



Interação e Detecção de Partículas

Projeto Final - Terapia de Protões

Universidade de Coimbra, 17 de Janeiro de 2022

Teórica Prática, Turma PL1

Prof. Doutor Alexandre Lindote

José Nuno da Cruz Faria - 2015252736

1) Introdução

A passagem de partículas carregadas pela matéria pode ser descrita através de dois fenómenos: perda de energia da partícula, e desvio da trajetória da partícula. Existem diversos processos físicos que causam estes efeitos (Bremsstrahlung, reações nucleares, emissão de radiação de Cherenkov), mas é principalmente causado por [1]:

- colisões inelásticas com os eletrões atómicos do material do alvo;
- *scattering* elástico com o núcleo.

A perda energética de uma partícula a atravessar um dado material é descrito pelo seu potencial de travagem (dE/dx). Figura 1 mostra a variação do potencial de travagem para diferentes partículas.

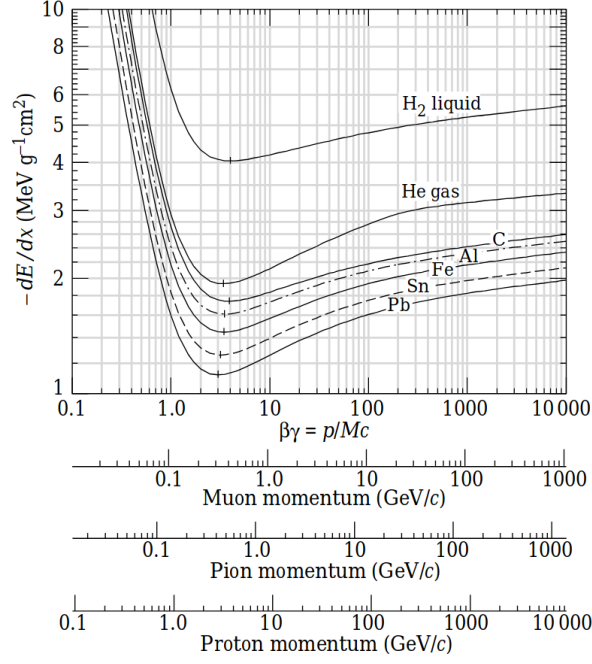


Figura 1: Variação de $-dE/dx$ em função de $\beta\gamma$ para diferentes partículas.

Para partículas pesadas carregadas (*e.g.* prótons, iões), o potencial de travagem pode ser descrito pela equação de Bethe-Block (sem correções de densidade e de camada), para $0.1 < \beta\gamma < 1000$:

$$-\frac{dE}{dx} = K\rho \frac{Z_{alvo}}{A_{alvo}} \frac{z^2}{\beta^2} \left[\ln \left(\frac{2m_e \gamma^2 \nu^2 W_{max}}{I^2} \right) - 2\beta^2 \right] \quad (1)$$

onde $K = 2\pi N_a r_e^2 m_e c^2 = 0.1535 \text{ MeV cm}^2/\text{g}$, ρ é a densidade do material do alvo, Z_{alvo} e A_{alvo} são o número atómico e massa atómica respetivamente, z é a carga da partícula incidente (em unidades de e), β é a velocidade relativa da partícula ($\beta = v/c$), γ é o fator de Lorentz, I é o potencial de excitação médio, e W_{max} é a energia máxima que pode ser transferida numa única colisão.

Podemos observar que o potencial de travagem que para energias não relativistas, a expressão é principalmente dominada pelo termo $1/\beta^2$, logo aumenta com a diminuição da velocidade. Isto significa que à medida que a partícula atravessa a materia, ela abranda devido às interações eletromagnéticas, que por sua vez aumenta o potencial de travagem, portanto deveremos observar a maioria da sua energia depositada no final da trajetória da partícula. Este efeito é chamado de pico de Bragg, e é extremamente utilizado em aplicações médicas onde pretendemos depositar uma alta dose de radiação em tumores com o mínimo de destruição do tecido envolvente.

Neste trabalho iremos efetuar uma simulação da interação de feixe de prótons com tecido mole, com o objetivo de determinar a energia necessária para o feixe depositar a energia num tumor que encontra-se 7 cm em profundidade da superfície do tecido. Foram efetuadas no total 11 simulações, uma por cada energia média diferente do feixe (de 50 a 150 MeV, em passos de 10 MeV).

2) Simulador

O código para o simulador de Monte-Carlo foi criado em Geant4 com base no *template* fornecido¹, e encontra-se no seguinte URL: https://github.com/jncfa/idp_exercices/tree/master/final_project.

Para criar o simulador é necessário adicionar 3 componentes ao *template*:

1. Geometria da Simulação: Definir a posição e orientação de cada componente a simular, *i.e.* posição e orientação do alvo de “tecido mole”.
2. Gerador da Partículas Primárias: Definir quais são as partículas emitidas e as suas propriedades (*i.e.* energia, posição, e orientação de cada partícula).
3. Recolha de informação para análise: Definir um mecanismo que permita a recolha dos dados da simulação.

Fora estas componentes, foi também necessário alterar a `PhysicsList` que define as interações físicas que serão simuladas, tendo sido usado a `PhysicsList` que vinha previamente configurada no *template*, para ter um corte de 0.01 mm.

Nas próximas secções, é detalhada a implementação de cada componente.

2).1 Geometria da Simulação

Para definir a geometria da simulação, utilizámos a classe `G4VUserDetectorConstruction`² para definir a posição, orientação dos materiais que compõe o nosso alvo. Foi usado um dos tecidos moles que encontrava-se na lista NIST de materiais, com o código `G4_TISSUE_SOFT_ICRP`, com densidade $\rho = 1.03 \text{ g/cm}^3$.

```
// Inicia o G4NistManager e procura pelo material G4_TISSUE_SOFT_ICRP
G4NistManager* manager = G4NistManager::Instance();
G4Material* softTissue = manager->FindOrBuildMaterial(
    "G4_TISSUE_SOFT_ICRP");
```

A dimensão do alvo tem a largura e altura do “mundo” da simulação e uma profundidade de 5 m (que é muito maior do que o alcance médio dos prótons para o espectro de energias que iremos simular). O material que compõe o ambiente envolvente é vácuo, de modo a considerarmos apenas as interações com o alvo.

```
// Tamanho do mundo da simulacao
G4double fWorldLength= 10.0*m;
G4double HalfWorldLength = 0.5*fWorldLength;
// Espessura do alvo
G4double targetThickness = 5*m;
// Criar alvo com o dado tamanho e material
G4Box* solidTarget = new G4Box("target",0.5*targetThickness,
    HalfWorldLength,HalfWorldLength);
G4LogicalVolume* logicTarget = new G4LogicalVolume(solidTarget,
    softTissue,"Target",0,0,0);
G4ThreeVector pos = G4ThreeVector(0.5*targetThickness,0,0);
```

¹<https://lip.pt/~alex/G4Classes/Examples/BraggPeak.zip>

²https://apc.u-paris.fr/~franco/g4doxy4.10/html/class_g4_v_user_detector_construction.html

```
G4VPhysicalVolume* physiTarget = new G4PVPlacement(0, pos,
    logicTarget, "Target", logicWorld, false, 0);
```

2).2 Gerador da Partículas Primárias

Para definir as partículas que são geradas na simulação, utiliza-se uma classe do tipo `G4VUserPrimaryGeneratorAction`³ para definir o evento de “geração” de partículas. Nessa classe, é criada uma `G4ParticleGun`⁴ que emite um próton.

```
// Criar uma particle gun de uma particula
G4ParticleGun particleGun = new G4ParticleGun(1);
// Obter informacao dos protoes
G4ParticleTable *particleTable =
    G4ParticleTable::GetParticleTable();
G4ParticleDefinition *particle =
    particleTable->FindParticle("proton");
particleGun->SetParticleDefinition(particle);
// Definir posicao, orientacao e momento da particula
// (igual para todas as particulas)
particleGun->SetParticleMomentumDirection(
    G4ThreeVector(1., 0., 0.));
particleGun->SetParticlePosition(
    G4ThreeVector(-1*m, 0. * cm, 0. * cm));
```

Para a nossa simulação, considerámos que o feixe de prótons tem uma distribuição energética que segue uma distribuição Gaussiana, com $\sigma = 2\%$ e $\mu = \{50, 60, \dots, 150\}$ MeV, onde efetuamos uma simulação para cada valor diferente da energia média do feixe. O valor foi gerado aleatoriamente utilizando o método de Box-Muller, para cada partícula do feixe.

```
// Distribuicao Normal (mu=1, std=0) usando metodo de Box-Muller
G4float Z = pow((-2*log(G4UniformRand())), 0.5) *
    cos(2*M_PI*G4UniformRand());
// Media e desvio padrao que sao pretendidos
G4float mean_value = 100*MeV;
G4float std = 0.02;
particleGun->SetParticleEnergy(Z*std*mean_value + mean_value);
```

2).3 Recolha informação para análise

Para recolher os dados da simulação, criamos uma classe do tipo `G4UserSteppingAction`⁵ que é chamada a cada passo da iteração da simulação.

Em cada passo, o programa verifica se a partícula que está a ser seguida neste passo encontra-se no volume do detector, e se é uma partícula primária e obtém a energia que foi depositada nessa interação, a profundidade a que a partícula se encontra e a variação de posição simulada nesta interação, registando os valores no ficheiro “BraggPeak.out”.

³https://apc.u-paris.fr/~franco/g4doxy4.10/html/class_g4_v_user_primary_generator_action.html

⁴<https://apc.u-paris.fr/~franco/g4doxy/html/classG4ParticleGun.html>

⁵<https://apc.u-paris.fr/~franco/g4doxy/html/classG4UserSteppingAction.html>

```

// {Na inicializacao da classe...}
std::ofstream bout;
bout.open("BraggPeak.out");
// {Codigo chamado a cada passo...}
G4Track* thisTrack = thisStep->GetTrack();
G4VPhysicalVolume* theVolume = thisTrack->GetVolume();
// Ver se evento aconteceu no alvo e se foi com uma partícula
// primaria (i.e. nao tem parent track)
const G4ParticleDefinition* thisParticle =
    thisTrack->GetParticleDefinition();
if(theVolume->GetName() == "Target" &&
    thisParticle->GetParticleName() == "proton"){
    bout << thisTrack->GetTrackLength() / cm << "\t"
        << thisStep->GetTotalEnergyDeposit() / MeV << "\t"
        << thisStep->GetStepLength() / cm << G4endl;
}

```

3) Resultados obtidos e Discussão

Como mencionado previamente, efetuamos uma simulação ($n = 10000$ partículas emitidas em cada simulação) para cada valor de energia média do feixe (50 até 150 MeV, em passos de 10 MeV). As figuras seguintes mostram a dosagem medida em função da profundidade para cada energia média inicial do feixe.

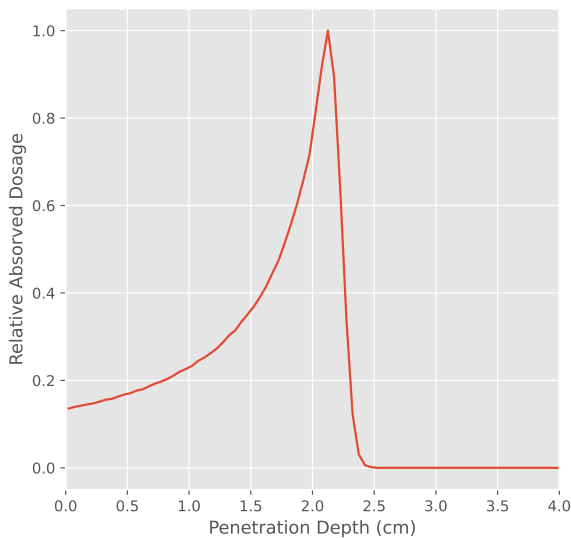


Figura 2: Dosagem medida em função da profundidade para um feixe com energia média de 50 MeV. O pico da dosagem é observado a uma profundidade de 2.13 ± 0.05 cm.

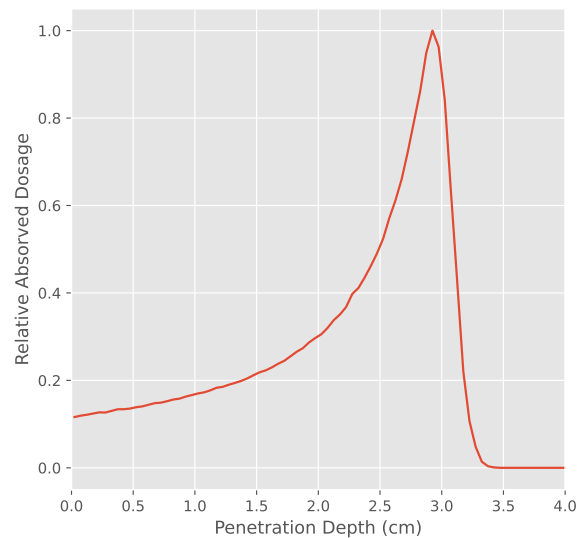


Figura 3: Dosagem medida em função da profundidade para um feixe com energia média de 60 MeV. O pico da dosagem é observado a uma profundidade de 2.93 ± 0.05 cm.

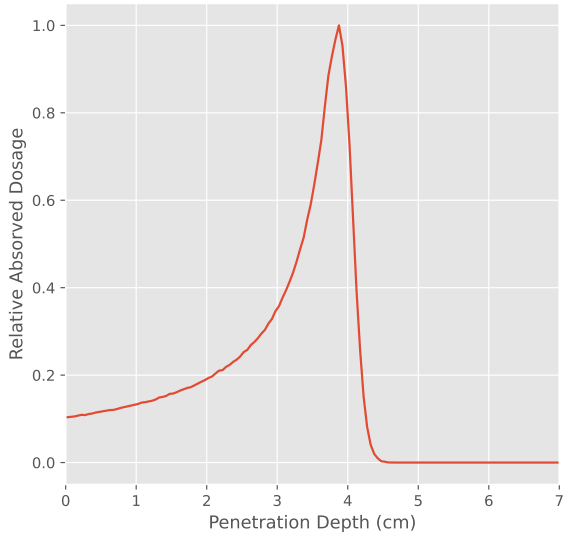


Figura 4: Dosagem medida em função da profundidade para um feixe com energia média de 70 MeV. O pico da dosagem é observado a uma profundidade de 3.88 ± 0.05 cm.

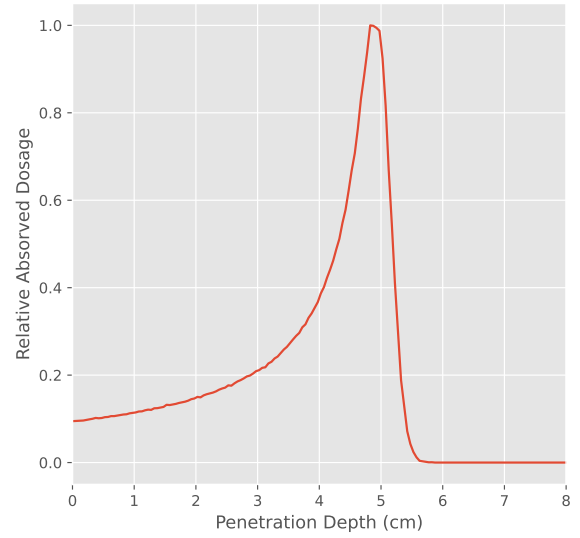


Figura 5: Dosagem medida em função da profundidade para um feixe com energia média de 80 MeV. O pico da dosagem é observado a uma profundidade de 4.83 ± 0.05 cm.

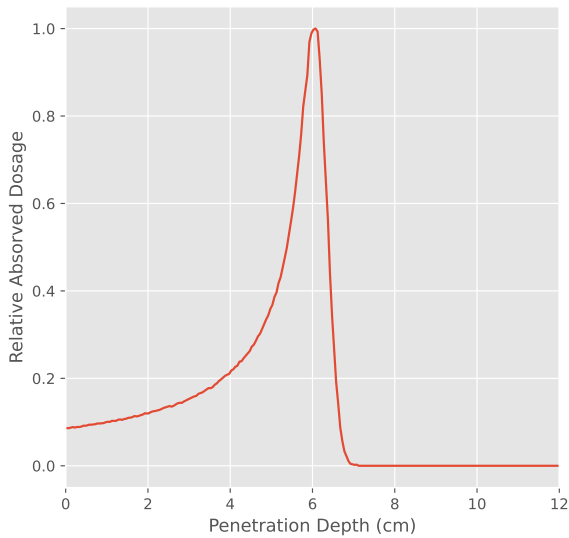


Figura 6: Dosagem medida em função da profundidade para um feixe com energia média de 90 MeV. O pico da dosagem é observado a uma profundidade de 6.08 ± 0.05 cm.

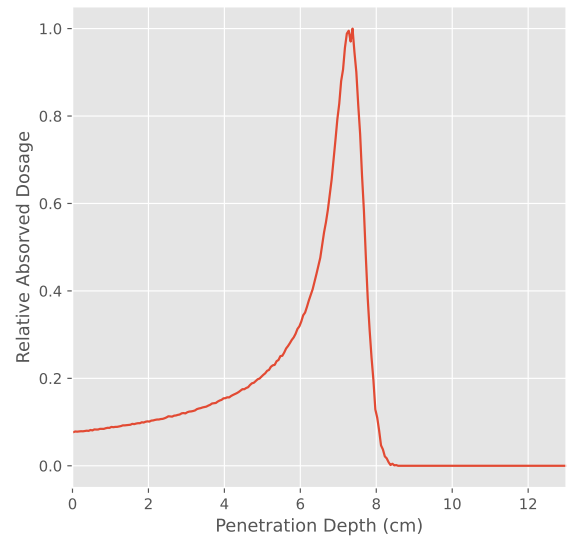


Figura 7: Dosagem medida em função da profundidade para um feixe com energia média de 100 MeV. O pico da dosagem é observado a uma profundidade de 7.38 ± 0.05 cm.

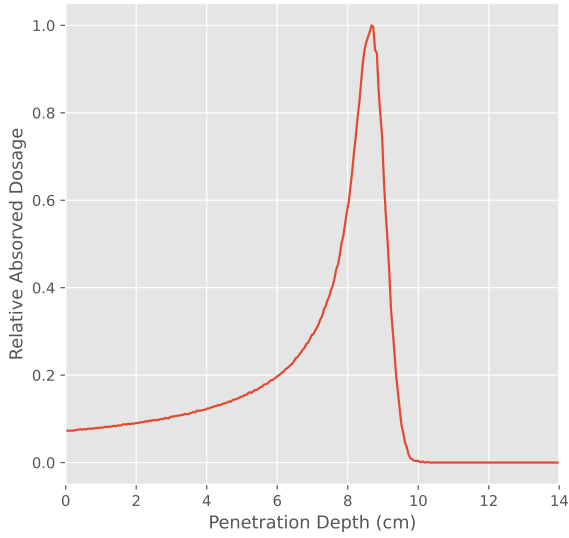


Figura 8: Dosagem medida em função da profundidade para um feixe com energia média de 110 MeV. O pico da dosagem é observado a uma profundidade de 8.68 ± 0.05 cm.

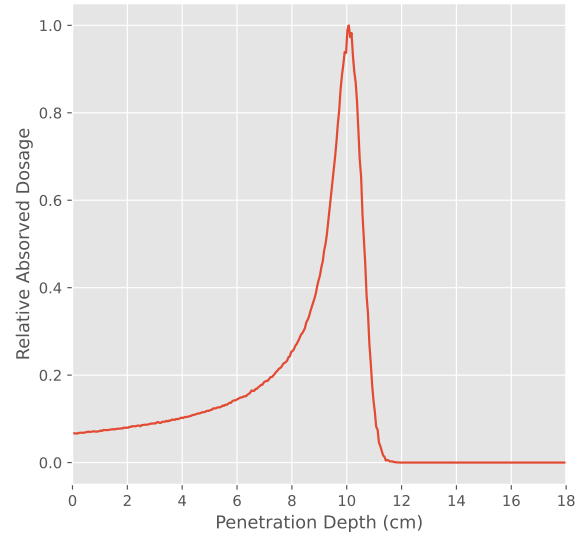


Figura 9: Dosagem medida em função da profundidade para um feixe com energia média de 120 MeV. O pico da dosagem é observado a uma profundidade de 10.08 ± 0.05 cm.

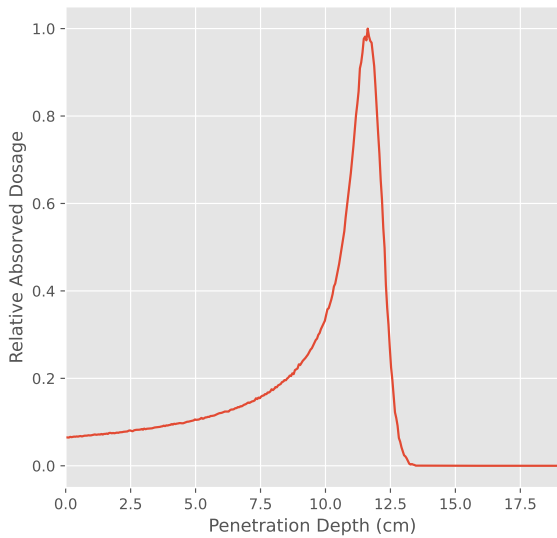


Figura 10: Dosagem medida em função da profundidade para um feixe com energia média de 130 MeV. O pico da dosagem é observado a uma profundidade de 11.63 ± 0.05 cm.

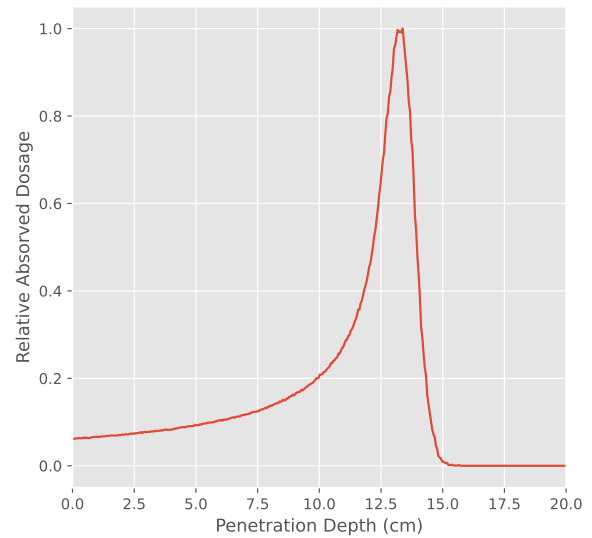


Figura 11: Dosagem medida em função da profundidade para um feixe com energia média de 140 MeV. O pico da dosagem é observado a uma profundidade de 13.38 ± 0.05 cm.

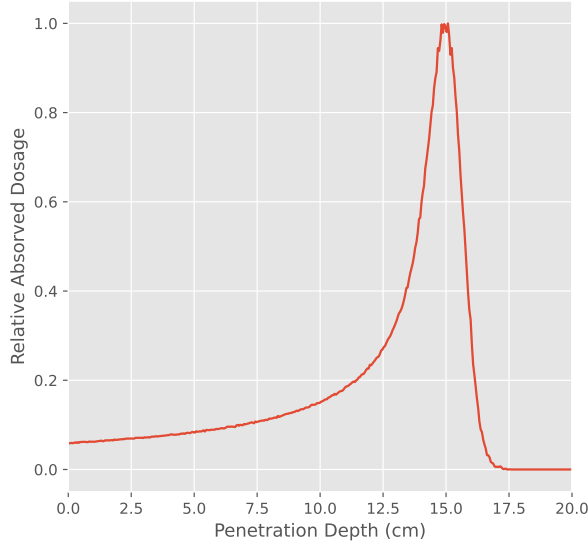


Figura 12: Dosagem medida em função da profundidade para um feixe com energia média de 150 MeV. O pico da dosagem é observado a uma profundidade de 15.08 ± 0.05 cm.

Para estimar a energia do feixe para tratar um tumor a 7 cm de profundidade, utilizámos os picos obtidos anteriormente para efetuar uma interpolação linear para obter o valor de energia médio associado, obtendo uma energia de $E = 97.12 \pm 0.30$ MeV para tratar o tumor minimizando a destruição do tecido envolvente.

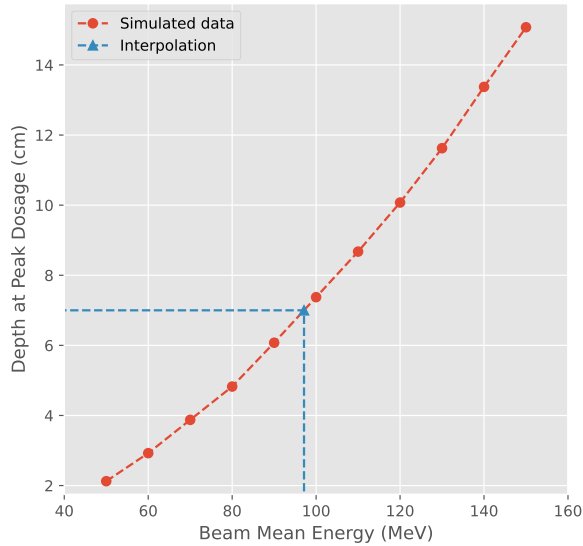


Figura 13: Pontos médios dos picos de Bragg obtidos em função da energia média do feixe de prótons. O ponto correspondente a 7cm está também marcado no gráfico, sendo obtido através de uma interpolação linear com os pontos de 90 e 100 MeV, correspondendo a uma energia de 97.12 ± 0.30 MeV⁶.

Dependendo da dimensão do tumor, poderá ser necessário modelar a distribuição energética do feixe para ter um *spread-out Bragg peak* (SOBP), de modo atingir o tumor inteiro.

⁶O erro foi obtido através da fórmula de propagação de erro: $\sigma f(x_1, x_2, \dots)^2 = \sum_n \left(\sigma x_n \frac{\partial f}{\partial x_n} \right)^2$.

Referências

- [1] W. R. Leo, *Techniques for nuclear and particle physics experiments: a how-to approach; 2nd ed.* Berlin: Springer, 1994. [Online]. Available: <https://cds.cern.ch/record/302344>