UCLouvain

École polytechnique de Louvain

Simultaneous Monte Carlo dose computation and optimization for proton therapy: the beamlet-free approach

Authors: **Valentin DECAT, Benjamin THARIN**
Supervisor: **John A. LEE**
Readers: **François GLINEUR, Guillaume JANSSENS, Romain SCHYNS**
Academic year 2024–2025
Master [120] in Biomedical Engineering

Abstract

Radiotherapy is one of the most common treatments for cancer. Proton therapy, a more advanced technique that uses proton beams instead of X-rays, is growing rapidly due to its ability to target tumors more precisely.
Today, proton therapy treatment planning is typically based on so-called "beamlet-based" methods, which involve dividing each beam into smaller units called "beamlets", whose intensities are individually optimized.
These techniques are effective but require storing all the beamlets and involve high computational demands, which can be a limitation in situations where speed is essential, such as in adaptive proton therapy.

This thesis introduces a new approach called beamlet-free, in which the optimization is performed without storing all individual beamlets. Instead, the method simulate randomly beamlet pairs to gradually estimate their optimal intensities.
We present a proof of concept of this algorithm and analyze its convergence on both a simplified case and a realistic patient case, comparing it to the traditional beamlet-based method.

Although our algorithm consistently converges toward a solution, our experiments on a simple case with few beamlets reveal residual noise stemming from its stochastic nature and the beamlet approximations. This noise slows convergence and prevents the algorithm from achieving a clinically acceptable treatment plan within a practical number of iterations. In contrast, for a more complex case with many beamlets, the beamlet-free approach appears to approximate the performance of the beamlet-based method more closely.

Further work is needed to assess whether these limitations can be overcome, enabling the full theoretical advantages of this approach to be realized. It is also important to investigate whether, in practice, the beamlet-free algorithm can outperform the beamlet-based method in terms of speed and computational cost.

Acknowledgment

We would like to thank everyone who contributed to this work, whether directly or indirectly. Without their support, this thesis would not have been possible.

First, we would like to thank John Aldo Lee for supervising our thesis, for providing us with valuable advice, and for being available throughout the year.

We also want to thank Romain Schyns for guiding us through this project and for generously giving his time during our various meetings.

A big thank you to the entire MIRO team for always being welcoming during our Friday afternoons at the lab, and for providing us with an office space on those occasions.

We are also grateful to our entire jury: John Aldo Lee, François Glineur, Guillaume Janssens, and Romain Schyns, for taking the time to read and evaluate our work. We truly appreciate it.

Thanks to Valentine Dormal and Danah Pross for providing us with patient images.

Finally, we would like to especially thank our families and friends for their unwavering support throughout our studies and during the writing of this thesis. Thank you!

Preliminary Notes

The AI ChatGPT form OpenAI was used for sentence improvement and translation into English. However, it was not used to generate explanations. DeepL was also used for translation.

Contents

1 Introduction **7**

2 State of the Art **10**
2.1 Radiotherapy 10
2.2 Protontherapy 12
2.2.1 Physics of Protontherapy 12
2.2.2 Depth-Dose Comparison Between Protons and X-rays . 16
2.2.3 Protontherapy Facility 17
2.3 Protontherapy Delivery Techniques 18
2.4 Protontherapy Workflow 22
2.5 Monte Carlo Method 23

3 Optimization Strategies for Treatment Planning **25**
3.1 Cost Function in Radiotherapy 25
3.2 Gradient Descent 27
3.2.1 Fundamentals of Gradient Descent 27
3.2.2 Stochastic Gradient Descent 28
3.2.3 Beamlet-Based Optimization 29
3.2.4 Beamlet-Free Optimization 31
3.3 Learning Rate Selection 33

4 Materials and Methods **34**
4.1 Materials 34
4.1.1 MCsquare 34
4.1.2 OpenTPS 34
4.1.3 Patient Imaging Data 34
4.2 Analytical Gradient Calculation 36
4.3 Practical Implementation 40
4.3.1 Pseudo Code 40
4.3.2 Initialization of Clinical Constraints 41
4.3.3 Beamlet Simulation 43

4.3.4 Computation of the Gradient . . . 43
4.3.5 Pre-Filling of the Correlation Matrix $\mathbf{C}_W$ . . . 46
4.3.6 Simulation Parameters . . . 49
4.3.7 Cost Evaluation . . . 50
4.3.8 Determining a Stopping Criterion . . . 50

5 Results **52**
5.1 Dosimetric Results . . . 52
5.2 Quantitative Results . . . 53
5.2.1 Cost Functions . . . 53
5.2.2 Dose Volume Histograms . . . 53
5.3 Learning Rate . . . 55
5.4 Stopping Criterion . . . 57
5.5 Application to a Realistic Patient Case . . . 59
5.6 Comparison with the Beamlet-Based Algorithm . . . 61
5.6.1 Example on a Simple Configuration . . . 61
5.6.2 Example on a Realistic Configuration . . . 62

6 Discussion **64**
6.1 Objectives of the Study . . . 64
6.2 Interpretation of Results . . . 64
6.2.1 Residual Noise in the Optimization Process . . . 64
6.2.2 Search for a Stopping Criterion . . . 67
6.2.3 Comparison with the Beamlet-Based Approach . . . 68
6.3 Limitations . . . 70
6.3.1 Impact of the Number of Beamlets on Algorithm Performance . . . 70
6.3.2 Limitation Due to Beamlet Reuse . . . 70
6.3.3 Pre-filling of the Correlation Matrix $\mathbf{C}_W$ . . . 71
6.3.4 Addition of an Organ at Risk . . . 71
6.4 Perspectives . . . 73
6.4.1 Code Optimization . . . 73
6.4.2 Improved Management of Organs at Risk . . . 74
6.4.3 Improved Pre-calculation of the Correlation Matrices $\mathbf{C}_W$ . . . 74
6.4.4 Learning Rate Adaptation . . . 75

7 Conclusion **76**

Chapter 1

Introduction

In Belgium, cancer was one of the leading causes of death in 2016. In 2018, 18.1 million cases were recorded and 9.6 million deaths were attributed to cancer. In addition, 43.8 million people were living with a cancer that was detected within 5 years in 2018. Experts expect 29 million cases in 2040 due to aging and population growth [1].

However, it is possible to significantly reduce the morbidity of cancer by using solutions that are already known. There are many effective treatments for cancer, and around 50% of patients worldwide are treated with radiotherapy [2]. Although proton therapy, which uses protons instead of conventional X-rays as in traditional radiotherapy, offers significant clinical advantages, less than 1% of patients are treated with protons or heavier ions [3]. This is primarily due to the technical and financial challenges involved in generating and accelerating protons, which demand large-scale equipment such as cyclotrons or synchrotrons, as well as sophisticated delivery systems. These infrastructures are significantly more expensive and complex to build and operate than those used in conventional X-ray–based radiotherapy.

In both conventional radiotherapy and proton therapy, establishing a treatment plan is a crucial step. Treatment planning involves defining the ideal dose to be delivered to the tumor while ensuring that organs at risk receive doses below predefined safety thresholds.

The most common delivery technique, known as pencil beam scanning, divides each radiation beam into a large number of smaller sub-units called beamlets. Each beamlet is characterized by its own energy, intensity, and direction.

Optimizing the treatment plan consists of adjusting the parameters of each beamlet to modulate the dose distribution as precisely as possible, in order to fulfill clinical objectives, maximizing tumor control while minimizing exposure to healthy tissues.

In proton therapy, optimizing the treatment is even more important than in traditional radiotherapy because the way the dose is delivered is very sensitive to uncertainties and changes in the patient's anatomy. Without proper optimization, this may lead to an inhomogeneous dose distribution within the tumor or high exposure of organs at risk, potentially causing severe side effects.

However, treatment plan optimization is often computationally intensive and time-consuming, which complicates its use in real-time or adaptive settings, something that could be particularly beneficial for tumors located near moving organs, such as the lungs. The high computational demand is largely due to the complexity of the calculations needed to simulate and store beamlets, as these involve detailed modeling of proton interactions with matter. Accurately computing dose distributions and storing individual beamlet data are essential steps in determining dose gradients and adjusting their weights (intensities) during optimization.

A new approach to overcome this limitation is the beamlet-free algorithm. Unlike traditional beamlet-based methods, which require storing all beamlets in memory, the beamlet-free approach estimates the gradient using only a small subset of beamlets. The goal is to drastically reduce computation time and memory usage, while maintaining an accuracy level comparable to conventional methods. These improvements could enable real-time treatment planning while still meeting clinical requirements.

An initial version of the "beamlet-free" algorithm was introduced by Danah Pross et al [4]. It also aims to optimize proton therapy treatment plans, but follows a different approach from the one used in this thesis : at each iteration, it simulates a batch of protons for a randomly selected spot, updates the dose map, and estimates the cost variation to compute the spot's gradient. The selection probability is then adjusted accordingly, creating a probability map used to guide future selections. The final spot weights correspond to how often each was chosen. We do not elaborate on this method, as it is not the focus of our study in this thesis.

A new version of the beamlet-free algorithm, which forms the basis of

this thesis, simulates only two beamlets at a time, each with a small number of protons. The method then computes the gradient for these two beamlets, which depends in particular on their correlation with the others. These correlations are stored in dedicated correlation matrices. The key advantage is that this approach retains useful information that becomes more accurate over iterations, without needing to store the full dose distribution of each individual beamlet.

The objective of this thesis is to implement the new approach of the beamlet-free algorithm, which uses a pair of beamlets, and to evaluate its performance. As a proof of concept, we focus on demonstrating the feasibility and convergence of the algorithm, rather than optimizing for computational efficiency. Consequently, we do not analyze convergence speed or runtime. The algorithm is tested using both a simplified, non-clinical scenario and a realistic patient image featuring a tumor located in the right lung. More specifically, the goal of this thesis is to answer the research question: Can the beamlet-free algorithm converge to a result comparable to the beamlet-based approach, and under what conditions?

Chapter 2

State of the Art

2.1 Radiotherapy

As we mentioned in the introduction, radiotherapy is one of the most common ways to treat cancer. Other methods such as surgery, chemotherapy, and immunotherapy are also available, but all of these have different side effects. Physicians select the most appropriate treatment depending on the type of cancer, the progression of the disease, and other clinical factors.

However, radiotherapy remains one of the standard treatments recommended in the early stages of cancer, when there are few or no metastases and local treatment is sufficient. Its objective is to ensure local-regional control of the tumor, by eradicating all cancerous cells to prevent extension or recurrence in neighboring tissues [5].

Radiotherapy uses ionizing radiation, which is radiation with enough energy to release electrons from atoms or molecules. This includes two main categories: particulate radiation (such as alpha, beta and neutron radiation) and electromagnetic ionizing radiation (such as X-rays and gamma rays). X-rays are most commonly used in radiotherapy. It is therefore necessary to use machines, called linear accelerators, capable of producing radiation at energies known as megavoltages, so that it can penetrate deep enough into the patient's body. The radiation absorbed will consequently cause damage to the DNA structures of the cells. The impact of this damage on these DNA structures can cause alterations at cellular level and eventually lead to cell death. Damage to DNA can be caused by direct radiation, by hitting the DNA itself, or indirect radiation via free radicals from water ionization. Among the different kinds of DNA damage, double-strand breaks typically

cause the most serious damage [6].

In order to deliver radiation to the tumor, the rays must inevitably pass through healthy tissue. Consequently, surrounding healthy cells are exposed to significant doses of radiation, which can damage their DNA, sometimes leading to cell dysfunction. However, cells possess DNA repair mechanisms that help correct such damage and prevent immediate dysfunction. Once radiation has caused DNA damage, three main cellular responses are possible [5]:

- Accurate repair: the cell correctly repairs the damage and maintain normal function.
- Inadequate repair: the repair process fails, leading to cell inactivation through mechanisms such as mitotic death, apoptosis or permanent growth arrest.
- Misrepair: the cell survives despite incorrect repair, resulting in permanent genetic alterations. This process can lead to long-term side effects, which may manifest years after treatment.

In order to avoid excessive doses in healthy tissues during planning, maximum tolerated doses are given according to the different tissues crossed in addition to the dose prescribed to the tumor itself. A unit of radiation dose is expressed in Gray (Gy) corresponding to joules of energy absorbed per kilogram of tissue. Fortunately, DNA repair mechanisms in cancer cells are often impaired, making them less effective than in healthy cells. This difference creates a therapeutic window that allows us to destroy most cancer cells while preserving the function of healthy tissue [7]. Different strategies can be used to maximize the damage to the tumor compared with healthy tissue.

Fractionation

In radiotherapy, the biological dose represents an adjusted version of the physical dose that accounts for tissue-specific sensitivity, providing a more accurate estimation of its biological impact. The aim of a treatment to maximize the biological dose delivered to the tumor while minimizing the dose in the rest of the tissues, particularly in organs at risk. Fractionation is a method designed to increase the difference between the biological dose delivered to the tumor and that received by healthy tissues. Which means that for the same total physical dose, the damage to healthy cells will be much less

than that to cancerous cells. The total dose is divided into several sessions of lower intensity, delivered regularly over several weeks. As mentioned above, cancer cells often have impaired DNA repair mechanisms, unlike healthy cells, which regenerate more efficiently. Rather than administering a single large beam likely to cause irreversible damage to many healthy cells, fractionation makes it possible to limit the biological dose collected by these cells, while maintaining a dose sufficient to affect the tumor. In this way, repair errors are relatively rare in healthy cells, whereas they accumulate in cancer cells, gradually leading to cell death.

Beam modulation

By using multiple beams, a more uniform dose can be distributed across the entire target while minimizing exposure to nearby organs at risk. Modulation of each beam intensity is also important, so that higher doses can be delivered where they are needed, while reducing doses in areas where organs at risk are affected.

Particle Properties

The kind of particles selected could be an additional choice to maximize damage to the tumor. This is where traditional radiation therapy (with photons) and proton therapy (with protons) differ, as we will explore in the following section.

2.2 Protontherapy

Proton therapy differs from radiotherapy in that protons rather than X-rays (photons) are used in the treatment. The behavior of protons in matter, and the specific facilities required for their production and guidance, differ significantly from those of photons. These aspects are detailed in the following subsections.

2.2.1 Physics of Protontherapy

In proton therapy, as in conventional radiotherapy, the dose delivered results from the interaction of particles with matter, which slows down or scatters the particles. It is the transfer of energy from the particles to the medium. However, depending on the nature of the particles used (alpha particles, protons, photons, etc.), these interactions are very different, which leads into different resulting doses. Protons are heavy, positively charged particles,

unlike photons, which are uncharged particles considered to have no mass. It's important to understand that the behavior of particles as they travel through matter is highly stochastic. As a result, it's difficult to predict the exact path a single proton will take once it's emitted. However, by simulating a large number of particles within the same beam, we can reliably predict the beam's overall interaction with matter. Because of their charge and mass, protons can interact with matter through several mechanisms. Figure 2.1 shows the 3 main types of proton interaction with matter at the energies used in proton therapy [5][8]:

a) Inelastic collision with atomically bonded electrons: the incident proton interacts with an atomically bonded electron, transferring some of its energy to it, resulting in ionization or excitation of the atom.

b) Elastic scattering: the incident proton interacts mainly with the nucleus of an atom, slightly modifying its trajectory with a slight loss of energy. Overall, protons are relatively weakly scattered.

c) Nuclear interactions: the incident proton interacts with the nucleus, generating secondary particles and the loss of the incident proton.

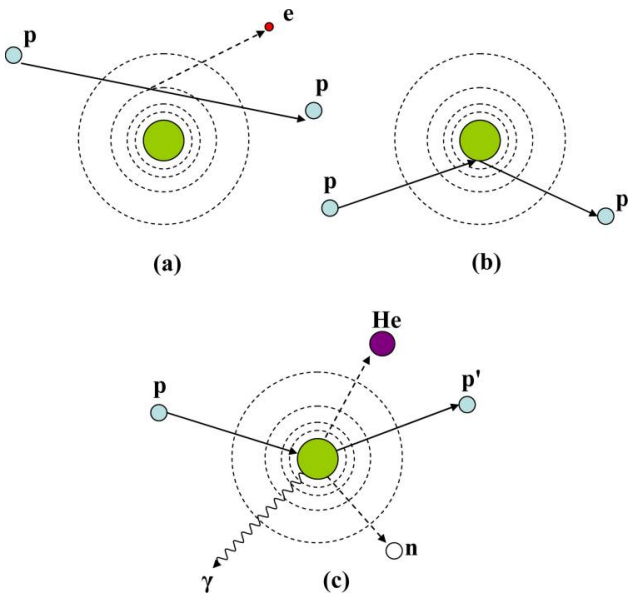


Figure 2.1: Illustration of the three main types of proton interactions with matter. Credits : [8]

To quantify these interactions, it is useful to define the rate of energy loss, given by the quotient S = dE/dx where E is the energy and x is the

distance. To make the equation independent of the density of the medium, it is possible to divide by the density, yielding the mass stopping power:

$$\frac{S}{\rho} = -\frac{dE}{\rho dx} \tag{2.1}$$

This rate is primarily due to inelastic collisions with bound atomic electrons. By considering only this type of interaction, we obtain the Bethe-Bloch formula [8]:

$$\frac{S}{\rho} = -\frac{1}{\rho}\frac{dE}{dx} = 4\pi N_e r_e^2 m_e c^2 \frac{Z}{A}\frac{z^2}{\beta^2}\left[\ln\left(\frac{2m_e c^2\gamma^2\beta^2}{I} - \beta^2 - \frac{\delta}{2} - \frac{C}{Z}\right)\right] \tag{2.2}$$

Figure 2.2 provides a graphical representation of this equation.

- The x-axis represents the particle energy in MeV.
- The y-axis represents the mass stopping power in eV/um.

In the figure 2.2, the red curve represents mass stopping power as a function of proton energy. It can be seen that, in general, the lower the particle energy, the higher the mass stopping power, meaning that the particle loses more energy over a short distance. This physical principle introduces the concept of the Bragg peak. When a particle emerges from a particle accelerator with a high energy (of the order of several hundred MeV), it passes through matter depositing only a small fraction of its energy, slowing it down very little over a long distance. On the other hand, once its energy has fallen to a few MeV, it releases a much greater quantity of its energy, which slows it down even more. This phenomenon causes protons to release a greater amount of energy at the end of their trajectory, a dose peak known as the Bragg peak [9].

Figure 2.3 illustrates the theoretical dose that should be observed when projecting high-energy protons into water. It shows a Bragg peak at around 30cm in the water. Before this peak, the deposited dose is relatively low, whereas after the peak, the deposited dose is almost zero, since almost all the particles have either stopped or have negligible energy left [10].

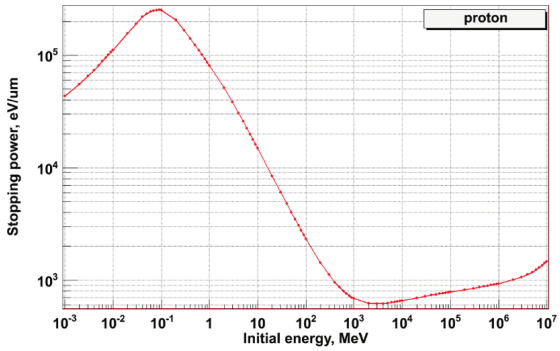


Figure 2.2: Variation of the mass stopping power as a function of particle energy. Credits : [9]

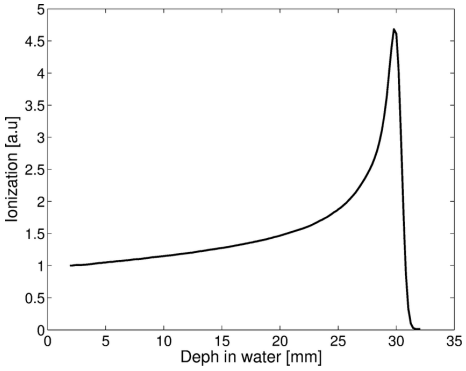


Figure 2.3: Depiction of the Bragg peak in water. Credits : [10]

Finally, figure 2.4 shows the evolution of the fluence of a proton beam as a function of depth. Fluence corresponds to the number of incident particles (in this case protons) crossing an elementary surface dA. It initially decreases slightly due to the protons' nuclear interactions and then, around the Bragg peak, decreases sharply: protons with no energy stop and are absorbed by the medium. The sigmoidal shape at the end of the curve reflects the stochastic fluctuations that particles undergo as they pass through matter [8].

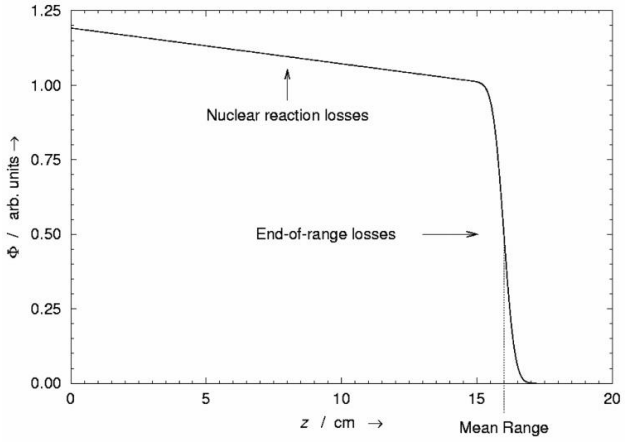


Figure 2.4: Relative fraction of the fluence ϕ in a broad beam of protons remaining as a function of depth z in water. Credits : [8]

2.2.2 Depth-Dose Comparison Between Protons and X-rays

One of the main advantages of proton therapy over conventional radiotherapy is the shape of the dose distribution.

In figure 2.5, the dotted red curve represents the dose distribution of a monoenergetic proton beam, which shows a clear Bragg peak at the target. By combining several proton energies, enabling Bragg peaks to be superimposed at different depths, this peak is "flattened" to deliver a homogeneous dose over the entire tumor zone (the shaded area). This produces the continuous red curve, which represents a proton beam modulated in energy to create a plateau: the Spread-Out Bragg Peak (SOBP). In contrast, the blue photon (X-ray) curve shows a significant release of energy before and after this same target region. This means that, to achieve an equivalent dose in the tumor, the surrounding healthy tissue receives a significantly higher dose with a photon beam than with a modulated proton beam [11].

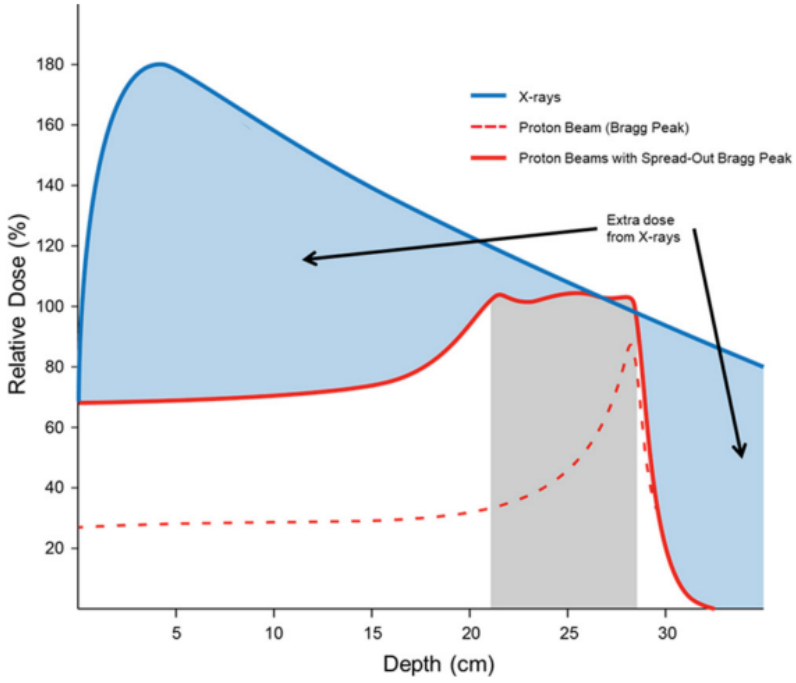


Figure 2.5: Depth-dose comparison of X-rays and proton beams, including a Spread-Out Bragg Peak. Credits : [11]

Figure 2.6 illustrates the difference between a conventional radiotherapy treatment plan and a proton therapy plan. The dose delivered to healthy tissues is significantly lower in proton therapy than in conventional radiotherapy. In proton therapy, almost no dose is observed beyond the target, unlike in conventional radiotherapy [12].

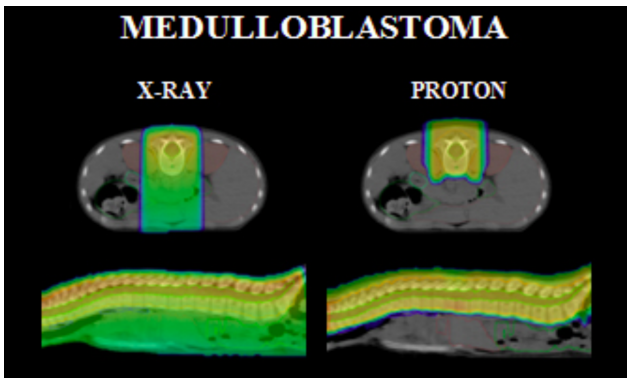


Figure 2.6: Dose distribution comparison of Radiotherapy and Protontherapy. Credits : [12]

2.2.3 Protontherapy Facility

Although proton therapy offers a major advantage thanks to proton-specific dose delivery, it also has its drawbacks.

One of the main limitations is the infrastructure required to accelerate, transport and deliver protons. Indeed, protons are far more complex to handle than photons. Figure 2.7 illustrates the main components of this type of installation. Firstly, protons are accelerated by a cyclotron or synchrotron. Although these two devices are based on different principles, they share the common goal of increasing particle energy to the level required for treatment. A single accelerator can generally power different treatment rooms via dedicated beamlines [5][13].

Once accelerated, particles are extracted from the accelerator into a beamline for transport to the treatment room. The beamline is equipped with various components to direct and focus the beam, preventing protons from coming into contact with its walls.

Finally, the beam enters a gantry, an imposing device designed to rotate the beam around the patient to target the tumor at different angles. To meet the technical constraints of these installations, a dedicated building generally has to be built within the hospital [5]. Because of this complexity, the cost of proton therapy is significantly higher than that of a conventional radiotherapy facility. Currently, only a limited number of hospitals worldwide have access to this type of infrastructure.

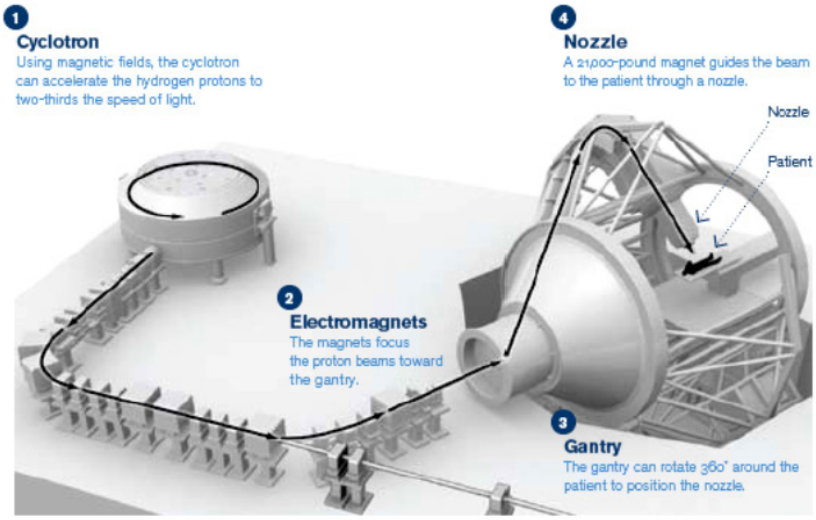


Figure 2.7: Schematic representation of a proton therapy system with its main components. Credits : [5]

2.3 Protontherapy Delivery Techniques

Once the proton beam has been brought into the treatment room, it is necessary to modulate its shape and energy, as well as the dose of protons that the patient will receive. Passive scattering is one of the two most commonly used techniques for delivering proton therapy. In this method, the narrow beam exiting the nozzle is scattered laterally and longitudinally by a modulator wheel and one or more scatterers, enabling the beam to cover the entire surface of the tumour. Next, a collimator is used to shape the beam so that it precisely matches the contour of the tumour. Any off-target protons are absorbed by this device. Finally, a compensator is placed in the beam path to modulate its penetration depth to conform to the target volume [13].

However, this technique has several disadvantages. The materials used must be tailored to the specific tumor; for instance, the shape of the compensator must be customised for each patient, which means that it has to be made to measure. In addition, a non-negligible dose is deposited between the tumour and the compensator, resulting in unnecessary irradiation of the healthy tissue located there [5].

The most commonly used technique today is pencil beam scanning (PBS). This method relies on the use of beamlets, which are very narrow, monoenergetic elementary beams. Unlike passive scattering, the aim of pencil beam scanning is not to scatter the beam, but to use magnets to change the direction of each beamlet. Depending on their energy, the beamlets are directed towards a precise point in the target, called a spot. The position of the spot corresponds to the position of the theoretical Bragg peak of the beamlet. In this way, the treatment plan is delivered layer by layer, thus resolving the two main problems with the passive scattering technique mentioned above [13].

Figure 2.8 shows a representation of a beamlet and its depth dose distribution. The spot corresponds to the location of the peak dose [14].

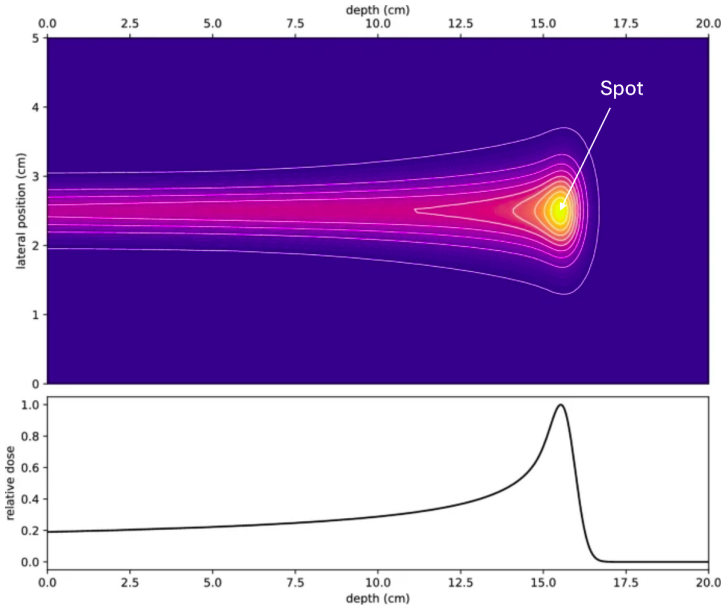


Figure 2.8: Representation and depth dose distribution of a beamlet. Credits : [14]

However, one of the main challenges of pencil beam scanning is the un-

certainty related to tumour positioning. Since this technique requires more time to deliver the dose compared to passive scattering, any movement of the tumour during treatment can lead to significant dose delivery errors. These two techniques are illustrated in Figure 2.9 [15].

In order to use the pencil beam scanning technique, it is necessary to define the position of each individual spot, and then generate the treatment plan based on these positions. All spot positions are stored in a spot matrix. Several parameters must therefore be defined, such as the spacing between spots and the spacing between layers. Pencil beam scanning will be the method used in this thesis. The concept of spots and beamlets is therefore fundamental for optimization in proton therapy, particularly for the beamlet-free algorithm that we detail in this thesis.

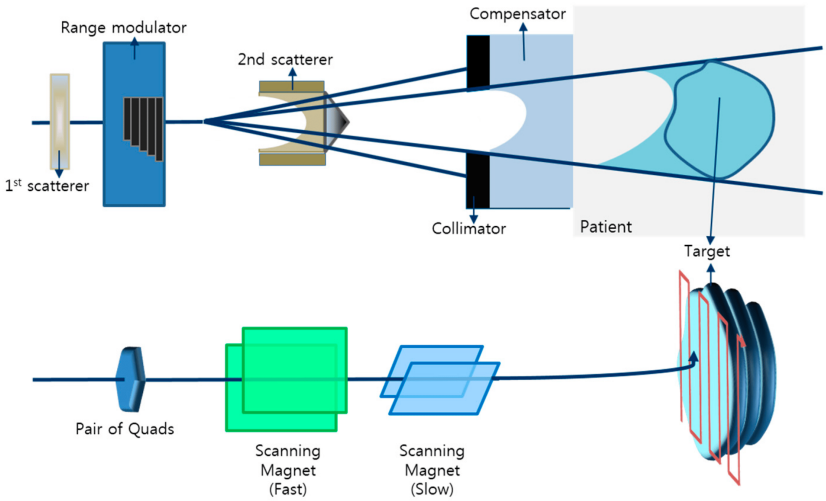


Figure 2.9: Diagram of passive scattering and pencil beam scanning techniques. Credits : [15]

Figure 2.10 shows the difference in dose distributions between the passive scattering technique on the left side and pencil beam scanning on the right side. The dose distributions appear to be more favorable with pencil beam scanning compared to passive scattering [3].

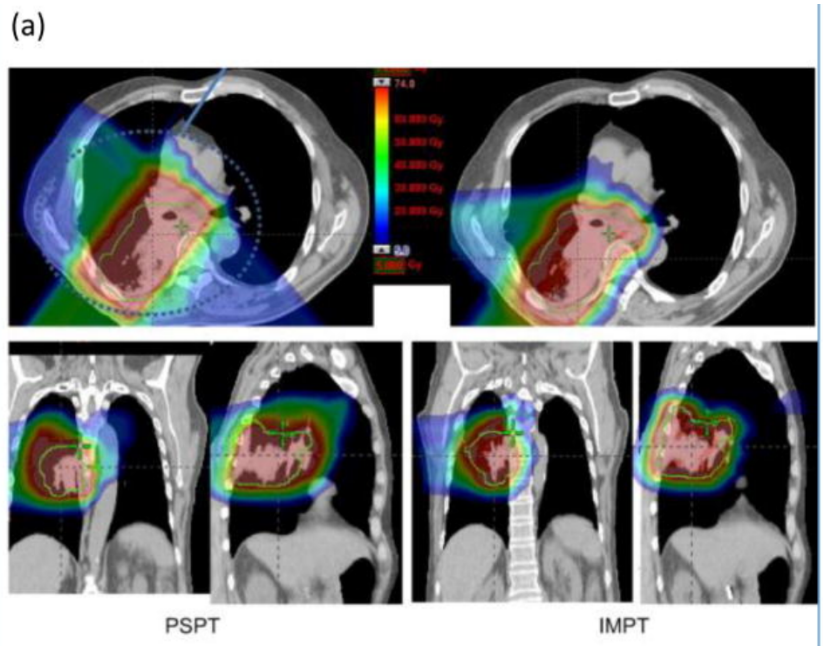


Figure 2.10: Passive scattering (PSPT) vs. Pencil beam scanning (IMPT) dose distributions. Credits : [3]

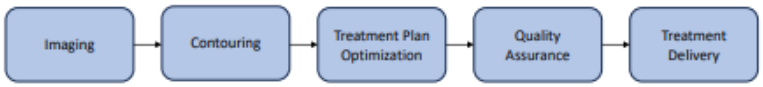


Figure 2.11: Key Steps in a Treatment Plan. Credits : [16]

2.4 Protontherapy Workflow

It is interesting to describe the 5 main stages in a proton therapy treatment. These are illustrated in the figure 2.11.

The first step of the workflow consists of taking images using a CT scan. The aim of this first stage is to estimate the density of the various tissues in the patient's body as accurately as possible. This is crucial because the stopping power of protons can vary considerably depending on tissue density, which can influence the depth of the bragg peak. The CT scanner generates images by analysing the attenuation of photons at different energies. This allows the images to be converted into Hunsfield units, which represent the attenuation of photons relative to water. The CT images also serve as an anatomical reference for the next steps. However, as we are using protons, it is necessary to convert the Hunsfield units into stopping power ratios, which represent the energy loss of protons relative to water. This conversion is only possible if the CT scanner has been calibrated beforehand [17].

The second step aims to establish the contouring on the CT images of the target(s) as well as the organ(s) at risk, and to prescribe the treatment plan. The prescription is given by the doctor and contains the desired dose in the target but also other objectives such as the maximum dose that can be absorbed in the organs at risk.

The third step is the optimization of the treatment plan. This stage will be described in more detail in the upcoming sections. In summary, the aim is to find a dose distribution that is realistic and that best complies with the prescription [5].

The fourth step is to analyse the quality of the treatment plan, using experimental measurements. One possible measurement method is to irradiate a water phantom (a tank filled with water) to record the actual dose distribution using detectors. Visually comparing different plans on the basis of their dose distribution can be complex. This is why Dose Volume Histograms (DVH) are also used to evaluate the design quantitatively. An example of a DVH is shown in Figure 2.12. On a DVH, the horizontal axis represents the normalised dose (in percentage) and the vertical axis represents the volume of the region of interest (in percentage) receiving at least that dose. Ideally, the curve for the target (the prostate in the example in Figure 2.12) should shift to the right, meaning that a large part of the target volume has been heavily irradiated. The organs at risk should remain further to the left, meaning

that they are relatively lightly irradiated. Nevertheless, in this example, a significant part of the bladder and rectum receive a high dose, indicating that the plan may not respect the constraints. DVHs can also be used to formulate quantitative objectives: for example, it may be required that 95% of the target volume should receive at least the prescribed dose, or that 90% of the volume of an organ at risk should remain within the prescribed dose range.

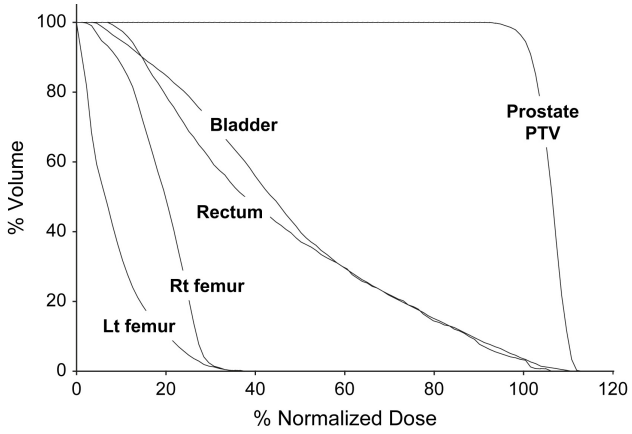


Figure 2.12: Example of a Dose Volume Histogram (DVH). Credits : [18]

The final stage is treatment delivery. The patient is precisely positioned on the treatment table and a proton beam is applied. As explained in section 2.1, fractionating the treatment over several weeks allows for better sparing of normal tissues and improved tumor control due to biological repair mechanisms.

2.5 Monte Carlo Method

In order to design a treatment plan adapted to the prescriptions using the pencil beam scanning method, it is essential to determine the optimum intensities of each beamlet to obtain an ideal dose distribution. To do this, it is necessary to simulate the beamlets passing through the material. Several methods can be used for this, but for the purposes of this thesis, as in many other studies in protontherapy, we will only use the Monte Carlo method.

This stochastic technique consists of simulating the path taken by a large number of protons through matter, taking into account all the physical laws

of interaction (presented in section 2.2.1). The greater the number of protons simulated, the more the dose distribution converges towards reality, by applying the law of large numbers.

The simulated beamlets can then be used in an optimisation process that finally adjusts the intensity that each beamlet will have in order to draw up the final treatment plan [19].

The challenge with Monte Carlo methods, compared to analytical algorithms like the pencil beam algorithm, is that they require significant computation time and considerable resources. On the other hand, analytical algorithms are much faster and more resource-efficient, but they tend to introduce greater uncertainty. This is partly because they handle lateral heterogeneities poorly [20].

Chapter 3

Optimization Strategies for Treatment Planning

Figure 3.1 illustrates the main steps involved in the optimisation process. The first step is to define the orientation and position of the beamlet spots. It is also at this stage that the medical prescriptions to be complied with are established. Next, one of the key steps is to optimise the intensities of the beamlets previously defined to achieve a dose distribution that best complies with the medical prescriptions and the constraints imposed. Once the optimisation has been carried out, the dose distribution corresponding to the intensities obtained is simulated. If the plan is validated by the medical physicist and the doctor, it can be delivered to the patient. If not, some optimisation parameters, such as increasing the weight of an organ at risk, can be modified to improve the quality of the plan [21].

3.1 Cost Function in Radiotherapy

The cost function is the function that our optimisation will minimise. In proton therapy, as in conventional radiotherapy, it evaluates the quality of the dose distribution delivered by all the beamlets. On the one hand, it penalises deviations between the prescribed dose and the dose delivered to the target, and on the other hand, it penalises any excess dose delivered to organs at risk. These two components are represented in the equation 3.1. The deviations are generally squared in order to penalise large deviations more heavily, which favours a more homogeneous dose distribution.

In proton therapy, the cost function generally takes the following form:

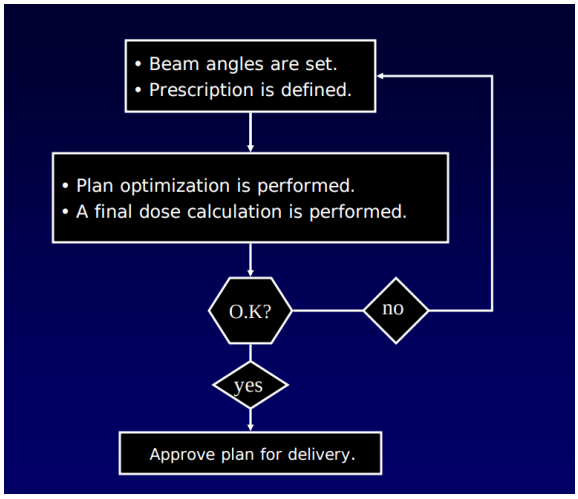


Figure 3.1: Optimisation Planning Process. Credits : [21]

$$\mathrm{f(x)} = \underbrace{\frac{w_{\mathrm{PTV}}}{n_{\mathrm{PTV}}} \sum_{i \in \mathrm{PTV}} (D_i - D_i^{\mathrm{presc}})^2}_{\text{Cost of under/over-dosing tumour (PTV)}} + \underbrace{\frac{w_{OAR}}{n_{OAR}} \sum_{j \in \mathrm{OAR}} \max(D_j - D_j^{\mathrm{tol}}, 0)^2}_{\text{Cost of overdosing a single organ at risk (OAR)}} \tag{3.1}$$

- $f(x)$: global cost function to be minimized where x corresponds to the intensity of the beamlets.
- w_{PTV}; w_{OAR}: weighting factor associated with the planning target volume (PTV) and the organ at risk (OAR) respectively.
- D_i; D_j: dose received by voxel $i; j$ in the PTV and OAR respectively.
- D_i^{presc}; D_j^{tol}: prescribed dose for voxel i and maximum tolerated dose for voxel j in the PTV; OAR respectively.
- n_{PTV}; n_{OAR}: number of voxels in PTV; OAR respectively.

In some cases, other types of objective may be used [22] :

- Biological objectives: for example, Tumor Control Planning (TCP), which represents the probability that the treatment will eliminate all the tumor cells in the target, or Normal Tissue Complication Probability (NTCP), which represents the probability of complication for normal tissues.

- Objectives based on the Dose Volume Histogram: for example, ensuring that at least 95% of the target receives the intended dose, or limiting the amount of radiation received by organs at risk.

3.2 Gradient Descent

The aim of this method is to minimise a cost function f(x). As a reminder, since we are using the pencil beam scanning technique, each beam consists of numerous narrow beamlets. Each beamlet has its own weight (i.e., intensity), spot position, and direction. In proton therapy, the variable "x" of the cost function corresponds to the weights of the beamlets. The aim is to find the ideal weights that minimise the cost function [4][23].

3.2.1 Fundamentals of Gradient Descent

Each beamlet is first assigned arbitrary initial weights, forming the starting point of the optimisation process. The algorithm then evaluates the cost function based on these weights and updates them in the direction opposite to the gradient, as this is the direction in which the cost function decreases most rapidly [23].

Once the new state has been obtained, the algorithm checks whether the clinical or planning objectives have been achieved. If not, the process is repeated by recalculating the gradient from the current state. The optimisation continues until the stopping criteria are met, indicating that the solution is sufficiently close to the optimum. At each iteration, we have [23] :

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

- x_k : Vector of beamlet weights at iteration k.
- η : The learning rate.
- $\nabla f(x_k)$: Gradient of the cost function f(x) evaluated at x_k, i.e.

$$\nabla f(x_k) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

One of the limitations of this method lies in the calculation of these gradients, which can sometimes be very costly in terms of resources and/or time. In addition, as we mentioned earlier, gradient descent ensures that we converge towards a local minimum, but not necessarily to the global minimum.

To be sure of reaching a global minimum, the cost function must be a 'convex' function, meaning it has a single minimum and no local traps.

In radiotherapy, we will therefore use a convex cost function, which ensures that we converge towards the global minimum.

3.2.2 Stochastic Gradient Descent

When there is a lot of data, calculating the full gradient is very expensive and not always feasible in practice [23]. This issue is particularly significant in proton therapy, where treatment plans typically consist of thousands of beamlets. Since each beamlet has an associated weight, a large number of gradient computations is required. The problem becomes even more critical when high precision is needed, as this requires simulating beamlets over a large number of voxels, making matrix operations increasingly computationally expensive.

To overcome this problem, we approximate the gradient using a subset of the data. This gives us [23]:

$$x_{k+1} = x_k - \eta \nabla f_{i_k}(x_k)$$

where f_{i_k} is the cost function evaluated over a randomly chosen subset i_k [23]. In the beamlet free algorithm we implemented, this subset is made up of pairs of beamlets used to approximate their respective gradient, in contrast to the beamlet based algorithm which uses all the beamlets. Updating the weights is therefore much faster, but more approximate. Stochastic gradient descent can therefore be more efficient when working with large datasets, because it consumes far fewer computational resources. However, it tends to introduce more noise and will tend to converge less smoothly compared to standard gradient descent. This behavior is illustrated by the figure 3.2. Since the gradient is only approximated, it will generally required more iterations to reach a satisfactory solution.

One important point to highlight is the relevance of using stochastic gradient descent in proton therapy. Indeed, when applying a Monte Carlo algorithm with only a limited number of sampled particles (forming a batch), we obtain only a partial estimate of the impact each beamlet has on the dose distribution. Each particle batch used to simulate a beamlet can be considered an i.i.d. (independent and identically distributed) sample that is, drawn independently and according to the same probability distribution as the others. In this context, using stochastic gradient descent becomes partic-

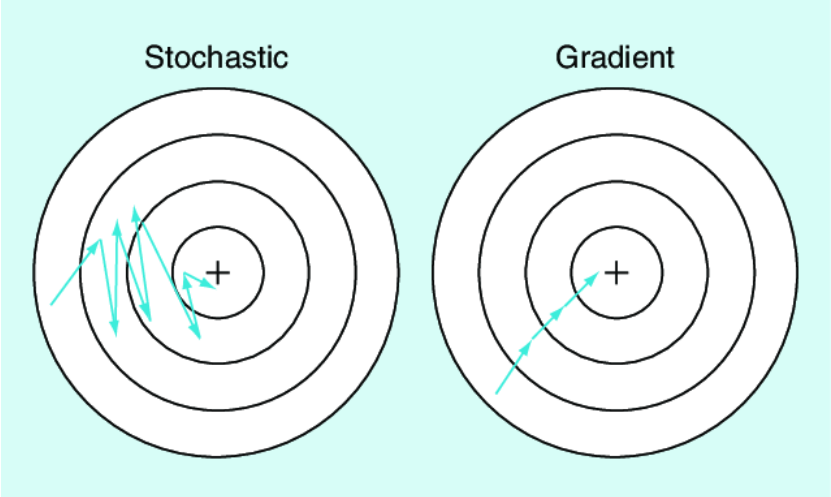


Figure 3.2: Stochastic gradient descent compared with standard gradient descent. Credits : [24]

ularly relevant: if all simulated batches are i.i.d. for each beamlet, then the expected value of the cost function gradient computed over a given particle subset i is equivalent to the gradient of the global cost function.

$$\mathbb{E}[\nabla f_i(x)] = \nabla f(x)$$

We can therefore say that, on average, the estimated gradient will point in the correct direction to minimize the cost function, which ensures convergence. In other words, the average of all partial dose estimates provides a reliable approximation of the final dose distribution. This explanation illustrates how a concept originating from machine learning can be effectively applied to proton therapy [25].

3.2.3 Beamlet-Based Optimization

The beamlet-based algorithm is a treatment planning method for conventional radiotherapy or proton therapy, whose goal is to optimize the dose distribution to make the treatment as effective as possible. This method involves dividing each beam into smaller elements named "beamlets". Each of these beamlets has its own energy and intensity, so they can be combined to modulate the dose more precisely compared to the original beam. The aim of this algorithm is to optimize the weights (i.e. the intensity) of each beamlet to achieve a dose distribution that meets clinical requirements.

Principle

The basic principle of the beamlet-based algorithm is to calculate the dose to each voxel by linearly summing the dose contributions from each beamlet weighted by their intensities. The total dose received by a voxel is given by the following formula:

$$d_i = \sum_j B_{ij} x_j$$

where :

- d_i is the total dose to voxel i.
- B_{ij} is the dose contribution from beamlet j to voxel i.
- x_j is the weight of the beamlet j.

The dose influence matrix ($\mathbf{B}$) is therefore essential, as it allows us to determine the dose received in each voxel. However, since it represents the dose deposited by each beamlet in each voxel, its dimensions are "number of voxels" by "number of beamlets". Its large size makes it computationally and memory intensive to use.

Steps in the algorithm:

Pre-calculation of the influence matrix: The algorithm starts by computing the dose delivered by each beamlet to every voxel in the regions of interest. These dose contributions constitute the influence matrix $\mathbf{B}$ (of size number of beamlets $\times$ number of voxels).

Optimising beamlet intensities: The algorithm iteratively adjusts the weights of each beamlet, aiming to deliver a uniform dose to the tumor while minimising irradiation of organs at risk, by following the opposite gradient direction. The gradients are computed using the previously calculated influence matrix.

Total dose calculation: Once the beamlet weights have been optimized, the total dose of the plan is computed, and the quality of the plan is evaluated based on the respect of the clinical requirements. If the results are not satisfactory, we can adjust certain optimisation parameters, such as the weights assigned to the objective function for organs at risk and the target, and rerun the optimisation to improve the treatment plan.

Strengths and limitations

The beamlet-based algorithm is highly accurate and well-suited for complex cases requiring robust optimisation. However, the dose influence matrix can become enormous in size, making it challenging to store and process. Furthermore, the computations demand significant computational resources, especially when using high-precision Monte Carlo simulations, which simulate the passage of numerous protons through the material.

3.2.4 Beamlet-Free Optimization

As mentioned earlier, the beamlet-based algorithm is an extremely powerful tool for calculating effective treatment plans. However, one of its main limitations is the high computational cost required to simulate the beamlets and optimise the treatment. This means the beamlet-based algorithm performs best when there is ample time available before treatment delivery. Conversely, it is relatively inefficient for real-time or near-real-time treatments. This significant drawback forces patients to undergo medical examinations, including imaging, several hours or even days prior to treatment. As a result, organs at risk may not be in exactly the same position, which can greatly reduce the treatment's effectiveness. Many organs are mobile and can shift slightly within the body. Furthermore, the beamlet-based algorithm does not support motion tracking, a treatment technique that involves "filming" the tumour in real time to monitor its movement during treatment. This approach is especially beneficial for tumours located near rapidly moving organs, such as the lungs.

The beamlet-free algorithm is an alternative approach to the beamlet-based algorithm that aims to overcome these limitations. The goal of this new method is to generate treatment plans more quickly, while maintaining sufficient accuracy to satisfy clinical constraints. This could allow patients to be treated soon after clinical examinations, and even make motion tracking feasible.

A key aspect of this approach is that the dose influence matrix ($\mathbf{B}$) is no longer computed. This eliminates the need to keep in memory all beamlets, which was necessary to build the $\mathbf{B}$ matrix, hence the term "beamlet-free." In the beamlet-free algorithm (BFA), only two beamlets need to be stored at a time, resulting in a significant reduction in resource usage, including RAM, especially when handling a large number of beamlets, as illustrated in Figure 3.3.

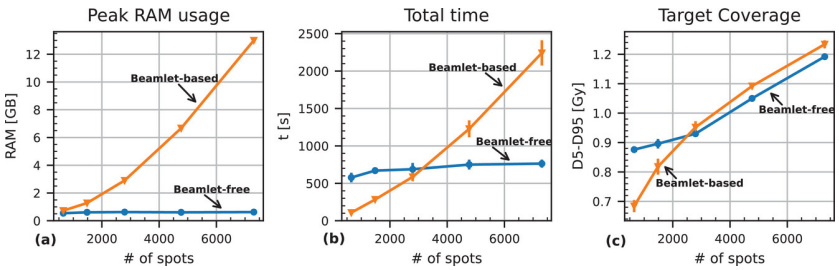


Figure 3.3: Comparison of beamlet-based and beamlet-free algorithms in terms of memory usage, computation time, and target coverage. Credits : [4]

In addition, instead of performing a single highly accurate Monte Carlo simulation that requires a substantial amount of computing time, as done with the beamlet-based algorithm, we will iteratively perform a much larger number of less precise simulations. This approach allows us to stop the optimisation as soon as clinical constraints are satisfied, potentially simulating significantly fewer protons than the beamlet-based algorithm.

Details of the beamlet-free algorithm, such as the gradient calculation, are presented in the next chapter, "Materials and Methods."

Proof of concept

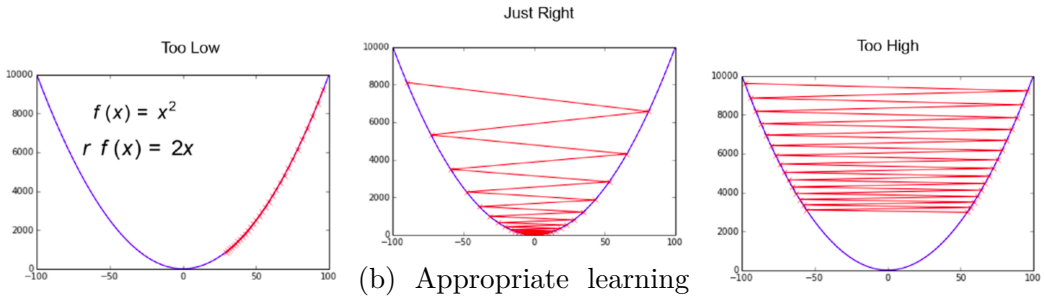
The aim of this thesis is to establish a proof of concept for the beamlet-free algorithm, focusing primarily on the feasibility and convergence of the method. This exploratory phase is based on a Python implementation that allows us to efficiently observe and analyse the results. The Monte Carlo simulations are performed in C for better performance, but the communication between the C and Python environments introduces a considerable time overhead. The beamlet-free algorithm requires a large number of Monte Carlo simulations, which significantly increases computing time. Our goal is to produce a working version and analyse its results, but, due to the limitations described above, we will not be able to compare the computational cost of this method with that of the beamlet-based algorithm.

The results obtained during this exploration will then enable the development of a more optimized version in C, eliminating the overhead from the Python-C interaction.

3.3 Learning Rate Selection

The learning rate (η) is a hyperparameter in optimisation algorithms that determines how quickly the beamlet weights are updated at each iteration [23].

Figure 3.4 shows a simple example of how the learning rate affects convergence, using the cost function $f(x) = x^2$. If the learning rate is too low, the algorithm will converge, but much too slowly (fig 3.4a). Conversely, if the learning rate is too high, the algorithm may diverge without ever reaching the minimum (fig 3.4c).



(a) Learning rate too low (b) Appropriate learning rate (c) Learning rate too high

Figure 3.4: Effect of the learning rate on convergence. Credits : [23]

To avoid choosing a learning rate that is too small or too large, it's interesting to use a degressive learning rate. The algorithm begins with a relatively high learning rate to quickly approach towards the optimum, and the rate is gradually reduced to enable more precise convergence.

Chapter 4

Materials and Methods

4.1 Materials

4.1.1 MCsquare

The MCsquare (Many core Monte Carlo) software was used to simulate the beamlets. One of its main advantages is its parallel architecture, which considerably speeds up Monte Carlo simulation calculations. MCsquare uses a class II condensed history transport algorithm. This type of algorithm aims to group together multiple successive physical interactions into a single simulation step. In the case of class II, the agorithm groups minor collisions together but treats hard collisions individually. This approach offers a good balance between computational speed and accuracy. Simulations performed with MCsquare show a difference of less than 2% compared with those obtained with GEANT4, a reference algorithm that is extremely accurate but much slower to run [19][26].

4.1.2 OpenTPS

The optimisation was implemented in OpenTPS, an open source software package for optimising treatment plans. This software allows us to use the beamlets simulated in MCsquare [27].

4.1.3 Patient Imaging Data

In this thesis, all simulations were conducted on two different cases: a simple scenario, which allowed for extensive testing to analyze the overall performance of the algorithm, and a realistic, complex patient case that required

significantly more computational resources.

Realistic tests were conducted on DICOM data using CT images from real patients. The organs at risk potentially intersected by the beams include the right lung, the heart, and the esophagus. As shown on Figure 4.1

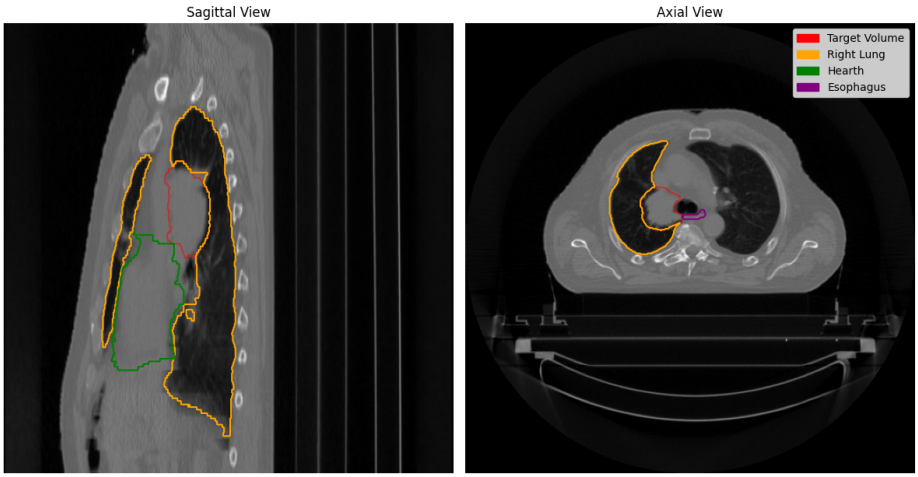


Figure 4.1: CT scan of the patient highlighting the organs at risk and the target volume

To ensure the proper functioning of our algorithm, voxels that were shared between the tumor and the OARs were removed from the OARs, while still being included in the target volume. Figure 4.3a illustrates this overlap removal for the right lung.

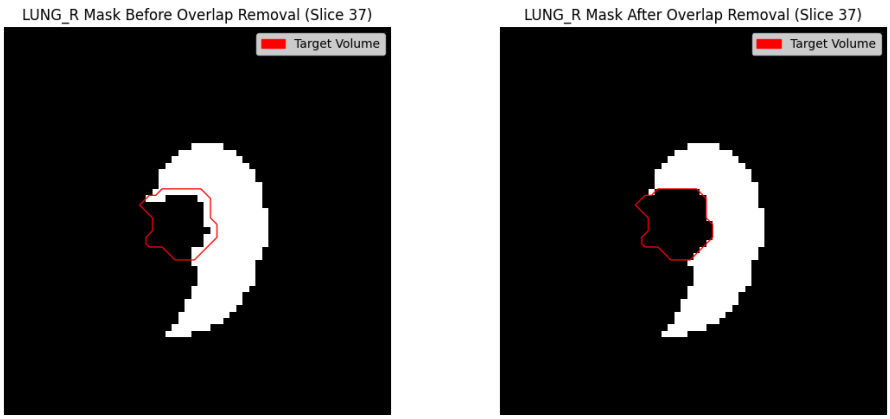


Figure 4.2: Right Lung Mask Before and After Overlap Removal

When a tumor is located close to the esophagus, as is the case in our study, treatment protocols recommend fractionating the uniform dose of 8 Gy into six separate fractions [28].

4.2 Analytical Gradient Calculation

The following section presents the mathematical development leading to the gradients used in optimisation. This development was developed by our supervisor John Aldo Lee, in an internal technical report entitled 'Beamlet-free Monte Carlo plan optimization revisited' [29].

We begin by defining the cost function, which aims to achieve a uniform dose distribution within the target volume. This cost is expressed as:

$$D_{\mathbf{W}} = \|\mathbf{d}^{pr} - \mathbf{B}\mathbf{x}\|^2_{\mathbf{W}} \tag{1}$$

The goal is to minimize $\mathbf{D_W}$ such that the delivered dose $\mathbf{Bx}$ matches the prescribed dose $\mathbf{d}^{pr}$. Here, $\mathbf{B}$ represents the matrix containing the dose contributions of each beamlets, $\mathbf{x}$ the corresponding weights of each beamlets, and $\mathbf{W}$ a weighting matrix reflecting the relative importance of different objectives for each region of interest.

With the cost function defined, it becomes possible to apply gradient descent to minimize it. The gradient of the cost function with respect to the beamlet weights $\mathbf{x}$ can be computed using the following expression:

$$\nabla_{\mathbf{x}} D_{\mathbf{W}} = 2\mathbf{B}^\top \mathbf{W}(\mathbf{B}\mathbf{x} - \mathbf{d}^{\mathrm{pr}}) \tag{2}$$

The weights are then updated iteratively according to the rule: $\mathbf{x} \leftarrow \mathbf{x} - \eta \nabla_{\mathbf{x}} D_{\mathbf{W}}$ where η denotes the learning rate.

Now consider the case where the beamlets are not yet accurately estimated, meaning that only a small subset of particles has been simulated. This situation is analogous to stochastic gradient descent, as each beamlet is based on different data samples drawn from the same underlying random process.

One first approach is to reformulate the gradients as follows:

$$\nabla_{\mathbf{x}} D_{\mathbf{W}} = 2\mathbf{B}_p^\top \mathbf{W}(\mathbf{B}_p\mathbf{x} - \mathbf{d}^{\mathrm{pr}}) \tag{3}$$

where each beamlet is simulated on a particle batch p. A key limitation of this approach is that if the batch size is too small, the gradient estimate becomes highly noisy, which may lead to scaling issues.

To mitigate these problems and avoid storing the full beamlet matrix, it is preferable to construct a total dose $\mathbf{d}^Q$ by adding together simulations of individual beamlets across multiple batches. The total dose $\mathbf{Bx}$ is thus obtained by summing the dose contributions of each beamlet over all batches. Given that the dose contribution of beamlet i is $\mathbf{b}_i x_i$ and that its dose over all batches Q_i of beamlet i is expressed as $\sum_{p=1}^{Q_i} \mathbf{b}_{ip}$, the total dose over all batches Q of all beamlets is given by:
$\mathbf{d}^Q = \sum_{i=1}^{B} \sum_{p=1}^{Q_i} \mathbf{b}_{ip} x_i$

Although we have not implemented this in our code, Q_i can be larger for beamlets deemed more important and smaller for less relevant beamlets.

Regarding the gradient calculation, multiplication by $\mathbf{B}$ is not feasible since only $\mathbf{b}_{ip}$ at the p^{th} batch is known. That's why we use the total dose $\mathbf{d}^Q$ instead.

To facilitate understanding, we first develop equation 2, which involves $\mathbf{B}$. We will then apply a similar reasoning using $\mathbf{d}^Q$.

$$\begin{aligned}
\mathbf{D_W} &= \|\mathbf{d}^{pr} - \mathbf{Bx}\|^2_{\mathbf{W}} && (4)\\
&= (\mathbf{d}^{pr} - \mathbf{Bx})^T \mathbf{W} (\mathbf{d}^{pr} - \mathbf{Bx}) && (5)\\
&= \|\mathbf{d}^{pr}\|^2_{\mathbf{W}} - 2(\mathbf{Bx})^T \mathbf{W} \mathbf{d}^{pr} + (\mathbf{Bx})^T \mathbf{W} (\mathbf{Bx}) && (6)\\
&= \|\mathbf{d}^{pr}\|^2_{\mathbf{W}} - 2\mathbf{x}^T \mathbf{B}^T \mathbf{W} \mathbf{d}^{pr} + \mathbf{x}^T (\mathbf{B}^T \mathbf{W} \mathbf{B}) \mathbf{x}. && (7)
\end{aligned}$$

The following steps follow the exact same reasoning, but with $\mathbf{Bx}$ replaced by $\mathbf{d}^Q$:

$$\begin{aligned}
\mathbf{D_W} &= \|\mathbf{d}^{pr} - \mathbf{d}^Q\|^2_{\mathbf{W}} && (8)\\
&= \left\| \mathbf{d}^{pr} - \sum_{i=1}^{B} \sum_{p=1}^{Q_i} \mathbf{b}_{ip} \frac{x_i}{Q_i} \right\|^2_{\mathbf{W}} && (9)
\end{aligned}$$

where B is the number of beamlets, and the division by Q_i ensures that the dose magnitudes of the beamlets are comparable, even if they were simulated with different numbers of protons.

By expanding the expression, we obtain:

$$\mathbf{D_W} = \left(\mathbf{d}^{pr} - \sum_{i=1}^{B}\sum_{p=1}^{Q_i} \mathbf{b}_{ip}\frac{x_i}{Q_i}\right)^T \mathbf{W} \left(\mathbf{d}^{pr} - \sum_{j=1}^{B}\sum_{q=1}^{Q_j} \mathbf{b}_{jq}\frac{x_j}{Q_j}\right) \tag{10}$$

The transition from equation (9) to (10) is made possible by renaming the beamlet index i to j in the second term, and similarly renaming the batch index p to q, without any loss of generality.

$$\begin{aligned}\mathbf{D_W} &= \|\mathbf{d}^{pr}\|^2_{\mathbf{W}} \\ &\quad - \sum_{i=1}^{B}\frac{x_i}{Q_i}\sum_{p=1}^{Q_i}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} - \sum_{j=1}^{B}\frac{x_j}{Q_j}\sum_{q=1}^{Q_j}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr} \\ &\quad + \left(\sum_{i=1}^{B}\sum_{p=1}^{Q_i}\mathbf{b}_{ip}\frac{x_i}{Q_i}\right)^T \mathbf{W} \left(\sum_{j=1}^{B}\sum_{q=1}^{Q_j}\mathbf{b}_{jq}\frac{x_j}{Q_j}\right)\end{aligned} \tag{11}$$

Note that the first term is independent of x.

By applying the general formula below to the second term,

$$\sum_{i=1}^{N_1} f(i) + \sum_{j=1}^{N_2} g(j) = \sum_{i,j=1}^{N_1,N_2}\left(\frac{f(i)}{N_2} + \frac{g(j)}{N_1}\right)$$

We obtain the following rearranged expression:

$$- \sum_{i=1}^{B}\frac{x_i}{Q_i}\sum_{p=1}^{Q_i}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} - \sum_{j=1}^{B}\frac{x_j}{Q_j}\sum_{q=1}^{Q_j}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr} \tag{12}$$

$$= -\frac{1}{B}\sum_{i,j=1}^{B}\left(\frac{x_i}{Q_i}\sum_{p=1}^{Q_i}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} + \frac{x_j}{Q_j}\sum_{q=1}^{Q_j}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr}\right) \tag{13}$$

$$= -\frac{1}{B}\sum_{i,j=1}^{B}\sum_{p,q=1}^{Q_i,Q_j}\left(\frac{x_i}{Q_iQ_j}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} + \frac{x_j}{Q_iQ_j}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr}\right) \tag{14}$$

where the batches associated with each pair of beamlets are now matched two by two.

The last term can then be rewritten as:

$$\sum_{i,j=1}^{B}\left(\frac{x_i}{Q_i}\sum_{p=1}^{Q_i}\mathbf{b}_{ip}\right)^T \mathbf{W}\left(\frac{x_j}{Q_j}\sum_{q=1}^{Q_j}\mathbf{b}_{jq}\right) \tag{15}$$

$$= \sum_{i,j=1}^{B}\sum_{p,q=1}^{Q_i,Q_j}\left(\frac{x_ix_j}{Q_iQ_j}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{b}_{jq}\right) \tag{16}$$

where the summations over the beamlets and over the batches are explicitly factored out.

By combining the three terms, we obtain:

$$\begin{aligned}\mathbf{D_W} &= \|\mathbf{d}^{pr}\|^2_{\mathbf{W}} \\ &- \frac{1}{B}\sum_{i,j=1}^{B}\sum_{p,q=1}^{Q_i,Q_j}\left(\frac{x_i}{Q_iQ_j}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} + \frac{x_j}{Q_iQ_j}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr}\right) \\ &+ \sum_{i,j=1}^{B}\sum_{p,q=1}^{Q_i,Q_j}\left(\frac{x_ix_j}{Q_iQ_j}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{b}_{jq}\right)\end{aligned} \tag{17}$$

Since the summations over the batches appear explicitly in both the second and the third terms, we can isolate a specific pair of batches (p,q). This allows us to extract the summations $\sum_{p,q=1}^{Q_i,Q_j}$ as well as the scaling factor $\frac{1}{Q_iQ_j}$. The cost function for a given beamlet pair (i,j) and batch pair (p,q) can thus be computed as follows:

$$(\mathbf{D_W})_{(p,q)} = \|\mathbf{d}^{pr}\|^2_{\mathbf{W}} - \frac{1}{B}\sum_{i,j=1}^{B}(x_i\mathbf{b}_{ip}^T + x_j\mathbf{b}_{jq}^T)\mathbf{W}\mathbf{d}^{pr} + \sum_{i,j=1}^{B}x_ix_j\mathbf{b}_{ip}^T\mathbf{W}\mathbf{b}_{jq} \tag{18}$$

Finally, we can compute the gradients for a pair of beamlets (i,j) simulated respectively with the batches of particles (p,q), as follows:

$$\frac{\partial(\mathbf{D_W})_{(p,q)}}{\partial x_i} = -\frac{1}{B}\sum_{j=1}^{B}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} + \sum_{j=1}^{B}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{b}_{jq}x_j \tag{21}$$

$$= -\mathbf{b}_{ip}^T\mathbf{W}\mathbf{d}^{pr} + \sum_{j=1}^{B}\mathbf{b}_{ip}^T\mathbf{W}\mathbf{b}_{jq}x_j \tag{22}$$

$$\frac{\partial(\mathbf{D_W})_{(p,q)}}{\partial x_j} = -\frac{1}{B}\sum_{i=1}^{B}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr} + \sum_{i=1}^{B}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{b}_{ip}x_i \tag{23}$$

$$= -\mathbf{b}_{jq}^T\mathbf{W}\mathbf{d}^{pr} + \sum_{i=1}^{B}\mathbf{b}_{jq}^T\mathbf{W}\mathbf{b}_{ip}x_i \tag{24}$$

The gradient equations are specific to a uniform dose objective with a cost function of the form $\|\mathbf{d}^{pr} - \mathbf{B}\mathbf{x}\|^2_{\mathbf{W}}$. Adding an organ at risk complicates the model, as it would require using a cost function of the form $\|max(\mathbf{d}^{pr} - \mathbf{B}\mathbf{x}, 0)\|^2_{\mathbf{W}}$. We did not carry out the derivation for such a cost function.

4.3 Practical Implementation

The core of our work involved implementing the beamlet-free algorithm in a Python environment.

The following section show the main steps of our beamlet-free algorithm implementation.

4.3.1 Pseudo Code

The basic structure of the algorithm is outlined in the pseudocode below. First, the main plan parameters need to be defined: the physician's dose prescription, the number of protons simulated per beamlet during optimization, the number of iterations, the learning rate, ...

Once the plan parameters are set, the following elements are initialized: $\mathbf{x}$ (the weight matrix), $\mathbf{W}$ (the weight values linked to the different dose objectives), $\mathbf{d}^{pr}$ (the prescribed dose objectives) and $\mathbf{C}_W$ (the correlation matrices for each ROI). These will be described in detail later in this thesis. Their purpose is to store information about the beamlets used during the optimization process, in the form of correlations between pairs of beamlets in the various regions of interest.

The optimization process then begins. At each iteration, two random beamlets, generated by MCsquare with a limited number of protons, are selected. The weights associated with these beamlets are updated based on their respective gradients, following the optimization rules.

Once the optimization ends (either after the maximum number of iterations is reached or an early stopping criterion is met), a dose map can be generated using MCsquare. This time, beamlets are simulated with a large number of protons to enable accurate evaluation of the treatment plan. These high-precision beamlets are only used at the evaluation stage, once the optimal weights have already been determined, they do not influence the optimization itself.

Some deviations from the pseudocode will be discussed later in this thesis. However, the core idea behind the algorithm remains the same.

Algorithm 1 Beamlet-free algorithm

Initialization

Import CT images
Define plan parameters
Initialize the weight matrix, the correlation matrices $\mathbf{C}_W$ for each ROI, the $\mathbf{W}$ matrix and the $\mathbf{d}^{pr}$ matrix

Optimization

for $i \leftarrow 1$ **to** *nb_iterations* **do**
    Simulate two random, coarse beamlets generated by MCsquare
    Add their correlations to the correlation matrices $\mathbf{C}_W$
    Compute gradients corresponding to the two beamlets
    Update the weight matrix $\mathbf{x}$
    Remove the beamlets
end for

Evaluation

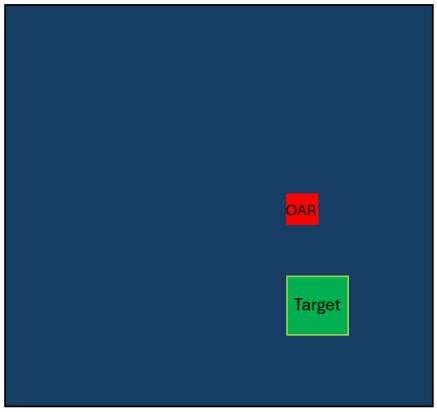
Generate a dose map using the final weights by calling MCsquare (with precise beamlets) — optional

4.3.2 Initialization of Clinical Constraints

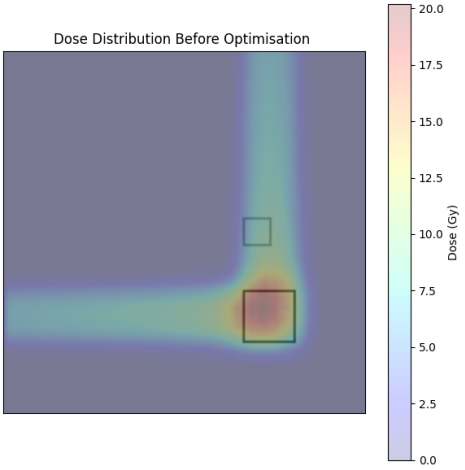
The first part of the algorithm consists of initialising the clinical constraints. To carry out our first tests, we created a simple case illustrated in Figure 4.3a, which contains a target and a single organ at risk. The purpose of this example is to check that a beam emitted from the top of the target naturally reduces its intensity as it passes through the OAR. A second beam, emitted from the left, is also simulated. If the algorithm is working correctly, it should compensate for the under-dosage in the part of the target located behind the OAR to obtain the most homogenous dose possible. The dose prescription matrix is of size (150,150,150), and we have included :

- A target with a prescribed homogeneous dose of 20 Gy.
- An organ at risk with a maximum tolerated dose of 5 Gy.
- Two beams: one from the left side, the other from the upper side.

The OpenTPS software allows the weights to be initialized with values other than zero. This option was used here to help guide the optimisation process toward a satisfactory solution from the start. Figure 4.3b shows the dose distribution before optimisation begins.



(a) Dose prescription matrix example



(b) Dose distribution before optimization

The cost function for this example can be :

$$COST = w_{TV}\frac{1}{n_T}\sum_{i\in T}(d_i - 20)^2 + w_{OAR}\frac{1}{n_{OAR}}\sum_{j\in T}(max(d_j - 5, 0))^2$$

where :

- w_{TV} and w_{OAR} represent the weights associated with the target volume (TV) and the organ at risk (OAR), respectively.
- n_T and n_{OAR} are the number of voxels in the TV and OAR, respectively.
- d_i and d_j correspond to the dose received in voxels i and j.
- 20 and 5 represent the prescribed dose values for the target and the OAR, respectively.

However, as previously mentioned in section 4.2, we were unable to determine the analytical gradient of a Dmax-type cost function, such as the function $\frac{w_{OAR}}{n_{OAR}}\sum_{j\in OAR}(max(d_j - 5, 0))^2$. For this reason, we were forced to limit ourselves to a Duniform-type formulation for the OAR. The issue with Duniform is that, if we request a uniform dose of 5 Gy in the OAR, the algorithm tends to artificially increase the dose in the under-irradiated voxels of the OAR to meet this uniform target, which goes against the goal of sparing healthy tissue. We therefore chose to request a Duniform of 0 Gy, not with the unrealistic aim of enforcing a strictly zero dose, but rather to encourage the algorithm to minimize the dose in the OAR as much as possible, while

preserving differentiability. This approximately corresponds to a Dmax minimization, since any nonzero dose is penalized. To prevent the algorithm from focusing too much on the OARs, which would otherwise incur a high cost even when the dose is below the acceptable threshold, we assigned them a lower relative weight than the tumor volume (TV), in order to prioritize tumor coverage during optimization.

Consequently, our cost function is defined as follows:

$$COST = \frac{w_{TV}}{n_{TV}} \sum_{i \in TV} (d_i - 20)^2 + \frac{w_{OAR}}{n_{OAR}} \sum_{j \in OAR} (d_j - 0)^2$$

4.3.3 Beamlet Simulation

At each iteration of our algorithm, we need to simulate 2 beamlets using MCsquare. However, since the communication between Python and MCsquare takes a significant amount of time, it would be extremely slow to send requests to MCsquare at every iteration. The trick to overcome this limitation is to precompute all beamlets, which we then keep in memory, and randomly select 2 at each iteration. This is in direct contradiction with the beamlet-free algorithm, as one of its main goals is precisely to avoid storing all beamlets in memory. Nevertheless, we allow ourselves this exception because our implementation is solely intended as a proof of concept to demonstrate the feasibility and convergence of the method, and is not designed to optimize performance. Moreover, this is not a limitation for a future optimized C implementation, which could simply simulate a pair of beamlets directly with MCsquare, without the overhead caused by communication between our implementation and MCsquare.

To avoid overusing the same beamlets, we simulate new beamlets every 1000 iterations. Since our basic simulation consists of 176 beamlets, this means that each beamlet is used on average 11.36 times. This is a good compromise: it's not frequent enough for the optimization to fully adapt to the beamlets, but still sufficient to save considerable computation time and allow optimization to be performed within reasonable time frames.

Therefore, our workaround should not significantly affect the results regarding the convergence of the algorithm.

4.3.4 Computation of the Gradient

After selecting a beamlet pair, the next step is to numerically compute their respective gradients, which are presented in analytical form in section 4.2:

$$\frac{\partial(\mathbf{Dw})_{(p,q)}}{\partial x_i} = -\mathbf{b}_{ip}^T \mathbf{W} \mathbf{d}^{pr} + \sum_{j=1}^{B} \mathbf{b}_{ip}^T \mathbf{W} \mathbf{b}_{jq} x_j$$
$$\frac{\partial(\mathbf{Dw})_{(p,q)}}{\partial x_j} = -\mathbf{b}_{jq}^T \mathbf{W} \mathbf{d}^{pr} + \sum_{i=1}^{B} \mathbf{b}_{jq}^T \mathbf{W} \mathbf{b}_{ip} x_i$$

The two gradients have a similar structure, differing only in their association with their respective beamlet. Therefore, we will detail the derivation of only one of them, as the other can be obtained in an analogous way. To facilitate understanding, the gradient can be decomposed into two terms:

$$\frac{\partial(\mathbf{Dw})_{(p,q)}}{\partial x_j} = \underbrace{-\mathbf{b}_{jq}^T \mathbf{W} \mathbf{d}^{pr}}_{\text{First term}} + \underbrace{\sum_{i=1}^{B} \mathbf{b}_{jq}^T \mathbf{W} \mathbf{b}_{ip} x_i}_{\text{Second term}}$$

It is important to note that the matrices $\mathbf{b}_{jq}$, $\mathbf{b}_{ip}$, $\mathbf{W}$, and $\mathbf{d}^{pr}$ are all of the same dimensions, corresponding to the size of the CT scan, which allows us to compute their dot product.

First Term

The calculation of the first term is relatively straightforward. Since $\mathbf{W}$ is a mask that selects only the voxels belonging to the region of interest (ROI), the product $\mathbf{b}_{jq}^T \mathbf{W} \mathbf{d}^{pr}$ amounts to computing a dot product between the dose distribution matrix delivered by beamlet j and the prescribed dose matrix, but restricted to the ROI.

In other words, this term measures the correlation between the dose distribution of beamlet j and the prescription in voxels of the ROI. If the dose delivered by this beamlet is high in voxels where the prescription is also high, this dot product will be large.

For the OARs, we can set this term to zero, since the prescribed dose is 0 Gy, as explained in section 4.3.2.

Second Term

The second term of the gradient, defined as $\sum_{j=1}^{B} \mathbf{b}_{ip}^T \mathbf{W} \mathbf{b}_{jq} x_j$, corresponds to the sum of dot products between $\mathbf{b}_{jq}$ and each of the other beamlets within the ROI, multiplied by their respective weights (i.e., their intensities). Since, by the nature of the beamlet-free algorithm, we cannot store all beamlets in

memory, we construct a correlation matrix $\mathbf{C}_W$ for each ROI. These matrices have dimensions (number of beamlets $\times$ number of beamlets) and store the correlations between all beamlets within each ROI. This approach is significantly less memory-intensive than storing the full influence matrix $\mathbf{B}$, which has dimensions (number of beamlets $\times$ number of voxels). As the number of voxels typically far exceeds the number of beamlets, this reduction in size leads to substantial memory savings. Furthermore, $\mathbf{C}_W$ is a symmetric matrix, so in principle, it would be possible to store only a triangular version, cutting the memory usage in half. However, we did not implement this optimization in our code.

In our case, we use 421,875 voxels and 176 beamlets, so the memory consumption could be significantly lower with the beamlet-free algorithm than with the beamlet-based approach.

To populate $\mathbf{C}_W$, at each iteration, we add the correlation between the two selected beamlets, i and j, into the matrix $\mathbf{C}_W$ at indices [i, j] and [j, i]. If a beamlet pair has already been selected in previous iterations, the new value is added using a weighted average with the previous observations. For instance, if the pair (i, j) has already been observed three times, the update is done as follows:

$$\mathrm{C}[i, j] \leftarrow \frac{3}{4}\,\mathrm{C}[i, j] + \frac{1}{4} \cdot \mathrm{Correlation}_{i,j},$$

and symmetrically for C[j, i].

This process allows for the progressive refinement of the $\mathbf{C}_W$ matrix approximation, reducing the impact of noise inherent in beamlet simulations with a low number of protons. The more iterations are performed, the more stable and representative the matrix becomes of the true correlations.

Then, to compute $\sum_{i=1}^{B} \mathbf{b}_{jq}^T \mathbf{W} \mathbf{b}_{ip}\, \mathbf{x}_i$, it is sufficient to take the dot product between column i of $\mathbf{C}_W$ and the beamlet weight vector.

Total gradient

Once the two terms are added together, we obtain the stochastic gradient associated with the selected beamlet. This gradient is then used to update the weight x_j of the beamlet according to the standard gradient descent rule:

$$x_j \leftarrow x_j - \eta \cdot \frac{\partial (D_{\mathbf{W}})_{(p,q)}}{\partial x_j}$$

It is interesting to note that, as the dose distribution approaches the prescribed dose, the quantity $\sum_{i=1}^{B} \mathbf{b}_{ip} x_i$, which corresponds to the actual dose distribution, gets closer to $\mathbf{d}^{pr}$. Theoretically, the two terms of the gradient should cancel each other out when the prescribed dose is achieved, although in practice, noise due to beamlet simulation and the stochastic nature of the method can affect this balance.

When dealing with multiple ROIs, it is necessary to have one $\mathbf{C}_W$ matrix per ROI, since the matrix $\mathbf{W}$ is specific to each ROI.

4.3.5 Pre-Filling of the Correlation Matrix $\mathbf{C}_W$

For the beamlet-free algorithm to converge properly, it is essential that the $\mathbf{C}_W$ matrix is adequately populated. Although we can compute the first term of the gradient without issue, the second term requires access to the entire column of $\mathbf{C}_W$ corresponding to the beamlet in question. This second term is crucial because it helps counterbalance the first term. It generally acts to increase the gradients when the weights (x) are too high, that is, when the dose exceeds the prescribed value. This increase in the gradient causes the weights to decrease, leading to a stable equilibrium.

However, when $\mathbf{C}_W$ is not sufficiently populated, the second term remains too weak, causing the algorithm to continuously increase the weights beyond what is necessary, which results in a dose well above the target.

In theory, we expected this matrix to be gradually filled during the iterations, but in practice, it appears to be filling up far too slowly.

Let us consider an example to estimate the number of iterations required to fill the matrix $\mathbf{C}_W$. Suppose the matrix has N distinct entries (i.e., only the upper triangular part including the diagonal, due to symmetry), and at each iteration, one entry and its symmetric counterpart are randomly filled. In this context, the expected number of iterations needed to fill a proportion P of the matrix of size N is given by $N \cdot \left(H_N - H_{N(1-P)}\right)$, where $H_N = \sum_{k=1}^{N} \frac{1}{k}$ [30].

Given that $\mathbf{C}_W$ is a symmetric matrix of size $n \times n$, the total number of unique entries (including the diagonal) is $\mathrm{N} = \frac{n(n+1)}{2}$.

In a relatively simple case with approximately 200 beamlets, we have n = 200, $\mathrm{N} = \frac{n(n+1)}{2} = 20100$. This yields the following table:

Filled Proportion (%)	Required Iterations (Average)
50	13 932
75	27 863
90	46 277
95	60 205
99	92 514

In a more realistic case with around 10000 beamlets, we have n = 10000, N = 50 005 000. This results in the following table:

Filled Proportion (%)	Required Iterations (Average)
50	34 660 824
75	69 321 648
90	115 140 763
95	149 801 583
99	230 281 486

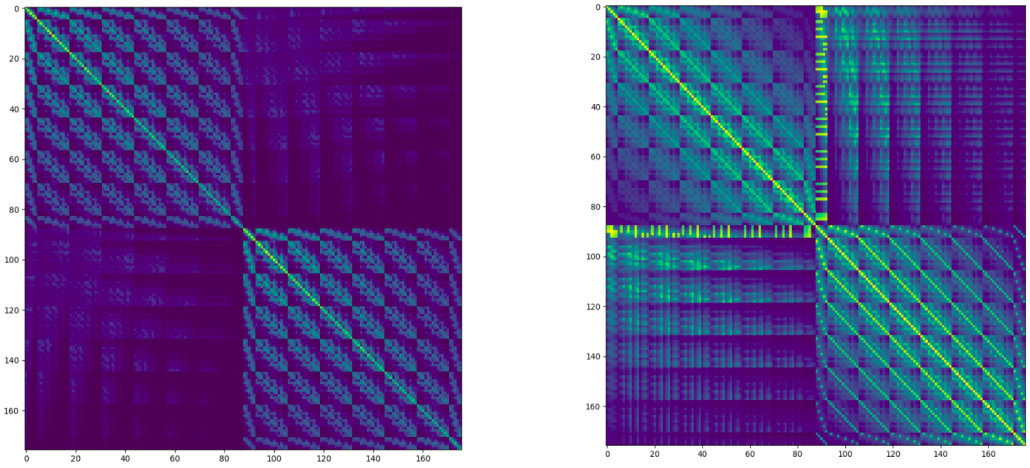
As we can see, even in a simple case, an extremely large number of iterations is required for $\mathbf{C}_W$ to be sufficiently filled. This means that convergence towards the solution only happens after many iterations. In the more complex case, with a realistic number of iterations, $\mathbf{C}_W$ will never be sufficiently populated to allow the algorithm to converge properly.

Solution:

To overcome this problem, we decided to pre-fill the $\mathbf{C}_W$ matrix. This is normally not allowed, as it would require keeping all beamlets in memory, which goes against one of the main advantages of the beamlet-free approach.

Therefore, instead of pre-filling $\mathbf{C}_W$ using all beamlets, we attempted to construct an approximation of $\mathbf{C}_W$. As mentioned in section 4.3.4, $\mathbf{C}_W$ is populated by computing the correlation between two beamlets ($\mathbf{B}$ dot $\mathbf{B}$). So, to estimate $\mathbf{C}_W$ without storing all beamlets, we tried to approximate the correlation between two beamlets based solely on their spot positions and angles.

We took into account not only the distance between their spots, but also whether their paths intersect or whether a spot lies close to the trajectory of the other beamlet. The result of this $\mathbf{C}_W$ approximation is illustrated in Figure 4.4b.



(a) Exact $\mathbf{C}_W$ Matrix (b) Our Approximtion of $\mathbf{C}_W$

Figure 4.4: Comparison Between Exact and Approximated $\mathbf{C}_W$

Although the $\mathbf{C}_W$ approximation we obtained reproduces some patterns present in the exact matrix, the result remains rather limited. Visually, a few similarities can be observed, but the overall structure is far from convincing.

That said, the presence of some recurring patterns suggests that it might theoretically be possible to achieve a better approximation with a more suitable method. Exploring this idea, and in particular the development of a more effective approximation strategy, could be the subject of more in-depth research.

In our simulations, we were not able to use our approximated $\mathbf{C}_W$ as the initial $\mathbf{C}_W$ matrix because it was too far from reality, which would have led to poor results. To be able to run our simulations nonetheless, we chose to pre-fill the $\mathbf{C}_W$ matrix using the exact product of the beamlet matrix, rather than relying on an approximation. Concretely, this means we computed $\mathbf{C}_W$ directly using the full matrix B derived from the beamlets. However, these beamlets were generated using the number of protons from a single iteration, which naturally introduces statistical noise.

Therefore, while the $\mathbf{C}_W$ matrix used is "exact" in terms of mathematical construction, it is based on noisy data, which makes it practically closer to what could be expected from a realistic approximation. We consider that if a method were capable of accurately approximating $\mathbf{C}_W$, it should be able to produce similar results to those obtained with this noisy version.

The development of such an approximation method, capable of efficiently reconstructing $\mathbf{C}_W$ from partial or noisy data, is an interesting challenge. However, it lies beyond the scope of this work and could be considered as a potential future research project.

4.3.6 Simulation Parameters

In our two simulations (simple example and patient), some fairly general parameters had to be defined:

- Spot spacing: the distance between two adjacent spots within the same layer is set to 5 mm and 7 mm in the simple example and patient simulation respectively.
- Layer spacing: the distance between successive layers is also set to 5 mm and 6 mm in the simple example and patient simulation respectively.
- Target margin: this margin is added around the target volume to irradiate a slightly larger area than the visible tumor. It accounts for microscopic disease not visible on imaging and compensates for setup and range uncertainties. The margin is set to 3 mm and 2 mm in the simple example and patient simulation respectively.
- Scoring spacing: set to [2, 2, 2] and [4, 4, 4] in the simple example and patient simulation respectively. This parameter implies that the dose is optimized on a grid with half and one-quarter of the original resolution per spatial dimension, respectively. Nevertheless, the final dose evaluation is performed on the original high-resolution grid. While this approach substantially accelerates the simulation, it comes at the cost of reduced spatial accuracy in the dose distribution and, consequently, in the optimization algorithm.
- Number of beams: We use 2 beams for both examples.
- Learning Rate: In most of our simulations, we used fixed learning rates, except in a dedicated section focused on finding an optimal value. In that section, we explored various types of decreasing learning rates, both linear and exponential, across different value ranges.

The number of spots increases if the spot spacing or layer spacing is reduced, or if the target margin is increased. In our case, we intentionally chose parameters that significantly reduce the number of spots. This results in a loss of precision, but allows us to run our simulations within a reasonable computation time.

4.3.7 Cost Evaluation

To assess the performance and convergence of our algorithm, we need to evaluate the cost function at each iteration to observe its evolution. To do so, we simulated high-precision beamlets, which are used solely for this purpose. These beamlets allow us to accurately compute the dose distribution based on the beamlet weights at each iteration.

All the dose distribution images, cost plots, and other visualizations in this report were generated using these high-precision beamlets, which are not used by the optimization algorithm itself.

In a realistic implementation of the beamlet-free algorithm, this cost would not be computable, as it requires access to the full beamlet matrix in order to calculate the dose distribution, given by $\sum_{i=1}^{B} \mathbf{b}_{ip} x_i$.

4.3.8 Determining a Stopping Criterion

As mentioned above, the cost function cannot be computed in the beamlet-free algorithm. Therefore, it cannot be used as a stopping criterion. The stopping criterion is typically used to stop the algorithm once sufficient convergence has been reached. Usually, this is done by analyzing the evolution of the cost function over iterations. Once the cost no longer decreases significantly, the optimization process can be stopped.

In the case of the beamlet-free algorithm, it is therefore essential to find an alternative criterion to determine when the algorithm has converged sufficiently. To address this, we use the norm of the gradient vector. Indeed, we expect that the closer the algorithm is to the optimum, the smaller the norm of the gradient becomes.

However, we cannot simply average the norms of all the individual gradients we compute. Since we are performing a stochastic optimization, the gradients are highly noisy and may vary significantly, they can be positive or negative for the same beamlet depending on the pairing at each iteration.

To obtain a more accurate estimate of the total gradient of one beamlet, we should sum all the individual gradients corresponding to each pairing with this beamlet first, and then compute the norm of this sum. This way, positive and negative contributions can cancel each other out, and when the algorithm reaches convergence, the norm of the summed gradient of one

beamlet should tend toward zero, even though individual gradients may still be non-zero.

Our strategy to implement this is to divide the iterations into windows of size X, and to evaluate the average gradient norm of each beamlet within each window.

The gradient norm is computed as follows:

1) Initialize a list (or vector) of the same size as the weight vector x, filled with zeros.

2) For each beamlet j, we sum all the gradients associated with it, i.e. all the gradients computed when it was paired with other beamlets during the considered iteration window. We store the result of this sum at index j of the list. This should approximate the gradient of beamlet j using all iterations within the window.

3) Compute the norm (e.g., the ℓ_2-norm) of the resulting list. This norm reflects the overall magnitude of the beamlet gradients.

4) To track the evolution of this norm over time, repeat steps 1–3 by sliding the iteration window forward. A step size of 1 offers maximum precision, while a larger step size can be used to reduce computational cost.

The choice of window size is critical. If the window is too small, the computed gradient norm will be highly noisy and may appear to stagnate even before the cost function reaches a plateau. Conversely, if the window size X, is too large, the norm, computed over last X iterations, will introduce a delay in detecting convergence, making the stopping criterion less responsive to the current state of the algorithm.

Chapter 5

Results

In this section, we present all the results obtained to evaluate the algorithm's performance. Interpretations and explanations of these results are provided in the Discussion (chapter 6). The results were first obtained using the simple example introduced earlier, and then using the patient image case also presented in a previous section.

To assess performance, we varied the number of protons simulated per beamlet, testing values from 32 to 1024 in powers of 2. An additional simulation with 100,000 protons was performed to evaluate the algorithm's behavior at a much finer beamlet granularity. We then examined how different learning rates affect the optimization process. Since it is important to stop the simulation once sufficient convergence is reached, we also analyzed the impact of various stopping criteria. Finally, we compared the beamlet-based and beamlet-free approaches

5.1 Dosimetric Results

To begin, we qualitatively examine the dose distributions generated by the algorithm. Figure 5.1 shows the distributions obtained with 32, 256, and 1024 protons per beamlet. The dose within the target remains consistently close to 20 Gray across all simulations. In contrast, the dose deposited in the organ at risk tends to decrease as the number of simulated protons increases, indicating a potential improvement in sparing healthy tissue.

This qualitative analysis provides initial insight into the algorithm's behavior. A more rigorous quantitative evaluation is presented in the following sections.

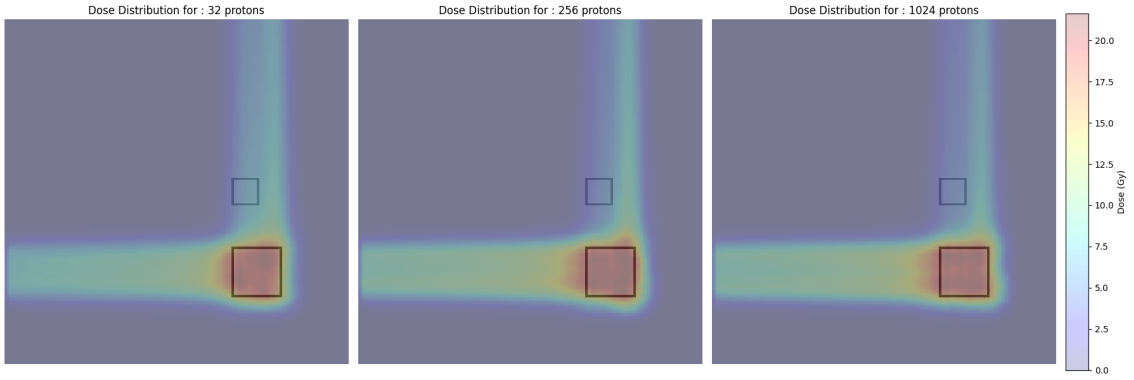


Figure 5.1: Dose distribution for 32, 256 and 1024 protons

5.2 Quantitative Results

5.2.1 Cost Functions

Figure 5.2 shows the evolution of the objective function cost over the course of the iterations. Each curve corresponds to a simulation performed with a different number of protons per beamlet. Overall, the cost tends to decrease with increasing iterations.

The curves associated with higher proton counts exhibit smoother and more stable convergence, whereas those using fewer protons tend to fluctuate more and show slower or incomplete convergence. The amplitude of the cost fluctuations also decreases with the number of protons.

The cost is evaluated at each iteration by comparing the current weight vector to a fixed reference dose matrix, which was computed at the beginning of the simulation. This reference matrix was generated by simulating each beamlet with one million protons to obtain a highly accurate dose estimate. While such a matrix would not be feasible in practice with the beamlet-free algorithm, due to computational constraints, it serves solely as a benchmark for performance evaluation and does not influence the optimization itself.

5.2.2 Dose Volume Histograms

Figure 5.3 presents the Dose Volume Histograms (DVHs) obtained for different numbers of protons per beamlet. As the number of protons increases, the DVH curves tend to approach the dose objectives, indicated by the vertical black lines, specifically 20 Gray for the target volume and 0 Gray for the

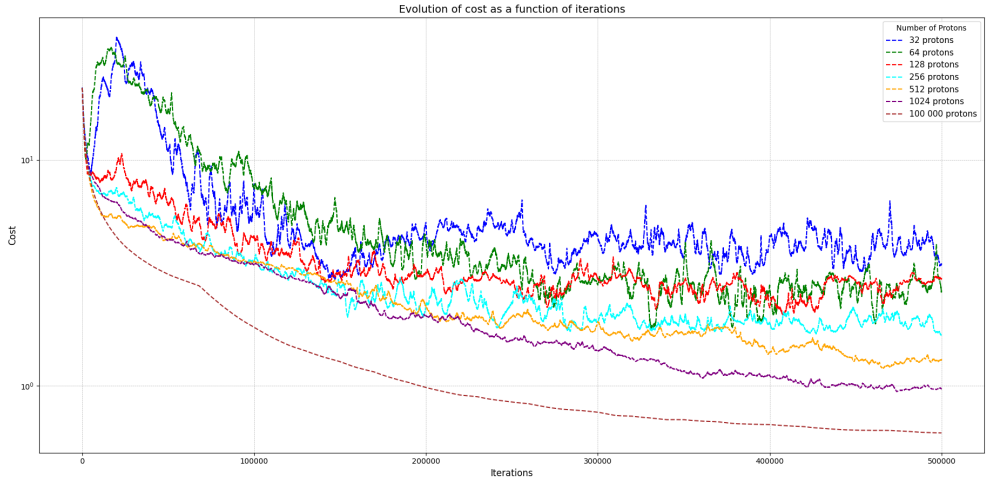


Figure 5.2: Evolution of costs as a function of iterations

organ at risk. Differences between curves are more pronounced in the organ at risk than in the target region.

Table 5.1 summarizes key DVH metrics for each region of interest and for varying proton counts. Reported values include the 2nd, 5th, 95th, and 98th percentiles of the dose distribution. For reference, the 5th percentile corresponds to the dose received by at least the 5% most irradiated volume. While these percentiles do not always improve monotonically with increasing proton count, an overall trend toward better dose conformity can be observed.

	Number of protons :						
Target	**32**	**64**	**128**	**256**	**512**	**1024**	**100000**
D2	21.64	21.86	21.67	21.67	21.30	21.37	21.37
D5	21.25	21.28	21.52	21.35	21.18	21.06	20.96
D95	17.15	18.10	17.91	18.25	18.71	18.69	18.81
D98	16.49	17.64	17.13	17.49	18.40	18.32	18.42

	Number of protons :						
Organe at risk	**32**	**64**	**128**	**256**	**512**	**1024**	**100000**
D2	6.97	6.87	6.75	5.33	5.36	4.92	3.50
D5	6.97	6.80	6.68	5.14	5.32	4.92	3.47

Table 5.1: Evolution of key dose metrics for the target and organ at risk as the number of simulated protons per beamlet increases.

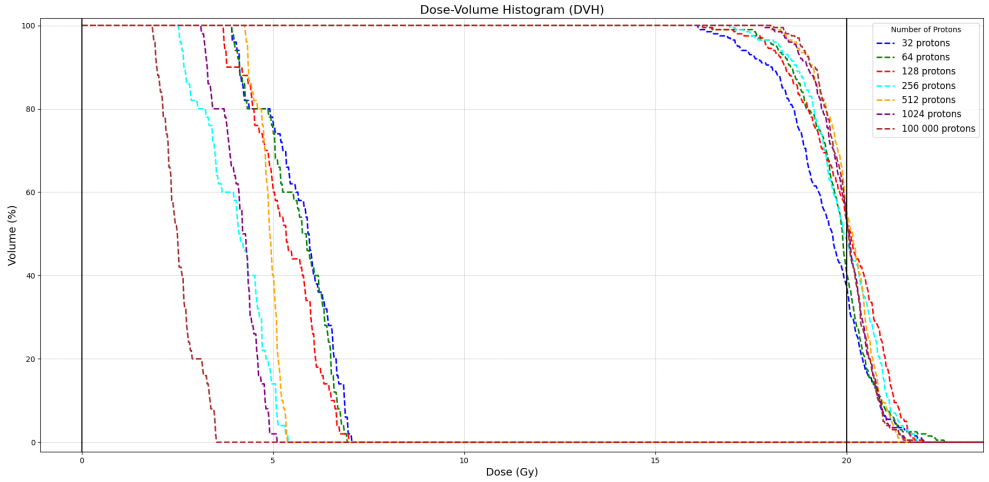


Figure 5.3: Dose Volume Histogram across the full range of simulated protons per beamlet.

5.3 Learning Rate

The previous results were obtained using a fixed learning rate. However, as discussed earlier in this thesis, it can be beneficial to adapt the learning rate over the course of the iterations. To investigate this, further simulations were performed using beamlets composed of 1024 protons.

Two types of learning rate decay were explored: linear and exponential. For both strategies, three decay profiles were tested: from 0.2 to 0.05, from 0.3 to 0.033, and from 0.4 to 0.025. Figure 5.4 shows how these learning rates evolve over iterations. Solid lines correspond to linear decay, while dashed lines represent exponential decay. A horizontal purple line indicates the fixed learning rate of 0.01 used in the previous sections. The markers on the curves indicate the iterations at which the learning rate was updated. In this work, the step size is updated each time a full batch of particles is processed. A batch is defined as the complete set of distinct beamlet pairs, which, in this case, amounts to 15,576 combinations.

Figure 5.5 shows the evolution of the cost as a function of the number of iterations for various learning rate schedules. The same color coding and line styles are used throughout. Overall, the curves display similar behavior, with two notable exceptions: the green dashed line (exponentially decreasing learning rate from 0.4 to 0.025), which results in a higher cost, and the solid

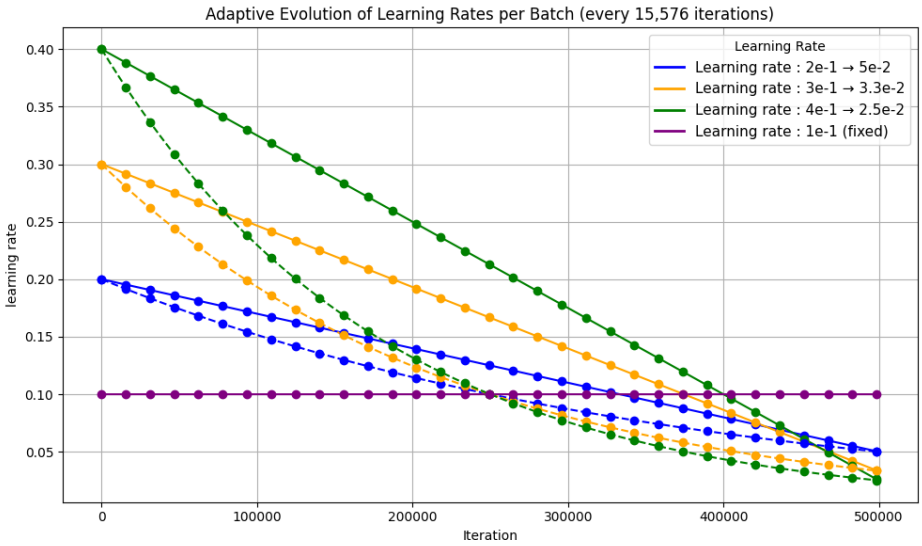


Figure 5.4: Adaptive Evolution of Learning Rates per Batch (every 15,576 iterations)

yellow line (linearly decreasing learning rate from 0.3 to 0.033), which yields a lower cost.

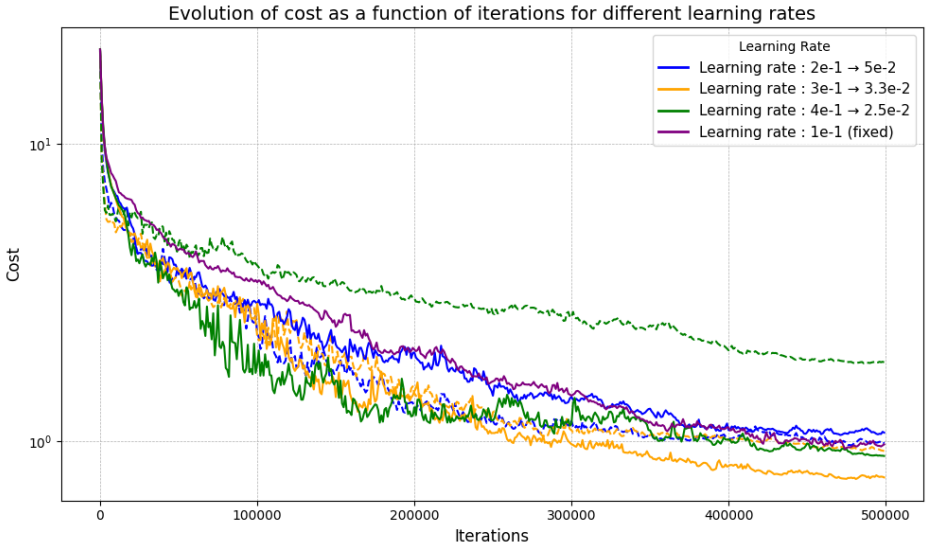


Figure 5.5: Evolution of cost as a function of iterations for different learning rates

5.4 Stopping Criterion

Just like the previous section, the analysis of the stopping criterion is performed using the optimization that simulates beamlets composed of 1024 protons.

To identify a stopping criterion, two approaches were analyzed: the evolution of the magnitude of the stochastic gradients and the norm of the gradient vector as a function of the iterations, as explained in Section 4.3.8.

Regarding the first approach, we initially applied a Gaussian filter to the evolution of the stochastic gradient magnitude. As shown in Figure 5.6, the gradient magnitude decreases sharply during the first 100,000 iterations before reaching a plateau.

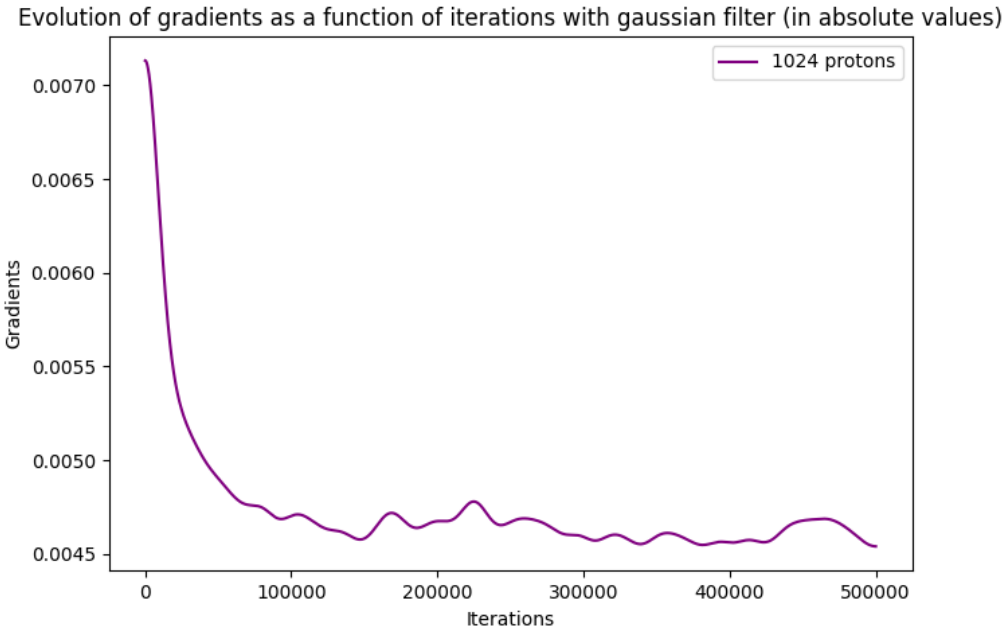


Figure 5.6: Variation of absolute gradient values across iterations with Gaussian filtering

The second approach considered was to compute the norm of the gradient vector. In this method, we sum all the gradients corresponding to the same beamlet that fall within the same iteration window. The behavior of the

gradient norm evolution varies significantly depending on the chosen window size. Figure 5.7 shows different windows. The curves representing windows of 50,000 and 150,000 iterations appear to plateau after a certain point, in contrast to the curve corresponding to a 250,000 iteration window.

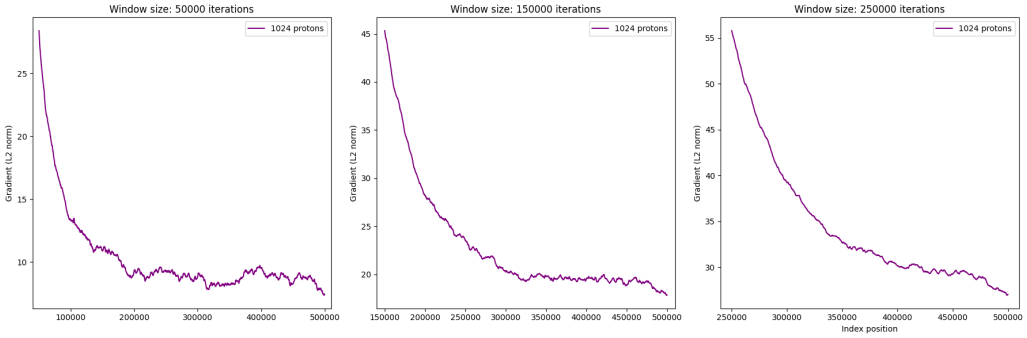


Figure 5.7: Gradient norm variation profile as a function of iterations for 3 different windows

Figures 5.8 and 5.9 allow us to compare this norm (using a 250,000 iteration window) with the evolution of the cost function over the iterations. We can observe that the gradient norm curve has an overall shape similar to that of the cost function, although clear differences remain.

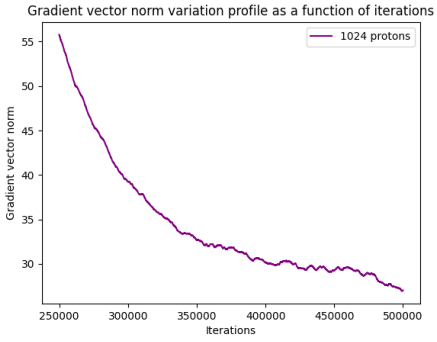


Figure 5.8: Gradient vector norm variation profile as a function of iterations

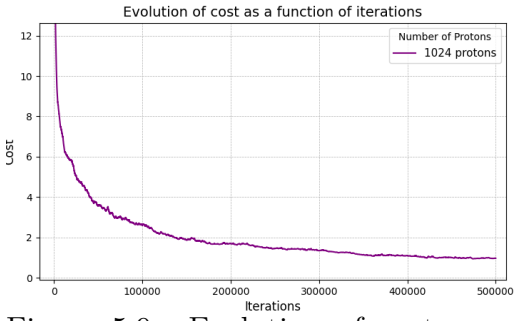


Figure 5.9: Evolution of cost as a function of iterations

Finally, the same process was performed for simulations using beamlets with different numbers of protons, as shown in Figure 5.10. The gradient vector appears to approach a plateau that decreases as the number of protons per beamlet increases. This decrease in gradients relative to the number of protons is also illustrated in Figure 5.11. We kept Figure 5.10 in linear

scale to remain consistent with the previous figures. As a result, the curves corresponding to a high number of protons may appear almost flat. However, they actually follow a similar shape to the other curves, but within a smaller value range.

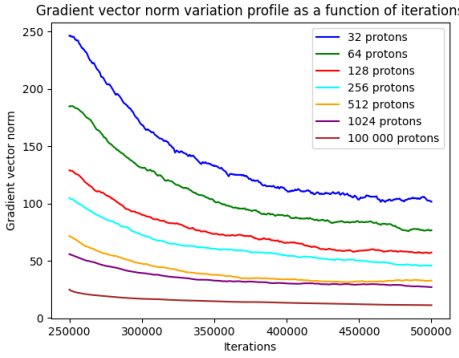


Figure 5.10: Gradient vector norm variation profile as a function of iterations

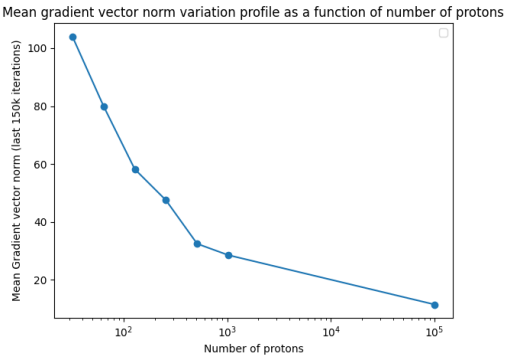


Figure 5.11: Mean gradient vector norm variation profile as a function of number of protons

5.5 Application to a Realistic Patient Case

At this stage, it is important to present the algorithm's performance on patient images. The optimization was performed using beamlets of 256 protons. This number of protons was chosen because, given that the goal of the beamlet-free approach is to use beamlets generated with a small number of protons, 256 already seems like a more reasonable choice than 1024. Figure 5.12 shows the dose distributions generated after optimization. Although there is a dose peak at the target location, the dose distribution within the target is slightly less uniform compared to the simple example analyzed earlier.

Next, Figure 5.13 shows the evolution of the cost function over iterations, together with the corresponding dose volume histogram. The goal within the target is to achieve a uniform dose of 8 Gray; the corresponding curve in the DVH indicates that the algorithm approaches this objective but does not fully satisfy it. Additionally, we observe significant doses delivered to a certain percentage of the right lung volume.

The dose statistics on tabular 5.2 confirm the DVH interpretation: the

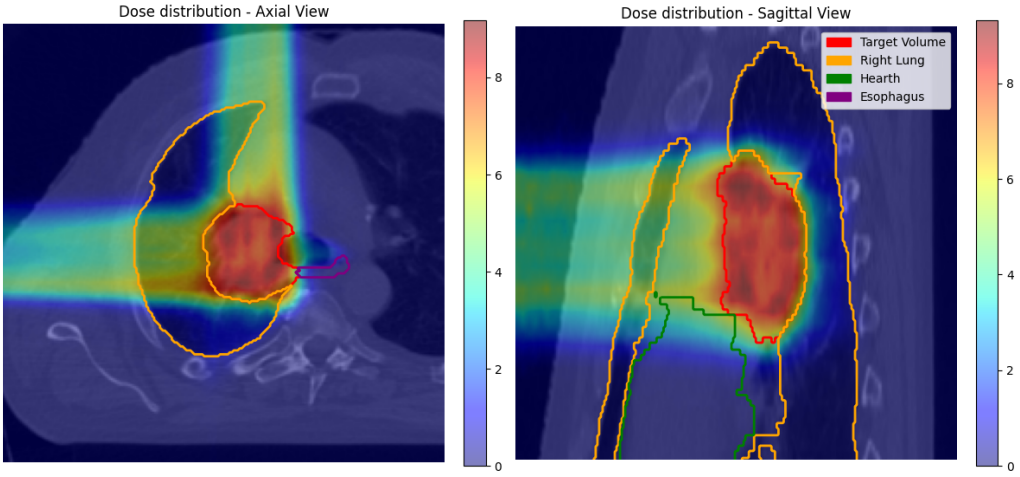


Figure 5.12: Dose distribution for the patient with Beamlet Free Algorithm (256 protons)

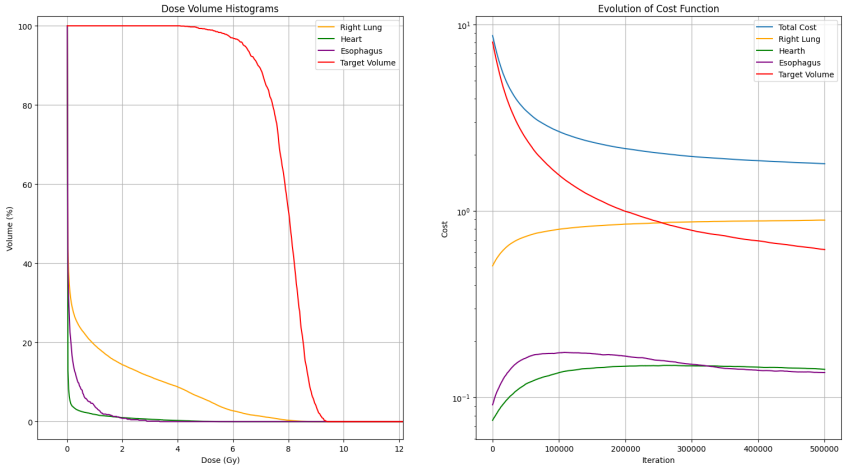


Figure 5.13: DVH and Cost as a function of iterations for the Patient Example with Beamlet Free Algorithm (with 256 protons)

relatively low D98 value (5.7 Gy) suggests underdosage in a significant portion of the volume. Furthermore, a slightly elevated D2 (9.14 Gy) suggests slight hotspots, indicating a lack of homogeneity within the target. Regarding the organs at risk, the esophagus and heart received relatively low radiation doses, as indicated by both their D2 and mean dose values. In contrast, the right lung appears to have been significantly affected, with the DVH revealing that a considerable portion of its volume was exposed to high doses relative to the prescribed target dose.

Region of Interest	**D2**	**D5**	**D95**	**D98**
Target Volume	9.14	8.95	6.41	5.70
Right Lung	6.46	5.21	–	–
Heart	0.89	0.13	–	–
Esophagus	1.31	0.82	–	–

Table 5.2: Dose metrics for target volume and organs at risk (in Gy)

5.6 Comparison with the Beamlet-Based Algorithm

Finally, to conclude this section, it is interesting to compare the beamlet free algorithm implemented in this work with a beamlet based algorithm. This comparison will be detailed in the following points for both the simple example and the patient images used in this thesis. All tests using the beamlet-based algorithm were performed with beamlets simulated using 100,000 protons.

5.6.1 Example on a Simple Configuration

Figure 5.14 shows the evolution of the cost function over iterations for the beamlet free algorithm (with 1024 and 100,000 protons simulated per beamlet) as well as for the beamlet based algorithm for our basic example. This figure also presents a dose-volume histogram comparing the two algorithms. It can be observed that the total cost is significantly lower for the beamlet based algorithm compared to the beamlet free algorithm (0.03 versus around 1, respectively), even with the same quality of beamlets simulation. Regarding the DVH, the beamlet based algorithm tends to better approach the target and OAR objectives compared to the beamlet free algorithm.

The iteration scale is not shown on the cost plot because one iteration of the beamlet-based algorithm does not correspond to one iteration of the beamlet-free algorithm. In the beamlet-based approach, the gradients of all beamlets are computed at each iteration, allowing the weights of all beamlets to be updated simultaneously. In contrast, the beamlet-free algorithm only updates the weights of two beamlets at a time. Moreover, a single iteration of the beamlet-based algorithm is significantly more computationally intensive than one of the beamlet-free algorithm. Therefore, the relevant information in this figure is the overall shape of the curve and the final cost reached, but not the apparent speed of convergence, since the x-axis does not represent a comparable scale between the two methods.

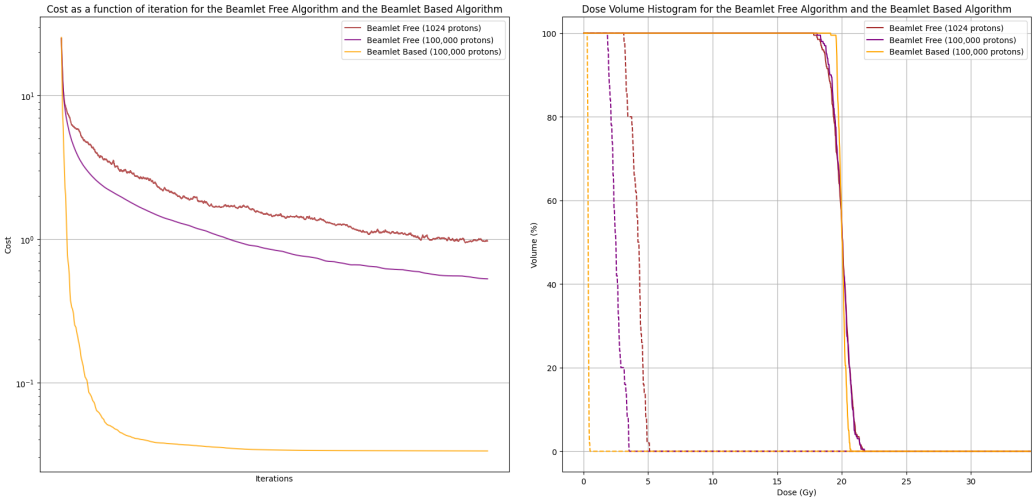


Figure 5.14: DVH and Cost comparison between the Beamlet Free Algorithm (with 100,000 and 1024 protons) and the Beamlet Based Algorithm

5.6.2 Example on a Realistic Configuration

Figures 5.15 and 5.16 present the DVHs and the cost evolution as a function of iterations on the patient's CT scan, comparing the beamlet-free algorithm (with 256 and 50,000 protons simulated per beamlet) and the beamlet-based algorithm. The results obtained with the beamlet-free algorithm are very similar regardless of the number of protons used. Once again, the beamlet-based approach performs better, but in this case, the difference between the two algorithms is much smaller. The final cost reaches approximately 1.3 with the beamlet-based method, compared to 1.7 with the beamlet-free one. As previously mentioned, the iteration scale is not shown in the cost plot,

since the notion of an "iteration" is not comparable between the two methods.

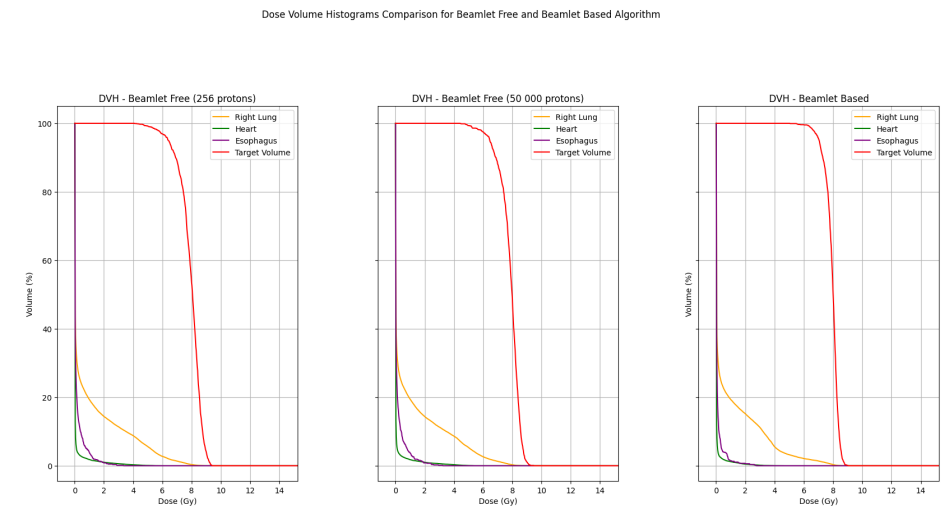


Figure 5.15: DVH comparison between the Beamlet Free Algorithm (with 50,000 and 256 protons) and the Beamlet Based Algorithm

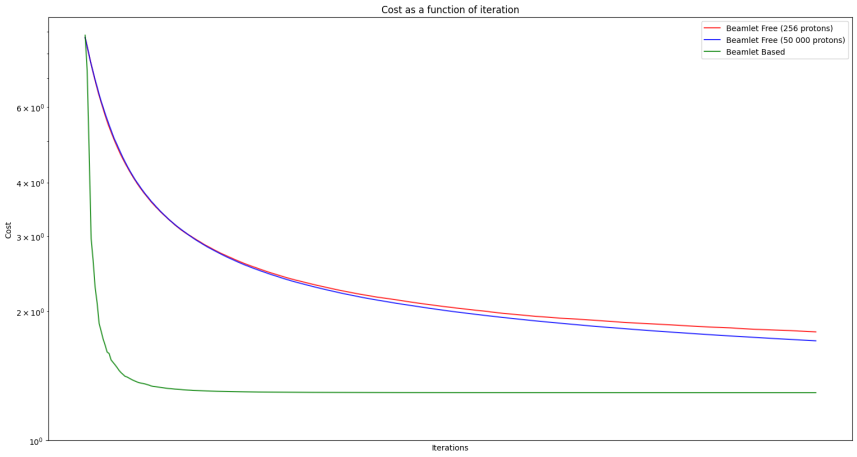


Figure 5.16: Cost in function of iterations comparison between the Beamlet Free Algorithm (with 50,000 and 256 protons) and the Beamlet Based Algorithm

Chapter 6

Discussion

6.1 Objectives of the Study

As mentioned earlier in Section 3.2.4, the aim of this study is to establish a proof of concept for the beamlet-free algorithm, focusing primarily on its feasibility and convergence. However, we are not able to evaluate performance in terms of computation time and numerical efficiency due to the hybrid architecture of the implementation: the algorithm is implemented in Python while the Monte Carlo simulations are run in C, which leads to a performance penalty caused by communication between the two languages. This study therefore serves as an intermediate step between the theoretical method and the implementation of an optimized version of the algorithm in C, which would eliminate the time loss caused by inter-language exchanges.

6.2 Interpretation of Results

6.2.1 Residual Noise in the Optimization Process

As an introduction, a first expected observation which will reappear throughout the discussion is the emergence of residual noise affecting the results. This originates from two main factors: the stochastic noise inherent to stochastic gradient descent, and the noise arising from beamlets generated with a limited number of protons.

Stochastic Noise

In addition to the noise induced by the beamlets, the use of a stochastic algorithm also generates noise in the results. Indeed, the gradients used are

estimates based on subsampling, which introduces additional variability in their computation. This noise is inherent to the method and therefore unavoidable, but it can be mitigated by adjusting certain parameters, such as the learning rate.

Theoretically, although the stochastic method introduces noise, the gradients should gradually balance out over the iterations, allowing the algorithm to reach convergence. This is a direct consequence of the mathematical formulation, which indicates that the theoretical gradient of a beamlet corresponds to the sum of all stochastic gradients computed when comparing this beamlet with each of the others. According to the law of large numbers, if a sufficient number of beamlet comparisons are made, the result should converge toward the exact gradient.

To isolate this stochastic noise, we used a simulation with beamlets generated using a large number of protons (100,000 protons), which allows us to minimize the noise caused by the beamlets. In Figure 5.2, the curve corresponding to beamlets simulated with 100 000 protons shows a clear trend toward convergence throughout the iterations, although its cost occasionally increases due to stochastic noise, as seen in the zoomed-in view in Figure 6.1. However, since the algorithm has not yet fully converged, it is difficult to assert with certainty that it would converge to the minimum of the cost function if the number of iterations were significantly higher. As shown in Figure 5.14, the beamlet-based algorithm converges to a much better solution, which demonstrates that it is possible to reach an better optimal solution.

It would be interesting to further investigate this limitation in convergence, as different hypotheses could lead to this result.

- The stochastic noise remains significant, even after a very large number of iterations, which prevents the algorithm from converging to the minimum of the cost function within a realistic number of iterations.
- The learning rate used in our implementation is not optimal. A decreasing learning rate specific to the optimization could potentially improve the results. However, we tested different decreasing learning rates, and our results do not appear to show any significant improvement.

These results suggest that the stochastic noise is too significant and prevents convergence within a realistic number of iterations.

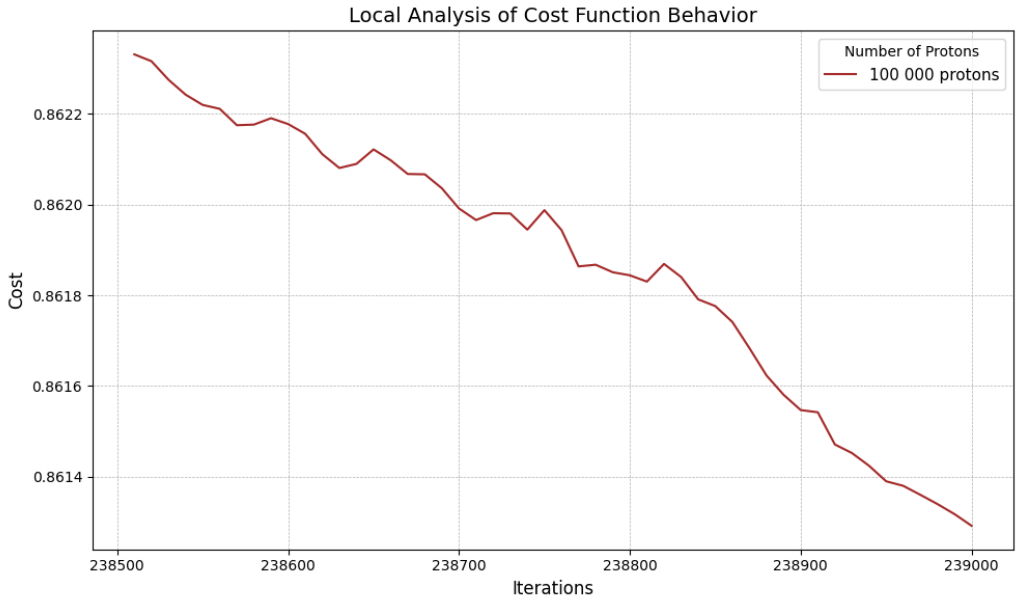


Figure 6.1: Local analysis of Cost Function Behavior

Beamlet Noise

The very principle of the beamlet-free algorithm relies on simulating a large number of low-precision beamlets. Using a small number of protons to simulate each beamlet inevitably introduces noise, which tends to decrease as the number of protons increases. The results obtained confirm this observation: the algorithm performs poorly when the number of protons is too low, while it improves significantly as this number increases. As shown in Figure 5.2, where the same learning rate is used for each simulation, those with few protons are extremely noisy. This suggests that an adaptive learning rate based on the number of protons could yield better results, as it would limit the impact of noise on the evolution of the weights.

However, it is quite difficult to determine the optimal learning rate "blindly" without running the code multiple times. Moreover, if we use an optimal learning rate for some proton numbers and suboptimal ones for others, the results are no longer truly comparable. For this reason, we chose to keep the learning rate fixed when analysing our results. The impact of using different learning rates is investigated separately, for a fixed number of protons.

6.2.2 Search for a Stopping Criterion

To define a proper stopping criterion, we should study the shape of the cost function: when it stagnates or decreases only very slightly, this suggests that a minimum has been almost reached. However, since we do not know the total cost (given by the matrix product $\mathbf{Bx}$), it is necessary to identify another quantity that exhibits a similar behavior.

Normally, the gradients are expected to follow roughly the same behavior as the cost function. As shown in Figure 5.6, it is not possible to directly use the stochastic gradients as a stopping criterion because they stagnate after fewer than 100,000 iterations, likely due to stochastic noise. In reality, it is not the individual gradient that faithfully reflects the cost curve, but rather the norm of the sum of gradients associated with a beamlet and all other pairs.

Thus, as detailed in section 4.3.8, we chose to compute these norms by summing, for each beamlet, the gradients over a window of iterations. In our specific case, to obtain a reliable measure, the size of this window must be large. However, this introduces a latency effect, since the norm is calculated over the entire window.

It is interesting to note that this norm eventually stagnates beyond a certain number of iterations. Moreover, as shown in Figures 5.10 and 5.11, the average value reached during this plateau decreases as the number of protons increases. This can be explained by the uncertainty associated with the beamlets, as well as the noise induced by the stochastic descent.

Indeed, highly noisy beamlets generate gradients that are themselves more noisy. Moreover, since these beamlets deviate further from the optimal solution, their gradients tend to remain higher, as gradients generally decrease when approaching the convergence point. It thus appears that the more imprecise the beamlets are, the less accurately the norm we compute reflects the cost curve.

Finally, Figures 5.8 and 5.9 show that, when the beamlets are sufficiently well simulated and the iteration window is large enough, the computed norm loosely follows the general trend of the cost function, although the two are not closely aligned. This appears to be a potential approach for defining a stopping criterion, although at present, the two curves still appear quite different. To obtain a satisfactory stopping criterion, it would be neces-

sary to ensure that it converges at the same time as the algorithm. Since, in our simulations, the algorithm does not seem to have fully converged, it is difficult to prove that the stopping criterion will converge at the same time.

6.2.3 Comparison with the Beamlet-Based Approach

Figure 5.14 and Figure 5.16 show the evolution of the cost across different algorithms, we will primarily focus on the final cost achieved rather than the speed in terms of iteration count. As mentioned in the Results section, comparing the number of iterations between the beamlet-free and beamlet-based algorithms is not meaningful: in a single iteration, the beamlet-based algorithm updates the weights of all beamlets, whereas the beamlet-free approach only modifies two beamlets.

It appears that in the case of a complex patient plan with a high number of beamlets, the beamlet-free algorithm performs proportionally closer to the beamlet-based algorithm than in a simple plan. For instance, in the patient case, the final cost reached was 1.7 with the beamlet-free algorithm compared to 1.3 with the beamlet-based algorithm, a difference of about 24%, while the beamlet-free algorithm had likely not yet fully converged. On the other hand, for the simple plan, the final cost was around 1 with the beamlet-free algorithm using 1024 protons, compared to a cost of just 0.03 with the beamlet-based, a difference of 97%.

These results suggest that the beamlet-free algorithm may be significantly more effective in complex plans than in simple ones, which is particularly relevant in realistic clinical scenarios.

Figure 6.2 comes from a study that used an early version of the beamlet-free algorithm, which is slightly different from ours. It shows that the beamlet-free algorithm is less effective than the beamlet-based approach when using fewer than 2000 spots, but appears to perform significantly better beyond that threshold.

In our patient case, a total of 1108 beamlets were used, compared to 176 for the simple case. It would therefore be particularly interesting to compare both algorithms on more complex cases. However, we were limited by the computational resources required for such simulations. Additionally, our implementation was intended as a proof of concept rather than an optimized version.

The performance difference between the beamlet-free and beamlet-based approaches depending on the number of spots can be explained by several factors. The beamlet-based algorithm stores the entire beamlet matrix in memory and uses it at each iteration to compute the gradients, resulting in a computational cost that is roughly linear with respect to the number of beamlets. On the other hand, the beamlet-free algorithm only keeps two beamlets in memory at a time, meaning the computational cost per iteration remains nearly constant regardless of the number of beamlets, although the number of iterations required increases with the complexity of the plan.

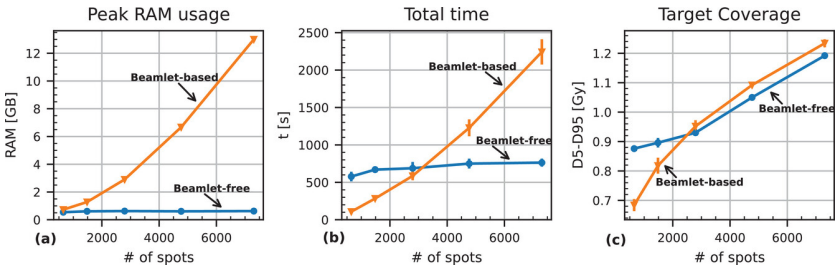


Figure 6.2: Comparison of beamlet-based and beamlet-free algorithms in terms of memory usage, computation time, and target coverage. Credits : [4]

It is important to note that the performance of the beamlet-free algorithm could likely be significantly improved with a better choice of learning rate. In contrast, the beamlet-based algorithm already uses an adaptive learning rate, which is presumably close to optimal, potentially limiting further improvements and narrowing the gap between the two methods.

Nevertheless, the current results do not appear to meet clinical constraints sufficiently for the treatment plan to be viable in practice. This may be due to simplifications made in the simulation parameters. Specifically, we were constrained to use a spot spacing of 7 mm and a layer spacing of 6 mm, with only two main beams, and employed voxels 64 times larger than those in the original CT scan in order to run the simulations within a reasonable time frame on our available computational resources.

Despite these limitations, the results demonstrate clear convergence of the beamlet-free algorithm, which appears to approach the performance of the beamlet-based method as the number of beamlets and the complexity of the plan increase. This is a very promising outcome and motivates further analysis of the computational cost and resource requirements needed to achieve full convergence.

6.3 Limitations

After discussing the results obtained, it is important to outline some significant limitations encountered throughout the thesis.

6.3.1 Impact of the Number of Beamlets on Algorithm Performance

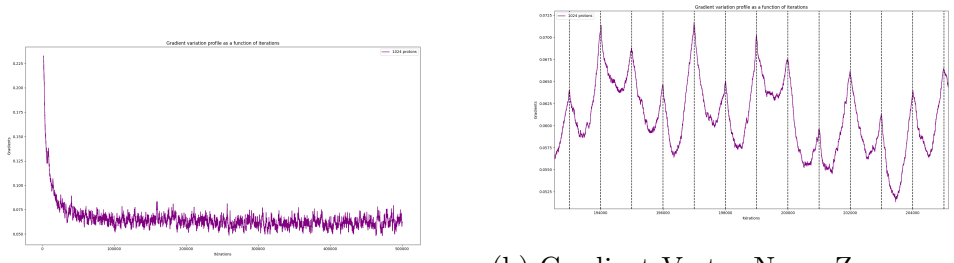
First, we observed that a large number of iterations is necessary for the algorithm to converge. Furthermore, the simulations conducted in this thesis use relatively few beamlets compared to what is required in real clinical cases. Since the square correlation matrices $\mathbf{C}_W$ have dimensions equal to the number of beamlets, increasing the number of beamlets leads to a quadratic growth in the size of these matrices. As a result, a proportionally higher number of iterations will be required on average to fill the correlation matrices with an equivalent level of precision.

6.3.2 Limitation Due to Beamlet Reuse

The optimization was implemented in Python, while the beamlet simulation was performed using the MC Square software, which is written in C. Communication between these two programs inevitably involves a significant time cost during algorithm simulation. Ideally, a random pair of beamlets would be selected and simulated by MC Square at each iteration. However, the latency caused by inter-language communication makes frequent calls to MC Square difficult to implement. To overcome this issue, we chose to simulate a full matrix of beamlets every 1000 iterations and then randomly select beamlet pairs from this matrix. Consequently, after 500,000 iterations, 500 matrices will have been generated.

Figures 6.3a and 6.3b show the evolution of the gradient norm over iterations using a small window. The gradient norm was calculated in the same way as in Section 4.3.8. Figure 6.3b is a zoomed-in view of Figure 6.3a and reveals an unexpected pattern: the curve exhibits numerous peaks, each spaced exactly 1000 iterations apart. This corresponds to the intervals at which new beamlet matrices are generated. For 1000 iterations, the algorithm minimizes the cost based on a fixed set of beamlets and, when the cost decreases, the gradients also decrease. Then, when new beamlets are simulated, the cost increases slightly (as the weights are adapted to the previous beamlets), and the gradients rise accordingly. These increases appear as peaks in Figure 6.3b. Theoretically, we would expect a slow decrease followed

by a sudden rise in the gradient norm. However, these peaks are smoothed due to the window used to compute the norm, which acts as a filter. By calculating a moving norm over several iterations, this window attenuates abrupt variations in the gradients.



(a) Gradient Vector Norm

(b) Gradient Vector Norm Zoom on image

Figure 6.3: Gradient vector norm profile for a tiny window

6.3.3 Pre-filling of the Correlation Matrix $\mathbf{C}_W$

As a reminder, one of the main advantages of the Beamlet-Free method is that it does not require storing all beamlets in memory. However, we chose to use the beamlet matrix to pre-fill the $\mathbf{C}_W$ matrix, which goes against this principle and constitutes a significant limitation of our approach.

A preliminary solution was developed in Section 4.3.5. The $\mathbf{C}_W$ matrix exhibits regular patterns, suggesting that it may theoretically be possible to develop an efficient approximation method without computing the full matrix. Even if such an approximation is not perfectly accurate, it would likely be sufficient to produce results comparable to ours, considering that our own $\mathbf{C}_W$ matrix was built from noisy, and thus imprecise, beamlets.

Designing such an approximation method could represent a promising direction for future work.

6.3.4 Addition of an Organ at Risk

Introducing an OAR into the model using a uniform dose objective set to 0 Gy is not optimal. Two alternative strategies were investigated in order to better reflect the goal of minimizing the maximum dose, rather than a uniform dose, in the OAR, while avoiding reliance on the mathematical gradient, whose computation appears prohibitively complex. However, each of these methods presents significant limitations that prevent us from achieving

satisfactory results.

To implement a D_{max} objective instead of a $D_{uniform}$, one could in theory apply the same approach as for $D_{uniform}$, but restrict it only to voxels receiving a dose above the prescribed threshold. However, the dose distribution is unknown during the optimization process, since computing it requires access to the full beamlet matrix through the equation: $\sum_{i=1}^{B} \mathbf{b}_{ip} x_i$.

First Method

A first approach to estimating the dose distribution is to construct an accumulated dose matrix. This matrix is initially set to 0 Gy in every voxel. Then, at each iteration, we update it by adding the contribution of the simulated pair of beamlets, scaled by their respective gradients.

If a gradient is negative, the algorithm aims to increase the beamlet's weight, meaning that the beamlet will contribute more dose, so we add its dose distribution to the accumulated matrix. Conversely, if the gradient is positive, we subtract its dose, reflecting the fact that the algorithm is trying to reduce the weight of that beamlet. With this accumulated dose matrix, we can identify voxels where the dose exceeds the prescribed threshold, and restrict the gradient computation to those voxels only. This mimics a D_{max} objective.

A major limitation of this method is the accumulation of error. Because only a small number of protons is simulated per beamlet, the dose contribution added to the matrix at each iteration is noisy and approximate. For example, a given beamlet might be added to the accumulated dose at one iteration, then removed at another. However, since each simulation of the beamlet is imprecise and varies from one iteration to the next, the subtraction doesn't perfectly cancel out the previous addition. This introduces an error into the accumulated dose, and over time, these discrepancies build up, leading to an increasingly inaccurate estimation of the actual dose distribution. Over time, this error accumulates, and the accumulated dose no longer reflects the actual dose corresponding to the current weights x. The smaller the number of protons simulated per beamlet, the larger this deviation becomes.

Second Method

A second approach would be to consider gradients in the organ at risk only when they are strictly positive. This would imply that the second term is greater than the first, meaning that $\mathbf{Bx} > \mathbf{d}_{\text{pr}}$, i.e., the received dose exceeds the prescribed dose.

This method has the advantage of not requiring the tracking of an accumulated dose. However, it does not allow us to identify which specific voxels exceed the dose objective, so the gradient is computed over the entire OAR regardless of the local dose distribution.

6.4 Perspectives

Finally, several perspectives that we find interesting to explore for potential future work are outlined. These perspectives are closely related to the limitations discussed in the previous section.

6.4.1 Code Optimization

Now that the proof of concept for the beamlet-free algorithm has been completed, it could be interesting to use the C programming language instead of Python. Beyond the fact that C is generally more efficient than Python for optimization tasks, it is important to recall that beamlet simulation is performed in C, while the optimization was implemented in Python. Communication between these two programs inevitably introduces latency during the algorithm's execution.

Moreover, integrating the code directly into MC Square (the software used for beamlet simulation) would help overcome the limitation discussed in the previous section regarding the reuse of beamlets.

Because many of the matrices in our approach are sparse (since we only look at voxels inside the ROI), we used sparse matrices as much as possible. It would be useful to keep using them, as sparse matrices help improve the speed and efficiency of the beamlet-free algorithm.

It is certainly possible to improve our code by finding strategies to reduce the amount of calculations required and/or increase simulation speed. As an illustration, three non-exhaustive examples are listed below :

- Implementing the algorithm to support multithreading would allow for the simultaneous comparison of multiple beamlets, thereby increasing the simulation speed.
- Use each beamlet for two consecutive iterations. The beamlet simulated at iteration t could be reused at iteration t+1, along with a newly simulated beamlet that would then be reused at iteration t+2, and so on. This approach would reduce the number of Monte Carlo simulations by approximately half, as each simulated beamlet would be used twice.
- Finally, a last example to improve the algorithm's efficiency could be to progressively refine the grid size (spatial resolution) over the course of the iterations. The simulation could begin with low-resolution images and less precise beamlets. After a certain number of iterations, the resolution of both the images and the beamlets could be increased. This approach would likely save computational resources and time during the initial phase, before enhancing the precision of the algorithm later on.

However, it would first be necessary to ensure that these strategies do not alter the theoretical convergence of the algorithm.

6.4.2 Improved Management of Organs at Risk

As presented in Section 6.3.4, it would be relevant to derive a theoretical expression for the gradient with regard to the specific cost of a given organ at risk, as represented below.

$$\mathbf{D_W} = \|max(\mathbf{d}^{pr} - \mathbf{Bx}, \mathbf{0})\|^2_{\mathbf{W}}$$

Another possibility is to explore alternative numerical approaches to obtain gradients that more closely approximate the ideal gradients. As an introduction, two methods that we implemented were briefly mentioned in Section 6.3.4, although they were not detailed in this thesis. These methods were only superficially explored, and it could be interesting to examine them more thoroughly.

6.4.3 Improved Pre-calculation of the Correlation Matrices $\mathbf{C}_W$

As explained in Section 6.3.3, achieving a good approximation of the correlation matrices $\mathbf{C}_W$ appears important to fully eliminate the beamlet matrix, as intended by the beamlet-free algorithm, while ensuring the algorithm

converges within a reasonable number of iterations. A potential solution is detailed in Section 4.3.5.

6.4.4 Learning Rate Adaptation

Finally, further research on a decreasing learning rate could lead to better results, and this learning rate could be adapted to the different parameters of the algorithm. Indeed the range of values should be correlated with certain parameters such as grid size, number of spots, and so on. Drawing inspiration from existing literature could help in adapting the learning rate effectively.

Chapter 7

Conclusion

In conclusion, the objective of this thesis was to realize a proof of concept for the beamlet-free algorithm. For this purpose, the optimization was performed according to the equations presented in Section 4.2. Through this work, the following points were chosen for analysis :

- The overall performance of the algorithm as a function of the number of protons simulated per beamlet.
- The search for a stopping criterion to stop the algorithm once it is judged to be sufficiently convergent.
- The exploration of an adaptive, decreasing learning rate that could lead to improved convergence of the algorithm.

These analyses were first performed on a simple example that is not representative of a clinical case, allowing us to evaluate the algorithm's performance without being influenced by the complexity of a real scenario. Then, after gaining a clear understanding of the overall functioning of the algorithm, a simulation was conducted on a patient image to assess the algorithm's performance in a real-world context.

This thesis also highlighted several major limitations related to the optimization process, most of which arise from a fundamental issue: residual noise. This noise arises from two different sources: first, the stochastic gradient descent approximating the true gradient at each iteration, and second, the uncertainty resulting from beamlets simulated with a relatively low number of protons.

Residual noise will therefore decrease as the number of simulated protons per beamlets increases, since the uncertainty generated by the beamlets will

be lower. Our results showed clear convergence of the beamlet-free algorithm, with its convergence point tending to approach that of the beamlet-based algorithm as the number of beamlets increases, which we find highly promising.

Contrary to our expectations, this thesis demonstrated that a large number of iterations is necessary for the algorithm to produce accurate results while continuing to approach the global minimum of the cost function.

As a proof of concept, the objective was not to directly optimize the algorithm's performance. That is why we chose a model developed in Python. The use of the C language would, of course, be more appropriate for optimizing the algorithm's performance in future work.

Without having conducted thorough analyses, we can say that the advantage of the beamlet-free algorithm lies in not needing to fully store the beamlet matrix, unlike other optimization methods used in proton therapy.

Several possible improvements have been suggested to enhance the performance of the beamlet-free algorithm and to assess whether it could be used in clinical settings and ideally, incorporated into adaptive treatment planning in proton therapy. The ultimate goal of this algorithm is to get as close as possible to the beamlet-based performance while achieving much faster optimization with significantly reduced computational resource requirements.

Bibliography

[1] *The Cancer Atlas*. en. URL: http://canceratlas.cancer.org/9WS (visited on 05/06/2025).

[2] Christine Allen, Sohyoung Her, and David A. Jaffray. "Radiotherapy for Cancer: Present and Future". en. In: *Advanced Drug Delivery Reviews* 109 (Jan. 2017), pp. 1–2. ISSN: 0169409X. DOI: 10.1016/j.addr.2017.01.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169409X17300133 (visited on 04/24/2025).

[3] Radhe Mohan and David Grosshans. "Proton therapy – Present and future". en. In: *Advanced Drug Delivery Reviews* 109 (Jan. 2017), pp. 26–44. ISSN: 0169409X. DOI: 10.1016/j.addr.2016.11.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169409X16303192 (visited on 04/24/2025).

[4] Danah Pross et al. "Technical note: Beamlet-free optimization for Monte-Carlo-based treatment planning in proton therapy". en. In: *Medical Physics* 51.1 (Jan. 2024). Publisher: Wiley, pp. 485–493. ISSN: 0094-2405, 2473-4209. DOI: 10.1002/mp.16813. URL: https://aapm.onlinelibrary.wiley.com/doi/10.1002/mp.16813 (visited on 05/01/2025).

[5] J. Lee, G. Janssens, and E. Sterpin. *Engineering challenges in proton-therapy, LGBIO2070 course*. 2024.

[6] Eric J. Hall and Amato J. Giaccia. *Radiobiology for the radiologist*. en. Eighth edition. Philadelphia: Wolters Kluwer, 2019. ISBN: 978-1-4963-3541-8.

[7] Thomas B. Brunner et al. "SBRT in pancreatic cancer: What is the therapeutic window?" In: *Radiotherapy and Oncology* 114.1 (Jan. 2015), pp. 109–116. ISSN: 0167-8140. DOI: 10.1016/j.radonc.2014.10.015. URL: https://www.sciencedirect.com/science/article/pii/S0167814014004654 (visited on 05/10/2025).

[8] Wayne D Newhauser and Rui Zhang. "The physics of proton therapy". en. In: *Physics in Medicine and Biology* 60.8 (Apr. 2015). Publisher: IOP Publishing, R155–R209. ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/60/8/r155. URL: https://iopscience.iop.org/article/10.1088/0031-9155/60/8/R155 (visited on 05/01/2025).

[9] E. Griesmayer and B. Dehning. "Diamonds for Beam Instrumentation". en. In: *Physics Procedia* 37 (2012), pp. 1997–2004. ISSN: 18753892. DOI: 10.1016/j.phpro.2012.02.526. URL: https://linkinghub.elsevier.com/retrieve/pii/S1875389212019232 (visited on 05/12/2025).

[10] *Ionizing Radiation Interactions | Oncology Medical Physics*. URL: https://oncologymedicalphysics.com/radiation-interactions/ (visited on 05/01/2025).

[11] Simon Deycmar et al. "The relative biological effectiveness of proton irradiation in dependence of DNA damage repair". en. In: *The British Journal of Radiology* 93.1107 (Mar. 2020). Publisher: Oxford University Press (OUP). ISSN: 0007-1285, 1748-880X. DOI: 10.1259/bjr.20190494. URL: https://academic.oup.com/bjr/article/doi/10.1259/bjr.20190494/7449338 (visited on 05/01/2025).

[12] *About proton therapy*. URL: https://www.ncc.re.kr/main.ncc?uri=english/sub11_Aboutprotontherapy.

[13] Radhe Mohan. "A review of proton therapy – Current status and future directions". en. In: *Precision Radiation Oncology* 6.2 (June 2022), pp. 164–176. ISSN: 2398-7324, 2398-7324. DOI: 10.1002/pro6.1149. URL: https://onlinelibrary.wiley.com/doi/10.1002/pro6.1149 (visited on 04/24/2025).

[14] Ben S. Ashby et al. "Efficient proton transport modelling for proton beam therapy and biological quantification". en. In: *Journal of Mathematical Biology* 90.5 (May 2025), p. 47. ISSN: 0303-6812, 1432-1416. DOI: 10.1007/s00285-025-02212-1. URL: https://link.springer.com/10.1007/s00285-025-02212-1 (visited on 05/06/2025).

[15] Jaeman Son et al. "Development of Optical Fiber Based Measurement System for the Verification of Entrance Dose Map in Pencil Beam Scanning Proton Beam". en. In: *Sensors* 18.1 (Jan. 2018), p. 227. ISSN: 1424-8220. DOI: 10.3390/s18010227. URL: https://www.mdpi.com/1424-8220/18/1/227 (visited on 04/24/2025).

[16] Felix Liu. "High Performance Computing for the Optimization of Radiation Therapy Treatment Plans". en. In: ().

[17] Patrick Wohlfahrt and Christian Richter. “Status and innovations in pre-treatment CT imaging for proton therapy”. en. In: *The British Journal of Radiology* 93.1107 (Mar. 2020), p. 20190590. ISSN: 0007-1285, 1748-880X. DOI: 10.1259/bjr.20190590. URL: https://academic.oup.com/bjr/article/doi/10.1259/bjr.20190590/7449297 (visited on 05/27/2025).

[18] Christopher R. King et al. “Stereotactic Body Radiotherapy for Localized Prostate Cancer: Interim Results of a Prospective Phase II Clinical Trial”. en. In: *International Journal of Radiation Oncology*Biology*Physics* 73.4 (Mar. 2009). Publisher: Elsevier BV, pp. 1043–1048. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2008.05.059. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360301608024590 (visited on 05/01/2025).

[19] John Lee and E. Sterpin. *Computational and Numerical Methods for Medical Physics*. 2025.

[20] Harald Paganetti. “Range uncertainties in proton therapy and the role of Monte Carlo simulations”. en. In: *Physics in Medicine and Biology* 57.11 (June 2012), R99–R117. ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/57/11/R99. URL: https://iopscience.iop.org/article/10.1088/0031-9155/57/11/R99 (visited on 04/24/2025).

[21] David Shepard. *IMRT Optimization Algorithms*. Slides.

[22] E. Sterpin. *WRDTH3160 : Technology, Dosimetry and Treatment Planning in Radiotherapy.* 2025.

[23] Nikhil Ketkar. “Stochastic Gradient Descent”. en. In: *Deep Learning with Python*. Berkeley, CA: Apress, 2017, pp. 113–132. ISBN: 978-1-4842-2765-7 978-1-4842-2766-4. DOI: 10.1007/978-1-4842-2766-4_8. URL: http://link.springer.com/10.1007/978-1-4842-2766-4_8 (visited on 04/24/2025).

[24] Kristy A Carpenter et al. “Deep Learning and Virtual Drug Screening”. en. In: *Future Medicinal Chemistry* 10.21 (Nov. 2018), pp. 2557–2567. ISSN: 1756-8919, 1756-8927. DOI: 10.4155/fmc-2018-0314. URL: https://www.tandfonline.com/doi/full/10.4155/fmc-2018-0314 (visited on 05/14/2025).

[25] Ian Goodfellow, Yoshua Bengio, and Aron Courville. *Deep learning*. MIT Press. 2016. URL: http://www.deeplearningbook.org.

[26] Kevin Souris, John Aldo Lee, and Edmond Sterpin. “Fast multipurpose Monte Carlo simulation for proton therapy using multi- and many-core CPU architectures”. en. In: *Medical Physics* 43.4 (Apr. 2016), pp. 1700–1712. ISSN: 0094-2405, 2473-4209. DOI: 10.1118/1.4943377. URL: https://aapm.onlinelibrary.wiley.com/doi/10.1118/1.4943377 (visited on 04/24/2025).

[27] S. Wuyckens et al. *OpenTPS – Open-source treatment planning system for research in proton therapy*. en. arXiv:2303.00365 [physics]. Mar. 2023. DOI: 10.48550/arXiv.2303.00365. URL: http://arxiv.org/abs/2303.00365 (visited on 04/24/2025).

[28] Noëlle C. Van Der Voort Van Zyp et al. “Clinical introduction of Monte Carlo treatment planning: A different prescription dose for non-small cell lung cancer according to tumor location and size”. en. In: *Radiotherapy and Oncology* 96.1 (July 2010), pp. 55–60. ISSN: 01678140. DOI: 10.1016/j.radonc.2010.04.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167814010002355 (visited on 05/19/2025).

[29] John A Lee. *Beamlet-free Monte Carlo plan optimization revisited*. en. 2024.

[30] *Coupon collector's problem*. URL: https://en.wikipedia.org/wiki/Coupon_collector%27s_problem.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**