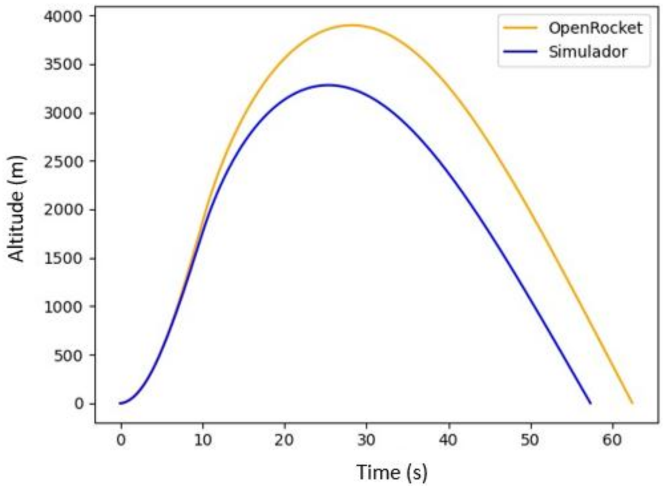


Source: Own Authorship.

BALLISTIC FLIGHT WITH WIND

This simulation is performed without the use of the recovery system and considering a constant wind speed of 5m/s in the guiding rods direction.

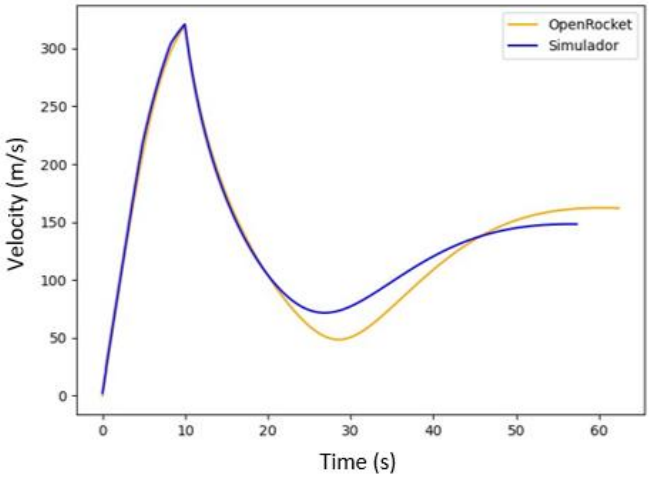
Figure 10 – Altitude comparison



Source: Own Authorship.

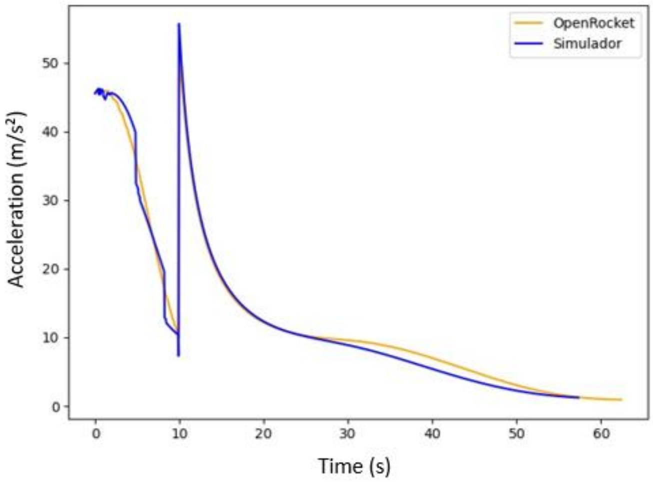
The difference between the apogee values is increased in this simulation, with a variation of 15.9%.

Figure 11 – Velocity comparison



Source: Own Authorship.

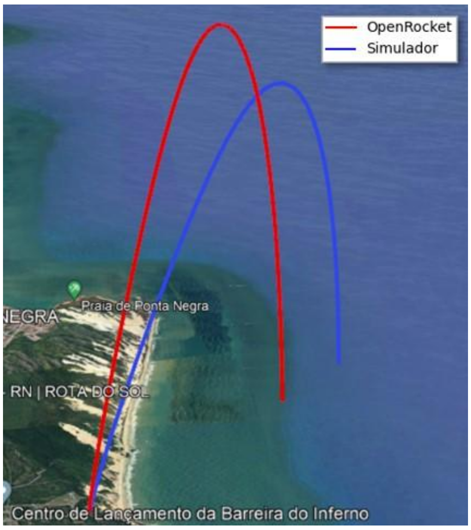
Figure 12 – Acceleration comparison



Source: Own Authorship.

The velocity and acceleration curve profiles have a slightly bigger divergence compared to the first simulation, with maximum variation of 9% for the velocity and 7.9% for the acceleration.

Figure 13 – Trajectory comparison



Source: Own Authorship.

The flight path is also similar, but it is clear that the impact point is more distant to the launch site in the simulator result, as shown in **Table 6**.

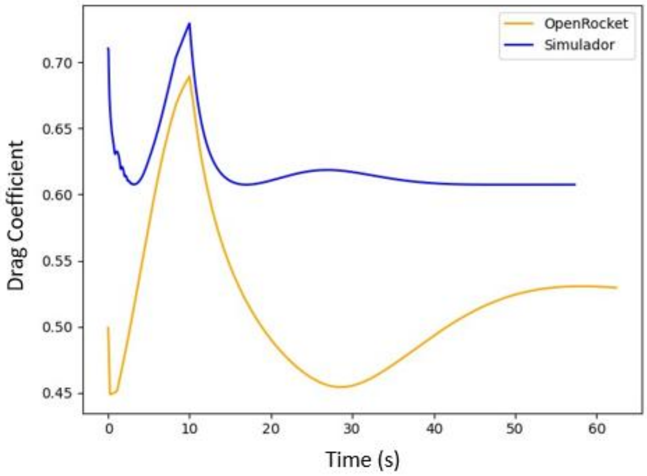
Table 6- Location of impact point

	Simulator	OpenRocket
East Direction (m)	2704.9	1942.5
North Direction (m)	2796.3	1958.2

Source: Own Authorship.

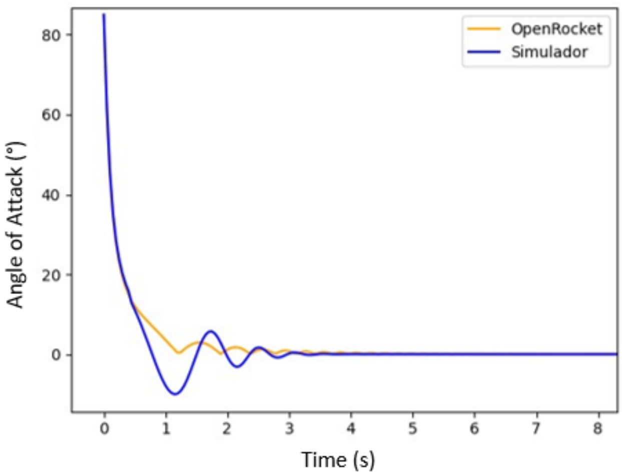
In this simulation, it is experienced bigger divergences between the results. This is caused not only by the different drag coefficient model, but also by the angle of attack development throughout the flight.

Figure 14 – Drag coefficient comparison



Source: Own Authorship.

Figure 15 – Angle of attack comparison



Source: Own Authorship.

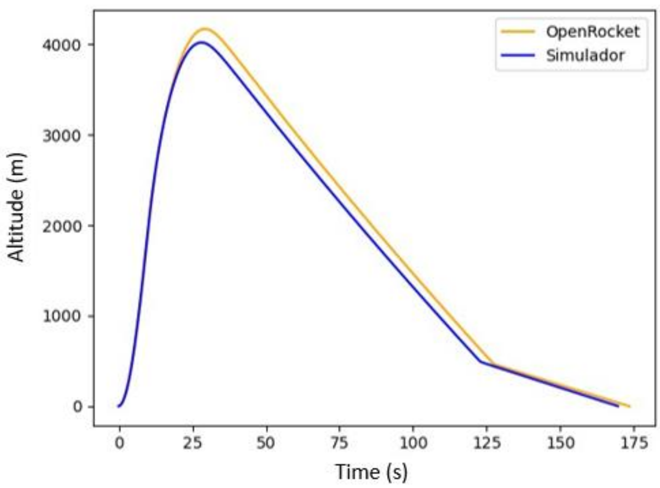
As shown in **Figure 15**, *OpenRocket* does not consider negative values for the angle of attack, not presenting the rocket's oscillation approached in this work. That increases the differences found between the simulations.

FLIGHT WITH PARACHUTES AND WITHOUT WIND

This simulation is performed with disregard to the wind influence and with the use of the recovery system, which is composed by two parachutes.

The pilot parachute is activated at the apogee and the main parachute is triggered 500 meters in altitude during the descent flight.

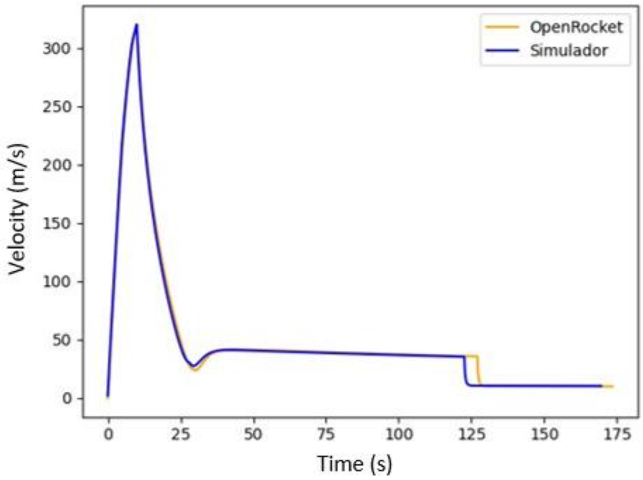
Figure 16 – Altitude comparison



Source: Own Authorship.

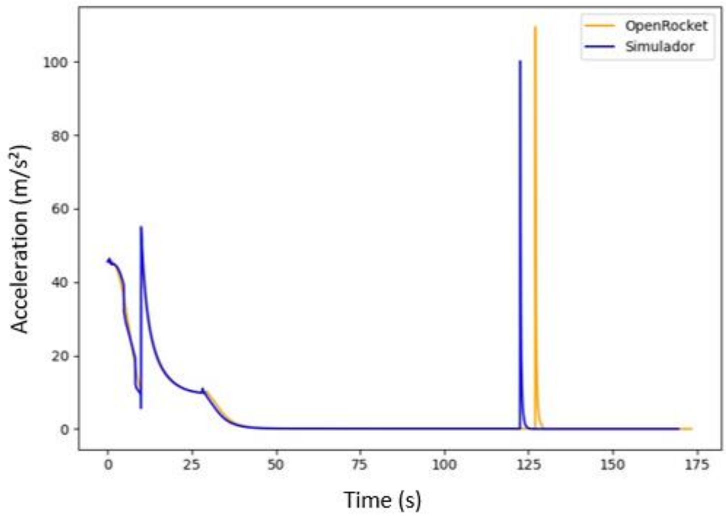
The apogee results are the same found as the ballistic flight without wind simulation, since it also does not consider wind effects. After this point, the altitude curves between each simulator are identical, with only a difference of 10 seconds to open each parachute and to the impact point.

Figure 17 – Velocity comparison



Source: Own Authorship.

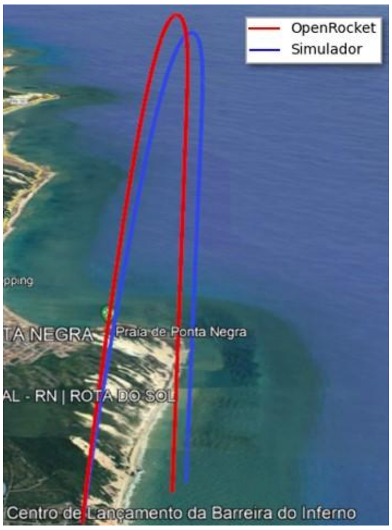
Figure 18 – Acceleration comparison



Source: Own Authorship.

The velocity and acceleration are again very similar, with really small differences between the curves' profiles. The main divergence is in the time to open the parachutes and to impact the ground.

Figure 19 – Trajectory comparison



Source: Own Authorship.

The trajectory is also very similar, with the rocket flying a little further in the simulator result, as shown in **Table 7**.

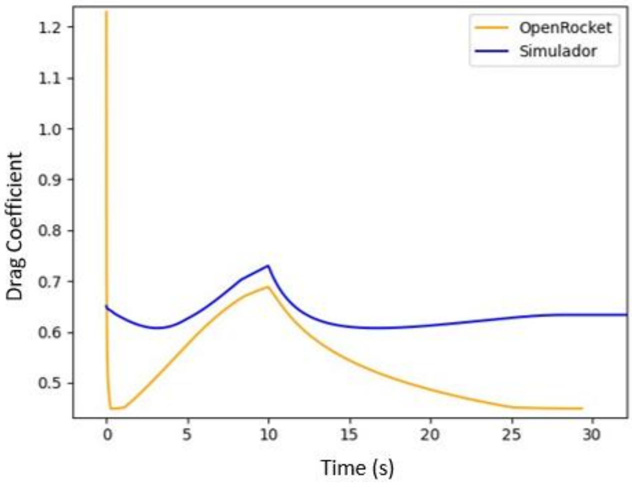
Table 7- Location of impact point

	Simulator	OpenRocket
East Direction (m)	844.7	712.7
North Direction (m)	910.1	724.8

Source: Own Authorship.

These small variations are once again due to the different drag coefficient models.

Figure 20 – Drag coefficient comparison

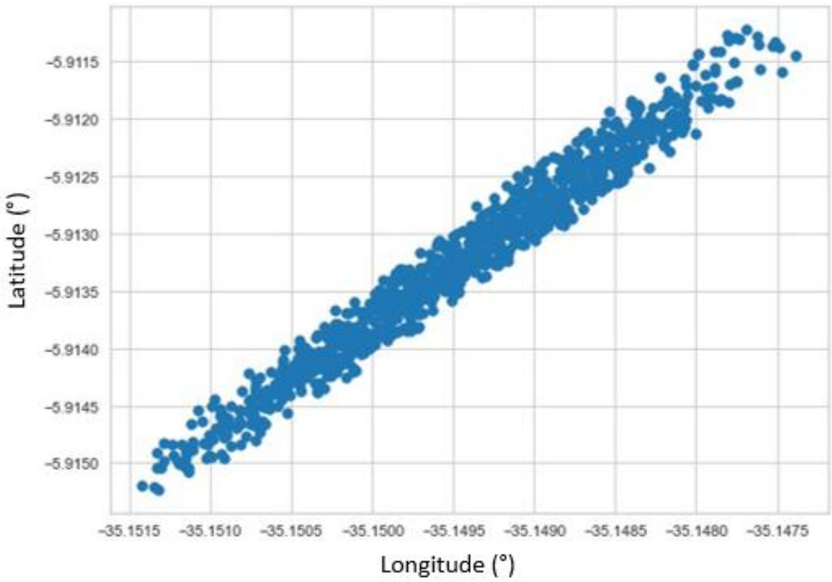


Source: Own Authorship.

IMPACT DISPERSON

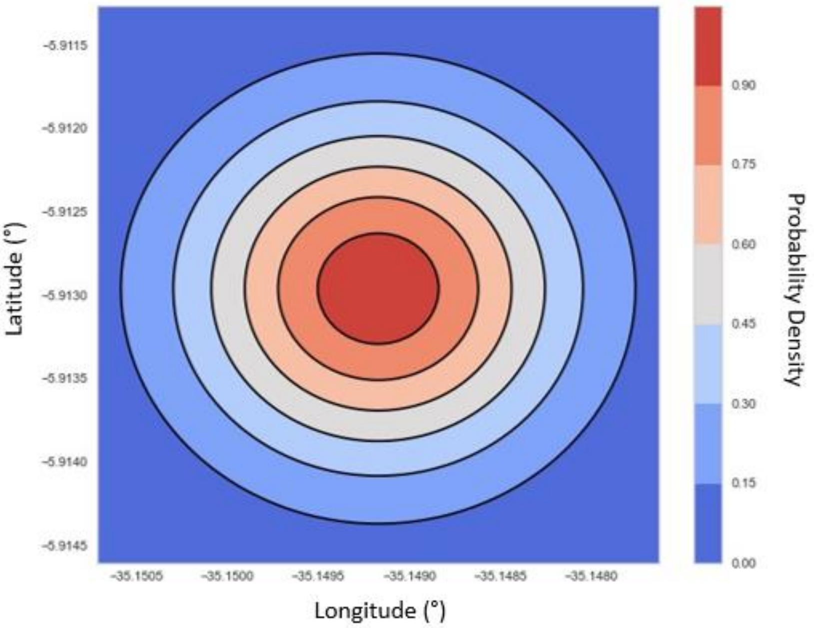
This stochastic simulation performed resulted in a nominal impact point with latitude of -5.913° and -35.149°. The randomized sample and the probability ellipses are shown in **Figures 21 and 22**.

Figure 21 – Randomized sample of impact points



Source: Own Authorship.

Figure 22 – Probability ellipses



Source: Own Authorship.

The results demonstrate that there is an approximated variation of 0.025° in both axes for the outermost ellipse, which covers 99.7% of the data. This leads to an estimate radius of dispersion of 2.8 kilometers surrounding the nominal impact point.

CONCLUSION

The tool accomplished all the goals proposed for this work. Its simulations have similar results compared to *OpenRocket*. As an open-source project, all variables can be customized. The 3D trajectory is plotted alongside the Google Earth view. Finally, it was estimated an impact point dispersion radius for the flight.

The divergences between the results from this tool and *OpenRocket* are due to the different model of calculating the drag coefficient and the angle of attack. These differences are great for users to be able to compare results between those tools and, if the user prefers, the models can be changed in the script.

Desenvolvimento de Ferramenta em Python para Simulação da Trajetória de Foguetes Acadêmicos

RESUMO

Dentro de qualquer missão espacial, a definição da trajetória que o veículo irá percorrer é um dos passos essenciais e mais importantes. Da mesma forma, para o lançamento de foguetes de menor porte também é fundamental que seja realizada a simulação de sua trajetória durante o desenvolvimento do projeto, assegurando que o foguete atinja seus objetivos de voo e possibilitando a definição de um raio de segurança com base na sua região de pouso. O presente trabalho propõe, então, o desenvolvimento de uma ferramenta para o cálculo da trajetória de foguetes de pequeno porte que seja satisfatória para uso em projetos acadêmicos, atuando como uma alternativa livre para softwares do tipo disponíveis no mercado. Para isso, é utilizada a linguagem de programação *Python* em razão de sua característica *open-source* e de tempos de execução menores comparados aos de linguagens similares durante a execução de métodos computação numérica, que são necessários para modelagem da dinâmica de voo do foguete. Os resultados encontrados com o uso da ferramenta para simulação da trajetória de um foguete acadêmico se mostram satisfatórios para o propósito desse trabalho, atingindo resultados semelhantes aos de simulações feitas no *OpenRocket* (software simulação de da trajetória de foguetes em *JAVA*, amplamente utilizado no espaço acadêmico). Ainda, é feita uma simulação de Monte Carlo, em que é estimado um raio de dispersão do ponto de impacto do foguete no solo.

PALAVRAS-CHAVE: trajetória; simulação; dinâmica de foguetes.

Desarrollo de una Herramienta Python para Simular la Trayectoria de Cohetes Académicos

RESUMEN

Dentro de cualquier misión espacial, definir la trayectoria que recorrerá el vehículo es uno de los pasos esenciales y más importantes. Asimismo, para el lanzamiento de cohetes más pequeños, también es fundamental simular su trayectoria durante el desarrollo del proyecto, asegurando que el cohete alcanza sus objetivos de vuelo y posibilitando la definición de un radio de seguridad en función de su región de aterrizaje. El presente trabajo propone, por tanto, el desarrollo de una herramienta para el cálculo de la trayectoria de pequeños cohetes que sea satisfactoria para su uso en proyectos académicos, actuando como una alternativa libre a software del tipo disponible en el mercado. Para ello se utiliza el lenguaje de programación Python por su característica de código abierto y tiempos de ejecución más cortos en comparación con lenguajes similares a la hora de ejecutar métodos de cómputo numérico, que son necesarios para modelar la dinámica de vuelo del cohete. Los resultados encontrados con el uso de la herramienta para simular la trayectoria de un cohete académico son suficientes para el propósito de este trabajo, llegando a resultados similares a los de las simulaciones realizadas en OpenRocket (software de simulación de la trayectoria de cohetes en JAVA, ampliamente utilizados en el espacio académico). Asimismo, se realiza una simulación de Monte Carlo, en la que se estima un radio de dispersión desde el punto de impacto del cohete contra el suelo.

PALABRAS CLAVE: trayectoria; simulación; dinamica de cohetes.

REFERENCES

BARROWMAN, J. **Calculating the center of pressure of a model rocket.** [S. l.]: Centuri Engineering Company, 1970.

BARROWMAN, J. **The practical calculation of the aerodynamic characteristics of slender finned vehicles.** NASA Technical Reports Server, 1967. Available at: https://ntrs.nasa.gov/citations/20010047838

FEODOSIEV, V. I.; SINIAREV, G. B. **Introduction to rocket technology.** [S. l.]: Academic Press, 2014.

NASA. **Earth Atmosphere Model.** 2021. Disponível em: https://www.grc.nasa.gov/www/k-12/ airplane/atmosmet.html.

NISKANEN, S. et al. **Development of an Open Source model rocket simulation software.** Thesis (Master), 2009.

MARION, J. B.; THORNTON, S. **Classical Dynamics of Particles and Systems;** 5 ed., 2004.

PINTO, T. M. et al**. A model and sounding rocket simulation tool with Mathematica.** Thesis (Master), 2015.

CURTIS, H. **Orbital mechanics for engineering students.** [S. l.]: Butterworth-Heinemann, 2013.

TEWARI, A. **Atmospheric and space flight dynamics.** [S. l.]: Springer, 2007.

UNICAMP. **The Multivariate Normal Distribution.** 2020. Disponível em: https: //www.ime.unicamp.br/~cnaber/mvnprop.pdf.

Recebido: 15 de fevereiro de 2022.
Aprovado: 2 de julho de 2022.

DOI:

Como citar: JUNIOR, D S P N; GURGEL, J F; REIS, M; WEHMANN, C F, Development of a Model Rocket Trajectory Simulation Tool with Python, **Revista Brasileira de Física Tecnológica Aplicada**, Ponta Grossa, v. 9, n.2, p. 48-70, dez. 2022.

Contato: Marcello Reis: mcarv.reis@gmail.com