Optimization of Radiotherapy Treatment Plans Based on Monte Carlo Dose Computations

Ludvig Håkansson

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6209
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2023

Abstract

Treatment planning plays a vital role in providing good treatment to cancer patients. In order to reach an adequate treatment plan, the algorithm used for simulating the dose in the patient must model the reality well. The most accurate algorithm for this is the Monte Carlo method, which in the context of radiotherapy treatment most often is used for pre-computing spot doses and optimizing the intensity for each spot. Due to storage limitations, it would be preferred to do the mentioned computations simultaneously as the optimization. This, however, makes the optimization problem non-convex due to the statistical noise introduced by Monte Carlo.

This thesis investigates the feasibility of using first-order optimization methods for treatment planning based on Monte Carlo simulations and addresses the challenges posed by the noise. A simplified proton Monte Carlo dose engine was implemented together with a matching analytical such, in order to assess the effect of the noise during optimization.

The results demonstrate that despite the noise, an adequate treatment plan can be achieved. Convergence is found to be dependent on how simulations are used within an iteration. Techniques such as accumulating total dose and computing the gradient and Hessian separately show promise for improving convergence and time efficiency, respectively. The impact of noise on error computation and the need for appropriate comparisons are highlighted.

This work provides insights for advancing Monte Carlo treatment planning and its integration into clinical settings. The findings are applicable not only to proton treatment plans but also to other ions and perhaps even to photons.

Preface

This thesis was carried out during spring of 2023 at Raysearch Laboratories for a master's degree project within the Engineering Physics program at the Faculty of Engineering (LTH), Lund University.

I would like to thank my main supervisor at RaySearch, Eric Landström, for the idea behind this exciting project and for continuous guidance throughout, from implementation details to the bigger picture of why what we do is important in the first place. I would also like to thank Erik Engwall and the rest of RaySearch for taking me in and providing interesting discussions.

Lastly, I would like to thank my supervisor at LTH, Pontus Giselsson, for giving feedback on the report.

Stockholm, June 2023
Ludvig Håkansson

Contents

1

Introduction

1.1 Background

Cancer is among the most deadly diseases in the world. Few can say that they do not have a friend or relative that have suffered from it. However, cancer treatment is an active field of research and is constantly advanced.

The most common treatment against cancer is radiotherapy treatment, i.e., irradiating with photons or particles which kills cells through ionization. The ionization damages the cell's DNA, and if the cell does not manage to repair the damage before it performs cell division (mitosis), it dies. Due to cancer cells performing cell division more rapidly than healthy cells, ionization is more deadly for cancer cells. For a successful radiotherapy treatment, the dose deposited in the tumor area must be weighed against the dose deposited in healthy tissue.

Traditionally, analytical methods are used to find how the dose is deposited in the body. While analytical methods in general are fast, they do not always produce a sufficiently accurate dose distribution, especially not in parts where there is much heterogeneity. Monte Carlo (MC) dose calculation is the golden standard when it comes to accuracy, due to the great understanding of particle interactions and cross sections in the body. In general, the MC method is based on statistical simulations from random sampling using pseudo-random numbers, in order to make an approximation of the underlying system. In particular, MC within radiotherapy treatment consists of sampling a particle's path and tracking where it deposit its energy through sampling the probability for different types of interactions. The drawback of MC is the need for many particle simulations to make a good approximation, which takes time. However, recent advancements in GPU computing and the use of *fast MC algorithms* have significantly reduced the computation time.

For protons, MC is widely used in treatment planning for *pencil beam scanning* (PBS), which is based on dividing a beam into small areas called spots, computing the dose each spot is contributing with, and then optimizing the intensity for each spot. When the parameter space becomes larger, this requires too much memory to be feasible. Instead, it would be beneficial to incorporate MC in the optimization, performing the spot dose computations during the optimization and thus eliminat-

ing the need for storing the large dose matrix that defines how the dose depends on each spot. However, this method introduces a new problem, namely that the objective function becomes non-deterministic and thus non-convex, due to the statistical noise. The calculated gradients will also be contaminated with noise, making many deterministic optimization methods fail.

This thesis aims at investigating the feasibility of using first order methods to optimize radiotherapy treatment plans based on MC dose computations and their ability to handle the noise introduced.

1.2 Objective

The objective of this master thesis is to further advance MC treatment planning in order to deliver a more accurate, and at the same time efficient, treatment for cancer patients. This will be done by investigating the effects of optimizing radiotherapy treatment plans based on MC computations. As MC computations contain noise, this must be accounted for.

The objective will be reached by answering the following research questions:

- Is it possible to optimize a treatment plan despite the noise introduced by MC?
- How does the convergence depend on the noise from different parts of an optimization step?
- Can a better convergence be achieved by smoothing the MC noise?
- How should current MC dose engines be incorporated in treatment plan optimization?

1.3 Constraints

As will be explained in the following chapter, simulating particles interacting with the body can be done very accurately. This thesis will not focus on simulating physical interactions as exactly as possible, but the model used will instead be something that resembles the optimization problem well so that the results of the experiments can serve as a guide when addressing the real optimization problem.

1.4 Outline of the report

In this report, it is the optimization part of radiotherapy treatment planning based on MC computations that is the important. However, in order to make the reader familiar with the topic and motivations for the model, some theory on radiotherapy treatment is presented. The chapters are as follows:

- Chapter 2 introduces some theory about radiation and how it is used in radiotherapy treatment. Here is also some background that motivates the model used. Finally, it briefly introduces inverse treatment planning which is used to optimize treatment plans using mathematical optimization.
- Chapter 3 covers what tools which will be used for the optimization and how they are formulated for the specific problem.
- Chapter 4 discusses different dose calculation algorithms, both analytical and based on MC. It further discusses more in depth how MC is used in radiotherapy treatment planning today.
- Regarding the rest of the chapters; Chapter 5 explains how the dose engines that was used for the experiment was implemented and the problem setup. Chapter 6, 7 and 8 consist of the results, discussion and conclusions, respectively.

2

Radiotherapy treatment

The following chapter introduces radiation and how it is used to cure cancer patients, especially radiotherapy treatment with protons. At last comes the motivation for how proton treatment plans are modeled and an introduction to optimization of treatment plans.

2.1 Radiation

Radiation can be divided into two groups, electromagnetic and particle radiation. The former constitutes of photons and is named differently depending on its wavelength, e.g., gamma rays and X-rays. The latter constitutes of matter such as electrons, protons, neutrons and ions traveling at high speeds. [Burton et al., 2023]

Absorbed radiation in body is measured as dose, which is how much energy from ionizing radiation is absorbed in the body per mass. Dose is most commonly measured in the SI unit Gray (Gy), where 1 Gy = 1 J/kg. The effect of absorbed dose is chemical reactions which causes damage to the DNA, consequently leading to cell death if the cell does not manage to repair it. The basics of radiotherapy treatment is the healthy cells' ability to repair themselves in higher extent than the cancerous cells, referred to as the *therapeutic window*. [Baskar et al., 2014]

It is often important to have high *tumor control*, i.e., high probability for all cancer cells to die, in order for the cancer treatment to be successful, since if only a fraction survives these will be able to continue their cell growth. However, when increasing the tumor control, the probability for complication increases as well, causing side effects such as cancer. Different tumors and organs have varying sensitivity to dose and in order to deliver the best possible treatment to the patient the dose must adhere to this difference. [Degerfält et al., 2008] To avoid side effects, advancements have been made to widen the therapeutic window [Baskar et al., 2014]. Another way to avoid complications is to irradiate the patience from different directions, since then the radiation does not travel through the same path to reach the tumor.

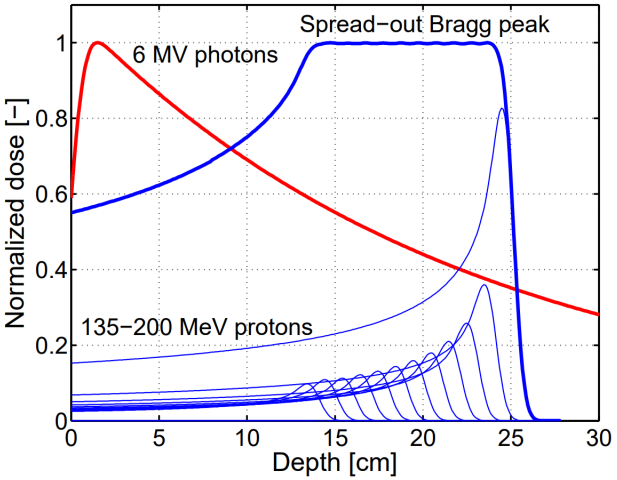


Figure 2.1 Depth dose curves along the central axis for 6 MV photon beam and 135-200 MeV proton beams. A uniform plateau at the peak is enabled when super positioning several Bragg peaks with different energy. This allows for delivering a uniform dose in the tumor while in high extent sparing other parts. Figure reproduced with permission from Bokrantz (2013).

Proton therapy

Protons for radiotherapy treatment are typically generated from hydrogen, where its electron is striped off. With the help of cyclotrons and synchrotrons the protons are accelerated in a circle or a spiral up to a desired energy whereas they are directed against the target. [Liu and Chang, 2011]

Because the particle is positively charged, when it moves through the body it constantly attracts and drags with atomic electrons, losing energy to the electrons and thus losing velocity. Just before it halts completely, after millions of interactions, it undergoes a lot more of these interactions per unit distance, which is what generates the so-called *Bragg peak* of the proton depth curve. The electron that is stripped from its atom is called a *secondary particle* and is an ionizing particle itself that deposits energy on its path. [Maughan, 2022]

Due to the Bragg peak, radiotherapy treatment with protons can in high extent deliver a desirable dose to the tumor while sparing other parts. This is done through dividing the same beam in many segments, where the protons in each segment have different energies, effectively having the peak where most of the dose is delivered at different depths in the body. Figure 2.1 shows how super positioning proton segments with different energies can create a uniform spread-out Bragg peak.

Pencil beam scanning. Due to the proton being a charged particle, after it has been accelerated and it is on the way towards the target, it passes through two sections of magnets that are perpendicular. Changing the magnetic field between the

magnets creates the ability to steer the proton to a specific location. This gives the possibility to change the form of the *fluence*, which will be described in Section 2.2 but simplified is an intensity map of the beam. This is the foundation of proton *pencil beam scanning* (PBS).

Lateral deviation

Besides the type of interaction where the proton excites an electron from its atom, the proton can also interact with the atom nucleus. If the proton comes close enough to the nucleus, it will be repelled, causing a change in its directory. This causes a dose that is smeared out along the main path. [Newhauser and Zhang, 2015] Furthermore, when the secondary particles, i.e., the electrons, are excited from their atom, their path can be in any direction. This also contributes to the smeared out dose along the main path, making a spot dose look pear shaped. [Maughan, 2022] Lastly, the proton can also undergo non-elastic nuclear interaction, where the proton is absorbed and a neutron is ejected [Newhauser and Zhang, 2015].

2.2 Fluence

Fluence is a physical property of a radiation beam, describing properties such as the direction, energy and position distribution. This project will regard a simplified view of it; a two dimensional intensity image of the radiation, as seen from the source of the radiation. The ability to in some extent decide the form of the fluence is the basic idea of *intensity modulated proton therapy* (IMPT). In these treatment models the fluence is divided into several spots, often as in a grid, where the intensity of each spot, the *spot weights*, can be chosen independently.

Spot weight optimization

This project is limited to *spot weight optimization*, i.e., the parameters that are optimized are the before mentioned spot weights. An advantage of spot weight optimization is that the dose is linear to the spot weights. This causes the optimization problem to be convex if *1)* the problem is convex in the dose and *2)* the parameter set is convex. [Bokrantz, 2013] As will be seen, the dose objectives will be (sometimes one-sided) quadratic, and the parameter set will be convex.

2.3 Treatment planning

For the patient to receive the best possible treatment, the treatment needs to be adapted to the patient. The process of finding a parameter setting that corresponds to a good dose distribution in the target is called *treatment planning*, and the outcome of the process is a *treatment plan*. The parameters for this thesis are, as already mentioned, the spot weights.

Characterization of a good treatment plan is such as a uniform dose in the tumor on a certain dose level and as little dose as possible in healthy tissue. Other criteria can be that the plan is robust to changes in the patient geometry [Sundström, 2021] and that it can be delivered fast.

Forward treatment planning

One intuitive way to find a treatment plan is to manually adjust the parameters and evaluate the dose corresponding to that setting. If the result is not good enough, the parameters are changed in a way that the person doing it thinks will yield a better result, something that requires experience. The mentioned steps are carried out until a satisfactory dose distribution is reached. Dose computation will be described in Chapter 4

Inverse treatment planning

In inverse planning, the procedure is the other way around. As mentioned in Section 2.1, different tumors need different amount of dose to achieve tumor control, and similarly for healthy tissue in the near region. With this information, a physician can specify both the upper limit and the lower limit of dose in different parts of the body, as well as the importance to fulfill them. These are the so-called clinical goals which are inserted in a mathematical model that can give a number of how well a certain dose satisfies the clinical goals.

A *treatment planning system* (TPS) is a program that models the target and delivery machine and with this handles the treatment planning. The TPS uses the previously mentioned mathematical model of the problem to optimize a delivery plan that satisfies the clinical goals. Chapter 3 describes the theory of the optimization.

3

Inverse treatment planning

The following chapter begins with introducing the optimization methods used in the thesis and a short motivation for them. The term improved precision is explained and the motivation for it. After that comes the mathematical formulation of the objective function and how the problem is formulated with the help of dose objectives.

3.1 Optimization methods

Machine learning algorithms

When training large neural networks the most common approach when it comes to optimization is using *stochastic gradient descent* (SGD). SGD replaces the gradient evaluation in *gradient descent* (GD), which is described next, with an unbiased stochastic approximation of it. In machine learning this amounts to performing the gradient evaluation on just a part of the whole data set, a so-called mini-batch, taking shorter time at the expense that noise is introduced in the gradient. When running SGD there is a trade-off between how fast each iteration takes and the accuracy of the step. The method has proved itself of performing well even for small batches. [Bottou, 2010]

Monte Carlo (MC) computes an unbiased approximation of the real solution, where more simulations correspond to higher certainty. Thus computing the gradient based on MC gives an unbiased stochastic approximation of the gradient. This serves as motivation why first order methods should also work in optimization based on noisy MC data.

Gradient descent

Gradient descent (GD), sometimes referred to as *steepest-descent*, is a so-called first order method, since it only uses the first derivative, in contrast to second order methods which also make use of the Hessian. The gradient $\nabla f : \mathbb{R}^m \to \mathbb{R}^m$ of a differentiable function $f : \mathbb{R}^m \to \mathbb{R}$ of m variables points in the direction of greatest ascent, thus a lower function value is reached in the opposite direction, which is beneficial when aiming to minimize f. The update rule for GD is as follows:

$$x_{n+1} = x_n - \alpha \nabla f(x_n) \tag{3.1}$$

where $\alpha > 0$ is the step length and the subscript indicates the iteration. If the step length is small enough and f is continuous, GD is certain to converge to a local minimum, except for some special initial points, e.g., saddle points. Under the same conditions but if the problem is convex, GD is certain to converge to the global minimum. A difficulty with GD is to choose the step size; a too small one will result in slow convergence and if it is too big the sequence of points might not converge, and maybe even diverge. This issue is often handled by including a *line search*, which is a one dimensional minimization problem in the search direction, i.e., the direction of the step.

Momentum

Another difficulty with GD is that it has trouble to navigate in so-called ravines, i.e., areas in the error landscape with curvature much sharper in one direction than in another. The result of using GD in such cases is that it tends to oscillate much in the direction with sharp curvature and only making small progress in the direction of not so sharp curvature. A solution to the mentioned problem is to include *momentum* of the gradient, i.e., including a decaying average of the past gradients when computing the step [Polyak, 1964]. The update formula for GD with momentum then becomes

$$\begin{aligned} g_n &= \gamma g_{n-1} + \alpha \nabla f(x_n) \\ x_{n+1} &= x_n - g_n \end{aligned} \tag{3.2}$$

where the scalar γ usually is about 0.9 [Ruder, 2017]. The result from the above update formula is that the updated momentum increases in directions where the gradient has the same sign as the momentum, whilst decreasing in the directions where the opposite holds. This will lead to less oscillation and at the same time acceleration in the not so sharp curvature dimension. [Qian, 1999]

In addition to hinder oscillation, taking the decaying average of current and past gradients may serve as a way to reduce the effect of the noise in the gradient. If there is no systematic error in the noise, averaging the gradients will cause the noise to decrease.

Nesterov accelerated gradient

The step g_n in (3.2) consists of two terms, the decayed sum of the previous gradients as well as the current gradient. Thus a *semi* step in the direction of the decayed sum of gradient will be taken regardless of the gradient in the current point. Because of this, it makes sense to instead of calculating the gradient in the current point x_n to calculate the gradient in the point $x_n - \gamma g_{n-1}$. This is called *Nesterov Accelerated Gradient* (NAG) [Nesterov, 1983] and can be written as

$$\begin{aligned} g_n &= \gamma g_{n-1} + \alpha \nabla f(x_n - \gamma g_{n-1}) \\ x_{n+1} &= x_n - g_n \end{aligned} \tag{3.3}$$

The effect of applying NAG is less overshooting as we have a sense of what the error landscape looks like around where the coming step will take us.

Pre-conditioner

While the gradient is cheap to calculate, it has some drawbacks; it only regards the steepness of the error landscape at the point investigated. It is desired to pre-condition each step with a term that regards also the curvature in the current point. The reason for having a pre-conditioner is to improve the condition number. Newton's method is an example of such a method, written as

$$x_{n+1} = x_n - H(x_n)^{-1} \nabla f(x_n) \tag{3.4}$$

where $H(x_n)^{-1}$ is the inverse Hessian at the current point x_n. This is derived from approximating the function with a second order Taylor expansion and taking a step to the minimum of this. Newton's method has local quadratic convergence if the right conditions hold, such as the initial point is near the solution [Nesterov and Nemirovskii, 1994]. However, calculating the Hessian is computationally expensive, the cost is $\mathcal{O}(m^2)$ as opposed to calculating the gradient whose cost is $\mathcal{O}(m)$, where m is the number of parameters and $\mathcal{O}(\cdot)$ represents the complexity w.r.t. input parameters. On top of this, the cost to invert the Hessian is $\mathcal{O}(m^3)$. One common technique is to approximate the Hessian with its diagonal elements, reducing the cost for both calculating and inverting it to $\mathcal{O}(m)$. The diagonal elements of the Hessian as an approximation of the Hessian, from now on referred to as *Hessian diagonal*, becomes a better approximation if the Hessian is diagonally dominant, i.e., its diagonal elements are large in comparison to the other elements on the same row. Approximating the Hessian with its diagonal elements has been done before, e.g., by Bordes et al. (2009).

Other methods that approximate the Hessian. *Quasi-Newton* (QN) methods are a family of optimization methods that iteratively approximate the inverse of the Hessian for each iteration with the help of the gradients. The difference between different QN methods is the update formula for calculating the next inverse Hessian approximation. The most common is BFGS (named after Broyden, Fletcher, Goldfarb and Shanno) and its limited memory version L-BFGS, which instead of storing the $m \times m$ big inverse Hessian approximation, it saves the gradients from the k latest iterations and at each iteration calculates the approximation. This saves a lot of storage when m is say >1000 and k is around 10 or 20.

The main advantage of QN methods compared to Newton's method is that the computationally expensive inverse operation is not needed, since the inverse Hessian is approximated directly.

Decaying average of Hessian

In the same way as the momentum term might decrease the effect of the noise in the gradient, treating the Hessian as a decaying average over the current and previous Hessians might prove to be a good idea. This is due to local curvature information for noisy functions might be misleading.

ADAHessian is a method that, in addition to spatial averaging which will not be discussed, preconditions a step with the inverted square root of decaying average of the squared Hessian diagonal. The reason for using the squared Hessian and then square root is due to stability reasons, which is also adopted in *Adaptive Moment Estimation* (Adam) and *Root Mean Squared Propagation* (RMSProp), except these methods use their squared gradient. [Yao et al., 2021].

Putting it all together

Equation (3.5) explains how the decaying average on the Hessian is handled when integrating it with NAG; the previous decaying average of gradients and Hessian diagonals are multiplied with their respective decay factor $\gamma_1, \gamma_2 \in (0,1)$.

$$\begin{aligned} g_{n+1} &= \nabla f\left(x_n - \alpha \frac{\gamma_1 m_n}{\sqrt{\gamma_2 v_n}}\right) \\ h_{n+1} &= \nabla^2 f\left(x_n - \alpha \frac{\gamma_1 m_n}{\sqrt{\gamma_2 v_n}}\right) \\ m_{n+1} &= \gamma_1 m_n + g_{n+1} \\ v_{n+1} &= \gamma_2 v_n + h_{n+1}^2 \\ x_{n+1} &= x_n - \alpha \frac{m_{n+1}}{\sqrt{v_{n+1}}} \end{aligned} \tag{3.5}$$

Improved precision

When evaluating an optimization algorithm for deterministic optimization, the wish is often to find a satisfactory point in as few iterations as possible, i.e., for as few optimization steps as possible. For the case of MC based optimization, it is probable that the fewest iterations for finding a satisfactory point is reached when running as many MC simulations as possible for each iteration, because the more simulations computed the better is the approximation as will be seen in Section 4.2, and then the optimization steps should have a higher quality. However, computing simulations is time consuming, thus it is not probable that the fewer iterations run, the less time the optimization takes. With that said, the same trade-off between time and quality of the step occurs for MC based optimization as for SGD, as discussed in the beginning of Section 3.1.

The idea of *improved precision* (IP) consists of having high uncertainty in the beginning of optimization and then gradually decreasing it, i.e., improving precision, in this case increasing the number of simulations. The reason for it is to re-

duce the total number of simulations for the optimization, thus reducing the time. It is motivated by that in the beginning, the current point is relatively bad and therefore a step calculated with high uncertainty should still improve the point. As the point becomes better, the error that a step is intended to decrease becomes smaller than the uncertainty of the step. Thus more simulations should be computed to decrease the uncertainty and make an improvement of the optimization parameters.

Although the concept of IP has been studied before, e.g., for solving least-square problems based on MC [Pfeiffer and Sato, 2018], it has to the knowledge of the author not been used before in the setting of performing more simulations over the procedure of an optimization.

3.2 Mathematical formulation

When considering vectors, all are assumed to be column vectors.

Patient

The part of the patient that is subject for any considerable amount of dose is modeled as a three dimensional volume which is discretized into a set of *voxels* in a grid, indexed as $v \in \mathbb{V}$. A voxel is a box-shaped volume which typically is a few millimeters wide.

Dose

Each voxel v holds a dose value d_v. The vector $D \in \mathbb{R}_+^{|\mathbb{V}|}$ contains the dose in all voxels, where $\mathbb{R}_+ = \{y \in \mathbb{R} \mid y \geq 0\}$.

Parameters

The optimization parameters are the spot weights x_s, each belonging to a spot $s \in \mathbb{S}$. The parameter vector $x \in \mathbb{R}_+^{|\mathbb{S}|}$ contains all the spot weights. Each spot belongs to a segment which in turn belongs to a beam. A segment consists of all the spots in a beam whose protons have a certain range. A beam consists of all segments from a certain source.

Minimization problem

The problem is formulated as the following; minimize the objective function Φ subject to the constraints $Ax \leq b$. Mathematically, this looks like

$$\begin{aligned} \min_{x \in \mathbb{R}^N} \quad & \Phi(x) \\ \text{s.t.} \quad & Ax \leq b \end{aligned} \tag{3.6}$$

where N is the number of parameters. The objective function Φ is a sum of the dose objectives, which will be defined in (3.8) and (3.9).

Constraints. The only constraint for the problem is that the parameters cannot be negative, since that would mean to have negative irradiation which is not physically possible. Thus the problem can be written compactly as

$$\min_{x \in \mathbb{R}^N_+} \quad \Phi(x) \tag{3.7}$$

Dose objectives

The objective function is dependent on the dose d_v in each voxel v. Before the optimization begins, each voxel in the body is given a dose objective, i.e., a desired dose set by a treatment planner/physician, as well as an importance factor which is a constant that signals the importance to fulfill the desired dose. This is done by having a desired upper and/or lower dose value in each voxel, and a function that sets a penalty when the desired dose level is not satisfied. [Siggel, 2012] The requested function is for the case of desired lower respectively upper dose written as

$$f_{\min}(D) = \sum_{v \in \mathbb{V}} (w_{\min,v}(d_{\min,v} - d_v))^2_+ \tag{3.8}$$

$$f_{\max}(D) = \sum_{v \in \mathbb{V}} (w_{\max,v}(d_v - d_{\max,v}))^2_+ \tag{3.9}$$

where the operator $(\cdot)_+ = \max(\cdot, 0)$, the scalar $w_{\min/\max,v} \geq 0$ is the weight of the lower/upper dose objectives in voxel v, a high value indicating a high importance to satisfy the dose objective, $d_{\min/\max,v}$ is the desired lower/upper dose for voxel v and D is a vector containing the dose in all voxels.

For some parts, there might be penalty just on one side of the desired dose, e.g., for healthy tissue where there is no need to have a lower bound on the dose. In contrast to this, for the tumor, where it is considered best to have as uniform dose as possible, penalty is set both when the dose is above and below the desired one.

Dose falloff region. Conflict of interest between dose objectives inside and outside of the tumor can cause the dose in the tumor near the edges to decrease. To avoid this, a dose falloff region can be used. This region might have linearly decreasing desired dose, ranging from d^*_{tumor} to d^*_{outside}, starting at the edge of the tumor and spreading outwards a given distance. The weight on this new dose objective, i.e., the importance to adhere to this linear decreasing dose, should be less than the weight on the desired dose in the tumor. [Bokrantz, 2013]

Objective function

The objective function Φ can now be written as follows:

$$\Phi(x) = f_{\min}(D(x)) + f_{\max}(D(x)) \tag{3.10}$$

The dose D in the voxel grid depends on the spot weights x through the dose computation, described in Chapter 4. For each dose D the objective function computes an objective value, also called *error*.

Gradient

The gradient for the objective function is computed by differentiating (3.10) w.r.t. each parameter x_s as follows:

$$\frac{\partial \Phi}{\partial x_s} = (\nabla_D \Phi)^T D_s = \sum_{v \in \mathbb{V}} \left(\frac{\partial \Phi}{\partial d_v} \frac{\partial d_v}{\partial x_s} \right) \tag{3.11}$$

where the *dose derivate field* $\nabla_D \Phi$ describes how the error depends on the dose in each voxel and the *spot dose* D_s describes how the dose in each voxel depends on parameter x_s, which will be described more in Chapter 4. Notice that the chain rule is applied; from first differentiating w.r.t. the parameter x_s, we then differentiate w.r.t. the dose in all voxels D and differentiate the dose in each voxel w.r.t. the same parameter x_s which is equal to the spot dose D_s.

The dose derivate field $\nabla_D \Phi$ is calculated by differentiating (3.8) and (3.9) w.r.t. to the dose in each voxel, for voxel v it can be written as follows:

$$P_{v,\min} = -2w_v^2 (d_{\min,v} - d_v)_+ \tag{3.12}$$

$$P_{v,\max} = 2w_v^2 (d_v - d_{\max,v})_+ \tag{3.13}$$

Let the vector $P \in \mathbb{R}^{|\mathbb{V}|}$ contain all $P_v = P_{v,\min} + P_{v,\max}$ for each voxel v. Then (3.11) can be written as

$$\frac{\partial \Phi}{\partial x_s} = P^T D_s \tag{3.14}$$

Hessian diagonal

Recall that creating and inverting the Hessian is computationally expensive, while approximating it with its diagonal demands a lot less computation. Each element of the Hessian diagonal is computed by taking the second partial derivative w.r.t. the specific parameter x_s as follows:

$$\frac{\partial^2 \Phi}{\partial x_s^2} = \sum_{v \in \mathbb{V}} \left(\frac{\partial^2 \Phi}{\partial d_v^2} \left(\frac{\partial d_v}{\partial x_s} \right)^2 \right) \tag{3.15}$$

The term $\frac{\partial^2 \Phi}{\partial d_v^2}$ is derived from differentiating (3.12) and (3.13) w.r.t. the dose in voxel v. For voxel v it is written as follows:

$$H_{v,\min} = 2w_v^2 \theta(d_{\min,v} - d_v) \tag{3.16}$$

$$H_{v,\max} = 2w_v^2\theta(d_v - d_{\max,v}) \tag{3.17}$$

where $\theta(\cdot)$ is the Heaviside function. Let the vector $H \in \mathbb{R}^{|\mathbb{V}|}$ contain $H_v = H_{v,\min} + H_{v,\max}$ for all v. Equation (3.15) can then be rewritten as follows:

$$\frac{\partial^2\Phi}{\partial x_s^2} = H^T D_s^2 \tag{3.18}$$

where D_s^2 is element-wise product of the elements.

Observe that the Heaviside function is not continuous, consequently the Hessian diagonal is not defined in all points.

Uncertainty propagation

There are several ways to find the uncertainty in the error. One way is to calculate it in batches and find the standard deviation through *propagation of uncertainty* as follows:

$$s_f^2 = \sum_i \left(\frac{\partial f}{\partial x_i}\right)^2 s_{x_i}^2 \tag{3.19}$$

where s_f represents the standard deviation of the objective function f and s_{x_i} represents the standard deviation of variable x_i, under the assumption that the variables are independent. However, since the objective function depends on the dose in each voxel, the uncertainty in the voxel dose should be mirrored in the objective function uncertainty. A better way to do so is to let x in (3.19) be the dose in the voxels [Siggel, 2012]. Then the standard deviation of the objective function can be approximated as follows:

$$s_f = \sqrt{\sum_{v\in\mathbb{V}} P_v^2 s_{d_v}^2} \tag{3.20}$$

Uncertainty affecting error

Suppose there is an underlying true dose distribution D_t that the MC based dose distribution D_{MC} approximates. Further suppose that D_t is close to the desired dose distribution D_d. Since D_{MC} corresponds to D_t with added noise, the dose in many voxels will deviate further from the desired dose in each voxel, and the objective function will increase if the dose deviates in such direction. Furthermore, with increasing uncertainty the objective function will increase even more. [Siggel, 2012]

Summary

The objective function depends on the dose distribution which is discretized into voxels, each voxel containing a dose value. The clinical goals are defined by the dose objectives in the voxel space, thus they build up the objective function. The

dose in each voxel depends on the spot weights and is computed by the dose engine, described in Chapter 4. The only constraint in the parameter space is non-negativeness.

4

Dose computation

A dose engine is an algorithm that computes the dose in the body. The inputs are such as the parameters (in this case the spot weights), geometry of the target, type of particle and source position.

This chapter begins with describing an analytical dose algorithm. Then it covers the background of Monte Carlo (MC), why it works and how it can compute dose. It ends with how MC is used in radiotherapy treatment today.

4.1 Analytical dose engine

Analytical dose engines are, in contrast to MC dose engines, completely deterministic. They are fast at the expense of accuracy, especially in heterogeneous mediums such as bone and cavities [Fiorini et al., 2018]. A common thing for analytical dose engines is that the whole dose is unambiguously determined by the input parameters. An example of an analytical algorithm used for proton dose calculation is *pencil beam algorithm* (PB).

Pencil beam algorithm. The idea behind PB algorithms is to first compute the kernel of an infinitesimally thin pencil beam in water, i.e., the dose this pencil beam deposits. The kernel is then used for convoluting the fluence, taking into account the thickness of the medium only on the central axis of each point on the fluence. Since only the middle of the beam is regarded, the dose is poorly represented in e.g. lungs or bones that happens to lie right beside the central axis. This is because these have different thickness, but the algorithm deposits dose in these volumes just as if the thickness there was the same as on the central axis. [Elcim et al., 2018]

4.2 Monte Carlo dose engine

General Monte Carlo

MC is a statistical method based on randomly sampling from a probability distribution many times and categorizing the result to find out about the underlying system.

It is often used for approximating multi-dimensional integrals and solving partial differential equations.

Monte Carlo dose engine

MC is often regarded as the most accurate way to compute dose. It is based on randomly simulating individual particles moving through the body and their interactions based on the different characteristics of the mediums in the body, such as density and cross-sections for particle interactions. This is done both for the primary particle as well as its secondary particles. Given that enough particles are simulated and that the physics of particle interactions are correctly modeled, the MC dose engine provides a good approximation. In recent years, fast MC algorithms have become available that speed up simulations by, e.g., only considering the most important interactions and stopping simulation of a particle after the energy have dropped below a certain threshold. [Teoh et al., 2020] The simulation of a particle and its secondary particles is called a history and is described below.

Particle history. A primary particle is sampled from the source, obeying the geometry of the problem on the way to the body. When entering the body, the distance to the first interaction is sampled based on the total cross section in its path. When a distance is found, the type of interaction is sampled based on the cross sections for the different types in the medium, as well as the interaction parameters, i.e., scattering angle for the primary particle, energy locally deposited and energy and angle for eventual secondary particles. Lastly, the distance until an interaction is sampled again for the primary and secondary particles. The steps are repeated, until all particles have left the simulated geometry, or their energy have fallen below a threshold in which case all their energy is deposited in that volume. Figure 4.1 illustrates a history. Due to the cross section's dependency of a correct model of the body, it is important that the CT images, which the model of the body often is based on, is correct and the different regions are correctly identified. [Seco and Verhaegen, 2013]

Statistics of Monte Carlo. Let X_{vs}^h represent the dose deposited at voxel v by spot s by the h:th history. X_{vs}^h is a random variable that depends on the fluence dependent initialization of the primary particle and the geometry of the body it travels through. Furthermore, the mean dose in voxel v from spot s is calculated by averaging over all N histories as follows:

$$D_{vs} = \frac{1}{N} \sum_{h=1}^{N} X_{vs}^h \tag{4.1}$$

Assuming that the histories are independent enough and that N is sufficiently large, according to the central limit theorem, the mean dose D_{vs} will be approximately normally distributed. This in turn implies that the standard deviation of the mean dose is approximately proportional to $1/\sqrt{N}$. This is the basic idea of the

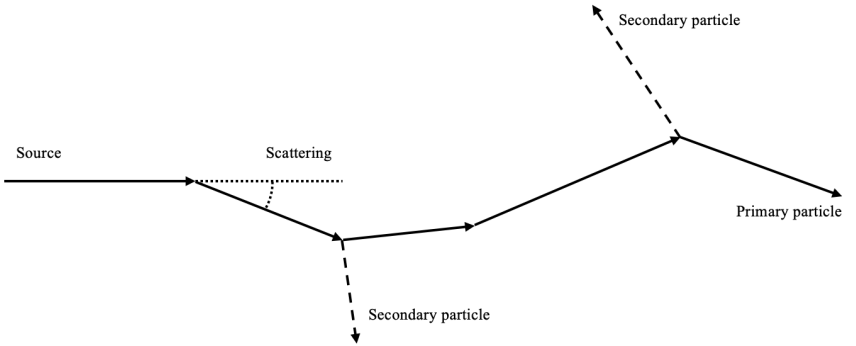


Figure 4.1 Illustration of an MC simulation. The particle is simulated from the source, whereas it undergoes stochastic interactions such as scattering and creation of secondary particles. The total effect from a simulated primary particle is called a history.

MC method, assuming that the system is well modeled the result of the method will converge toward the real solution as N grows large.

Fast Monte Carlo dose engine

In recent years so-called fast MC dose engines have been introduced to the market. The fast differ from the general purpose MC dose engines in the way that they are designed to be able to compute dose in time frames that are acceptable for clinical use - this is achieved, e.g., by only considering the most likely interactions. They are also designed to run on multiple threads (about 100,000) on GPUs, where each of these threads is typically assigned a history which it simulates.

Due to the large number of threads, the memory for each thread is limited, and thus the big data chunks such as the voxel grid and gradient vector is stored centrally.

Deposit dose. Each time dose is deposited in a voxel simulated by a thread, the thread directly writes this to the centralized dose voxel grid, or in the case the history is simulated to compute the gradient, the dose in the voxel is multiplied with the voxel element in the dose derivative field P as in (3.14). When the thread has written the dose to the correct instance, it is thrown away.

Monte Carlo dose normalization

The dose computed with MC must be normalized with the number of histories, so that the result is not dependent on the number of histories simulated, except for generating a result with less noise. Except for this, the dose must also be multiplied

with the sum of all the spot weights to correspond to the magnitude of the total beam.

4.3 Monte Carlo in radiotherapy today

Today, MC in radiotherapy treatment is mainly used in two situations; for evaluation and for pre-computing spot doses.

Treatment plan evaluation using Monte Carlo

After a parameter setting has been found, e.g., through inverse planning based on an analytical dose engine or through forward planning, we may want to evaluate the solution to make sure that the plan fulfills the clinical goals, in which case MC can be used. The procedure is then to perform a dose computation with MC based on the found parameter settings. If the computed dose is satisfactory, the plan is used for treatment.

Optimization using pre-computed spot doses

MC can be used to pre-compute each spot dose D_s, i.e., how much spot s contributes with dose to each voxel. The total dose is a linear combination of all the spot doses, and can be expressed as follows:

$$D(\omega) = \sum_{s \in \mathbb{S}} D_s \omega_s \tag{4.2}$$

where ω_s is the weight of spot s and $\mathbb{S}$ is the set of spots. The objective function Φ depends on the total dose through the dose objectives, as described in Section 3.2. The gradient can then be computed as the scalar product between the dose derivative field, i.e., the objective function differentiated w.r.t. the dose in each voxel, and the dose contribution from each spot D_s as follows:

$$\frac{\partial \Phi}{\partial \omega_s} = (\nabla_D \Phi)^T D_s \tag{4.3}$$

The spot weights can now be optimized deterministically using the pre-computed spot doses. The spot doses still have noise in them, since they were computed with a MC dose engine, and the quality of the optimal point found by the optimizer will be limited by the noise in the spot doses. However, the noise will not affect the procedure of the optimization.

The main limitation with using MC to pre-compute spot doses is the storage limitation. The amount of storage needed is linear with the number of spots used, as well as with the number of voxels the target is represented with. The size of each spot dose D_s is the number of voxels, and the number of such spot dose contributions is the number of spots. This makes it infeasible to use the method for e.g. carbon PBS, due to the number of spots needed (around 40k instead of 10k).

Another problem for pre-computed spot doses is that the desired statistical uncertainty must be decided at the start of the optimization, generally applying for all the spots. This is done because the solution reached by the discussed method is limited by the quality of the spot doses. Since there are many spots that will have a low weight in the end, or even zero weight, the long calculation time for these spots was in vain.

A beamlet-free MC based optimization method

To avoid the computationally expensive procedure of calculating the spot doses, Pross et al. (2023) developed a beamlet-free algorithm which incorporates MC in the spot weight optimization. The algorithm views the spot weights as a probability distribution, in each iteration sampling a spot from this probability distribution and simulating a batch of particles. The spot probability corresponding to the simulated spot is changed according to the impact of the simulated particles, taking into account the current dose, similarly to gradient descent, while the dose from the simulated particles is accumulated to the total dose. The probability distribution in the end is the result from the optimization. Due to constantly sampling from the probability distribution and regarding all simulations for the total dose, there is no need for a final dose computation.

5

Implementation

This chapter begins with explaining how the parameters are connected to the fluence and then the implementations of the two dose engines are explained. It goes on with describing how some features of the optimization are handled and declares the optimization algorithm used. Lastly, the experimental problem and general settings for the optimization are described.

5.1 Parameters to fluence map

The fluence map is represented with a matrix with the same size as the voxel grid looking from the current radiation source, and with the same resolution. The spots in a segment, explained in Section 2.1, are distributed as in Figure 5.1, where spot s has the weight x_s. Also, between segments the spots are moving in the pattern shown in the figure, as described also in Section 2.1. The pattern and the moving of the spots makes all positions in the matrix filled for four consecutive segments, except for the edges, which if the parameter space is big enough for the problem, will be zero either way.

5.2 Dose engine

For the project, the optimizer was not implemented in a developed treatment planning system with an integrated Monte Carlo (MC) dose engine, described in Section 2.3. Instead, a simplified MC dose engine was implemented in Matlab, with the aim of capturing the stochastic qualities, and not the accuracy with which commercial MC engines model real particle interactions. Along with the simplified MC dose engine, a matching analytical dose engine was implemented such that the dose from the MC dose engine converges to the dose from the analytical dose engine as the number of histories increases.

The fact that the MC dose converges to the analytical dose creates the ability to compare the convergence based only on the noise introduced, which would not be possible if a commercial analytical and MC dose engine had been used.

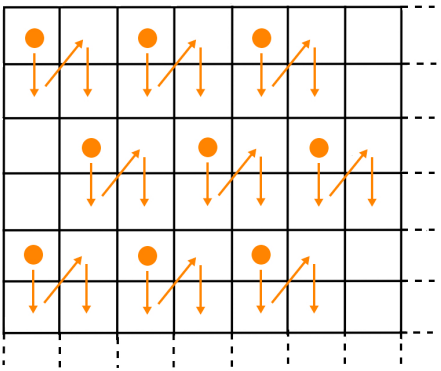


Figure 5.1 The figure shows the pattern of the spots in a segment, as well how the position of each spot moves between segments. Note that the same spot can have different intensity between segments.

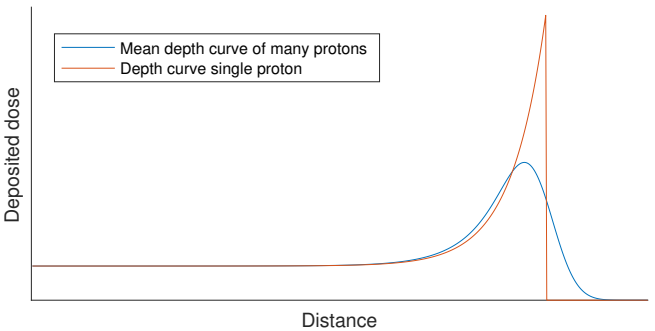


Figure 5.2 The depth curve for a mean proton at a specific energy (red) and the mean depth curve for many protons (blue), i.e., the Bragg curve. The Bragg curve is smeared out due to the energy of the protons deviate as well as they take different paths due to scattering. The curves are not formed from real data but are approximates.

Analytical dose engine

Proton depth curve. As described earlier, the proton continuously deposits energy along its path, thus the proton is modeled such that it deposits energy at every voxel it passes, according to its depth curve. As previously mentioned, the Bragg curve for many protons is smoother than the curve for a single particle, due to the range is normally distributed around the mean, as in Figure 5.2. For the analytical dose engine this is modeled through convoluting the idealized depth curve for a particle with the mean energy for the specific segment with a normal distribution kernel. The depth curves used, as seen in Figure 5.2, are simplified.

Lateral deviation of fluence. The fluence map is convoluted with a two dimensional normal distribution kernel to model the lateral deviation of the protons and how the secondary particles spread the dose out, as described in Section 2.1.

Fluence to dose. The dose distribution on each depth layer in the voxel grid is equal to the convoluted fluence times the value of the depth curve at the current depth.

Monte Carlo dose engine

Proton depth curve. To model the smeared Bragg curve for the MC dose engine, for each history within a segment the depth curve of a single proton is offset by an amount sampled from the normal distribution, where the normal distribution has the same properties as the kernel used for the analytical dose engine.

Lateral deviation of fluence. To sample the position of a particle in the fluence for MC, first an index is sampled from the spot weights, resulting in more sampled particles from a spot with high weight. To that position is then a deviation added in both cardinal directions, sampled form a normal distribution, with the same properties as the kernel with which the fluence was convoluted for the analytical dose engine.

Fluence to dose. When the position of the particle for the current history is sampled, the voxels that are on a straight line from that position is subject for receiving dose. The dose is deposited according to the depth curve for that specific history, as just described. The dose is also multiplied by the MC dose normalization factor, described in Section 4.2, which is the sum of the spot weights divided by the number of histories simulated to calculate the dose.

Computing gradient and Hessian diagonal

The elements of the gradient describes how the error is changed when taking a small step in each direction, and the Hessian diagonal in a similar way, thus they dependend on the dose at the current point. This can be seen in (3.14) and (3.18) which depend on P respectively H, which in turn depend on the dose at the current point. To compute the gradient and Hessian diagonal, both the *total dose* and *spot dose* must be computed, i.e., what is the dose distribution for the current spot weights and what is the dose contribution for each individual spot s, respectively. The above creates the ability to choose how many histories are used to compute the different parts of the gradient and Hessian diagonal.

The procedure of computing the total dose is described above. The spot dose D_s for spot s is computed by setting the spot weight x_s to one and the rest to zero and computing the dose.

Addressing fast MC dose engine´s way of throwing dose away. As discussed in Section 4.2, the procedure of fast MC dose engines is to throw away a deposited

dose directly after it has been computed and written. This creates no problem when computing the total dose and the gradient, since these properties are linear to the dose deposited by a single history. However, the Hessian diagonal is not linear to the dose from a single history, as seen in (3.18).

From Cauchy-Schwarz inequality it can be derived that

$$\left(\sum_{h=1}^{N} d_{vh}\right)^2 \leq \sum_{h=1}^{N} d_{vh}^2 N \tag{5.1}$$

where d_{vh} is the dose in voxel v from history h from a spot and N is the number of histories from the current spot. Working with the fast MC dose engine as is would require to use the right hand side of (5.1) when calculating the Hessian diagonal, due to there is not enough memory to store all the spot doses at the same time. Then the dose from a history deposited in a voxel is directly squared and multiplied by the number of histories N and H, as in (3.18). This would overestimate the Hessian, thus causing the optimizer to take shorter steps, which is arguably better than taking steps too long.

Ideally, however, the left hand side would be used, i.e., the total dose in voxel v from all histories for a certain spot squared. To overcome the problem of having to store all spot doses at the same time, instead only a fraction of the spots could be accumulated simultaneously. When enough histories have been simulated for a spot, the entire dose in each voxel is squared and multiplied with H as in (3.18), thus according to the left hand side in (5.1). The dose engine implemented assumes that the total dose for a spot is known, except for one case, presented in Section 6.3.

The above is not an issue for the analytical dose engine, as discussed in Section 4.3 the spot dose is generally precomputed and saved before the optimization begins, thus the whole spot dose is known for each voxel and the left hand side of (5.1) can be used.

5.3 Optimization features

Improved precision

The idea of improved precision (IP) is, as explained in Section 3.1, that higher uncertainty is accepted in the beginning of the optimization, when the current point is of poorer quality compared to later points. It is implemented as follows; given start values from sequences representing how many histories are to be used for both total dose and spot dose, from a given iteration the current point is compared to some previous point. If no improvement has been made considering the errors and their uncertainty, the next values in the sequences are used. The motivation for starting after some iterations have passed is partly to have some value before to compare with, partly because some oscillation might be to expect in the beginning due to

uncertain steps, but the trend might be downwards anyways and that should not be wasted.

Handling constraints

As stated in (3.7) the only constraint of the problem is for the spot weights to be non-negative. This is handled by projecting any parameter that happens to become negative back to zero.

Avoid calculating zeroed parameters

To decrease computing time the spot dose was not calculated for parameters that had been zero for the last three iterations. The motivation for this was that the step with high certainty would be zero or negative, since it had been so at least the three last iterations. To not fall in the trap of never giving the parameter a chance to become positive again, the spot dose was calculated for all parameters every 30th iteration.

Stopping criterion

A necessary condition for a point x to be a minimum is that $\nabla f(x) = 0$. A common stopping criterion for deterministic optimization, i.e., optimization without noise, is thus to investigate if the norm of the gradient at each point is less than a certain tolerance, and if so consider the point to be the minimum. For MC based optimization, the above procedure does not work since the gradient will contain noise and thus it will differ from its actual value. The stopping criterion used was instead to set a maximum number of iterations and a maximum number of histories, and stop the optimization when any of them was passed. The maximum number of histories was compared with the number of histories used, i.e., accumulated histories used for both computing total dose as well as computing all the spot doses. If a parameter's spot dose not was computed for a certain iteration, as explained above, it did not contribute with any histories to the total number of histories used.

5.4 Optimization algorithm

The fundamental difference of integrating MC dose calculation in the optimization process compared to optimizing spot weights on pre-computed spots, as explained in Section 4.3, is that the dose from each spot is not stored but constantly re-computed. For each iteration, to find the step first the total dose must be computed based on the current spot weights to find P and H, and then each spot dose is computed and the dose from each history is written to the gradient and Hessian diagonal vector according to (3.14) and (3.18). Lastly, a step is taken based on (3.5).

The optimization algorithm used can be found written in pseudo-code in Algorithm 1.

Algorithm 1 The procedure of the optimizer.

$x \leftarrow x_0$ ▷ Initiate variables
$m_0 \leftarrow 0$
$v_0 \leftarrow 0$
$no\_hist \leftarrow$ `first value in history sequence`
for $i = 1 : max\_itr$ **do**
  $D_t = f(x_i - \alpha \frac{\gamma_1 m_i}{\sqrt{\gamma_2 v_i}}, no\_hist)$ ▷ Calculate dose
  if `need to improve precision` **then**
    $no\_hist \leftarrow$ `next value in history sequence`
  end if
  `Compute` P `and` H `from` D_t
  for $s = 1 : length(x)$ **do** ▷ For all spots
    for $h \leq no\_hist$ **do** ▷ For all histories in that spot
      `Simulate particle` h `to get dose` D_h
      $g(s) += PD_h$
      $h(s) += HD_h^2$
    end for
  end for
  $m_{i+1} = \gamma_1 m_i + g$ ▷ Update decaying average
  $v_{i+1} = \gamma_2 v_i + h^2$
  $x_{i+1} = x_i - \alpha \frac{m_{i+1}}{\sqrt{v_{i+1}}}$ ▷ Update point, project negative to zero
end for

5.5 Problem

The experimental problem that is used for this project is a simplified one that aims to resemble the geometry in head and neck cancer. It consists of the tumor volume which is the tonsils, shaped like a 'c', surrounding the trachea which should have minimal dose, see Figure 5.3. The voxel grid representing the area of interest is 50x50x20 voxels and the area constituting the tumor and trachea is 20x30x16 voxels.

The target was irradiated with two beams from opposite sides, each with 12 segments such that consecutive Bragg peaks was at equidistant and covered the whole target volume.

5.6 Optimization settings

If not stated differently in a certain section, the settings that were used to get the results in Chapter 6 are found in Table 5.1.

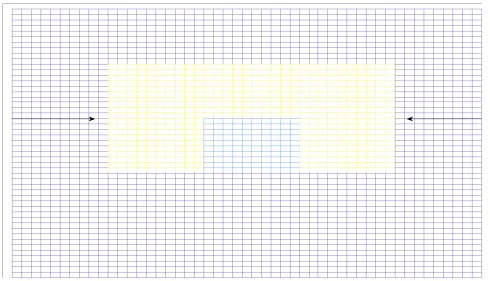


Figure 5.3 The problem setup with the tumor (yellow), organ at risk (blue) and healthy tissue (purple). The figure is at a cross section in the middle. The two beam directions are where the arrows come from.

Table 5.1 The standard settings for the optimization.
*Notice that the number of histories is to calculate the spot dose for one spot.

Entity	Value
Step size α MC	0.1
Step size α analytical	0.05
Momentum factor γ_1	0.9
Decaying average Hessian diagonal factor γ_2	0.9
History sequence total dose	$10^4 2^k$, $k = 0, 1, 2, ...$
Histories sequences spot dose*	$10^2 2^k$, $k = 0, 1, 2, ...$
Iterations until uncertainty can decrease	10
Number of segments	12
Initial point	All ones

6

Results

This chapter presents the results from different experiments, all based on the experimental problem introduced in Section 5.5

Three different errors will be discussed; analytical, MC and ideal MC. The analytical error comes from optimization based on the analytical dose engine. Both MC and ideal MC error is computed on points that come from optimization based on the Monte Carlo (MC) dose engine, with the exception that the ideal MC error is computed with the analytical dose engine. The ideal MC error thus gives a "correct" quality measure of the point, whereas the MC error provides a quality measure with noise, similar to what will be visible in a clinical setting.

When taking the norm, it is the Euclidean norm that is referred to.

6.1 General dose distribution and convergence results

Figure 6.1 shows the dose distribution at a cross section of the voxel grid, for the solution of the problem based on the MC dose engine. The desired dose in the tumor is 4. Notice the relatively uniform dose in the tumor.

It is common practice to present a *cumulative dose-value histogram* (DVH) for different regions in the target, such as tumor and nearby organ, when investigating methods for treatment planning, bench marking the investigated to the most common or the state-of-the-art method. A DVH relates the absorbed dose to tissue volume for each such region. Since the dose engines implemented do not try to be as physically correct as possible, there is no point in presenting such results. Instead, Figure 6.1 shows that the MC based optimization yields something that looks like the desired dose, and from now on only convergence plots or similar will be regarded.

To investigate how the dose computed with the MC dose engine converges to the dose computed with the analytical dose engine, the norm of the difference between the analytical dose and the MC dose was computed for an increasing number of histories for the MC dose. The results are shown in Figure 6.2. As can be seen, the norm of the difference is monotonically decreasing. These results have no value

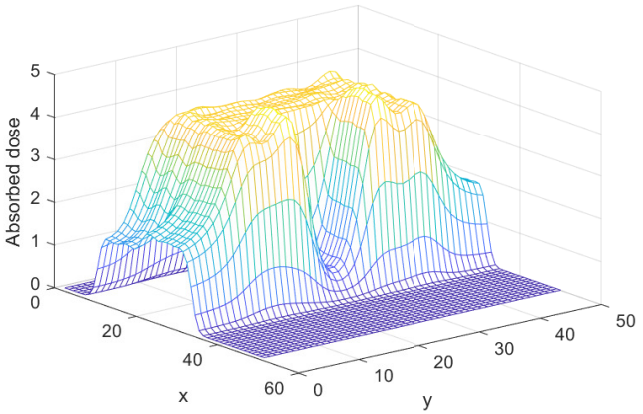


Figure 6.1 Dose distribution in a cross section of the target. The plateau is the 'c' shaped tumor and in the middle of it is the trachea where the dose should be minimized, all surrounded by healthy tissue. The treatment plan that corresponds to the dose distribution comes from optimization based on the MC dose engine, however, the dose distribution was computed with the analytical dose engine.

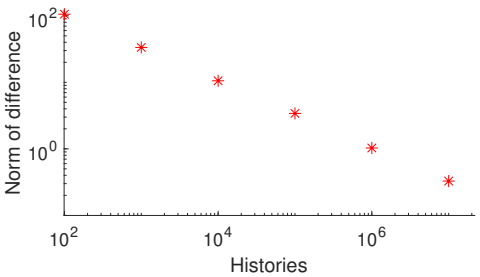
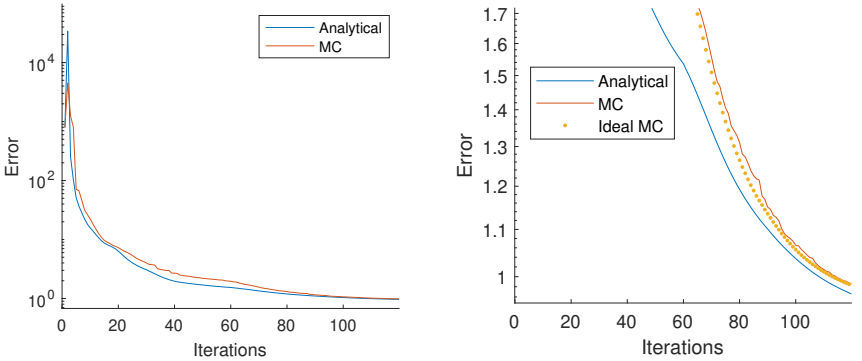


Figure 6.2 The norm of the difference between the analytical dose and the MC dose for increasing number of histories used to compute the MC dose. The point was chosen as one near the solution.

other than motivation that the analytical dose engine equals the MC dose engine computed with infinite number of histories.

Figure 6.3 shows the MC convergence that led to the point giving the dose distribution in Figure 6.1, as well as an analytical convergence. Notice that the two plots follow each other well, the analytical always a bit below.



(a) Convergence of analytical and MC dose engine.

(b) Same as (a) but zoomed in on the later part. Also contains the ideal MC convergence (orange).

Figure 6.3 Convergence both based on analytical dose engine (blue) and MC dose engine (red) for 120 iterations. The ideal MC error for each iteration during the MC optimization (orange) can also be seen in figure (b) which is zoomed in on the later part of the optimization for easier view.

6.2 Increased error due to uncertainty

As can be seen in Figure 6.3b the ideal MC error lies below the MC error for the same point. That can also be seen in Figure 6.4 which shows the difference between the MC and ideal MC error for each point in the MC based optimization, normalized with the ideal MC error at that point. After the initial increase, the difference is decreasing in clear steps which corresponds to when the number of histories to compute the total dose is increased, starting at around iteration 35, which aligns with the results in Figure 6.2. The initial increase is probably due to the choice of initial point, which is all ones according to Table 5.1. The point causes the noise not to have a big effect, since all the spot weights are equal and the noise probably cancels out. Furthermore, the increase within each step is due to the normalization number is constantly decreasing, since a better point is being reached.

The reason for the MC error to be higher than the ideal MC error is as previously mentioned in Section 3.2, the noise from the MC dose calculation causes the dose in each voxel to deviate from the ideal value. Since the dose penalties are (sometimes one sided) quadratic, a deviation towards a better dose level for a voxel causes less improvement in error than an equally big deviation towards a worse dose level causes degradation.

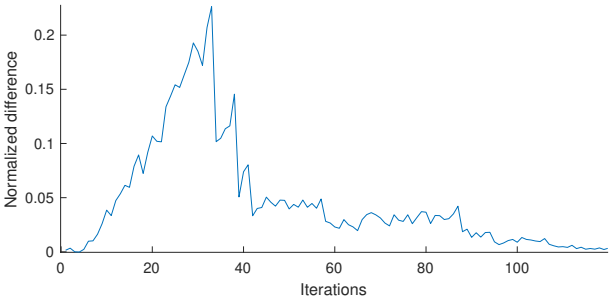


Figure 6.4 The difference between the MC and ideal MC error for each iteration during an MC based optimization, normalized with the ideal error. After the initial increase due to lower normalization number, the difference decreases as more histories are being simulated.

6.3 Quality of step dependence on number of histories

The following results intend to show how the quality of the gradient and the Hessian diagonal compared to the analytical such depends on the number of histories used to compute the total dose and the spot dose. The gradient and Hessian diagonal in a given point was computed for an increasing number of histories for two cases, namely *1)* when computing the total dose with the analytical dose engine and the spot dose with the MC dose engine (A-MC), and *2)* when computing the total dose with the MC dose engine and the spot dose with the analytical dose engine (MC-A). The entities were compared with an analytical gradient respectively Hessian diagonal in three ways; by taking the norm of the difference between them normalized with the norm of the analytical such, calculating the angle between them and comparing the length of them. The point was chosen as one near the optimal solution.

The results are shown in Figure 6.5. The MC dose converges to the analytical dose as seen in Figure 6.2, thus there is no surprise that the normalized difference for the gradient and the Hessian diagonal is decreasing for both the MC-A and the A-MC case. This, of course, also implies that the angles decrease and the fraction of the lengths converges to one.

Comparing the y axes in Figure 6.5 for the gradient figures (left) and the Hessian diagonal figures (right), the axes for the gradient has a larger scale. This implies that the gradient is much more sensitive to the number of histories used to calculate it than the Hessian diagonal is. That in turn means that there is not as important to have a high accuracy when computing the Hessian diagonal as for computing the gradient.

Looking at Figure 6.5a, it is much more beneficial to increase the number of histories to compute total dose than to compute spot dose, as the steepness of the MC-A case is greater. Note also that the histories for computing a spot dose must be scaled with the number of spots N_s, i.e., when computing the spot dose with 100 histories the number of histories spent is really $100 \cdot N_s$, where N_s often is around

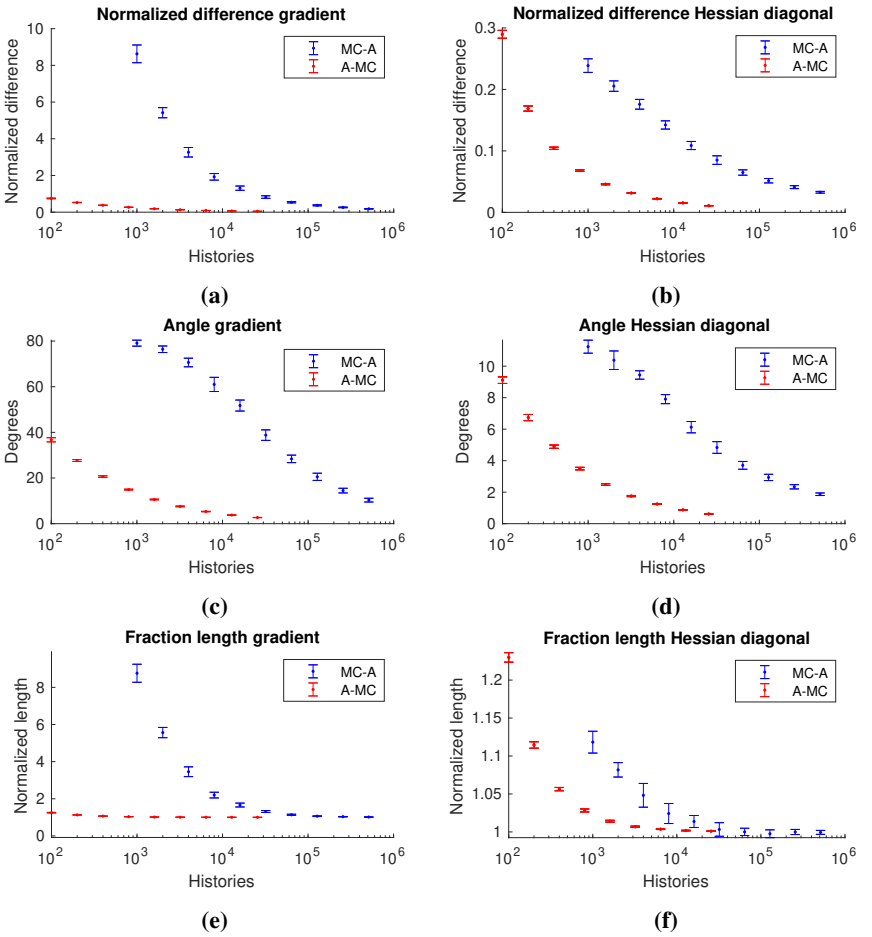


Figure 6.5 The analytical gradient and Hessian diagonal are compared with two semi MC gradient and Hessian diagonal for increasing number of histories. The two cases of semi MC is MC total dose and analytical spot dose (MC-A) and analytical total dose and MC spot dose (A-MC). The upper figures show the norm of difference between the semi MC and the analytical normalized with the norm of the analytical, the middle figures show the angle between the semi MC and the analytical and the lower figures show the norm of the semi MC divided by the norm of the analytical. The left figures are the comparisons of the gradient and the right figures are for comparisons of the Hessian diagonal. The error bars are one standard deviation, calculated from 20 batches. All gradient and Hessian diagonal computations were done for the same point which was one near the solution.

10,000.

Approximation of Hessian diagonal

The results in Figure 6.6 show the comparison with an analytical step for the case A-MC but where the Hessian diagonal is approximated (A-MC-approx), as discussed in Section 5.2, based on the qualities of fast MC dose engines. Instead of comparing the gradient and Hessian diagonal separately they are combined as a whole step, i.e., $H^{-1}g$ where H is the Hessian diagonal and g is the gradient. Interestingly, the direction for the A-MC-approx step is degraded as more histories are used to compute the step after 200 histories, as seen in 6.6b. Also, the length of the A-MC-approx step is decreasing from at the beginning being not even 1.5% the length of the analytical step, as seen in 6.6c. The fact that the step is shorter than the analytical is according to the hypothesis discussed in Section 5.2. This overall causes the normalized difference between the steps to degrade, which can be seen in Figure 6.6a, after 200 histories.

The degradation is probably due to that the approximation for the Hessian diagonal, where the number of histories is used as a scale factor, is a bad one. Notice that as the gradient is linear to the dose from each history, we can know that it has no effect on these results comparing to the results in Figure 6.5.

The overall results suggest that fast MC dose engines might need to be changed so that the whole spot dose can be computed to find the correct Hessian diagonal when incorporating it in the optimization process.

6.4 Improved precision

Only increase total dose histories

The results from Section 6.3 indicate that it might be of more interest to increase the number of histories to compute total dose than number of histories to compute spot dose for gradient and Hessian diagonal. The present section investigates how the convergence is affected by only increasing the number of histories to compute the total dose.

The convergence for MC dose engine, both for standard sequence of histories to compute spot dose as in Table 5.1 (standard) as well as not increasing histories to compute spot dose from the start value 100 histories (light), was computed for 70 iterations. As seen in Figure 6.7a, the convergence for the two cases (red and blue) is quite similar w.r.t. iterations. However, looking at 6.7b, where the error is plotted w.r.t. the total number of histories used, it can be seen that the convergence is significantly better for the light case. The error at the end for the two cases is 1.49509 and 1.54834 for standard respective light case, i.e., the standard reaches a lower error, but this is done at over 6 times the amount of histories used, while the standard case is around the same error as the light case at about 5.5 times the amount of histories used.

Since simulating histories is, despite fast MC dose engines, a time consuming

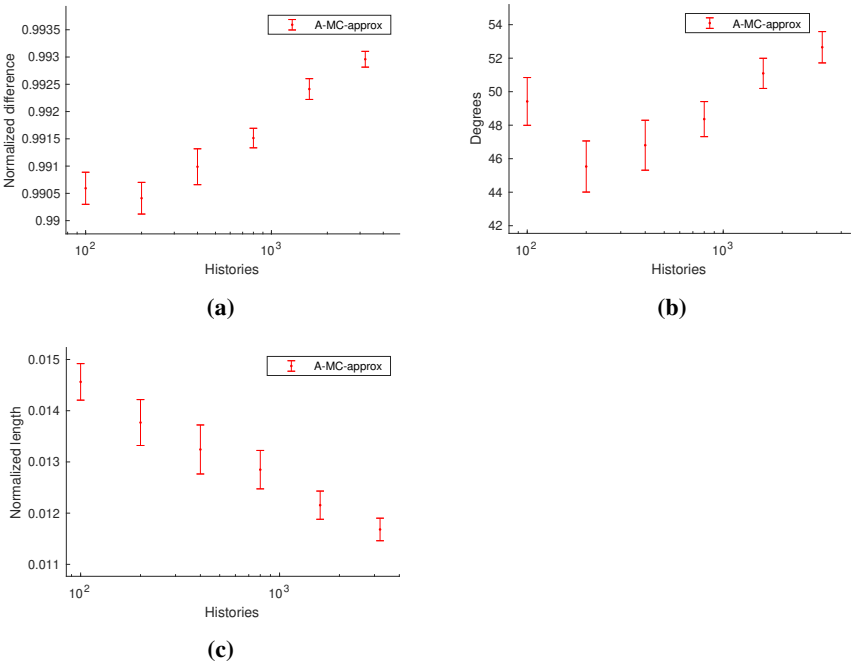


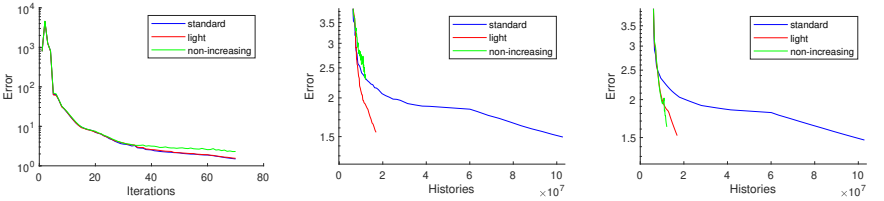
Figure 6.6 A semi MC step is compared with an analytical step for increasing number of histories. The semi MC step comes from calculating the total dose with the analytical dose engine and the spot dose with the MC dose engine (A-MC-approx), where the Hessian diagonal is approximated based on the discussion in Section 5.2. The figures show (a) the norm of difference between the semi MC steps and the analytical step normalized with the norm of the analytical step, (b) the angle between the semi MC steps and the analytical step and (c) the norm of the semi MC steps divided by the norm of the analytical step. The error bars are one standard deviation, calculated from 20 batches. All step computations was done for the same point which was one near the solution.

part, reducing it with such a big factor at the cost of not reaching as good point, might be worth it.

Increasing neither total nor spot dose

The following section investigates how the convergence is affected by not increasing histories to compute total dose nor spot dose.

The convergence based on MC dose engine was computed when the histories for total and spot dose was not increased from their start values at 10,000 and 100 (non-increasing) and compared to the standard case as described above. The optimization was performed for 70 iterations, and results are in Figure 6.7 (blue and green). The convergence is quite similar until around iteration 30, where the non-increasing case



(a) MC convergence w.r.t. iterations.

(b) MC convergence w.r.t. total number of simulated histories.

(c) Ideal MC convergence w.r.t. total number of simulated histories.

Figure 6.7 Convergence based on MC dose engine for standard history sequence (blue), when only increasing number of histories to compute total dose (red) and when not increasing total nor spot dose (green), w.r.t. iterations (a) and accumulated histories (b). (c) illustrates the same as (b) with the difference that the error in each iteration is the analytical one. Note that only the later part of the convergence is visible in (b) and (c), for easier view.

has an inferior convergence compared to the standard case, as seen in Figure 6.7a. Looking at convergence w.r.t. histories in Figure 6.7b, the non-increasing case has an inferior convergence until it catches up in the end of the optimization, i.e., at around iteration 70.

Analytical convergence

As seen in Section 6.2, the ideal MC error for a given point is lower than the MC error, as well as that the difference between the errors decreases as more histories are used for the total dose calculation. To see how this affects the convergence, the ideal MC convergence is illustrated in Figure 6.7c, for the same optimization runs as in the other two sub-figures.

As can be seen when comparing the ideal MC convergence in Figure 6.7c with the MC convergence in Figure 6.7b, the non-increasing case makes the best improvement. This is due that the total dose is always counted with 10,000 histories for the non-increasing case, while the standard and light cases have increased their number of histories to compute the total dose, due to IP.

6.5 Varying history sequence start values

In order to see how the convergence was affected by different start values on the history sequences, the MC based optimization was run for different combinations of start values both for calculating total dose as well as spot dose. The start values for the total dose history sequence was set to 100, 1,000, 10,000 and 100,000 histories, and for the spot dose 10 and 100 histories. Figure 6.8 shows the results from varying start values of number of histories w.r.t. the total number of histories spent. The optimization was run for around $1.5 \cdot 10^8$ histories.

The general result is that, for the given amount of histories spent, it is better to start at a high number of histories for total dose and a low number of histories for the spot dose. This agrees with the results from Section 6.3, which states that it is better for the convergence to increase the number of histories to compute the total dose. It should, however, be noted that for both start values on number of histories for spot dose the convergence is best when starting with 10,000 histories to compute total dose (yellow) in the beginning. Later during the optimization the case of starting with 100,000 histories for total dose (purple) reaches lower error. This might be due to that in the beginning, based on the quality of the points, simulating 10,000 histories saturates the step qualities and because the convergence is w.r.t. histories it is more effective to not spend more histories than that. Then, as the quality of the points increase, it is beneficial to have an increase in step quality, thus starting with 100,000 histories for total dose has better convergence.

The fact that the optimizations starting with 10 histories for computing spot dose have better convergence than when starting with 100 histories, see Figure 6.8c and 6.8d, is somewhat overshadowed by the fact that this causes the optimization to go through more iterations. This of course also holds when starting value for total number of histories decrease. Simulating histories is a time consuming task, but the optimization time also increase when the number of iterations increase, for a given number of total histories.

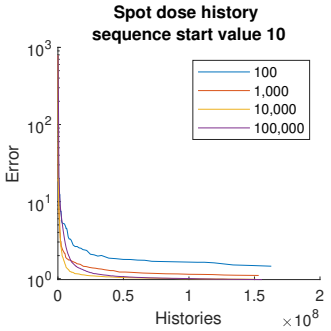
6.6 Uncertainty propagation

The two ways to compute the error uncertainty as described in the end of Section 3.2 was compared. The results are presented in Figure 6.9, illustrating the uncertainty both from standard deviation of error from 10 batches, as well as from propagating the dose uncertainty in each voxel from the same 10 batches. It can be seen that the error uncertainty based on dose uncertainty in general lies above. It can also be seen that while the uncertainty based on error batches oscillate much, the uncertainty based on dose uncertainty is much smoother, making distinct decreases corresponding to when the number of histories for computing total dose is increased.

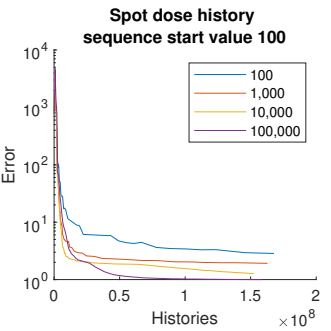
The reason for the initial great oscillation for both cases has not been found, but it might have to do with the characteristics of the initial point, which is all ones.

6.7 Smooth dose derivative field

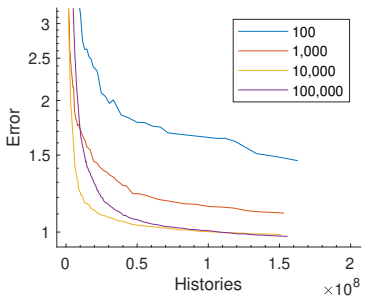
As seen in Section 6.1 the MC dose converges to the analytical dose as more histories are used to compute it, i.e., when the noise is reduced. As a MC dose calculation is closer to the analytical, a better convergence should be yielded. To investigate how reducing the noise in the data from a MC dose calculation affects the convergence,



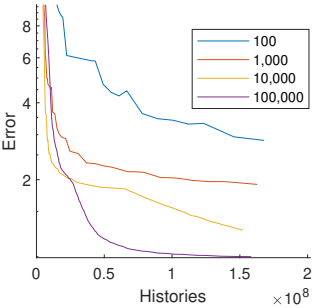
(a) Spot dose start value 10 histories. Total dose starts at value according to legend.



(b) Spot dose start value 100 histories.
Total dose starts at value according to legend.



(c) Same as (a) but zoomed in.



(d) Same as (b) but zoomed in.

Figure 6.8 The figures illustrates the convergence for different start values of the history sequence. The start values for histories to compute spot dose is 10 for (a) and (c) respectively 100 for (b) and (d), each with start values for calculating the total dose according to legend. The lower figures are zoomed in to the later part of the convergence, corresponding to the upper figures, for easier view. The plots are w.r.t. total number of histories simulated during the optimization.

the *dose derivative field* tensor P [1] was smoothed by convoluting it with an unbiased 3x3x3 normal distribution kernel with varying standard deviation.

As can be seen in the results in Figure 6.10, the convergence is best for the non convoluted and degrades for increasing standard deviation for the convolution kernel. This might be due to the convolution causes the parameters to be more dependent which causes the conditional number to degrade. At the same time, causing the step to be sub-optimal due to the dose derivative field is smeared out and reso-

[1] P is normally handled as a vector but it can instead be seen as a matrix such that its elements are arranged as the voxels they represent lie in the voxel grid.

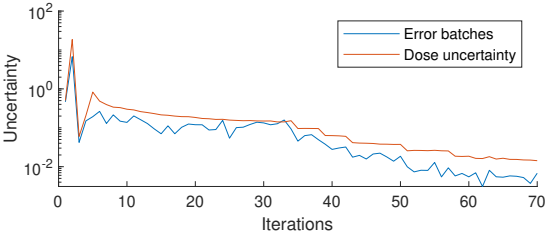


Figure 6.9 The error uncertainty computed by standard deviation of the error from several batches (blue) and the error uncertainty by propagating the dose uncertainty in each voxel from several batches (red) by means of (3.20). The computations were done during optimization based on MC dose engine for 70 iterations.

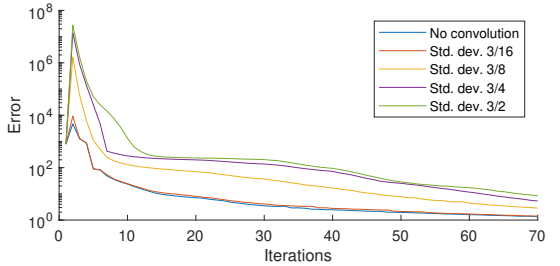


Figure 6.10 Convergence based on the MC dose engine where the dose derivative field is convoluted with a non biased normal distribution kernel with varying standard deviation. Notice how the non-convoluted has the best convergence and decreasing for larger standard deviation.

lution is lost.

7

Discussion

7.1 General discussion

For the specific experiment case and with the simplified dose engine used, an adequate treatment plan can be reached by optimization based on Monte Carlo (MC). The convergence is degraded compared to the optimization based on the analytical dose engine.

Of course, the analytical convergence curve comes from optimizing the parameters with the same algorithm as the MC based is optimized with. In reality, when optimizing on pre-computed spot doses a more advanced optimization algorithm is used, e.g., including a line search. This is not incorporated due to be able to compare the MC based to the analytical better.

7.2 Importance of total dose

The convergence seems to be most positively impacted by decreasing the uncertainty in the total dose than in the spot dose. This can be seen in Figure 6.5 where the quality of the gradient only depend on the number of histories used to compute either the total dose or the spot dose. It can be seen that it converges much more quickly towards the completely analytical step when increasing the number of histories to compute the total dose.

It can also be seen in Figure 6.7b, which compares optimizations that increase both history sequences, only total dose history sequence and neither. The case when only total dose history sequence is increased has significantly better convergence than for when both sequences are increased.

One way to benefit from this is to instead of computing the total dose based on the current point at each iteration, to compute the difference in dose and accumulate it to the dose from the previous iteration. I.e., computing the dose of the *step* and adding this to the previous dose according to some update formula. The update formula must make sure that the new dose, i.e., the old dose together with the dose difference, corresponds to the new point. The benefit from this is that the number of

histories used to compute the total dose increases, without the need to increase the number of histories to compute total dose from one iteration to another.

The method of accumulating the total dose has similarities with the beamlet-free algorithm by Pross et al. (2023) described in Section 4.3. However, instead of letting the spot weights be represented with a probability distribution, the point is updated exact and the dose corresponding to this point is updated with means of the step to get to the point and the previous dose.

The problem with the discussed method is that it only works if the dose is linear to the optimization parameters. This is not the case when optimizing treatment plans with e.g. photons.

7.3 Computing the Hessian diagonal

It is clear from Figure 6.6 that the approximation of the Hessian diagonal described in Section 5.2 is bad, since neither the direction nor the length converges to the analytical such. The fact that the corresponding do converge when using the correct Hessian diagonal, as seen in Figure 6.5, shows there is much to benefit from using the correct Hessian diagonal, instead of the approximation.

As previously discussed, all spot doses cannot be computed and stored, due to it takes too much memory when optimizing treatment plans for, e.g., carbon, which uses more spots than proton does. However, the procedure for computing gradient and Hessian diagonal can instead be to compute say 200-300 spots simultaneously; when a spot dose is computed it is used to calculate the two instances according to (3.14) and (3.18), whereas the spot dose is discarded and next spot dose is calculated. This way the correct Hessian diagonal can be computed without storing spot doses. This of course makes the optimization more time consuming as the full potential of multi-threading on GPUs cannot be reached.

Another way to solve the problem is to compute the gradient and Hessian diagonal separately. The gradient can be computed in the regular way, there is no need to compute the whole spot dose since the component of the gradient is linear to the dose. When computing the Hessian diagonal, it is done as described above but with less accuracy; either by simulating significantly less histories and/or decreasing the resolution in the voxel grid. This is motivated by the results in Figure 6.5 which show that the Hessian diagonal has much higher accuracy than the gradient for a given number of histories to calculate it. Additionally, for convex problems it can often be assumed that the Hessian varies less than the gradient. Thus it is probable that a less exact Hessian diagonal can still contribute to a good step. Due to the use of lower accuracy, using this method should not demand too much overhead when computing gradient and Hessian diagonal spot doses separately.

7.4 Uncertainty of error

As was seen in Figure 6.7c, the error from the MC dose can not be trusted to accurately give a good representation of the actual point if the uncertainty for the total dose is high. As a clinical MC dose engine does not have a matching analytical dose engine that can give the correct error for a given point, the above can cause problem, e.g., when investigating different optimization methods and applying IP, as explained in Section 3.1. IP will cause bias in convergence for methods that have decreased their uncertainty in total dose, even though they do not necessarily have reached a better treatment plan.

A possible solution to the above problem is to make an estimation of the current ideal MC error based on the MC error and the error uncertainty. For this purpose, the error uncertainty based on propagating the uncertainty in dose as described in Section 3.2 is probably good, due to it being fairly constant while calculating total dose with a constant number of histories, as seen in Figure 6.9.

Another solution could be to at a given time perform a total dose computation with a set number of iterations for all the methods, in order to be able to perform a bias-free comparison between them.

7.5 Smooth dose derivative field

The idea of smoothing the dose derivative field by convolution with a normal kernel in order to reduce the effect of noise when computing the total dose was not successful. This, however, does not imply that smoothing the dose derivative field is not a good idea.

7.6 Stopping criterion

A common way to choose a stopping criterion, i.e., a criterion for when to stop the optimization and consider the point found to be the solution, is when the improvement in error between iterations is within a certain tolerance. As the noise causes the error to fluctuate much, this does not work.

What could be done instead is to do a linear regression over the last say 10 points and evaluate the slope, if it is within tolerance the best of these points (or the last) is chosen as the solution.

8

Conclusions

8.1 Conclusions

The presented results can with certainty only be considered to hold for the simplified experiment they are based on. However, despite the simplification, the hope is for the results to serve as a guide for treatment planning based on Monte Carlo (MC) simulations in a clinical setting. The main conclusions are that...

- ...an adequate treatment plan can be achieved despite the noise introduced by MC. The optimization converges even though the uncertainty is high,
- ...for better convergence, it is of more importance to have a low uncertainty in the total dose than in the spot dose,
- ...the noise gives bias for error that was computed with low uncertainty. This must be accounted for when comparing solutions, and comparing solutions with the same uncertainty is encouraged,
- ...the fast MC dose engines need to be changed so that the correct Hessian diagonal can be computed. This leads to the last point, namely that,
- ...it might be better to compute the gradient and Hessian diagonal separately for time efficiency. This due to the Hessian diagonal's less need for low uncertainty in combination with it being more computationally expensive to compute.

8.2 Outlook

This thesis has investigated the effects of optimizing proton treatment plans based on MC and the results should serve as a guide towards performing the same in a clinical setting. However, in the background the method was motivated by being able to perform treatment planning with MC even when the number of spots grew too large to be able to store their pre-computed dose. With that said, the results

should serve as a guide even for treatment planning with other ions such as carbon, which uses about 4 times as many spots.

Stretching it even further, the problem was to optimize spot weights which was linear to the total dose. The results could in some sense also be translatable to optimization where the dose is not linear to the optimization parameters, where the main case of this is *intensity-modulated radiotherapy treatment* where photons are used as radiation and the shape and intensity of the beam segments are optimized.

Bibliography

Baskar, R., J. Dai, N. Wenlong, R. Yeo, and K.-W. Yeoh (2014). "Biological response of cancer cells to radiation treatment". *Frontiers in Molecular Biosciences* **1**, p. 24. ISSN: 2296-889X. DOI: 10.3389/fmolb.2014.00024. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4429645/ (visited on 2023-05-22).

Bokrantz, R. (2013). "Multicriteria optimization for managing tradeoffs in radiation therapy treatment planning". en.

Bordes, A., L. Bottou, and P. Gallinari (2009). "SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent". en.

Bottou, L. (2010). "Large-Scale Machine Learning with Stochastic Gradient Descent". en. In: Lechevallier, Y. et al. (Eds.). *Proceedings of COMPSTAT'2010*. Physica-Verlag HD, Heidelberg, pp. 177–186. ISBN: 978-3-7908-2604-3. DOI: 10.1007/978-3-7908-2604-3_16.

Burton, M., A. Mozumber, and A. Canfield Upton (2023). *Radiation | Definition, Types, Effects, & Facts | Britannica*. en. URL: https://www.britannica.com/science/radiation (visited on 2023-03-08).

Degerfält, J., I.-M. Moegelin, and L. Sharp (2008). *Strålbehandling*. 2nd. Studentlitteratur, Lund. ISBN: 978-91-44-04523-8.

Elcim, Y., B. Dirican, and O. Yavas (2018). "Dosimetric comparison of pencil beam and Monte Carlo algorithms in conformal lung radiotherapy". *Journal of Applied Clinical Medical Physics* **19**:5, pp. 616–624. ISSN: 1526-9914. DOI: 10.1002/acm2.12426. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6123106/ (visited on 2023-05-05).

Fiorini, F., N. Schreuder, and F. Van den Heuvel (2018). "Technical Note: Defining cyclotron-based clinical scanning proton machines in a FLUKA Monte Carlo system". en. *Medical Physics* **45**:2. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mp.12701, pp. 963–970. ISSN: 2473-4209. DOI: 10.1002/mp.12701. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mp.12701 (visited on 2023-04-19).

Liu, H. and J. Y. Chang (2011). "Proton therapy in clinical practice". *Chinese Journal of Cancer* **30**:5, pp. 315–326. ISSN: 1000-467X. DOI: 10.5732/cjc.010.10529. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4013396/ (visited on 2023-05-12).

Maughan, R. L. (2022). *The Physics of Proton Therapy | OncoLink*. URL: https://www.oncolink.org/healthcare-professionals/oncolink-university/proton-therapy-professional-education/oncolink-proton-education-modules/the-physics-of-proton-therapy (visited on 2023-04-26).

Nesterov, Y. (1983). "A method for unconstrained convex minimization problem with the rate of convergence o(1/k^2)". *Doklady AN USSR* **269**, pp. 543–547. URL: https://cir.nii.ac.jp/crid/1570572699326076416 (visited on 2023-05-31).

Nesterov, Y. and A. Nemirovskii (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9781611970791. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611970791.

Newhauser, W. D. and R. Zhang (2015). "The physics of proton therapy". *Physics in medicine and biology* **60**:8, R155–R209. ISSN: 0031-9155. DOI: 10.1088/0031-9155/60/8/R155. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4407514/ (visited on 2023-05-22).

Pfeiffer, G. T. and Y. Sato (2018). *On stochastic optimization methods for Monte Carlo least-squares problems*. arXiv:1804.10079 [math]. DOI: 10.48550/arXiv.1804.10079. URL: http://arxiv.org/abs/1804.10079 (visited on 2023-05-10).

Polyak, B. T. (1964). "Some methods of speeding up the convergence of iteration methods". en. *USSR Computational Mathematics and Mathematical Physics* **4**:5, pp. 1–17. ISSN: 0041-5553. DOI: 10.1016/0041-5553(64)90137-5. URL: https://www.sciencedirect.com/science/article/pii/0041555364901375 (visited on 2023-05-31).

Pross, D., S. Wuyckens, S. Deffet, E. Sterpin, J. A. Lee, and K. Souris (2023). *Beamlet-free optimization for Monte Carlo based treatment planning in proton therapy*. arXiv:2304.08105 [physics]. URL: http://arxiv.org/abs/2304.08105 (visited on 2023-04-18).

Qian, N. (1999). "On the momentum term in gradient descent learning algorithms". en. *Neural Networks* **12**:1, pp. 145–151. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00116-6. URL: https://www.sciencedirect.com/science/article/pii/S0893608098001166 (visited on 2023-04-24).

Ruder, S. (2017). *An overview of gradient descent optimization algorithms*. en. arXiv:1609.04747 [cs]. URL: http://arxiv.org/abs/1609.04747 (visited on 2023-05-27).

Seco, J. and F. Verhaegen (2013). *Monte Carlo Techniques in Radiation Therapy*. en. 1st. CRC Press. URL: https://www.routledge.com/Monte-Carlo-Techniques-in-Radiation-Therapy/author/p/book/9781138199903 (visited on 2023-04-19).

Siggel, M. (2012). "Concepts for the efficient Monte Carlo-based treatment plan optimization in radiotherapy". en.

Sundström, J. (2021). "Scenario Dose Calculation for Robust Optimization in Proton Therapy Treatment Planning". eng. *Master's Theses in Mathematical Sciences*. ISSN: 1404-6342. URL: http://lup.lub.lu.se/student-papers/record/9059444 (visited on 2023-05-31).

Teoh, S., F. Fiorini, B. George, K. A. Vallis, and F. Van den Heuvel (2020). "Is an analytical dose engine sufficient for intensity modulated proton therapy in lung cancer?" *The British Journal of Radiology* **93**:1107, p. 20190583. ISSN: 0007-1285. DOI: 10.1259/bjr.20190583. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7066954/ (visited on 2023-04-19).

Yao, Z., A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. W. Mahoney (2021). *ADAHESSIAN: An Adaptive Second Order Optimizer for Machine Learning*. arXiv:2006.00719 [cs, math, stat]. URL: http://arxiv.org/abs/2006.00719 (visited on 2023-03-20).

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2023

Document Number
TFRT-6209

Author(s)
Ludvig Håkansson

Supervisor
Eric Landström, Raysearch Laboratories, Sweden
Pontus Giselsson, Dept. of Automatic Control, Lund University, Sweden
Emma Tegling, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Optimization of Radiotherapy Treatment Plans Based on Monte Carlo Dose Computations

Abstract

Treatment planning plays a vital role in providing good treatment to cancer patients. In order to reach an adequate treatment plan, the algorithm used for simulating the dose in the patient must model the reality well. The most accurate algorithm for this is the Monte Carlo method, which in the context of radiotherapy treatment most often is used for pre-computing spot doses and optimizing the intensity for each spot. Due to storage limitations, it would be preferred to do the mentioned computations simultaneously as the optimization. This, however, makes the optimization problem non-convex due to the statistical noise introduced by Monte Carlo.

This thesis investigates the feasibility of using first-order optimization methods for treatment planning based on Monte Carlo simulations and addresses the challenges posed by the noise. A simplified proton Monte Carlo dose engine was implemented together with a matching analytical such, in order to assess the effect of the noise during optimization.

The results demonstrate that despite the noise, an adequate treatment plan can be achieved. Convergence is found to be dependent on how simulations are used within an iteration. Techniques such as accumulating total dose and computing the gradient and Hessian separately show promise for improving convergence and time efficiency, respectively. The impact of noise on error computation and the need for appropriate comparisons are highlighted.

This work provides insights for advancing Monte Carlo treatment planning and its integration into clinical settings. The findings are applicable not only to proton treatment plans but also to other ions and perhaps even to photons.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-47

Recipient's notes

Security classification

http://www.control.lth.se/publications/