

## PENELOPE: INTERFACE EM PYTHON PARA AVANÇOS NAS SIMULAÇÕES DE RADIOTERAPIA

Luana Rodrigues dos Santos<sup>1</sup>; Everaldo Arashiro<sup>2</sup>; Suzielli Martins Mendonça<sup>3</sup>

<sup>1</sup>FURG: Universidade Federal do Rio Grande – driggres@gmail.com

<sup>2</sup>FURG: Universidade Federal do Rio Grande – earashiro@furg.br

<sup>3</sup>FURG: Universidade Federal do Rio Grande – suzielli@furg.br

### 1. INTRODUÇÃO

A física médica abrange múltiplas áreas de interesse, radioterapia sendo uma delas, é uma etapa importante no combate ao câncer, especialmente no cálculo preciso da distribuição de dose, cuja imprecisão pode comprometer a eficácia do tratamento e causar danos aos tecidos sadios, o que é de extrema preocupação. Nesse contexto, e considerando que a incidência global de câncer tem aumentado significativamente nas últimas décadas, como reporta SUNG *et al.*, (2021), é imprescindível que físicos médicos continuem contribuindo para lidar com desafios atuais da radioterapia, explorando tecnologias emergentes para aprimorar modelos preditivos voltados à radioterapia (FIORINO *et al.*, 2020).

Diante desse cenário, o sistema de códigos PENELOPE (*Penetration and Energy Loss of Positrons and Electrons*) surge como uma ferramenta de interesse, pois contém em sua estrutura de código cálculos do transporte de partículas de baixa energia, úteis para a radioterapia e fundamentados em uma base física sólida, o que possibilita modelagens realistas para o planejamento do tratamento radioterápico (PENELOPE-2014 (Bank, 2015)).

Neste trabalho, propõe-se o uso de linguagens modernas como o Python para ampliar a utilidade dos cálculos avançados do PENELOPE, a fim de interligá-los a lógicas de bibliotecas que permitam a visualização do caminho da radiação no paciente, seja por meio de geometrias fornecidas ou criadas por nós ou por outros interessados em simular a radiação em geometrias próprias.

### 2. METODOLOGIA

O código PENELOPE foi fornecido por um de seus autores e, como passo fundamental para o desenvolvimento do projeto, foi realizada uma separação das sub-rotinas que compõem o código-fonte principal *penelope.f*, que é estruturado em quatro blocos principais de subprogramas: Cálculos preparatórios, Rotinas de I/O (*Input/Output*), Simulação de interações, e Rotinas numéricas de transporte.

O sistema utiliza o método Monte Carlo, o que viabiliza simulações rápidas de transporte de radiação e cálculo de dose em radioterapia (JIA *et al.*, 2011).

A base de dados do PENELOPE contém informações físicas essenciais sobre os materiais, tais como composições químicas, densidade e seções de choque, mais detalhes sobre a estrutura dessa base de dados podem ser encontrados no documento PENELOPE-2014 (Bank, 2015). Esses dados são essenciais para simular diversas situações que envolvam a interação de partículas com a matéria.

A organização da pasta do projeto, ilustrada no Fluxograma da Figura 1, é composta por:

1. Dois arquivos principais: **main.py**, que funciona como controlador central da aplicação, e **int.ui**, que contém a interface gráfica criada no *QtCreator*;
2. Um modelo geométrico, **geo.obj**, que representa a geometria de um ser humano, obtido de sites que fornecem gratuitamente esses arquivos;
3. Duas pastas: uma com o conteúdo original do PENELOPE e outra contendo algumas sub-rotinas separadas.

A pasta sub-rotinas abriga rotinas extensas que não serão reescritas em Python, mas sim chamadas pelo **main.py** por meio de bibliotecas facilitadoras como o *f2py* (PETERSON, 2009), com objetivo de manter a velocidade dos códigos Fortran, mas permitir interação com as bibliotecas Python. O arquivo **geo.obj** será a geometria na qual faremos os testes por meio da biblioteca VEDO, para verificar se a interface está ou não funcionando na simulação da interação da radiação com a geometria.

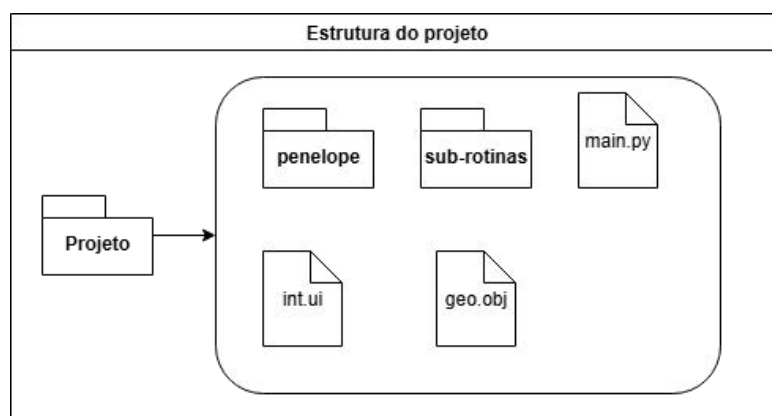


Figura 1 - Fluxograma da estrutura de pastas do projeto. (Fonte: do autor)

A biblioteca VEDO é especializada em visualização científica e análise de dados tridimensionais, como exemplo a Figura 2, pode-se notar que a utilização dessa biblioteca serve para melhorar a visualização atual do PENELOPE, pois possibilita uma melhor exploração de ângulos e sistematizar um método de corte que facilite a visualização de curvas de isodose e segmentação de partes anatomicas.

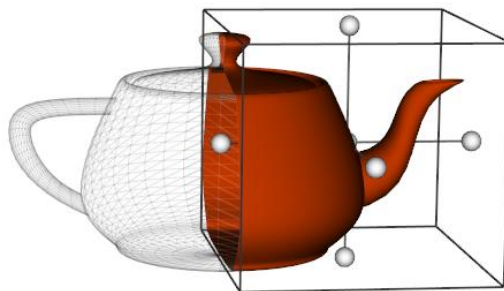


Figura 2 - Exemplo de um modelo 3D sendo visualizado utilizando a biblioteca VEDO (Fonte: *API documentation*, VEDO).

### 3. RESULTADOS E DISCUSSÃO

Atualmente, possuímos a leitura funcional das seções de choque e das informações dos materiais contidos no banco de dados, sendo possível utilizando o Matplotlib, permitindo a criação de gráficos de alta qualidade para análise científica da interação de fótons, elétrons e pósitrons no material selecionado (HUNTER, 2007). A partícula selecionada para a plotagem do gráfico será a mesma escolhida para a simulação da interação na geometria. Esses recursos estão ilustrados na Figura 3.

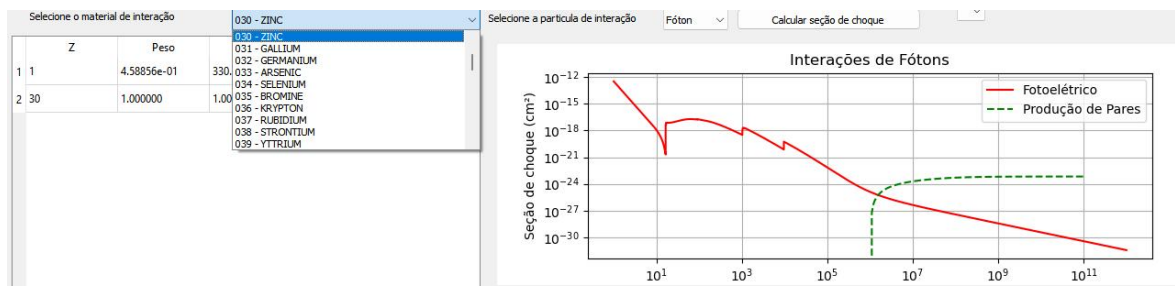


Figura 3 - Gráfico das seções de choque e interações das partículas no material selecionado exibido na interface. (Fonte: do autor)

Essa etapa do desenvolvimento da interface é a base de todo o processo, pois, quando a simulação da interação ocorrer, será necessário garantir que as seções de choque estejam corretamente carregadas para o material selecionado. Caso contrário, o método Monte Carlo poderá, de forma incorreta, aumentar a probabilidade de uma interação em um intervalo de energia que não corresponda à realidade física.

Dentro da interface também é possível o carregamento das geometrias propriamente ditas, essas geometrias serão utilizadas para os cálculos e para visualização das simulações. O usuário pode selecionar a área de interesse para realizar a interação e deve posteriormente selecionar o material ao qual a geometria será formada por meio do método *drag and drop*, em tradução direta: arrastar e soltar. Esses recursos estão ilustrado na Figura 4.

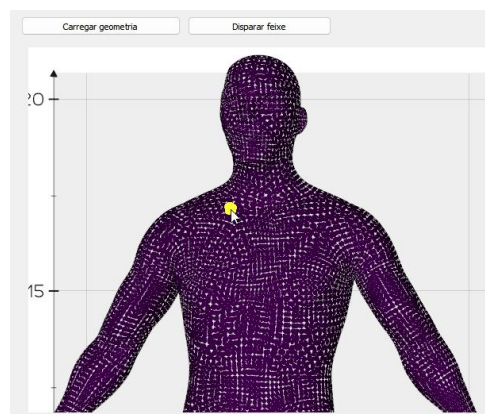


Figura 4 - Modelo **geo.obj** carregado na interface, em amarelo o ponto selecionado para a interação ocorrer. (Fonte: do autor)

#### **4. CONCLUSÕES**

O presente trabalho encontra-se em uma etapa inicial de desenvolvimento, mas os resultados obtidos até o momento já demonstram avanços significativos na integração dos dados. Essa integração estabelece uma base sólida para a incorporação de funcionalidades mais avançadas.

Nesta etapa, as principais conquistas incluem a implementação eficiente da leitura das seções de choque e das propriedades dos materiais, a criação de um recurso de visualização gráfica interativa dessas informações e o desenvolvimento de um ambiente de simulação que integra, de forma consistente, a escolha do material à geometria utilizada.

Como limitação atual, ressalta-se que a simulação ainda não contempla segmentações específicas dos volumes anatômicos, podendo apenas atribuir materiais a geometrias separadas, mas não separando elas em si.

Para as próximas etapas, está prevista a utilização da biblioteca VEDO para realizar a segmentação da geometria, permitindo a atribuição precisa de propriedades físicas a cada região anatômica. Com isso, será possível calcular interações e curvas de isodose considerando as seções de choque correspondentes a cada tipo de tecido.

#### **5. REFERÊNCIAS BIBLIOGRÁFICAS**

NUCLEAR ENERGY AGENCY DATA BANK. PENELOPE 2014: Code System for Monte Carlo Simulation of Electron and Photon Transport. Issy-les-Moulineaux, France: Nuclear Energy Agency Data Bank, 2015.

FIORINO, C. et al. Grand challenges for medical physics in radiation oncology. *Radiotherapy and Oncology*, v.153, p.7-14, 2020.

SUNG, H. et al. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, v.71, n.3, p.209-249, 2021.

PETERSON, P. F2PY: a tool for connecting Fortran and Python programs. *International Journal of Computational Science and Engineering*, v.4, n.4, p.296-305, 2009.

HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, v.9, n.3, p.90-95, 2007.

JIA, X. et al. GPU-based fast Monte Carlo simulation for radiotherapy dose calculation. *Physics in Medicine & Biology*, v.56, n.22, p.7017, 2011.