

Thesis for the acquisition of the academic degree
Master of Science

**Development of magnetic field
influenced pencil beam spot selection
for proton therapy treatment planning**

Yosef Emshani
born in Castrop-Rauxel

2024

Lühr Group
Department of Physics
TU Dortmund University

Primary reviewer: Prof. Dr. Armin Lühr
Secondary reviewer: Prof. Dr. Kevin Kröninger
Submission date: December 2, 2024

Abstract

This thesis presents the development and validation of an analytical algorithm for calculating magnetic field influenced proton pencil beam spots, forming the basis for integration into a candidate treatment planning system (TPS). Proton therapy, renowned for its precision and reduced healthy tissue damage, benefits from image guidance to further enhance treatment efficacy. However, MR-integrated proton therapy (MRiPT) presents several challenges, one of which arises from the deflection of charged proton beams by the magnetic field of the magnetic resonance imaging (MRI) scanner. The developed proton transfer algorithm addresses this challenge by calculating proton trajectories through various materials under the influence of a magnetic field, validated against Monte Carlo (MC) simulations. The algorithm was tested across cases of increasing complexity, from vacuum to patient-specific phantoms, with differences between calculated and reference results analyzed. Notably, minimal differences were observed for water phantoms, while bone phantoms exhibited the largest discrepancies. The inclusion of optimization via a gradient descent algorithm allowed for adjustment of pencil beam spots to account for deflections. Median stopping position distances increased from 0.49 mm without a magnetic field to 1.06 mm under a $B_z = 1.5$ T field, highlighting the need for further refinement. Future work includes enhancing stopping power accuracy, extending to three-dimensional (3D) volumetric analyses, and exploring computationally efficient optimization techniques.

Kurzfassung

In dieser Arbeit wird die Entwicklung und Validierung eines analytischen Algorithmus zur Berechnung magnetfeldbeeinflusster Protonen-Pencil-Beam-Spots vorgestellt, der die Grundlage für die Integration in einem TPS bildet. Die Protonentherapie, die für ihre Präzision und geringere Schädigung des gesunden Gewebes bekannt ist, profitiert von der Bildführung, um die Wirksamkeit der Behandlung weiter zu verbessern. Die MRiPT stellt einige Herausforderungen dar, wovon eine die Ablenkung der geladenen Protonenstrahlen durch das Magnetfeld des Scanners ist. Der entwickelte Algorithmus für den Protonentransfer geht auf diese Herausforderung ein, indem er die Trajektorien von Protonen durch verschiedene Materialien unter dem Einfluss eines Magnetfeldes berechnet und anhand von MC Simulationen validiert. Der Algorithmus wurde in Fällen mit zunehmender Komplexität getestet, vom Vakuum bis hin zu patientenspezifischen Phantomen, wobei die Unterschiede zwischen den berechneten und den Referenzergebnissen analysiert wurden. Insbesondere wurden minimale Unterschiede bei Wasserphantomen beobachtet, während Knochenphantome die

größten Diskrepanzen aufwiesen. Die Optimierung mittels eines Gradientenabstiegsverfahren ermöglichte die Anpassung der Pencil Beam Spots, um Ablenkungen zu berücksichtigen. Die mittleren Abstände stiegen von 0.49 mm ohne Magnetfeld auf 1.06 mm unter einem $B_z = 1.5$ T Feld, was die Notwendigkeit einer weiteren Verfeinerung unterstreicht. Zukünftige Arbeiten umfassen die Verbesserung der Genauigkeit des Bremsvermögens, die Ausweitung auf 3D volumetrische Analysen und die Erforschung rechnerisch effizienter Optimierungstechniken.

Contents

1	Introduction	1
2	Theoretical background	3
2.1	Interaction of charged particles with matter	3
2.2	Interaction of charged particles with magnetic fields	5
2.3	Gradient descent optimization method	6
2.4	Monte Carlo simulations	7
2.5	Radiation therapy	8
2.6	matRad treatment planning system	10
2.7	Hounsfield look-up table	12
3	Materials and methods	14
3.1	Patient dataset	14
3.2	Treatment plan setup	15
3.3	Analytical calculation of proton stopping position	16
3.3.1	Developed proton transfer algorithm	16
3.3.2	matRad	18
3.4	Monte Carlo based calculation of proton stopping position	20
3.4.1	Simulation setup	20
3.4.2	Estimation of proton stopping position for $B = 0$ T	21
3.4.3	Estimation of proton stopping position for $B > 0$ T	21
3.5	Optimization for magnetic field introduction	23
4	Results	25
4.1	Verification of proton stopping position	25
4.1.1	Vacuum state	25
4.1.2	Water phantom	27
4.1.3	Bone phantom	32
4.1.4	Prostate patient	35
4.2	Recalculation of spot selection for a matRad treatment field	39
4.2.1	Proton treatment field	39
4.2.2	Verifying treatment field beam positions	41
4.2.3	Calculation of deflected spots	43
4.2.4	Optimization of deflected spots	45

5	Discussion	47
5.1	Verification of proton stopping position	47
5.1.1	Vacuum state	47
5.1.2	Water phantom	48
5.1.3	Bone phantom	50
5.1.4	Prostate patient	51
5.2	Recalculation of spot selection for a matRad treatment field	53
5.2.1	Proton treatment field	53
5.2.2	Verifying treatment field beam positions	54
5.2.3	Calculation of deflected spots	54
5.2.4	Optimization of deflected spots	55
6	Conclusion and outlook	56
A	Appendix	58
A.1	Developed proton transfer algorithm	58
A.1.1	Proton transfer algorithm	58
A.1.2	Initialization of proton transfer algorithm	66
A.2	TOPAS	68
A.2.1	Simulation setup	68
A.2.2	Estimation of proton stopping positions for $B > 0$ T	69
A.3	matRad	76
A.3.1	Treatment plan setup	76
A.3.2	Estimation of matRad proton stopping positions	79
A.3.3	Hounsfield look-up table	81
A.4	Optimization	81
A.4.1	Computational gradient descent implementation	81
	List of figures	86
	List of tables	90

List of abbreviations

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
CSDA	continuous slowing down approximation
CT	computed tomography
CTV	clinical target volume
DICOM	Digital Imaging and Communications in Medicine
DVH	dose volume histogram
GUI	graphical user interface
HLUT	Hounsfield look-up table
ICRU	International Commission on Radiation Units and Measurements
LINAC	linear accelerator
MATA	material assignment
MC	Monte Carlo
MRI	magnetic resonance imaging
MRiPT	MR-integrated proton therapy
OAR	organ at risk
RBE	relative biological effectiveness
rSPR	relative stopping power ratio
TOPAS	Tool for Particle Simulation
TPS	treatment planning system

List of abbreviations

1 Introduction

Cancer therapy currently comprises three primary modalities: surgery, chemotherapy, and radiotherapy [zeman_basics_2020]. Among these, radiotherapy is widely applied, either as a standalone treatment or in combination with other therapeutic approaches [p_world_nodate]. It is a well-established and highly effective method for treating cancer. Conventional radiotherapy predominantly uses photons to deliver the prescribed dose to tumor sites. However, proton therapy offers notable advantages, particularly in its ability to spare healthy tissue, owing to the unique dose deposition characteristics of protons [suit_clinical_1977]. Consequently, the number of treatments and centers utilizing particle therapy, particularly proton therapy, has significantly increased in recent years [han_current_2019].

The high precision of energy deposition in proton therapy, while beneficial, also makes it particularly sensitive to morphological changes in patient anatomy. Consequently, image-guided proton therapy has the potential to further enhance the sparing of healthy tissue. MRI is a promising candidate for image guidance due to its excellent soft-tissue contrast and the absence of ionizing radiation [schellhammer_integrating_2018]. This has led to growing interest in exploring MRiPT. Although MR-guided photon therapy already has commercial systems in clinical use [duetschler_fast_2023], MRiPT remains a novel concept under investigation. Unlike photons, the primary proton beam is deflected by the magnetic field of an MRI system due to its charged nature. This deflection must be considered when determining proton stopping positions within the specified target in the patient.

This thesis establishes the foundation for integrating magnetic field influenced pencil beam spot selection within a candidate TPS. To achieve this, an algorithm is developed to calculate proton trajectories through various materials under the influence of a magnetic field. The algorithm is validated using a candidate MC simulation platform. Proton pencil beam properties, such as initial energy and starting position, are exported from an example treatment plan and imported into the developed algorithm. The resulting pencil beam spots are compared to those initially calculated in the TPS. An optimization algorithm is then applied to account

1 Introduction

for magnetic field induced deflections, ensuring that the adjusted pencil beam spots closely align with those calculated in the absence of a magnetic field.

2 Theoretical background

This chapter outlines the theoretical foundations for developing a magnetic field influenced pencil beam spot algorithm. Key concepts include charged particle interactions with matter (see section 2.1) and magnetic fields (see section 2.2), optimization techniques for beam positioning (see section 2.3), and Monte Carlo simulations (see section 2.4). Additionally, the chapter reviews the radiotherapy workflow (see section 2.5) and introduces the treatment planning system used in this thesis (see section 2.6). Finally, the concept of a Hounsfield look-up table (HLUT) is explained (see section 2.7).

2.1 Interaction of charged particles with matter

In this thesis, it is assumed that only charged particles are considered. Focusing on proton therapy, protons serve as the primary charged particles under analysis. Their charge generates an electric field that increases the likelihood of interactions compared to neutral particles such as neutrons or photons. The use of charged particles leads to the emission of secondary electrons, which contributes to tissue damage. The energy loss of these particles encompasses nuclear, electronic, and radiative components [krieger_wechselwirkungen_2023].

The total stopping power S_{tot} is described with

$$S_{\text{tot}} = \left[\left(\frac{dE}{dx} \right)_{\text{nucl}} + \left(\frac{dE}{dx} \right)_{\text{col}} + \left(\frac{dE}{dx} \right)_{\text{rad}} \right]. \quad (2.1)$$

At lower energy levels, nuclear energy loss becomes dominant [hull_ion_2011]. However, this interaction is generally negligible in clinical contexts, as protons typically have energies in the range of $200 \text{ MeV} \leq E \leq 250 \text{ MeV}$ [sengbusch_maximum_2009].

Radiative energy loss is also negligible, as it is minimal compared to electronic energy loss [hull_ion_2011].

In the clinically relevant proton energy range, the electronic energy loss is significant [**krieger_wechselwirkungen_2023**, **hull_ion_2011**]. This total energy loss results from inelastic interactions between the charged particle and the shell electrons, as well as interactions with the nucleus. Clinically, the main focus is on inelastic scattering leading to the emission of secondary electrons [**krieger_wechselwirkungen_2023**].

With these assumptions, the total stopping power (equation 2.1) can be approximated with

$$S_{\text{tot}} \approx \left(\frac{dE}{dx} \right)_{\text{col}}. \quad (2.2)$$

The approximate total stopping power, S_{tot} , can be calculated using the Bethe Bloch formula [**bethe_zur_1930**, **bloch_bremsvermogen_1933**, **bloch_zur_1933**]. A simplified form of this formula is used, tailored specifically for applications in proton therapy [**newhauser_physics_2015**]

$$-\left(\frac{dE}{dx} \right)_{\text{col}} \approx \Gamma \rho \frac{z_T}{A_T} \frac{Z^2}{\beta_{\text{rel}}^2} \left[\frac{1}{2} \ln \left(\frac{2m_e c^2 \beta_{\text{rel}}^2 W_{\text{max}}}{I^2} \right) - \beta_{\text{rel}}^2 - \frac{\delta}{2} - \frac{C}{z} \right], \quad (2.3)$$

where $\Gamma = 2\pi N_A r_e^2 m_e c^2$ and $W_{\text{max}} = \frac{2m_e c^2 \beta_{\text{rel}}^2 \gamma^2}{1 + \frac{2\gamma m_e}{M_0} + \left(\frac{m_e}{M_0} \right)^2}$. Here Γ is a constant that includes the Avogadro constant N_A , the electron radius r_e , the electron mass m_e , and the speed of light c . The term W_{max} represents the maximum energy transfer to an electron. The properties of the target material are defined by the atomic number z_T , the mass number A_T , the density ρ , and the mean excitation potential I . This mean excitation potential I is discussed further in the following chapters (see subsection 3.3.1). The projectile properties include the atomic number Z , the mass M_0 , the Lorentz factor γ , and the relativistic velocity $\beta_{\text{rel}} = \frac{v}{c}$, where v is the projectile velocity. The density correction term δ accounts for reduced energy loss at higher energies due to changes in the electric field and its interaction with shell electrons [**newhauser_physics_2015**]. Finally the shell correction term C becomes significant at lower energies [**newhauser_physics_2015**].

2.2 Interaction of charged particles with magnetic fields

Given a static, uniform electric field \vec{E} and magnetic field \vec{B} , the trajectory of a charged particle in a vacuum is influenced by the Lorentz force \vec{F}_L [hoffmann_proton_2015]. In the context of relativistic dynamics, the Lorentz force law for a charge q can be expressed as

$$\vec{F}_L = \frac{d\vec{p}}{dt} = q(\vec{E} + \vec{v} \times \vec{B}) , \quad (2.4)$$

where the relativistic momentum is described by

$$\vec{p} = m\vec{v} = \frac{m_0\vec{v}}{\sqrt{1 - \frac{v^2}{c^2}}} , \quad (2.5)$$

with the speed of light c . For relativistic dynamics, the mass m is expanded by the Lorentz factor

$$\gamma = \left(1 - \frac{v^2}{c^2}\right)^{-\frac{1}{2}} , \quad (2.6)$$

relative to the rest mass m_0 . Assuming $\vec{E} = 0$ leads to a constant velocity v . Since m and γ are constants as well, the motion of a particle in a static uniform magnetic field can be treated as if it were non-relativistic

$$\vec{F}_L = m\frac{d\vec{v}}{dt} = q(\vec{v} \times \vec{B}) , \quad (2.7)$$

except that the particle's mass is greater than its rest mass by a factor of γ .

To derive the radius r of the circular component of a charged particle's helical trajectory in a magnetic field, the balance between the centripetal force \vec{F}_C and the Lorentz force \vec{F}_L is considered. Setting $\vec{F}_C = \vec{F}_L$, we have

$$\frac{mv^2}{r} = qvB , \quad (2.8)$$

where v is the particles' velocity relative to the magnetic field B . Solving for r yields

$$r = \frac{mv}{qB} . \quad (2.9)$$

Thus, the particle follows a helical path with a radius r .

The trajectory of a positively charged particle is generally demonstrated in figure 2.1.

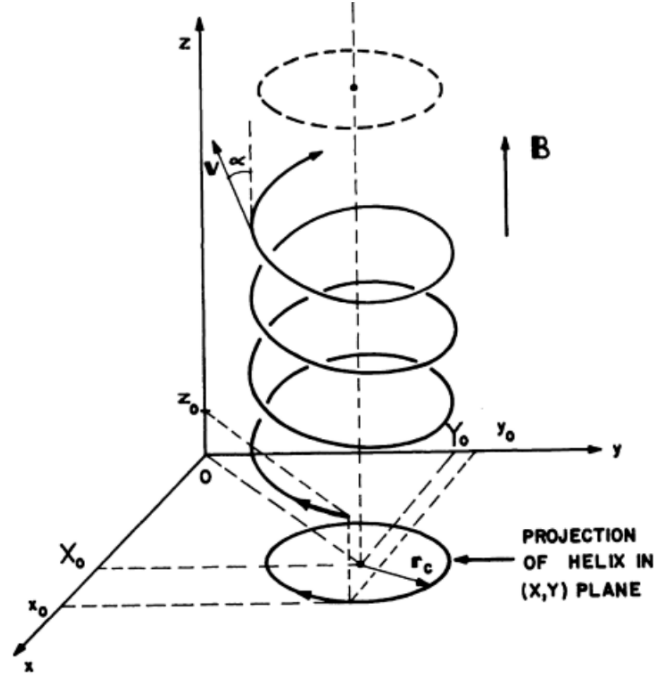


Figure 2.1: Helical trajectory of a positively charged particle in a magnetic field with marked radius [hoffmann_proton_2015].

2.3 Gradient descent optimization method

The gradient descent algorithm is an iterative optimization method used to minimize a differentiable function $f(\vec{x})$ [boyd_convex_2004]. Starting from an initial position \vec{x}_0 , the algorithm updates \vec{x} in the direction opposite to the gradient $\nabla f(\vec{x})$, which points toward the steepest ascent. By moving against the gradient, we approach a local minimum. The update rule is given by

$$\vec{x}_{i+1} = \vec{x}_i - \eta \nabla f(\vec{x}_i) , \quad (2.10)$$

where η is the learning rate, a parameter that determines the step size. This iterative process continues until $\nabla f(\vec{x}) \approx 0$, indicating that a local minimum is reached.

2.4 Monte Carlo simulations

MC simulations are a computational technique used to model complex systems by simulating the random processes within them. This method leverages repeated random sampling to approximate solutions to problems that are analytically intractable, making it valuable for applications requiring high precision. In radiotherapy, MC simulations are particularly beneficial as they enable detailed modeling of particle interactions within human tissue, accounting for the stochastic nature of radiation transport and dose deposition [andreo_monte_2018].

Tool for Particle Simulation (TOPAS) is a MC simulation platform designed for particle transport simulations, especially in medical physics. Built on the Geant4 toolkit, TOPAS enables highly detailed simulations of particle interactions with various materials, including tissue-equivalent substances used in radiotherapy and dosimetry [perl_topas_2012] [faddegon_topas_2020]. Configured through a set of editable parameter files, TOPAS supports a wide range of particle types. In this thesis, TOPAS serves as the ground truth for estimating proton stopping positions.

A TOPAS simulation is set up by a set of parameter files. These files can be edited using a plain text editor. The syntax of a parameter is defined as:

```
Parameter_Type : Parameter_Name = Parameter_Value # Optional comment
```

The parameter type indicates the type of value being used, such as an integer "i", a string "s", or a decimal "d" with an associated unit. Parameter names follow an object-oriented, hierarchical structure and always begin with a prefix assigned by TOPAS that indicates the parameter type. For example, "Ge" stands for geometric parameters, while "So" refers to particle source-related parameters. The prefix is followed by the object name for which the parameter is defined. Finally, the value of the parameter is specified, which can be either a string or a number with a unit, depending on its type. This structure provides clarity in the parameter file, making it immediately clear what type of parameter is being specified.

2.5 Radiation therapy

Radiotherapy is a medical treatment that uses high-energy radiation to destroy cancer cells, aiming to reduce or eliminate tumors while minimizing damage to surrounding healthy tissue. The radiotherapy workflow is visualized in figure 2.2. The first step, **Consultation**, includes an initial patient evaluation to determine suitability for radiotherapy. During **Simulation**, computed tomography (CT) imaging is performed with the patient in the treatment position to guide subsequent planning. In the **Contouring** stage, target volumes and critical organs are delineated to define areas requiring treatment and those to be protected. The **Planning** phase involves calculating an optimal dose distribution to maximize tumor control while sparing healthy tissue. This thesis focuses on this step of radiation therapy. A candidate TPS and further introductions into treatment planning will be discussed in the following section 2.6. **Delivery** is the administration of radiation using a linear accelerator (LINAC) according to the treatment plan. Finally, **Follow-up** includes monitoring and assessing patient response to treatment to evaluate its effectiveness.

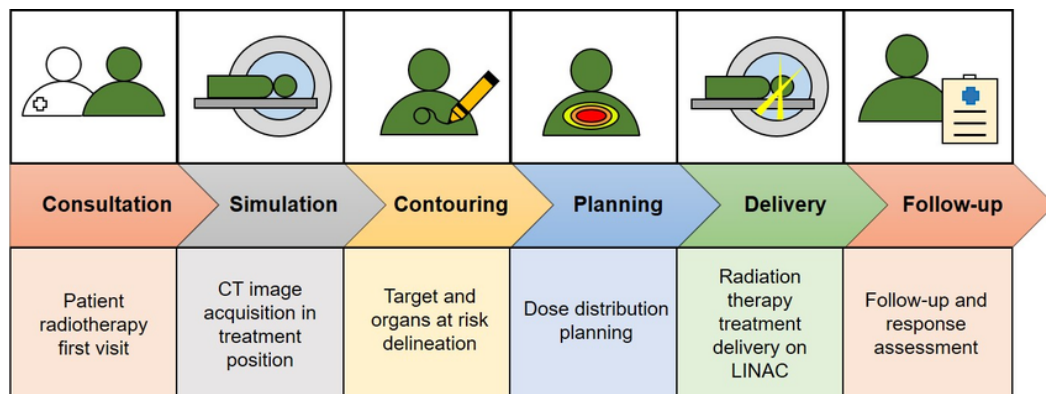


Figure 2.2: Illustration of radiotherapy workflow consisting of consultation, simulation, contouring, planning, delivery and follow-up stages [marvaso_virtual_2022].

When analyzing treatment planning in proton therapy, the proton range is of critical importance, as it determines the depth at which the maximum dose is delivered within tissue. Unlike conventional X-ray radiotherapy, protons deposit most of their energy at the end of their path, forming a characteristic Bragg peak in the depth-dose profile.

The proton range can be approximated using the continuous slowing down approximation (CSDA), which calculates the average path length a proton travels before coming to rest. Clinically, the proton range is often defined by the 80 % distal fall-off point of the Bragg peak, R_{80} , where the dose drops sharply. This is illustrated in figure 2.3. It has been shown that

$$\text{CSDA} \approx R_{80} , \quad (2.11)$$

making it a useful approximation for clinical applications [paganetti_proton_2018].

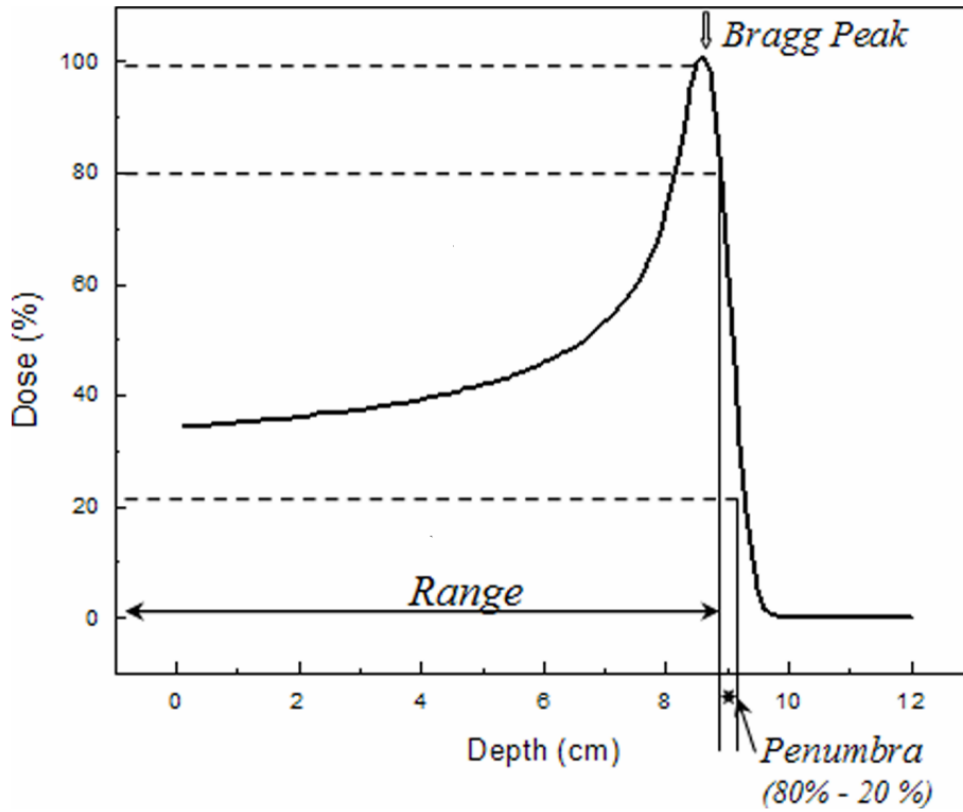


Figure 2.3: Visualization of proton percentage depth dose curve example with marked characteristic Bragg peak as well as highlighted proton range, R_{80} , at 80 % distal fall-off of the Bragg peak [park_variation_2011].

Accurate determination of the proton range is essential in proton therapy to ensure precise dose delivery to the tumor while minimizing exposure to surrounding healthy tissue.

2.6 matRad treatment planning system

The TPS analyzed in this thesis is the open source software matRad. This TPS supports intensity modulated photon, proton, and carbon ion therapy. It is developed for educational and research purposes only and is not intended for clinical use. The software environment is completely coded in MATLAB [wieser_development_2017]. In this thesis, the feature of proton therapy is focused on.

The matRad workflow is visually illustrated in figure 2.4.

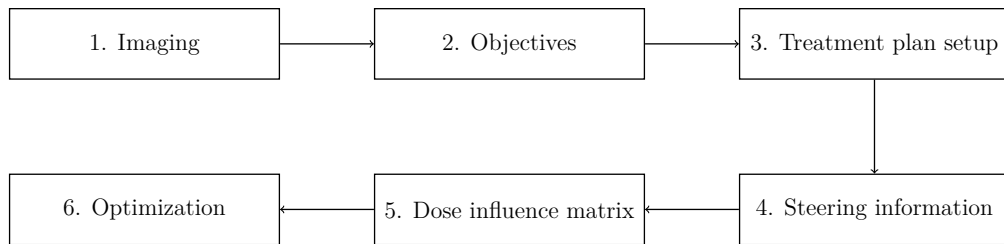


Figure 2.4: Illustration of the matRad workflow consisting of six different steps.

Imaging

The first step is to import an imaging dataset, e.g. MRI or CT. matRad allows the transformation of common imaging formats, such as Digital Imaging and Communications in Medicine (DICOM), into ".mat" files for compatibility reasons. This transformation can be done by using the graphical user interface (GUI) provided by matRad. All of the other functions in the following steps can be used from the source code.

Objectives

For the imported image and contours of organs at risk (OARs) and clinical target volume (CTV), objectives and constraints must be set, such as limiting the maximum dose or achieving a mean dose. In matRad, users can also define the goal's priority and set a penalty for unmet targets, with higher penalties increasing computational cost.

Treatment plan setup

The treatment plan setup sets field values, such as the gantry and couch angle, the radiation mode, e.g. photons, carbons or protons and the relative biological effectiveness (RBE) calculation method. The isocenter and number of beams are also set. Furthermore, the bixel width will be set. This bixel concept is explained further in the next step.

Steering information

All geometric information about the irradiation is stored in the "stf" structure, which uses prior steps to calculate steering information. The overall geometry follows a ray and bixel concept, illustrated in figure 2.5. In the steering information the end position coordinates are calculated, which is the focus of this thesis.

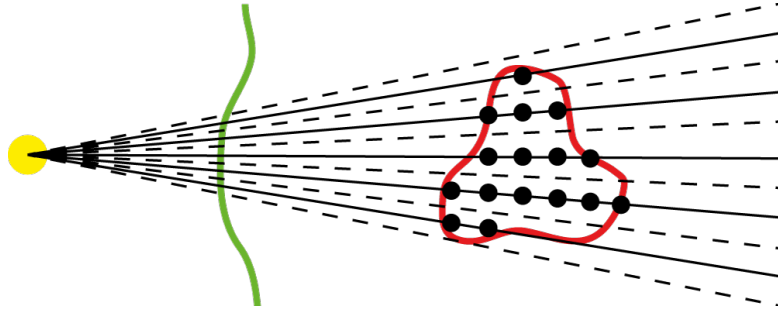


Figure 2.5: A virtual radiation source (yellow) emits equidistant rays (solid black) to cover the target volume (red) within the patient (green). In the isocenter plane (not shown), the distance between beams equals the bixel width, representing discrete fluence elements (dashed black). The depth of the target is determined along each ray, and spots (black dots) are placed accordingly [bangert_dose_nodate].

Dose influence matrix

A conventional pencil beam model is used for dose calculation [bangert_dose_nodate]. The result of the dose calculation is a dose influence matrix, which is stored in a "dij" structure. In general, the "dij" entries are calculated row by row. This matrix has the following syntax:

$$D_{ij} = \begin{pmatrix} D_{11} & \cdots & D_{1j} \\ \vdots & \ddots & \vdots \\ D_{i1} & \cdots & D_{ij} \end{pmatrix}$$

Optimization

The goal of optimization is to determine bixel and spot weights that achieve the best possible dose distribution based on the clinical objectives and constraints of the radiation treatment. With the interior optimization algorithm the dose can be calculated with

$$d_i = \sum_j D_{ij} \cdot w_j \quad , \quad (2.12)$$

where D_{ij} is the dose influence matrix and w_j is the weighting factor for each bixel j resulting from the optimization algorithm with consideration of specified objectives and constraints. This step is particularly important for creating treatment plans that provide approximate clinically relevant dose volume histogram (DVH) results. Finally, matRad offers a visualization of the resulting treatment plan with its dose distribution, as well as a printout of a DVH.

2.7 Hounsfield look-up table

A HLUT is used to convert CT numbers into corresponding tissue materials. Each CT number reflects the attenuation properties of a specific tissue relative to water, allowing differentiation between various tissue types. A $n \times m$ relative stopping power ratio (rSPR) map is generated from an $n \times m$ CT number map, representing a CT slice. This approach approximates the stopping power of different tissues by relating it to the stopping power in water through the rSPR map. With the help of linear interpolation, the material assignment based on these conversions is crucial in simulations and treatment planning, especially in radiotherapy, where precise tissue characterization directly impacts dose calculations and treatment accuracy. A candidate HLUT conversion curve is illustrated in figure 2.6.

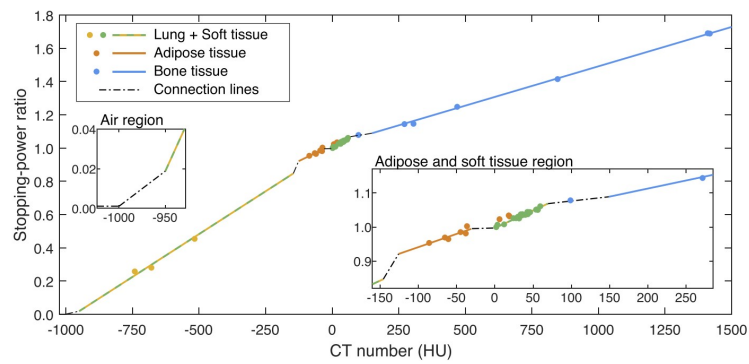


Figure 2.6: Visualization of candidate HLUT conversion. Modified from [peters_consensus_2023].

3 Materials and methods

To integrate a magnetic field influenced pencil beam spot selection algorithm for a given patient dataset (see section 3.1) and treatment plan setup (see section 3.2), analytical calculations of the proton stopping position are discussed (see section 3.3). To validate these calculations, a comparison is made with proton stopping positions determined in a MC environment (TOPAS) (see section 3.4). Finally, an optimization algorithm for magnetic field influenced pencil beam spot selection is introduced (see section 3.5).

3.1 Patient dataset

The patient dataset used in this thesis is publicly available as part of the Gold Atlas project [nyholm_mr_2018]. This project includes data from 19 patients, each with defined anatomical structures and delineation details. The delineations were independently performed in RayStation v4.7.2 via remote connections by four experienced radiation oncologists and one radiologist. Each patient’s imaging data is stored in DICOM format.

For this thesis, the dataset labeled "2_04_P" (patient ID 9) is used as an example. This dataset includes five structures: the urinary bladder, rectum, both femoral heads, and prostate. The CT images were acquired using a Toshiba Aquilion CT scanner.

The voxel dimensions are $(1.09375 \times 1.03975 \times 2) \text{ mm}^3$, with an overall resolution of $341 \times 225 \times 62$. Of the 62 slices available, the slice labeled "IMG0036" and its pixel data is presented for explanation in this thesis.

Throughout this thesis, this analyzed dataset will be referred to as the prostate patient dataset.

3.2 Treatment plan setup

A treatment plan is made for the given prostate patient dataset consisting of three sample slices with the same pixel data, for which one slice is visualized in the following figure 3.1.

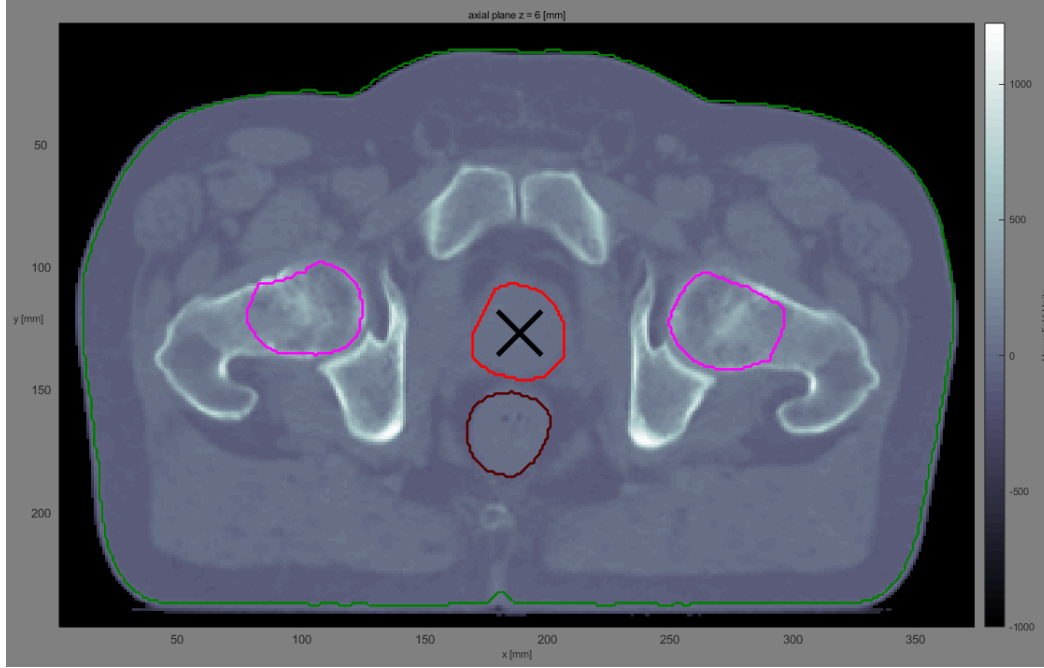


Figure 3.1: Given prostate patient CT dataset imported into matRad and visualized using the matRad GUI. A candidate CTV, representing the prostate, is highlighted (red) as well as possible OARs, representing both femoral heads and the rectum (pink and dark red).

As for objectives, the prostate is set as the only target with the goal of achieving a mean dose of 50 Gy and a default priority and penalty of 1. Further objectives and constraints for the OARs are not set, since this work does not focus on a clinically relevant result, but on the investigation of the pencil beam spot selection.

Radiation is delivered from a couch angle of 0° and a gantry angle of 270° . Protons are used, assuming a constant RBE of 1.1. The default pencil beam width of 5 mm is used. For patient materials, the matRad HLUT approach is used. The dose grid resolution is set equal to the CT grid resolution of $(1.09375, 1.09375)$ mm² for x and y. The slices show a spacing of $z = 2$ mm.

The analytical absorbed dose is calculated using the matRad default pencil beam model. Finally, the default fluence optimization algorithm is applied.

This matRad treatment plan simulation is documented and explained in more detail in the appendix (subsection A.3.1).

3.3 Analytical calculation of proton stopping position

3.3.1 Developed proton transfer algorithm

In this section, an analytical method is proposed to provide a rough estimation of the expected proton stopping position in the presence of a magnetic field.

The CT slice is assumed to be perpendicular to the magnetic field ($\vec{B} = B_z$). Since the goal is to estimate the approximate stopping location of the proton, random effects such as particle collisions and scattering are not calculated in detail. Instead, the simulation focuses on the general effects of energy loss and deflection due to the magnetic field.

The simulation of the expected proton trajectory is calculated in a step-wise manner, with two steps per voxel. In each step, the state of a given incident proton is described by its position \vec{r} , velocity \vec{v} , magnetic field \vec{B} , and material-specific stopping power (rSPR), determined by the CT number.

The initialization of a given simulation is conducted by:

$$\begin{aligned}\gamma_0 &= 1 + \frac{E_0}{m_u} \\ \beta_0 &= \sqrt{1 - \frac{1}{\gamma_0^2}} \\ \vec{v}_0 &= \beta_0 \cdot c \\ \vec{F}_0 &= -q \cdot (\vec{v}_0 \times \vec{B}_0) \\ \vec{a}_0 &= \frac{\vec{F}_0}{m_p} \quad ,\end{aligned}$$

where γ_0 is the Lorentz factor, E_0 is the initial energy in MeV, m_u is the atomic mass unit, β_0 is the relativistic velocity, c is the light of speed and \vec{v}_0 is the initial velocity of the proton. Then, the initial Lorentz force \vec{F}_0 is computed with the charge of the proton and magnetic field strength \vec{B}_0 . Finally, the initial acceleration of the proton \vec{a}_0 can be described with the initial Lorentz force and mass of the proton m_p .

Given the status of the proton at step $i-1$ ($r_{i-1}^{\rightarrow}, v_{i-1}^{\rightarrow}, \vec{B}_i, \text{rSPR}_i$), the proton status at step i can be calculated iteratively with

$$\begin{aligned}\vec{F}_i &= -q \cdot (\vec{v}_{i-1} \times \vec{B}_i) \\ \vec{a}_i &= \frac{\vec{F}_i}{m_p} \\ \vec{v}_i &= \vec{v}_{i-1} + \vec{a}_i \cdot dt_i \\ \vec{r}_i &= \vec{r}_{i-1} + \vec{v}_i \cdot dt_i \quad ,\end{aligned}$$

where the deflection, due to a magnetic field \vec{B}_i , is calculating using the Lorentz force \vec{F}_i with the charge of the proton q , the norm of the prior velocity status \vec{v}_{i-1} . With the Lorentz force and the mass of the proton m_p , the updated deflection \vec{a}_i can be computed. The next step is calculating the next velocity step \vec{v}_i using the updated acceleration, time step and prior velocity step \vec{v}_{i-1} . Finally, the updated proton position \vec{r}_i is computed using the prior position step \vec{r}_{i-1} , updated velocity and time step. The time step itself is calculated by

$$dt_i = \frac{\Delta x}{\|\vec{v}_{i-1,x}\|} \quad ,$$

where dt_i is the time step, Δx is half of the grid step size and $\|\vec{v}_{i-1,x}\|$ is the norm of the prior calculated velocity in x-direction.

The update of the velocity magnitude depends on the energy of the given proton status. Energy loss is calculated using the CSDA. The implementation of energy loss calculation using CSDA is inspired by "libamtrack" [**grzanka__libamtrack__nodate**]. This implementation allows to calculate the energy loss up to $E = 0.49$ MeV. Lower energy values are neglected. Thus, this energy is set as the energy threshold of the proton stopping position calculation in this thesis. The initial mean excitation value for Bethe Bloch (equation 2.3) is $I = 75$ eV. This was updated with the International Commission on Radiation Units and Measurements (ICRU) report 90 to $I = 78$ eV [**czarnecki__impact__2018**], which is also used in TOPAS [**perl__30__nodate**]. Thus, the developed calculation for the proton stopping position in this thesis used

the same mean excitation value $I = 78$ eV.

In order to calculate proton energy loss, a rSPR map is imported. With the help of

$$\begin{aligned}\Delta E_{i,\text{SPR}} &= \Delta E_i \cdot \text{SPR}_i \\ E_i &= \Delta E_{i,\text{SPR}} \cdot \Delta x \quad ,\end{aligned}$$

where $\Delta E_{i,\text{SPR}}$ is the energy loss with rSPR map value applied relative to the current proton position, ΔE_i is the energy loss calculated using the "libamtrack" implementation [grzanka_libamtrack_nodate] and SPR_i is the rSPR map value relative to the current proton position. Finally, the energy is updated using the updated energy loss and step size Δx .

A detailed documentation of this developed algorithm can be found in the appendix (section A.1).

To verify the developed algorithm of proton transfer, cases of increasing geometric complexity are analyzed. First, the helical radius (equation 2.9) is verified via a simulation in a vacuum. For this simulation an energy loss of $\frac{dE}{dx} = 10^{-14} \frac{\text{keV}}{\mu\text{m}}$ is assumed. Next, simulations are conducted in water (CT number: 0 HU [denotter_hounsfield_2024]), bone (CT number: 1000 HU [denotter_hounsfield_2024]), and prostate patient phantoms (CT number: on patient CT) with and without a magnetic field. The voxel dimensions and overall resolutions of all phantoms are set equal to the patient dataset (see section 3.1). Each of these cases is compared to MC based calculations of proton stopping positions under matching conditions, including initial energy, beam source position, and the same HLUT. The specific HLUT used in verification is the "Schneider" HLUT [schneider_correlation_2000].

3.3.2 matRad

The dose influence matrix D_{ij} , generated during matRad treatment planning according to given objectives and constraints, is analyzed to obtain the planned proton pencil beam stopping position, which is determined by the distal fall-off at 80 % of the relative maximum dose R_{80} on the laterally integrated dose distribution. With the stopping positions of all pencil beams used in the initial treatment plan, a spot grid, that is supposed to fill a given CTV, can be achieved. The CTV is visualized by exporting the variable "voiTarget" inside the steering information in "matRad_generateStf.m" is exported. This analysis is documented and can be seen in the appendix (subsection A.3.2).

3.3 Analytical calculation of proton stopping position

To further analyze the developed algorithm, initial positions and energy values for each bixel in the treatment field are exported and input into the algorithm code with the matRad HLUT. The resulting spots are then compared. Finally, a magnetic field is introduced to assess resulting differences, and deflected spots are optimized.

3.4 Monte Carlo based calculation of proton stopping position

The parameter setup used in this thesis and two candidate methods for verifying the analytical calculation of the proton stopping position for both $B = 0$ T and $B > 0$ T are described in subsection 3.4.1, subsection 3.4.2 and subsection 3.4.3 respectively.

3.4.1 Simulation setup

The default physics settings are applied for all TOPAS simulations in this thesis.

Additionally, the beam model is configured with an energy spread of 0, a "Flat" position distribution, and an "Ellipse" cutoff shape. The cutoffs in both the x- and y-directions are set to 0.5 cm, and the position spread is set to 0.1 mm. No angular distribution is applied. All simulations use the default random seed and a constant initial proton count of $N = 10^5$. The beam is positioned at the left, center-aligned starting voxel of the CT dataset, as illustrated in figure 3.2. Detailed simulation settings and magnetic field configurations are provided in subsection A.2.1.

Intuitively, each voxel in a CT slice has a different CT number. For all simulations, the "Schneider" conversion method is used to assign CT numbers to tissue parameters. The "Schneider" conversion method is implemented with the following lines:

```
includeFile = HUtoMaterialSchneider.txt  
s:Ge/Patient/ImagingtoMaterialConverter = "Schneider"
```

For absorbed dose measurements, a volume scorer is used. This scorer divides a geometric volume into symmetrical bins. Imported CT datasets for different materials are divided into multiple voxels. Each voxel contains the absorbed dose value for the corresponding bin. The bin size for all dimensions is set equivalent to the default CT grid size.

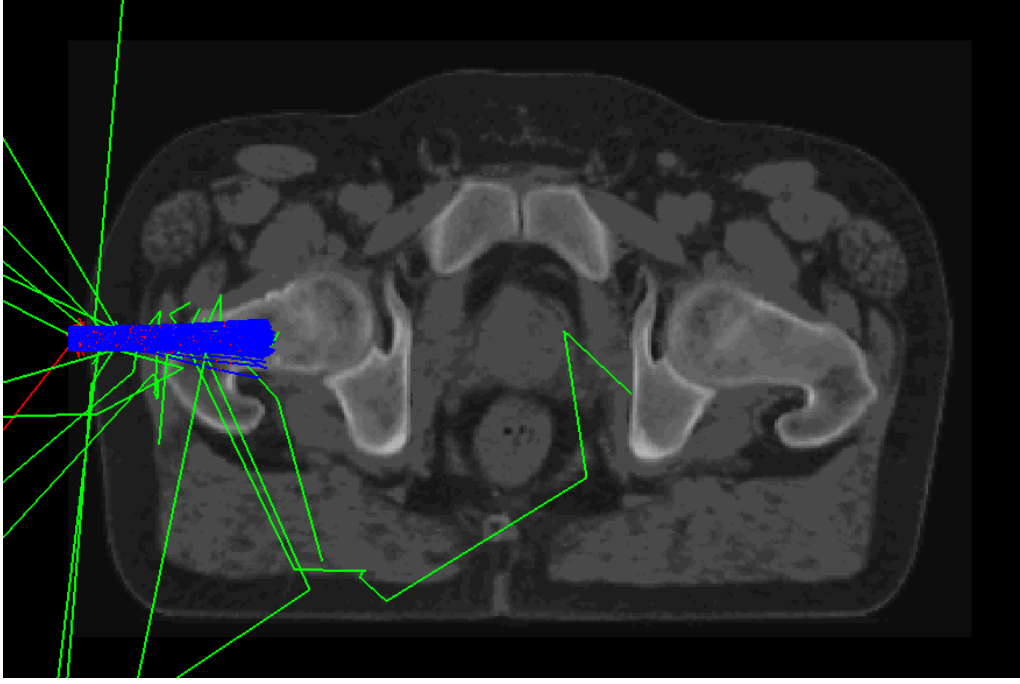


Figure 3.2: Visualization of the beam start position in TOPAS. In this example, a prostate patient CT dataset is irradiated with protons ($E = 100$ MeV, $N = 10^3$). Primary particles, i.e. protons are marked (blue lines), as well as secondary particles, i.e. electrons (red lines) and gamma rays (green lines).

3.4.2 Estimation of proton stopping position for $B = 0$ T

In the absence of a magnetic field, no deflection occurs in proton trajectories. The 80 % distal fall-off of the laterally integrated proton dose distribution is taken as the stopping position, consistent with the proton stopping calculation used in matRad (see subsection 3.3.2).

3.4.3 Estimation of proton stopping position for $B > 0$ T

To estimate the proton stopping position in a magnetic field, the following method is proposed. First, the 3D dose distribution $D(i, j, k)$ is first simplified to a simple proton trajectory curve, $y(x)$, by applying a Gaussian fit to the two-dimensional (2D) dose cross section at each depth x , i.e.,

$$y(x) = \max \left(\text{Gaussian} \left(D(x, :, :) \right) \right), \quad (3.1)$$

3 Materials and methods

where $y(x)$ is regarded as the expected trajectory of the corresponding magnetic field influenced deflected proton pencil beam (see figure 3.3).

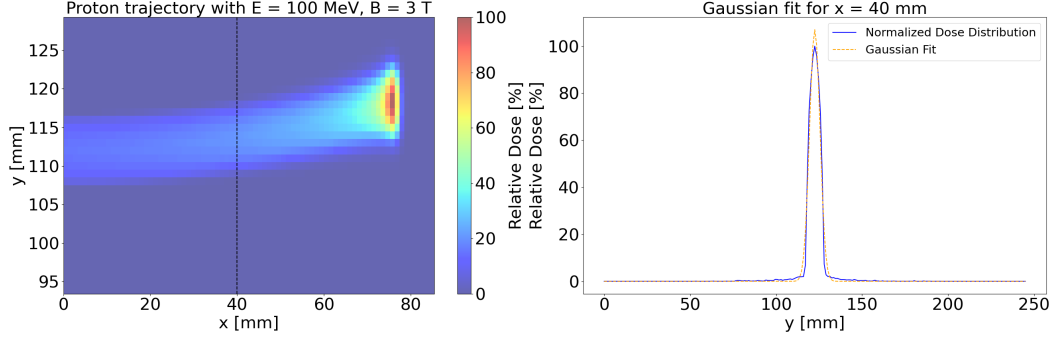


Figure 3.3: Visualization of magnetic field influenced proton trajectory simulated in TOPAS (left). A candidate value of $x = 40$ mm is marked and the one-dimensional (1D) dose distribution along y with Gaussian fit for specified x value is demonstrated (right).

Once the proton trajectory is determined, the proton stopping position is calculated using the following method. It is assumed that the path length a proton travels is primarily governed by stopping power, which is minimally influenced by the deflection induced by a magnetic field. Therefore, the trajectory lengths of protons with and without a magnetic field are considered approximately equivalent. The proton stopping position is determined by equating the path length of a proton traveling along the deflected trajectory in a magnetic field to the path length of the same proton pencil beam (with identical energy and initial position) in the absence of a magnetic field. The stopping position is then calculated as follows: First, the trajectory length of P_2 , denoted s_{L2} , is determined based on the 80 % distal fall-off of the corresponding laterally integrated dose distribution. Subsequently, the trajectory length of P_1 , denoted s_{L1} , is calculated by

$$s_{L1} = \sum_i \sqrt{\Delta x_i^2 + \Delta y_i^2} \quad \text{with} \quad \Delta x_i = x_i - x_{i-1} \wedge \Delta y_i = y_i - y_{i-1} \quad , \quad (3.2)$$

until step i for which $s_{L1} = s_{L2}$. Finally, (x_i, y_i) is regarded as the proton stopping position of P_2 . This approximation is visualized in figure 3.4.

A detailed code implementation can be found in appendix (subsection A.2.2).

The results of the MC based calculation of proton stopping position are assumed to be the ground truth for verifying the analytically calculated proton stopping position.

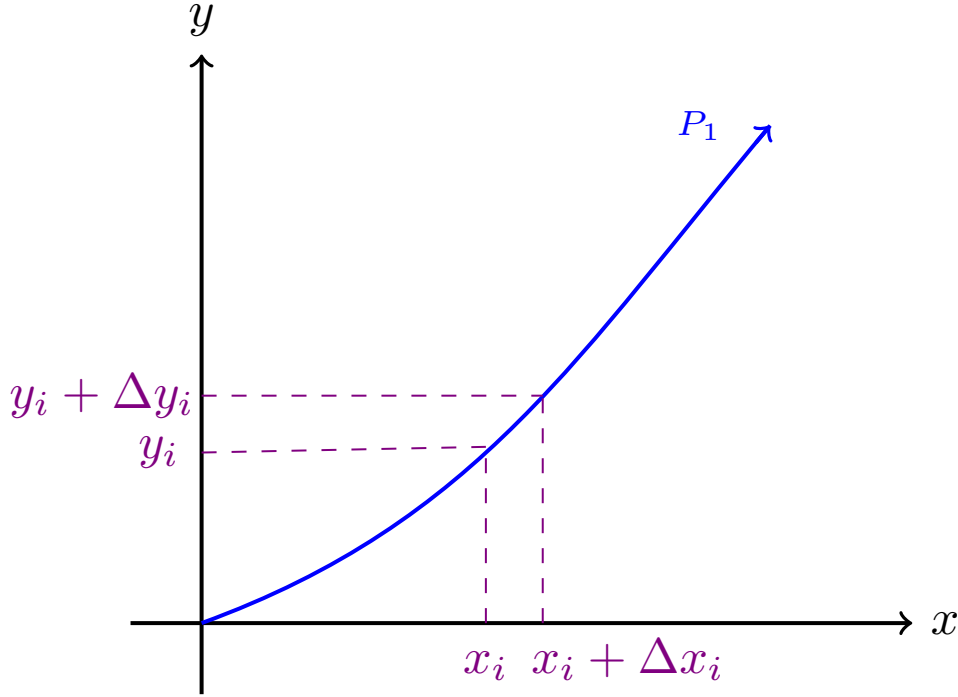


Figure 3.4: Illustration of magnetic field influenced proton beam trajectory approximation. P_1 refers to the magnetic field influenced pencil beam.

3.5 Optimization for magnetic field introduction

The purpose of this section is to present a method for modifying and adapting the pencil beam spots of the initial treatment field considering the impact of the influence of an additional given magnetic field, ensuring that the modified spots provide approximately the same coverage of the CTV as the original spots without a magnetic field.

First, for a given pencil beam i in the initial treatment field with initial position \vec{r}_i and energy E_i , the proton stopping position without a magnetic field, \vec{V}_i , is calculated using the developed algorithm (see subsection 3.3.1). Next, for the same pencil beam, the proton stopping position in the presence of the given magnetic field, $\vec{V}_{M,i}$, is determined. Finally, the updated new position $\vec{r}_{op,i}$ and energy $E_{op,i}$ is obtained via gradient descent optimization (see section 2.3) by minimizing the difference between $\vec{V}_{M,i}$ and \vec{V}_i . Thus, $\vec{r}_{op,i} \approx \vec{V}_i$ is assumed. This is visualized in figure 3.5.

3 Materials and methods

The gradient descent parameters used are:

```
learning_rate_E = 0.0005; % Step size for energy
learning_rate_y = 0.0005; % Step size for position (if y is adjusted)
tolerance = 1e-4; % Convergence tolerance
max_steps = 200; % Maximum iterations
```

A detailed documentation of the computational gradient descent implementation can be seen in the appendix (subsection A.4.1).

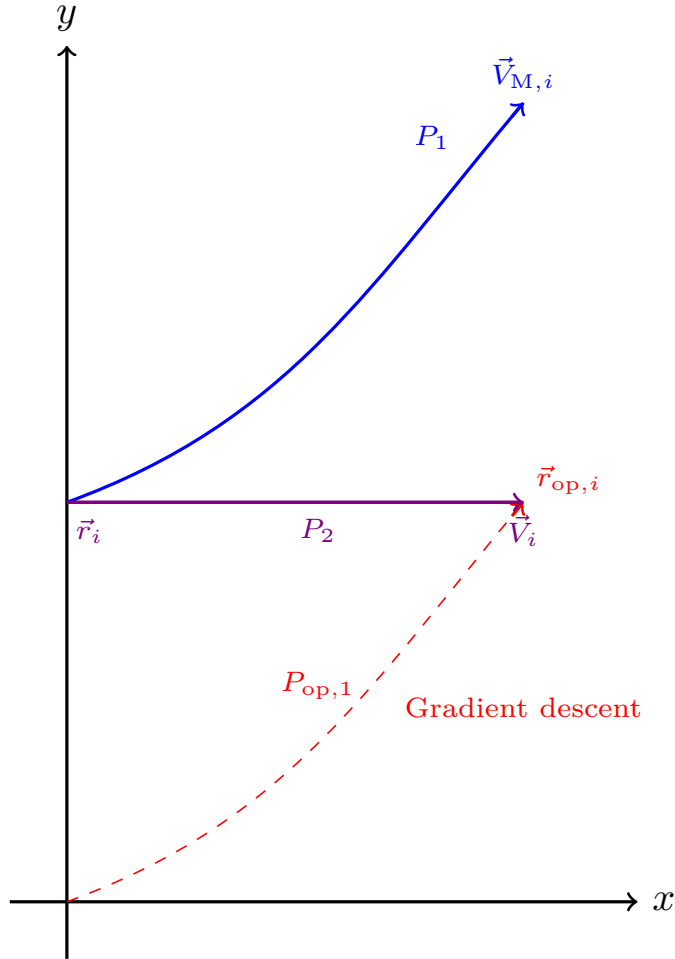


Figure 3.5: Illustration of gradient descent optimization. P1 marks the magnetic field influenced pencil beam with proton stopping position $\vec{V}_{M,i}$. P2 refers to the pencil beam without a magnetic field with initial position \vec{r}_i and proton stopping position \vec{V}_i . After gradient descent optimization, the pencil beam $P_{op,1}$ stopping position $\vec{r}_{op,i}$ is marked.

4 Results

This chapter presents the results of the developed algorithm across cases of increasing complexity, verified against ground truth references and previously discussed methods (see section 4.1). The analysis then extends to the initial results of a candidate treatment field using the TPS, comparing it to the developed algorithm, and exploring pencil beam spot optimization under the influence of a magnetic field (see section 4.2).

4.1 Verification of proton stopping position

4.1.1 Vacuum state

The comparison between radii calculated using the developed proton transfer algorithm (r_{ana}) and analytical calculation using equation 2.9 (r_{eq}) is listed in table 4.1.

In a vacuum, with a fixed magnetic field strength of 1.5 T, relative differences remain positive and converge around an average of 0.3075 %. Additionally, relative differences decrease as energy increases. For a set energy of 100 MeV, analysis across four different magnetic field strengths shows that positive relative differences continue, converging around 0.3275 %. Varying magnetic field strength thus appears to produce a stable relative difference. Absolute differences slightly increase with higher energy values but decrease with increasing magnetic field strength.

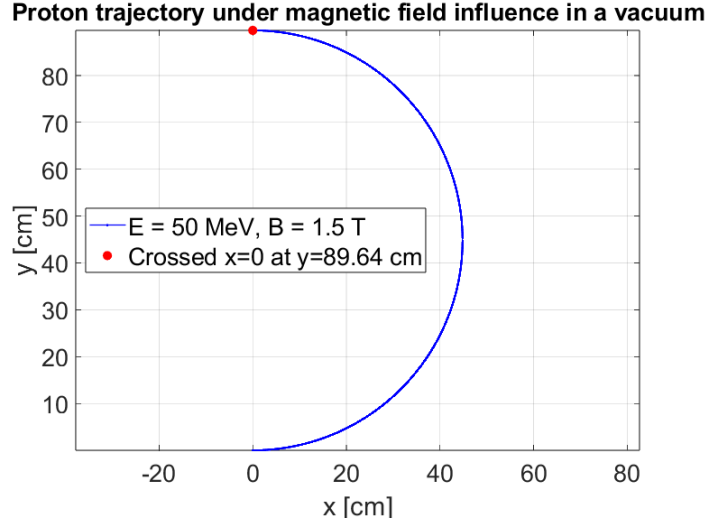


Figure 4.1: Example proton trajectory in a vacuum under influence of a magnetic field with $E = 50$ MeV and $B_z = 1.5$ T resulting from MATLAB simulation. Highlighted position (red dot) marks the diameter of the circular trajectory.

Table 4.1: Comparison of resulting radii calculated using the developed proton transfer algorithm (r_{ana}) and analytical equation for the helical radius (r_{eq}) under influence of a magnetic field (B_z) in a vacuum state. The first table has a fixed magnetic field strength with four different energy values, while the second table shows the results for a fixed energy value and four different magnetic field strengths. Absolute and relative differences between those radii are also listed.

B = 1.5 T

E [MeV]	r_{ana} [cm]	r_{eq} [cm]	Abs. Difference [cm]	Rel. Difference [%]
50	65.76	65.53	0.23	0.35
100	89.63	89.35	0.28	0.31
150	106.03	105.71	0.32	0.30
200	118.46	118.14	0.32	0.27

E = 100 MeV

B [T]	r_{ana} [cm]	r_{eq} [cm]	Abs. Difference [cm]	Rel. Difference [%]
0.5	268.89	268.04	0.85	0.32
1	134.45	134.02	0.43	0.32
2	67.23	67.01	0.22	0.33
3	44.82	44.67	0.15	0.34

4.1.2 Water phantom

B = 0 T

An example of proton beam trajectories calculated using the developed proton transfer algorithm and MC simulations is visualized in figure 4.2. For the MC simulated pencil beam, the laterally integrated dose and corresponding proton range, R_{80} , are shown in figure 4.3. A comparison of proton ranges calculated using the developed proton transfer algorithm (r_{ana}) and those extracted from MC simulations (r_{MC}) for proton beams with different energies in a water phantom without a magnetic field is listed in table 4.2.

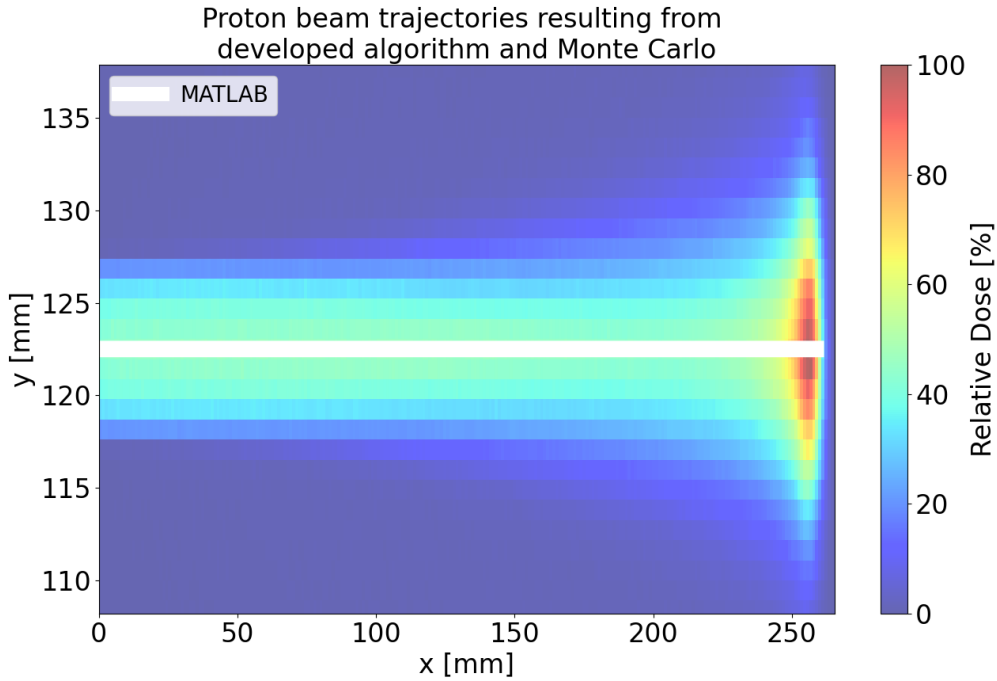


Figure 4.2: Resulting proton beam trajectories from the developed algorithm and MC simulations (TOPAS) in water for $B_z = 0$ T and $E = 200$ MeV. The white line shows the trajectory calculated in the developed algorithm, while the heatmap dose distribution refers to the MC simulated proton beam trajectory.

Qualitatively, the proton trajectory from a MC simulation shows a dimensional spread, contrasting with the simpler trajectory from the developed proton transfer algorithm. Furthermore, the stopping position calculated using the developed algorithm occurs shortly after the maximum relative dose observed in the MC based calculation.

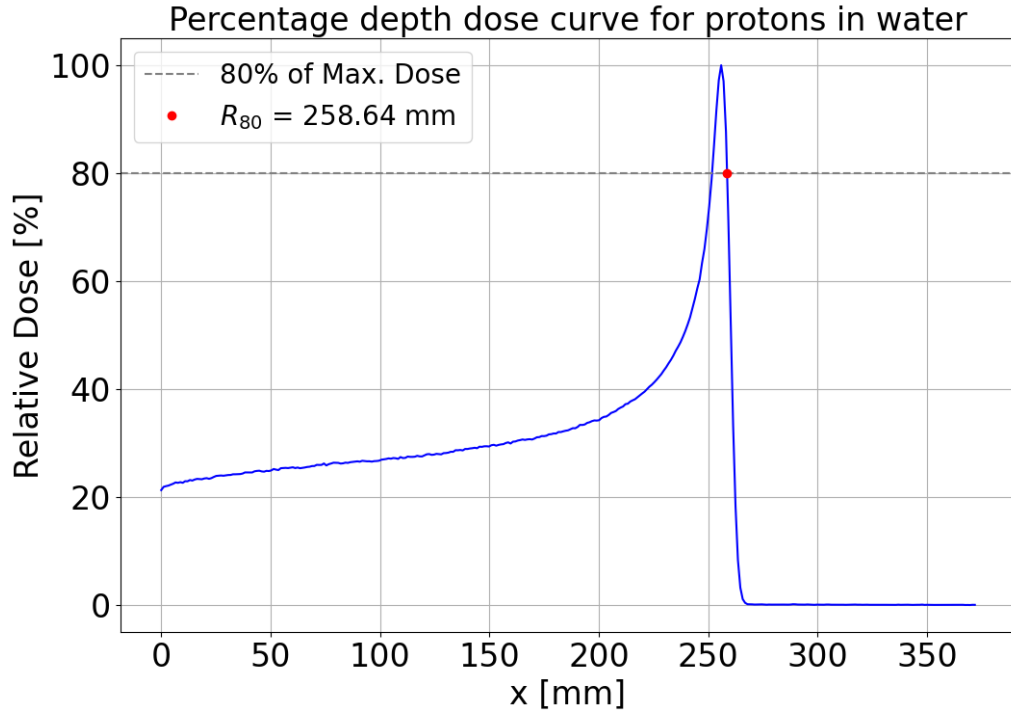


Figure 4.3: Resulting percentage depth dose curve with marked proton range, R_{80} , from a MC simulation (TOPAS) in water for $E = 200$ MeV and $B = 0$ T with $N = 10^5$ protons.

Table 4.2: Comparison of proton ranges calculated analytically using the developed proton transfer algorithm (r_{ana}) and proton ranges extracted from MC simulation (TOPAS) depth dose curves (r_{MC}) for a water phantom without a magnetic field. Four different energy values are analyzed and the resulting relative and absolute differences between the two ranges can be seen.

B = 0 T				
E [MeV]	r_{ana} [mm]	r_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
50	22.19	21.26	0.93	4.37
100	76.92	76.28	0.64	0.84
150	157.21	156.87	0.34	0.22
200	258.63	258.64	-0.01	-0.0039

Without a magnetic field and for a water phantom, relative differences decrease significantly with increasing energy. In all cases, the range calculated analytically using the developed algorithm is higher, except for $E = 200$ MeV. Absolute differences also decrease, with a maximum of 0.93 mm.

The relative dose value required to achieve $s_{L1} = s_{L2}$, for further magnetic field introductions, is visualized in figure 4.4. This example corresponds to the candidate energy value previously examined in the MC simulation in water.

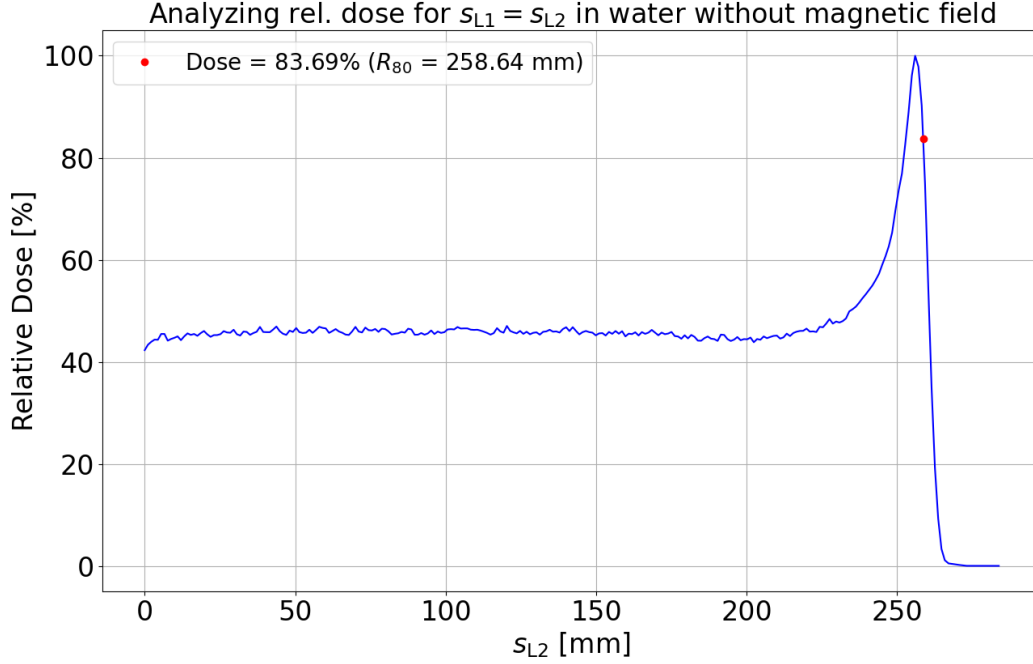


Figure 4.4: Percentage depth dose curve and marked R_{80} range with the corresponding relative absorbed dose value from a MC simulation (TOPAS) in water with $B_z = 0$ T and $E = 200$ MeV with $N = 10^5$ protons. s_{L1} refers to the proton trajectory length in a magnetic field and s_{L2} refers to the proton trajectory length without a magnetic field.

Multiple relative absorbed dose values for four different energy values with the calculation of R_{80} for $s = s_{L1} = s_{L2}$ for the analyzed water phantom are listed in table 4.3.

Table 4.3: Relative dose values calculated using MC simulations (TOPAS) in a water phantom with $B_z = 0$ T and four different energy values with prior determined proton ranges and trajectory lengths.

E [MeV]	Rel. Dose [%]
50	79.90
100	81.60
150	82.60
200	83.69

$B > 0 \text{ T}$

The relative dose value for the same example previously shown without a magnetic field (83.69 %) and its corresponding trajectory length s_{L1} for $B_z = 1.5 \text{ T}$ is analyzed and visualized in figure 4.5.

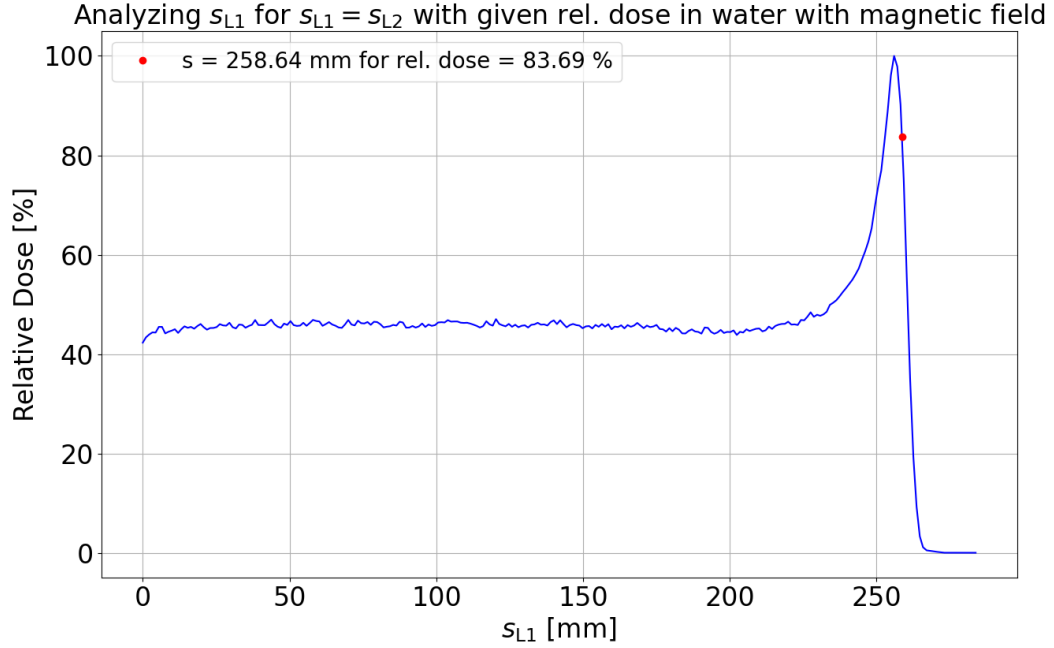


Figure 4.5: Percentage depth dose curve and marked $s = s_{L1} = s_{L2}$ value with the corresponding relative absorbed dose value from a MC simulation (TOPAS) in water with $B_z = 1.5 \text{ T}$ and $E = 200 \text{ MeV}$ with $N = 10^5$ protons. s_{L1} refers to the proton trajectory length in a magnetic field and s_{L2} refers to the proton trajectory length without a magnetic field.

The trajectory lengths s calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for the analyzed water phantom are listed in table 4.4.

In addition, table 4.5 lists the coordinates for the proton stopping positions calculated using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for the analyzed water phantom.

Table 4.4: Comparison of trajectory lengths calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for a water phantom. Different energy values and magnetic field strengths are analyzed with their absolute and relative differences listed.

B = 1.5 T				
E [MeV]	s_{ana} [mm]	s_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
50	22.19	21.26	0.93	4.37
100	76.92	76.22	0.70	0.92
150	157.21	157.01	0.20	0.13
200	258.63	258.29	0.34	0.13

E = 100 MeV				
B [T]	s_{ana} [mm]	s_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
0.5	76.92	76.26	0.66	0.87
1	76.92	76.27	0.65	0.85
2	76.92	76.24	0.68	0.89
3	76.92	76.32	0.60	0.79

Table 4.5: Comparison of end positions calculated analytically using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for a water phantom with $\vec{V} = (x, y)$. Different energy values and magnetic field strengths are analyzed and their vectorial absolute differences ($\Delta\vec{V}$) are listed. Additionally, the Euclidean distances ($|\vec{V}| = \sqrt{(x_{\text{MC}} - x_{\text{ana}})^2 + (y_{\text{MC}} - y_{\text{ana}})^2}$) are shown.

B = 1.5 T				
E [MeV]	\vec{V}_{ana} [mm]	\vec{V}_{MC} [mm]	$\Delta\vec{V}$ [mm]	$ \vec{V} $ [mm]
50	(22.19, 122.87)	(21.26, 122.89)	(0.93, -0.02)	0.93
100	(76.83, 125.79)	(76.11, 125.87)	(0.72, -0.08)	0.72
150	(156.63, 134.13)	(156.38, 133.97)	(0.25, 0.16)	0.30
200	(256.58, 150.62)	(256.21, 149.24)	(0.37, 1.38)	1.43

E = 100 MeV				
B [T]	\vec{V}_{ana} [mm]	\vec{V}_{MC} [mm]	$\Delta\vec{V}$ [mm]	$ \vec{V} $ [mm]
0.5	(76.91, 123.60)	(76.24, 123.64)	(0.67, -0.04)	0.67
1	(76.88, 124.70)	(76.22, 124.76)	(0.66, -0.06)	0.66
2	(76.76, 126.89)	(76.04, 126.99)	(0.72, -0.10)	0.73
3	(76.55, 129.09)	(75.89, 129.21)	(0.66, -0.12)	0.67

For trajectory lengths calculated and listed in table 4.4, absolute differences vary from 0.34 mm to 0.93 mm. Generally, relative differences decrease with increasing energy, while they remain nearly constant across varying magnetic field strengths. In table 4.5, absolute differences for increasing energy values appear to decrease until 200 MeV, where a slight increase in x and a larger increase in y can be observed. Additionally, distances initially decrease until reaching 200 MeV, at which point an increase in distance is seen. In contrast, for a fixed energy value and varying magnetic field strengths, dimensional absolute differences and distances appear to converge to an average value.

4.1.3 Bone phantom

B = 0 T

A comparison of proton ranges calculated using the developed proton transfer algorithm (r_{ana}) and those extracted from MC simulations (r_{MC}) for proton beams with different energies in a bone phantom without a magnetic field is listed in table 4.6.

Table 4.6: Comparison of proton ranges calculated analytically using the developed proton transfer algorithm (r_{ana}) and proton ranges ranges extracted from MC simulation (TOPAS) depth dose curves (r_{MC}) for a bone phantom without a magnetic field. Four different energy values are analyzed and the resulting relative and absolute differences between the two ranges can be seen.

B = 0 T				
E [MeV]	r_{ana} [mm]	r_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
50	15.19	14.38	0.81	5.63
100	52.63	50.17	2.46	4.90
150	107.55	103.07	4.48	4.35
200	176.93	169.84	7.09	4.17

In the bone phantom, with increasing energy values and without a magnetic field, both absolute and relative differences are larger than the results for the water phantom (cf. subsection 4.1.2). Generally, all analytically calculated proton stopping positions (r_{ana}) are higher than the MC based calculated proton stopping positions (r_{MC}). With increasing energy, absolute differences rise from an initial value of 0.81 mm up to 7.09 mm.

Relative dose values calculated with R_{80} needed for $s = s_{\text{L1}} = s_{\text{L2}}$ are analyzed for the bone phantom. These results are listed in table table 4.7.

Table 4.7: Relative dose values calculated using MC simulations (TOPAS) in a bone phantom with $B_z = 0$ T and four different energy values with prior determined proton ranges and trajectory lengths.

E [MeV]	Rel. Dose [%]
50	87.76
100	82.85
150	82.28
200	82.48

B > 0 T

The trajectory lengths s calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for the analyzed bone phantom are listed in table 4.8.

Table 4.8: Comparison of trajectory lengths calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for a bone phantom. Different energy values and magnetic field strengths are analyzed with their absolute and relative differences listed.

B = 1.5 T				
E [MeV]	s_{ana} [mm]	s_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
50	15.19	14.29	0.9	6.29
100	52.63	50.04	2.59	5.18
150	107.55	103.11	4.44	4.31
200	176.93	169.82	7.11	4.19

E = 100 MeV				
B [T]	s_{ana} [mm]	s_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
0.5	52.63	50.15	2.48	4.95
1	52.63	50.08	2.55	5.09
2	52.63	50.00	2.63	5.26
3	52.63	49.90	2.73	5.47

Relative differences decrease with increasing energy, though less pronounced than in the water phantom analysis (cf. table 4.4). Absolute differences, however, increase with energy, reaching a maximum of 7.11 mm. For varying magnetic field strengths, both absolute and relative differences converge to nearly constant values, mirroring observations in the water phantom analysis.

4 Results

For this bone phantom, table 4.9 lists the coordinates for the proton stopping positions calculated using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}).

Table 4.9: Comparison of end positions calculated analytically using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for a bone phantom with $\vec{V} = (x, y)$ assuming a constant CT number of 1000 HU. Different energy values and magnetic field strengths are analyzed and their vectorial absolute differences ($\Delta\vec{V}$) are listed. Additionally, the Euclidean distances ($|\vec{V}| = \sqrt{(x_{\text{MC}} - x_{\text{ana}})^2 + (y_{\text{MC}} - y_{\text{ana}})^2}$) are shown.

B = 1.5 T				
E [MeV]	\vec{V}_{ana} [mm]	\vec{V}_{MC} [mm]	$\Delta\vec{V}$ [mm]	$ \vec{V} $ [mm]
50	(15.19, 122.68)	(14.29, 122.67)	(0.9, 0.01)	0.90
100	(52.60, 124.05)	(50.01, 123.93)	(2.59, 0.12)	2.59
150	(107.36, 127.95)	(102.92, 127.38)	(4.44, 0.57)	4.48
200	(176.27, 135.69)	(169.18, 133.87)	(7.09, 1.82)	7.32

E = 100 MeV				
B [T]	\vec{V}_{ana} [mm]	\vec{V}_{MC} [mm]	$\Delta\vec{V}$ [mm]	$ \vec{V} $ [mm]
0.5	(52.63, 123.02)	(50.14, 122.97)	(2.49, 0.06)	2.49
1	(52.62, 123.53)	(50.06, 123.46)	(2.56, 0.07)	2.56
2	(52.58, 124.56)	(49.94, 124.42)	(2.64, 0.14)	2.64
3	(52.51, 125.59)	(49.78, 125.38)	(2.73, 0.21)	2.73

Evaluating proton stopping positions for the bone phantom shows that, with increasing energy, the difference between \vec{V}_{ana} and \vec{V}_{MC} increases significantly along x , while the increase along y is present but less pronounced. Additionally, the analytically developed proton transfer algorithm consistently yields higher proton stopping positions than those calculated using MC simulations. For fixed energy and varying magnetic field strengths, differences along x are more prominent than along y .

4.1.4 Prostate patient

B = 0 T

Finally, for the given prostate patient CT dataset, a comparison of proton ranges calculated using the developed proton transfer algorithm (r_{ana}) and those extracted from MC simulations (r_{MC}) for proton beams with different energies without a magnetic field is listed in table 4.6.

Table 4.10: Comparison of proton ranges calculated analytically using the developed proton transfer algorithm (r_{ana}) and proton ranges ranges extracted from MC simulation (TOPAS) depth dose curves (r_{MC}) for given prostate patient CT dataset without a magnetic field. Four different energy values are analyzed and the resulting relative and absolute differences between the two ranges can be seen.

B = 0 T				
E [MeV]	r_{ana} [mm]	r_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
50	31.48	30.74	0.74	2.41
100	81.03	80.80	0.23	0.28
150	150.66	152.33	-1.67	-1.11
200	244.97	249.97	-5.00	-2.04

Without a magnetic field, absolute differences between r_{ana} and r_{MC} decrease, starting from 0.74 mm and reaching -5.00 mm. Up to 100 MeV, $r_{\text{ana}} > r_{\text{MC}}$ is given, which is consistent with previous results. However, at higher energies, $r_{\text{ana}} < r_{\text{MC}}$ can be observed. Relative differences follow a similar decreasing trend.

Multiple relative absorbed dose values for four different energy values with the calculation of R_{80} for $s = s_{\text{L1}} = s_{\text{L2}}$ for the prostate patient dataset are listed in table 4.3.

Table 4.11: Relative dose values calculated using MC simulations (TOPAS) in given prostate patient CT dataset with $B_z = 0$ T and four different energy values with prior determined proton ranges and trajectory lengths.

E [MeV]	Rel. Dose [%]
50	83.33
100	83.44
150	82.95
200	79.04

$B > 0 \text{ T}$

Additionally, the trajectory lengths s calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for the given prostate patient CT dataset are listed in table 4.12.

Table 4.12: Comparison of trajectory lengths calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for given prostate patient CT dataset. Different energy values and magnetic field strengths are analyzed with their absolute and relative differences listed.

$B = 1.5 \text{ T}$				
E [MeV]	s_{ana} [mm]	s_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
50	31.49	30.41	1.08	3.55
100	82.94	80.62	2.32	2.75
150	154.96	153.60	1.36	0.89
200	254.35	256.92	-2.57	-1.01

$E = 100 \text{ MeV}$				
B [T]	s_{ana} [mm]	s_{MC} [mm]	Abs. Difference [mm]	Rel. Difference [%]
0.5	81.04	80.69	0.35	0.43
1	81.55	80.68	0.87	1.08
2	84.36	80.61	3.75	4.65
3	84.09	80.93	3.16	3.90

Relative differences between s_{ana} and s_{MC} appear to decrease with increasing energy. A trend of $r_{\text{ana}} > r_{\text{MC}}$ is observed up to the final energy value of $E = 200 \text{ MeV}$, where $r_{\text{ana}} < r_{\text{MC}}$ occurs. For a fixed energy and increasing magnetic field strength, a notable jump in absolute and relative difference values is observed from 1 T to 2 T.

The table 4.13 lists the coordinates for the proton stopping positions calculated using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for the given prostate patient CT dataset.

Table 4.13: Comparison of end positions calculated analytically using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for given prostate patient CT dataset with $\vec{V} = (x, y)$. Different energy values and magnetic field strengths are analyzed and their vectorial absolute differences ($\Delta\vec{V}$) are listed. Additionally, the Euclidean distances ($|\vec{V}| = \sqrt{(x_{\text{MC}} - x_{\text{ana}})^2 + (y_{\text{MC}} - y_{\text{ana}})^2}$) are shown.

B = 1.5 T				
E [MeV]	\vec{V}_{ana} [mm]	\vec{V}_{MC} [mm]	$\Delta\vec{V}$ [mm]	$ \vec{V} $ [mm]
50	(31.49, 123.25)	(30.39, 123.24)	(1.10, 0.01)	1.10
100	(82.82, 126.33)	(80.48, 126.10)	(2.34, 0.23)	2.35
150	(154.41, 133.80)	(152.73, 134.26)	(1.68, -0.46)	1.74
200	(252.41, 149.70)	(254.33, 150.72)	(-1.92, -1.02)	2.17

E = 100 MeV				
B [T]	\vec{V}_{ana} [mm]	\vec{V}_{MC} [mm]	$\Delta\vec{V}$ [mm]	$ \vec{V} $ [mm]
0.5	(81.03, 123.72)	(80.65, 123.84)	(0.38, -0.12)	0.40
1	(81.49, 124.72)	(80.62, 125.00)	(0.87, -0.28)	0.91
2	(84.14, 127.79)	(80.35, 127.18)	(3.79, 0.61)	3.84
3	(83.61, 130.34)	(80.19, 129.01)	(3.42, 1.33)	3.67

For the complex case of a prostate patient, no consistent trend of increase or decrease in absolute differences is observed, contrasting with previous analyses (cf. table 4.5, table 4.9). Increasing energy values lead to both increases and decreases in absolute differences. Distances show similar results, ranging from 1.10 mm to 2.35 mm. However, for a fixed energy value and increasing magnetic field strengths, a shift in absolute differences and distances is observed when increasing the field from 1 T to 2 T. Across varying energy values and magnetic field strengths, no clear trend or correlation between \vec{V}_{ana} and \vec{V}_{MC} is apparent.

An example of two magnetic field proton beam trajectories for analyzed prostate patient CT dataset is visualized in figure 4.6.

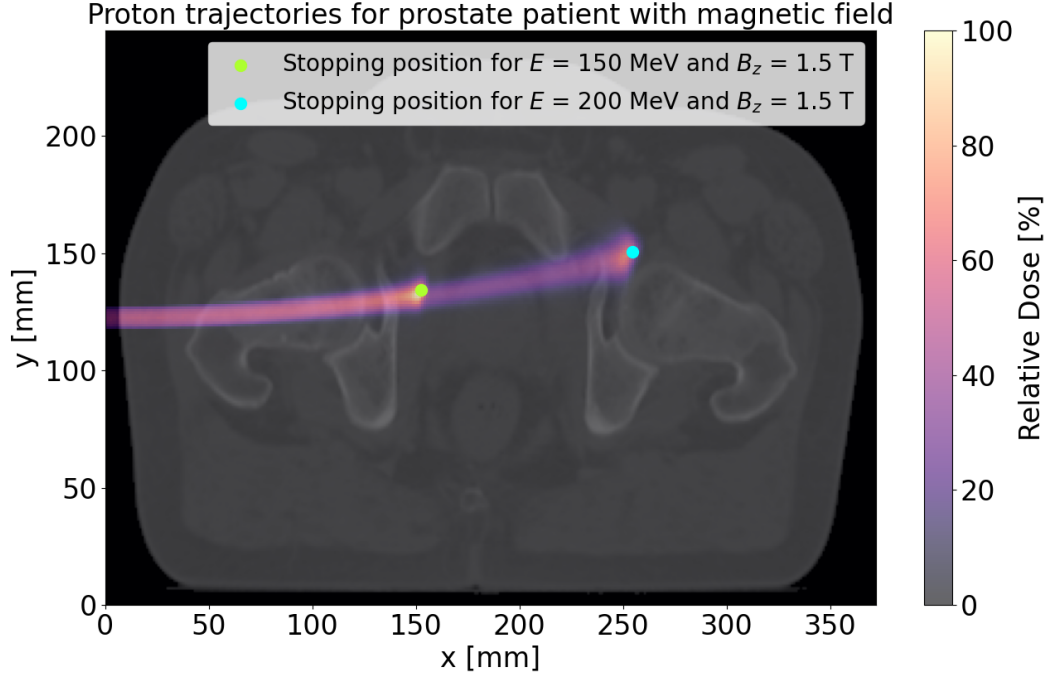


Figure 4.6: Visualization of proton beam trajectories from MC simulations (TOPAS) with an introduced magnetic field ($B_z = 1.5$ T) and two candidate energy values ($E = 150$ MeV and $E = 200$ MeV) for the prostate patient CT dataset. The proton stopping positions are highlighted for $E = 150$ MeV at $\vec{V}_{150 \text{ MeV}} = (152.73, 134.26)$ mm (light green dot) and for $E = 200$ MeV at $\vec{V}_{200 \text{ MeV}} = (254.33, 150.72)$ mm (cyan dot), where $\vec{V} = (x, y)$.

4.2 Recalculation of spot selection for a matRad treatment field

After comparing the results from the developed proton transfer algorithm with those obtained from MC simulations, the focus shifts to the TPS matRad. Initially, a treatment field is introduced along with its resulting dose distribution (see subsection 4.2.1). The initial starting positions and energy values are exported for comparison of bixel end-positions within the CTV (see subsection 4.2.2). Furthermore, a magnetic field is introduced, and position differences are analyzed (see subsection 4.2.3). Finally, the gradient descent algorithm is applied to optimize and recalculate deflected coordinate positions, shifting them towards the initial end positions (see subsection 4.2.4).

4.2.1 Proton treatment field

The prostate patient CT is overlaid with the combined dose distribution of 88 proton pencil beams, generated by the TPS to cover the CTV, as visualized in figure 4.7.

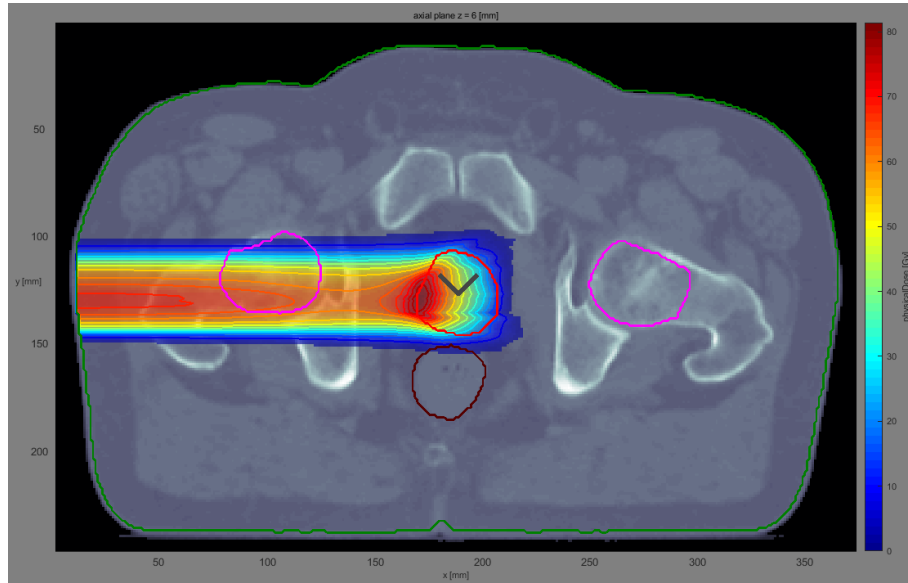


Figure 4.7: Dose distribution for given prostate CT dataset with objective of achieving a mean dose of 50 Gy. A candidate CTV (prostate) is highlighted in red, along with potential OARs (femoral heads and rectum) in pink and dark red.

The exported stopping positions for each pencil beam for this treatment field are visualized in figure 4.8.

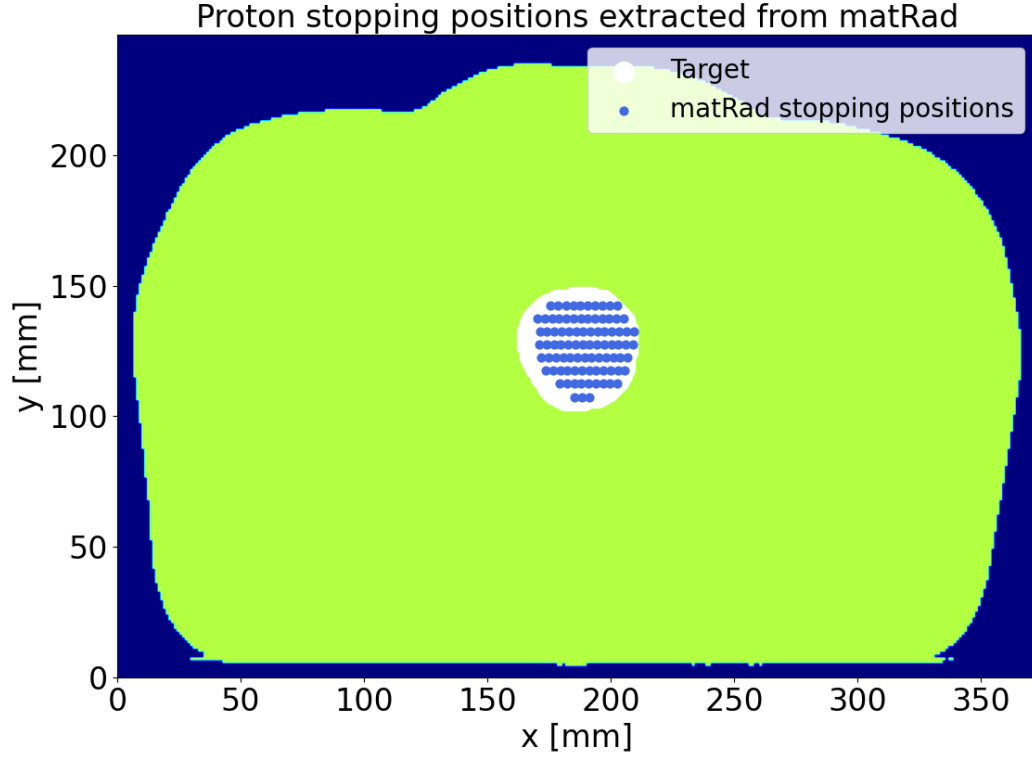


Figure 4.8: Visualization of a simplified contour representing only the body contour of given prostate patient CT dataset. The CTV is highlighted (white) as well as the proton stopping positions for each pencil beam retrieved from the "dij" matrix after matRad irradiation (blue dots).

For the following analysis, a section of the body contour around the CTV is focused on to facilitate visualization and to better differentiate pencil beam spots.

4.2.2 Verifying treatment field beam positions

The following figure 4.9 visualizes the proton stopping positions calculated using the developed proton transfer algorithm and compares them with previously mentioned matRad extracted proton stopping positions.

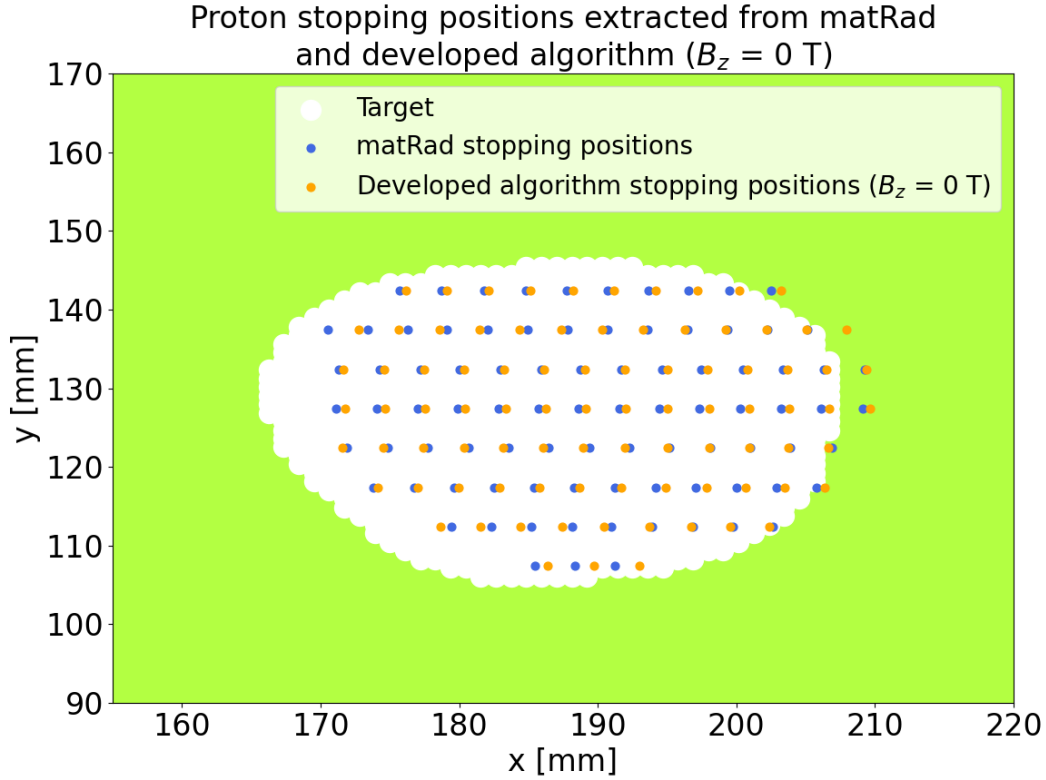


Figure 4.9: Visualization of the enlarged prostate patient CT dataset with high-lighted CTV (white). Qualitative comparison between proton stopping positions for each pencil beam calculated via the developed proton transfer algorithm (orange dots) and matRad (blue dots) using the same initial positions and energy values.

Using identical initial starting positions and energy values, a trend is observed across the rows. For instance, in the first row, the proton stopping positions are closer to the matRad stopping positions compared to those in the second row. However, within each row, a consistent difference is maintained between the proton stopping positions calculated using the developed proton transfer algorithm and the matRad stopping positions.

A quantitative analysis is done by calculating the Euclidean distances of matRad and proton stopping positions calculated via the developed proton transfer algorithm. This is visualized as a histogram in figure 4.10.

Distances between unoptimized ($B_z = 0$ T) and matRad proton stopping positions

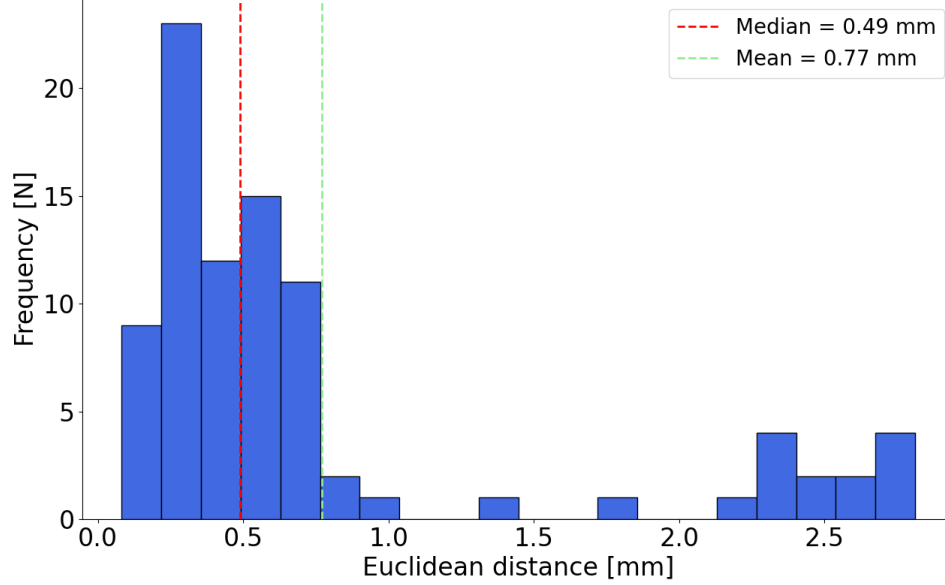


Figure 4.10: Histogram for quantitatively analyzing difference between stopping positions calculated via the developed proton transfer algorithm ($B_z = 0$ T) and matRad extracted stopping positions using initial matRad starting positions and energy values. Median (red) and mean (green) values are highlighted.

A general trend is observed, with most pencil beams showing distances below 1.0 mm. The median distance is 0.49 mm, and the mean distance is 0.77 mm.

4.2.3 Calculation of deflected spots

The initial proton stopping positions extracted from matRad are visually compared to the proton stopping positions influenced by a magnetic field ($B_z = 1.5$ T) in figure 4.11. The magnetic field influenced stopping positions are calculated using the developed proton transfer algorithm.

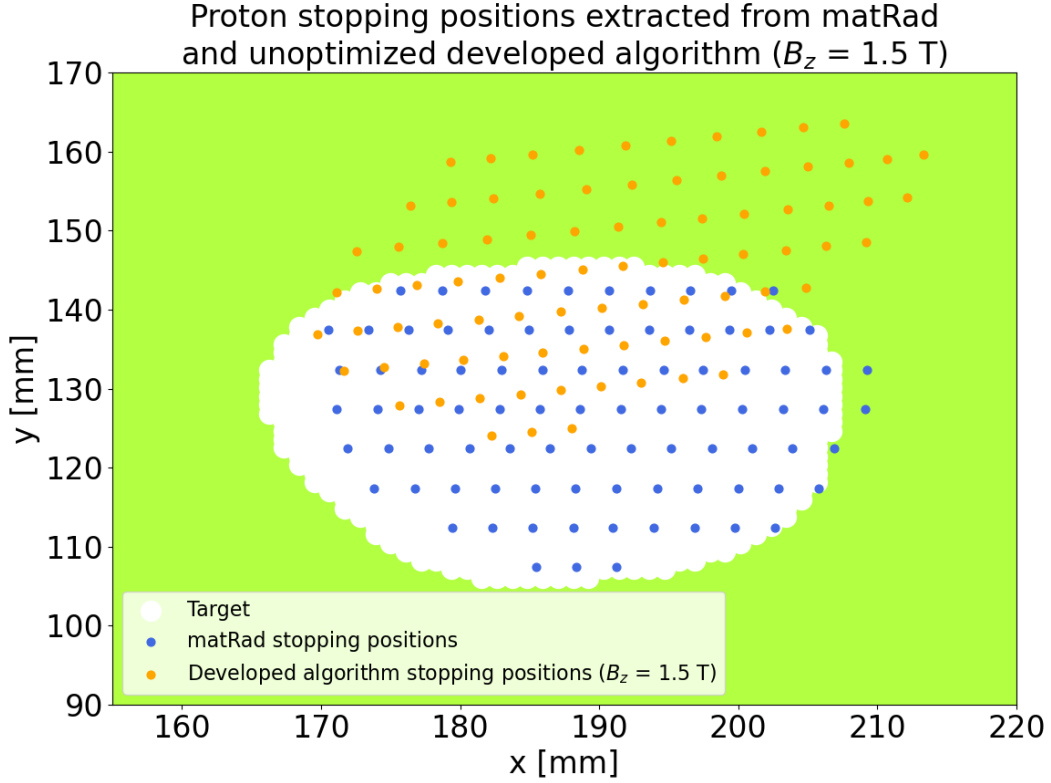


Figure 4.11: Visualization of the enlarged prostate patient CT dataset with highlighted CTV (white). Qualitative comparison between stopping positions calculated using the developed proton transfer algorithm (orange dots) and extracted from matRad (blue dots) using the same initial positions and energy values. A magnetic field ($B_z = 1.5$ T) is introduced in the developed proton transfer algorithm.

Without further optimization of the original pencil beams in terms of energy and position, the CTV would no longer be fully covered under the influence of a magnetic field.

A histogram visualizing the Euclidean distances between stopping positions calculated using the developed proton transfer algorithm and extracted from matRad is visualized in figure 4.12.

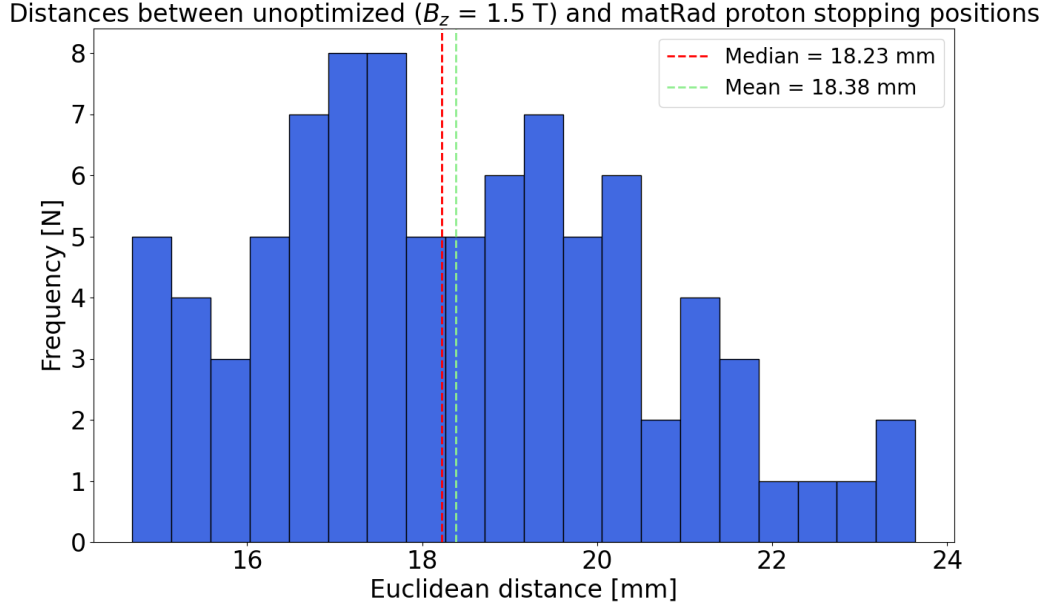


Figure 4.12: Histogram for quantitatively analyzing difference between stopping positions calculated using the developed proton transfer algorithm and stopping positions extracted from matRad using initial matRad starting positions and energy values. A magnetic field ($B_z = 1.5$ T) is introduced in the developed proton transfer algorithm. Median (red) and mean (green) values are highlighted.

The introduction of a magnetic field results in larger distances between the proton stopping positions calculated using the developed proton transfer algorithm and those extracted from matRad. The median distance is 18.23 mm, while the mean distance is 18.38 mm.

4.2.4 Optimization of deflected spots

Optimized proton stopping positions, calculated using the developed proton transfer algorithm with the introduction of a magnetic field ($B_z = 1.5$ T), are compared with the initial matRad stopping positions without a magnetic field, as visualized in figure 4.13.

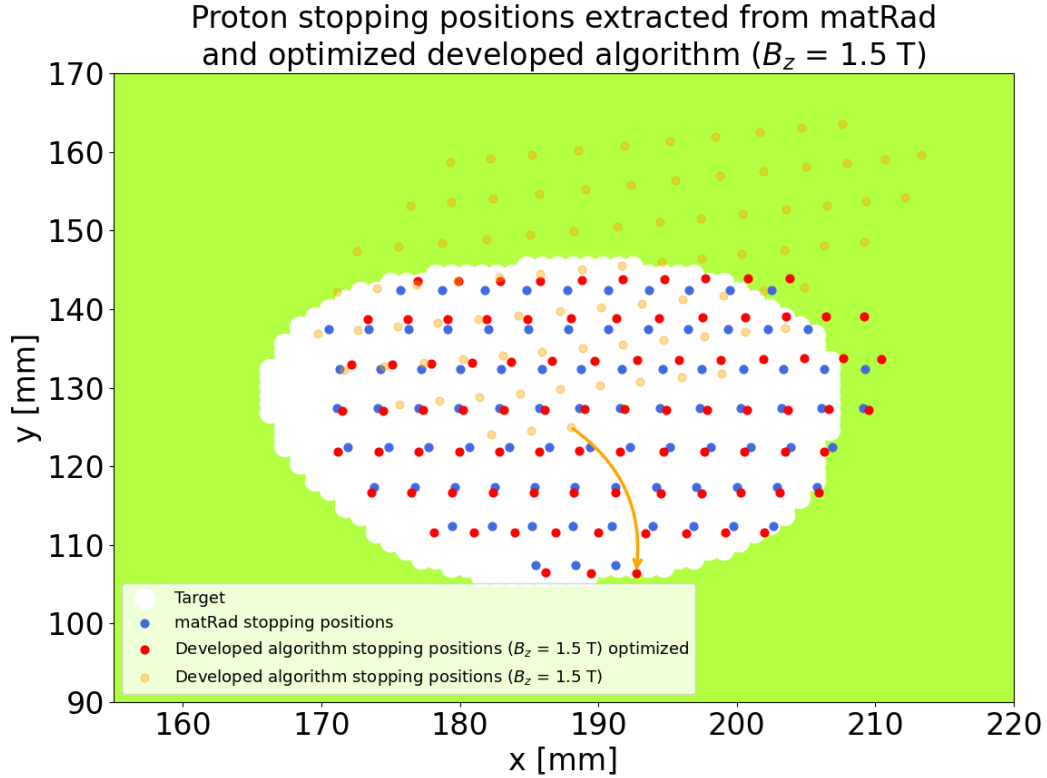


Figure 4.13: Visualization of the enlarged prostate patient CT dataset with highlighted CTV (white). Qualitative comparison between magnetic field influenced and optimized spots calculated using the developed proton transfer algorithm (red dots) and matRad extracted proton stopping positions (blue dots) for each pencil beam. A candidate spot without optimization (orange dot) with its updated position is shown (orange arrow).

Qualitatively, the optimization shows that the deflected proton stopping positions calculated using the developed proton transfer algorithm are shifted around a position approximately matching the initial matRad proton stopping positions.

The following figure 4.14 presents the quantitative analysis of the optimized spots in a histogram, comparing this dataset to the previously calculated stopping positions using the developed proton transfer algorithm without a magnetic field (see subsection 4.2.2).

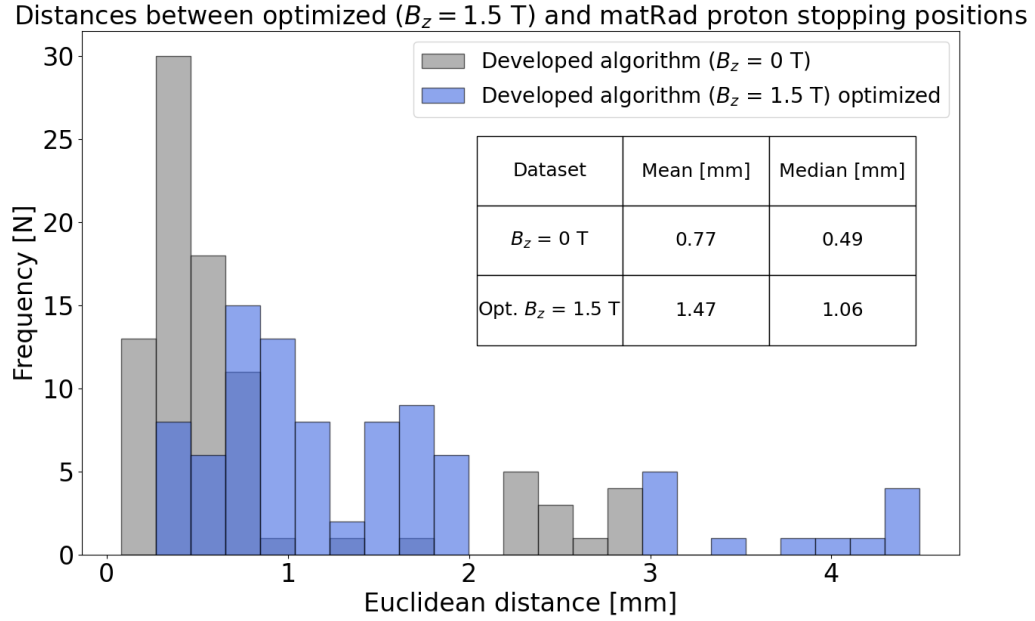


Figure 4.14: Histogram for quantitatively analyzing the differences between optimized stopping positions calculated using the developed proton transfer algorithm with an introduced magnetic field ($B_z = 1.5$ T) and proton stopping positions extracted from matRad (blue bars). For comparison, initial stopping positions calculated using the developed proton transfer algorithm without a magnetic field are shown (gray bars). Euclidean distances are used, with their frequency distribution displayed. Median (red) and mean (green) values for both datasets are highlighted.

After optimization, the majority of distances between the optimized stopping positions calculated using the developed proton transfer algorithm and proton stopping positions extracted from matRad are below 2 mm, as indicated by the blue bars. In contrast, the initial distances between stopping positions calculated using the developed proton transfer algorithm and those extracted from matRad, without the introduction of a magnetic field, were predominantly below 1 mm (cf. figure 4.10). After introducing a magnetic field and performing optimization, the mean distance increased to 1.47 mm, with the median distance at 1.06 mm.

5 Discussion

In order to achieve a possible magnetic field influenced pencil beam spot selection for the TPS matRad, several assumptions and approximations were made. In addition, simplified settings and initializations were made. In this chapter, the results presented so far are discussed and possible limitations are mentioned.

5.1 Verification of proton stopping position

5.1.1 Vacuum state

5.2 Discussion

In the analysis of a vacuum state, the calculated proton trajectory was compared to the analytical results derived using equation 2.9. Systematic uncertainties may arise from inaccuracies in the variables used for calculations. For the four candidate energy values analyzed, as shown in table 4.1, the radii computed by the developed proton transfer algorithm were consistently slightly larger than the corresponding analytical results from equation 2.9. The difference between r_{ana} and r_{eq} decreased with increasing energy values and converged to a constant relative difference as the magnetic field strength varied.

A possible explanation for this observation is that the developed proton transfer vacuum calculation assumes a small energy loss of $\frac{dE}{dx} = 10^{-14} \frac{\text{keV}}{\mu\text{m}}$ to approximate vacuum conditions. In an ideal vacuum, no energy loss would occur. Consequently, the analytical results from equation 2.9 are likely closer to the true values than those obtained using the developed proton transfer algorithm. However, the magnitude of this error in the developed algorithm is minimal. Testing with $\frac{dE}{dx} = 10^{-9} \frac{\text{keV}}{\mu\text{m}}$ yielded identical results.

Voxel-wise spatial parameterization introduces additional systematic uncertainties. While the analytical calculation assumes an infinitesimally small grid size, this

thesis used a grid size of 1.09375 mm, limiting the resolution for results below this threshold. Updating the proton position only at the start of each voxel may further contribute to the observed discrepancies. This uncertainty applies to all subsequent cases analyzed.

5.2.1 Water phantom

B = 0 T

For this analysis, as well as the analyses of the bone phantom and prostate patient CT dataset, the results of the developed proton transfer algorithm were compared with MC simulations in TOPAS. This introduces the possibility of statistical uncertainties, particularly since all TOPAS simulations used a constant number of initial protons ($N = 10^5$) (see subsection 3.4.1). For the water phantom without a magnetic field, further simulations with $N = 10^7$ were conducted. These showed a proton range difference of 0.01 mm. Thus, the error magnitude from using $N = 10^5$ is relatively low. All subsequent statements assume that TOPAS serves as the reference for validating the results of the developed proton transfer algorithm. Result discrepancies may also stem from uncertain systematic inputs in TOPAS.

For $B = 0$ T, the assumption $CSDA \approx R_{80}$ was applied to compare analytical proton stopping positions with MC based stopping positions in TOPAS, using the percentage depth dose curve (equation 2.11). Results in table 4.2 indicate that absolute and relative differences between the analytical and MC proton ranges decrease with increasing energy. Notably, the relative difference converges toward 0 % at $E = 200$ MeV. Additional simulations at $E = 225$ MeV confirmed a similar convergence near 0 %, with only a 0.0088 % increase in relative difference.

Differences between proton ranges calculated using the developed proton transfer algorithm and MC simulations may arise from systematic uncertainties due to the grid size resolution in the developed algorithm. Additionally, while the Schneider HLUT implementation in TOPAS is assumed to match that used in the developed algorithm, subtle differences could affect stopping power calculations and, consequently, proton stopping positions. Variations in underlying physics models also contribute. For instance, greater scattering effects modeled in TOPAS may shorten proton ranges. The default physics models in TOPAS include six modules [**perl_default_nodate**], which account for processes such as particle decay and elastic, inelastic, and capture interactions [**noauthor_decay_nodate**, **noauthor_qgsp_bic_nodate**]. In contrast, the Bethe Bloch implementation

in "libamtrack" used in this thesis focuses primarily on electromagnetic processes [grzanka_libamtrack_nodate].

The TOPAS beam model, briefly described in subsection 3.4.1 and detailed in appendix A.2.1, introduces additional systematic uncertainties. A simplified beam model was employed to approximate the developed narrow proton transfer beam. Further analysis of different beam settings may reduce discrepancies between the MC-simulated proton ranges in TOPAS and the analytically calculated ranges from the developed proton transfer algorithm.

Another source of systematic uncertainty arises from the positioning of the beam source relative to the width of the irradiated CT dataset. For the water phantom, bone phantom, and prostate patient, 62 slices with a slice thickness of 0.2 cm were imported, yielding a total width of 12.4 cm. In this thesis, it is assumed that the current beam setup fully covers the entire phantom geometry. Investigating this factor, along with alternative beam configurations, could uncover smaller absolute and relative differences between proton ranges calculated analytically using the developed proton transfer algorithm and those derived from the MC platform TOPAS. These uncertainties are expected to persist across all subsequent TOPAS simulations.

B > 0 T

In this thesis, primarily homogeneous magnetic fields were investigated. However, in MRiPT, proton pencil beam delivery is also influenced by the heterogeneous fringe fields of an MR scanner. Machine-specific look-up tables could incorporate the effects of such fringe fields [duetschler_fast_2023].

For both trajectory lengths, listed in table 4.4, and proton stopping positions, shown in table 4.5, the absolute and relative differences between results from the developed proton transfer algorithm and MC simulations appear energy dependent. Trajectories and stopping positions were analyzed using a fixed magnetic field strength with varying energy values and a fixed energy value with varying magnetic field strengths.

From table 4.5, it can be observed that for $B_z = 1.5$ T and varying energy values, the Euclidean distances between positions calculated in the developed proton transfer algorithm and with MC simulations decrease up until the analysis of $E = 200$ MeV.

When energy is fixed and magnetic field strength varies, no significant changes in distance values are observed, indicating an energy dependence. This trend could be attributed to systematic uncertainties in the developed proton transfer algorithm. Additionally, the proposed method for calculating proton stopping positions in magnetic fields may introduce further uncertainties. Exploring alternative methods based on MC simulated pencil beam trajectories influenced by magnetic fields could improve accuracy.

5.2.2 Bone phantom

B = 0 T

For the bone phantom without a magnetic field, proton ranges are listed in table 4.6. The ranges calculated using the developed proton transfer algorithm are consistently larger than those obtained from MC simulations. Absolute differences increase from 0.81 mm at $E = 50$ MeV to 7.09 mm at $E = 200$ MeV. Relative differences appear to converge toward a constant value of approximately 4.76 %, indicating the presence of systematic uncertainties in proton range calculations.

A major source of these uncertainties could be differences in how CT numbers, here 1000 HU, are converted into rSPR values between the developed proton transfer algorithm and the MC simulations. Further analysis of the Schneider HLUT method and its implementation may confirm or refute this hypothesis. Investigating alternative stopping power calculation methods could also be beneficial. For example, the material assignment (MATA) approach maps rSPR values to material properties using 40 predefined material compositions representative of human tissues, with mass density determined through linear interpolation [[permatasari_material_2020](#)].

B > 0 T

The absolute trajectory length differences for a fixed magnetic field strength and varying energy values (see table 4.8) calculated using the developed proton transfer algorithm and MC simulations increase with energy. Conversely, for a fixed energy value and varying magnetic field strength, these differences converge to a nearly constant range of 2.48 mm to 2.73 mm. Relative differences similarly stabilize around a constant value, as observed in the analysis without a magnetic field. A comparable trend is evident when examining proton stopping positions instead of distances (see table 4.9).

In the bone phantom analysis, each voxel is assigned the same CT number. For a fixed magnetic field strength and varying energy values, the increasing absolute differences suggest an energy dependent uncertainty. This trend is further supported by the nearly constant absolute differences observed for fixed energy values. These uncertainties appear multifactorial. Similar to the analysis of magnetic field influenced proton stopping positions in the water phantom (see subsection 5.1.2), the method used to calculate proton stopping positions from trajectory lengths may contribute to the observed discrepancies. Additionally, differences in rSPR value calculation and other systematic uncertainties within the developed proton transfer algorithm could further amplify the variations in proton stopping positions for fixed energy and magnetic field strength values.

5.2.3 Prostate patient

B = 0 T

The imaging for the investigated prostate patient dataset was performed using CT, which directly provides CT numbers for conversion into rSPR values. The availability of the prostate patient dataset and the simplicity of stopping power calculation are the primary reasons for focusing on CT slices in this thesis. Since the goal is MRiPT, developing a method to convert MRI datasets into CT datasets, potentially in real-time, could represent a significant step toward the realization of MRiPT. In this context, synthetic CT methods could be explored [boulanger_deep_2021].

The results in table 4.10 show that, with increasing energy values, the absolute difference between proton ranges calculated using the developed proton transfer algorithm and those obtained from MC simulations decreases. At a specific energy value ($E = 150$ MeV), the proton ranges derived from MC simulations exceed those calculated by the developed proton transfer algorithm. This observation may stem from differences in the stopping power calculation methods employed in the proton transfer algorithm and the MC simulations. This systematic uncertainty is particularly evident in the prostate patient dataset, where each voxel has a unique CT number. Further investigations using more sophisticated stopping power calculation algorithms, such as the MATA approach, could help address these discrepancies. Additionally, uncertainties in proton stopping positions, both with and without a magnetic field, are amplified by statistical effects. For example, the proton transfer algorithm computes the beam trajectory using a single proton under idealized conditions, whereas MC simulations account for the most probable stopping positions of multiple initial protons.

B > 0 T

For the trajectory lengths listed in table 4.12, similar observations and explanations can be drawn as for the proton ranges without a magnetic field discussed previously.

The proton stopping positions calculated using the developed proton transfer algorithm and derived from MC simulations, as shown in table 4.13, do not exhibit a consistent trend in absolute differences. For a fixed magnetic field strength and increasing energy values, the absolute differences and distances show no clear trend of increase or decrease, supporting the hypothesis of discrepancies in stopping power calculations. Similarly, for a fixed energy value and varying magnetic field strengths, a jump of approximately 3 mm is observed when increasing from $B_z = 1$ T to $B_z = 2$ T.

This discrepancy may be attributed to differences in stopping power calculations, which result in trajectories passing through voxels with different CT numbers. For instance, a specific proton state i calculated using the developed algorithm might pass through a voxel with a CT number representing air, while the same proton state i in the MC simulation might pass through a voxel with a CT number representing bone. Consequently, the proton stopping position calculated using the developed algorithm would yield a larger value compared to that derived from the MC simulation.

5.3 Recalculation of spot selection for a matRad treatment field

5.3.1 Proton treatment field

As shown in figure 4.7, a candidate proton treatment field was generated using matRad. However, this field is unsuitable as a treatment plan due to inadequate coverage of the CTV and the potential for overdosing. Additionally, the OARs were not considered during its creation. Further analysis is required to evaluate the use of additional beams, potentially from opposing directions, and to define dose constraints for the OARs, ensuring clinically relevant dose coverage of the CTV. Moreover, a constant proton RBE value of 1.1 was assumed in the dose calculations. Recent studies have demonstrated that the RBE of protons varies along the particle track, which could influence treatment effectiveness [[paganetti_relative_2002](#)].

5.3.2 Verifying treatment field beam positions

The figure 4.9 suggests a consistent trend within each row. Specifically, an approximately uniform difference is observed between the proton stopping positions calculated using the developed proton transfer algorithm and those extracted from matRad. A possible explanation for row-specific differences is the traversal of different materials in the prostate patient CT slice. This highlights the potential for stopping power miscalculations in the developed proton transfer algorithm, discrepancies in stopping power calculation methods between the developed algorithm and matRad, or both.

Systematic uncertainties in proton stopping position calculations are further emphasized when analyzing the Euclidean distances between stopping positions calculated using the developed algorithm and matRad (see figure 4.10). The median distance of 0.49 mm, approximately half the resolution size, supports the presence of systematic uncertainty. Additionally, random errors in stopping position estimation may arise within each voxel in matRad. For this thesis, it was assumed that proton stopping positions calculated using both the developed proton transfer algorithm and matRad occur at the same point within a voxel.

Further investigation into stopping power calculation methods, grid resolution, and the analysis of stopping positions within a voxel could help reduce these differences and improve the agreement between the developed proton transfer algorithm and matRad.

5.3.3 Calculation of deflected spots

For the calculation of deflected spots, only a single magnetic field strength ($B_z = 1.5$ T) was analyzed. Consequently, the results concerning the order of magnitude of the distance between magnetic field influenced pencil beam spots calculated using the developed proton transfer algorithm and spots extracted from matRad without a magnetic field allow for only a single hypothesis. At the analyzed magnetic field strength, an approximate distance of 18 mm is observed (see figure 4.12). This significant shift in spot positioning, also visualized in figure 4.11, intuitively suggests the need for optimization.

Further analysis incorporating additional magnetic field strengths and advanced statistical approaches is required to confirm or refute the hypothesis that optimization of pencil beam spot positioning is necessary to account for the effects of the magnetic field.

5.3.4 Optimization of deflected spots

When analyzing figure 4.13, optimized spots calculated using the developed proton transfer algorithm in the middle row appear to converge more closely to matRad spots than those in the first and last rows. Ideally, the distances between optimized stopping positions under a magnetic field and those without a magnetic field should converge. However, figure 4.14 shows that the median distance of optimized spots under a magnetic field is approximately double that of spots without a magnetic field. Specifically, the median value increased from 0.49 mm to 1.06 mm. This discrepancy likely arises from multiple factors.

First, the introduced magnetic field deflects spots in both the x and y directions, adding an extra dimension of uncertainty to the pre-existing systematic uncertainties in spot positioning. Second, the gradient descent optimization method introduces its own uncertainties, as it depends on parameters such as the learning rate, convergence criteria, and maximum iterations. The settings used (see section 3.5) may have been insufficient for achieving optimal results due to computational constraints. Additionally, the optimization was applied to a single CT slice, limiting its generalizability.

The observed increase in distances may result from a combination of uncertainties in the proton stopping position calculations by the developed algorithm and those introduced by suboptimal optimization of pencil beam spots.

To improve optimization under a magnetic field, the stopping position calculations in the developed algorithm should first be re-evaluated. Further analysis of the optimization algorithm and its parameters is also necessary. Expanding the analysis to multiple CT slices could provide further validation of the current results. Additionally, exploring alternative optimization techniques could reduce computational costs while enhancing accuracy.

A potential improvement involves utilizing the same gradient descent algorithm with modifications in spot selection to reduce computational expense, as introduced in the following chapter.

6 Conclusion and outlook

In this thesis, a method for analytically calculating magnetic field influenced pencil beam spots was successfully proposed and validated as a foundation for further integration within a research TPS.

The vacuum state analysis demonstrated converging constant relative differences between the analytical calculations and the developed proton transfer algorithm. Among the phantoms analyzed, the water phantom exhibited the lowest differences between proton ranges calculated using the developed algorithm and MC simulations. Conversely, the bone phantom showed the largest discrepancies, while the prostate patient phantom revealed no consistent trend in differences between stopping positions calculated with the developed algorithm and MC simulations for magnetic field influenced pencil beams.

The analysis identified several sources of uncertainty in the developed proton transfer algorithm, with stopping power calculations emerging as the most significant. To enhance stopping position accuracy, implementing the MATA method and recalculating all simulations is recommended. Further investigation into the resolution of the developed algorithm also appears promising. Additionally, the precise determination of stopping positions within a voxel warrants further study to ensure maximal accuracy.

The method of estimating proton stopping positions under magnetic fields based on trajectory lengths, assumed to be accurate in this thesis, requires further validation. All calculations performed in this study were limited to a single CT slice. Expanding the analysis to volumetric data would generalize the algorithm for 3D systems, which is a crucial step for its integration into a clinical TPS. Moreover, the analysis was restricted to a single irradiation starting position. Evaluating multiple and varied starting positions would provide a more comprehensive validation of the algorithm. Despite the identified systematic uncertainties and the need for further refinement, the current method provides an approximate estimation of proton stopping positions. For the prostate patient analysis, the method demonstrated an approximate error of 2 to 3 mm, which underscores its potential for application with further optimization.

Furthermore, for a candidate proton treatment field in the analyzed TPS matRad, proton stopping positions, along with the required variables for their calculation, were successfully exported and integrated into the developed proton transfer algorithm.

The analysis of initial proton stopping positions, without optimization or an introduced magnetic field, yielded a median distance of 0.49 mm. After optimization and introducing a magnetic field ($B_z = 1.5$ T), the median distance increased to 1.06 mm, approximately double the value compared to the case without a magnetic field or optimization.

The incorporation of inhomogeneous magnetic fields is crucial to improve the realism of calculations and further advance the feasibility of MRiPT.

For further analysis of the TPS matRad, a clinically relevant treatment plan needs to be explored. Practical objectives and constraints must be established for the CTV and the OARs, while incorporating multiple beam sources and varied starting positions. Additionally, developing methods to convert MRI datasets into CT datasets is essential if the current CT slice analysis is pursued. In this context, synthetic CT methods, as previously mentioned, could provide significant advantages.

Optimization was carried out using a gradient descent algorithm. A critical next step is to investigate the optimization parameters, such as the learning rate, convergence criteria, and maximum iterations, to improve its effectiveness. Moreover, extending the optimization to account for different magnetic field strengths, as well as adapting the method to a 3D system with multiple beams and CT slices, should be prioritized.

To reduce computational costs, an alternative method for spot selection is recommended. Currently, stopping positions for each pencil beam are optimized individually. By focusing the optimization on an outer layer of pencil beam spots for a given CTV, an optimized outer grid could be created for the target. The required dose for the inner layer of the target could then be delivered using standardized pencil beam scanning techniques.

Finally, after maximizing proton stopping position accuracy and optimizing magnetic field influenced pencil beam spots, the developed proton transfer algorithm must be integrated into the TPS for clinical application.

A Appendix

A.1 Developed proton transfer algorithm

A.1.1 Proton transfer algorithm

```
classdef ProtonSimulation_half
properties
    % Constant variables
    mp_MeV = 938.272013;
    mp = 1.67262158e-27; % [kg]
    c = 299792458 * 100; % [mm/s]
    atomic_mass_unit_MeV_c2 = 931.494; % MeV/c^2
    q = 1.602e-19; % Elementary charge

    % Variables one might change
    E0_MeV; % Initial proton energy
    B; % Magnetic field [T]
    initial_position; % Initial position [mm]
    initial_velocity; % Initial velocity [mm/s]

    % Arrays to store results
    positions;
    velocities;
    accelerations;
    times;
    energy_losses_adjusted;
    energies; % To store energy at each step
    total_trajectory_length;

    % Initial conditions
    grid_step; % [m]
    total_distance; % [mm]
    num_steps;
    SPR_values;
    E_MeV;
    x_distance = 0;
end

methods
```

```

function obj = ProtonSimulation_half(SCR_values, EO_MeV, B,
                                     grid_step, total_distance,
                                     initial_position,
                                     initial_velocity)

    % Constructor to initialize properties and load SCR values
    obj.SCR_values = SCR_values;
    obj.EO_MeV = EO_MeV;
    obj.B = B;
    obj.grid_step = grid_step;
    obj.total_distance = total_distance;
    obj.initial_position = initial_position;
    obj.initial_velocity = initial_velocity;
    steps = round(obj.total_distance / obj.grid_step);
    obj.num_steps = steps;
    obj.positions = zeros(3, steps);
    obj.velocities = zeros(3, steps);
    obj.accelerations = zeros(3, steps);
    obj.times = zeros(1, steps);
    obj.energy_losses_adjusted = zeros(1, steps);
    obj.energies = zeros(1, steps); % Initialize energies array
    obj.E_MeV = obj.EO_MeV;
    obj.total_trajectory_length = 0; % Initialize total
                                     trajectory length
end

function obj = initializeStep(obj)
    % Initialize first step of velocity and acceleration
    gamma0 = 1 + obj.EO_MeV / obj.atomic_mass_unit_MeV_c2;
    beta0 = sqrt(1 - 1/gamma0^2);
    v0 = beta0 * obj.c;
    v = obj.initial_velocity; % Use the initial velocity input
    F = - obj.q * cross(v, obj.B);
    a = F / obj.mp;
    obj.positions(:, 1) = obj.initial_position; % Use the initial
                                                position input
    obj.velocities(:, 1) = v;
    obj.energies(1) = obj.EO_MeV; % Store initial energy
end

function obj = simulate(obj)
    [num_rows, num_cols] = size(obj.SCR_values);

    for step = 2:obj.num_steps
        % Time step calculation based on half the grid step
        half_grid_step = obj.grid_step / 2;
        half_time_step = half_grid_step / norm(obj.velocities(1,
                                                                step - 1));

        % First half-step to mid-point of the voxel
    end
end

```

```

F_mid = - obj.q * cross(obj.velocities(:, step - 1), obj.
                        B);
a_mid = F_mid / obj.mp;

mid_velocities = obj.velocities(:, step - 1) + a_mid *
                half_time_step;
mid_positions = obj.positions(:, step - 1) + obj.
                velocities(:, step -
                1) * half_time_step
                ;

% Calculate energy loss at the mid-point
energy_loss_keV_um = obj.calculate_energy_loss(obj.E_MeV)
                ;

if obj.E_MeV < 0.49
    energy_loss_keV_um = 0;
    obj.E_MeV = 0;
    %disp(['Energy dropped below Bethe lower limit, E=0
            MeV']);
    break;
else
    x_pos = mid_positions(1) * 10; % Convert to mm
    y_pos = mid_positions(2) * 10; % Convert to mm
    x_index = min(max(round(x_pos), 1), num_cols);
    y_index = min(max(round(y_pos), 1), num_rows);
    SPR_value = obj.SPR_values(y_index, x_index);
    energy_loss_keV_um_adjusted = energy_loss_keV_um *
                                SPR_value;

    obj.energy_losses_adjusted(step) =
    energy_loss_keV_um_adjusted;
    obj.E_MeV = obj.E_MeV - energy_loss_keV_um_adjusted *
                obj.grid_step;
    obj.energies(step) = obj.E_MeV; % Store energy at
                                current step
end

% Update gamma and beta
gamma = 1 + obj.E_MeV / obj.atomic_mass_unit_MeV_c2;
beta = sqrt(1 - 1/gamma^2);
if gamma < 1
    %disp(['Gamma dropped below 1 at step ', num2str(step
        )]);
    break;
end

% Complete the second half-step to the end of the voxel
F_end = - obj.q * cross(mid_velocities, obj.B);
a_end = F_end / obj.mp;

```

```

        obj.velocities(:, step) = mid_velocities + a_end *
                                half_time_step;
        obj.positions(:, step) = mid_positions + mid_velocities *
                                half_time_step;
        obj.accelerations(:, step) = a_end;
        obj.times(step) = obj.times(step - 1) + 2 *
                                half_time_step;

        % Calculate trajectory length increment for this step
        delta_position = obj.positions(:, step) - obj.positions(:,
                                step - 1);
        delta_s = norm(delta_position); % Calculate delta_s using
                                sqrt(Δx2 + Δy2 +
                                Δz2)
        obj.total_trajectory_length = obj.total_trajectory_length
                                + delta_s; %
                                Accumulate
                                trajectory length

    end
end

function energy_loss_MeV_cm = calculate_energy_loss(obj, E_MeV)
% constant variables
me_MeV = 0.510998918; %(see https://github.com/libamtrack/library/
                                blob/master/include/AT_Constants
                                .h#L38 l. 38)
mp_MeV = 938.272013; %(see https://github.com/libamtrack/library/
                                blob/master/include/AT_Constants.h#
                                L35 l. 35)
m_MeV = 1.0079 * mp_MeV; %(see https://github.com/libamtrack/library/
                                blob/master/src/
                                AT_PhysicsRoutines.c#L278 l. 280
                                )
c = 299792458; % Speed of light in m/s
re = 2.8179403262 * 10-15; % electron radius in m
NA = 6.02214086 * 1023; % 1/mol
atomic_mass_unit_MeV_c2 = 931.494; %MeV/c2
Dirac_constant_J_s = 1.054571628e-34;
BETHE_LOWER_LIMIT_E_MEV_U = 0.49;
phase_undefined = 0;
phase_condensed = 1;
phase_gas = 2;
Z = 1; %for protons

% variables one might change!
E_restricted_keV = 1000;
I_eV = 78; % 75 eV for liquid water / 78 in TOPAS. for others,
%see https://github.com/libamtrack/library/blob/
%44dc48cfa977c05008ad12646d798d6c4b6ea504/include/AT_DataMaterial.h l
                                . 197-203

```

```

AT = 18.0006; % mass number for H2O
ZT = 10; % ordinary number for target: water (H2O: 2 * 1 + 1 * 8 = 10
)
phase = phase_condensed; %for liquid water! for others, see https://
github.com/libamtrack/library/
blob/master/include/
AT_DataMaterial.h#L246
%lines 240

% simple calculations and conversion factors
gamma = 1 + E_MeV / atomic_mass_unit_MeV_c2; %(see https://github.com
/libamtrack/library/blob/master/
src/AT_PhysicsRoutines.c#L45
lines 65-82)

assert(gamma >= 1.0);

beta = sqrt(1 - 1/gamma^2); %(see https://github.com/libamtrack/
library/blob/master/src/
AT_PhysicsRoutines.c#L45 line 75
)
mass_correction_term = 1 + (2 * (me_MeV / m_MeV) / gamma) + (me_MeV /
m_MeV)^2; %(see https://github.
com/libamtrack/library/blob/
master/src/AT_PhysicsRoutines.c#
L278 l. 281)
Wm_MeV = 2 * me_MeV * beta^2 / (1 - beta^2) / mass_correction_term; %
(see https://github.com/
libamtrack/library/blob/master/
src/AT_PhysicsRoutines.c#L285 l.
288)

beta2 = beta^2;
I_MeV = I_eV * 1e-6;
m_to_cm = 100;
MeV_to_J = 1.60217646e-13;
I_J = I_MeV * MeV_to_J;

% Stopping power

% Restricted stopping number requested?
restricted = false;

```

```

if E_restricted_keV > 0.0 && (E_restricted_keV / 1000.0) < Wm_MeV
    restricted = true;
end

% First part of stopping number
SN11 = (2.0 * me_MeV * beta2) / (1.0 - beta2);
assert(I_MeV > 0);
SN11 = SN11 / I_MeV;

if restricted
    Wm_MeV = E_restricted_keV * 1e-3;
end
SN12 = Wm_MeV / I_MeV;

% Second part of stopping number
SN2 = beta2;
if restricted
    SN2 = SN2 / 2;
    SN2 = (SN2 + (1.0 - beta2) * Wm_MeV) / (4.0 * me_MeV);
end

% Third part of stopping number (density correction following
                                Sternheimer, 1971)
delta = 0.0;
if phase ~= phase_undefined
    gamma_single = 1.0 + E_MeV / atomic_mass_unit_MeV_c2;
    assert(gamma_single >= 1.0);

    assert(E_MeV >= 0);
    beta_single = sqrt(1 - (1/gamma_single^2));

    assert(beta_single * gamma_single > 0);
    kinetic_variable_single = log10(beta_single*gamma_single);
    kinetic_variable = kinetic_variable_single;

    rho_gcm3 = 1;
    numbers_of_atoms_per_g = NA / AT;
    numbers_of_electrons_per_g = numbers_of_atoms_per_g * ZT;
    electron_density_per_cm3 = numbers_of_electrons_per_g * rho_gcm3;
    electron_density_m3_single = electron_density_per_cm3 * m_to_cm *
                                m_to_cm * m_to_cm;
    electron_density_m3 = electron_density_m3_single;

    %https://github.com/libamtrack/library/blob/master/src/
                                AT_DataMaterial.c#L294
    %line 307
    plasma_energy_J = sqrt((4 * pi * electron_density_m3 * re) *
                            Dirac_constant_J_s * c);

    C = 1.0 + 2.0 * log(I_J / plasma_energy_J);

```

```

% Find x_0 and x_1 dependent on phase, I-value, and C
if phase == phase_condensed
    if I_eV < 100
        x_1 = 2.0;
        if C <= 3.681
            x_0 = 0.2;
        else
            x_0 = 0.326 * C - 1.0;
        end
    else % I_eV >= 100
        x_1 = 3.0;
        if C <= 5.215
            x_0 = 0.2;
        else
            x_0 = 0.326 * C - 1.5;
        end
    end
else % gaseous material
    x_0 = 0.326 * C - 2.5;
    x_1 = 5.0;
    if C < 10.0
        x_0 = 1.6;
        x_1 = 4.0;
    end
    if C >= 10.0 && C < 10.5
        x_0 = 1.7;
        x_1 = 4.0;
    end
    if C >= 10.5 && C < 11.0
        x_0 = 1.8;
        x_1 = 4.0;
    end
    if C >= 11.0 && C < 11.5
        x_0 = 1.9;
        x_1 = 4.0;
    end
    if C >= 11.5 && C < 12.25
        x_0 = 2.0;
        x_1 = 4.0;
    end
    if C >= 12.25 && C < 13.804
        x_0 = 2.0;
        x_1 = 5.0;
    end
end

x_a = C / 4.606;
m = 3.0;
a = 4.606 * (x_a - x_0) / ((x_1 - x_0)^m);

```

```

    if kinetic_variable >= x_0 && kinetic_variable <= x_1
        delta = 4.606 * kinetic_variable - C + a * (x_1 -
                                                    kinetic_variable)^m;
    end
    if kinetic_variable > x_1
        delta = 4.606 * kinetic_variable - C;
    end
end
SN3 = delta;

% Forth part of stopping number (shell correction) TODO: implement

%assert(SN11 > 0);
%assert(SN12 > 0);

stopping_number = 0.5 * log(SN11 * SN12) - SN2 - SN3;

% Leading energy loss term
%(we assume that effective charge is used! otherwise,
%change code with the help of
%https://github.com/libamtrack/library/blob/
%44dc48cfa977c05008ad12646d798d6c4b6ea504/src/
%AT_StoppingPowerDataBethe.c#L49
%1. 62-65)

assert(AT > 0);
assert(beta2 > 0);

% calculation of effective charge according to Barkas-Bethe
% approximation
if Z ~= 1
    effective_charge = Z * (1.0 - exp(-125.0 * beta / (Z^(2.0/3.0))))
    ;
else
    effective_charge = 1.0 - exp(-125.0 * beta);
end

%since we only use effective charge we can say z = Z;
z = Z;

% ICRU49, p.6, after Cohen and Taylor (1986), k_MeV_cm2_g = 0.307075
energy_loss_leading_term_MeV_cm2_g = 0.307075 * (ZT / AT) * (z^2) /
    beta2;

% Energy loss
% Compute only above 1.0 MeV, otherwise theory is too wrong below
return zero

```



```

% TODO: Find smarter criterion because this may cause problems in the
% code (as it did
% TODO: with the inappropriately set lower limit for CSDA range
% integration (was 0, now 1.0 MeV)

if E_MeV >= BETHE_LOWER_LIMIT_E_MEV_U
    energy_loss_MeV_cm = energy_loss_leading_term_MeV_cm2_g *
        stopping_number;
else
    energy_loss_MeV_cm = 0;
end

% unit conversion
energy_loss_keV_um = energy_loss_MeV_cm / 10; % [keV/m] AND [MeV/mm]
energy_loss_MeV_m = energy_loss_MeV_cm / 100; % [MeV/m]
end

% Function for saving results
function saveResults(obj)
    non_zero_mask = any(obj.positions ~= 0, 1);
    filtered_positions = obj.positions(:, non_zero_mask);
    filtered_energyloss = obj.energy_losses_adjusted(:,
        non_zero_mask);
    csvwrite('trajectory.csv', filtered_positions');
    disp(['Total trajectory length s: ', num2str(obj.
        total_trajectory_length)
        , ' mm']);
end

function energy = getEnergyAtStep(obj, step)
    % Method to retrieve energy at a given step
    energy = obj.energies(step);
end

% Function for storing and saving step range
function num_non_zero_steps = displayStepRange(obj)
    non_zero_mask = any(obj.positions ~= 0, 1);
    num_non_zero_steps = sum(non_zero_mask);
end
end
end
end

```

A.1.2 Initialization of proton transfer algorithm

```

y_position = 11.2; % Example initial y-position
EO_MeV = 100; % Example initial energy in MeV
magnetic_field = [0; 0; 3]; % Example magnetic field in T
SPR = readmatrix('SPR_values.csv');

```

```
% Load SPR values SPR_values(_slice).csv watertestSPR.csv bone_SPR.csv
                                heavybone_SPR.csv

final_pos = analyzeProtonTrajectory(y_position, E0_MeV, magnetic_field,
                                SPR);

% Display last x-position
disp([num2str(final_pos(1)')]);

% New Function to Analyze Proton Trajectory
function final_position = analyzeProtonTrajectory(y, E, B, SPR)
    % Constants (needed for velocity calculation for example)
    atomic_mass_unit_MeV_c2 = 931.494; % MeV/c^2
    c = 299792458 * 100; % [mm/s]

    grid_step = 0.00109375; % [m] This is the CT grid step /"PixelSpacing
                                " one can acquire by looking at
                                metadata using 3D Slicer for
                                example
    total_distance = 83.91796875; % [mm] CT rows * CT columns * CT grid
                                step

    % Pre-calculations for initial velocity
    gamma0 = 1 + E / atomic_mass_unit_MeV_c2;
    beta0 = sqrt(1 - 1/gamma0^2);
    v0 = beta0 * c;

    % Set initial conditions based on input parameters
    initial_position = [0; y; 0]; % Initial y-position
    initial_velocity = [v0; 0; 0]; % Initial velocity

    % Create an instance of ProtonSimulation
    protonSim = ProtonSimulation_half(SPR, E, B, grid_step,
                                total_distance, initial_position
                                , initial_velocity);

    % Initialize and run the simulation
    protonSim = protonSim.initializeStep();
    protonSim = protonSim.simulate();
    protonSim.saveResults();

    % Display the number of steps with non-zero values
    % Get the number of steps with non-zero values
    num_non_zero_steps = protonSim.displayStepRange();

    % Get the final position using the number of non-zero steps
    if num_non_zero_steps > 0
        final_position = protonSim.positions(:, num_non_zero_steps); %
                                Extract the last valid
                                position
    end
end
```

```

        %disp(['Final valid position (x, y, z): ', num2str(final_position
            ')]);
    else
        %disp('No valid positions found.');
```

A.2 TOPAS

A.2.1 Simulation setup

```

# General settings

i:Ts/NumberOfThreads = 0 # Max CPU threads
includeFile = HUtoMaterialSchneider.txt # Import Schneider HULUT

# Proton source positioning
s:Ge/BeamPosition/Parent="World"
s:Ge/BeamPosition/Type="Group"
d:Ge/BeamPosition/TransX= -18.6484375 cm
d:Ge/BeamPosition/TransY= 0. cm
d:Ge/BeamPosition/TransZ= 0. cm
d:Ge/BeamPosition/RotX=90. deg
d:Ge/BeamPosition/RotY=270. deg
d:Ge/BeamPosition/RotZ=0. deg

# Proton beam model
s:So/Demo/Type = "Beam"
s:So/Demo/Component = "BeamPosition"
s:So/Demo/BeamParticle = "proton"
d:So/Demo/BeamEnergy = X. MeV # set X to desired energy value
u:So/Demo/BeamEnergySpread = 0
s:So/Demo/BeamPositionDistribution = "Flat"
s:So/Demo/BeamPositionCutoffShape = "Ellipse"
d:So/Demo/BeamPositionCutoffX = 0.5 cm
d:So/Demo/BeamPositionCutoffY = 0.5 cm
d:So/Demo/BeamPositionSpreadX = 0.1 mm
d:So/Demo/BeamPositionSpreadY = 0.1 mm
s:So/Demo/BeamAngularDistribution = "None"
d:So/Demo/BeamAngularCutoffX = 90. deg
d:So/Demo/BeamAngularCutoffY = 90. deg
d:So/Demo/BeamAngularSpreadX = 0.000000032 rad
d:So/Demo/BeamAngularSpreadY = 0.000000032 rad
i:So/Demo/NumberOfHistoriesInRun = 100000

# CT import
```

```

s:Ge/Patient/Type = "TsDicomPatient"
s:Ge/Patient/Parent = "World"
s:Ge/Patient/ImagingtoMaterialConverter = "Schneider"
d:Ge/Patient/RotX = 0. deg
d:Ge/Patient/RotY = 180. deg
d:Ge/Patient/RotZ = 180. deg
s:Ge/Patient/Field = "DipoleMagnet"
u:Ge/Patient/MagneticFieldDirectionX = 0.0
u:Ge/Patient/MagneticFieldDirectionY = 0.0
u:Ge/Patient/MagneticFieldDirectionZ = 1.0 # for Bz
s:Ge/Patient/MagneticField3DTable = "PurgMag3D.TABLE"
d:Ge/Patient/MagneticFieldStrength = X.X tesla # set X.X to desired
                                         magnetic field strength

# Dose scorer

s:Sc/MyScorer/Quantity = "DoseToMedium"
s:Sc/MyScorer/Component = "Patient"
s:Sc/MyScorer/OutputFile = "TOPAS_Beam"
s:Sc/MyScorer/OutputType = "csv"
s:Sc/MyScorer/IfOutputFileAlreadyExists = "Overwrite"

```

A.2.2 Estimation of proton stopping positions for $B > 0$ T

```

import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from scipy.optimize import curve_fit
from scipy.interpolate import interp1d

# Define Gaussian function for fitting
def gaussian(y, a, mean, sigma):
    return a * np.exp(-0.5 * ((y - mean) / sigma) ** 2)

# Read the csv-file
outputfile_topas = '../Data/TOPAS_prostate_B3_100MeV.csv'
#outputfile_topas = '../TOPAS/TOPAS_Beam.csv'
df = pd.read_csv(outputfile_topas, comment='#', header=None)

# Convert the dataframe to a numpy array
topas_datamatrix = np.array(df)

# Extract depth (x), height (y), and dose values
depth = topas_datamatrix[:, 0] # depth (x)
y_positions = topas_datamatrix[:, 1] # y positions (height)
dose = topas_datamatrix[:, 3] # dose values

```

```

# Reshape the dose data into a 2D grid based on 341 x bins, 225 y bins,
# and 7 z bins
n_x_bins = 341
n_y_bins = 225
n_z_bins = 62

dose_grid = dose.reshape(n_x_bins, n_y_bins, n_z_bins)

# Define x axis for the plot (convert from cm to mm)
x_bin_size = 0.109375 * 10 # mm per bin in x
x_axis = np.arange(n_x_bins) * x_bin_size # depth in mm

# Sum dose values across all z-bins (integrating across slices)
integrated_dose_z = np.sum(dose_grid, axis=2)

# Integrate the dose along the y-axis (collapse into 1D dose profile
# along x)
integrated_dose = np.sum(integrated_dose_z, axis=1)

# Normalize the integrated dose by its maximum value
integrated_dose_norm = (integrated_dose / np.max(integrated_dose)) * 100

# Find the peak dose index (where max dose occurs)
peak_idx = np.argmax(integrated_dose_norm)

# Search for the position after the peak where dose crosses below 80% of
# the peak dose
r80_value = 80 # 80% of the maximum dose
post_peak_dose = integrated_dose_norm[peak_idx:]
crossing_idx = np.where(post_peak_dose <= r80_value)[0][0] + peak_idx

# Get the values at the two points surrounding the R80 crossing
x1, x2 = x_axis[crossing_idx - 1], x_axis[crossing_idx]
y1, y2 = integrated_dose_norm[crossing_idx - 1], integrated_dose_norm[
    crossing_idx]

# Perform linear interpolation to find the exact x position for R80
r80_x = x1 + (r80_value - y1) * (x2 - x1) / (y2 - y1)

print(f"R80 value at depth: {r80_x:.2f} mm")

# Define y axis (height) in mm for the plot
y_bin_size = 0.109375 * 10 # mm per bin in y
y_axis = np.arange(n_y_bins) * y_bin_size # height in mm

# Initialize a list to store the (x, y, dose) values
yshift_array = []

# Boolean flag to track when the first Gaussian fit fails
fitting_failed = False

```

```
# Iterate over all x_bins
for x_bin in range(n_x_bins):
    if fitting_failed:
        x_value = x_bin * x_bin_size # Convert x_bin to depth in mm
        yshift_array.append([x_value, 0.0, 0.0])
        continue

    # Extract dose values along the y-axis for the current x_bin (summed
    # over z)
    y_dose_distribution = np.sum(dose_grid[x_bin, :, :], axis=1) # Sum
    # over all z-bins

    # Check if the dose is all zeros or close to zero
    if np.max(y_dose_distribution) <= 0:
        x_value = x_bin * x_bin_size # Convert x_bin to depth in mm
        yshift_array.append([x_value, 0.0, 0.0])
        continue

    # Normalize the dose distribution
    y_dose_distribution_norm = (y_dose_distribution / np.max(
        y_dose_distribution)) * 100

    if np.isnan(y_dose_distribution_norm).any() or np.isinf(
        y_dose_distribution_norm).any():
        x_value = x_bin * x_bin_size # Convert x_bin to depth in mm
        yshift_array.append([x_value, 0.0, 0.0])
        continue

    try:
        # Fit a Gaussian function to the normalized y-dose distribution
        popt, _ = curve_fit(gaussian, y_axis, y_dose_distribution_norm,
            p0=[np.max(y_dose_distribution_norm), np.mean(
                y_axis)
                , np.std(
                y_axis)
            ],
            maxfev=4000)

        # Extract the mean (which corresponds to the y position of the
        # Gaussian peak)
        mean_fit = popt[1]

        # Get the absolute dose value at the position where the Gaussian
        # peak occurs
        closest_y_idx = np.argmin(np.abs(y_axis - mean_fit))
        absolute_dose = y_dose_distribution[closest_y_idx]

        # Append the current x_bin (converted to depth), y value (
        # mean_fit), and absolute dose
```

```

                                to the array
x_value = x_bin * x_bin_size # Convert x_bin to depth in mm
yshift_array.append([x_value, mean_fit, absolute_dose])

except RuntimeError:
    # If curve fitting fails, store 0.0 values for this x_bin
    fitting_failed = True
    x_value = x_bin * x_bin_size # Convert x_bin to depth in mm
    yshift_array.append([x_value, 0.0, 0.0])

# Convert the list to a DataFrame
yshift_df = pd.DataFrame(yshift_array, columns=['x (mm)', 'y (mm)', 'dose
                                (absolute)'])

# Filter out rows where both y and dose are zero
yshift_df_filtered = yshift_df[(yshift_df['y (mm)'] > 0) & (yshift_df['
                                dose (absolute)'] > 0)].copy()

# Calculate cumulative distance 's'
s_values = [0.0] # Start with s = 0 for the first point
for i in range(1, len(yshift_df_filtered)):
    delta_x = yshift_df_filtered['x (mm)'].iloc[i] - yshift_df_filtered['
                                x (mm)'].iloc[i - 1]
    delta_y = yshift_df_filtered['y (mm)'].iloc[i] - yshift_df_filtered['
                                y (mm)'].iloc[i - 1]
    delta_s = np.sqrt(delta_x**2 + delta_y**2)
    s_values.append(s_values[-1] + delta_s)

yshift_df_filtered.loc[:, 's (mm)'] = s_values

# Normalize the dose
yshift_df_filtered.loc[:, 'dose_norm'] = (yshift_df_filtered['dose (
                                absolute)'] / np.max(
                                yshift_df_filtered['dose (absolute)']
                                )) * 100

# Apply a threshold to remove very small dose values (e.g., less than 1%)
dose_threshold = 1.0
threshold_idx = np.where(yshift_df_filtered['dose_norm'] < dose_threshold
                                )[0]

# Keep the next 4 points after the dose drops below the threshold
if len(threshold_idx) > 0:
    first_threshold_idx = threshold_idx[0]
    end_idx = first_threshold_idx + 4 # Keep 4 more points
    yshift_df_filtered = yshift_df_filtered.iloc[:min(end_idx, len(
                                yshift_df_filtered))]

# Toggle to show Gaussian fit for a specific bin

```

```

show_gaussian_fit = True # Set to False to disable

# Specify the bin for Gaussian example (this is where the Gaussian will
                                be plotted)
example_bin = 20 # You can change this bin as needed

# Plot the Gaussian fit for a specific bin if requested
if show_gaussian_fit:
    # Extract dose values along the y-axis for the chosen x_bin (summed
                                over z)
    y_dose_distribution_example = np.sum(dose_grid[example_bin, :, :],
                                axis=1) # Sum over z-bins

    # Normalize the dose distribution
    y_dose_distribution_example_norm = (y_dose_distribution_example / np.
                                max(y_dose_distribution_example)
                                ) * 100

    # Fit a Gaussian function
    popt, _ = curve_fit(gaussian, y_axis,
                                y_dose_distribution_example_norm,
                                p0=[np.max(y_dose_distribution_example_norm), np.
                                mean(y_axis)
                                , np.std(
                                y_axis)],
                                maxfev=4000)

    # Extract fitted Gaussian
    gaussian_fit_example = gaussian(y_axis, *popt)

    # Plot Gaussian fit
    plt.figure(figsize=(12, 8))
    plt.plot(y_axis, y_dose_distribution_example_norm, label='Normalized
                                Dose Distribution', color='blue'
                                )
    plt.plot(y_axis, gaussian_fit_example, label='Gaussian Fit', color='
                                orange', linestyle='--')
    plt.xlabel('Y Position [mm]')
    plt.ylabel('Normalized Dose [%]')
    plt.title(f'Gaussian fit for x bin {example_bin}')
    plt.legend()
    plt.grid(True)
    plt.show()

# Plot the depth-dose distribution for all z-bins (summarized result)
plt.figure(figsize=(12, 8))

# Plot the depth-dose distribution averaged across all z-bins

```



```

plt.plot(x_axis, integrated_dose_norm, color='blue')

# Add a horizontal line for 80% of the max dose
plt.axhline(y=r80_value, color='gray', linestyle='--', label='80% of Max.
Dose')

# Add a red dot at the R80 position
plt.plot(r80_x, r80_value, 'ro', label=f'R$_{{80}}$ = {r80_x:.2f} mm') #
Add label to legend

# Add labels and title
plt.xlabel('Depth [mm]')
plt.ylabel('Relative Dose [%]')
plt.title('Percentage depth dose curve for protons (N =  $10^{15}$ , E = 200
MeV, B = 0 T) in water')

plt.legend()
plt.grid(True)

# Show the depth-dose distribution plot
plt.show()

# Interpolation to find the dose at the calculated R80 value
r80_s = r80_x # Use the calculated R80 value
s_values = yshift_df_filtered['s (mm)']
dose_values = yshift_df_filtered['dose_norm']

# Create interpolation function
interp_function = interp1d(s_values, dose_values, bounds_error=False,
fill_value="extrapolate")

# Get interpolated dose at r80_s
interpolated_dose = interp_function(r80_s)

# Plot the dose vs distance 's' (mm)
plt.figure(figsize=(12, 8))
plt.plot(yshift_df_filtered['s (mm)'], yshift_df_filtered['dose_norm'])
plt.xlabel('Distance (s) [mm]')
plt.ylabel('Relative Dose [%]')
plt.grid(True)

# Plot the interpolated R80 point
plt.plot(r80_s, interpolated_dose, 'ro', label=f'Dose = {
interpolated_dose:.2f}% (R$_{{80}}$
= {r80_s:.2f} mm)')
#plt.plot(r80_s, interpolated_dose, 'ro', label=f'R$_{{80}}$ = {r80_s:.2f
} mm (Dose = {interpolated_dose:.2f
%})')

```

```
plt.title('Distance (s) vs. Relative Dose: Searching dose')
plt.legend()

# Show the plot
plt.show()

# Search for `s` value based on input relative dose
search_rel_dose = True # Toggle on or off

# Allow user to input the desired relative dose % (default: 83.69%)
target_rel_dose = 83.44 #interpolated_dose # ENTER HERE HOW MUCH r80 we
                        had for B = 0 T !!!

# Find the maximum dose and its corresponding index
max_dose_idx = np.argmax(yshift_df_filtered['dose_norm'])
max_dose_value = yshift_df_filtered['dose_norm'].iloc[max_dose_idx]

# Now, slice the data after the maximum dose
falling_side_df = yshift_df_filtered.iloc[max_dose_idx:] # Data after
                                                         reaching 100%

# Check if the target relative dose exists on the falling side
if target_rel_dose <= max_dose_value:
    # Interpolation function to find `s` for the given relative dose,
    # only after the peak

    interp_s_for_dose_falling =
    interp1d(falling_side_df['dose_norm'],
    falling_side_df['s (mm)'],
    kind='linear',
    bounds_error=False,
    fill_value="extrapolate")

    # Get the corresponding `s` value for the target relative dose (83.69
    %)
    target_s_value_falling = interp_s_for_dose_falling(target_rel_dose)

    # Plot the dose vs distance 's' without the R80 point, but with the
    # searched target dose point

    plt.figure(figsize=(12, 8))
    plt.plot(yshift_df_filtered['s (mm)'], yshift_df_filtered['dose_norm'],
             color='blue')

    # Plot the interpolated point at the target dose
    plt.plot(target_s_value_falling, target_rel_dose, 'ro', label=f's = {
        target_s_value_falling:.2f} mm
        for rel. dose = {target_rel_dose
        :.2f} %')

plt.xlabel('Distance (s) [mm]')
plt.ylabel('Relative Dose [%]')
```

```

plt.title(f'Distance (s) vs. Relative Dose: Searching distance')
plt.legend()
plt.grid(True)
plt.show()
else:
    print(f"Target relative dose {target_rel_dose}% exceeds the maximum
          dose.")

# Define the midpoint for the y-axis (example: 122.5 mm for B = 0 T)
y_midpoint = 122.5

# Find the two surrounding s-values for interpolation
s_values = yshift_df_filtered['s (mm)']
s_before_idx = np.where(s_values <= target_s_value_falling)[0][-1] #
                    closest s below target
s_after_idx = np.where(s_values >= target_s_value_falling)[0][0]   #
                    closest s above target

# Get corresponding x and y values
x1, x2 = yshift_df_filtered['x (mm)'].iloc[s_before_idx],
        yshift_df_filtered['x (mm)'].iloc[
            s_after_idx]
y1, y2 = yshift_df_filtered['y (mm)'].iloc[s_before_idx],
        yshift_df_filtered['y (mm)'].iloc[
            s_after_idx]
s1, s2 = s_values.iloc[s_before_idx], s_values.iloc[s_after_idx]

# Perform linear interpolation to find exact x and y for the
                    target_s_value_falling
interpolated_x = x1 + (target_s_value_falling - s1) * (x2 - x1) / (s2 -
                    s1)
interpolated_y = y1 + (target_s_value_falling - s1) * (y2 - y1) / (s2 -
                    s1)

# Check if y values need to be inverted based on the mirroring effect
if interpolated_y < y_midpoint:
    interpolated_y = 2 * y_midpoint - interpolated_y # Flip the y
                    position

print(f"Interpolated x = {interpolated_x:.2f} mm, y = {interpolated_y:.2f
        } mm for s = {target_s_value_falling
        :.2f} mm")

```

A.3 matRad

A.3.1 Treatment plan setup

```

%% Patient Data Import

matRad_rc; %If this throws an error, run it from the parent directory
           first to set the paths

load('2_04_P_CT_CST.mat');

%% Objective & Constraints

for i = 1:size(cst, 1)
    if strcmp(cst{i, 2}, 'Prostate')
        cst{i, 3} = 'TARGET';
        cst{i, 5}.Priority = 1;
        cst{i, 6}{1, 1}.className = 'DoseObjectives.
                                   matRad_MeanDose';
        cst{i, 6}{1, 1}.parameters{1,1} = 50;
        cst{i, 6}{1, 1}.penalty = 1;
    end
end

%% Treatment Plan

pln.radiationMode = 'protons';
pln.machine = 'Generic';
pln.propOpt.bioOptimization = 'const_RBExD';
pln.propDoseCalc.calcLET = 1;

pln.numOfFractions = 1;
pln.propStf.gantryAngles = [270];
pln.propStf.couchAngles = [0];
pln.propStf.bixelWidth = 5;
pln.propStf.numOfBeams = numel(pln.propStf.gantryAngles);
pln.propStf.isoCenter = ones(pln.propStf.numOfBeams,1) *
                        matRad_getIsoCenter(cst,ct,0);

csvwrite('isoCenter.csv', pln.propStf.isoCenter');
disp(['Iso Center positions have been saved as isoCenter.csv!']);

pln.propOpt.runDAO = 0;
pln.propOpt.runSequencing = 0;

% dose calculation settings
pln.propDoseCalc.doseGrid.resolution.x = ct.resolution.x; % [mm]
pln.propDoseCalc.doseGrid.resolution.y = ct.resolution.y; % [mm]

```

```

pln.propDoseCalc.doseGrid.resolution.z = ct.resolution.z; % [mm]

%% Generate Beam Geometry STF
stf = matRad_generateStf(ct,cst,pln);

%% Dose Calculation
% Lets generate dosimetric information by pre-computing dose
%                               influence
% matrices for unit beamlet intensities. Having dose influences
%                               available
% allows for subsequent inverse optimization.
dij = matRad_calcParticleDose(ct,stf,pln,cst);

%% Inverse Optimization for IMPT
% The goal of the fluence optimization is to find a set of bixel/spot
% weights which yield the best possible dose distribution according
%                               to the
% clinical objectives and constraints underlying the radiation
%                               treatment
resultGUI = matRad_fluenceOptimization(dij,cst,pln);

%% Plot the Resulting Dose Slice
% Let's plot the transversal iso-center dose slice
slice = round(pln.propStf.isoCenter(1,3)./ct.resolution.z);
figure
imagesc(resultGUI.RBExDose(:,:,slice)),colorbar,colormap(jet)

plane = 3;
doseWindow = [0 max([resultGUI.RBExDose(:);])];

figure,title('original plan')
matRad_plotSliceWrapper(gca,ct,cst,1,resultGUI.RBExDose,plane,slice,[
    ],0.75,colorcube,[],doseWindow,[
    ]);

```

A.3.2 Estimation of matRad proton stopping positions

```
% Initialization for storing x, y values, 80% dose values, and energy
x_y_energy_values_at_80_percent = [];

% Counter for the current bixel (across all rays)
current_bixel_id = 1;

% Loop through all rays
for ray_id = 1:length(stf.ray)
    % Get the x and y position of the current ray relative to the
                                isocenter
    ray_y_pos = stf.ray(ray_id).rayPos(2);

    % Get the isocenter
    isoCenter_y = stf.isoCenter(2);

    % Actual y position of the ray (relative to the isocenter)
    ray_y_absolute = isoCenter_y + ray_y_pos;

    % Number of bixels for this ray
    num_bixels = stf.numOfBixelsPerRay(ray_id);

    % Loop through all bixels in this ray
    for bixel_idx = 1:num_bixels
        % Step 1: Extract the 1D vector for the current bixel
        doseVector = full(dij.physicalDose{1,1}(:, current_bixel_id));
        LETVector = full(dij.mLETDose{1,1}(:, current_bixel_id));

        % Dimensions of the dose grid
        x_dim = dij.doseGrid.dimensions(2); % Number of x coordinates
        y_dim = dij.doseGrid.dimensions(1); % Number of y coordinates
        z_dim = dij.doseGrid.dimensions(3); % Number of z layers

        % Step 2: Convert the 1D vector into a 3D matrix
        doseMatrix = reshape(doseVector, [y_dim, x_dim, z_dim]);
        LETMatrix = reshape(LETVector, [y_dim, x_dim, z_dim]);

        % Step 3: Restrict to the second z layer
        doseMatrix_z2 = doseMatrix(:, :, 2); % Dose values in the second z
                                                layer
        LETMatrix_z2 = LETMatrix(:, :, 2); % LET values in the second z
                                                layer

        % Step 4: Sum the dose values along the y-axis for each x
                                                coordinate
        depthDoseCurve = sum(doseMatrix_z2, 1); % Sum over y for each x
                                                coordinate
        depthLETCurve = sum(LETMatrix_z2, 1); % Sum over y for each x
                                                coordinate
```

```

% Step 5: Normalize the dose values to the maximum value (
                                relative dose)
max_dose = max(depthDoseCurve);
relativeDepthDoseCurve = depthDoseCurve / max_dose;

max_LET = max(depthLETCurve);
relativeDepthLETCurve = depthLETCurve / max_LET;

% Step 6: Find the x value where the dose falls to 80% of the
                                maximum (after the peak)
dose_threshold = 0.8; % 80% of the maximum value
% Find the peak
[~, max_index] = max(relativeDepthDoseCurve);
% Find the first value after the peak that falls below 80%
idx_above_threshold = find(relativeDepthDoseCurve(max_index:end)
                                >= dose_threshold) +
                                max_index - 1;
idx_below_threshold = find(relativeDepthDoseCurve(max_index:end)
                                < dose_threshold, 1, 'first'
                                ) + max_index - 1;

% Interpolation between the two points
if ~isempty(idx_above_threshold) && ~isempty(idx_below_threshold)
    && idx_below_threshold >
    idx_above_threshold(end)
    x1 = idx_above_threshold(end); % Last index where dose >= 80%
    x2 = idx_below_threshold; % First index where dose < 80%
    D1 = relativeDepthDoseCurve(x1); % Dose at x1
    D2 = relativeDepthDoseCurve(x2); % Dose at x2

    % Linear interpolation for the exact x value where the dose
                                falls to 80%
    x_80_percent = x1 + (dose_threshold - D1) / (D2 - D1) * (x2 -
                                x1);
else
    x_80_percent = NaN; % If no suitable point is found
end

% Extract the energy of the current bixel from the current ray
energy = stf.ray(ray_id).energy(bixel_idx);

% Store the x value (at 80% dose), the corrected y value (Ray +
                                Isocenter), and the energy
x_y_energy_values_at_80_percent =
[x_y_energy_values_at_80_percent;
x_80_percent*dij.ctGrid.resolution.x, ray_y_absolute, energy];

% Increment the bixel counter
current_bixel_id = current_bixel_id + 1;

```

```
        end
    end

    % Define the filename
    filename = 'x_y_energy_values_at_80_percent.csv';

    % Save the matrix to a CSV file
    writematrix(x_y_energy_values_at_80_percent, filename);
```

A.3.3 Hounsfield look-up table

```
def matRad_conversion(HU):
    return np.interp(HU, [-1024, 200, 449, 2000, 2048, 3071],
                     [0.00324, 1.2, 1.20001, 2.49066, 2.5306, 2.53061])
```

A.4 Optimization

A.4.1 Computational gradient descent implementation

```
    % Load data and initialize variables as before
    data = readmatrix('x_y_energy_values_at_80_percent.csv');
    y_positions = data(:, 2); % Column 2 represents y-position
    energy_values = data(:, 3); % Column 3 represents energy values
    new_x_positions = zeros(size(y_positions));
    new_y_positions = zeros(size(y_positions)); % To store updated y-
                                                positions if adjusted
    new_energy_values = zeros(size(energy_values)); % To store updated
                                                energy values

    % Initialize arrays to store initial positions and adjusted starting
    positions
    adjusted_starting_x = zeros(size(y_positions));
    adjusted_starting_y = zeros(size(y_positions));
    adjusted_starting_energy = zeros(size(energy_values));

    % Initialize arrays to store the initial positions under the magnetic
    field (for verification)
    initial_x_positions_B = zeros(size(y_positions));
    initial_y_positions_B = zeros(size(y_positions));
    initial_energy_values_B = zeros(size(energy_values));

    magnetic_field = [0; 0; 1.5]; % Magnetic field in Tesla
    SPR = readmatrix('SPR_values.csv');
```



```

% Parameters for gradient descent
learning_rate_E = 0.0005; % Step size for energy
learning_rate_y = 0.0005; % Step size for position (if y is adjusted)
tolerance = 1e-4;          % Convergence tolerance
max_steps = 200;           % Maximum iterations

% Error threshold definitions
small_error_threshold = 0.5; % Reduce small difference threshold for
                             % finer control
big_error_threshold = 1.5;   % Reduce big difference threshold for finer
                             % control

% Loop over each row in the CSV data
for i = 1:length(y_positions)
    % Current y and energy
    y_position = y_positions(i) / 10.9375; % Divide y-position by pixel
                                           % factor
    E0_MeV = energy_values(i); % Energy value

    % Target without magnetic field
    Target = analyzeProtonTrajectory(y_position, E0_MeV, [0; 0; 0], SPR);

    % Initial position with magnetic field (Target_start)
    Target_start = analyzeProtonTrajectory(y_position, E0_MeV,
                                           magnetic_field, SPR);

    % Save the initial positions under magnetic field for verification (
    % before gradient descent)
    initial_x_positions_B(i) = Target_start(1) * 10; % Scale back to
    % original x-position
    initial_y_positions_B(i) = Target_start(2) * 10.9375; % Scale back
    % to original y-position
    initial_energy_values_B(i) = E0_MeV; % Initial energy before
    % gradient descent

    % Gradient Descent Loop
    Ei = E0_MeV; % This energy will be adjusted
    yi = y_position; % This y-position will be adjusted

    % Initialize variables to track the starting positions during the
    % gradient descent
    initial_yi = yi; % Keep track of the starting y_position
    initial_Ei = Ei; % Keep track of the starting energy

    for step_i = 1:max_steps
        % Compute the difference between the current position and target
        position_this_step = analyzeProtonTrajectory(yi, Ei,
                                                    magnetic_field, SPR);
        error_x = position_this_step(1) - Target(1);
        error_y = position_this_step(2) - Target(2);
    end
end

```

```

% Check for convergence
if abs(error_x) < tolerance && abs(error_y) < tolerance
    break;
end

% Adjust learning rates based on error size
if abs(error_x) >= big_error_threshold
    % Larger errors require larger steps, but within smaller
    % thresholds now
    dE = -error_x * EO_MeV * learning_rate_E;
    dy = error_x * y_position * learning_rate_y;
elseif abs(error_x) <= small_error_threshold
    % For small errors, fine-tune the adjustments
    dE = -error_x * EO_MeV * learning_rate_E * 0.5;
    dy = error_x * y_position * learning_rate_y * 0.5;
end

% Update energy and position using gradient descent
Ei = Ei + dE; % Update energy
yi = yi + dy; % Update y-position
end

% Save the final x-position, updated y-position, and energy after
% gradient descent
new_x_positions(i) = position_this_step(1) * 10; % Convert x to
% original scale
new_y_positions(i) = yi * 10.9375; % Convert back to original scale
% for saving
new_energy_values(i) = Ei; % Store the updated energy

% Calculate the difference in y-position, x-position, and energy
% between Target_start and final
% values
adjusted_starting_x(i) = Target_start(1) - position_this_step(1); %
% Difference in x
adjusted_starting_y(i) = Target_start(2) - yi; % Difference in y-
% position
adjusted_starting_energy(i) = EO_MeV - Ei; % Difference in energy
end

% Combine new x-positions, updated y-positions, and updated energy values
% into a matrix
new_data = [new_x_positions, new_y_positions, new_energy_values];
writematrix(new_data, 'x_y_energy_values_at_80_percent_gradient.csv');

% Save the initial starting positions before gradient descent under the
% magnetic field (Target_start) in a
% CSV file

```

```

initial_B_data = [initial_x_positions_B, initial_y_positions_B,
                  initial_energy_values_B];
writematrix(initial_B_data, 'x_y_energy_values_at_80_percent_gradient_B.
                           csv');

% Save the adjusted starting positions (differences in x, y, and energy
                                     to reach the target) in a CSV file
adjusted_starting_data = [adjusted_starting_x * 10, adjusted_starting_y *
                          10.9375, adjusted_starting_energy];
writematrix(adjusted_starting_data, 'x_y_new_starting.csv');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Function to Analyze Proton Trajectory %%%%%%%%%%
%%

function final_position = analyzeProtonTrajectory(y, E, B, SPR)
    % Constants (needed for velocity calculation for example)
    atomic_mass_unit_MeV_c2 = 931.494; % MeV/c^2
    c = 299792458 * 100; % [mm/s]

    grid_step = 0.00109375; % [m] This is the CT grid step /"PixelSpacing"
    total_distance = 83.91796875; % [mm] CT rows * CT columns * CT grid
                                   step

    % Pre-calculations for initial velocity
    gamma0 = 1 + E / atomic_mass_unit_MeV_c2;
    beta0 = sqrt(1 - 1/gamma0^2);
    v0 = beta0 * c;

    % Set initial conditions based on input parameters
    initial_position = [0; y; 0]; % Initial y-position
    initial_velocity = [v0; 0; 0]; % Initial velocity

    % Create an instance of ProtonSimulation
    protonSim = ProtonSimulation_half(SPR, E, B, grid_step,
                                      total_distance, initial_position
                                      , initial_velocity);

    % Initialize and run the simulation
    protonSim = protonSim.initializeStep();
    protonSim = protonSim.simulate();
    protonSim.saveResults();

    % Get the final position using the number of non-zero steps
    num_non_zero_steps = protonSim.displayStepRange();
    if num_non_zero_steps > 0
        final_position = protonSim.positions(:, num_non_zero_steps); %
                                                                    Extract the last valid
                                                                    position
    end

```

A Appendix

```
    else
        final_position = [0; 0; 0]; % Default if no valid positions found
    end
end
```

List of figures

2.1	Helical trajectory of a positively charged particle in a magnetic field with marked radius [hoffmann_proton_2015].	6
2.2	Illustration of radiotherapy workflow consisting of consultation, simulation, contouring, planning, delivery and follow-up stages [marvaso_virtual_2022].	8
2.3	Visualization of proton percentage depth dose curve example with marked characteristic Bragg peak as well as highlighted proton range, R_{80} , at 80 % distal fall-off of the Bragg peak [park_variation_2011].	9
2.4	Illustration of the matRad workflow consisting of six different steps.	10
2.5	A virtual radiation source (yellow) emits equidistant rays (solid black) to cover the target volume (red) within the patient (green). In the isocenter plane (not shown), the distance between beams equals the bixel width, representing discrete fluence elements (dashed black). The depth of the target is determined along each ray, and spots (black dots) are placed accordingly [bangert_dose_nodate].	11
2.6	Visualization of candidate HLUT conversion. Modified from [peters_consensus_2023].	13
3.1	Given prostate patient CT dataset imported into matRad and visualized using the matRad GUI. A candidate CTV, representing the prostate, is highlighted (red) as well as possible OARs, representing both femoral heads and the rectum (pink and dark red).	15
3.2	Visualization of the beam start position in TOPAS. In this example, a prostate patient CT dataset is irradiated with protons ($E = 100$ MeV, $N = 10^3$). Primary particles, i.e. protons are marked (blue lines), as well as secondary particles, i.e. electrons (red lines) and gamma rays (green lines).	21
3.3	Visualization of magnetic field influenced proton trajectory simulated in TOPAS (left). A candidate value of $x = 40$ mm is marked and the 1D dose distribution along y with Gaussian fit for specified x value is demonstrated (right).	22
3.4	Illustration of magnetic field influenced proton beam trajectory approximation. P_1 refers to the magnetic field influenced pencil beam.	23

3.5	Illustration of gradient descent optimization. P1 marks the magnetic field influenced pencil beam with proton stopping position $\vec{V}_{M,i}$. P2 refers to the pencil beam without a magnetic field with initial position \vec{r}_i and proton stopping position \vec{V}_i . After gradient descent optimization, the pencil beam $P_{op,1}$ stopping position $\vec{r}_{op,i}$ is marked.	24
4.1	Example proton trajectory in a vacuum under influence of a magnetic field with $E = 50$ MeV and $B_z = 1.5$ T resulting from MATLAB simulation. Highlighted position (red dot) marks the diameter of the circular trajectory.	26
4.2	Resulting proton beam trajectories from the developed algorithm and MC simulations (TOPAS) in water for $B_z = 0$ T and $E = 200$ MeV. The white line shows the trajectory calculated in the developed algorithm, while the heatmap dose distribution refers to the MC simulated proton beam trajectory.	27
4.3	Resulting percentage depth dose curve with marked proton range, R_{80} , from a MC simulation (TOPAS) in water for $E = 200$ MeV and $B = 0$ T with $N = 10^5$ protons.	28
4.4	Percentage depth dose curve and marked R_{80} range with the corresponding relative absorbed dose value from a MC simulation (TOPAS) in water with $B_z = 0$ T and $E = 200$ MeV with $N = 10^5$ protons. s_{L1} refers to the proton trajectory length in a magnetic field and s_{L2} refers to the proton trajectory length without a magnetic field. . . .	29
4.5	Percentage depth dose curve and marked $s = s_{L1} = s_{L2}$ value with the corresponding relative absorbed dose value from a MC simulation (TOPAS) in water with $B_z = 1.5$ T and $E = 200$ MeV with $N = 10^5$ protons. s_{L1} refers to the proton trajectory length in a magnetic field and s_{L2} refers to the proton trajectory length without a magnetic field.	30
4.6	Visualization of proton beam trajectories from MC simulations (TOPAS) with an introduced magnetic field ($B_z = 1.5$ T) and two candidate energy values ($E = 150$ MeV and $E = 200$ MeV) for the prostate patient CT dataset. The proton stopping positions are highlighted for $E = 150$ MeV at $\vec{V}_{150\text{ MeV}} = (152.73, 134.26)$ mm (light green dot) and for $E = 200$ MeV at $\vec{V}_{200\text{ MeV}} = (254.33, 150.72)$ mm (cyan dot), where $\vec{V} = (x, y)$	38
4.7	Dose distribution for given prostate CT dataset with objective of achieving a mean dose of 50 Gy. A candidate CTV (prostate) is highlighted in red, along with potential OARs (femoral heads and rectum) in pink and dark red.	39

4.8	Visualization of a simplified contour representing only the body contour of given prostate patient CT dataset. The CTV is highlighted (white) as well as the proton stopping positions for each pencil beam retrieved from the "dij" matrix after matRad irradiation (blue dots).	40
4.9	Visualization of the enlarged prostate patient CT dataset with highlighted CTV (white). Qualitative comparison between proton stopping positions for each pencil beam calculated via the developed proton transfer algorithm (orange dots) and matRad (blue dots) using the same initial positions and energy values.	41
4.10	Histogram for quantitatively analyzing difference between stopping positions calculated via the developed proton transfer algorithm ($B_z = 0$ T) and matRad extracted stopping positions using initial matRad starting positions and energy values. Median (red) and mean (green) values are highlighted.	42
4.11	Visualization of the enlarged prostate patient CT dataset with highlighted CTV (white). Qualitative comparison between stopping positions calculated using the developed proton transfer algorithm (orange dots) and extracted from matRad (blue dots) using the same initial positions and energy values. A magnetic field ($B_z = 1.5$ T) is introduced in the developed proton transfer algorithm.	43
4.12	Histogram for quantitatively analyzing difference between stopping positions calculated using the developed proton transfer algorithm and stopping positions extracted from matRad using initial matRad starting positions and energy values. A magnetic field ($B_z = 1.5$ T) is introduced in the developed proton transfer algorithm. Median (red) and mean (green) values are highlighted.	44
4.13	Visualization of the enlarged prostate patient CT dataset with highlighted CTV (white). Qualitative comparison between magnetic field influenced and optimized spots calculated using the developed proton transfer algorithm (red dots) and matRad extracted proton stopping positions (blue dots) for each pencil beam. A candidate spot without optimization (orange dot) with its updated position is shown (orange arrow).	45

4.14	Histogram for quantitatively analyzing the differences between optimized stopping positions calculated using the developed proton transfer algorithm with an introduced magnetic field ($B_z = 1.5$ T) and proton stopping positions extracted from matRad (blue bars). For comparison, initial stopping positions calculated using the developed proton transfer algorithm without a magnetic field are shown (gray bars). Euclidean distances are used, with their frequency distribution displayed. Median (red) and mean (green) values for both datasets are highlighted.	46
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

List of tables

4.1	Comparison of resulting radii calculated using the developed proton transfer algorithm (r_{ana}) and analytical equation for the helical radius (r_{eq}) under influence of a magnetic field (B_z) in a vacuum state. The first table has a fixed magnetic field strength with four different energy values, while the second table shows the results for a fixed energy value and four different magnetic field strengths. Absolute and relative differences between those radii are also listed.	26
4.2	Comparison of proton ranges calculated analytically using the developed proton transfer algorithm (r_{ana}) and proton ranges ranges extracted from MC simulation (TOPAS) depth dose curves (r_{MC}) for a water phantom without a magnetic field. Four different energy values are analyzed and the resulting relative and absolute differences between the two ranges can be seen.	28
4.3	Relative dose values calculated using MC simulations (TOPAS) in a water phantom with $B_z = 0$ T and four different energy values with prior determined proton ranges and trajectory lengths.	29
4.4	Comparison of trajectory lengths calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for a water phantom. Different energy values and magnetic field strengths are analyzed with their absolute and relative differences listed.	31
4.5	Comparison of end positions calculated analytically using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for a water phantom with $\vec{V} = (x, y)$. Different energy values and magnetic field strengths are analyzed and their vectorial absolute differences ($\Delta\vec{V}$) are listed. Additionally, the Euclidean distances ($ \vec{V} = \sqrt{(x_{\text{MC}} - x_{\text{ana}})^2 + (y_{\text{MC}} - y_{\text{ana}})^2}$) are shown.	31
4.6	Comparison of proton ranges calculated analytically using the developed proton transfer algorithm (r_{ana}) and proton ranges ranges extracted from MC simulation (TOPAS) depth dose curves (r_{MC}) for a bone phantom without a magnetic field. Four different energy values are analyzed and the resulting relative and absolute differences between the two ranges can be seen.	32

4.7	Relative dose values calculated using MC simulations (TOPAS) in a bone phantom with $B_z = 0$ T and four different energy values with prior determined proton ranges and trajectory lengths.	33
4.8	Comparison of trajectory lengths calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for a bone phantom. Different energy values and magnetic field strengths are analyzed with their absolute and relative differences listed.	33
4.9	Comparison of end positions calculated analytically using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for a bone phantom with $\vec{V} = (x, y)$ assuming a constant CT number of 1000 HU. Different energy values and magnetic field strengths are analyzed and their vectorial absolute differences ($\Delta\vec{V}$) are listed. Additionally, the Euclidean distances ($ \vec{V} = \sqrt{(x_{\text{MC}} - x_{\text{ana}})^2 + (y_{\text{MC}} - y_{\text{ana}})^2}$) are shown.	34
4.10	Comparison of proton ranges calculated analytically using the developed proton transfer algorithm (r_{ana}) and proton ranges ranges extracted from MC simulation (TOPAS) depth dose curves (r_{MC}) for given prostate patient CT dataset without a magnetic field. Four different energy values are analyzed and the resulting relative and absolute differences between the two ranges can be seen.	35
4.11	Relative dose values calculated using MC simulations (TOPAS) in given prostate patient CT dataset with $B_z = 0$ T and four different energy values with prior determined proton ranges and trajectory lengths.	35
4.12	Comparison of trajectory lengths calculated analytically in the developed proton transfer algorithm (s_{ana}) and with MC simulations (s_{MC}) for given prostate patient CT dataset. Different energy values and magnetic field strengths are analyzed with their absolute and relative differences listed.	36
4.13	Comparison of end positions calculated analytically using the developed proton transfer algorithm (\vec{V}_{ana}) and using MC simulations (\vec{V}_{MC}) for given prostate patient CT dataset with $\vec{V} = (x, y)$. Different energy values and magnetic field strengths are analyzed and their vectorial absolute differences ($\Delta\vec{V}$) are listed. Additionally, the Euclidean distances ($ \vec{V} = \sqrt{(x_{\text{MC}} - x_{\text{ana}})^2 + (y_{\text{MC}} - y_{\text{ana}})^2}$) are shown.	37

Acknowledgments

I would like to express my deepest gratitude to Prof. Dr. Armin Lühr for his guidance, insightful discussions, and the opportunity to conduct my Master's thesis within his research group.

I extend my sincere thanks to Prof. Dr. Kevin Kröninger for kindly agreeing to serve as the second referee for my thesis.

I am deeply grateful to the entire Lühr research group for their invaluable feedback, support, and collaborative spirit. I would like to extend special thanks to Cihangir, Dennis, Lisa, Paulin, Robin and Roman for their dedicated assistance throughout the course of my work.

Finally, I would like to thank my family and friends for their unwavering emotional support during challenging times.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this thesis, the author utilized the GPT-4 model and DeepL to address grammatical and spelling errors, as well as to ensure linguistic consistency and coherence. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

December 2, 2024

Yosef Emshani

Eidesstattliche Versicherung

(Affidavit)

Emshani, Yosef

Name, Vorname
(surname, first name)

217564

Matrikelnummer
(student ID number)

☐ Bachelorarbeit
(Bachelor's thesis)

☒ Masterarbeit
(Master's thesis)

Titel
(Title)

Development of magnetic field influenced pencil beam spot selection
for proton therapy treatment planning

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem oben genannten Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present thesis with the above-mentioned title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution before.

Dortmund, 2.12.2024

Ort, Datum
(place, date)

Unterschrift
(signature)

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -):

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:

Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to EUR 50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the Chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, Section 63 (5) North Rhine-Westphalia Higher Education Act (*Hochschulgesetz, HG*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund University will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:*

Dortmund, 2.12.2024

Ort, Datum
(place, date)

Unterschrift
(signature)

***Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**