university of groningen / faculty of science and engineering / biomedical engineering

Pyramid Filter: Design of a 3D Range Spreader for FLASH Irradiations

Clara Popescu-Boboc

S5067421

PARTREC (PArticle Therapy Research Center)

Period: 22/04/2025 - 08/07/2025

Bachelor's Project

1st Examiner: Dr. Alexander Gerbershagen (Head of PARTREC, Team Leader for Accelerator and Radiation Physics)

2nd Examiner: Dr. Peter Dendooven (Associate Professor at University Medical Center Groningen, Professor at the Helsinki Institute of Physics)

Daily supervisor: MSc Atia Ibrahimi (PhD student at PARTREC)

Contents

Acknowledgements

First and foremost, I extend my heartfelt gratitude to my two supervisors, Dr. Alexander Gerbershagen and Dr. Peter Dendooven, for organizing weekly meetings and offering valuable feedback, despite their demanding schedules. I am also deeply thankful to Prof. Marco Schippers for giving me the opportunity to conduct this research at PARTREC. My special thanks go to PhD Atia Ibrahimi, my daily supervisor, who was consistently generous with her time, deeply invested in ensuring I understood every detail, proactive in offering assistance, and who often steered me away from fruitless efforts. I would also like to acknowledge Ms. Lluna López Llorens, the previous student whose work I continued, for generously sharing her codes and an unpublished manuscript that helped me understand the concepts and allowed me to bring new contributions. Finally, I am grateful to my parents, grandparents, and my older sister for their emotional and financial support throughout my academic journey, from childhood to the present. ”Nihil sine Deo”.

Abstract

The thesis investigates the design and optimisation of pyramid-shaped proton range modulator filters to improve dose distribution uniformity in high-speed FLASH radiotherapy. To address this, the study asks: (1) How does the spatial configuration of pyramid filters affect dose distribution and SOBP uniformity in proton therapy? (2) What is the effect of introducing a collimator in combination with pyramid filters on beam shaping and dose conformity? (3) How do variations in the base layer design influence proton beam penetration range and modulation for targeting irregularly-shaped tumours? To answer these questions, Python-based models were developed and validated using TOPAS Monte Carlo simulations. A 17×17 array of pyramids significantly improved dose uniformity across the SOBP, compared to a single pyramid setup. Integrating a copper collimator with a 3×3 cm² air-gap effectively confined the modulated beam to the tumour region, without degrading the SOBP. Additionally, adjusting the filter's base layer thickness modulated penetration depth, and a split base design enabled lateral range variation for better conformity to asymmetric tumour geometries. Based on these results, the project recommends tailored pyramid configurations and collimation for achieving homogeneous, conformal dose delivery in FLASH proton therapy. In conclusion, the work shows that a simulation-driven filter design can strengthen FLASH treatment precision, while sparing surrounding healthy tissue.

1 Introduction

The first chapter of the thesis details the pyramid filter design rationale for FLASH irradiations (1.1), formulates the project research questions (1.2), and reviews relevant theoretical concepts in proton therapy physics (1.3).

1.1 Background & Motivation

In the present age, cancer care delivery has undergone a shift from a disease-focused management toward a patient-centric approach (Vincenzo et al., 2020). Within this evolving landscape, radiotherapy (RT) remains a fundamental component of effective treatment, contributing to nearly 40% of therapeutic regimens (Roomi, 2023).

Among the most significant advancements in modern external beam RT is proton therapy, a non-invasive and highly precise modality used to treat both malignant and benign tumours (Press and Mehta, 2024; Roomi, 2023). The radiological use of fast protons was first proposed in 1946 by the American physicist Robert R. Wilson, who identified that protons deposit their energy primarily at the end of their path, allowing ionization of tissue at a specific depth while sparing bordering structures (Wilson, 1946). Following decades of research and development, the technique received regulatory approval from the U.S. Food and Drug Administration (FDA) in 1988 for clinical use (Northwestern Medicine, 2025).

One of the ways in which proton therapy distinguishes itself from conventional photon-based techniques is through its ability to minimise irradiation of surrounding healthy tissues and organs at risk (e.g. lungs, heart, spinal cord), as it is visible in Figure 1.1. While photons deposit energy continuously along their entire path through the body (Patient Resource, 2024), protons release most of theirs at a specific tissue depth, a phenomenon known as the Bragg peak, beyond which the dose falls off sharply (Graeff et al., 2023). However, the Bragg peak of a monoenergetic proton beam is too narrow to fully cover the depth of a tumour, which is often irregular in shape. To address this limitation, proton beams of varying energies are superimposed (Figure 1.2), effectively stacking multiple Bragg peaks to create a Spread-Out Bragg Peak (SOBP). This results in a uniform dose across the entire tumour volume, while preserving the steep dose gradient that protects distal healthy tissues (Hazem, 2023).

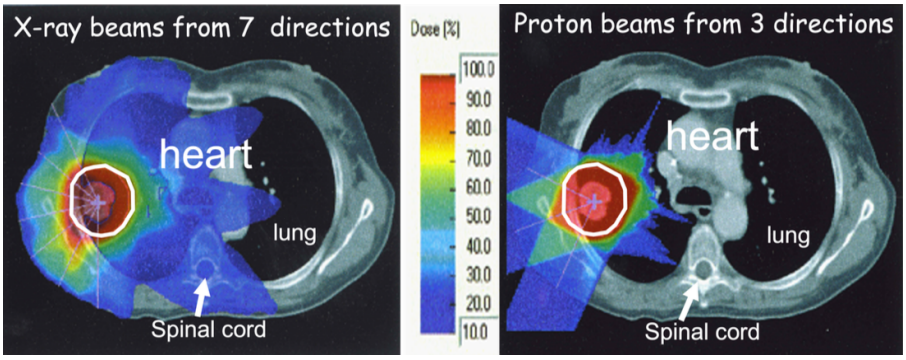

Figure 1.1: Comparison of dose distributions for a lung tumour irradiation with X-rays from seven angles (left) and with proton beams from three angles (right) (Auberger et al., 2004).

To modulate the energy, proton beams of a fixed energy are passed through degraders, which are typically composed of low-Z materials such as graphite or Lucite. These reduce beam energy while

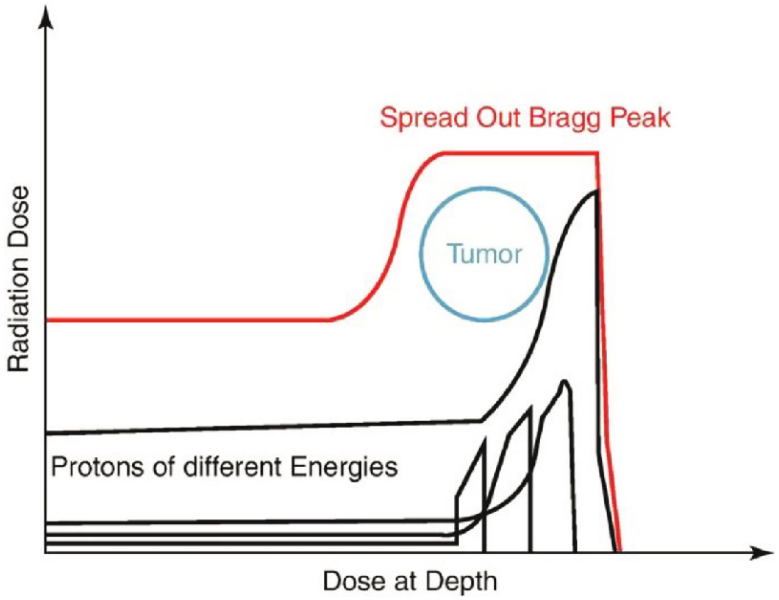

Figure 1.2: Dose deposition as a function of depth in tissue, showing the formation of the SOBP by superposition of Bragg peaks of different energies (Hazem, 2023).

minimising scattering (Das et al., 2015). An alternative method to spread the Bragg peak range in a beam delivery system is the use of for instance brass modulation wedges or aluminium ridge filters. The latter consists of repeated bar-shaped structures whose step thickness determines the pullback of individual Bragg peaks, and whose width sets their weighting. Its main drawback is that it can only modulate energy in one dimension. Figure 1.3 shows a ridge filter designed for a spatial modulation width of 6 cm (Akagi et al., 2003).

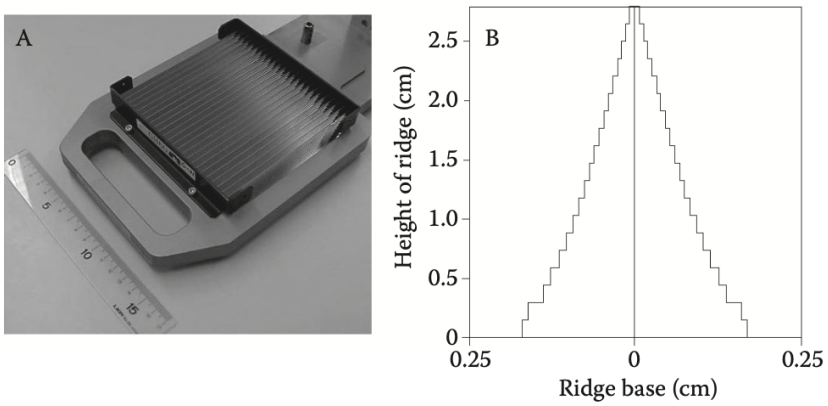

Figure 1.3: (A) Photograph of a bar ridge filter. (B) Cross-section of a ridge (Akagi et al., 2003)).

Proton therapy not only improves tumour targeting, but also leads to a substantial reduction in the integral dose received by adjacent tissues, following the ALARA principle (As Low As Reasonably Achievable). Clinically, this should translate into a lower incidence of serious, often irreversible, late side effects and a reduced risk of radiation-induced secondary malignancies (Health Council of the Netherlands, 2009).

On a molecular level, the therapeutic effect of proton irradiation is primarily mediated by the induction of DNA double-strand breaks (DSBs), which are among the most cytotoxic forms of dam-

age (Figure 1.4). DSBs trigger a cascade of cellular responses involving repair mechanisms, such as non-homologous end joining (NHEJ) and homologous recombination repair (HRR) (Deycmar et al., 2020). Proton beams, toward the distal end of the SOBP where both the linear energy transfer (LET) and relative biological effectiveness (RBE) increase, lead to more complex and challenging-to-repair DNA lesions (Failla and Henshaw, 1931). When repair is unsuccessful, cells may undergo mitotic catastrophe due to chromosomal aberrations, ultimately leading to cell death. Additionally, the accumulation of fragmented DNA within micronuclei can stimulate innate immune responses via the cGAS–STING pathway, contributing to immunogenic cell death (Deycmar et al., 2020).

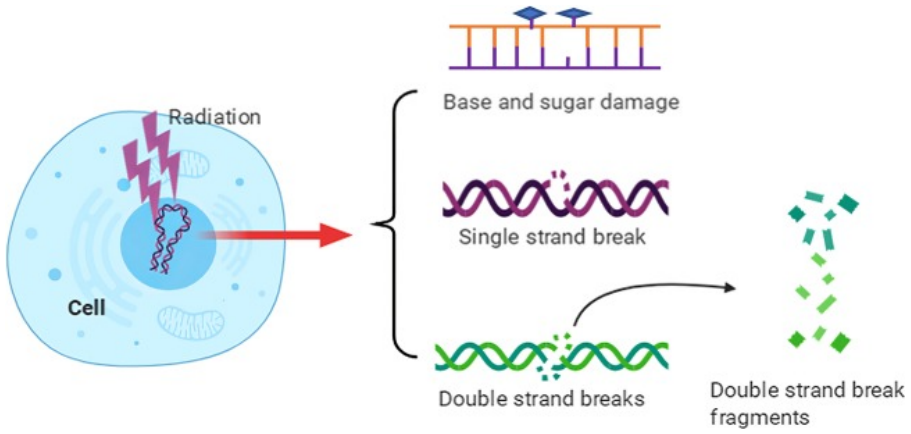

Figure 1.4: Illustration of proton-induced DNA damage: bases, sugars, single and double-strand breaks (Huang and Zhou, 2020).

There is an evolving role for novel imaging and delivery techniques in modern radiotherapy (RT) planning. Among the most recent advancements shaping the field are modalities such as: stereotactic ablative body radiotherapy (SABR), proton-beam therapy (PBT), on-line adaptive radiotherapy (ART), re-irradiation (re-RT), etc. The focus of this paper will be on FLASH RT.

FLASH RT is an emerging technique that delivers therapeutic doses at ultra-high dose rates ($\geq$ 40 Gy/s) within milliseconds, eliciting a unique biological response that preferentially spares healthy tissues, while maintaining tumour control (Favaudon et al., 2024). Despite its preclinical success, the translation of FLASH RT into proton therapy remains limited by speed constraints in the dose delivery process. An important limitation in this context is the absence of a mechanical system capable of modulating energy and spatial distribution at the required speed and precision.

One approach currently under investigation to overcome these challenges is the use of pyramid filters (see Figure 1.5), which are three-dimensional, static range modulators composed of vertical, pyramid-shaped pins. Unlike traditional one-dimensional ridge filters, these allow for advanced modulation without relying on time-intensive accelerator adjustments, making them particularly well suited for FLASH-capable delivery systems (Jolly et al., 2020; Simeonov et al., 2017).

1.2 Research Questions

1. How does the spatial configuration of pyramid filters affect dose distribution and SOBP uniformity in proton therapy?

2. What is the effect of introducing a collimator in combination with pyramid filters on beam shaping and dose conformity?

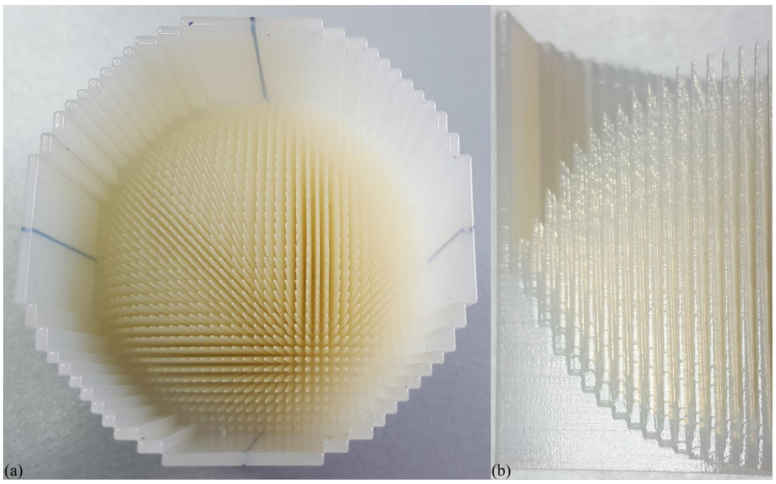

Figure 1.5: Photograph of a pyramid filter (Simeonov et al., 2017).

3. How do variations in the base layer design influence proton beam penetration range and modulation for targeting irregularly-shaped tumours?

1.3 Theoretical Background

At therapeutic energies (typically 70–250 MeV), the dominant mechanisms through which protons interact with matter are Coulomb interactions with atomic electrons (causing ionization and excitation), elastic Coulomb scattering with atomic nuclei (deflecting the proton's path), and inelastic nuclear reactions (leading to particle loss and secondary radiation). These processes collectively determine the range, dose distribution, and biological impact of the proton beam (FitzGerald, 2024; Olivari, 2024; Paganetti, 2012).

1.3.1 Stopping Power and Absorbed Dose

The average energy loss of a proton per unit path length in matter is known as the linear stopping power S. It can be expressed as the sum of the energy lost due to Coulomb interactions with electrons and the energy loss through elastic collisions with atomic nuclei (FitzGerald, 2024; ICRU, 2011):

$$S = -\frac{dE}{dx} = -\left.\frac{dE}{dx}\right|_{\text{elec}} + -\left.\frac{dE}{dx}\right|_{\text{nuc}} \quad (\text{MeV/mm}) \tag{1.1}$$

The absorbed dose D is defined as the mean energy ε deposited per unit mass of material (ICRU, 2011):

$$D = \frac{d\varepsilon}{dm} \quad (\text{Gy}) \tag{1.2}$$

1.3.2 Coulomb Interactions with Electrons: The Bethe–Bloch Formula

The energy loss due to the ionization and excitation of electrons is quantified by the Bethe–Bloch equation (Bloch, 1933; Bethe, 1930):

$$\left(-\frac{dE}{dx}\right)_{\text{elec}} = \frac{4\pi r_e^2 m_e c^2 z^2}{\beta^2} \cdot \frac{Z}{A} \cdot \rho N_A \left[\ln\left(\frac{2m_e c^2 \beta^2}{1-\beta^2}\right) - \beta^2 - \ln I\right] \quad (\text{MeV/mm}), \tag{1.3}$$

where r_e and m_e are the radius and mass of an electron, z refers to the charge number of the incident particle (with $z = 1$ for protons), $\beta = \frac{v}{c}$ denotes the particle's speed v relative to the speed of light c. The parameters Z, A, and ρ represent the atomic number, atomic mass, and density of the absorber material, respectively. N_A is Avogadro's number, and I is the mean excitation energy of the absorbing medium.

1.3.3 Range and Range Straggling

The range of a proton is the average depth in tissue at which the particle comes to rest. Due to stochastic variations in energy-loss events, not all protons stop at exactly the same depth, leading to a phenomenon known as range straggling. This effect results in a finite width of the distal fall-off of the Bragg peak and becomes more significant at higher proton energies (Bortfeld, 1997; Olivari, 2024).

To estimate the depth a proton will reach in tissue, the Bragg–Kleeman rule provides an empirical relationship between residual range and energy (Bragg and Kleeman, 1905):

$$E(z) = \left(\frac{1}{\alpha^{1/p}} (R_0 - z) \right)^{1/p} \tag{1.4}$$

where R_0 is the initial proton range in water, z is the depth, and α and p are material-dependent parameters. Equation 1.4 is particularly relevant for beam energy tuning in simulations, such that the Bragg peak aligns with the target tumour depth.

1.3.4 Elastic Coulomb Scattering with Nuclei: The Highland Formula

As protons traverse matter, they undergo elastic Coulomb scattering with atomic nuclei, resulting in angular deflections that broaden the beam's lateral spread (approximately Gaussian in profile). This process, known as multiple Coulomb scattering (MCS), is described in clinical physics by the following approximation (Highland, 1975; Bethe, 1953):

$$\sigma_\theta = \frac{14.1\,\text{MeV}}{\beta\, p\, c} z \sqrt{\frac{L}{L_R}} \left(1 + \frac{1}{9} \log_{10} \left(\frac{L}{L_R} \right) \right) \quad \text{rad}, \tag{1.5}$$

where σ_θ is the lateral angular spread, β and z have the same meanings as in Equation 1.3, p is the proton momentum, L denotes the thickness of the target material, and L_R is its radiation length.

1.3.5 Inelastic Nuclear Reactions

Although less frequent than electromagnetic interactions, inelastic nuclear reactions between protons and atomic nuclei can affect the dose deposition. These lead to the loss of primary protons and the production of secondary particles, such as: neutrons, gamma rays, and light nuclear fragments. While inelastic nuclear events contribute only about 1–2% of dose per centimetre at 160 MeV, they can influence the distal dose tail and must be accounted for in dosimetric models and shielding calculations (Olivari, 2024).

1.3.6 Material Compensation and Pyramid Array Geometry

When non-water materials (e.g., PMMA, Nylon-12) are placed in the beam path, their impact on proton range in water must be accounted for using the water-equivalent thickness (WET) formula (Zhang, 2009):

$$t_w = t_m \frac{\rho_m}{\rho_w} \frac{S_m}{S_w} \tag{1.6}$$

where t_m is the physical thickness of the material, and ρ and S are the density and stopping power of the material and water, respectively. This adjustment is needed to properly alter the beam energy in order to preserve tumour-targeting accuracy.

Additionally, the lateral modulation grid used in the pyramid filter must be centred on the tumour and wide enough to span its cross-sectional area. If the tumour has width W and pyramids are spaced by l, the number of pyramid elements along one dimension is given by:

$$N = \left\lfloor \frac{W}{l} \right\rfloor + 1 \tag{1.7}$$

Equation 1.7 was derived to fully cover the target (including both tumour edges) and enforces symmetry by using an odd number of elements. This guarantees that the array is centred on the beam axis ($x = 0$, $y = 0$) and the pyramid at the origin is aligned to the middle of the tumour.

Lastly, after the number of pyramid elements N is found, the total lateral beam aperture A required to cover the array can be calculated as:

$$A = N \cdot l \tag{1.8}$$

2 Materials & Methods

2.1 Experimental Setup

The simulation environment was constructed using TOPAS (TOol for PArticle Simulation) (Faddegon et al., 2020; Perl et al., 2012), a Monte Carlo platform built on GEANT-4 (Agostinelli et al., 2003). Figure 2.1 shows a lateral view of the defined World's geometry: the monoenergetic proton beam (dark blue trajectories) enters from the left, passes through the polymethylmethacrylate (PMMA) pyramid filter consisting of a base layer (light blue) and smaller layers on top (alternating yellow-pink pyramid layers), then through the copper collimator (yellow), and finally deposits dose within a cubic water phantom (transparent blue box), where the scoring volume is shown in pink. For dimensions and specs, see Table 2.1.

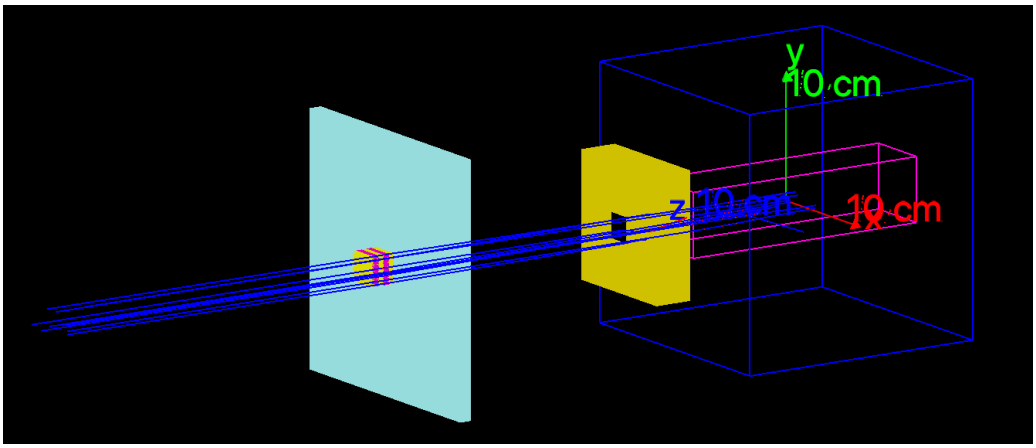

Figure 2.1: TOPAS simulation setup showing the main coordinate system. The x-axis (red) represents lateral spread, the y-axis (green) corresponds to vertical height, and the z-axis (dark blue) indicates beam penetration depth.

2.2 Base Case Definition

The tumour region of interest was modelled as a cubic volume of $2 \times 2 \times 2$ cm^3, centred at a depth of 11 cm along the z-axis, as illustrated in Figure 2.2. A cube was chosen instead of a sphere or anatomically realistic tumour shape to simplify geometry handling and dose scoring. Furthermore, its relatively small size was selected not only to represent a high-precision treatment scenario, but also with the intention of scaling down to even smaller target volumes in future work, particularly in alignment with preclinical testing on small animal models.

To resemble human tissue, the simulations used a homogeneous, water phantom with dimensions of $20 \times 20 \times 20$ cm^3. A scoring volume of $5.05 \times 5.05 \times 20$ cm^3 was defined to encompass the entire beam path and allow spatial dose analysis in both lateral and depth directions.

All tests were performed in vacuum to isolate and study the physical effects of beam modulation and filtering without the influence of air scattering or energy loss prior to the phantom. While Figure 2.1 shows only 10 representative proton trajectories for visual clarity, each configuration presented in chapter 3 Results & Discussion was simulated with 100 million protons to achieve high statistical precision in dose scoring. The proton beam was defined with a flat spatial distribution and an energy spread of 0.6. In TOPAS, this value is interpreted by default as a Gaussian energy distribution with

Table 2.1: Simulation components and physical parameters.

Component	**Material**	**Dimensions**	**Position**	**Function**
Proton Beam	N/A	Energy: 136.7 MeV	TOPAS Built-in "BeamPosition"	Delivers therapeutic radiation; beam width matches pyramid base array.
Pyramid Filter	PMMA	Base layer: 1 cm; 17×17 array of smaller layers	Directly downstream of beam	Modulates energy deposition; shapes a flat SOBP via stacked pyramid layers.
Collimator	Copper	$5 \times 5 \times 3$ cm^3	After filter; 3 cm upstream of phantom	Restricts lateral spread; includes a central $3 \times 3 \times 3$ cm^3 air opening.
Phantom	Water	$20 \times 20 \times 20$ cm^3	25 cm away from pyramid filter	Homogeneous medium for realistic dose deposition and measurement in human tissue
Scoring Volume	Water	$5.05 \times 5.05 \times 20$ cm^3, $101 \times 101 \times 400$ bins	Within water phantom	Region of interest where the dose deposition is scored for assesing SOBP flatness and treatment accuracy.

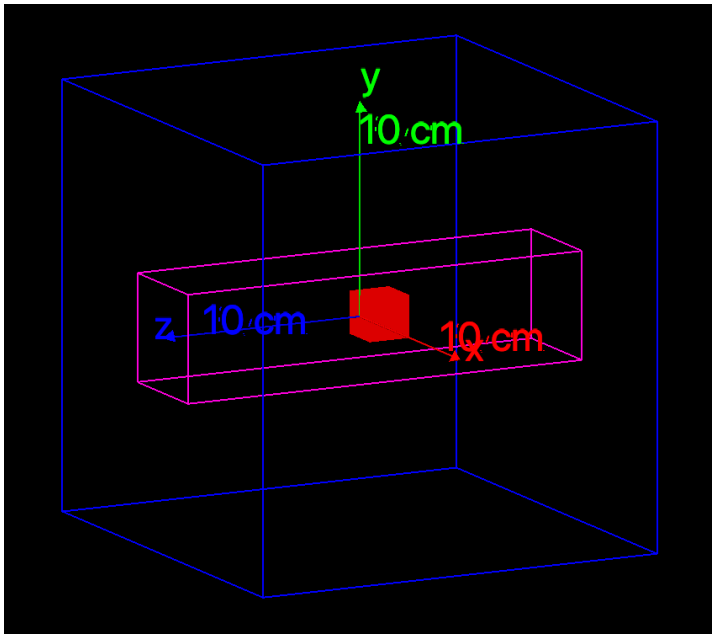

Figure 2.2: Visualisation of the scoring volume (pink) within the water phantom (dark blue), with the virtual tumour shown in red.

a standard deviation σ equal to 0.6% of the central beam energy. The energy spread itself describes the natural statistical variation in particle energies around the nominal value, with a Gaussian profile indicating that most protons are close to the mean energy and fewer deviate significantly. To illustrate, for a beam energy of 136.7 MeV, this corresponds to $\sigma \approx 0.82$ MeV.

These parameters were kept constant across all experiments to allow direct comparison between simulation scenarios and to ensure that observed effects were attributable solely to filter geometry and modulation strategy.

2.3 Starting Point from Previous Work

This work builds upon the simulation framework developed by a former University of Groningen Physics student (Lopez Llorens, 2024). Her implementation was based on two scripts: one that employed the Georges code documentation (Tesse et al., 2023) to generate optimised SOBP weights, and another that created based on the obtained weights the pyramid filter layers. While the core logic of both codes was preserved, they were extended and automised as much as possible to allow greater flexibility and applicability across different tumour sizes, depths, and setup configurations, especially with the intention of upcoming students to take over.

Importantly, it was also her current work that motivated the decision to use a single beam in this paper. Although the specifics of her spot placement studies fall outside the scope of this thesis, her new findings demonstrated that even with complex beam spot arrangements, one could provide sufficiently conformal dose coverage.

The updated code versions, as used in this thesis, retain credit to the original author where appropriate and are documented in Appendix B.

2.4 Simulation Steps

This section outlines the steps that were followed to design, validate, and evaluate the pyramid filter. While the numerical results are presented in the next chapter, the following describes how the computations were conducted. Moreover, a dedicated user guide is provided in the Appendix for Linux and macOS users, with instructions on how to generate the SOBP, as well as running simulations on Hábrók, the University of Groningen's high-performance computing (HPC) cluster.

2.4.1 Python-Based Generation of SOBP Weights and Pyramid Geometry

The simulation workflow began in Python, where the Georges code was used to generate the weights tailored to the tumour depth. A voxel resolution of 0.05 cm was chosen to discretise the 2 cm tumour thickness into six Bragg peaks (see Figure 2.3). The initial energy of the beam was estimated using the Bragg-Kleeman relation (Equation 1.4) and was found to be $E_i = 129.5$ MeV.

To physically support and integrate the pyramid elements into a single filter structure, a continuous base layer is required beneath. Consequently, the WET formula (Equation 1.6) was applied to quantify the additional range introduced by a 1 cm PMMA base layer and was calculated as 1.158 cm. Therefore, the beam range had to be increased to maintain the distal edge at 12 cm, which led to an adjusted beam energy of $E_f = 136.7$ MeV.

The obtained weights were then fed into the Python code responsible for determining the pyramid layer dimensions. The updated version introduced in this thesis allows the user to adjust the input parameters, such as the distance between the pyramid filter and the water phantom and/or the thickness and material composition of the layers.

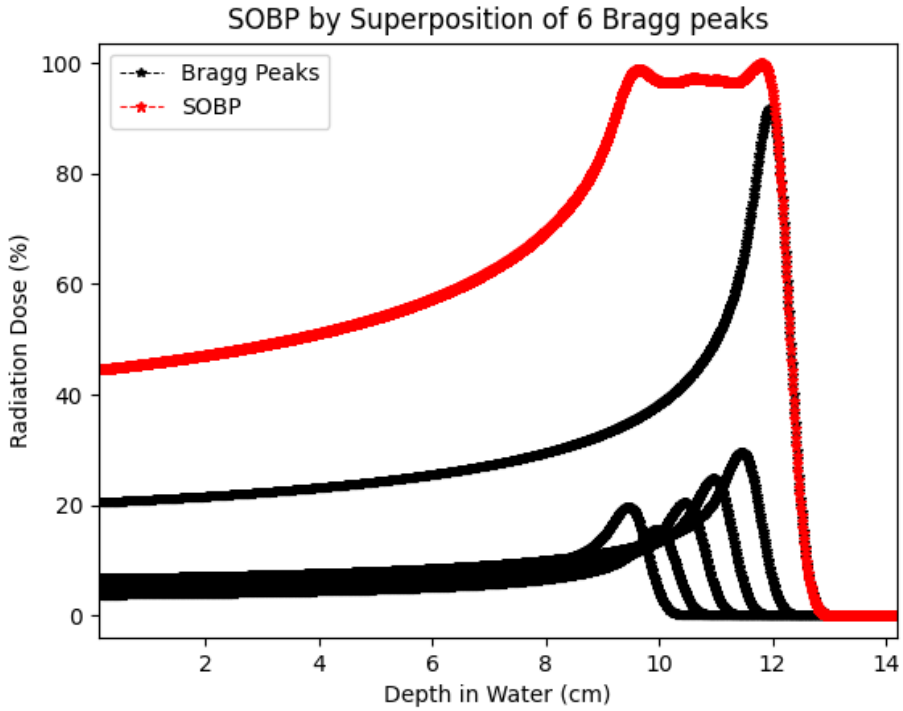

Figure 2.3: Depth-dose curve demonstrating the superposition of 6 weighted Bragg peaks (black), forming the SOBP (red).

2.4.2 TOPAS Simulations for Dose Validation

The geometry output was imported into TOPAS for Monte Carlo simulations. To account for the lateral coverage of the full tumour cross-section (2×2 cm^2), the number of pyramids in the filter array was computed using the lateral spacing formula (Equation 1.7), yielding a 17×17 array, with each pyramid spaced 0.14 cm apart. The beam aperture was determined using Equation 1.8, resulting in a total width of 0.14 cm for only one pyramid and 2.38 cm for the 17×17 array. An illustration of the pyramid filter is shown in Figure 2.4.

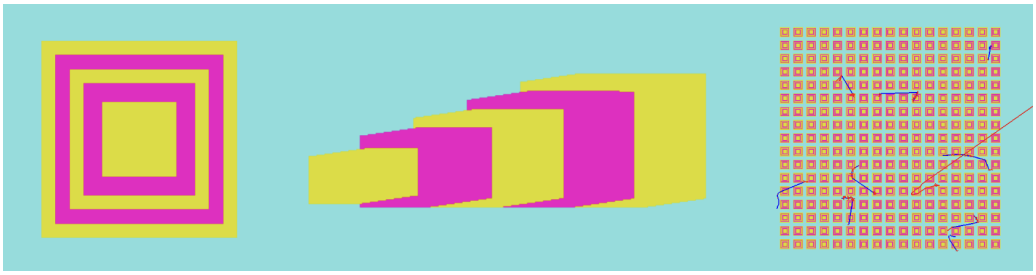

Figure 2.4: Geometry of the pyramid filter. From left to right: top view of one pyramid, side view showing stacked modulation, and top view of the complete 17×17 array with 10 particles entering for representation. Different colours are assigned for visual clarity only; all layers are made from the same material.

Fine-tuning of the filter design was conducted through iterative testing in which specific features were varied. In particular, the cross-sectional area of the topmost pyramid layers was reduced to shape

the proximal dose, and the distance between the filter and phantom was adjusted to reach a dose distribution in the lateral (x, y) directions. These optimisations are mentioned here for completeness, but their effects are fully demonstrated in subchapter 3.1 of Results & Discussion.

2.4.3 Python-Based Post-Processing and Dose Translation

The last step after obtaining the dose scoring file from TOPAS was analysing the full 3D dose distribution, including the SOBP and longitudinal depth-dose profiles, as well as transverse dose profiles and overall dose uniformity within the tumour region, using Python visualizations. Additionally, a dose addition tool was implemented to reduce computational demands for multi-pyramid configurations. This code (see Appendix B.2.5) replicates the dose matrix from a single-pyramid simulation by spatially shifting and summing the distributions across a defined grid, where each shift corresponds to the spacing between pyramid elements -approximately 0.14 cm-. In doing so, the tool emulates full-array behaviour without the need for repeatedly running large-scale simulations. This approximation was validated against full-array TOPAS simulations using the same pyramid layout (see Figures B.1 and B.2 from Appendix B.2.5), confirming the reliability of the approach. An illustration of how the code works and the pyramid spacing is shown in Figure 2.5.

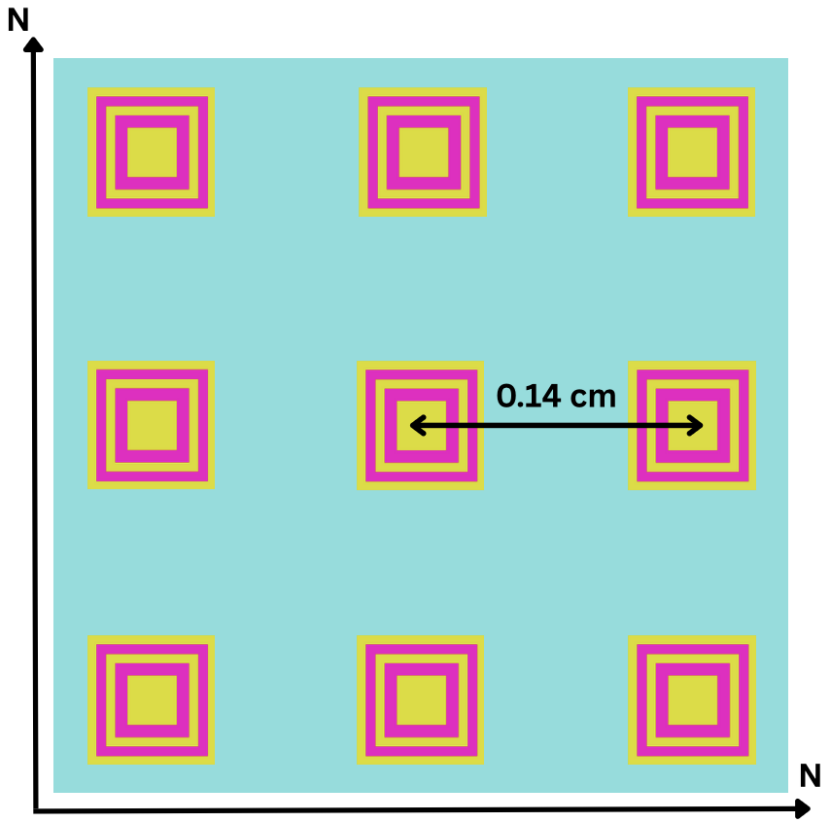

Figure 2.5: Schematic representation of translating the dose from one pyramid into an $N \times N$ array of identical pyramids. Each stack of 5 pyramid layers was spaced by 0.14 cm in both x and y directions.

3 Results & Discussion

This chapter sheds light upon the findings of the study, organised into three sections corresponding to each of the research questions, to ensure a clear and logical presentation. Firstly, in Section 3.1 it is examined how the spatial configuration of pyramid filters influences proton dose distributions and the uniformity of the SOBP. Then, in Section 3.2 the use of a collimator is analysed, with the intention of improving the beam shaping, dose conformity to the target, and reduction of out-of-field dose. Finally, in Section 3.3 it is evaluated how variations in the base layer design of the pyramid filter affect the proton beam's penetration range and modulation, especially with an eye towards treating irregularly-shaped tumours. It is worth pointing out that all results showcased are based on Monte Carlo simulations performed with a consistent number of 100 million protons per configuration, as previously described in the methodology.

3.1 Effect of Pyramid Spatial Configuration on Dose and SOBP Uniformity

In this section, results are presented for various spatial arrangements of the pyramid filter, specifically comparing the use of a single pyramid versus an array of multiple (17×17). Their impact on the uniformity of the lateral dose distribution and the SOBP in the target region was investigated by normalising all dose distributions to a common reference: the mean dose within the defined region of interest (ROI) spanning $x, y \in [-1, +1]$ cm and $z \in [10, 12]$ cm in the water phantom was scaled to 20 Gy. Normalising to the average dose in this central region was needed in order to compare effectively how uniformly the SOBP covers the target under different configurations.

To characterize the SOBP and dose profiles, the depth–dose data was analysed in two complementary ways:

1. Plane-integrated dose: obtained by summing the dose over the cross-sectional x-y plane at each depth z, where the plane spans 5.05 × 5.05 cm², discretized into 101 × 101 square voxels of 0.05 cm × 0.05 cm (essentially the total or average dose in each depth slice as a function of z).

2. Central-axis dose: extracted from the voxel located at the beam's geometrical centre (x = y = 0 cm), across all depths. The sampled region is a 0.05 × 0.05 cm² square voxel in the transverse plane, extending along the full depth of the phantom.

These two analyses highlight different aspects of uniformity. On the one hand, the slope (derivative) of the plane-integrated curve indicates how uniform the dose is across the field at a given depth, since any lateral non-uniformity will reduce the integrated dose at that depth relative to a perfectly flat field. On the other hand, the central-axis curve shows the depth–dose experienced by the central ray of the beam. Depending on the beam size and shape, the central-axis SOBP is often the highest dose path and comparing it to the plane-integrated SOBP reveals if the central region is receiving disproportionately more dose than the field average, which would signify non-uniform lateral distribution. In a narrow beam, protons that are scattered out of the beam deposit their dose at larger distance from the beam axis. At larger depth, more protons are lost due to this outscattering. On the beam axis, this reduces the dose as a function of depth.

Figure 3.1 and Figure 3.2 show the central-axis depth–dose profiles for the single pyramid (1×1) and the 17×17 pyramid array configurations, respectively. Likewise, Figure 3.3 and Figure 3.4 show the plane-integrated dose versus depth for the same two configurations. Comparing these results, it is evident that using an array of pyramids produces a more uniform SOBP across the ROI than using a single one.

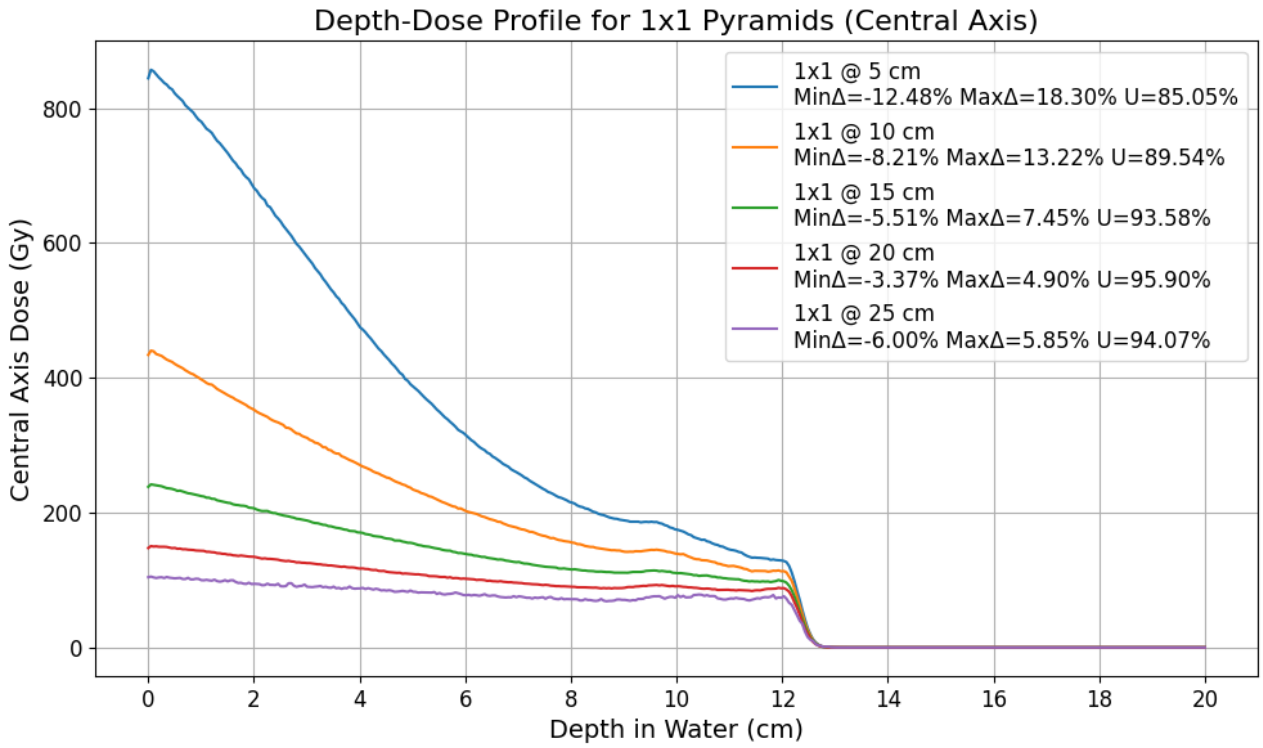

Figure 3.1: Central-axis depth–dose profile for a single pyramid (1×1) configuration.

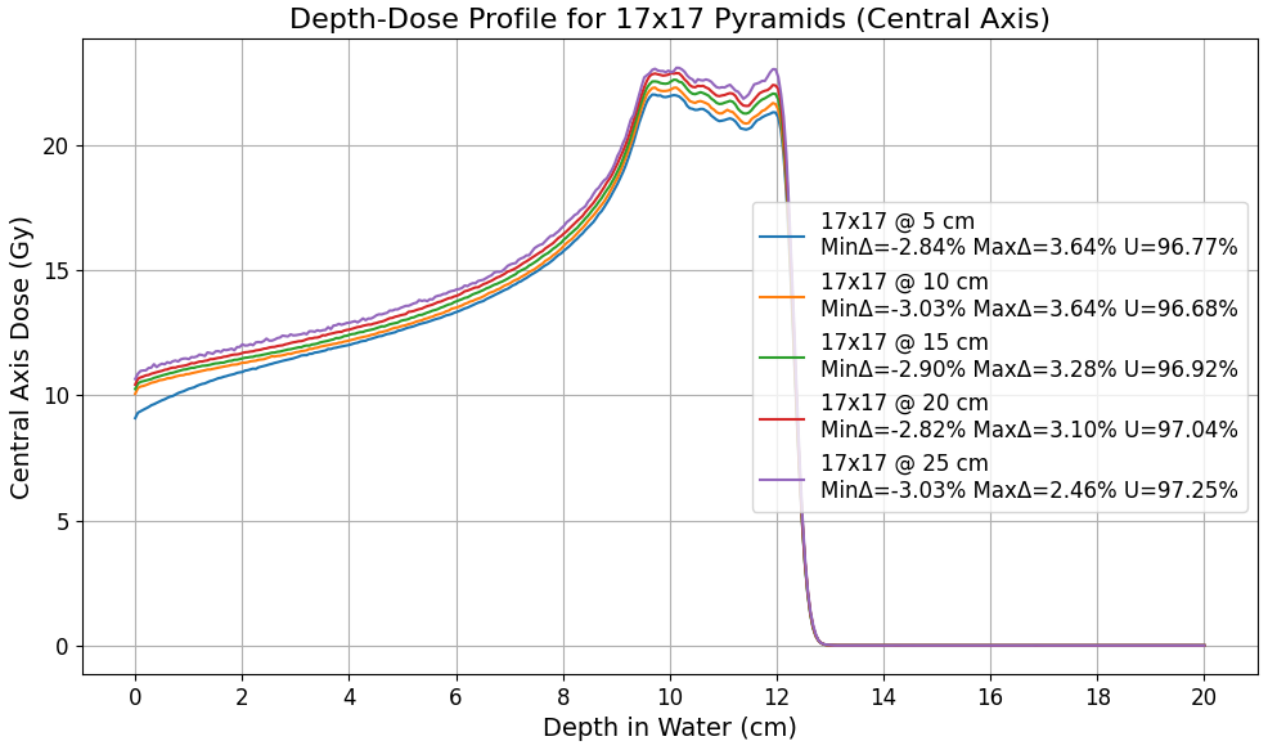

Figure 3.2: Central-axis depth–dose profile for a 17×17 pyramid array configuration.

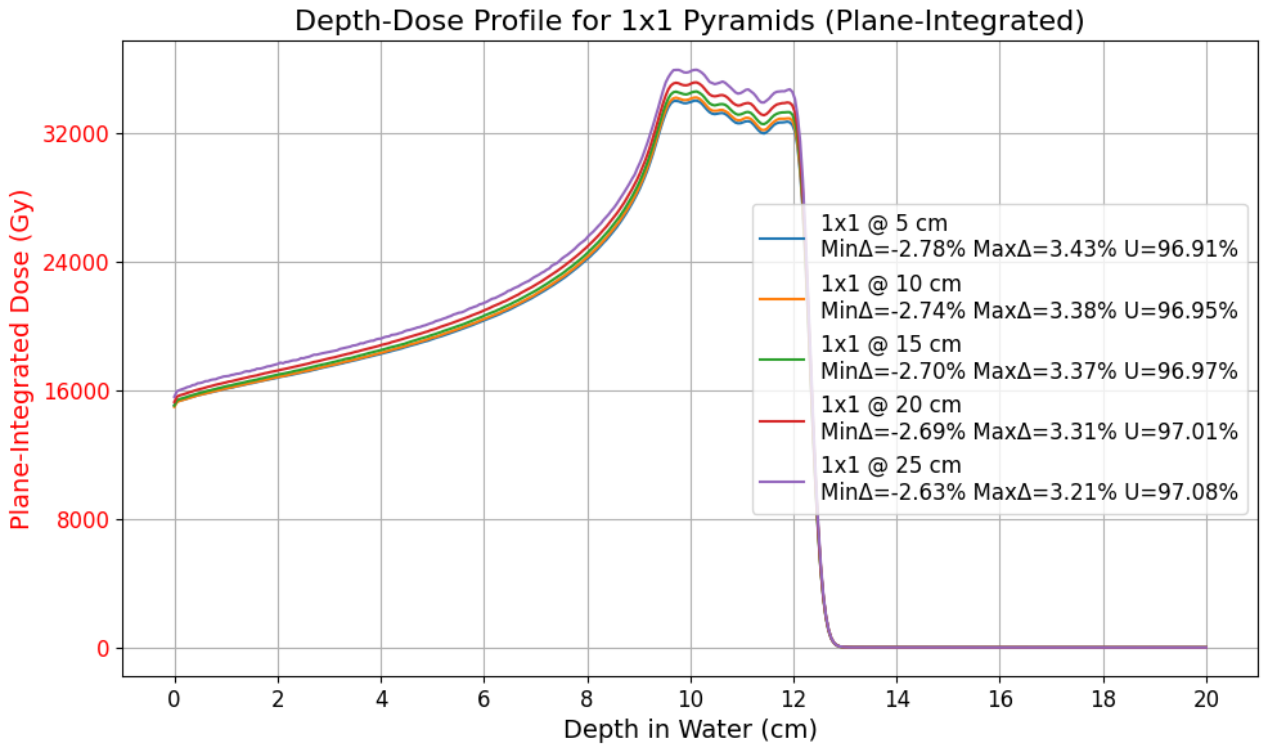

Figure 3.3: Plane-integrated depth–dose profile for a single pyramid (1×1) configuration.

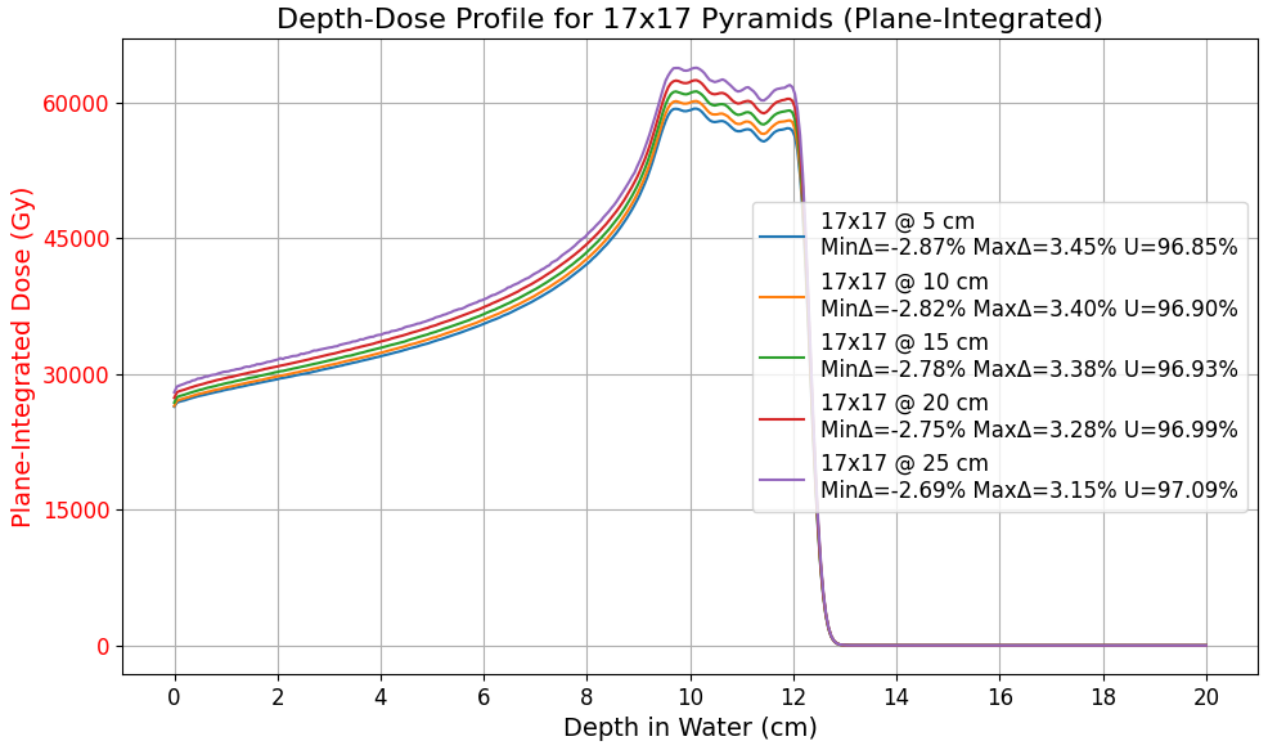

Figure 3.4: Plane-integrated depth–dose profile for a 17×17 pyramid array configuration.

In the single pyramid case (1×1), the plane-integrated curve shows a distinct and well-formed SOBP. However, the central-axis dose is less flat and slightly underperforms in the SOBP region, not in terms of absolute dose, but in uniformity across the beam plane. This is because only the centre of the beam receives the full modulation from the pyramid, while off-axis regions receive less contribution from Bragg peaks, resulting in lower dose at those positions. The statistical fluctuations inherent to Monte Carlo simulations may also exaggerate this local non-uniformity. In contrast, the 17×17 pyramid array configuration yields a flat and uniform SOBP across the field, which is substantially due to having a wider beam and less dose in the centre. The plane-integrated and central-axis curves are nearly identical in the SOBP region, both reflecting a uniform 20 Gy plateau by design of the normalisation. This means the dose at depth is distributed much more evenly laterally -the multitude of pyramids effectively superimpose many slightly offset Bragg peaks such that any point in the target, not just the centre, receives contributions from several pyramid "pixels"-. The result is that the SOBP is very uniform across the entire 2 cm field for the array of pyramids, whereas the single pyramid had a slight non-uniformity, a central hot spot with a few percent higher dose than the edges of the ROI.

Another investigated aspect of the spatial configuration was the distance between the pyramid filter and the water phantom. Increasing this distance resulted in better transversal uniformity within the SOBP region. Five spacing values were evaluated (5, 10, 15, 20 and 25 cm) using trial-and-error. Among these, the 25 cm separation produced the highest uniformity (see Figure 3.4), and was therefore selected for the remaining simulations. This outcome suggests that allowing the modulated proton beam to travel some distance in air before entering the phantom enables the lateral dose contributions from individual pyramids to blend more effectively. The result is a smoother, more uniform dose distribution, especially across the full irradiated plane. While one might assume that placing the modulator closer to the phantom would reduce beam divergence and maintain tighter lateral confinement, in this case, the increased separation allowed for better lateral overlap of dose spots, increasing overall uniformity. A disadvantage of the larger lateral spread is intended dose variations and penumbra in the lateral direction will be less sharp.

In addition to testing different setup configurations, the geometric design of the pyramid filter layers was altered to optimise the SOBP's alignment with the desired depth. Through iterative adjustment, it was found that a subtle change in the transverse cross section of the top pyramid layer had a significant effect on the SOBP range coverage. In the initial pyramid design, the proximal Bragg peak of the SOBP occurred shallower than the planned 10 cm depth, at around 8.5 cm. After reducing the top layer's base area from 0.44 mm to 0.40 mm, the weight of the shallowest Bragg peak was decreased, which induced an increase in the relative contribution of the subsequent Bragg peaks. This adjustment shifted the dose balance across the SOBP, improving the uniformity of the modulation, which remained within approximately ±3%.

3.2 Impact of Collimator Use on Beam Shaping and Dose Distribution

This section presents the effects of introducing a collimator into the experimental setup and how it influences the lateral beam shape, the dose conformity to the target area, and the dose to regions outside the ROI. The need for adding a collimator arose from the observation that even with a large pyramid array (17 × 17), the dose was not sufficiently uniform throughout the 3D target volume, with variation across the defined ROI reaching ±43% (shown in Figure 3.5), far exceeding the intended ±3% uniformity margin. This non-uniformity prompted further optimisation of the spatial modulation approach. To achieve improved dose coverage within the ROI, both the beam and pyramid array had to be enlarged, which in turn necessitated the addition of a collimator to constrain the expanded field and reduce dose spillage to surrounding regions.

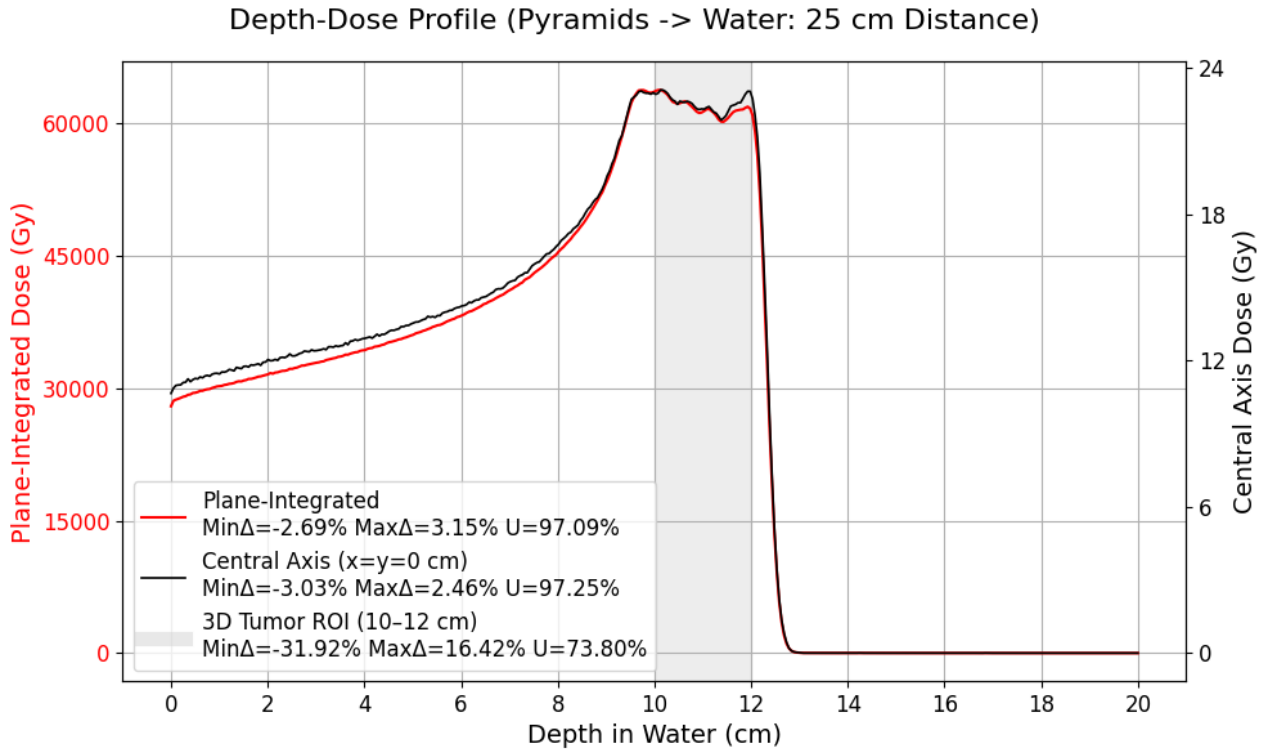

Figure 3.5: Dose-depth profile when the distance between the pyramids and water phantom is 25 cm; plane-integrated curve in red, central-axis curve in black, shaded ROI rectangle of 3D tumour region for visualisation.

To begin with, the pyramid filter array size was increased to reach a better coverage of the tumour target. Several configurations -21 × 21, 25 × 25, 27 × 27, 31 × 31- were tested, among which the 27 × 27 array of pyramids provided the flattest dose distribution across the whole ROI: ±5%. This confirmed that a broader lateral modulation can majorly improve the homogeneity, yet with the cost of a significantly enlarged beam footprint, since now the high-dose region extended well beyond the intended 2 cm field (see Figure B.2). Each 2D image shown corresponds to a single voxel-thick slice extracted from the 3D dose distribution array at the specified coordinates (e.g., XY at z = 10 cm), with a slice thickness of 0.05 cm, matching the voxel size in the z-direction. Then, a custom copper collimator was designed to constrain the lateral spread of the modulated beam to the ROI (see Figure 3.7). The thickness of the collimator was set to 3 cm: for a proton beam energy of 136.7 MeV, the corresponding projected range in copper is 20.23 g/cm^2 (NIST, 2025), which, when divided

by the material density of 8.96 g/cm^3, yields a required thickness of approximately 2.26 cm. This value was rounded up to ensure that any proton which tries to travel outside the intended aperture will be stopped or at least significantly attenuated, providing a comfortable safety margin beyond the 2.3 cm range. Consequently, the collimator aperture size was chosen based on the uncollimated beam's lateral profile, respectably an opening just large enough to fully cover the target area, while blocking as much of the stray peripheral beam as possible. After multiple iterations, the best configuration for the final design employed a 3 $\times$ 3 cm^2 aperture in the Cu block, which was sufficient to cover the tumour ROI.

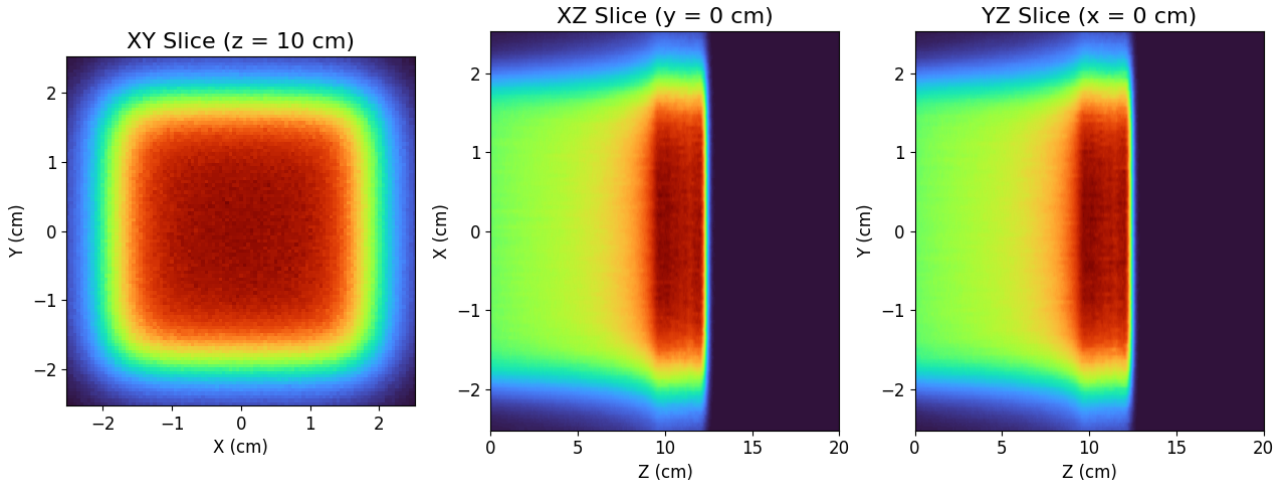

Figure 3.6: 2D dose slices for a 27 $\times$ 27 pyramid array without a collimator.

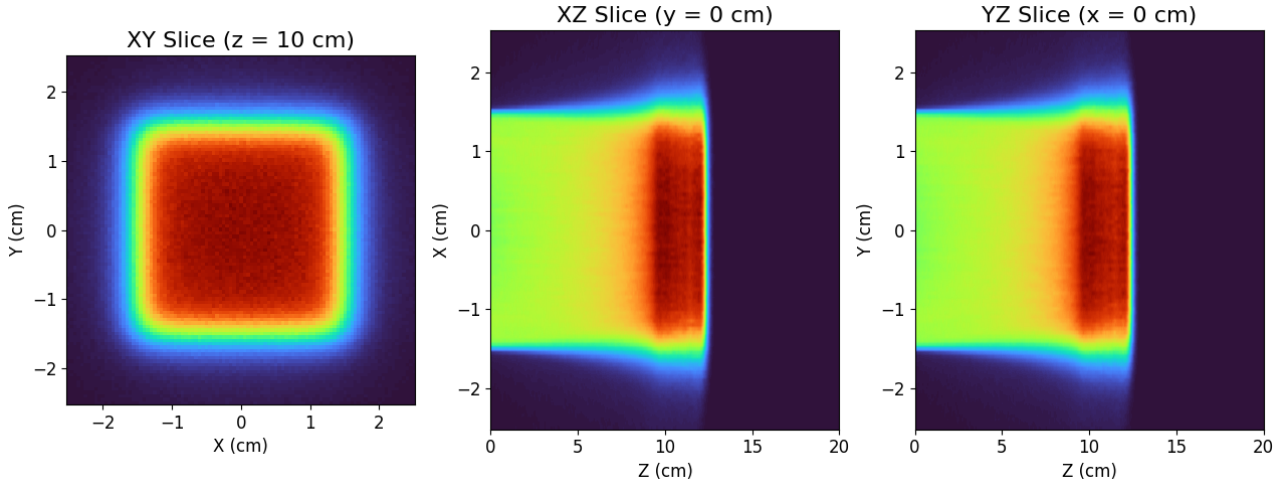

Figure 3.7: 2D dose slices for a 27 $\times$ 27 pyramid array with a 3 $\times$ 3 air-gap collimator.

The beam's depth-dose characteristics with and without the collimator were compared to confirm that the SOBP itself was not adversely affected by the collimation. Figure 3.8 overlays the central-axis depth–dose curves for an uncollimated beam versus the collimated one. As shown in this figure, the SOBP plateau remains essentially unchanged when the collimator is added, which indicates that the collimator does not distort or degrade the desired SOBP in the target, but rather the lateral distribution. Moreover, the dose uniformity within the central field remains high with the

collimator. In fact, since the uncollimated 27 × 27 array already provided good uniformity in the core (±5.5%), the collimator's presence kept this at a similar level (within about ±12.5%).

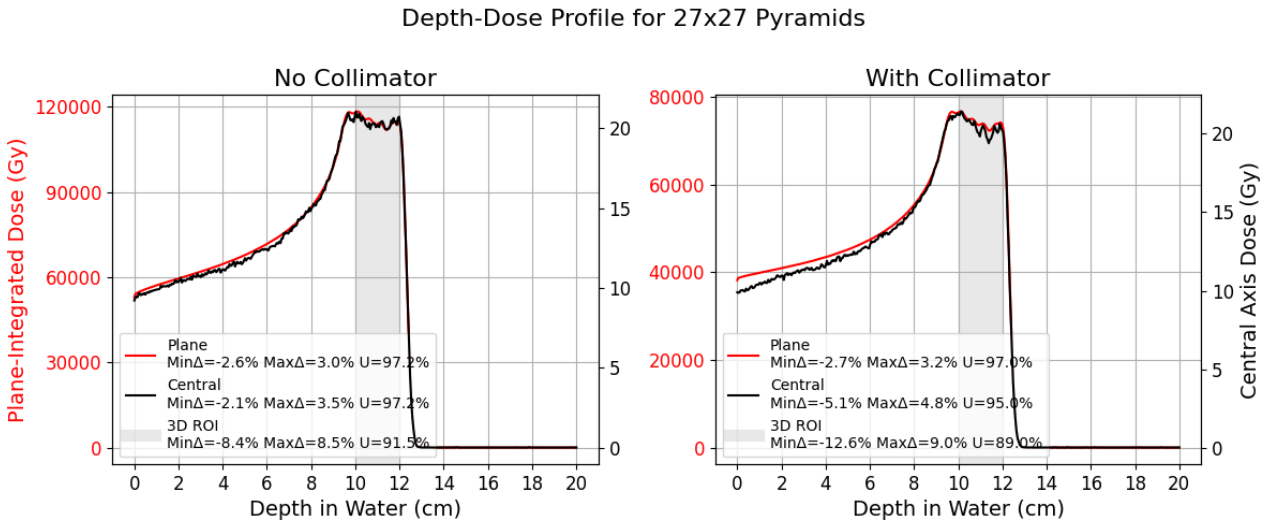

Figure 3.8: SOBP comparison of not collimated versus collimated beam (27×27 array of pyramids); plane-integrated curve in red, central-axis curve in black, shaded ROI rectangle of 3D tumour region for visualisation.

Figure 3.9 shows the horizontal and vertical dose profiles through the centre of the field at the depth of 11 cm, comparing the two scenarios: the no collimator case (black dashed curve) and the collimator one (solid red curve). Several key observations can be made. First, the uncollimated beam has a relatively gradual fall-off at about 1.5 cm from the centre, while the collimated one drops much more sharply, indicating a much tighter field.

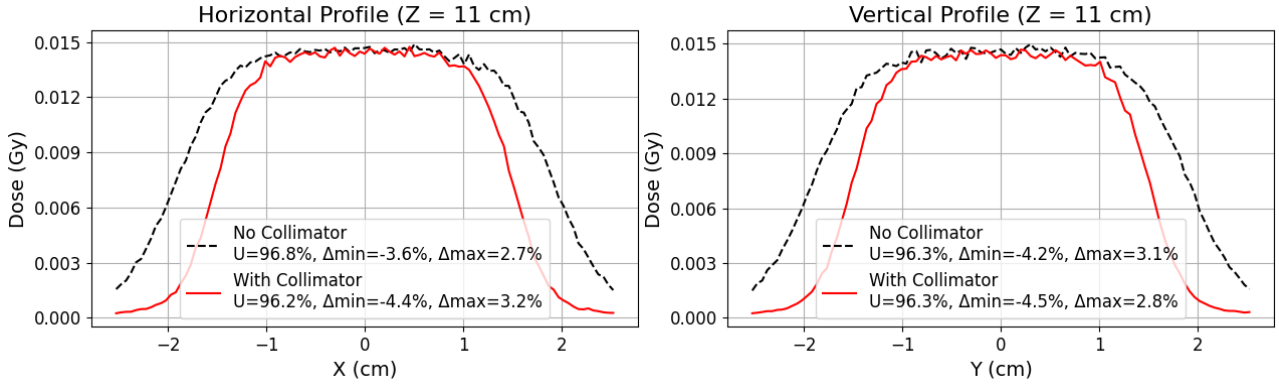

Figure 3.9: Horizontal (left plot) and vertical (right plot) line profiles at $z = 11$ cm (middle depth of ROI) comparing the transversal dose with (red) and without (black) a collimator.

3.3 Influence of Base Layer Thickness on Proton Beam. Potential for Treating an Irregularly-Shaped Tumour

The last section tackles how altering the base layer of the pyramid filter affects the proton beam's penetration range and modulation. It also briefly targets how these effects could potentially treat irregularly-shaped tumours.

The base layer is the widest, broadest layer in the pyramid filter stack, and it serves to reduce the energy of the proton beam by a roughly constant amount across the field; it sets the highest energy, the most range-reduced component of the modulated beam, thereby determining the distal end of the SOBP. Simulations were performed with several different base layer thicknesses to quantify their impact: 0.5, 1.0, 3.0, and 8.0 cm. The obtained 2D slices are visible in Figure 3.10) and were acquired using the same approach described previously, where each image represents a single voxel-thick slice extracted from the 3D dose matrix at the specified coordinates, with a slice thickness of 0.05 cm in the depth direction. In each case, the rest of the pyramid filter geometry (the pyramid layers on top of the base) and the initial beam energy spectrum were kept identical, with the beam tuned such that the 1.0 cm base case delivered the SOBP to a depth of 12 cm in the phantom.

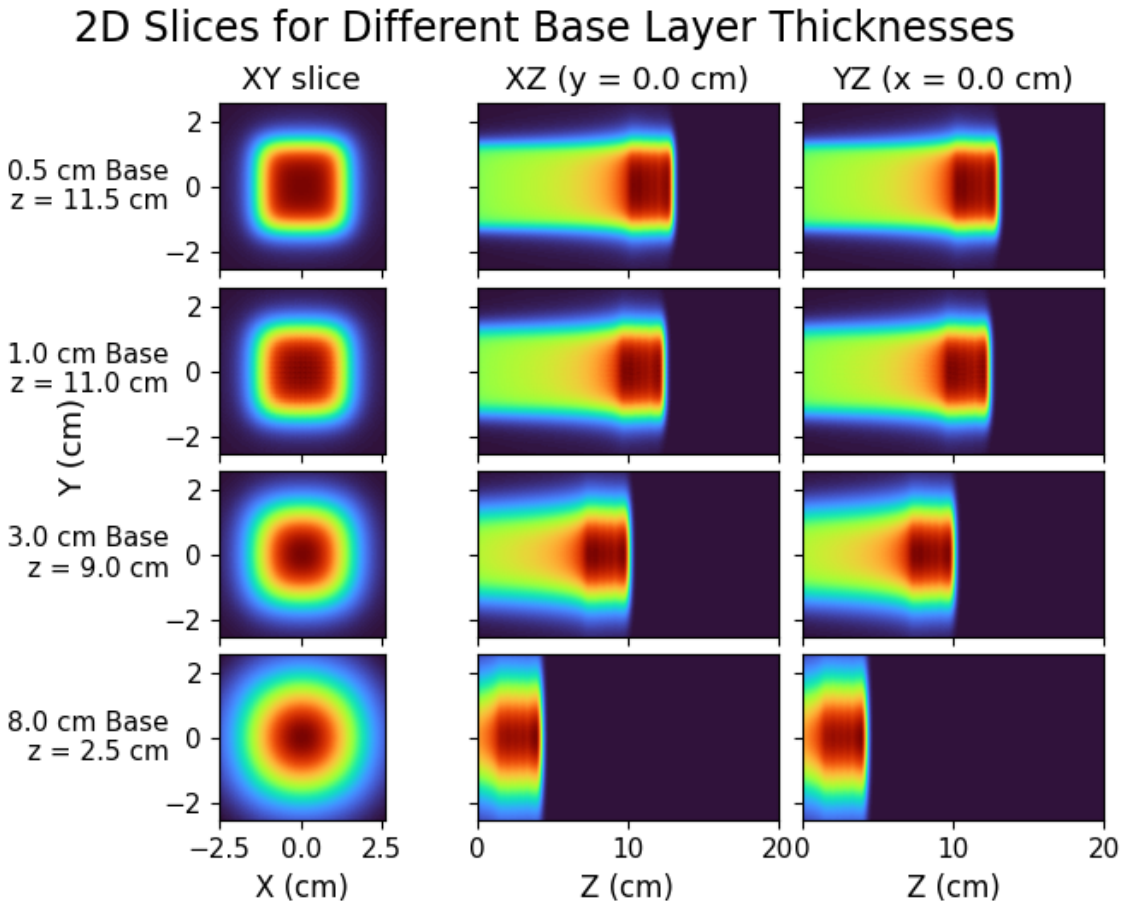

Figure 3.10: 2D dose slices for PMMA base layers of various thicknesses (0.5, 1.0, 3.0, and 8.0 cm) without the use of a collimator; XY slices are plotted at the middle depth of the corresponding SOBP.

As expected, increasing the base layer thickness causes a substantial reduction in the proton beam's penetration range. With a wider base, protons lose more energy before entering the phantom, so the resulting SOBP is shifted closer to the surface. For example, in the extreme case of an 8.0 cm

base, the range of the SOBP was dramatically shortened: distal fall-off occurred much sooner than in the 1.0 cm base scenario. Conversely, with a very thin 0.5 cm base, the beam retained higher energy and penetrated deeper, beyond the 12 cm depth mark, since less energy was removed up front. Thus, there is a clear trade-off: a thicker base yields a shallower SOBP, whereas a thinner one allows the beam to reach deeper targets.

Furthermore, a secondary effect was identified with respect to the shape of the beam. When using a much thicker base, such as the 8 cm one, without adjusting the pyramid contributions, the beam narrowed because the highest energy components were removed. This can be seen in the first column of Figure 3.10, the XY dose plots indicating that as the thickness increases, the beam shape becomes more round. Thus, if the base layer thickness influences the amplitude of the most distal Bragg peak in a SOBP, the distance between the pyramids mounted on the base layer (0.14 cm) should be adapted accordingly to reflect this shift in weight distribution.

The above findings hinted that for anatomically complex tumours that have regions of varying depth, a uniform base layer thickness may not be ideal for all parts of the target. In the case where, for example, one portion of the tumour lies deeper in the body and another one is more superficial or if part of the beam's path needs to spare a sensitive structure at depth, a compromise base thickness might undertreat one part or overtreat another. To address this challenge, another design was made in which the base layer was segmented into lateral regions of different thicknesses. By providing a spatially non-uniform base, the beam's range can be made to vary across the field, potentially matching the depth profile of a tumour with an asymmetric structure. As a preliminary demonstration, a simulation was run in which half of the field was covered by a 1.0 cm thick base section and the other half by a 3.0 cm thick base section (shown in Figure 3.11).

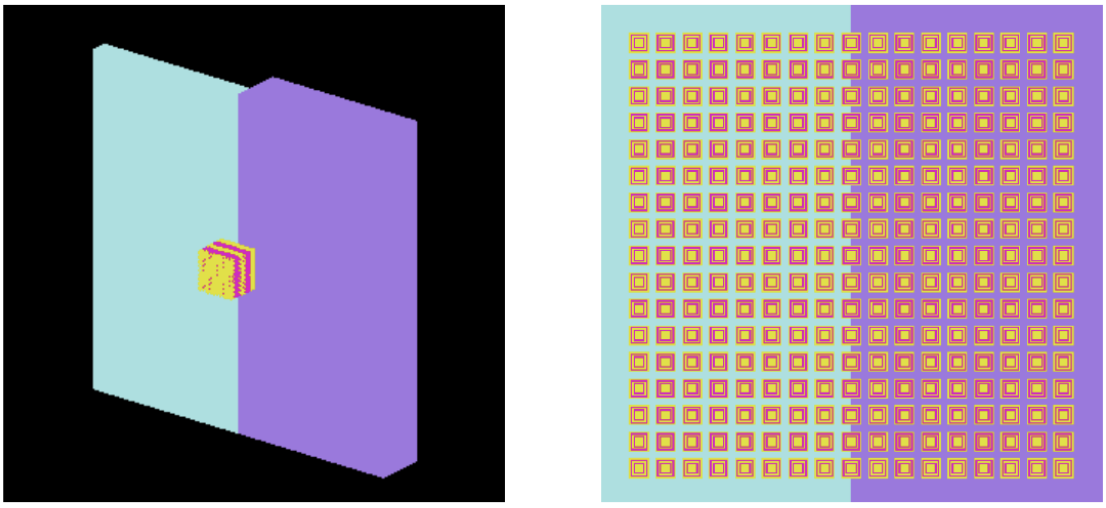

Figure 3.11: Side view (left) and top view (right) of 17 × 17 pyramid filter configuration; 1 cm thick base layer (light blue), 3 cm thick base layer (purple).

The XZ and YZ dose slices (Figure 3.12) revealed a lateral gradient in beam range: the shallower penetration on the thicker (3 cm) side transitions smoothly into deeper coverage on the thinner (1 cm) side. Meanwhile, the XY plot confirmed that the dose is laterally symmetric within each region. These qualitative observations were further supported through quantitative analysis of the plane-integrated longitudinal depth-dose curves. Figure 3.13 presents four different cases: a full 1 cm base (light blue), as well as a full 3 cm base (purple), the superposition of the two before-mentioned

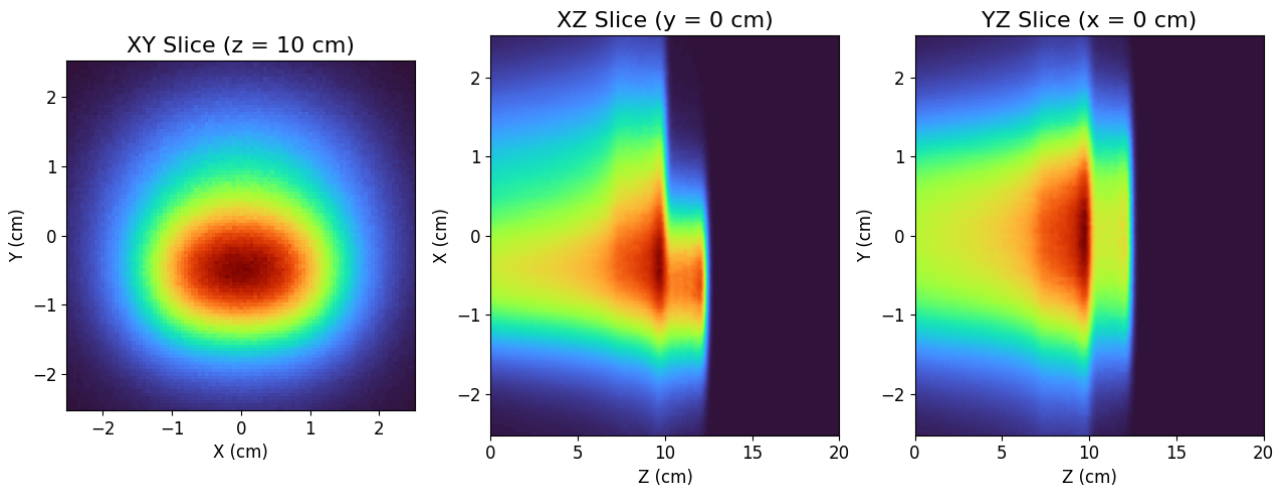

Figure 3.12: 2D dose slices for a split base design with 1 cm and 3 cm thick sections.

(gray) and the split base design (red). By comparing the theoretical expectation (dashed gray curve) to the actual depth-dose profile obtained (solid red curve), it was confirmed that the resulting SOBP is consistent with the predicted superposition, demonstrating two distinct plateaus of comparable amplitude and location. Moreover, the non-uniformity in the proximal SOBP arises from the fact that, although the initial energy of the beam is constant, it is modulated differently as it passes through the two distinct support layer thicknesses, resulting in dose components that begin depositing at different depths and locally sum to a non-uniform entrance dose.

This experiment demonstrated the feasibility of using segmented base layers to shape dose distribution not only in depth, but also laterally. Such an approach could offer better conformality when targeting irregular tumours or avoiding healthy structures with differing ranges.

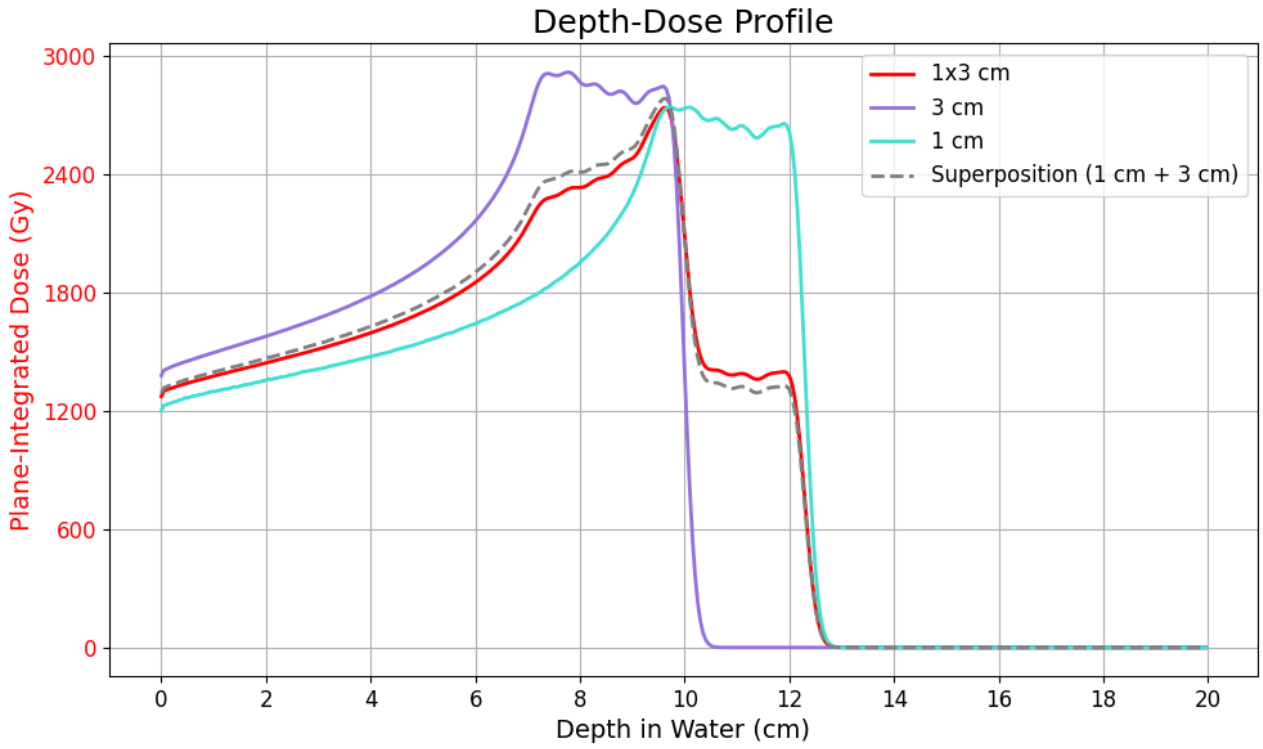

Figure 3.13: Depth–dose SOBP for the split base configuration (red), as well as for a homogeneous 1 cm base (light blue), 3 cm base (purple), and the superposition of the two (gray).

4 Conclusion

The final section of the thesis highlights the main findings of the 10-week project 4.1, as well as its limitations 4.2 and proposals for future work 4.3.

4.1 Summary of Main Contributions

For the first research question, the results showed that broad, multi-element pyramid filters produce much more uniform dose distributions than a single pyramid. A 17 $\times$ 17 pyramid array yielded a flat SOBP plateau with virtually identical central-axis and plane-integrated profiles across the 2 cm field, whereas the single-pyramid case had a pronounced central hot spot. Spacing the filter farther from the phantom further smoothed the dose. Under the optimised configuration (17×17 array, 25 cm distance), the SOBP uniformity was maintained on the order of ±3% across for both the plane-integrated and central-axis approaches.

For the second subject of inquiry, a custom 3 cm thick copper collimator with a 3 $\times$ 3 cm² aperture was introduced to trim the field of the expanded 27 $\times$ 27 array of pyramids. The SOBP depth–dose profile within the target remained essentially unchanged by collimation, showing that the range modulation was preserved. In contrast, the lateral profile became much steeper: with the collimator in place the dose fell off sharply at the aperture edge, whereas without it the uncollimated beam exhibited a gradual 1.5 cm penumbra. As a result, stray dose outside the target was effectively eliminated and the beam was tightly confined to the ROI.

For the third and last topic of investigation, varying the base layer thickness was found to directly control the beam's shape and depth deposition. Thicker bases shifted the SOBP proximally: for example, an 8.0 cm base layer brought the distal fall-off much closer to the surface compared to the 1.0 cm baseline, whereas a very thin 0.5 cm base allowed the beam to extend beyond the original 12 cm depth. This reflected a clear trade-off: thicker bases yielded shallower SOBPs, while thinner bases allowed deeper penetration. Moreover, the study demonstrated that the base can also be laterally segmented to match irregular tumours. In the presented simulation in which half of the field used a 1.0 cm base and the other half a 3.0 cm base, the resulting depth–dose distribution showed two distinct plateaus, one for each thickness, exactly as predicted. Therefore, it can be stated that spatially varying the base design can become of interest for treating anatomically complex tumours.

4.2 Limitations

4.2.1 Monte Carlo Simulation Constraints

One limitation of this study was the computational intensity of TOPAS Monte Carlo simulations. Due to time constraints, the number of proton histories had to be limited to 100 milion to keep simulation times reasonable. As a consequence, some depth-dose results could have relatively higher statistical uncertainty than desired. Additionally, the simulations were not re-run to collect detailed statistical metrics (e.g., standard deviation or variance in the scoring volume), so such quantitative uncertainty analysis is absent from the presented findings. Increasing the particle count or incorporating statistical error analysis would provide more realistic dose uniformity assessments, however this was not feasible within the project timeline.

4.2.2 Idealised Beam Model

The simulations were performed with an ideal pencil beam, a perfectly parallel, non-diverging beam as the radiation source. The main advantage of this approach is isolating the fundamental Bragg peak superposition effects without additional complications from beam divergence or lateral spreading, which is useful in early-stage research. Nevertheless, it also greatly simplifies the experimental setup.

4.2.3 Small Size of Pyramid Layers

Each pyramid layer, with the exception of the base, is approximately 0.44 cm thick, resulting in a total pyramid height of just 2.2 cm. This compact geometry, inherited from previous work, was adopted without further exploration of alternative dimensions. On the one hand, the small lateral width of the pyramids facilitates sharper lateral dose variations and allows the filter to be positioned closer to the patient, advantages particularly relevant in compact or preclinical setups. On the other hand, this design also introduces potential drawbacks. Due to increased lateral scattering from the narrow pyramids, a low-dose tail may form distal to the SOBP, which can reduce dose conformity beyond the target volume. In contrast, larger pyramids would suppress this distal tail by better containing scattered protons, but would compromise the ability to generate steep lateral dose gradients.

4.3 Future Work

4.3.1 Optimised SOBP Weight Calculation Algorithm

An essential next step is to develop an improved algorithm for determining the weights of individual Bragg peaks in the SOBP, by combining the strengths of the current method with optimisation techniques. The used Georges documentation (Tesse et al., 2023) could be augmented with a deconvolution-based approach to solve for the optimal weights. For example, a numerical deconvolution method, such as that demonstrated by Bortfeld's analytical solution, could be used to derive the weight distribution that yields a flat-top dose profile. By doing so, it could take as input the desired number of Bragg peaks and output the required weights for a uniform SOBP of a specified width. An important feature would be enforcing a monotonic decrease in weights for shallower layers, for example each subsequent Bragg peak would have a smaller or equal weight to the previous one ($W_{i+1} \leq W_i$).

4.3.2 Investigating Nylon-12 as a Modulator Material

A second important direction is to continue the exploration of Nylon-12 as an alternative to PMMA for the range-modulating filter. Nylon-12 is appealing because of its close similarity to water in terms of radiation interaction. Its density (1.03 g/cm³) and elemental composition (approximately $(CH2)11CONH$ for the monomer unit) result in attenuation and stopping power characteristics that are essentially water-equivalent within a few percent. For instance, the mass attenuation coefficient of Nylon-12 has been shown to agree with that of liquid water and PMMA to within about 3% (Ade et al., 2020). In Appendix B.2.2, the TOPAS material definition prepared for Nylon-12 can be found, where the material is defined by four constituent elements (Carbon, Hydrogen, Nitrogen, Oxygen with mass fractions 0.7301, 0.1175, 0.0710, 0.0814, respectively) corresponding to the Nylon-12 monomer's formula, a mass density of 1.03 g/cm³ (very close to water's 1.0 g/cm³), and a mean excitation energy of 75 eV (using water's mean excitation energy as an approximation). Due to time constraints, only a test simulation was run and no detailed analysis was completed, thus future work could fully utilise

this model as such: first by running dedicated TOPAS simulations to produce SOBPs with a Nylon-12 modulator, and then by comparing the dose uniformity and beam attenuation results against the baseline PMMA case. This comparison is relevant because although Nylon-12 is easier to 3D print than PMMA (Buyenne, 2025), it has to be checked whether or not it can achieve equal or even superior SOBP flatness, and whether any adjustments in layer weights are needed due to its slightly different stopping-power ratio.

Bibliography

Ade, N., van Eeden, D., and du Plessis, F. (2020). Characterization of Nylon-12 as a water-equivalent solid phantom material for dosimetric measurements in therapeutic photon and electron beams. *Applied Radiation and Isotopes*, 155. DOI: https://doi.org/10.1016/j.apradiso.2019.108919.

Agostinelli, S., Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce, P., Asai, M., Axen, D., and ... Zschiesche, D. (2003). Geant4—a simulation toolkit. *Nucl. Instr. Methods Phys. Res.*, 506(3):250–303. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8.

Akagi, T., Higashi, A., Tsugami, H., Sakamoto, H., Masuda, Y., and Hishikawa, Y. (2003). Ridge filter design for proton therapy at Hyogo Ion Beam Medical Center. *Phys Med Biol.*, 48(22):301–312. DOI: https://doi.org/10.1088/0031-9155/48/22/n01.

Auberger, T., Griesmayer, E., and Pleško, M. (2004). Das Projekt MedAustron: Designstudie (Fotec - Forschungs - und Technologietransfer GmbH, Wr. Neustadt. ISBN: 3-200-00141-0.

Bethe, H. (1930). Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie. *Ann Phys*, 397(3):325–400. DOI: https://doi.org/10.1002/andp.19303970303.

Bethe, H. (1953). Molière's theory of multiple scattering. *Phys. Rev.*, 89(6):1256–1266. DOI: https://doi.org/10.1103/PhysRev.89.1256.

Bloch, F. (1933). Zur Bremsung rasch bewegter Teilchen beim Durchgang durch Materie. *Ann. d. Phys.*, 408(3):285–320. DOI: https://doi.org/10.1002/andp.19334080303.

Blog Buyenne (2025). Comparison of PMMA vs. Nylon (PA-12, PA-6, PA-66, and Blends) for 3D Printing. URL: https://blog.buyenne.com/?p=221848.

Bortfeld, T. (1997). An analytical approximation of the Bragg curve for therapeutic proton beams. *Med. Phys.*, 24(12):2024–2033. DOI: https://doi.org/10.1118/1.598116.

Das, I., Paganetti, H., and Schippers, M. (2015). Principles and Practice of Proton Beam Therapy. URL: https://www.medicalphysics.org/SimpleCMS.php?content=tablecontents.php&isbn=9781936366446.

Deycmar, S., Faccin, E., Kazimova, T., Knobel, P., Telarovic, I., Tschanz, F., Waller, V., Winkler, R., Yong, C., Zingariello, D., and Pruschy, M. (2020). The relative biological effectiveness of proton irradiation in dependence of DNA damage repair. *British Institute of Radiology*, 93. DOI: https://doi.org/10.1259/bjr.20190494.

Faddegon, B., Ramos-Méndez, J., Schuemann, J., McNamara, A., Shin, J., Perl, J., and Paganetti, H. (2020). The TOPAS tool for particle simulation, a Monte Carlo simulation tool for physics, biology and clinical research. *Physica Medica*, pages 114–121. DOI: https://doi.org/10.1016/j.ejmp.2020.03.019.

Failla, G. and Henshaw, P. (1931). The Relative Biological Effectiveness of X-rays and Gamma Rays. *RSNA*, 17(1). DOI: https://doi.org/10.1148/17.1.1.

Favaudon, V., Caplier, L., Monceau, V., Pouzoulet, F., Sayarath, M., Fouillade, C., Poupon, M., Brito, I., Hupé, P., Bourhis, J., Hall, J., Fontaine, J., and Vozenin, M. (2014). Ultrahigh dose-rate FLASH irradiation increases the differential response between normal and tumor tissue in mice. *Science Translational Medicine*, 6(245):245–293. DOI: https://doi.org/10.1126/scitranslmed.3008973.

FitzGerald, T.J. (2024). Proton Therapy - Scientific Questions and Future Direction. DOI: https://10.5772/intechopen.111250.

Graeff, C., Volz, L., and Durante, M. (2023). Emerging technologies for cancer therapy using accelerated particles. *Progress in Particle and Nuclear Physics*, 131. DOI: https://doi.org/10.1016/j.ppnp.2023.104046.

Hazem, R. (2023). Interaction of Proton Beam with Human Tissues in Proton Therapy. *Proton Therapy - Scientific Questions and Future Direction*. DOI: https://10.5772/intechopen.1003186.

Health Council of the Netherlands (2009). Proton radiotherapy. Horizon scanning report. URL: https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.healthcouncil.nl/binaries/healthcouncil/documenten/advisory-reports/2009/12/11/proton-radiotherapy/advisory-report-proton-radiotherapy.pdf&ved=2ahUKEwitqe2jnoKNAxU71AIHHd9CLi4QFnoECBkQAQ&usg=AOvVaw3IQKzk7WuGp8FiMgoV0_Yf.

Highland, V. (1975). Some practical remarks on multiple scattering*. *Nucl. Instr. Methods Phys. Res.*, 129:497–499. DOI: https://10.5772/intechopen.1003186.

Huang, R. and Zhou, P. (2020). DNA damage response signaling pathways and targets for radiotherapy sensitization in cancer. *Sig Transduct Target Ther*, 5(60). DOI: https://doi.org/10.1038/s41392-020-0150-x.

ICRU Report 85 (2011). Fundamental Quantities and units for ionizing radiation (Revised). *Journal of ICRU*, 11:27. DOI: https://doi.org/10.1093/jicru/ndr010.

Jolly, S., Owen, H., Schippers, M., and Welsch, C. (2020). Technical challenges for FLASH proton therapy. *Physica Medica: European Journal of Medical Physics*, 78:71–82. DOI: https://doi.org/10.1016/j.ejmp.2020.08.005.

Lopez Llorens, L. (2024). Design of a Pyramid Filter by Performing a Comparative Analysis of Spread-Out Bragg Peaks Using Analytical Models and GEANT-4 TOPAS. URI: https://fse.studenttheses.ub.rug.nl/id/eprint/33302.

NIST (2025). PSTAR: Stopping Power and Range Tables for Protons. Copper. URL: https://physics.nist.gov/cgi-bin/Star/ap_table.pl.

Northwestern Medicine (2025). How Does Proton Therapy Work? URL: https://www.nm.org/conditions-and-care-areas/cancer/proton-therapy/how-does-proton-therapy-work.

Olivari, F. (2024). Quantitative proton radiography with a single-2D-detector system: a feasibility study. *Thesis fully internal (DIV), University of Groningen]*, pages 15–17. DOI: https://doi.org/10.33612/diss.1048400629.

Paganetti, H. (2012). Proton Therapy Physics.

Patient Resource (2024). Radiation Therapy. URL: https://www.patientresource.com/Radiation_Therapy_Options.

Perl, J., Shin, J., Schumann, J., Faddegon, B., and Paganetti, H. (2012). TOPAS: An innovative proton Monte Carlo platform for research and clinical applications. *Medical Physics*, 39(11):6818–6837. DOI: https://doi.org/10.1118/1.4758060.

Press, R. and Mehta, M. (2024). Proton Therapy: Current Status and Controversies. *JCO Oncology Practice*, 20(6)):747–749. DOI: https://doi.org/10.1200/OP.24.00132.

Roomi, M. (2023). Radiation Therapy for Cancer Treatment. *Journal of Otolaryngology Research & Reports*, 2(4)):1–10. DOI: https://doi.org/10.47363/JOLRR/2023(2)116.

Simeonov, Y., Weber, U., Penchev, P., Ringbæk, T., Schuy, C., Brons, S., Engenhart-Cabillic, R., Bliedtner, J., and Zink, K. (2017). 3D range-modulator for scanned particle therapy: development, Monte Carlo simulations and experimental evaluation. *Phys. Med. Biol.*, 62(17)):7075–7096. DOI: https://doi.org/10.1088/1361-6560/aa81f4.

Tesse, R., Hernalsteens, C., Gnacadja, E., Pauly, N., Ramoisiaux, E., and Vanwelde, M. (2023). Georges: A modular Python library for seamless beam dynamics simulations and optimization. *SoftwareX*, 24:101579. DOI: https://doi.org/10.1016/j.softx.2023.101579.

Vincenzo, V., Boldrini, L., Mariani, S., and Massaccesi, M. (2020). Role of radiation oncology in modern multidisciplinary cancer treatment. *Molecular Oncology*, 14(7):1431–1441. DOI: https://doi.org/10.1002/1878-0261.12712.

W.H., B. and Kleeman, R. (1905). XXXIX. On the particles of radium, and their loss of range in passing through various atoms and molecules. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 10:318–340. DOI: https://doi.org/10.1080/14786440509463378.

Wilson, R. (1946). Radiological Use of Fast Protons. 47(5):487–491. DOI: https://doi.org/10.1148/47.5.487.

Zhang, R. and Newhauser, W. (2009). Calculation of water equivalent thickness of materials of arbitrary density, elemental composition and thickness in proton beam irradiation. *Phys Med Biol*, 54(6):1383–95. DOI: https://doi.org/doi:10.1088/0031-9155/54/6/001.

Appendices

This section compiles all supplementary materials supporting the findings of this thesis. These include a user guide detailing how to run TOPAS simulations on Hábrók (Appendix A), as well as the Python codes used to compute the SOBP weight factors (Appendix B.1), generate the pyramid geometry B.2.1, B.2.2, B.2.3), design a collimator B.2.4, and translate the dose from one pyramid into a specific grid size B.2.5, all analysing dose distribution data. Furthermore, the signed AI declaration according to the Biomedical Engineering BSc Project Guidelines can be found in Appendix C.

A How to Run TOPAS on Hábrók (for Mac and Linux Users)

Listing 1: *Hábrók User Guide.*

```
##############################################
  HOW TO RUN TOPAS ON HABROK
##############################################

Step 1: Open Terminal on your Mac and connect to Habrok
-------------------------------------------------------
ssh username@login1.hb.hpc.rug.nl

Step 2: (Only if needed) Transfer your simulation and G4Data folder to Habrok
-------------------------------------------------------------
scp -r /path username@login1.hb.hpc.rug.nl:~
scp -r /path/G4Data username@login1.hb.hpc.rug.nl:~

Step 3: Go to your simulation folder on the cluster
-------------------------------------------------------------
cd ~/path

Step 4: Ensure your run_topas.sh file contains SLURM + TOPAS setup
-------------------------------------------------------------
Access file:
nano run_topas.sh

Input (adjust numerical values according to own needs):

#!/bin/bash
#SBATCH --time=0:10:00 # Max runtime
#SBATCH --nodes=1 # Number of nodes
#SBATCH --ntasks-per-node=32 # Number of CPU tasks (parallel processing)
#SBATCH --mem-per-cpu=2G # Memory per task
#SBATCH --job-name=test_topas # Job name (seen in squeue)
#SBATCH --output=slurm-%j.out # Output file (%j = JobID)
#SBATCH --error=slurm-%j.err # Error file (%j = JobID)

module load TOPAS
module load Geant4-data

# Set the path to the Geant4 data directory
export TOPAS_G4_DATA_DIR=$G4INCLDATA/..

# Run TOPAS
topas main.txt
```

```
This script contains both the SLURM job configuration and the commands to run TOPAS.
It must be submitted using `sbatch`, not run directly.

Step 5: Submit the job
-----------------------------------------------------------
sbatch run_topas.sh

It will respond with:
Submitted batch job <jobid>

Step 6: Monitor the job
-----------------------------------------------------------
squeue -u username

Once the job finishes, it will disappear from the list.

Step 7: View the output
-----------------------------------------------------------
Output will be in the file named:
slurm-<jobid>.out

For example:
less slurm-17600317.out

If you want to see it live as it is being written:
tail -f slurm-17600317.out

Step 8: Download results to your Mac (if needed)
-----------------------------------------------------------
From your Mac terminal:
scp -r username@login1.hb.hpc.rug.nl:~/path .
```

B Python Codes

B.1 SOBP Weight Factors

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
from scipy.interpolate import interp1d
from scipy.optimize import Bounds, minimize
from georges.ptw import BraggPeakAnalysis

# Constants
p = 1.772
alpha = 0.0022
E_0 = 129.5    # MeV
R_initial = alpha * (E_0 ** p)
z = np.linspace(0, 40, 3999)  # Depth in cm
sigma = np.sqrt(((0.012 * (R_initial) ** 0.935) ** 2) + (((0.01 * E_0) ** 2) * (
    alpha ** 2) * (p ** 2) * (E_0 ** ((2 * p) - 2))))

primary_fluence = 10 ** 5
```

```
mass_density = 1  # in g/cm^3 of water
gamma = 0.6  # unit = 1
beta = 0.012  # cm^-1
e = 0.01

# Bragg peak depths from 10 to 12 cm in 0.5 cm steps
R_values = [R_initial] + [R_initial - i * 0.5 for i in range(1, 5)]

D_parabolic_func1_array = []
D_parabolic_func2_array = []

def compute_dose(primary_fluence, R_initial, sigma, e):
    z = np.arange(0, 39.99, 0.01)
    energy_deposits = []
    for i in range(len(z)):
        energy_deposit = calculate_D(z[i], R_initial, sigma, D, primary_fluence,
    e)
        energy_deposits.append(energy_deposit)
    return energy_deposits

def calculate_D(z, R_initial, sigma, D, primary_fluence, e):
    if z <= (R_initial + (5 * sigma)):
        z1 = -(R_initial - z) / sigma
        D_parabolic_func1 = sp.special.pbdv(-(1 / p), z1)[0]
        D_parabolic_func2 = sp.special.pbdv(-(1 / p) - 1, z1)[0]
        gamma_function = sp.special.gamma(1 / p, out=None)
        D_parabolic_func1_array.append(D_parabolic_func1)
        D_parabolic_func2_array.append(D_parabolic_func2)
        return D(primary_fluence, R_initial, sigma, z, D_parabolic_func1,
   D_parabolic_func2, e, gamma_function, mass_density, alpha, beta)
    else:
        return 0

def D(primary_fluence, R_initial, sigma, z, D_parabolic_func1, D_parabolic_func2
   , e, gamma_function, mass_density, alpha, beta):
    z1 = -(R_initial - z) / sigma
    return ((primary_fluence * np.exp(-(z1 ** 2 / 4)) * sigma ** (1 / p) *
   gamma_function) / (np.sqrt(2 * np.pi) * mass_density * p * (alpha ** (1 / p))
    * (1 + (beta * R_initial)))) * (((1 / sigma) * D_parabolic_func1) + (((beta
   / p) + (alpha * beta) + (e / R_initial)) * D_parabolic_func2))

# Compute dose deposits for each depth
dose_deposits_scaled = []
for depth in R_values:
    E_0 = (depth / alpha) ** (1 / p)
    R0 = depth
    dose_deposits = compute_dose(primary_fluence, R0, sigma, e)
    max_dose = max(np.max(dose) for dose in dose_deposits)
    dose_deposits_scaled.append([dose / max_dose for dose in dose_deposits])

# Convert to DataFrame
dose_data_df = pd.DataFrame(dose_deposits_scaled)

# Define analysis class
class SpreadOutBraggPeakAnalysis:
    def __init__(self, dose_data, z_axis, method="scipy.optimize", color="red",
   str_on_legend="", adjust_last_peak=1.0):
```

```
68          self.dose_data = dose_data
69          self.z_axis = z_axis
70          self.method = method
71          self.color = color
72          self.str_on_legend = str_on_legend
73          self.adjust_last_peak = adjust_last_peak
74
75      def get_library_max_ranges(self):
76          max_ranges = np.zeros(self.dose_data.shape[0])
77          for i in range(self.dose_data.shape[0]):
78              normalized_bragg_peak = 1e2 * self.dose_data.iloc[i, :] / np.max(
    self.dose_data.iloc[i, :])
79              bp_analysis = BraggPeakAnalysis(
80                  bp=pd.DataFrame({"centers": self.z_axis, "dose":
    normalized_bragg_peak}),
81                  method="scipy.optimize",
82              )
83              max_ranges[i] = bp_analysis.get_xrange(100)
84          return max_ranges
85
86      def sobp_data(self):
87          sobp_data = np.zeros((self.dose_data.shape[0], self.dose_data.shape[0]))
88          max_ranges = self.get_library_max_ranges()
89          for i in range(self.dose_data.shape[0]):
90              for j in range(self.dose_data.shape[0]):
91                  bp_analysis = BraggPeakAnalysis(
92                      bp=pd.DataFrame({"z": self.z_axis, "dose": self.dose_data.
    iloc[j, :]}),
93                      method="scipy.optimize",
94                  )
95                  f = interp1d(bp_analysis.data["z"].values, bp_analysis.data["
    dose"].values, kind=2, bounds_error=False)
96                  sobp_data[i, j] = f(max_ranges[i])
97          return sobp_data
98
99      def compute_weights(self):
100         a_matrix = self.sobp_data() / self.sobp_data().max()
101         goal_dose_values = np.full(a_matrix.shape[0], 1)
102         b = goal_dose_values
103         n = len(b)
104         fun = lambda x: np.linalg.norm(np.dot(a_matrix, x) - b)
105         sol = minimize(fun, np.zeros(n), method="L-BFGS-B", bounds=[(0.0, None)]
     * n)
106         return sol["x"]
107
108     def get_final_weights(self):
109         weights = self.compute_weights()
110         weights[0] *= self.adjust_last_peak
111         return weights
112
113     def compute_sobp_profile(self):
114         weights = self.get_final_weights().reshape(1, self.dose_data.shape[0])
115         weighted_dose_data = np.matmul(weights, self.dose_data.values)
116         sobp_profile = weighted_dose_data.reshape(self.dose_data.shape[1])
117         return 1e2 * sobp_profile / sobp_profile.max()
118
119     def view_sobp(self, with_pristine_peaks=False):
```

```
        fig, ax = plt.subplots()
        if with_pristine_peaks:
            weights = self.compute_weights()
            weighted_dose = self.dose_data.multiply(weights, axis=0)
            for index, row in weighted_dose.iterrows():
                ax.plot(self.z_axis, 1e2 * row.values / self.sobp_data().max(), linestyle="dashed", linewidth=0.7, color="black", marker="*", markersize=4, label="Bragg Peaks" if index == 0 else None)
        ax.plot(self.z_axis, self.compute_sobp_profile(), linestyle="dashed", linewidth=0.8, color=self.color, marker="*", markersize=5, label="SOBP")
        ax.set_title(f"SOBP by Superposition of {self.dose_data.shape[0]} Bragg peaks")
        ax.set_xlabel("Depth in Water (cm)")
        ax.set_ylabel("Radiation Dose (%)")
        ax.legend()
        ax.set_xlim(0, np.max(self.z_axis))
        plt.show()

sobp_analysis = SpreadOutBraggPeakAnalysis(dose_data=dose_data_df, z_axis=z, method="scipy.optimize", color="red")
sobp_analysis.view_sobp(with_pristine_peaks=True)

weights = sobp_analysis.compute_weights()
print("Weights:", ', '.join([f"{w:.6f}" for w in weights]))
sobp = sobp_analysis.get_library_max_ranges()
print("Peak Ranges (cm):", ', '.join([f"{S:.6f}" for S in sobp]))
```

Listing 2: *Python Code for determining the SOBP weights using the Georges library.)*

B.2 Pyramid Filter Geometry

B.2.1 PMMA Pyramids

```
import math

def compute_X_from_W(weights):
    X = [0] * 6
    X[5] = math.sqrt(weights[5])
    for i in range(4, -1, -1):
        X[i] = math.sqrt(weights[i] + X[i + 1] ** 2)
    return X

def compute_half_values(weights):
    return [w / 2 for w in weights]

# ========== USER CONFIGURATION ==========
desired_distance_to_phantom = 25.0  # cm     Change distance between phantom and pyramids here
phantom_front_z = 10.0              # cm
base_layer_thickness = 1.0          # cm     Change thickness of base layer here
# ========================================

# Geometry setup
pyramid_HLZ = 2.1585  # mm
pyramid_layer_thickness = 2 * pyramid_HLZ / 10  # cm

# Compute offset to align base top with bottom of first pyramid layer
```

```
base_bottom_z = phantom_front_z + desired_distance_to_phantom
base_layer_center = base_bottom_z + base_layer_thickness / 2
base_layer_top = base_layer_center + base_layer_thickness / 2

# Input weights (for 2cm tumour at 12cm)
weights = [0.918003, 0.295703, 0.247931, 0.204536, 0.156234, 0.197178]
X = compute_X_from_W(weights)
X.remove(max(X))
X_div_2 = compute_half_values(X)

# Constants
distance = 1.4039055  # mm
num_ridge_groups = 1
rows, cols = 1, 1
center_row, center_col = 0, 0

zero_x_groups = [0 + i * cols + 1 for i in range(rows)]
zero_y_groups = list(range(center_row * cols + 1, center_row * cols + cols + 1))

# Begin writing geometry
content = f"""# PMMA base layer ({base_layer_thickness} cm)
#######################################

s:Ge/RidgeGroup/Parent = "World"
s:Ge/RidgeGroup/Type   = "Group"
d:Ge/RidgeGroup/RotX   = 0.0 deg
d:Ge/RidgeGroup/RotY   = 0.0 deg
d:Ge/RidgeGroup/RotZ   = 0.0 deg
d:Ge/RidgeGroup/TransX = 0.0 mm
d:Ge/RidgeGroup/TransY = 0.0 mm
d:Ge/RidgeGroup/TransZ = 0.0 cm

s:Ge/Sq1/Parent       = "RidgeGroup"
s:Ge/Sq1/Type         = "TsBox"
s:Ge/Sq1/Material     = "PMMA"
s:Ge/Sq1/Color        = "lightblue"
s:Ge/Sq1/DrawingStyle = "Solid"
d:Ge/Sq1/HLX          = 10.0 cm
d:Ge/Sq1/HLY          = 10.0 cm
d:Ge/Sq1/HLZ          = {base_layer_thickness / 2:.4f} cm
d:Ge/Sq1/TransZ       = {base_layer_center:.5f} cm
"""

for group_number in range(1, num_ridge_groups + 1):
    row = (group_number - 1) // cols
    col = (group_number - 1) % cols

    if group_number in zero_x_groups:
        transX = 0
        transY = (row - center_row) * distance
    elif group_number in zero_y_groups:
        transX = (col - center_col) * distance
        transY = 0
    else:
        transX = (col - center_col) * distance
        transY = (row - center_row) * distance

```

```
    content += f"""
# RidgeFilter Group {group_number}
#####################################

s:Ge/RidgeGroup{group_number}/Parent = "World"
s:Ge/RidgeGroup{group_number}/Type   = "Group"
d:Ge/RidgeGroup{group_number}/RotX   = 0.0 deg
d:Ge/RidgeGroup{group_number}/RotY   = 0.0 deg
d:Ge/RidgeGroup{group_number}/RotZ   = 0.0 deg
d:Ge/RidgeGroup{group_number}/TransX = {round(transX, 5)} mm
d:Ge/RidgeGroup{group_number}/TransY = {round(transY, 5)} mm
d:Ge/RidgeGroup{group_number}/TransZ = 0.0 cm

# Instantiate squares for RidgeGroup {group_number}
"""

    for sq_number in range(1, 6):
        if sq_number == 5:  # Comment this if-cond to not have it pre-defined
            hlx = 0.20  # mm
            hly = 0.20  # mm
        else:
            hlx = X_div_2[(sq_number - 1) % len(X_div_2)]
            hly = hlx

        colour = "yellow" if (sq_number % 2) != 0 else "pink"
        transZ = base_layer_top + pyramid_layer_thickness / 2 + (sq_number - 1)
    * pyramid_layer_thickness

        content += f"""
s:Ge/Sq{sq_number}.{group_number}/Parent       = "RidgeGroup{group_number}"
s:Ge/Sq{sq_number}.{group_number}/Type         = "TsBox"
s:Ge/Sq{sq_number}.{group_number}/Material     = "PMMA"
s:Ge/Sq{sq_number}.{group_number}/Color        = "{colour}"
s:Ge/Sq{sq_number}.{group_number}/DrawingStyle = "Solid"
d:Ge/Sq{sq_number}.{group_number}/HLX          = {hlx} mm
d:Ge/Sq{sq_number}.{group_number}/HLY          = {hly} mm
d:Ge/Sq{sq_number}.{group_number}/HLZ          = {pyramid_HLZ} mm
d:Ge/Sq{sq_number}.{group_number}/TransZ       = {transZ:.5f} cm
"""

# Save to file
output_filename = "pyramid_dimensions_PMMA.txt"
with open(output_filename, "w") as file:
    file.write(content)

print(f"File '{output_filename}' has been successfully written.")
```

Listing 3: Python code for determining the PMMA pyramid layers geometry.

B.2.2 Nylon-12 Pyramids

```
import math

def compute_X_from_W(weights):
    X = [0] * 6
    X[5] = math.sqrt(weights[5])
    for i in range(4, -1, -1):
```

```
        X[i] = math.sqrt(weights[i] + X[i + 1] ** 2)
    return X

def compute_half_values(weights):
    return [w / 2 for w in weights]

# ========== USER CONFIGURATION ==========
desired_distance_to_phantom = 25.0  # cm     Distance between phantom and pyramids
phantom_front_z = 10.0              # cm     Phantom front face Z-position
base_layer_thickness = 1.0          # cm     Base thickness
base_material = "Nylon12"           # Change material name here
pyramid_HLZ = 2.4272                # mm     Pyramid HLZ for Nylon-12
# ========================================

# Derived constants
pyramid_layer_thickness = 2 * pyramid_HLZ / 10  # cm

# Geometry alignment
base_bottom_z = phantom_front_z + desired_distance_to_phantom
base_layer_center = base_bottom_z + base_layer_thickness / 2
base_layer_top = base_layer_center + base_layer_thickness / 2

# Input weights
weights = [0.918003, 0.295703, 0.247931, 0.204536, 0.156234, 0.197178]
X = compute_X_from_W(weights)
X.remove(max(X))
X_div_2 = compute_half_values(X)

# Grid setup
distance = 1.4039055  # mm
num_ridge_groups = 289
rows, cols = 17, 17
center_row, center_col = 8, 8
zero_x_groups = [8 + i * cols + 1 for i in range(rows)]
zero_y_groups = list(range(center_row * cols + 1, center_row * cols + cols + 1))

# Header
content = f"""# {base_material} base layer ({base_layer_thickness} cm)
########################################

s:Ge/RidgeGroup/Parent = "World"
s:Ge/RidgeGroup/Type   = "Group"
d:Ge/RidgeGroup/RotX   = 0.0 deg
d:Ge/RidgeGroup/RotY   = 0.0 deg
d:Ge/RidgeGroup/RotZ   = 0.0 deg
d:Ge/RidgeGroup/TransX = 0.0 mm
d:Ge/RidgeGroup/TransY = 0.0 mm
d:Ge/RidgeGroup/TransZ = 0.0 cm

s:Ge/Sq1/Parent       = "RidgeGroup"
s:Ge/Sq1/Type         = "TsBox"
s:Ge/Sq1/Material     = "{base_material}"
s:Ge/Sq1/Color        = "lightblue"
s:Ge/Sq1/DrawingStyle = "Solid"
d:Ge/Sq1/HLX          = 10.0 cm
d:Ge/Sq1/HLY          = 10.0 cm
```

```
d:Ge/Sq1/HLZ          = {base_layer_thickness / 2:.4f} cm
d:Ge/Sq1/TransZ       = {base_layer_center:.5f} cm
"""

# Ridge Filter Squares
for group_number in range(1, num_ridge_groups + 1):
    row = (group_number - 1) // cols
    col = (group_number - 1) % cols

    transX = 0 if group_number in zero_x_groups else (col - center_col) *
    distance
    transY = 0 if group_number in zero_y_groups else (row - center_row) *
    distance

    content += f"""
# RidgeFilter Group {group_number}
#####################################

s:Ge/RidgeGroup{group_number}/Parent = "World"
s:Ge/RidgeGroup{group_number}/Type   = "Group"
d:Ge/RidgeGroup{group_number}/RotX   = 0.0 deg
d:Ge/RidgeGroup{group_number}/RotY   = 0.0 deg
d:Ge/RidgeGroup{group_number}/RotZ   = 0.0 deg
d:Ge/RidgeGroup{group_number}/TransX = {round(transX, 5)} mm
d:Ge/RidgeGroup{group_number}/TransY = {round(transY, 5)} mm
d:Ge/RidgeGroup{group_number}/TransZ = 0.0 cm
"""

    for sq_number in range(1, 6):
        if sq_number == 5:  # Comment this if-cond to not have it pre-defined
            hlx = 0.20  # mm
            hly = 0.20
        else:
            hlx = X_div_2[(sq_number - 1) % len(X_div_2)]
            hly = hlx

        colour = "yellow" if (sq_number % 2) != 0 else "pink"
        transZ = base_layer_top + pyramid_layer_thickness / 2 + (sq_number - 1)
    * pyramid_layer_thickness

        content += f"""
s:Ge/Sq{sq_number}.{group_number}/Parent       = "RidgeGroup{group_number}"
s:Ge/Sq{sq_number}.{group_number}/Type         = "TsBox"
s:Ge/Sq{sq_number}.{group_number}/Material     = "{base_material}"
s:Ge/Sq{sq_number}.{group_number}/Color        = "{colour}"
s:Ge/Sq{sq_number}.{group_number}/DrawingStyle = "Solid"
d:Ge/Sq{sq_number}.{group_number}/HLX          = {hlx} mm
d:Ge/Sq{sq_number}.{group_number}/HLY          = {hly} mm
d:Ge/Sq{sq_number}.{group_number}/HLZ          = {pyramid_HLZ} mm
d:Ge/Sq{sq_number}.{group_number}/TransZ       = {transZ:.5f} cm
"""

# Save to file
output_filename = "pyramid_dimensions_Nylon12.txt"
with open(output_filename, "w") as file:
    file.write(content)

```

```
print(f"File '{output_filename}' has been successfully written.")
```

Listing 4: Python code for determining the Nylon-12 pyramid layers geometry.

B.2.3 PMMA Base Layer with Different Thickness Proportions

```
import math

def compute_X_from_W(weights):
    X = [0] * 6
    X[5] = math.sqrt(weights[5])
    for i in range(4, -1, -1):
        X[i] = math.sqrt(weights[i] + X[i + 1] ** 2)
    return X

def compute_half_values(weights):
    return [w / 2 for w in weights]

# ========== USER CONFIGURATION ==========
desired_distance_to_phantom = 25.0  # cm
phantom_front_z = 10.0              # cm
blockA_thickness = 1.0              # cm
blockB_thickness = 3.0              # cm
base_material = "PMMA"
block_HLX = 5.0                     # cm (half-width)
# ========================================

# Geometry setup
pyramid_HLZ_mm = 2.1585
pyramid_HLZ_cm = pyramid_HLZ_mm / 10
pyramid_layer_thickness = 2 * pyramid_HLZ_cm

# Compute offset to align base top with bottom of first pyramid layer
top_surface_z = phantom_front_z + desired_distance_to_phantom
centerA = top_surface_z - pyramid_HLZ_cm - (blockA_thickness / 2)
centerB = top_surface_z - pyramid_HLZ_cm - (blockB_thickness / 2)

# Input weights (for 2cm tumour at 12cm)
weights = [0.918003, 0.295703, 0.247931, 0.204536, 0.156234, 0.197178]
X = compute_X_from_W(weights)
X.remove(max(X))
X_div_2 = compute_half_values(X)

# Constants
distance = 1.4039055  # mm
num_ridge_groups = 289
rows, cols = 17, 17
center_row, center_col = 8, 8
zero_x_groups = [8 + i * cols + 1 for i in range(rows)]
zero_y_groups = list(range(center_row * cols + 1, center_row * cols + cols + 1))

# Begin geometry content
content = f"""# Split PMMA Base Layer (A: {blockA_thickness} cm, B: {
    blockB_thickness} cm)
########################################

# RidgeGroup to hold base
```

```
s:Ge/RidgeGroup/Parent = "World"
s:Ge/RidgeGroup/Type   = "Group"
d:Ge/RidgeGroup/TransX = 0.0 mm
d:Ge/RidgeGroup/TransY = 0.0 mm
d:Ge/RidgeGroup/TransZ = 0.0 cm
"""

# Block A
content += f"""
# Block A (left)
s:Ge/BaseBlockA/Parent       = "RidgeGroup"
s:Ge/BaseBlockA/Type         = "TsBox"
s:Ge/BaseBlockA/Material     = "{base_material}"
s:Ge/BaseBlockA/Color        = "lightblue"
s:Ge/BaseBlockA/DrawingStyle = "Solid"
d:Ge/BaseBlockA/HLX          = {block_HLX:.2f} cm
d:Ge/BaseBlockA/HLY          = 10.0 cm
d:Ge/BaseBlockA/HLZ          = {blockA_thickness / 2:.4f} cm
d:Ge/BaseBlockA/TransX       = {-block_HLX:.4f} cm
d:Ge/BaseBlockA/TransZ       = {centerA:.5f} cm
"""

# Block B
content += f"""
# Block B (right)
s:Ge/BaseBlockB/Parent       = "RidgeGroup"
s:Ge/BaseBlockB/Type         = "TsBox"
s:Ge/BaseBlockB/Material     = "{base_material}"
s:Ge/BaseBlockB/Color        = "skyblue"
s:Ge/BaseBlockB/DrawingStyle = "Solid"
d:Ge/BaseBlockB/HLX          = {block_HLX:.2f} cm
d:Ge/BaseBlockB/HLY          = 10.0 cm
d:Ge/BaseBlockB/HLZ          = {blockB_thickness / 2:.4f} cm
d:Ge/BaseBlockB/TransX       = {block_HLX:.4f} cm
d:Ge/BaseBlockB/TransZ       = {centerB:.5f} cm
"""

# Ridge groups and pyramids
for group_number in range(1, num_ridge_groups + 1):
    row = (group_number - 1) // cols
    col = (group_number - 1) % cols

    if group_number in zero_x_groups:
        transX = 0
        transY = (row - center_row) * distance
    elif group_number in zero_y_groups:
        transX = (col - center_col) * distance
        transY = 0
    else:
        transX = (col - center_col) * distance
        transY = (row - center_row) * distance

    content += f"""
# RidgeFilter Group {group_number}
#####################################

s:Ge/RidgeGroup{group_number}/Parent = "World"
```

```
s:Ge/RidgeGroup{group_number}/Type    = "Group"
d:Ge/RidgeGroup{group_number}/RotX    = 0.0 deg
d:Ge/RidgeGroup{group_number}/RotY    = 0.0 deg
d:Ge/RidgeGroup{group_number}/RotZ    = 0.0 deg
d:Ge/RidgeGroup{group_number}/TransX = {round(transX, 5)} mm
d:Ge/RidgeGroup{group_number}/TransY = {round(transY, 5)} mm
d:Ge/RidgeGroup{group_number}/TransZ = 0.0 cm
"""

    for sq_number in range(1, 6):
        if sq_number == 5:  # Comment this if-cond to not have it pre-defined
            hlx = 0.20
            hly = 0.20
        else:
            hlx = X_div_2[(sq_number - 1) % len(X_div_2)]
            hly = hlx

        colour = "yellow" if (sq_number % 2) != 0 else "pink"
        transZ = top_surface_z + (sq_number - 1) * pyramid_layer_thickness

        content += f"""
s:Ge/Sq{sq_number}.{group_number}/Parent        = "RidgeGroup{group_number}"
s:Ge/Sq{sq_number}.{group_number}/Type          = "TsBox"
s:Ge/Sq{sq_number}.{group_number}/Material      = "{base_material}"
s:Ge/Sq{sq_number}.{group_number}/Color         = "{colour}"
s:Ge/Sq{sq_number}.{group_number}/DrawingStyle = "Solid"
d:Ge/Sq{sq_number}.{group_number}/HLX           = {hlx} mm
d:Ge/Sq{sq_number}.{group_number}/HLY           = {hly} mm
d:Ge/Sq{sq_number}.{group_number}/HLZ           = {pyramid_HLZ_mm} mm
d:Ge/Sq{sq_number}.{group_number}/TransZ        = {transZ:.5f} cm
"""

# Save to file
output_filename = "pyramid_splitted_base_layer.txt"
with open(output_filename, "w") as file:
    file.write(content)

print(f"File '{output_filename}' has been successfully written.")
```

Listing 5: Python code for determining the PMMA pyramid layers geometry in the case when the base layer has different thickness proportions (e.g. one half is 1 cm thick, the other half is 3 cm thick).

B.2.4 Collimator

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

def compute_uniformity_metrics(profile):
    mean_val = np.mean(profile)
    min_val = np.min(profile)
    max_val = np.max(profile)
    dmin = 100 * (min_val - mean_val) / mean_val
    dmax = 100 * (max_val - mean_val) / mean_val
    uniformity = 100 * (1 - (max_val - min_val) / (max_val + min_val))
```

```
13      return uniformity, dmin, dmax
14
15  def analyze_and_compare_collimator_effects(dose_with_path, dose_without_path,
        z_index=int((11.0 / 20.0) * 400)):
16      x_bins, y_bins, z_bins = 101, 101, 400
17      x_size_cm, y_size_cm, z_size_cm = 5.05, 5.05, 20.0
18      x = np.linspace(-x_size_cm / 2, x_size_cm / 2, x_bins)
19      y = np.linspace(-y_size_cm / 2, y_size_cm / 2, y_bins)
20      z = np.linspace(0, z_size_cm, z_bins)
21
22      dose_with = pd.read_csv(dose_with_path, comment="#", delim_whitespace=True,
        header=None)[3].values
23      dose_without = pd.read_csv(dose_without_path, comment="#", delim_whitespace=
        True, header=None)[3].values
24      dose_with = dose_with.reshape((x_bins, y_bins, z_bins))
25      dose_without = dose_without.reshape((x_bins, y_bins, z_bins))
26
27      dose_diff = dose_with - dose_without
28      abs_diff = np.abs(dose_diff)
29      rel_diff = np.divide(dose_diff, dose_without + 1e-10)
30
31      mean_abs_diff = np.mean(abs_diff)
32      mean_rel_diff = np.mean(np.abs(rel_diff))
33
34      X, Y = np.meshgrid(x, y, indexing='ij')
35      mask = (X >= -1.0) & (X <= 1.0) & (Y >= -1.0) & (Y <= 1.0)
36
37      dose_z_with = dose_with[:, :, z_index]
38      dose_z_without = dose_without[:, :, z_index]
39
40      dose_in_roi_with = dose_z_with[mask]
41      dose_in_roi_without = dose_z_without[mask]
42      dose_out_roi_with = dose_z_with[~mask]
43      dose_out_roi_without = dose_z_without[~mask]
44
45      conformity_with = np.sum(dose_in_roi_with) / np.sum(dose_z_with)
46      conformity_without = np.sum(dose_in_roi_without) / np.sum(dose_z_without)
47      out_of_field_reduction = (np.sum(dose_out_roi_without) - np.sum(
        dose_out_roi_with)) / (np.sum(dose_out_roi_without) + 1e-10)
48
49      print("Collimator Effect Analysis")
50      print(f"Mean absolute dose difference: {mean_abs_diff:.4f}")
51      print(f"Mean relative dose difference: {mean_rel_diff:.2%}")
52      print(f"Conformity WITH collimator: {conformity_with:.2%}")
53      print(f"Conformity WITHOUT collimator: {conformity_without:.2%}")
54      print(f"Out-of-field dose reduction: {out_of_field_reduction:.2%}")
55
56      fig, axs = plt.subplots(1, 3, figsize=(18, 5))
57      dose_plot = axs[0].imshow(dose_z_with, extent=[x[0], x[-1], y[0], y[-1]],
        origin='lower', cmap='turbo')
58      axs[0].set_title("Dose WITH Collimator", fontsize=14)
59      axs[1].imshow(dose_z_without, extent=[x[0], x[-1], y[0], y[-1]], origin='
        lower', cmap='turbo')
60      axs[1].set_title("Dose WITHOUT Collimator", fontsize=14)
61      diff_min = -np.max(np.abs(dose_diff[:, :, z_index]))
62      diff_max = np.max(np.abs(dose_diff[:, :, z_index]))
63      axs[2].imshow(dose_diff[:, :, z_index], extent=[x[0], x[-1], y[0], y[-1]],
```

```
                origin='lower', cmap='bwr', vmin=diff_min, vmax=diff_max)
    axs[2].set_title("Dose Difference (With - Without)", fontsize=14)

    cbar = fig.colorbar(dose_plot, ax=axs[2], orientation='vertical')
    cbar.set_label("Dose (Gy)", fontsize=12)
    cbar.ax.tick_params(labelsize=11)

    for ax in axs:
        ax.set_xlabel("X (cm)", fontsize=12)
        ax.set_ylabel("Y (cm)", fontsize=12)
        ax.tick_params(labelsize=11)
        ax.yaxis.set_major_locator(MaxNLocator(nbins=5))

    plt.tight_layout()
    plt.show()

    fig2, ax2 = plt.subplots(1, 2, figsize=(14, 4))
    center_x = x_bins // 2
    center_y = y_bins // 2

    x_roi = (x >= -1.0) & (x <= 1.0)
    y_roi = (y >= -1.0) & (y <= 1.0)
    h_with = dose_z_with[x_roi, center_y]
    h_without = dose_z_without[x_roi, center_y]
    v_with = dose_z_with[center_x, y_roi]
    v_without = dose_z_without[center_x, y_roi]

    U_h_w, dmin_h_w, dmax_h_w = compute_uniformity_metrics(h_with)
    U_h_wo, dmin_h_wo, dmax_h_wo = compute_uniformity_metrics(h_without)
    U_v_w, dmin_v_w, dmax_v_w = compute_uniformity_metrics(v_with)
    U_v_wo, dmin_v_wo, dmax_v_wo = compute_uniformity_metrics(v_without)

    # Horizontal profile
    ax2[0].plot(x, dose_z_without[:, center_y], label=f"No Collimator\nU={U_h_wo
    :.1f}%,  min ={dmin_h_wo:.1f}%,  max ={dmax_h_wo:.1f}%", color="black",
    linestyle="--")
    ax2[0].plot(x, dose_z_with[:, center_y], label=f"With Collimator\nU={U_h_w
    :.1f}%,  min ={dmin_h_w:.1f}%,  max ={dmax_h_w:.1f}%", color="red")
    ax2[0].set_title(f"Horizontal Profile (Z = 11 cm)", fontsize=16)
    ax2[0].set_xlabel("X (cm)", fontsize=14)
    ax2[0].set_ylabel("Dose (Gy)", fontsize=14)
    ax2[0].tick_params(labelsize=12)
    ax2[0].yaxis.set_major_locator(MaxNLocator(nbins=6))
    ax2[0].legend(fontsize=12)
    ax2[0].grid(True)

    # Vertical profile
    ax2[1].plot(y, dose_z_without[center_x, :], label=f"No Collimator\nU={U_v_wo
    :.1f}%,  min ={dmin_v_wo:.1f}%,  max ={dmax_v_wo:.1f}%", color="black",
    linestyle="--")
    ax2[1].plot(y, dose_z_with[center_x, :], label=f"With Collimator\nU={U_v_w
    :.1f}%,  min ={dmin_v_w:.1f}%,  max ={dmax_v_w:.1f}%", color="red")
    ax2[1].set_title(f"Vertical Profile (Z = 11 cm)", fontsize=16)
    ax2[1].set_xlabel("Y (cm)", fontsize=14)
    ax2[1].set_ylabel("Dose (Gy)", fontsize=14)
    ax2[1].tick_params(labelsize=12)
    ax2[1].yaxis.set_major_locator(MaxNLocator(nbins=6))
```

```
    ax2[1].legend(fontsize=12)
    ax2[1].grid(True)

    plt.tight_layout()
    plt.show()
```

Listing 6: Python code for analysing the effect of using a collimator for both addition and simulation results.

B.2.5 Dose Translation into a Specific Grid Size

```
import numpy as np
import pandas as pd

def create_combined_SOBP(input_csv, output_csv, grid_size=17, shift_mm
    =1.4039055):
    """
    Create a combined SOBP from a single pyramid file by translating it into a
    grid of shifted copies.

    Parameters:
        input_csv: path to base single-pyramid file
        output_csv: where to save combined dose
        grid_size: odd number, like 3, 5, 9, 17, etc.
        shift_mm: physical spacing between pyramid centers in mm
    """

    # Geometry
    x_bins, y_bins, z_bins = 101, 101, 400
    z_size_cm = 20.0
    dz = z_size_cm / z_bins
    voxel_size_mm = 0.5  # mm per voxel (from your setup)

    # Convert physical shift to voxel shift
    shift_vox = int(round(shift_mm / voxel_size_mm))

    # Load dose and reshape
    df = pd.read_csv(input_csv, comment="#", delim_whitespace=True, header=None)
    dose_flat = df[3].values
    dose_3d = dose_flat.reshape((x_bins, y_bins, z_bins))

    # Generate shifts
    half_grid = grid_size // 2
    shifts = [(dy * shift_vox, dx * shift_vox)
              for dy in range(-half_grid, half_grid + 1)
              for dx in range(-half_grid, half_grid + 1)]

    print(f"Generating {grid_size}x{grid_size} = {len(shifts)} pyramid shifts")

    # Combine all shifted doses
    combined_dose = np.zeros_like(dose_3d)

    for dy, dx in shifts:
        shifted = np.roll(np.roll(dose_3d, dy, axis=0), dx, axis=1)

        # Clear wrap-around areas
        if dy != 0:
```

```
            if dy > 0:
                shifted[:dy, :, :] = 0
            else:
                shifted[dy:, :, :] = 0
        if dx != 0:
            if dx > 0:
                shifted[:, :dx, :] = 0
            else:
                shifted[:, dx:, :] = 0

        combined_dose += shifted

    df_new = df.copy()
    df_new[3] = combined_dose.reshape(-1)
    df_new.to_csv(output_csv, sep=' ', header=False, index=False)
    print(f"Combined SOBP saved to '{output_csv}'.")

create_combined_SOBP("/Applications/topas/Clara/Simulations/Practice/Week10
   /27-06/25cm/SOBP.csv", "SOBP_27x27.csv", grid_size=27)
```

Listing 7: Python code for translating and summing the dose from only one pyramid into a specific grid size to use less computational power.

The plots below (Figure B.1 and Figure B.2) show that the designed code for translating the dose leads to a similar outcome by simply running the simulation in TOPAS.

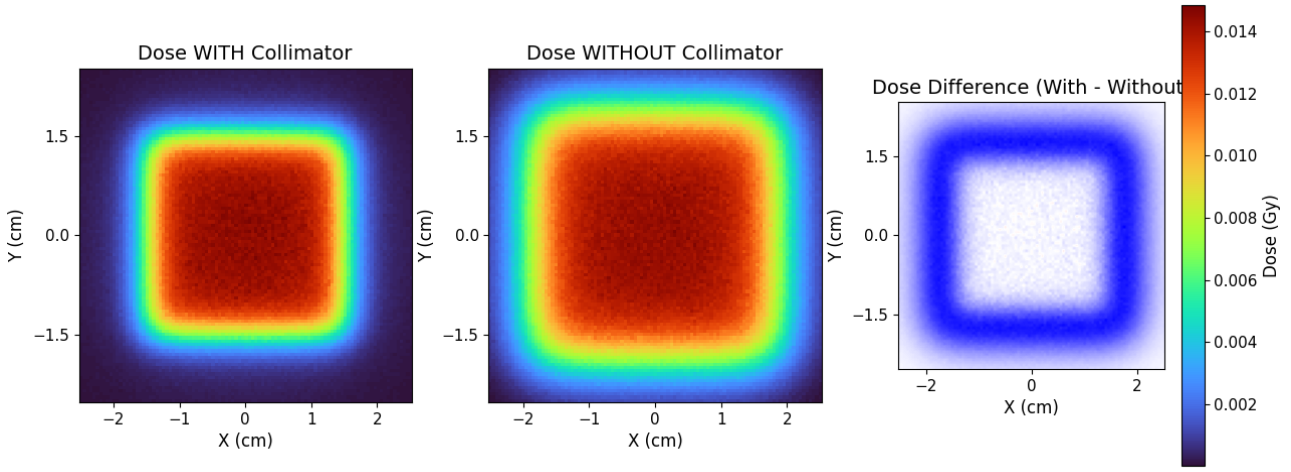

Figure B.1: Comparison of XY dose slice for a 27 × 27 array of pyramids with and without a collimator by directly simulating in TOPAS.

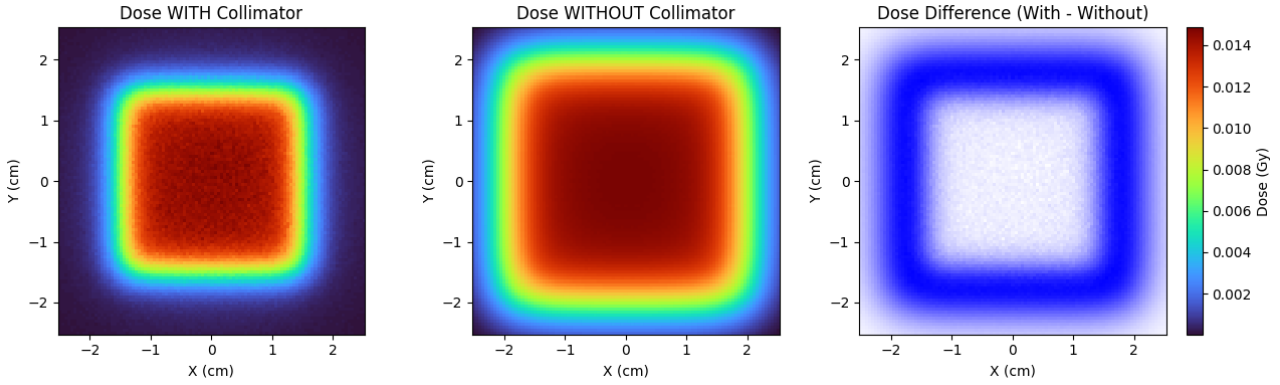

Figure B.2: Comparison of XY dose slice for a 27 × 27 array of pyramids with and without a collimator by translating the dose from only one pyramid.

C AI Signed Declaration

university of groningen / faculty of science and engineering / biomedical engineering

Appendix: Declaration on the use of generative AI systems during BME projects

Title of the project:	Pyramid Filter: Design of a 3D Range Spreader for FLASH Irradiations
Your full name:	Clara Popescu-Boboc
Student number:	S5067421

In the project I have used systems based on generative artificial intelligence (AI)[1, 2]
(please check one of the boxes with X).

☒ Yes ☐ No

If you have selected "Yes", complete the rest of the form. If you have selected "No", simply fill in the place, date and signature below.

I have used the following generative AI based systems in the creation of this thesis: [1, 2]
(please list all systems used below)

1.	OpenAI. (2025). ChatGPT with GPT-4o (April 30 version) [Large language model]. https://chat.openai.com/chat
2.	
3.	
Other:	

I further declare that I
(please check one of the boxes with X).)

☐ have actively informed myself about the capabilities and limitations of the above-mentioned AI systems to the extent that I can use them responsibly,
☐ have labelled the content taken from the AI systems listed above with my details in the table below,
☐ have verified that the content generated by the above-mentioned AI systems and adopted by me is factually correct,
☒ am aware that, as the author of this work, I am responsible for the information and statements made in it,
☐ am aware that the violation of the disclosure of the use of generative AI in my work is a deception and leads to an evaluation with an insufficient grade.

- Indicate in the table on the next page when the above-mentioned AI systems have been used during your project.
- When you have completed and signed the form, please add it to the beginning of your thesis/report, straight after the standard title page).

[1] This declaration does not apply to the use of basic widely used tools for checking spelling and grammar, translating texts and improving software quality for data analysis and software prototypes.
[2] If you are unsure whether an IT system used is a generative AI system and/or whether you need to declare it, declare it.

I have applied the above-mentioned AI systems as indicated below.

Areas of contribution	Number AI system(s) used	Description of the manner of use and compliance with good scientific practice, if necessary separately by chapter of the work
Development and conception of the research project	1	Used for better understanding the concepts
Collection and evaluation of literature sources	1	Used for better understanding the concepts from literature
Elaboration, collection and/or procurement of data		
Processing of data	1	Explanation of certain acquired plots or ideas on how to continue when got stuck
Selection of methodology		

Programming	1	Finding bugs, optimisation and automatisation of codes
Analysis/evaluation of the data/sources	1	Explanation of certain acquired plots and physics concepts
Interpretation of the analysis /evaluation and derivation of conclusions	1	Avoiding repetition, better use of English grammar and vocabulary
Writing of the manuscript: Creation of visualizations		
Writing of the manuscript: Structuring the text	1	LaTeX syntax for correctly formatting figures, tables, title page and bibliography
Writing of the manuscript: Formulation of text	1	Avoiding repetition, better use of English grammar and vocabulary

Writing of the manuscript: Revision of text		

Further contributions / additional information:

Place	Date	Signature
Groningen, NL	07-07-2025	Clara Es