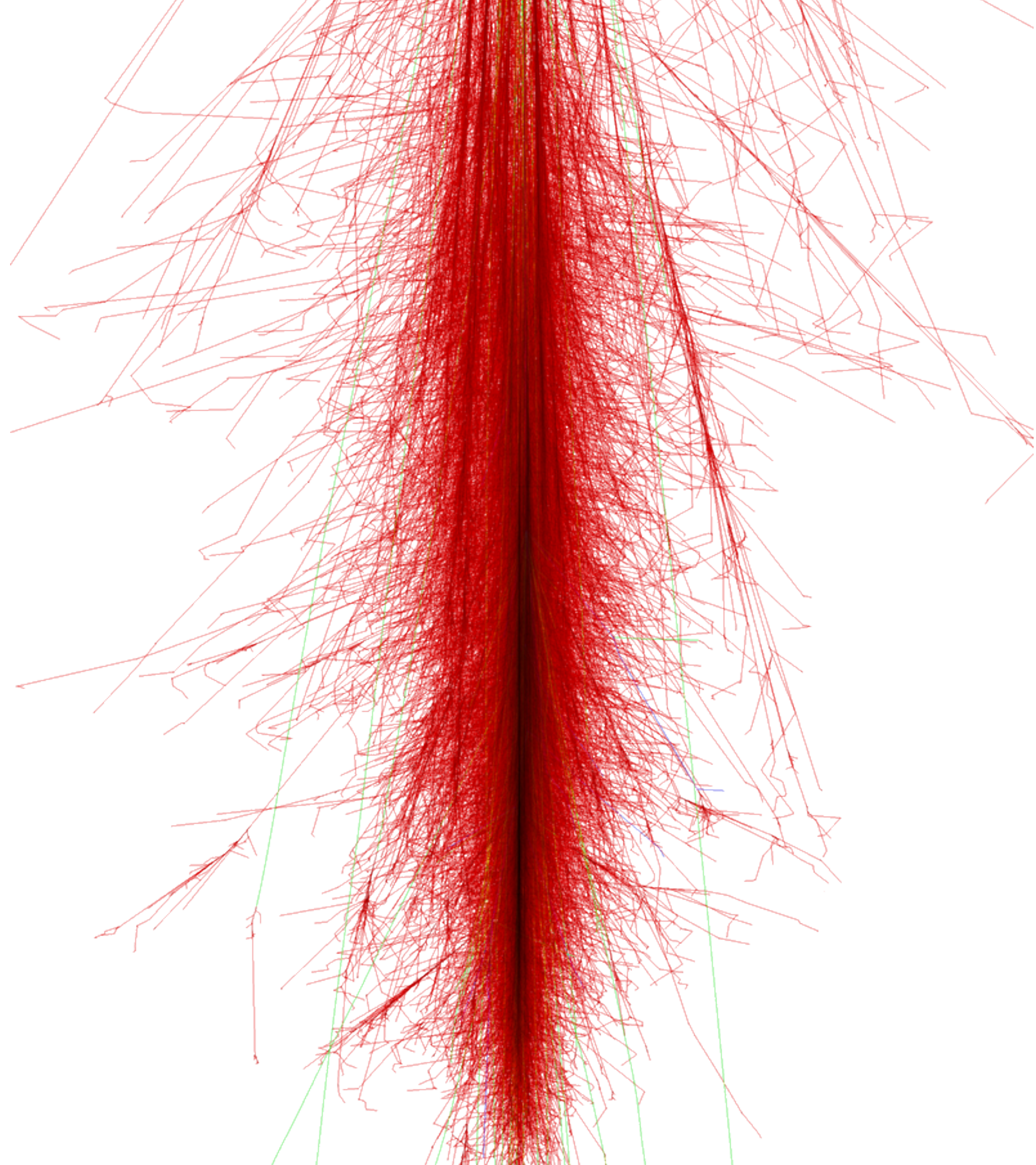


WG3: SIMULAÇÃO

**GUSTAVO GIL DA SILVEIRA
LEONARDO PEZZIN
MARCO LEITE**

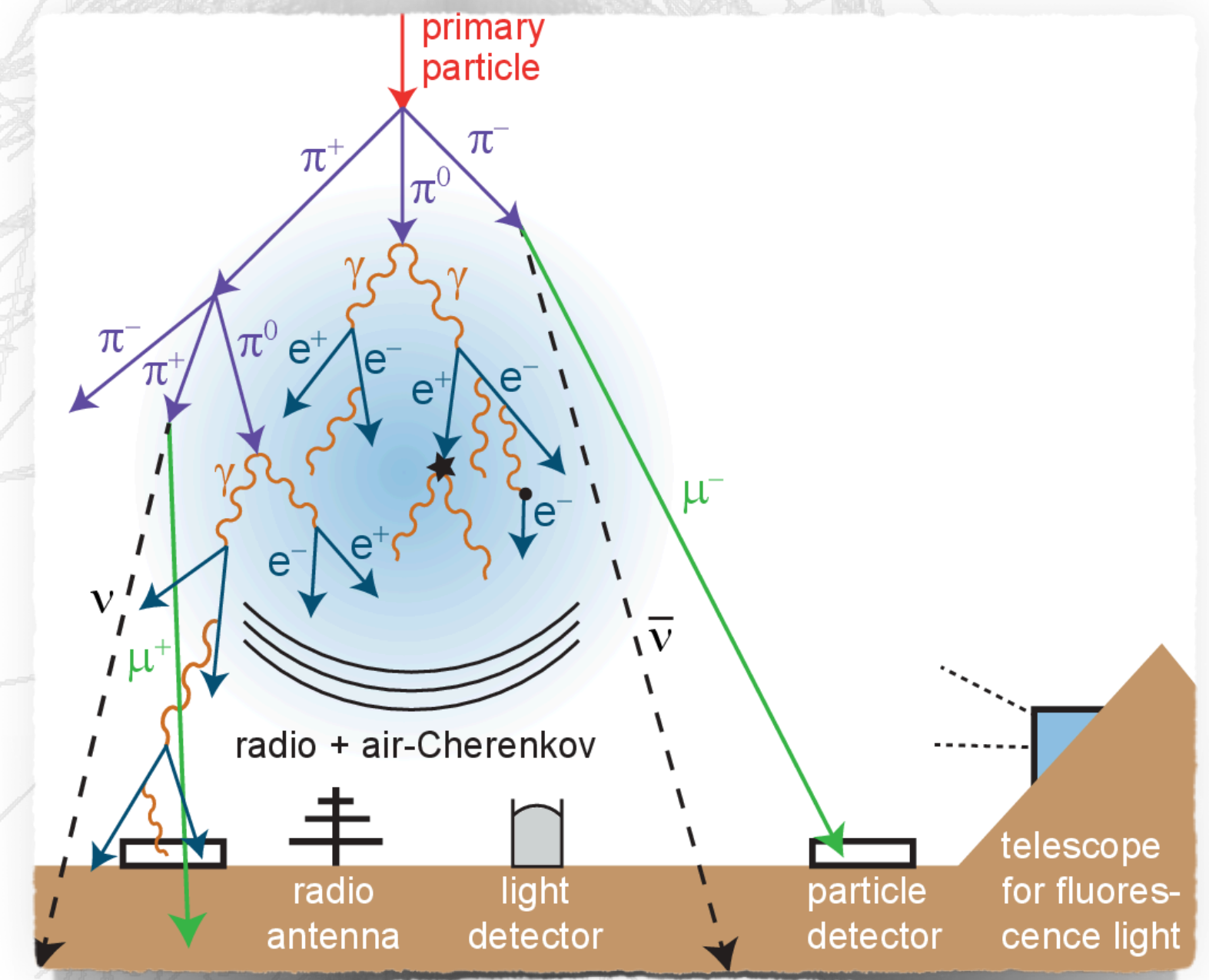
**PÁGINA DO GRUPO DE TRABALHO
(INEXISTENTE) — CULPA DO INDICO...**



PLANO

CRIAR INTERFACE INTERATIVA PARA INICIATIVA À LA MASTERCLASS

- Permitir a apresentação da Física subjacente aos raios cósmicos por meio de interface interativa e/ou vídeo;
- Introduzir conceitos básicos de cinemática;
- Disponibilizar uma ferramenta para análise dos dados:
 - seleção de partículas: tipo, carga, energia, etc.
 - seleção de partícula primária: próton, íon, Ferro.
 - separação dos tipos de partículas no chuveiro.



PLANO

INTERFACE SIMPLIFICADA PARA USO DO PROFESSOR

- Interface simples e intuitiva.
- Servidor local/remoto com aplicativos instalados para uso;
 - A depender dos aplicativos, um servidor remoto facilita a instalação e manutenção do *software*;
 - Número de sessões remotas pode afetar o desempenho da ferramenta.
- Linguagem a ser usada deve ser de fácil compreensão aos estudantes;
 - Vantagem em se usar python de forma global.



INTERFACE

COMO FAZER USO DAS FERRAMENTAS

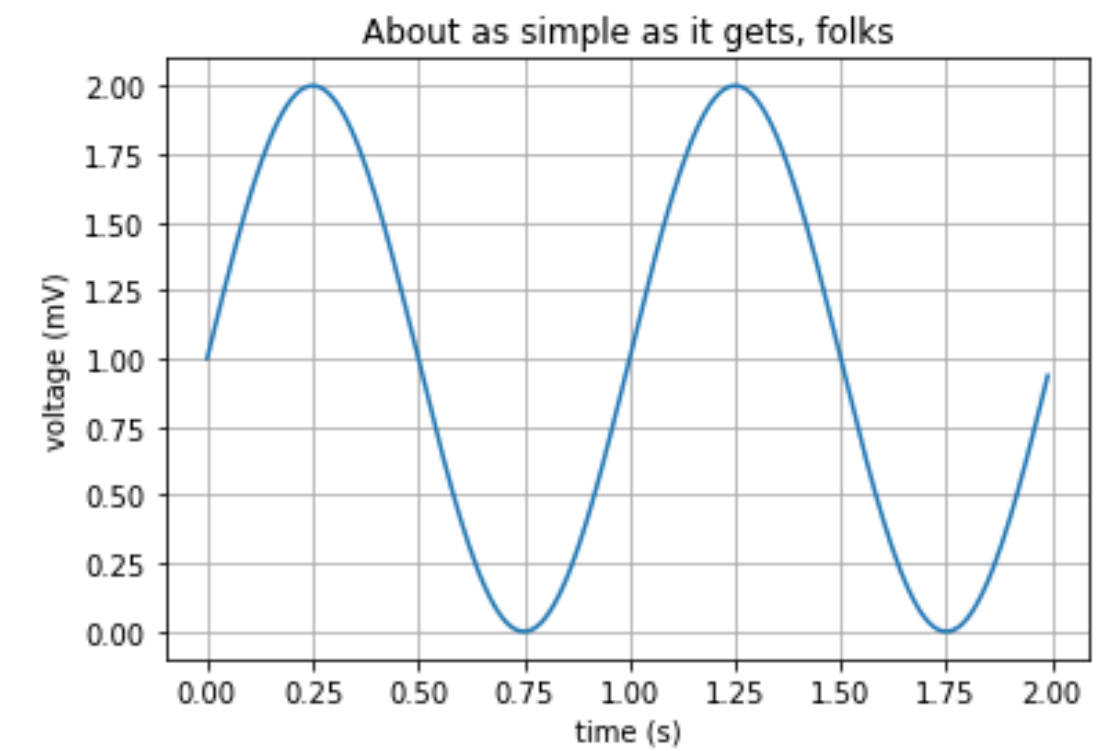
- A plataforma amigável para rodar as ferramentas de análise em python.
- Como tipicamente interfaces para usuário envolvem o uso do Terminal, uma alternativa em navegador de internet seria o mais intuitivo.
- O **Jupyter Notebook** se mostra a opção mais vantajosa a ser usada com estudantes visto a facilidade de uso sem exigir conhecimentos avançados de Windows ou Linux.
- Atualmente grande parte dos aplicativos utilizam ou possuem módulos de interface com o python, o que facilita a manutenção e atualização do servidor.





```
In [1]: print("Hello World")  
Hello World
```

```
In [2]: import matplotlib  
import matplotlib.pyplot as plt  
import numpy as np  
  
# Data for plotting  
t = np.arange(0.0, 2.0, 0.01)  
s = 1 + np.sin(2 * np.pi * t)  
  
fig, ax = plt.subplots()  
ax.plot(t, s)  
  
ax.set(xlabel='time (s)', ylabel='voltage (mV)',  
       title='About as simple as it gets, folks')  
ax.grid()  
  
fig.savefig("test.png")  
plt.show()
```



```
In [ ]: |
```

COMO GERAR RAIOS CÓSMICOS?

CORSIKA

A MONTE CARLO CODE TO SIMULATE EXTENSIVE AIR SHOWERS

- Geração de partículas de chuveis atmosféricos extensos.
- Possui física avançada para descrição das características do chuveiro.
- Permite exportar a informação de traços das partículas:
 - Posições inicial e final;
 - Tempos inicial e final;
 - Tamanho do arquivo: ~1GB 🤯
- Descreve os parâmetros para elétrons, múons, mésons e demais partículas.

CHIRITOI

CONVERSÃO DE BINÁRIO PARA TEXTO

- A informação dos traços do CORSIKA é fornecida em arquivo binário, pois é lida diretamente por rotina para construção de gráfico.
- Chiritoi et al (Romênia) construíram uma ferramenta para conversão dos dados e interface com o Blender (renderizador de animações).

3D VISUALIZATION FOR SIMULATED EXTENSIVE AIR SHOWERS (EAS) USING CORSIKA AND BLENDER 3D

G. CHIRIȚOI¹, A. JIPA², E. M. POPESCU¹

¹*Institute of Space Science, Atomiștilor 409, RO-077125, Bucharest-Măgurele, Romania E-mail: gabriel.chiritoi@spacescience.ro Email: empopescu@spacescience.ro*

²*Atomic and Nuclear Physics Department, Faculty of Physics, University of Bucharest, P.O.Box MG-11, RO-077125 Bucharest-Magurele, Romania, E-mail: jipa@brahms.fizica.unibuc.ro*

Blender* [~/home/bruno/bolsa raios cosmicos/SEGUNDA ETAPA/raiosCosmicos292.blend]

```
107 listofVectors = [(x,y,z),(xend,yend,zend)]
108
109 def MakePolyLine(objname, curvename, cList):
110     curvedata = bpy.data.curves.new(name=curvename, type='CURVE')
111     curvedata.dimensions = '3D'
112     curvedata.resolution_u = 2
113     curvedata.bevel_object = ob
114
115     objectdata = bpy.data.objects.new(objname, curvedata)
116     objectdata.location = (0,0,0) #object origin
117     bpy.context.scene.objects.link(objectdata)
118
119     polyline = curvedata.splines.new('NURBS')
120     polyline.points.add(len(cList)-1)
121     for num in range(len(cList)):
122         polyline.points[num].co = (cList[num])+w,)
123
124     polyline.order_u = len(polyline.points)-1
125     polyline.use_endpoint_u = True
126
127     r=0.04
128     g=0.79
129     b=0.17
130     color=(r,g,b)
131     mat = createNewMaterial("myNewMaterial",color)
```

File: *//corsikablender/codearrumado.py (unsaved)

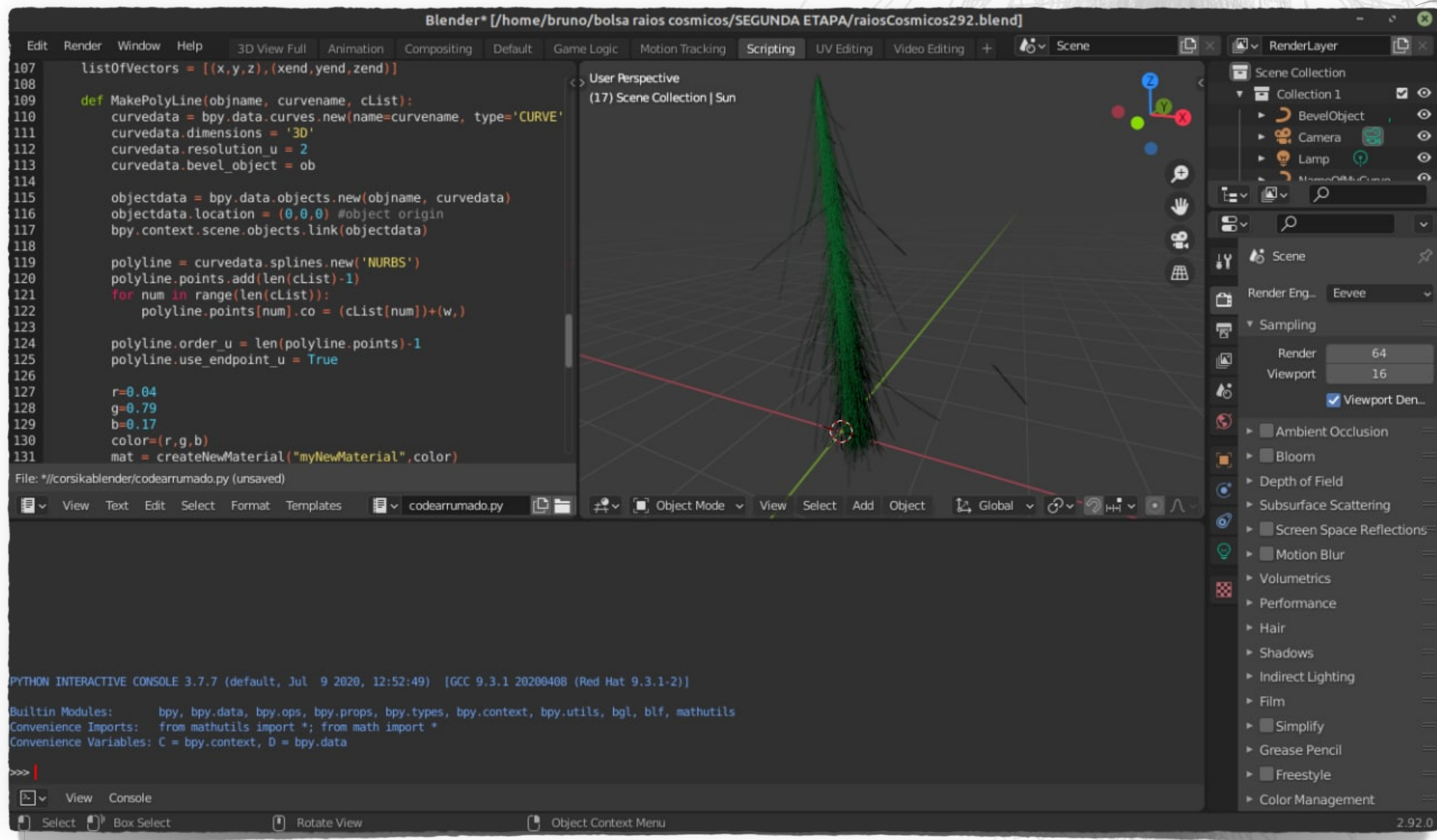
View Text Edit Select Format Templates codearrumado.py Object Mode View Select Add Object Global

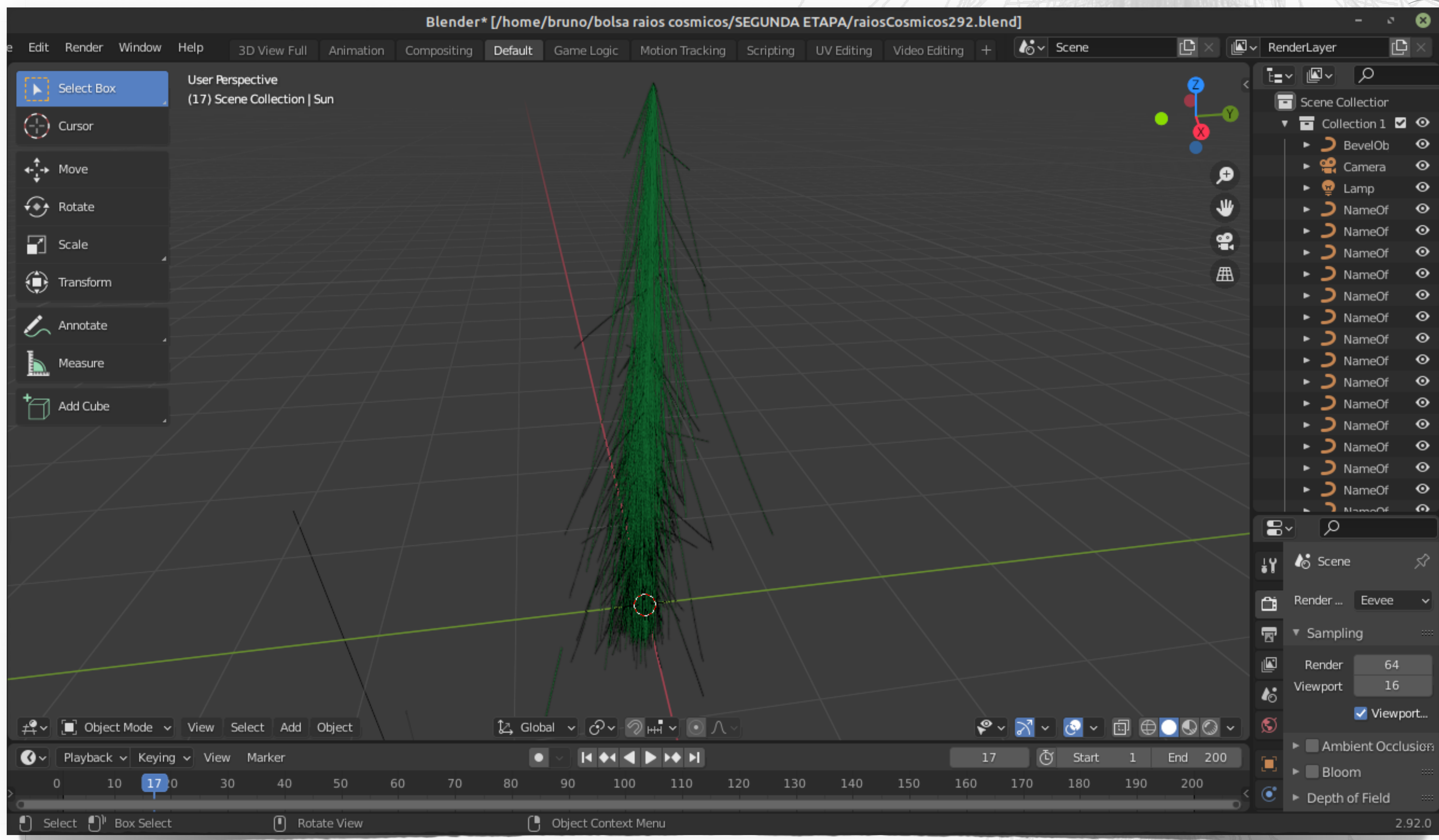
PYTHON INTERACTIVE CONSOLE 3.7.7 (default, Jul 9 2020, 12:52:49) [GCC 9.3.1 20200408 (Red Hat 9.3.1-2)]

Builtin Modules: bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data

>>> |

View Console Select Box Select Rotate View Object Context Menu 2.92.0

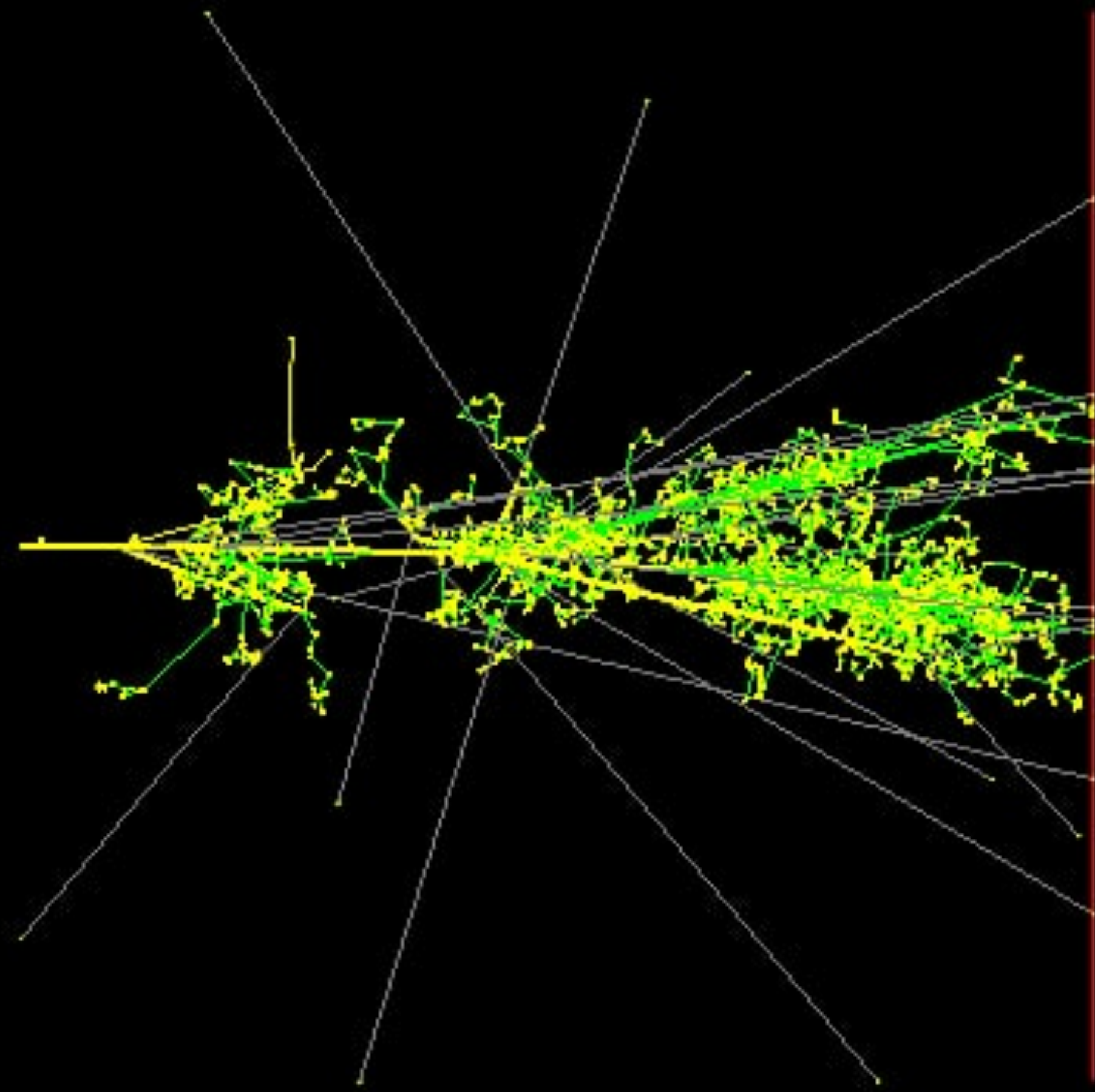


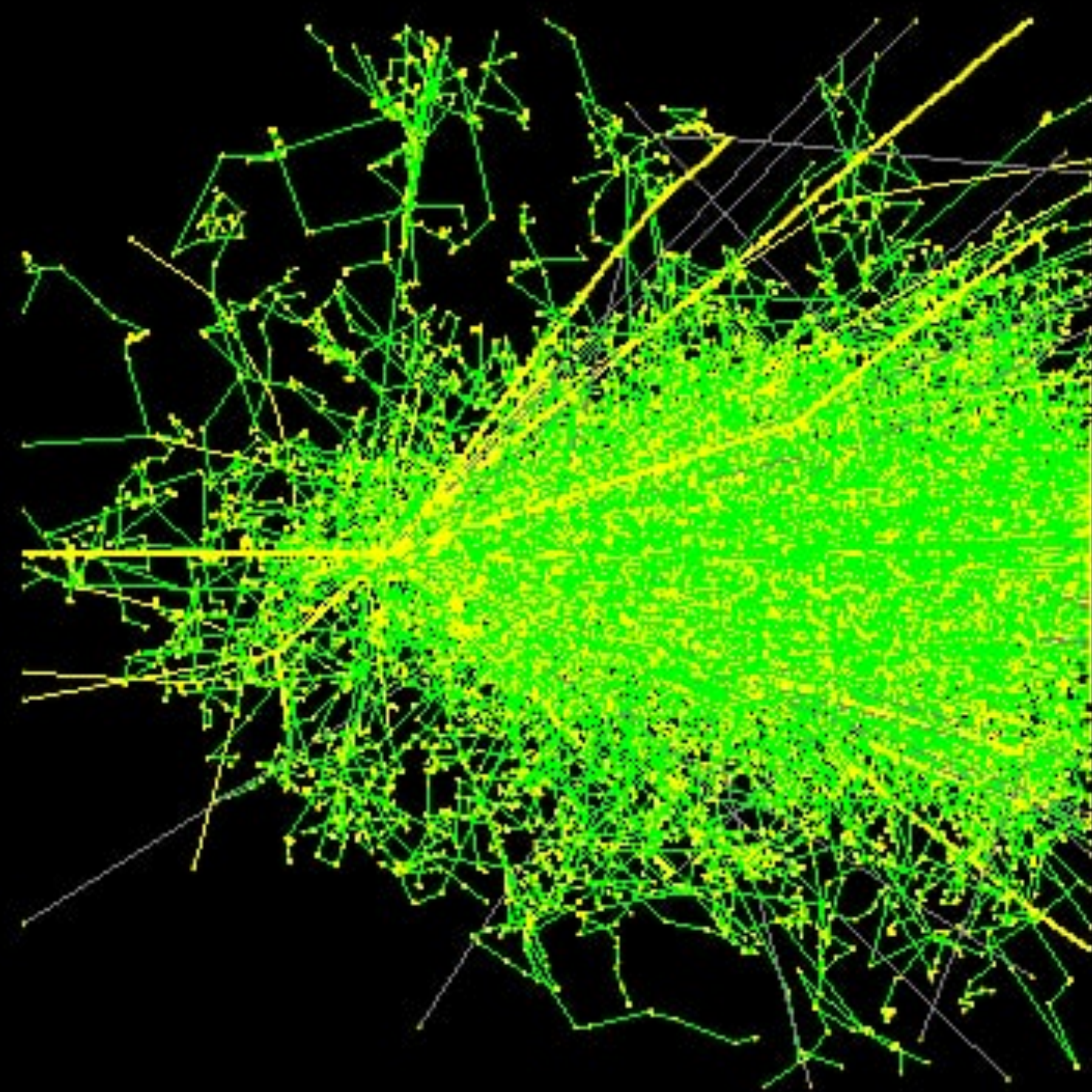


GEANT4

TOOLKIT PARA DESCRIÇÃO DE DETECTORES

- Grande flexibilidade em introduzir a física de interações e construção de sólidos para a modelagem de detectores.
- Interface moderna e de fácil acesso, com possibilidade de uso no Terminal.
- Possui módulo em python para ser acessado via jupyter.
 - Falta averiguar a performance em gerar evento para visualização rápida.
 - Podemos construir arquivos prontos para análise das partículas para agilizar o processo.





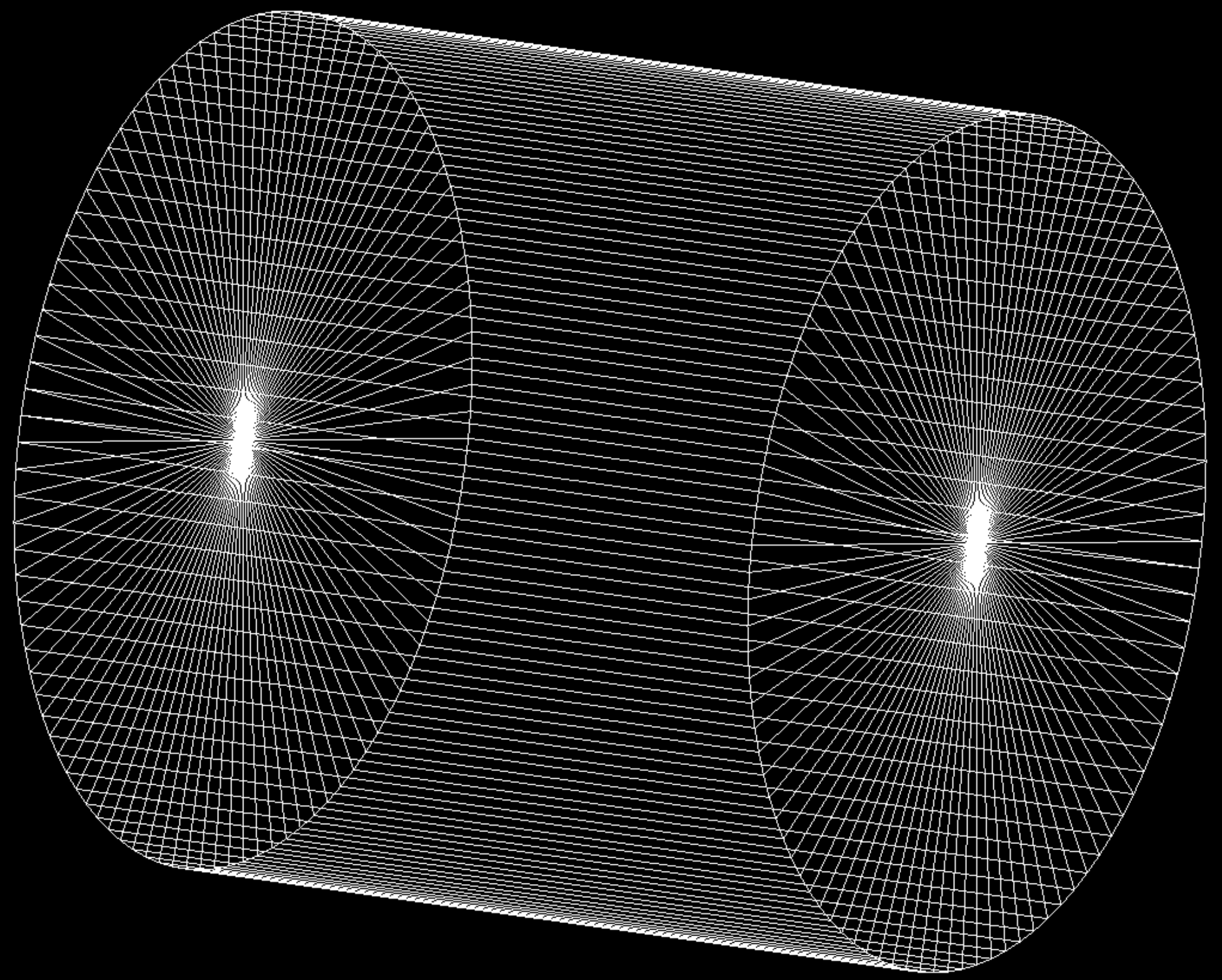
Network visualization



Search :

Command

- ▶ control
- ▶ units
- ▶ profiler
- ▶ gui
- ▶ geometry
- ▶ tracking
- ▶ particle
- ▶ event
- ▶ cuts
- ▶ run
- ▶ random
- ▶ process
- ▶ material
- ▶ physics_lists
- ▶ vis
- ▶ hits
- ▶ heptst
- ▶ physics_engine
- ▶ gun



FLUKA

PACOTE DE SIMULAÇÃO DE FÍSICA DE PARTÍCULAS - CERN

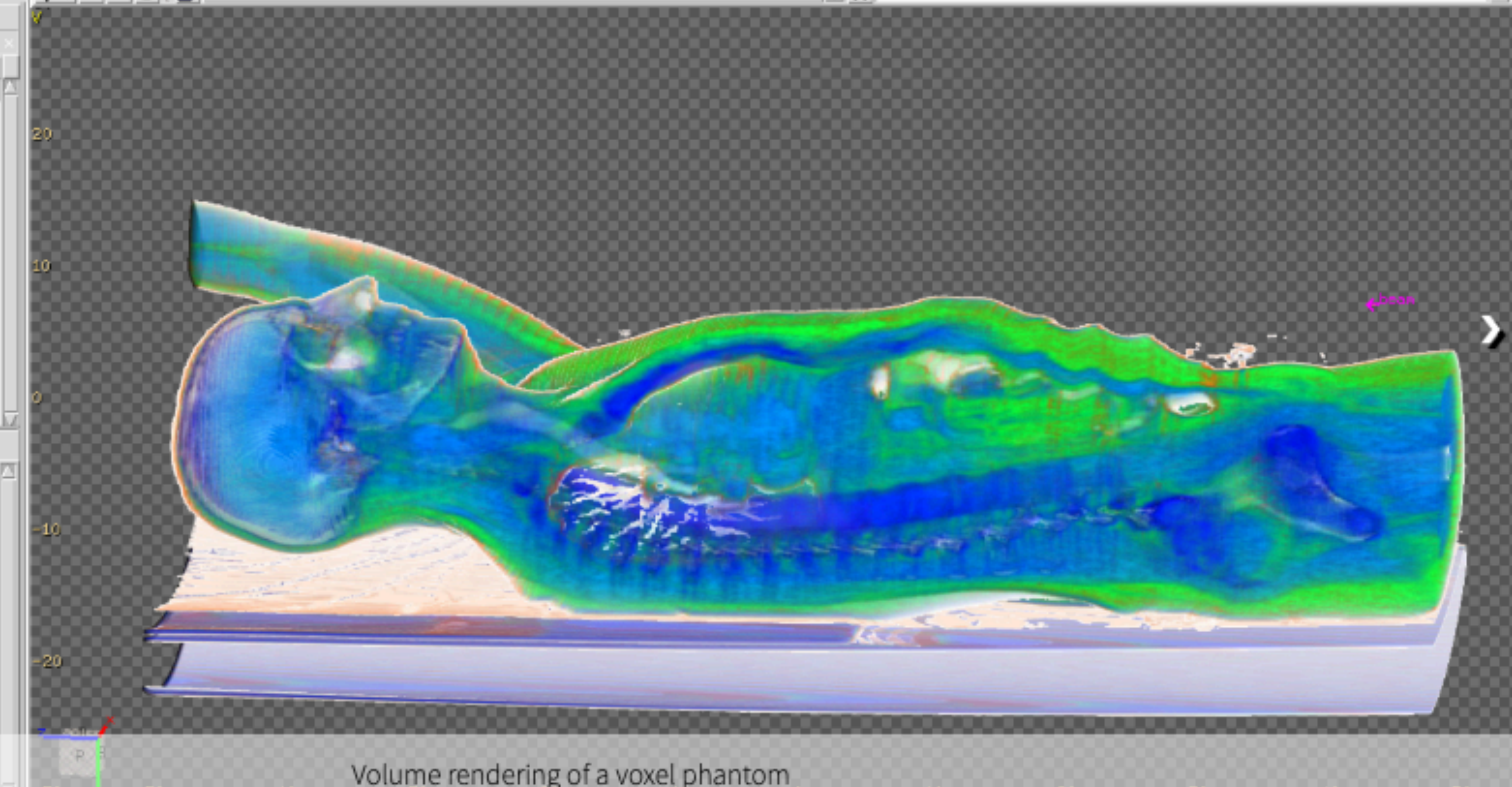
- Usado para estudo de radiação.
- Ferramenta muito completa, porém de difícil interface.
 - Módulos antigos (1998), baseados em Fortran77 e difíceis de integrar.
- Possui interface gráfica com o FLAIR.
- Possui uma boa descrição da atmosfera que pode ser transportada para GEANT4.

Geometry

Type	Name
BEAM	

Properties Attributes

Name visible



Volume rendering of a voxel phantom

PROPOSTA

SOLUÇÃO COMPLETA E MAIS INTUITIVA

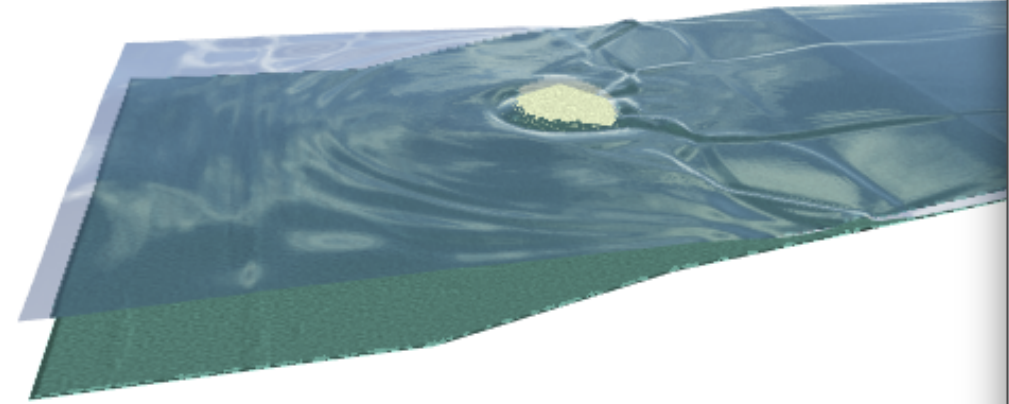
- Usar o GEANT4 como ferramenta de geração dos raios cósmicos.
- Exportar os dados para o VTK e usar na interface de **jupyter** com **ipygani**.
- Vantagem de fácil manutenção e melhor visualização da física.
- Precisa de um servidor para uso nas análises e que garanta estabilidade no volume de uso por diferentes instâncias.

JupyterLab interface showing a code editor with the following Python code:

```
water = Water(
    actual_water,
    under_water_blocks=(UnderWater(floor), ),
    caustics_enabled=True,
)

scene = Scene((water, ))

scene
```



JupyterLab interface showing a code editor with the following Python code:

```
# Turn the PyVista mesh into a PolyMesh
mesh = PolyMesh.from_vtk(ugrid)

colored_mesh = IsoColor(mesh, min=-10421.0, max=6527.0)
warped_mesh = WarpByScalar(colored_mesh, input='altitude', factor=0.5e-5)

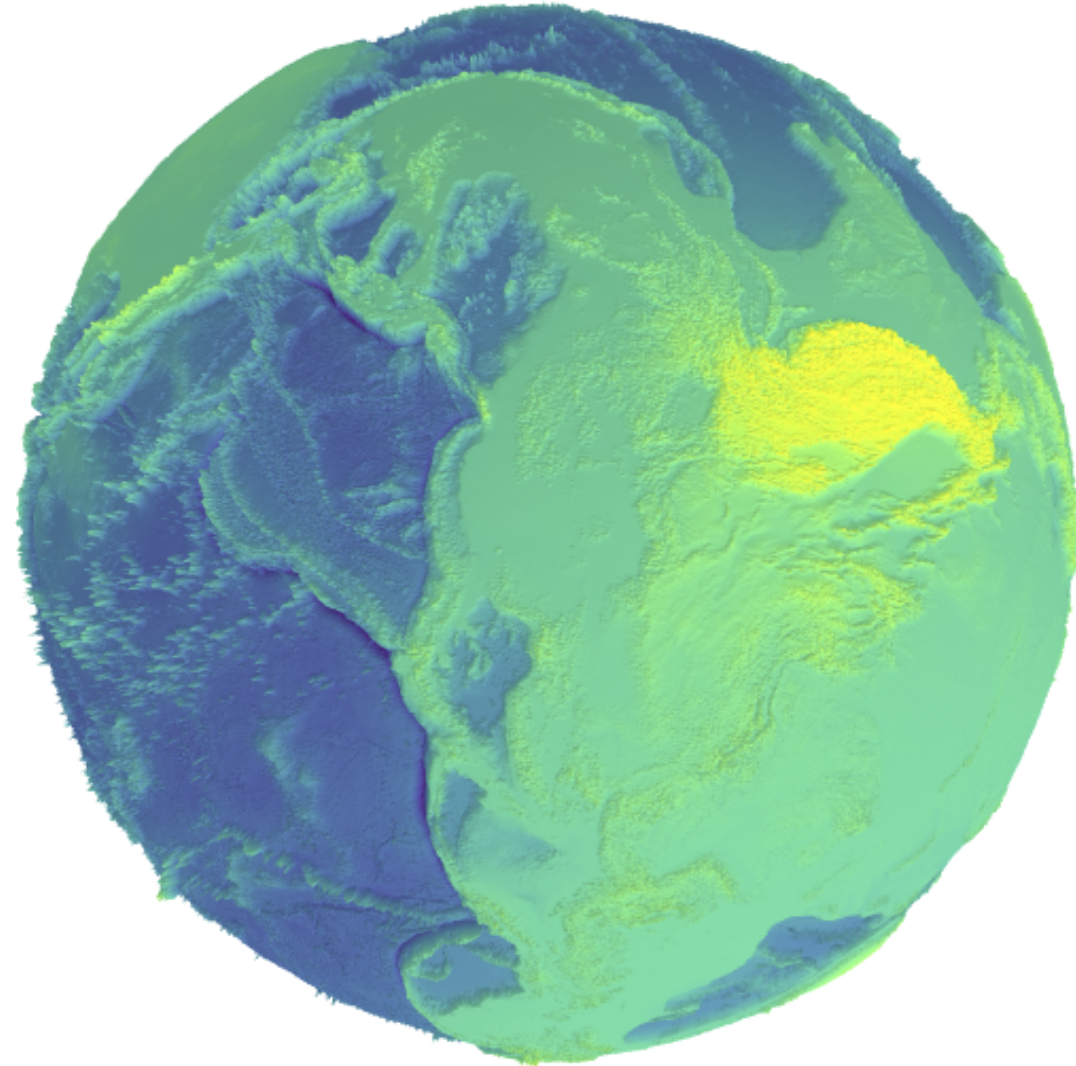
# Link a slider to the warp value
warp_slider = FloatSlider(min=0., max=5., value=0.5)

def on_slider_change(change):
    warped_mesh.factor = change['new'] * 1e-5

warp_slider.observe(on_slider_change, 'value')

VBox((warp_slider, Scene([warped_mesh], layout={'height': '600px'})))
```

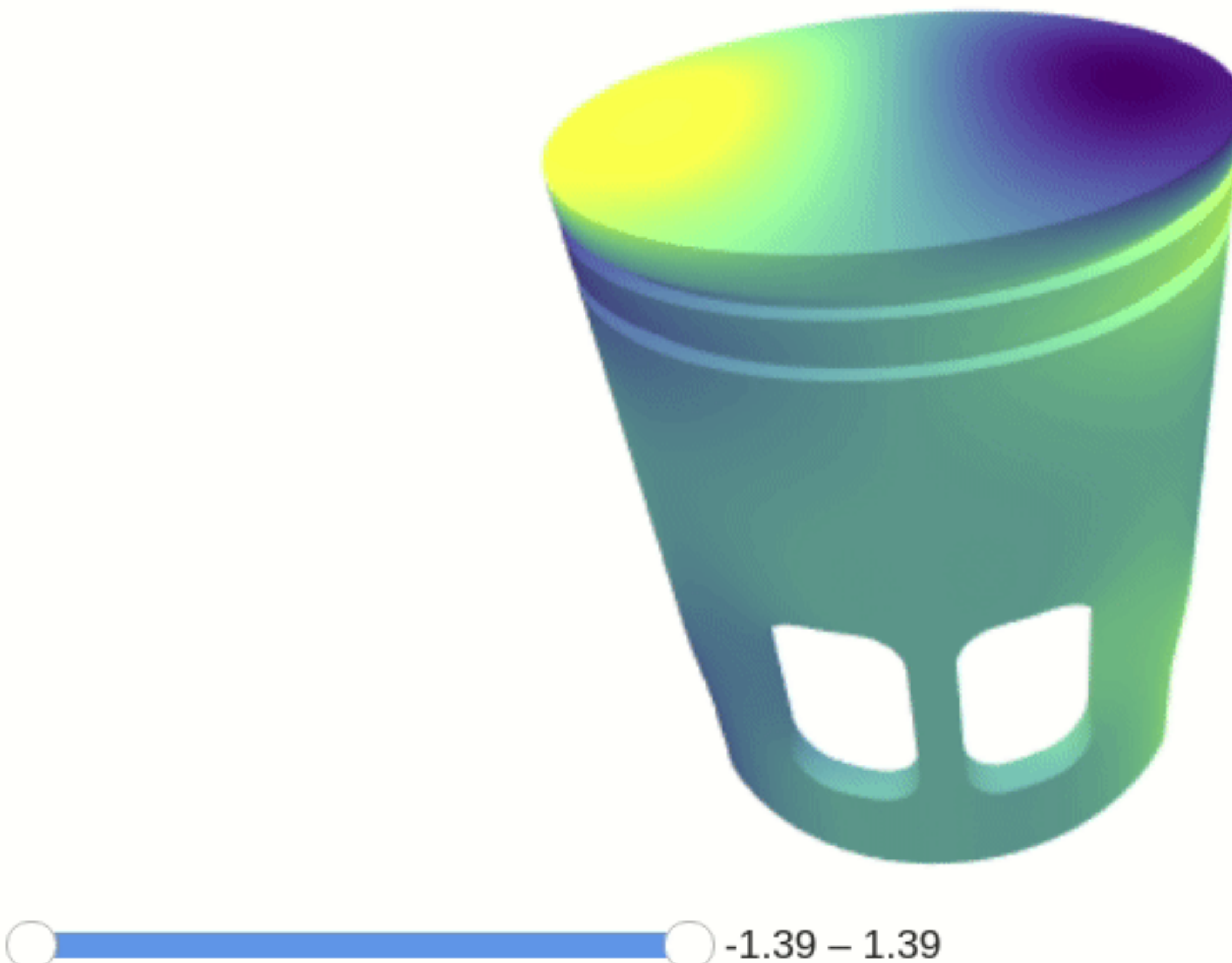
Below the code is a slider control set to 0.50.



File Edit View Run Kernel Tabs Settings Help

Python 3

```
[11]: VBox((scene2, range_slider))
```



ipygany

AGRADECIMENTOS

- Bruno Zanette, Marcos Derós Elisa Garcia (UFRGS) pelo auxílio na geração dos dados com CORSIKA, GEANT4 e Blender.
- Marco pelas rotinas de GEANT4 e Fluka para testes (que ainda estão ocorrendo).