

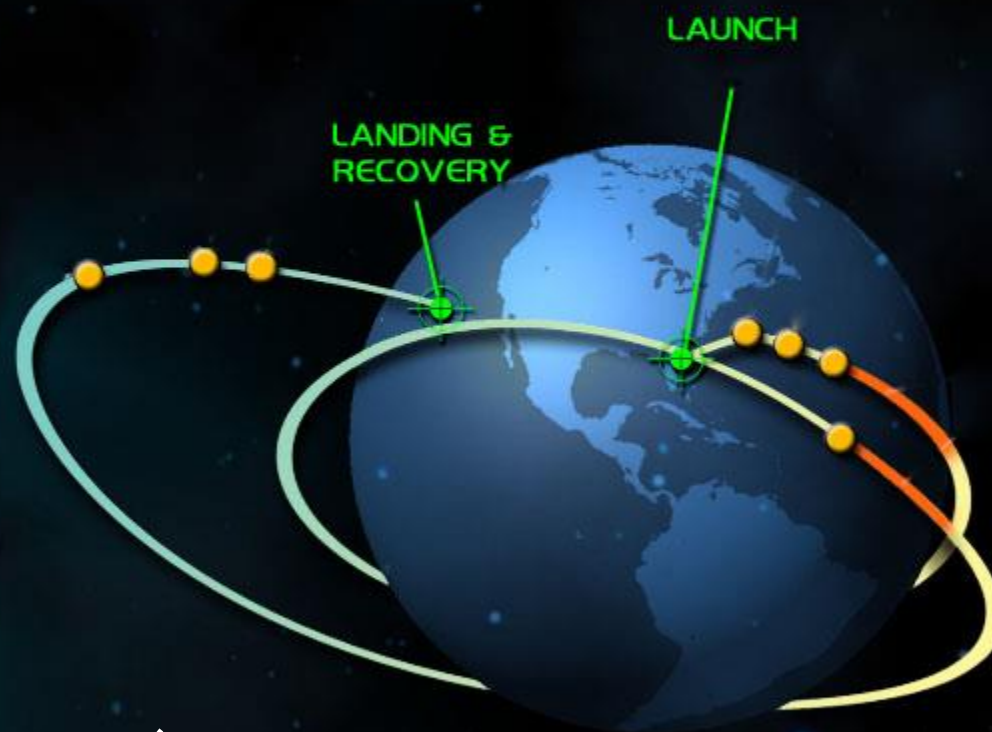
SIMULAÇÃO DE TRAJETÓRIA E ANÁLISE DE DISPERSÃO DE IMPACTO APLICADOS AO FOGUETE MODELISMO



UFC



ENCONTROS
UNIVERSITÁRIOS



Domingos Sávio Pinheiro do Nascimento Júnior

Co-Autores: Anderson Leandro, Paulo Mateus, João Victor Nogueira, Sidney Roberto

Orientador: Claus Franz Wehmann

INTRODUÇÃO

Foguete Modelismo

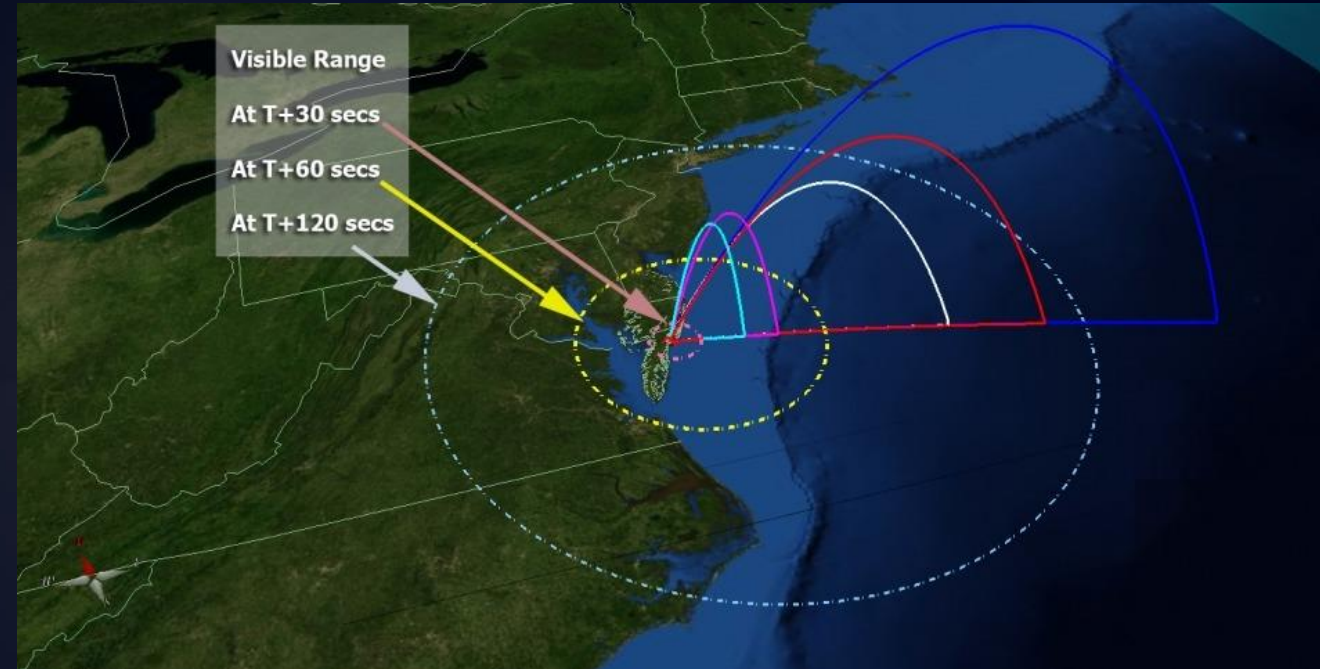
- Foguetes construídos em escala reduzida e com complexidade e custos também diminutos, utilizados em competições ou em pesquisas e projetos científicos.
- Aprendizado prático de Tecnologia de Foguetes.
- Pesquisa e desenvolvimento no campo Aeroespacial.
- A atividade vem crescendo no Brasil, principalmente, no âmbito acadêmico, com o surgimento de novas equipes, como o GDAe da UFC.



INTRODUÇÃO

Simulação de Trajetória

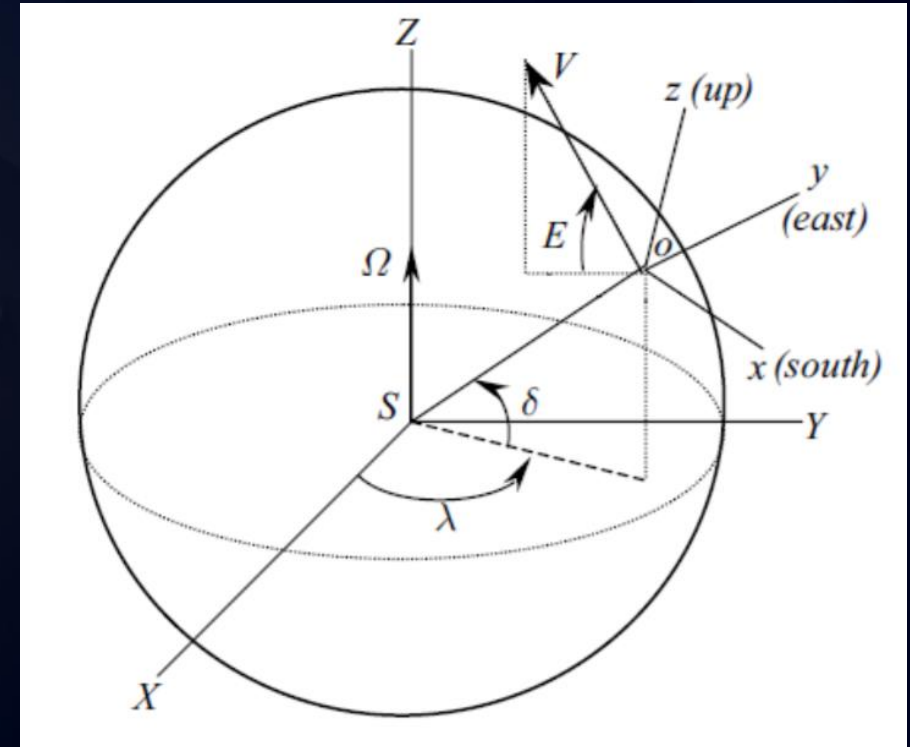
- A simulação de trajetória desses foguetes é um conceito altamente importante por apresentar previamente o caminho percorrido pelo veículo, possibilitando prever diversas variáveis de voo, bem como o ponto de impacto, o que é essencial para que acidentes sejam evitados.
- Apesar de sua importância, esse tópico ainda é pouco difundido e estudado, em razão, principalmente, de sua alta complexidade.
- Essa simulação é, geralmente, feita com auxílio de software já disponíveis, como o OpenRocket.



INTRODUÇÃO

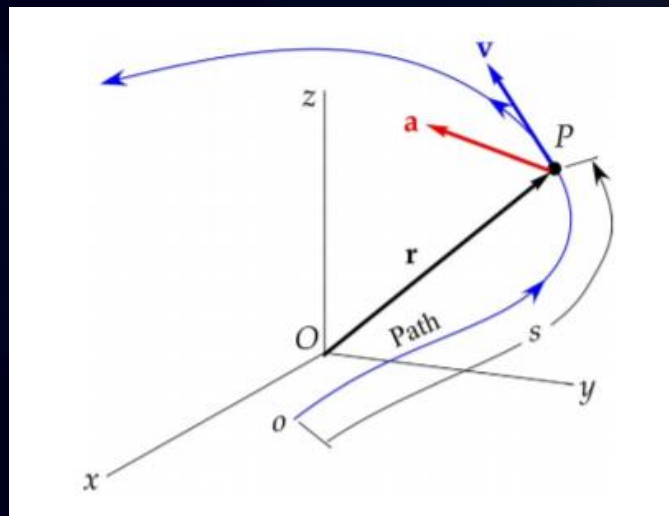
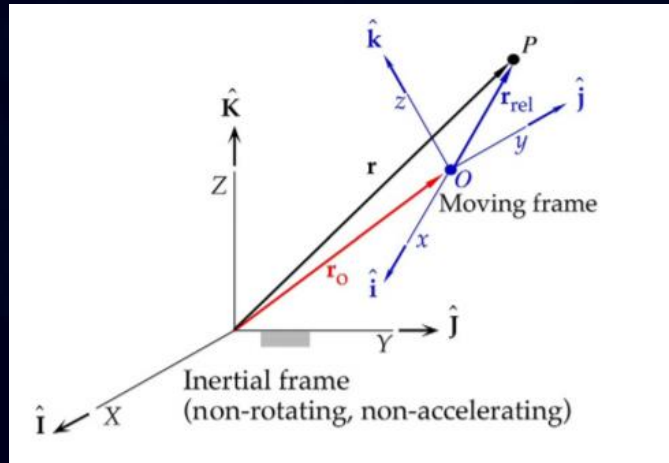
Simulação de Trajetória

- Para implementação e interpretação da simulação realizada, é importante ter o conhecimento de conceitos básicos do campo aeroespacial, visto que a simulação é feita com 6 "DOF", considerando a Terra esférica.
- São considerados um plano inercial, com origem no centro da Terra, e um plano que se move de acordo com a velocidade do rotação do planeta, com origem na superfície terrestre.
- Almeja-se, então, encontrar a posição do foguete em relação ao tempo, relativos ao plano na superfície.



INTRODUÇÃO

Simulação de Trajetória



$$\mathbf{r} = \mathbf{r}_O + \mathbf{r}_{rel}$$

$$\mathbf{v} = \mathbf{v}_O + \frac{d\mathbf{r}_{rel}}{dt}$$

$$\mathbf{a} = \mathbf{a}_O + \frac{d^2\mathbf{r}_{rel}}{dt^2}$$

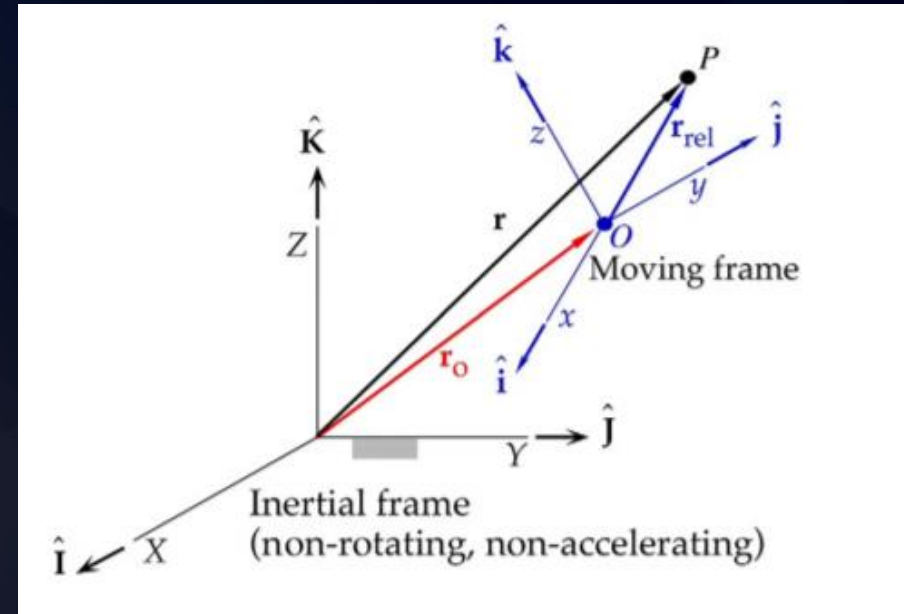
$$\mathbf{a} = \mathbf{a}_O + \dot{\boldsymbol{\Omega}} \times \mathbf{r}_{rel} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}_{rel}) + 2\boldsymbol{\Omega} \times \mathbf{v}_{rel} + \mathbf{a}_{rel}$$



INTRODUÇÃO

Simulação da Trajetória

- Com as equações apresentadas, é possível encontrar a **equação de movimento do foguete**.
- Integrando a equação, podemos encontrar sua velocidade (v_{rel}) e, integrando novamente, sua posição (r_{rel}).
- Esse par de variáveis (r_{rel}, v_{rel}) compõe o que é conhecido como **state vector**, que define o formato e a orientação em qualquer ponto da trajetória.
- Possuindo o vetor em função do tempo, pode-se encontrar o caminho percorrido pelo veículo.



$$m \frac{\partial^2}{\partial t^2}(r_{rel}) = -m \frac{d^2}{dt^2}(R_o) - m \frac{d}{dt}(w) \times r - mw \times (w \times r) - 2mw \times \frac{\partial}{\partial x}(r_{rel}) + \frac{GMm}{(R_o + r_{rel})^2} + Empuxo + Arrasto$$



OBJETIVOS

- Esse trabalho objetiva estabelecer um método de análise prévia da trajetória e análise de dispersão de impacto de foguetes de escala reduzida, simulando o voo por meio de ferramentas computacionais e prevendo os prováveis pontos de impactos por meio de métodos estatísticos, de modo a obter de dados mais realísticos e uma visualização mais harmônica dos dados.
- Objetiva-se, então, desenvolver uma estratégia que realize a simulação de maneira coerente com a teoria e obtendo os resultados necessários para participação e pontuação na COBRUF, de modo que possa ser uma solução mais viável a ser adotada por equipes que utilizem somente softwares livres, como o *OpenRocket*.



METODOLOGIA

➤ O processo de análise do voo do veículo será dividido em quatro etapas:

- Simulação da Trajetória
- Tratamento dos Dados
- Análise de Dispersão de Impacto
- Visualização dos Dados Finais

Simulação
da Trajetória

Tratamento
dos Dados

Análise de
Dispersão de
Impacto

Visualização
dos Dados

OpenRocket 

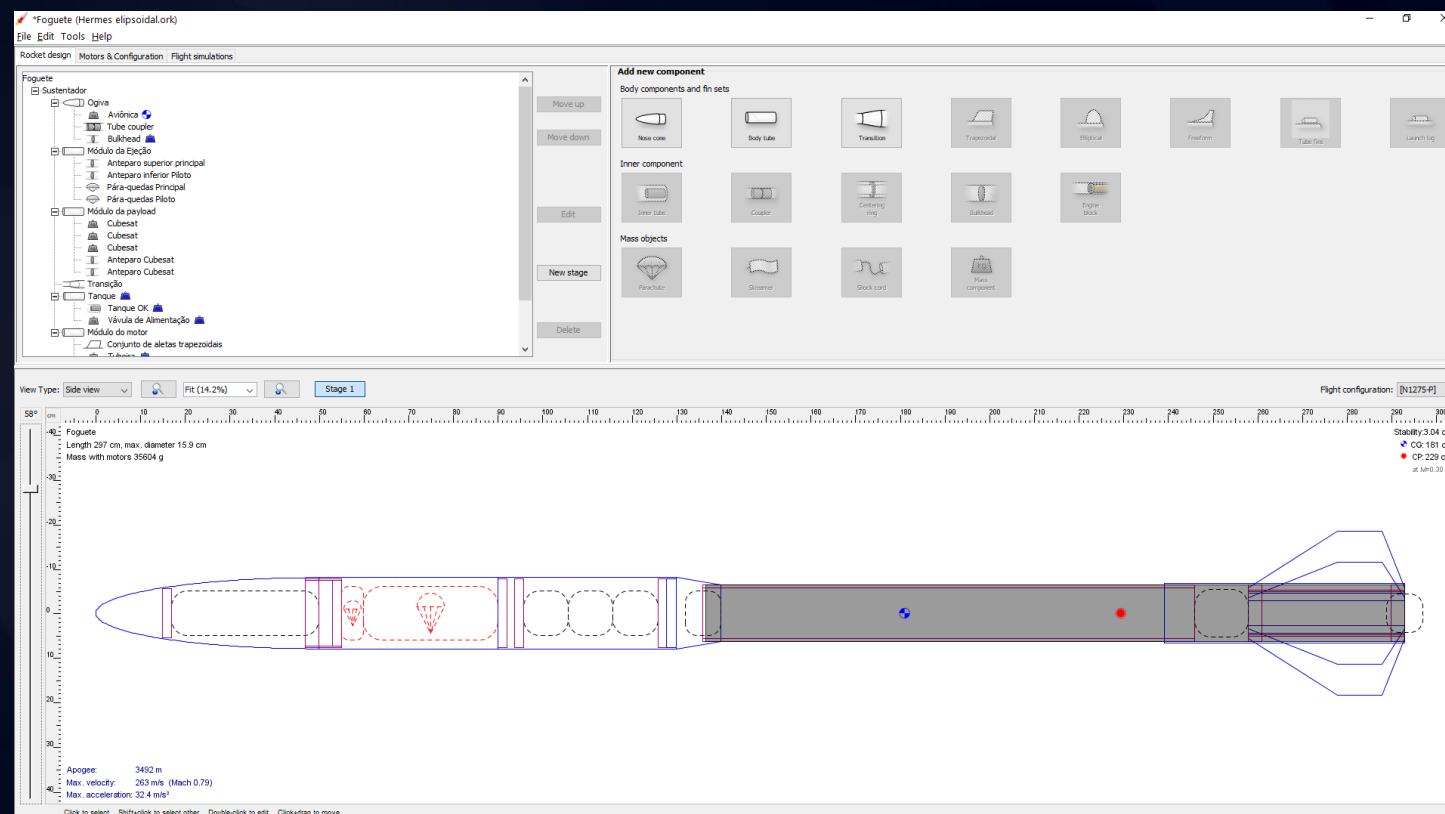
 python™



METODOLOGIA

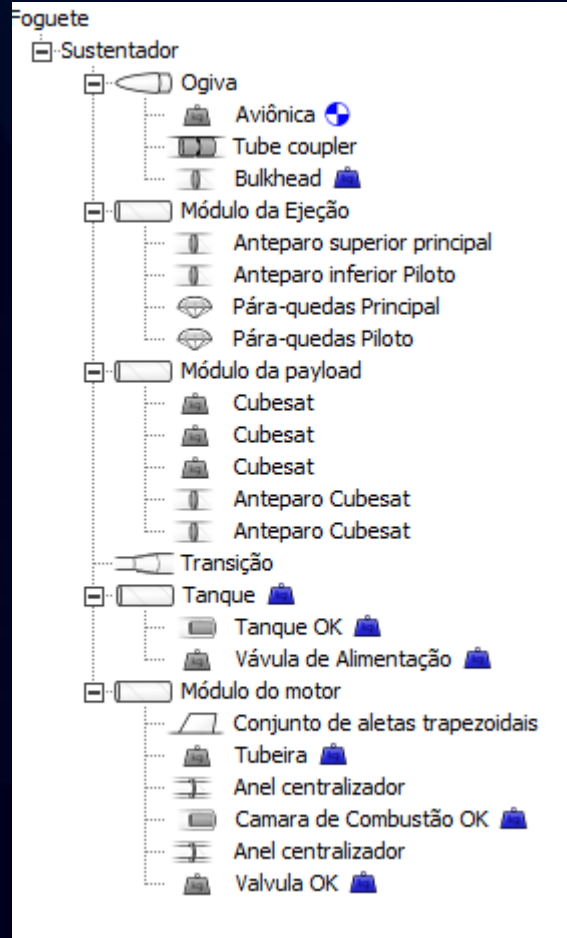
Simulação da Trajetória

- Para a simulação de trajetória é utilizado o software *OpenRocket*, em que é fornecido como entrada, inicialmente, os **materiais e dimensões de cada componente** do foguete, bem como o **tipo de propulsão e os dados do motor**.
- São fornecidos os **dados do ambiente** de lançamento do foguete e definido o passo para a integração da equação de movimento, determinando o vetor de estado da trajetória do foguete.
- Os dados podem, então, serem exportados para uma planilha.



METODOLOGIA

Simulação da Trajetória



Total impulse: 12750 Ns (25% N)

Avg. thrust: 1275 N

Max. thrust: 1500 N

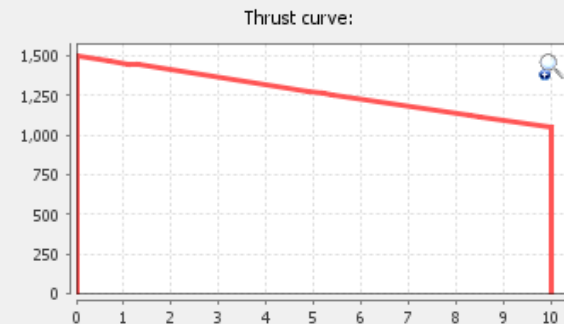
Burn time: 10 s

Launch mass: 19739 g

Empty mass: 9739 g

Data points: 11

@File: agnesteorico.eng, @Pts-I: 12, @Pts-O: 11, @Sm: 0, @CO: 5%
@TI: 12749.9, @TIIa: 12749.89, @TIIe: 0.0%,
@ThMax: 1499.926, @ThAvg: 1274.862, @Tb: 10.001



Wind

Average windspeed: 4 m/s

Standard deviation: 0.4 m/s

Turbulence intensity: 10 % Medium

Wind direction: 90 °

Launch site

Latitude: -5.92 ° N

Longitude: -35.2 ° E

Altitude: 50 m

Launch rod

Length: 600 cm

☒ Always launch directly up-wind or down-wind

Angle: 5 °

Direction: 90 °

Atmospheric conditions

☒ Use International Standard Atmosphere

Temperature: 25 °C

Pressure: 1013 mbar



METODOLOGIA

Tratamento dos Dados

- A etapa de tratamento de dados objetiva verificar a coerência dos dados obtidos e, então, corrigi-los caso necessário.
- Para tal, é utilizado um código em *python* que recebe os dados e possibilita que as devidas alterações sejam realizadas.

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from pandas import DataFrame as df
import os
import time

timestr = time.strftime("%Y%m%d%H%M%S")

x, y = [], [] # EAST, NORTH
x2, y2 = [], [] # LONGITUDE, LATITUDE
z = [] # ALTITUDE

file = input('Type the file name right below (containing the respective file extension).\nOBS. ')
with open(file, 'r') as arq:
    for data in arq:
        line = data.split(',')
        x2.append(float(line[4])) #LONGITUDE
        y2.append(float(line[3])) #LATITUDE
        x.append(float(line[1])) #EAST
        y.append(float(line[2])) #NORTH
        z.append(float(line[0])) #ALTITUDE

new_file_e = input('How would like to name the Google Earth track file?:\n - ')
new_file = new_file_e + timestr + '.txt'

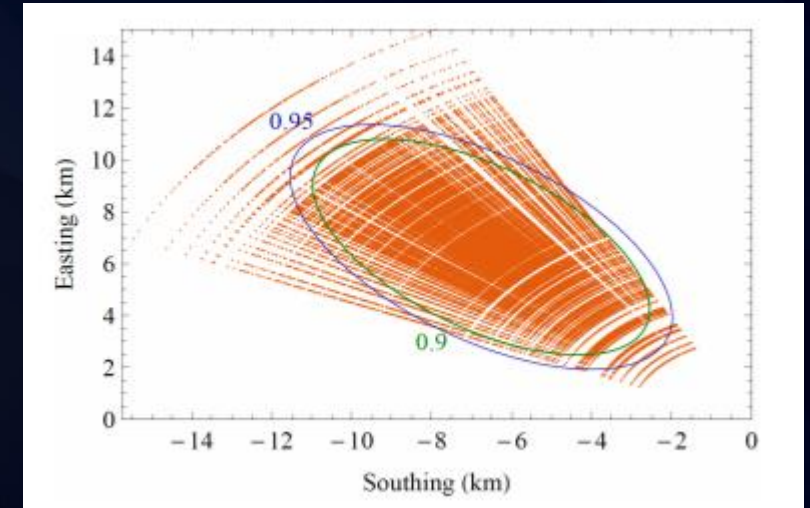
try:
    os.mkfifo(new_file)
except AttributeError:
    print("We're not able to create the file. Please create a file named " + new_file + ".")
finally:
    with open(new_file, 'w') as new_arq:
        for j in range(len(x)):
            new_arq.write(str(x2[j]) + ',' + str(y2[j]) + ',' + str(z[j]) + '\n')
```



METODOLOGIA

Análise da Dispersão de Impacto

- A análise de dispersão de impacto é uma das mais importantes etapas do processo, já que demonstra os mais prováveis pontos de impacto do foguete na superfície, permitindo ter um controle de onde o foguete irá cair e fazer as devidas alterações para que aquela área seja deserta.
- Para tal, métodos estatísticos codificados em python são utilizados. No caso, o Método de Monte Carlo é utilizado, em que simulações aleatórias do mesmo foguete, Hermes, são realizadas e os pontos de impacto resultantes são plotados em um plano, tendo como resultado final a formação de uma elipse contendo a maioria dos pontos encontrados.
- A probabilidade do foguete aterrissar naquela elipse depende do nível de confiabilidade requisitado.



METODOLOGIA

Análise da Dispersão de Impacto

- Mostrar como é feita a análise e foto do código



METODOLOGIA

Visualização dos Dados

- A etapa final é a de visualização de dados, em que o output final da simulação e da análise de dispersão de impacto são apresentados.
- Para isso, códigos em *python* e em *html* são utilizados para plotagem em 3D e plotagem no Google Earth.

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from pandas import DataFrame as df

x, y, z = [], [], []
file = input('Type the file name right below (containing the respective file extension).\nOBS.
with open(file, 'r') as arq:
    for data in arq:
        line = data.split(',')
        x.append(float(line[1]))
        y.append(float(line[2]))
        z.append(float(line[0]))

for i in range (0, len(z)-1):

    if z[i] == max(z):
        x_ap = x[i]
        y_ap = y[i]

table_data = {'START': [0, 0, 0], 'APOGEE': [max(z), x_ap, y_ap], 'END': [0, x[len(y)-1], y[len(x)-1]]}
print(df(table_data, index = ['ALTITUDE', 'NORTH DIRECTION', 'EAST DIRECTION']))

fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')
ax.set_title('Trajectory Simulation', color = 'c')
ax.set_xlabel('North (m)', color = 'black')
ax.set_ylabel('East (m)', color = 'black')
ax.set_zlabel('Altitude (m)', color = 'black')
ax.plot(x, y, z, 'blue')
plt.show()
```



METODOLOGIA

Visualização dos Dados

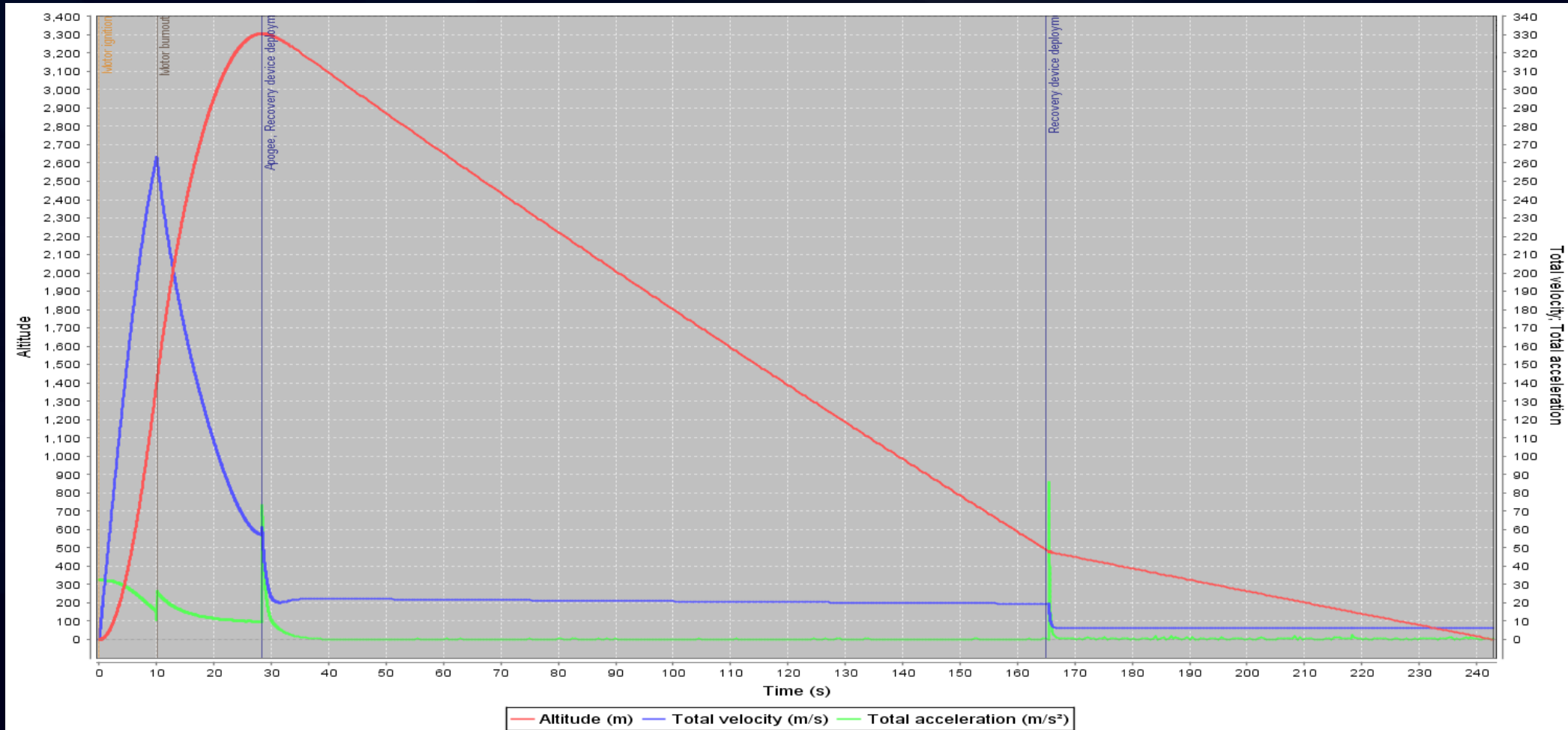
```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name><![CDATA[traj_e]]></name>
    <visibility>1</visibility>
    <open>1</open>
    <Folder id="Tracks">
      <name>Tracks</name>
      <visibility>1</visibility>
      <open>0</open>
      <Placemark>
        <name><![CDATA[traj_earth]]></name>
        <Snippet></Snippet>
        <description><![CDATA[&nbsp;]]></description>
        <Style>
          <LineStyle>
            <color>ff0000e6</color>
            <width>4</width>
          </LineStyle>
        </Style>
        <LineString>
          <tessellate>0</tessellate>
          <altitudeMode>absolute</altitudeMode>
          <coordinates>
</coordinates>
          </LineString>
        </Placemark>
      </Folder>
    </Document>
  </kml>
```



DISCUSSÃO

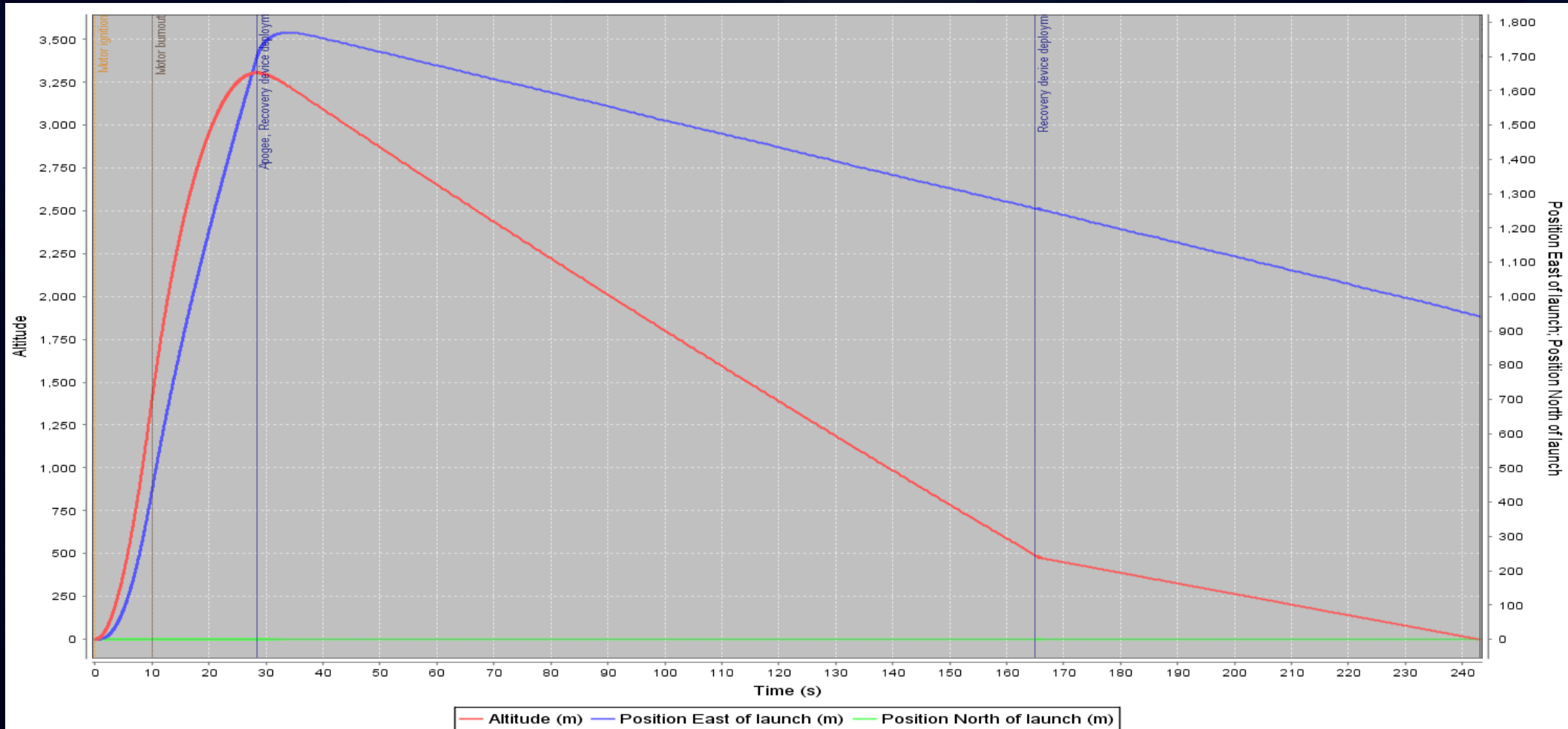
Simulação da Trajetória

| Velocity off rod | Apogee | Velocity at depl... | Optimum delay | Max. velocity | Max. acceleration | Time to apogee | Flight time | Ground hit velocity |
|------------------|--------|---------------------|---------------|---------------|-----------------------|----------------|-------------|---------------------|
| 19.8 m/s | 3305 m | 19.6 m/s | 18.4 s | 263 m/s | 32.4 m/s ² | 28.3 s | 243 s | 6.14 m/s |



DISCUSSÃO

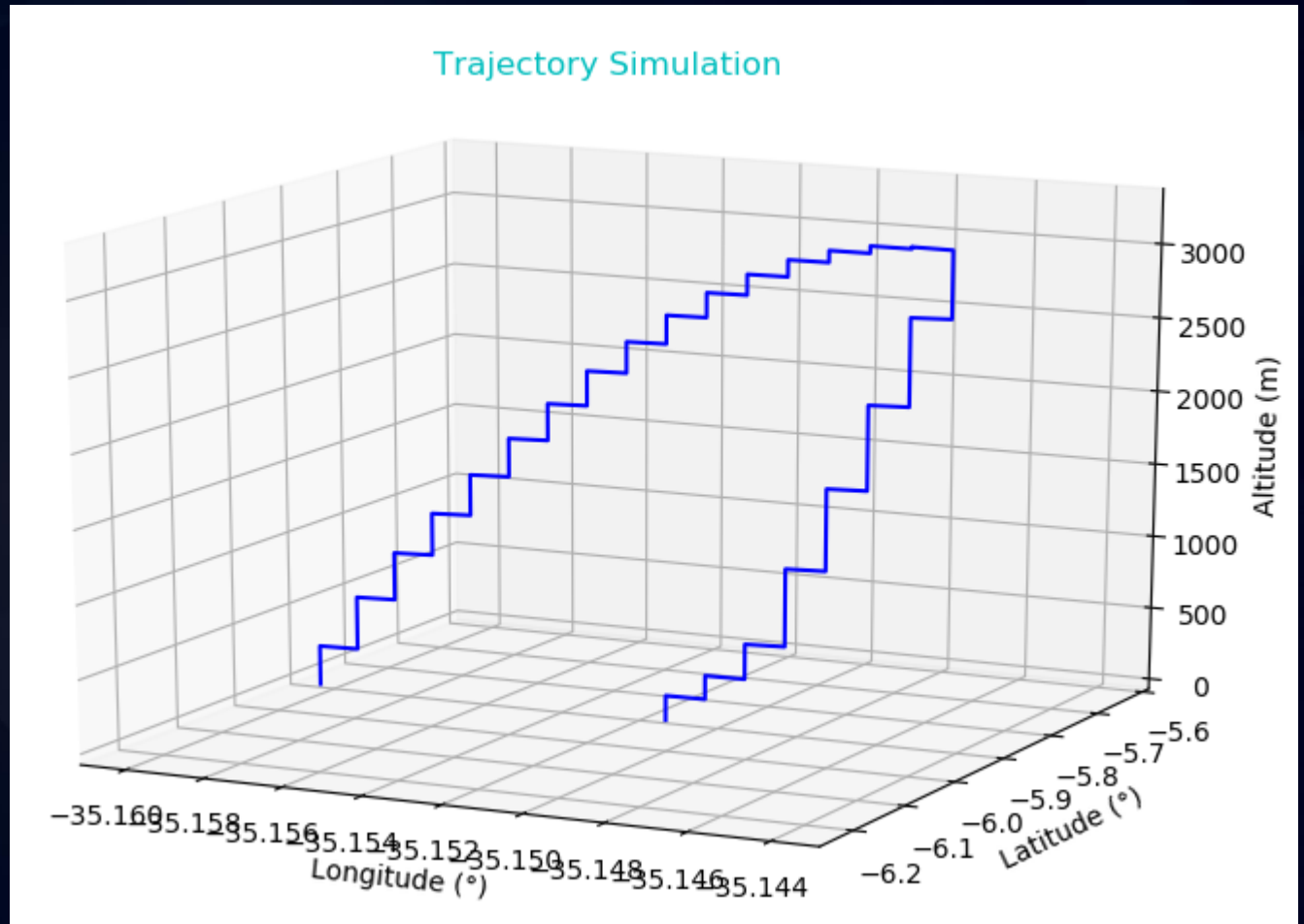
Simulação da Trajetória



DISCUSSÃO

Simulação da Trajetória

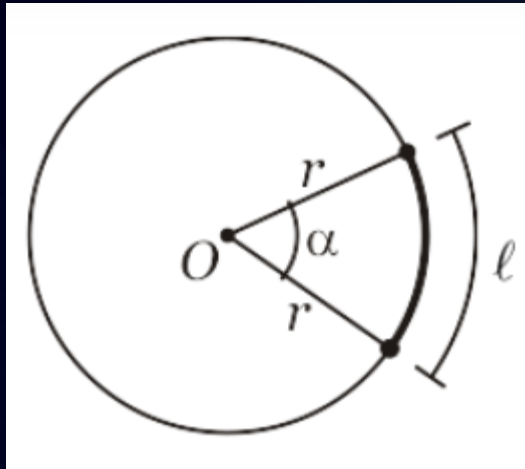
- Output de dados de trajetória completo.
- Só é possível colocar dados de entrada com até uma casa decimal.
- Deficiente no quesito de visualização da trajetória.
- Inconsistência nos dados de longitude e latitude.



DISCUSSÃO

Tratamento de Dados

- Com a obtenção de resultados insatisfatórios pela simulação, foi necessário corrigir os dados.
- Elaborou-se, então, um código em *python* para calcular as variações das coordenadas de longitude e latitude, considerando o planeta perfeitamente esférico.



$$\alpha = \frac{(180 \cdot \ell)}{\pi \cdot r}$$

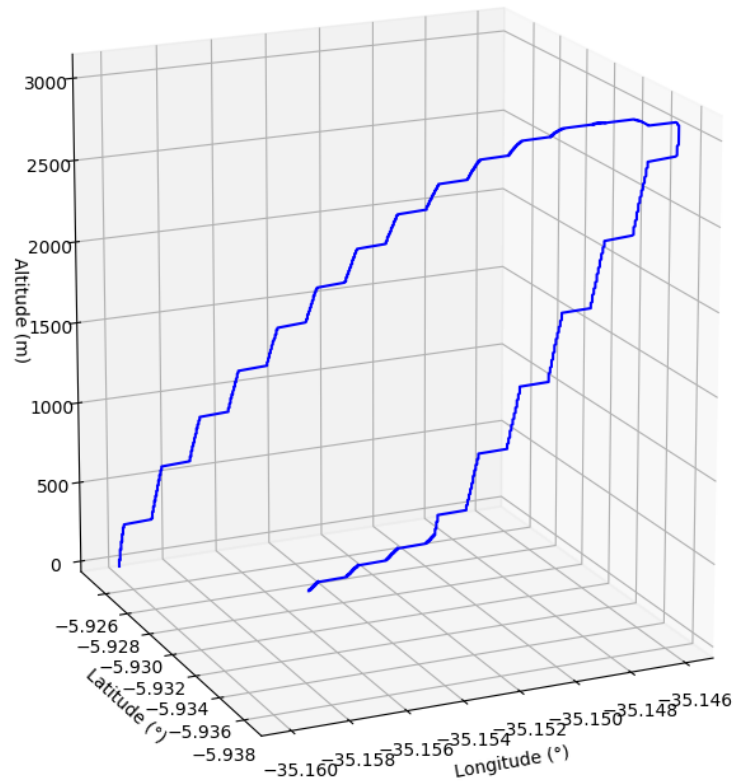
- Quando ℓ for a distância percorrida na direção Leste, α será o incremento da longitude.
- Quando ℓ for a distância percorrida na direção Norte, α será o incremento da latitude.



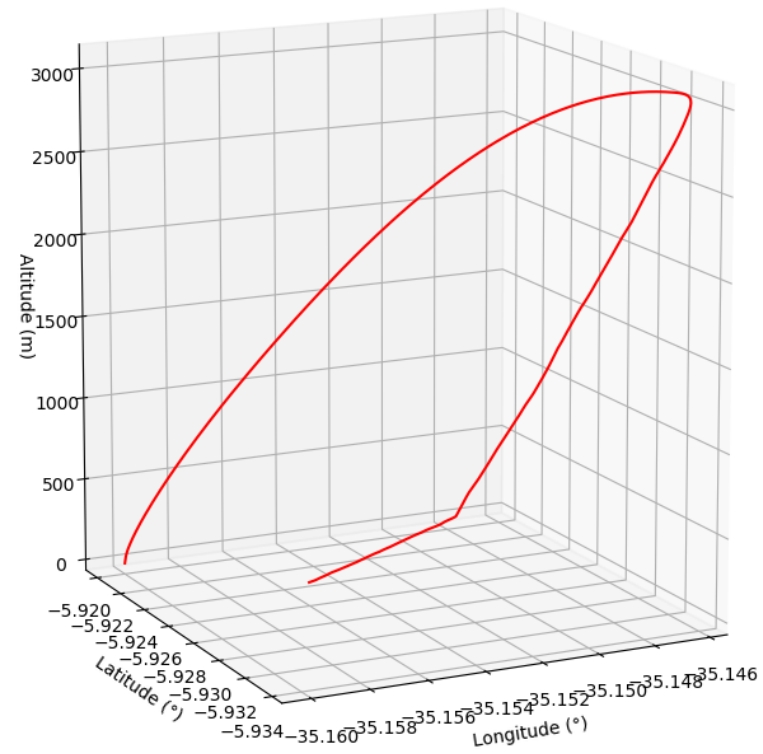
DISCUSSÃO

RESULTADOS ENCONTRADOS

Trajectory Simulation



Trajectory Simulation



DISCUSSÃO

RESULTADOS ENCONTRADOS

- Com os dados da trajetória encontrados e tratados, pode-se, então, plotar. No caso do *OpenRocket*, não é possível plotar a trajetória em 3D, bem como em um mapa 3D, como no Google Earth, o que é requisitado pela COBRUF.

➤ INSERIR IMAGEM DAS PLOTAGENS



DISCUSSÃO

RESULTADOS ENCONTRADOS

- Demonstrar resultados da DISPERSÃO DE IMPACTO



DISCUSSÃO

RESULTADOS ESPERADOS

- A estratégia adotada nesse trabalho de simulação completa da trajetória de foguetes de pequeno porte, incluindo a análise de dispersão de impacto, se mostrou consistente e coerente com a Teoria.
- No entanto, Vide acontecimentos recentes, a COBRUF 2019, que já havia sido adiada para 2020, teve os lançamentos cancelados e não foi possível comparar os resultados teóricos com os práticos. Espera-se então que a participação do GDAe em uma próxima edição ou em uma diferente competição possa trazer resultados para comparação com o que foi atingido nesse trabalho.
- Ainda, a imprecisão desses softwares livres, como é o caso do *OpenRocket*, e a falta de recursos que são importantes para o lançamento de foguetes, trazendo segurança e confiabilidade, mostra que novas soluções devem ser encontradas.
- Para tal, foi iniciado em 2019 um projeto de elaboração de um software para simulação da trajetória de foguetes, que hoje faz parte da bolsa PIBI 2020, em que espera-se a obtenção de resultados mais satisfatórios.



CONCLUSÃO

- A estratégia de simulação de trajetória de foguetes e análise de dispersão de impacto apresentada nesse trabalho se mostra coerente com a teoria, apresentando resultados melhores e mais completos do que simulações feitas somente com o uso do OpenRocket. Sendo assim, essa será a metodologia a metodologia utilizada



AGRADECIMENTOS

- GDAe – UFC
- Departamento de Física – UFC



REFERÊNCIAS

- Orbital Mechanics For Engineering Students
- Atmospheric and Space Flight Dynamics
- A model and sounding rocket simulation tool with mathematica

