# Dissertation

submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany,
for the degree of
Doctor of Natural Sciences

Put forward by

Dipl.-Phys. Martin Siggel
Born in Neuruppin

Oral examination: May 23, 2012

# Concepts for the efficient Monte Carlo-based treatment plan optimization in radiotherapy

Referees: Prof. Dr. Uwe Oelfke
Prof. Dr. Peter Bachert

## Konzepte für die effiziente Monte Carlo-basierte Optimierung von Bestrahlungsplänen in der Radiotherapie

Monte Carlo (MC) Algorithmen zur Dosisberechnung gelten als Goldstandard in der intensitätsmodulierten Radiotherapie (IMRT). Das einfache Hinzufügen von MC Dosisberechnungen in gängige Systeme zur Optimierung von Bestrahlungsplänen ist zwar möglich, jedoch zu ineffizient und resultiert in zu langen Rechenzeiten für eine klinische Verwendung. In dieser Arbeit wurde ein hybrider Optimierungsalgorithmus entwickelt, welcher die Genauigkeit von MC Simulationen mit der Effizienz von weniger genauen Algorithmen zur Dosisberechnung verbindet. Wir präsentieren zwei Methoden, die eine schnelle Konvergenz des iterativen Optimierungsverfahrens erlauben und die Effizienz der MC Dosisberechnung erhalten. Die Funktionsweise des hybriden Optimierungsalgorithmus wurde an verschiedenen Körperregionen demonstriert. Die daraus resultierenden Bestrahlungspläne werden gegen die Ergebnisse eines Referenzalgorithmus verglichen, welcher auf MC Simulationen im gängigen IMRT Framework basiert. Neben verschiedenen Indikatoren zur Qualität der Bestrahlungspläne wurden Konvergenzeigenschaften, Rechenzeiten und Effizienzen für diesen Vergleich ausgewertet. Die Effizienz der Optimierung konnte mit dem neuen Algorithmus von urspünglich 10–30 % auf 80–95 % gesteigert werden. Aufgrund dieser Steigerung konnten wir – je nach Bestrahlungsplan – die Rechenzeiten auf 2 bis 28 Minuten verkürzen. Dabei konnte im Vergleich zum Referenzalgorithmus die Qualität der Bestrahlungspläne beibehalten werden.

## Concepts for the efficient Monte Carlo-based treatment plan optimization in radiotherapy

Monte Carlo (MC) dose calculation algorithms are regarded as the gold standard in intensity-modulated radiation therapy (IMRT). Simply adding a MC dose calculation engine to a standard IMRT optimization framework is possible but computationally inefficient. Thus, the optimization would be too time consuming for clinical practice. In this work we developed a hybrid algorithm for the treatment plan optimization that combines the accuracy of MC simulations with the efficiency of less precise dose calculation algorithms. Two methods are introduced that allow a rapid convergence of the iterative optimization algorithm and preserve the efficiency of the MC dose calculation. The performance of the hybrid optimization algorithm is analyzed on different treatment sites. The results are compared against a reference optimization algorithm, which is based on MC simulations in the standard IMRT framework. For this comparison we evaluated several indicators of treatment plan quality, convergence properties, calculation times and efficiency ratios. The efficiency of the optimization could be improved from originally 10–30 % to 80–95 %. Due to this improvement the calculation times could be reduced to 2–28 minutes, depending on the treatment plan complexity. At the same time, the treatment plan quality could be maintained compared to the reference algorithm.

# Contents

# 1 Introduction

Radiation therapy is one of the three major columns for the treatment of localized cancer. Depending on the type and progression of the tumor, it can be applied on its own or in combination with chemotherapy or surgery. The aim of radiation therapy is to prevent a further tumor growth by causing severe damage to the DNA of tumor cells.

Radiation therapy is the use of ionizing radiation for medical purposes. The first appliance of X-rays for cancer treatment is dated back to end the of the 19th century. The breakthrough of clinical radiation therapy was marked by the discovery of radioactivity. Even nowadays, cobalt-60 sources are used in some treatment devices for the generation of high energetic photons. In current state of the art devices however, the radioactive source is replaced with a linear accelerator – also called linac – in which electrons are accelerated to high energies in the range of several MeV. Due to a deceleration of these electrons in a tungsten target (or other high-Z materials), a spatially broad photon beam is created for the therapeutic appliance. These high energy photons transfer some of their energy to electrons in the patient, which then interact with the tissue. This interaction is responsible for a cascade of biological effects, including the controlled cell death (apoptosis). In radiotherapy, the physical effect of the irradiation of a volume is specified with the irradiation dose. It is defined as the absorbed energy in the volume divided by its mass.

## 1.1 The treatment planning process

In clinical practice, a specific irradiation dose is prescribed to a target volume. According to the ICRU report 50 (ICRU 1993), this target volume comprises the solid tumor (gross tumor volume/GTV), the surrounding tissue with the microscopic spread of cancer cells (tumor + spread = clinical target volume/CTV) and a safety margin to account for organ motion and setup uncertainties of the patient (tumor + spread + margin = planning target volume/PTV). The physical nature of photons and charged particles (e.g. electrons, protons, heavy ions) and their interaction with matter prevent an irradiation of the tumor only. Hence, the art of treatment planning is to find a trade-off between a high homogeneous target dose while sparing dose in normal tissue as much as possible, especially in organs at risk (OAR). Traditionally, the treatment plan creation was a complicated manual and time consuming process. For each treatment plan a number of decisions have to be made:

- Decision about the number of incident beams and their directions as their superposition can be exploited to reduce high doses in healthy tissue and to shape the high dose region (Mackie et al. 1993).
- Decision about the geometrical field shape of each beam to achieve a target-conformal dose distribution. The irradiation field can be adjusted to match e.g. the outline of the tumor using a multileaf collimator (MLC).
- Decision about the relative weight (monitor units or irradiation time) of each beam.
- Decision about tolerated doses in organs at risk and minimum and maximum doses for the target volume.

Hence, treatment planning was and still is a trial and error process. For a given set of treatment parameters, a three-dimensional dose distribution is calculated by a computer program. These parameters have to be adjusted repeatedly until the dose distribution satisfies the clinical requirements. This manual process of treatment plan creation is often referred to as forward planning (see figure 1.2(a)).

## 1.2 IMRT and inverse planning

The intensity-modulated radiation therapy (IMRT) was invented in the early 1980s (Brahme et al. 1982). Due to the great advantages of IMRT compared to conventional

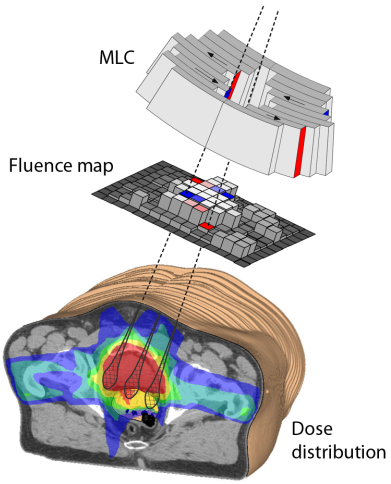
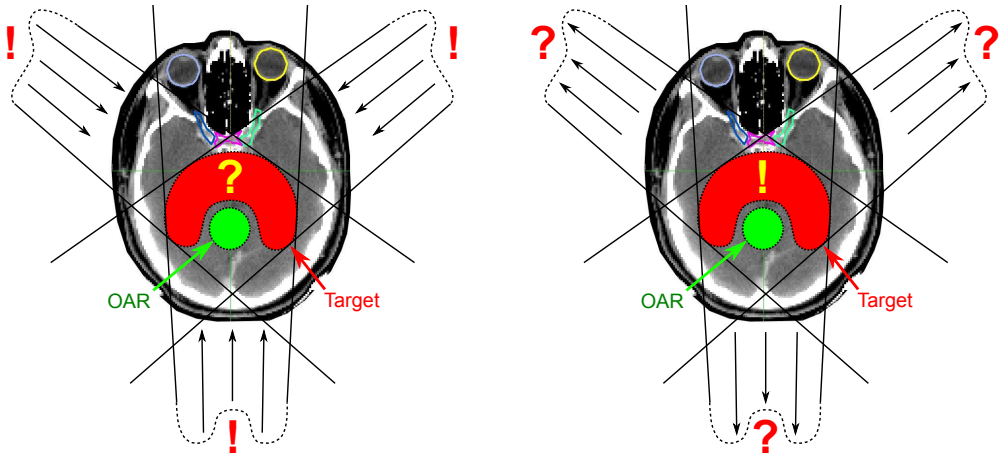



**Figure 1.1:** Scheme of the inverse planning principles. The desired 2-dimensional fluence map is created by irradiating a sequence of different fields shaped with a multileaf collimator (MLC). An appropriate modulation of the fluence results in a target conformal dose distribution. Image taken from Hårdemark et al. (2003, RaySearch white paper), RaySearch Laboratories AB, Copyright © 2003

radiotherapy and not least due to the increasing computing power, IMRT is nowadays widely used in clinical practice (Bortfeld 2006). IMRT extends 3D-conformal radiotherapy by additionally modulating the intensity of each treatment field. The modulation of the photon fluence is realized with a MLC in practice by either irradiating successively differently shaped fields (step and shoot) or by dynamically adapting the velocity of each leaf (dMLC). This fluence modulation process with an MLC is illustrated in figure 1.1. An alternative to a MLC for the fluence modulation are compensators, which are blocks of absorbing material (e.g. brass or aluminum) with varying thickness. IMRT incorporates many advantages in comparison to conformal radiotherapy: most importantly, IMRT allows the creation of dose distributions with concave target conformity (Brahme 1988), for example for the treatment of horse shoe shaped targets like the intrathoracic lymphatic system (Németh & Schlegel 1987) or the prostate.

The challenge in creating good IMRT treatment plans is to determine the fluence modulation for each beam, as the number of free parameters is enormous. The solution to avoid the cumbersome and time consuming manual treatment plan creation is the algorithmic optimization of the fluence patterns. This so-called inverse planning, which was first described by Webb (1989), simplifies the manual iterative decision making process described above. In inverse planning, the treatment planner derives organ doses constraints from the medical requirements. A fluence map optimization (FMO) algorithm then tries to find the fluence map that matches these requirements best. To take account for the finite width of the MLC leafs and to reduce algorithmic complexity, the




(a) Forward planning (b) Inverse planning

**Figure 1.2:** Difference between forward and inverse treatment planning. In forward planning, the beam parameters as e.g. the fluence maps are defined by the treatment planner. A dose calculation algorithm then determines the corresponding dose distribution. In inverse planning, dose constraints like the target dose and tolerated doses in OARs are defined by the planner. An algorithm tries to find the fluence map, that matches these constraints best.

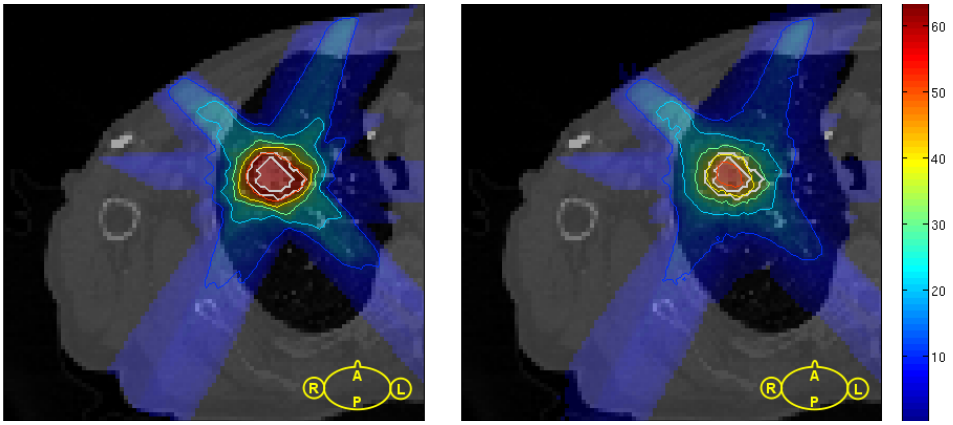
two-dimensional fluence map of each beam is discretized into small quadratic sub-beams (illustrated in figure 1.1). In the following these sub-beams will be called beamlets.

During optimization, the plan quality is estimated by an objective function, which converts the differences between a given dose distribution and the prescribed dose constraints into a single value (Bortfeld et al. 1990, Spirou & Chui 1998). Then, the FMO algorithm searches for the fluence weight configuration that minimizes this function. With the tool of algorithmic optimization, the process of creating treatment plans is considerably simplified and it is reduced to a selection of incident beam directions, beam energies, the definition of the dose constraints and additional penalty factors, which penalize the violation of the dose constraints. For a "good" settings of these penalty factors and dose constraints, this mathematically optimal treatment plan resembles an acceptable tradeoff between high tumor conformity and sparing organs at risk. Still, the finding of this good set of penalty factors etc. is one crucial part of inverse planning and requires much experience. The conceptual difference between the forward and inverse planning process is depicted in figure 1.2.

Technically, each iteration of the optimization consists of two parts: calculating the dose distribution of a given fluence map and evaluating its corresponding objective function value. As each dose calculation algorithm incorporates however a systematic error, the planned/optimized dose distribution and the actually delivered dose to the patient differ. In particular at treatment sites with strong tissue heterogeneities, the relative error of established dose calculation algorithms can exceed 20% (Scholz et al. 2003, Krieger & Sauer 2005). In order to get reliable treatment plans for these body sites, more accurate dose calculation algorithms have to used for inverse planning.

## 1.3 The need of Monte Carlo algorithms in inverse planning

The dose calculation with Monte Carlo (MC) simulations is considered one of the most accurate techniques today (Chetty et al. 2007). Particle transport and scattering in the patient and at the treatment head are accurately handled. Especially in low density tissue like in the lung, traditional convolution based methods cannot achieve the accuracy of MC simulations (Scholz et al. 2003). It is known that for example pencil beam algorithms significantly overestimate doses in the lung. As a consequence, lung tumors would be severely underdosed with an optimized treatment plan that is based on pencil beams. An example of the clinical effect of an inaccurate dose calculation algorithm is shown in figure 1.3, which compares a planned dose distribution with the actually delivered dose of a lung treatment plan. In these cases, the inclusion of MC dose calculation algorithms into the optimization process is highly desirable.




(a) Planned dose (b) Delivered dose

**Figure 1.3:** Example treatment plan of a lung tumor: the figure on the left shows an optimized dose distribution based on the inaccurate pencil beam dose calculation method. The dose recalculation by Monte Carlo simulation (b) reveals a significant underdosage of the tumor of about 20 %.

## 1.4 Problems of the Monte Carlo-based inverse planning

Since MC methods are stochastic by nature, many particles (photons will be considered as "particles" from now on) have to be simulated in order to achieve a low statistical uncertainty of the dose distribution. The computation of a single dose distribution with multi-purpose MC frameworks, such as EGSnrc (Kawrakow & Rogers 2001) or Geant4 (GEANT4 Collaboration 2003), can take several hours. To achieve clinically acceptable dose calculation times, modern MC codes were developed in the last decade that utilize a range of variance reduction techniques, such as XVMC (Kawrakow 1996), VMC$^{++}$ (Kawrakow & Fippel 2000) and DPM (Sempau et al. 2000).

Fluence map optimization can be formulated as a convex optimization problem (Bortfeld et al. 1990). Therefore, gradient based optimization algorithms are commonly used (Bortfeld et al. 1990, Spirou & Chui 1998) as they usually converge faster than stochastic optimization methods. In order to calculate the gradient of the objective function, the dose contribution of each beamlet has to be known und thus requires a dose calculation for each beamlet. In the case of MC algorithms, this calculation can be very time consuming (Jeraj & Keall 1999) or require computing clusters for reasonable calculation times (Bergman et al. 2006) and require the simulation of a high number of particles. In many cases, the final fluence map contains elements with a zero weight, that is, the irradiation will be blocked from those beamlets. This is often the case if an organ at risk is placed before or behind the target inside the treatment beam. As a consequence, simulated particles from these beamlets do not contribute to the dose distribution of the optimized treatment plan. Therefore, many particles of the original dose calculation are

wasted and the computation takes more time than necessary. It should be noted that a waste of particle does not only originate from closed beamlets but also from beamlets with a small fluence, as the number of simulated particles is higher than required.

In order to decrease computation time, waste of particles has to be avoided. The aim of this work was to develop an optimization algorithm that increases the efficiency of the treatment plan optimization by exploring the search space of the optimization problem without simulating additional particles. Because a large number of patients are treated each day in clinical practice, long calculation times of several hours are noneconomical. For that reason, the ultimate goal of this work was to achieve clinically acceptable times for the total MC-based treatment plan optimization of only a few minutes in combination with the fast VMC$^{++}$ package.

# 2 Methods

## 2.1 Algorithmic Optimization

Optimization is part of our everyday life, even if we are not aware of it. Our navigation systems find the shortest or the fastest route to our destination. Time schedules and routes of bus lines are set in order to minimize delays and maximize the number of transported people. The shape of our cars is tuned to reduce drag and fuel consumption. Many systems on the financial market are built to maximize profit or minimize risk. Even the basic physical principles can be understood as an optimization problem as each (physical) system moves to the state of its lowest total energy. These are only a few examples of an endless list.

Mathematically speaking, optimization is a minimization (or maximization) of a function. To optimize a real world problem, it has to be converted into a function first. In navigation systems this function may be the traveling time, the length of the route or even the expected fuel consumption. This so-called objective function depends on the free parameters of the system. In our example the parameters are the roads and their order that we should drive to get to our destination. However, the conversion of the problem into an objective function is not always as obvious as it seems. Often, different aspects have to be combined and trade-offs have to be made. Therefore, the quality of the result of the optimization depends to a great extend on this modelling process.

Let $f(\vec{x})$ be the objective function of $n$ free parameters that are combined in the vector $\vec{x} \in \mathbb{R}^n$. Then, a general optimization problem can be formulated as follows (Nocedal & Wright 1999):

$$\min_{\vec{x}} f(\vec{x}) \quad \text{subject to} \begin{cases} c_i(\vec{x}) \geq 0,\ i \in I \\ c_j(\vec{x}) = 0,\ j \in E \end{cases} \tag{2.1}$$

Here the functions $c_i(\vec{x})$ are called constraints, which define the space in which the function $f$ should be minimized. We call this space the feasible region. It can be shown that any optimization problem can be transformed into standard form (2.1). For example maximization problems, which often occur in the financial business, can be transformed into the standard form by simply multiplying their objective function with a negative number.

Optimization problems can be divided into two classes: convex and global optimization problems. In convex problems, the objective function has only one local minimum that

resembles the optimal solution. All algorithms for these kinds of problems use basically the same idea: they start at some initial point $\vec{x}_0$: From there they choose a search direction that points downhill i.e. they go to a new point $\vec{x}_1$ in order that $f(\vec{x}_1) < f(\vec{x}_0)$. This strategy is continued, until the minimum is reached. On unconstrained problems this minimum is reached, when the gradient of the objective function is zero $\vec{\nabla} f(\vec{x}) = 0$. Some of the typical convex optimization algorithms are the steepest descend method, the conjugate gradients, Newtons methods and quasi-Newton algorithms. The objective function of global problems usually has a large number of local minima. If we applied a convex optimization algorithm on a global problem, the optimization would end up in a local minimum. To avoid this, algorithms for global optimization try to explore the complete solution space in a stochastic fashion. Typical algorithms for global optimization are genetic algorithms, simulated annealing and Monte Carlo sampling.

In radiation therapy, the optimization of the fluence maps can be modeled as a convex problem (Bortfeld et al. 1990). Therefore only optimization algorithms for convex problems will be further discussed.

A function $f : S \subset \mathbb{R}^n \to \mathbb{R}$ is convex on a convex set $S$ if for any two arbitrary points $\vec{x}_1, \vec{x}_2 \in S$ the function continues below a line spanning from $(\vec{x}_1, f(\vec{x}_1))$ to $(\vec{x}_2, f(\vec{x}_2))$. This requirement can be mathematically expressed with

$$f\left(\alpha\vec{x}_1 + [1-\alpha]\,\vec{x}_2\right) \leq \alpha f(\vec{x}_1) + (1-\alpha) f(\vec{x}_2), \quad \forall\alpha \in [0,1]. \tag{2.2}$$

It can be shown that this expression holds if and only if the Hessian matrix $H(\vec{x}) := \vec{\nabla}^2_{xx} f(\vec{x})$ of a twice differentiable function $f(\vec{x})$ is positive-definite at each point $\vec{x} \in S$. That is, if

$$\vec{p}^{\top} H(\vec{x}) \vec{p} > 0, \quad \forall \vec{p} \in \mathbb{R}^n \neq 0. \tag{2.3}$$

### 2.1.1 The steepest descent method

Most of the convex optimization methods are iterative methods. In every iteration $k$ a direction $\vec{p}_k$ is calculated at the current solution $\vec{x}_k$ that defines the new path to explore. Of course, in the function space this search direction $\vec{p}_k$ has to point in a direction where $f$ is decreasing.

The most basic strategy is to choose the direction of the steepest descent. It can be shown that this direction is given by the negative gradient $\vec{p} = -\vec{\nabla} f(\vec{x})$ at the current iterate $\vec{x}$. Depending on the complexity of the optimization problem, this gradient can be calculated either analytically, by finite differences or with automatic differentiation. Giving this steepest descend direction, the next iterate $\vec{x}_{k+1}$ is determined by

$$\vec{x}_{k+1} = \vec{x}_k - \alpha_k \vec{\nabla}_x f(\vec{x}_k), \tag{2.4}$$

where the parameter $\alpha_k$ is called the step-length. As a rule, this step-length $\alpha_k$ is found

by minimizing the function along the search direction $\vec{p}_k$, i.e.

$$\min_{\alpha_k} f(\vec{x}_k + \alpha_k \vec{p}_k). \tag{2.5}$$

This one-dimensional optimization is relatively easy to solve. It turns out that the solution does not have to be exact. An approximate solution suffices for most of the optimization algorithms. More details for efficient line search strategies can be found in section 2.1.4.

In practice, optimization algorithms will not exactly reach the optimal solution but they will converge against it. Therefore, the algorithm is usually terminated if the norm of the gradient is small enough, i.e. if $\|\vec{\nabla}_x f(\vec{x}_k)\|_2 < \epsilon$.

### 2.1.2 Newton's method in optimization

For most optimization problems, the steepest descend method converges very slowly and requires a large number of iterations. The reason is that it completely ignores the curvature of the objective function (Nocedal & Wright 1999). Figure 2.1(a) illustrates the convergence of the steepest descent algorithm on a quadratic function. The zig-zag pattern is typical for the steepest descend method.

The idea of Newton's method is to use the curvature information to calculate a better search direction. At the current iterate $\vec{x}_k$ a local quadratic model of the objective function $f_{qp}(\vec{x})$ is created by a second order Taylor expansion:

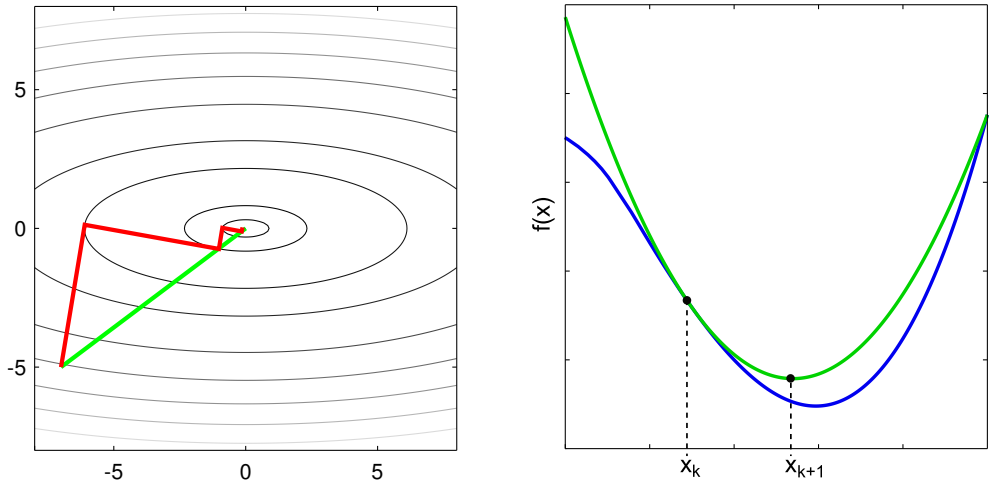
$$f_k^{\text{qp}}(\vec{p}) := f(\vec{x}_k) + \vec{\nabla}_x f(\vec{x}_k)^\top \vec{p} + \vec{p}^\top \nabla^2_{xx} f(\vec{x}_k) \vec{p} \tag{2.6}$$

The vector $\vec{p}_k$, which minimizes the quadratic model, will be the new search direction. Fortunately, the optimization of a quadratic model can be analytically solved. Calculating the derivative of (2.6) and setting it to zero leads to the solution:

$$\vec{\nabla}_p f_k^{\text{qp}}(\vec{p}_k) = \vec{\nabla}_x f(\vec{x}_k) + \nabla^2_{xx} f(\vec{x}_k) \vec{p}_k \stackrel{!}{=} 0 \tag{2.7}$$

$$\Rightarrow \vec{p}_k = -\left[\nabla^2_{xx} f(\vec{x}_k)\right]^{-1} \vec{\nabla}_x f(\vec{x}_k) \tag{2.8}$$

The process of finding Newton's direction is depicted in figure 2.1(b) on an one-dimensional problem. At the current iterate $\vec{x}_k$ a quadratic model (green) approximates the objective function (blue). The position of the minimum of the parabola determines the next iterate $\vec{x}_{k+1}$. If we look back at figure 2.1(a), it can be seen that contrary to the steepest descent method, the newton-step instantaneously solves the problem since the objective function is already quadratic. If the objective function $f(\vec{x})$ is continuously differentiable and its gradient at the starting point $\vec{\nabla}_x f \vec{x}_0) \neq 0$, the convergence of Newton's method or Newton based methods is quadratic (Nocedal & Wright 1999). Practically this means that in each iteration of the algorithm the number of digits of the current objective function value $f_k$, that match the digits of global minimum $f^*$,




(a) (b)

**Figure 2.1:** Illustration of Newton's method in optimization: Figure (a) compares the convergence of the steepest descent method (red line) with Newton's method (green line) on the quadratic ellipsoid $f(x, y) = \frac{1}{2}x^2 + 2y^2$. The optimization starts at $(-7, -5)$. In this example, the steepest-descent method theoretically requires an infinite number of iterations to reach the optimum at $(0, 0)$. Practically after 5 iterations the solution is found. In contrast, the newton step instantaneously leads to the exact solution. (b) illustrates how the next iterate is found in Newton's method: at the current iterate $\vec{x}_k$, a quadratic model (green) of the objective function (blue) is built. The position of its minimum defines the next iterate $\vec{x}_{k+1}$.

doubles. However, if these conditions are not met, convergence to the minimum cannot be guaranteed.

It should be noted, that it is not necessary to calculate the inverse of the Hessian matrix $\nabla^2_{xx} f^{-1}$. Instead, it is much more efficient to solve the linear equation $\nabla^2_{xx} f(\vec{x}_k)\vec{p}_k = -\vec{\nabla}_x f(\vec{x}_k)$ with the LU-decomposition or the conjugate gradient method (Hestenes & Stiefel 1952). Still, Newton's method has the disadvantage that the Hessian matrix has to be calculated. While this may be an option for low-dimensional optimization problems, the full computation of $\nabla^2_{xx} f$ might be too expensive in terms of memory requirements and calculation time on problems with a high dimension number.

### 2.1.3 Quasi-Newton methods: BFGS and limited-memory BFGS

Although Newton's method has excellent convergence properties, it also implies important disadvantages when it comes to high-dimensional real-life problems. Often, the objective function is not differentiable twice in the complete feasible domain $S$. Also,

an analytical representation of its second-derivatives is not always available. On $n$-dimensional problems, the calculation of the Hessian $\nabla^2_{xx} f$ with finite differences takes about $n^2$ evaluations of the objective function. This calculation is much too expensive if the dimensionality is high.

Quasi-Newton methods create an approximation of the Hessian matrix without explicitly calculating second derivatives. Only information of the iterates $\vec{x}_i$ and their gradients $\vec{\nabla}_x f(\vec{x}_i)$ are used to calculate and update this model of the Hessian in each iteration of the optimization. Similarly as for Newton's method, a convex quadratic model of the objective function at the current iterate $\vec{x}_k$ is created:

$$f_k^{\text{qn}}(\vec{p}) = f(\vec{x}_k) + \vec{\nabla}_x f(\vec{x}_k)^\top \vec{p} + \vec{p}^\top B_k \vec{p} \tag{2.9}$$

Here, $B_k \in \mathbb{R}^{n \times n}$ is a symmetric positive-definite matrix that acts as an approximation of the current Hessian $\nabla^2_{xx} f(\vec{x}_k)$. Herewith, the search direction $\vec{p}_k$ is given by the vector that minimizes this objective function model and therefore solves the equation $B_k \vec{p}_k = -\vec{\nabla}_x f(\vec{x}_k)$.

The BFGS method for the calculation of the matrix $B_k$ (Broyden 1970, Fletcher 1970, Goldfarb 1970, Shanno 1970) is probably the most popular quasi-Newton variant. It was proven that the convergence of the BFGS method is superlinear. That is, the rate of convergence lies between the steepest descent method and Newton's method. Because there is no curvature information available, the general idea of this method is that the gradients of the quadratic model $f_k^{\text{qn}}(\vec{p})$ should match the gradients of $f$ at the last two iterates $\vec{x}_k$ and $\vec{x}_{k-1}$. These two conditions can be written as

$$\vec{\nabla}_p f_k^{\text{qn}}(0) = \vec{\nabla} f_k \stackrel{!}{=} \vec{\nabla} f_k, \quad (\text{always true}, p = \vec{x}_k - \vec{x}_k = 0) \tag{2.10}$$

$$\vec{\nabla}_p f_k^{\text{qn}}(\vec{x}_{k-1} - \vec{x}_k) = \vec{\nabla} f_k + B_k(\vec{x}_{k-1} - \vec{x}_k) \stackrel{!}{=} \vec{\nabla} f_{k-1}. \tag{2.11}$$

If we rewrite the second expression, we get the so-called secant equation:

$$B_k(\vec{x}_k - \vec{x}_{k-1}) = \vec{\nabla} f_k - \vec{\nabla} f_{k-1}. \tag{2.12}$$

Because $B_k$ has to be positive-definite, the secant equation has only a solution if $(\vec{x}_k - \vec{x}_{k-1})^\top (\vec{\nabla} f_k - \vec{\nabla} f_{k-1}) > 0$. If this is not the case, $B_k$ is usually reset to the unit matrix (Nocedal & Wright 1999) and $\vec{p}_k$ gets the steepest descent direction. The secant equation (2.12) is however not very strong as it is underdetermined. Hence, there are almost an infinite number of solutions. To limit the number of possible solutions, the BFGS method additionally forces this matrix to be as close as possible to the matrix of the previous iteration $B_{k-1}$. That is, a symmetric positive-definite matrix $B_k$ has to be chosen that minimizes the Frobenius norm $\|B_k - B_{k-1}\|$ and fulfills the secant equation. A final requirement for the BFGS update is that the change of the model matrix $B_k - B_{k-1}$ should be a symmetric rank-2 matrix. If we define the vectors $\vec{s}_{k-1} := \vec{x}_k - \vec{x}_{k-1}$ and $\vec{y}_k := \vec{\nabla} f_k - \vec{\nabla} f_{k-1}$, then the BFGS update that satisfies all these requirements is given by

$$B_k = B_{k-1} - \frac{B_{k-1}\vec{s}_{k-1}\vec{s}_{k-1}^\top B_{k-1}}{\vec{s}_{k-1}^\top B_{k-1}\vec{s}_{k-1}} + \frac{\vec{y}_{k-1}\vec{y}_{k-1}^\top}{\vec{y}_{k-1}^\top \vec{s}_{k-1}}. \tag{2.13}$$

A more detailed derivation of this formula can be found in Nocedal & Wright (1999, pp. 194). The initial model matrix $B_0$ is often set to the unit matrix if no additional information about the problem and its scaling exist. In comparison to the calculation of the true Hessian, the computation of the BFGS update (2.13) is light-weight and can be executed very efficiently. Instead of first calculating $B_k$ and then solving $B_k\vec{p}_k = -\vec{\nabla}f_k$, the inverse Matrix $B_k^{-1}$ can be directly determined by applying the Sherman-Morrison-Woodbury formula (Sherman & Morrison 1950, Woodbury 1950) to the update formula (2.13). This makes the calculation of the search direction $\vec{p}_k$ even simpler.

Despite the computational efficiency in calculation speed, there is still the problem that the storage of the model matrix $B$ requires a lot of memory for a large number of dimensions. The limited-memory BFGS method (Nocedal 1980) solves this problem by calculating directly the product $-B_k^{-1}\vec{\nabla}f_k$ on the fly from the recent iterates $\vec{x}_i$ and the corresponding gradients $\vec{\nabla}f_i$. Obviously, a history of these iterates and the gradients has to be stored, in order to execute the computation. Fortunately the algorithm even shows super-linear convergence if not the complete history is stored but only the last $m$ iterations. The typical number of stored iterations is $m = 5 \dots 10$. An outline for the recursive calculation of $-B_k^{-1}\vec{\nabla}f_k$ is presented in algorithm 2. We would like to refer to Nocedal & Wright (1999, pp. 224) for the derivation of this calculation. The limited-memory BFGS (L-BFGS) method requires the storage of only $2n \cdot m$ values. For a high number of dimensions $n$ and a moderate history size $m$ this is much less than the requirement of the standard BFGS method, which has to store $n^2$ values. The complete L-BFGS optimization method is presented in algorithm 1.

**Algorithm 1** Limited-memory BFGS algorithm for unconstrained optimization

set the starting parameters for $\vec{x}_0$
compute $f_0 \leftarrow f(\vec{x}_0)$ and $\vec{\nabla}f_0 \leftarrow \vec{\nabla}_x f(\vec{x}_0)$
$k \leftarrow 0$
**while** $\|\vec{\nabla}f_k\|_2 > \epsilon$ **do**
  calculate BFGS direction $\vec{p}_k$ according to algorithm 2.
  $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k\vec{p}_k$, where $\alpha_k$ is determined by a line search
  compute $f_{k+1} \leftarrow f(\vec{x}_{k+1})$ and $\vec{\nabla}f_{k+1} \leftarrow \vec{\nabla}_x f(\vec{x}_{k+1})$
  calculate $\vec{s}_k \leftarrow \vec{x}_{k+1} - \vec{x}_k$ and $\vec{y}_k \leftarrow \vec{\nabla}f_{k+1} - \vec{\nabla}f_k$
  **if** $\vec{y}_k^\top \vec{s}_k > 0$ **then**
    insert $\vec{y}_k$ and $\vec{s}_k$ into l-BFGS history. If history is full, remove oldest element first.
  **else**
    clear l-BFGS history
  **end if**
  $k \leftarrow k + 1$
**end while**
**return** $\vec{x}_k$

**Algorithm 2** Recursive calculation of the search direction $\vec{p}_k = -B_k^{-1}\vec{\nabla} f_k$ with the limited-memory BFGS update, adapted from Nocedal & Wright (1999, p. 225)

$\vec{p}_k \leftarrow -\vec{\nabla} f_k$;
**for** $i = k-1 \dots k-m$ **do**
  $\alpha_i \leftarrow \left(\vec{s}_i^\top \vec{p}_k\right)\left(\vec{y}_i^\top \vec{s}_i\right)^{-1}$;
  $\vec{p}_k \leftarrow \vec{p}_k - \alpha_i \vec{y}_i$;
**end for**
$\vec{p}_k \leftarrow \frac{\vec{s}_{k-1}^\top \vec{y}_{k-1}}{\vec{y}_{k-1}^\top \vec{y}_{k-1}} \vec{p}_k$
**for** $i = k-m \dots k-1$ **do**
  $\beta \leftarrow \left(\vec{y}_i^\top \vec{p}_k\right)\left(\vec{y}_i^\top \vec{s}_i\right)^{-1}$;
  $\vec{p}_k \leftarrow \vec{p}_k + \vec{s}_i(\alpha_i - \beta)$;
**end for**
**return** $\vec{p}_k$

### 2.1.4 Line search strategies for unconstrained and box-constrained optimization

After the calculation of a search direction $\vec{p}_k$ according to one of the presented schemes, a so-called line search has to be performed that minimizes the function $f(\vec{x})$ along this direction. This one-dimensional optimization problem can be formulated as

$$\alpha_k = \underset{\alpha>0}{\operatorname{argmin}} f(\vec{x}_k + \alpha \vec{p}_k). \tag{2.14}$$

It turns out, that the steepest descent and the BFGS method do not need an exact line search. It is adequate if the parameter $\alpha_k$ provides a "sufficient" decrease of the objective function. Still an inaccurate line search is required for two reasons:

1. Too long steps have to be prevented: the minimum of this 1-d problem can be overshot by far. In this case, the objective function $f(\vec{x}_k + \alpha_k \vec{p}_k)$ can be even larger than $f(\vec{x}_k)$. Amplifying oscillations would be the result and convergence would be destroyed.

2. Prevent too short steps: although short steps will keep convergence intact, the optimization will have a very slow convergence rate.

To prevent too long steps, the step-length $\alpha$ has to provide a sufficient decrease in $f$. This can be expressed by the Armijo-condition

$$f(\vec{x}_k + \alpha \vec{p}_k) \leq f(\vec{x}_k) + \alpha c_1 \vec{\nabla}_x f(\vec{x}_k)^\top \vec{p}_k, \quad c_1 \in (0,1). \tag{2.15}$$

The interpretation of this equation is the following: the right hand of the Armijo-condition is a linear function with a slope that is less steep than the slope of the objective

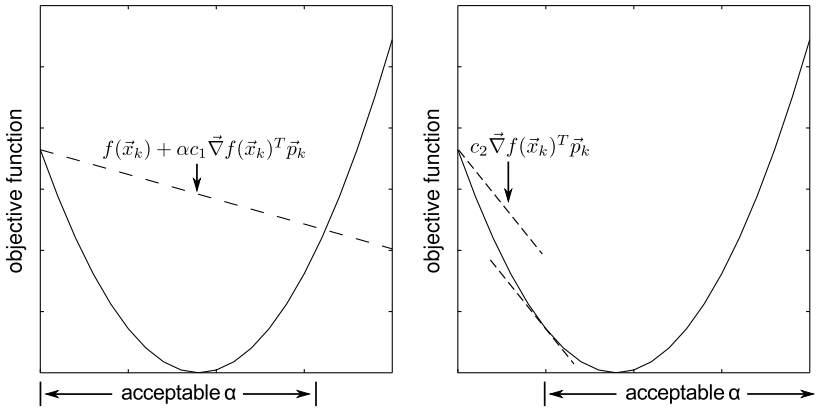



**Figure 2.2:** Illustration of the Armijo-condition (a) and the Wolfe-condition (b). In combination, they prevent too long and too short step sizes $\alpha$ during the line search.

function along $\vec{p}_k$. The latter is given by the directional derivative $\vec{\nabla}_x f(\vec{x}_k)^\top \vec{p}_k$. Thus, only those step lengths $\alpha$ are accepted where the objective function is located "under" this line. The parameter $c_1$ defines the steepness of the line. Usually, it is set to a small value of e.g. $c_1 = 10^{-4}$. The Armijo-condition is illustrated in figure 2.2(a).

The second requirement for the line search is to prevent too short steps. This can be realized with the Wolfe-condition, which is given by

$$\vec{\nabla} f(\vec{x}_k + \alpha \vec{p}_k)^\top \vec{p}_k \geq c_2 \vec{\nabla} f(\vec{x}_k)^\top \vec{p}_k, \quad c_2 \in (c_1, 1). \tag{2.16}$$

This condition forces the step-length $\alpha$ to come closer to the minimum because the slope of $f$ at a given step-length has to be larger (less steep, because $\vec{\nabla} f_k^\top \vec{p}_k$ is negative) than some threshold (see figure 2.2(b)). This threshold is defined by the directional derivative $\vec{\nabla} f_k^\top \vec{p}_k$ times a factor $c_2 \in (c_1, 1)$. For quasi-Newton and Newton methods this parameter is set to a relatively high value of $c_2 = 0.9$ (Nocedal & Wright 1999), which means that this condition is not very strong.

The drawback of the Wolfe-condition (2.16) is that the gradient of $f$ along the search direction has to be evaluated for each $\alpha$. Often, the gradient calculation is computationally very expensive. The standard method that prevents the calculation of gradients is known as *back-tracking*, which starts from an initially high step-length $\alpha_{\max}$. Then, the step-length is iteratively decreased until the Armijo-condition (2.15) is met. In Newton and quasi-Newton algorithms the maximum step-length is normally set to $\alpha_{\max} = 1$, because we expect the minimum of $f$ along the search direction $p_k$ to be at about $\alpha = 1$.

Later it will be shown that the treatment plan optimization has to be modelled as a constraint optimization problem, where one forces the free parameters (the fluence weights) to be greater or equal zero. These type of constraints are called box constraints,

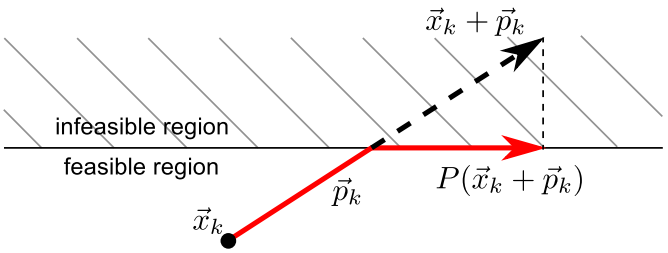



**Figure 2.3:** Illustration of the line search with boundary constraints. The search direction $\vec{p}_k$ points into the infeasible region from the current iterate $\vec{x}_k$. When the boundary is crossed, the vectors are projected onto the boundary. Thus, this one-dimensional search is performed on the red path.

which can be generally described by

$$l_i \leq x_i \leq u_i, \quad \forall i = 1 \dots n. \tag{2.17}$$

During optimization it has to be ensured, that all iterates lie in the feasible region which is defined by the lower bounds $\vec{l}$ and the upper bounds $\vec{u}$. Suppose, the current iterate $\vec{x}_k$ is located near to one of the bounds and the search direction $\vec{p}_k$ points towards this bound. Then, there is a high chance that this bound will be violated during the line search. To prevent the violation of the constraints, the vector $\vec{x}_k + \alpha\vec{p}_k$ is projected into the feasible domain (i.e. into the box) by a projection operator

$$P(\vec{x})_i := \begin{cases} l_i, & \text{if } x_i < l_i \\ x_i, & \text{if } l_i \leq x_i \leq u_i \\ u_i, & \text{if } x_i > u_i \end{cases}. \tag{2.18}$$

With this projection method, the line search changes to

$$\min_{\alpha > 0} f(\vec{P}(\vec{x}_k + \alpha\vec{p}_k)) \tag{2.19}$$

for box-constraints, which ensures that the objective function is only evaluated in the feasible region. It should be noted, that the direction of the line search is changing when one of the boundaries is reached, i.e. a constraint becomes active. Hence, at the beginning of the line search, there will be an effective search direction $\vec{P}(\vec{x}_k + \vec{p}_k) - \vec{x}_k$, which differs from $\vec{p}_k$. This has to be considered for the formulation of the Armijo-condition that includes the slope of $f$ along the search direction $\vec{p}_k$. The pseudo-code for the back-tracking line search with box constraints can be found in algorithm 3.

## 2.2 Treatment plan optimization in IMRT

The aim of optimization in intensity modulated radiation therapy is to find the intensity configuration for each beam direction that satisfies the medical requirements as good as

**Algorithm 3** Back-tracking line search with box-constraints

$\alpha \leftarrow \alpha_{\max}$
set backtracking factor $b \in (0, 1)$
$s_k \leftarrow \vec{\nabla} f_k^{\top} \left( \vec{P}(\vec{x}_k + \vec{p}_k) - x_k \right)$
**while** $f(\vec{P}(\vec{x}_k + \alpha\vec{p}_k)) > f_k + \alpha c_1 s_k$ **do**
  $\alpha \leftarrow \alpha \cdot b$
**end while**
**return** $\alpha$

possible. In the IMRT approach each beam is partitioned into small rectangular sub-beams that we call *beamlets*. Each of these beamlets has a corresponding weight/intensity that has to be optimized. The dimensionality of optimization problems in IMRT is comparably high. Typical values for the number of beamlets $n$ vary from 100 up to 10000, depending on the number of incident beams, the beamlet size and the volume of the target. In proton therapy, the dimensionality can be even higher. The combination of the weights of all beamlets is called a fluence map, which we denote with the vector $\vec{w}$. During optimization the fluence map $\vec{w}$ is iteratively changed until the irradiation dose $\vec{d}(\vec{w})$ satisfies the target dose prescriptions and the tolerances for organs at risk (OAR) doses.

### 2.2.1 Mathematical formulation of inverse planning

In the IMRT optimization framework, the quality of a treatment plan is quantified with an objective function. This objective function measures the violations of the treatment constraints in each voxel of the patient. Typical clinical constraints are:

- Minimum and maximum doses for the target volume (tumor).
- Maximum tolerated dose for an OAR.
- Dose-volume constraints for OARs.
- Biologically motivated constraints for OARs as e.g. the equivalent uniform dose (EUD) (Niemierko 1999).

The widely used objective function quadratically sums up the differences between the prescribed/tolerated dose and the actual dose value of each voxel $i$ (Brahme 1988, Bortfeld et al. 1990, Spirou & Chui 1998):

$$f(\vec{d}) := \sum_i p_i(d_i) \, (d_i - d_i^p)^2 \tag{2.20}$$

In this equation the parameter $p_i$ is a factor that penalizes a dose deviation from its prescribed dose $d_i^p$. For each volume of interest (VOI), a penalty factor is set by the

treatment planner to indicate the priorities of the clinical goals. Suppose, the voxel index $i$ belongs to the VOI $v$ whose penalty factor for underdosage is $p^{\text{vu}}$ and $p^{\text{vo}}$ for overdosage. Accordingly, there are the dose threshold values $d^{\text{vo}}$ and $d^{\text{du}}$ that define over- and underdosage. If $v$ belongs to an OAR, the lower dose threshold $d^{\text{du}}$ is set to zero. With these definitions, the penalty factor of each voxel $i$ is given by

$$p_i(d_i) := \begin{cases} p^{\text{vo}}, & \text{if } d_i > d^{\text{vo}} \\ 0, & \text{if } d^{\text{vu}} \leq d_i \leq d^{\text{vo}} \\ p^{\text{vu}}, & \text{if } d_i < d^{\text{vu}} \end{cases}. \tag{2.21}$$

Accordingly, the prescribed dose per voxel $d_i^p$ is set to the upper dose threshold $d^{\text{vo}}$ on overdosage and to the lower dose threshold $d^{\text{vu}}$ on underdosage. For some organs it is clinically important that some fraction of the organ does not exceed a specific dose. These kind of clinical goals are called *dose-volume constraints*. We implemented these constraints into the optimization with the same formalism of prescribed dose values and penalties (Spirou & Chui 1998): suppose a dose-volume constraint is violated so that more than $p\%$ of the organ $v$ receives a dose greater than $d^c$ . Then each voxel, whose dose exceeds $d^c$, gets a prescribed dose of $d^c$ with a penalty factor of $p^{vo}$. In our implementation each organ can have up to five different dose-volume constraints. A detailed description of this inclusion of dose-volume constraints into the optimization framework is given by Bortfeld et al. (1997).

The objective function 2.20 can be reformulated as

$$f(\vec{d}) = (\vec{d} - \vec{d^p})^\top P(\vec{d})(\vec{d} - \vec{d^p}), \tag{2.22}$$

where $P$ is a diagonal matrix with the voxel penalty factors on its diagonal, i.e. $P_{ii} = p_i$.

If we neglect some effects like photon scattering at the leaf edges, inter-leaf leakage and the tongue-and-grove effect, the dose distribution $\vec{d}$ is a linear function of the fluence weights $\vec{w}$ and can be characterized by the linear equation

$$\vec{d}(\vec{w}) = J\vec{w}. \tag{2.23}$$

The so-called *dose-influence matrix* $J$ contains the normalized dose contribution of each beamlet to the total dose distribution $\vec{d}$. The advantage of this method is, that the dose calculation is simplified to a matrix-vector-product which can be calculated very efficiently. The matrix is usually calculated with a dose calculation algorithm once per patient prior to the actual optimization loop. With the linearized dose function, we can denote optimal fluence $\vec{w}^*$ as the solution of the actual optimization problem:

$$\vec{w}^* = \underset{\vec{w} \geq 0}{\arg\min}\ (J\vec{w} - \vec{d^p})^\top P(\vec{d})(J\vec{w} - \vec{d^p}) \tag{2.24}$$

The positivity of $\vec{w}$ is a physical constraint because a "negative fluence" cannot be realized (that is, it is impossible to remove irradiation dose). In the following this optimization problem (2.24) is called fluence map optimization (FMO).

### 2.2.2 Convexity of the FMO problem

The definition of convexity was already given in equation (2.2) which is equivalent to the positive-definiteness of the Hessian matrix $\nabla^2_{xx} f$ over the feasible domain. The first and second derivative of our objective function (2.24) are given by

$$\vec{\nabla}_w f(\vec{w}) = 2\left(J^\top P J\vec{w} - J^\top P\vec{d^p}\right) \tag{2.25}$$

$$\nabla^2_{ww} f(\vec{w}) = 2J^\top P J. \tag{2.26}$$

To prove convexity, we show that $\vec{p}^\top \left[\nabla^2_{ww} f(\vec{w})\right] \vec{p} > 0$ for all $\vec{w} \geq 0$ and $\vec{p} \in \mathbb{R}^n \neq 0$:

$$\vec{p}^\top \left[\nabla^2_{ww} f(\vec{w})\right] \vec{p} \tag{2.27}$$

$$= 2\vec{p}^\top J^\top P J\vec{p} \tag{2.28}$$

$$= 2\vec{d}(\vec{p})^\top P\vec{d}(\vec{p}) \tag{2.29}$$

$$\geq 0,\ \forall\vec{w} \in \mathbb{R}^n_+,\ \forall\vec{p} \neq 0 \tag{2.30}$$

The last conclusion holds due to positive-semidefiniteness of the diagonal matrix $P$ as the penalty values on the diagonal $P_{ii}$ are greater or equal zero. Up to now, we only prove positive-semidefiniteness of the Hessian. If we suppose that there is some dose deposited in the patient ($\vec{d}(\vec{w}) \neq 0$) and also some of the clinical constraints are violated, then this expression is strict positive. If no constraints are violated and hence the matrix $P$ is zero, the problem could be formulated more aggressively by reducing for example tolerated doses for organs at risk. Alternatively, the optimization can be simply halted. □

### 2.2.3 The IMRT optimization cycle

Due to the convexity of the objective function (2.24), all the methods described in section 2.1 can be directly applied to this problem. Because the objective function is almost quadratic (there are some plateaus when a medical constraint is not violated), quasi-Newton methods are well suited for the optimization. Due to the large number of parameters/dimensions, the calculation of the Hessian matrix for Newton's method is too expensive with respect to calculation time and memory usage.

With the gained knowledge of algorithmic optimization, we created a general purpose optimization library for the use in C++ programs. Our implementation of the limited-memory BFGS algorithm is an adaptation of the bound-constraint BFGS of Kelley (1999). The great convergence performance of the limited-memory BFGS algorithm on IMRT problems was already demonstrated before (Pflugfelder et al. 2008). In addition to the presented algorithms, we also created a wrapper for the famous NPSOL library (Gill et al. 1984) for general constraint optimization and for the L-BFGS-B package (Zhu et al. 1997) for box constraints.

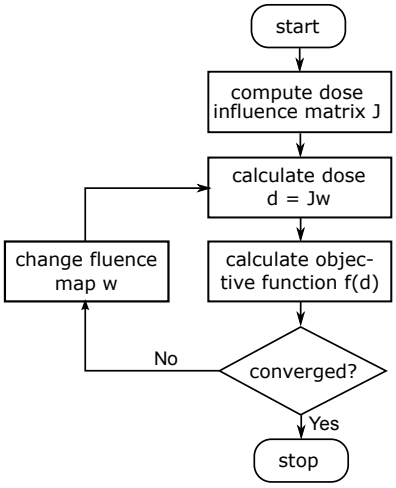



**Figure 2.4:** Optimization cycle in IMRT

The typical IMRT optimization cycle is presented in figure 2.4. Due to the linearization of the dose function $\vec{d}(\vec{w})$, the dose influence matrix $J$ can be calculated before entering the optimization loop. Not only the dose calculation in each iteration is faster by doing this, but it also allows a strict decoupling of dose-calculation and optimization algorithms. Independent of which optimization method is chosen, the optimization follows the general scheme of figure 2.4. In each iteration the dose distribution $\vec{d} = J\vec{w}$ of the current fluence map $\vec{w}_k$ is calculated first. With these data available, the objective function $f(\vec{d}(\vec{w}_k))$ and its gradient $\vec{\nabla}_w f(\vec{d}(\vec{w}_k))$ are computed. If we derive our objective function (2.24) with respect to the fluence maps $\vec{w}$, its gradient is given by the following equation:

$$\vec{\nabla}_w f(\vec{w}) = 2\left(J^\top P J\vec{w} - J^\top P\vec{d}^p\right) \tag{2.31}$$

$$\Leftrightarrow \left[\vec{\nabla}_w f\left(\vec{w}\right)\right]_k = 2\sum_i p_i(d_i)\left(\sum_j J_{ij}w_j - d_i^p\right)J_{ik}. \tag{2.32}$$

In each iteration, these gradients are used to calculate the search direction $\vec{p}_k$ according to an update scheme that depends on the optimization algorithm (see section 2.1). With this search direction and the line search parameter $\alpha_k$, the fluence map is changed as $\vec{w}_{k+1} = \vec{w}_k + \alpha_k\vec{p}_k$. Convergence is detected if the relative change of the objective functions falls below a threshold. That is, if

$$|f(\vec{w}_k) - f(\vec{w}_{k-1})| < \epsilon_f \max\left(|f(\vec{w}_k)|, 1\right). \tag{2.33}$$

A second convergence criterion is the relative step size. The optimization stops, if

$$\|\Delta\vec{w}\|_2 < \epsilon_w\|\vec{w}\|_2. \tag{2.34}$$

In our implementation, the threshold values $\epsilon_f$ and $\epsilon_w$ can be configured by the user. Our current configuration uses $\epsilon_f = 10^{-5}$ and $\epsilon_w = 10^{-3}$.

Currently, the only hard constraint used in the optimization is the positivity of the fluence map. The clinical goals are treated by the objective function as soft constraints which are penalized on violation. Other hard constraints, like a maximum monitor-unit limit are not included in the optimization at the moment. In order to limit the number of monitor units, an alternative is to treat it as a soft constraint by inserting a term into the objective function that penalizes the variance of a fluence map, as proposed by Webb (2001).

### 2.2.4 Clinical plan quality vs. objective function

It is indisputable that the objective function is only a mathematical concept, trying to convert a 3-dimensional dose distribution into one single number. Obviously, this approach has its disadvantages:

- The objective function is not aware of local features of the dose distribution. Therefore it is impossible to influence directly some part of the dose distribution for example to reduce dose hot spots. A typical workaround is to give these local regions different dose prescriptions or penalties by creating additional "virtual" OARs.
- Penalty factors are abstract. The quality of the resulting treatment plan depends to a great extend on these factors. The right choice of these penalties requires much experience.
- Inverse planning is not just "click & go". Often many correction of dose prescriptions, DVH-constraints and penalties have to be made in order to create a clinically good plan. As a consequence, dose prescription values in the algorithm may be different from the clinical goals; often they are tweaked to guide the optimizer in the right direction.
- There is no connection between mathematical optimality of $f(\vec{d})$ and the clinical optimality of the dose $\vec{d}$. Also, after a certain number of iterations of optimization, the clinical quality of a treatment plan is changing only marginally. However, the further decrease of the objective function is caused by a "fine-tuning" of the fluence map that often leads to unsmooth or noisy fluence maps.
- The created fluence maps are not directly deliverable. For the common step-and-shoot irradiation mode, each 2-dimensional fluence map has to be converted into a sequence of deliverable apertures. This step is called sequencing and is often executed as a post-processing step only (Xia & Verhey 1998). In this case, sequencing leads to a degradation of the treatment plan (Siebers et al. 2002).

To give the treatment planner more control over the result of a treatment plan, IMRT algorithms were developed with the multi-criteria optimization approach, where each VOI has its own objective function (Hamacher & Küfer 2002). There, the optimization algorithm creates a database of Pareto optimal treatment plans. After this calculation, the treatment planner is able to intuitively navigate through the database with a graphical user interface and hence has direct control over the dose of each VOI (Monz et al. 2008). In comparison to the standard IMRT optimization algorithm, this approach is however much more computing intensive.

To avoid the degradation of the treatment plan due to the leaf sequencing step, new algorithms were proposed that directly incorporate only deliverable apertures into the optimization. The so-called direct-aperture-optimization (DAO) is unfortunately not convex. Therefore, the fast gradient based optimization methods do not lead to an optimal solution. Instead, several methods from global optimization like simulated annealing (Shepard et al. 2002) or a column generation approach from linear programming (Romeijn et al. 2005) seem promising for the solution of this complex problem.

### 2.2.5 Compact storage of the dose influence matrix

Before entering the actual optimization loop (see figure 2.4), a dose influence matrix is calculated that contains the dose contribution of each fluence element (beamlet) to the total dose. However, the storage requirement for this matrix is huge: the complete dose influence matrix with $n$ voxels and $m$ beamlets at floating point precision requires $4 \times n \times m$ bytes of local memory. Thus, a typical treatment plan with $200^3$ voxels and 2000 beamlets would require 64 GB, which is far too much in comparison to the memory size of current computers (about 4 GB). Because of the finite size of each beamlet and the limited range of secondary electrons that lead to a dose smearing, the dose contribution of each beamlet is locally bounded. Practically this means that there are many voxels that do not receive any dose from the beamlet. Hence, the dose influence matrix $J$ is sparse.

Due to the size of $J$ and its sparsity, current state-of-the-art algorithms do not store the complete matrix but a list of dose values and indices, in which the dose contribution is larger than zero. A typical dose cube has a size of up to $500^3 \approx 1.3 \cdot 10^8$ voxels. To store the dose indices, a data-type has to be chosen that covers this range. Our version of the inverse planning software KonRad (Preiser et al. 1998, Oelfke & Bortfeld 2001, Nill et al. 2004), which is currently used for inverse treatment planning at the German Cancer Research Center, uses 4 byte integers for the storage of the voxel indices. To reduce memory usage, the dose values are discretized into 2 byte integers. That is, the encoding of the voxel index takes twice as much memory than the actual dose value. If the memory requirement of the voxel indices can be reduced, we could handle treatment plans with an even larger number of beamlets or with a finer spatial resolution.

Here we present a lossless compression method that reduces the required memory of the stored voxel index to an average of only 1 byte. With the new method only 3 bytes are required for the storage of one dose element in comparison to 6 byte with the old method. Therefore, the size of the dose influence matrix is halved. The problem: with one byte per voxel index we can only encode voxel indices from $0 \dots 255$. The solution: we do not store absolute voxel indices but the difference to the previous ones. Obviously, voxels with dose contributions are physically and spatially connected. Due to the three-dimensional locality we conclude that the local connection also exists in the one-dimensional memory model. If one voxel received a dose, its neighbors in memory often also have dose values larger than zero. Still, there can be gaps between voxel indices that are larger than 255 and hence cannot be encoded by one byte. In this case, additional bytes are required to encode the difference to the previous voxel index.

The general idea is that the memory usage of each encoded index difference is dynamically adapted. Because small index differences are more common than large differences, their storage should require the least memory. To support this argument, the average distribution of index differences in dose influence matrices is illustrated in figure 2.5. In about 97% of the cases the difference of two consecutive dose indices is less than 192. Let $I_k, k \in 0 \dots n-1$ be the ordered list of the voxel indices with a dose $d_{I_k} > 0$. Then, the list of index differences is defined by

$$\Delta I_k = I_k - I_{k-1}, \quad \text{and } \Delta I_0 = I_0. \tag{2.35}$$

In our compression algorithm, the index differences are encoded as a list of 1-byte integers. The first bits of the current 1-byte integer determine, how many following bytes of the list have to be read, in order to decode the current index difference. According to the bit pattern of the current 1-byte integer $b_i$, one of each rules have to be applied for the decoding of $\Delta I_k$:

**1111xxxx** If this pattern is discovered (the current integer is larger than 240), three more bytes have to be read to decode the current $\Delta I_k$. With four bits from the current byte and 24 bits from the three following, we have in total 28 bits which corresponds to a maximum voxel index difference of about $2.68 \cdot 10^8$. For dose cube sizes currently used in treatment planning, this should be more than enough. If the dose cube sizes are increasing in the future, this method can be adapted to include more additional bytes.

**11xxxxxx** Bytes $b_i$ with $192 \leq b_i < 240$ can be recognized by the bit pattern 11xxxxxx. If this is the case, only one additional byte is read to decode the current index. 14 bits are available in total. However, if the first of the available bits is set, the second must be zero. Else, we would have case one. Still, this allows the encoding of integers up to a value of 12287.

**else** If we exclude all number with the first two highest bit set, we have a range of integer values from 0 to 191. This is the most probable case, according to figure 2.5.

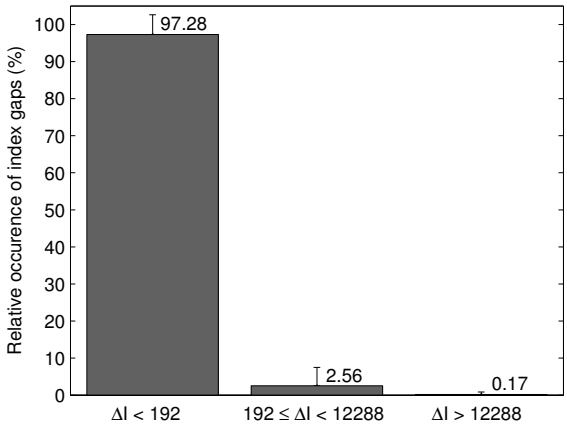



**Figure 2.5:** Motivation for the binning scheme (0–191, 192–12287, 12288–) for the compression/decompression of sparse dose cubes: illustrated is the distribution of the index differences $\Delta I_k$ in dose influence matrices. These data where generated from 930 matrices with different sizes. Most index differences are smaller than 192 and can be encoded with only 1 byte.

With these rules, the compression of an index difference list is straight-forward: if the current voxel difference $\Delta I_k$ is smaller than 192, it is stored in one byte. However, if it is larger or equal than 192 but smaller than 12288, the first two bits are "masked" with ones and the following 14 bits can be used to encode $\Delta I_k$. In the rare case of $\Delta I_k \geq 12288$, we set the first 4 bits of the first byte to one and use the remaining 28 bits for the storage of $\Delta I_k$.

The presented method has one significant disadvantage so far. The decoding of the voxel indices is a serial process because the computation of a voxel index $I_k$ requires the calculation of the previous index $I_{k-1}$ first. As a consequence, the algorithm cannot be parallelized or used in the parallelized matrix-vector product for the dose calculation. However, the method can be expanded with little sacrifice of memory usage if every $l-$th voxel index of the list $I$ is directly stored instead of saving the difference to its predecessor. Due to the varying memory usage of each dose element, the position of each so-called key index (every $l-$th index of the list) has to be calculated and stored once. If $l$ is chosen large enough, the additionally memory required for the storage of the key index list and for the directly stored voxel indices is negligible. In the current implementation of the optimization framework, we use a key index interval of $l = 256$. With the list of key index positions, the key indices can be subdivided amongst all threads. Then, each thread in a parallelized algorithm can decode a part of the whole list by jumping directly to its assigned key index.

To evaluate the performance of the decompression algorithm, we compare the runtimes for the dose calculation and for the calculation of gradients of the objective function with and without compression. These measurement are performed on the sequential and the parallel version of those calculations.

## 2.3 Monte Carlo simulations

### 2.3.1 Random number generation from nonuniform distributions

At the heart of each Monte Carlo simulation is the generation of random numbers from probability distributions. That is, each random event $\vec{r}$ has a certain probability $p(\vec{r})$. In the following we will discuss methods, how to generate these random events that obey such a probability distribution. The next methods require that there is access to a (pseudo) random number generator that creates uniformly distributed random numbers in the range $(0, 1)$.

#### Direct/Analytical Sampling

Probably the most elegant method of drawing random numbers is direct sampling. Suppose we have an analytical representation of the one-dimensional probability distribution $p(r)$. Its cumulative distribution function $P(r)$ is then defined by the integral

$$P(r) := \int_{-\infty}^{r} p(t)dt. \tag{2.36}$$

If $P(r)$ is invertible and $u \in (0, 1)$ is an uniformly distributed random variable, the random number $r = P^{-1}(u)$ obeys the distribution function p(r) (Devroye 1986, pp. 27).

Consider the example of normally distributed random variables: the normal distribution is defined by $p(r) := \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}r^2}$. Calculating the integral (2.36), the cumulative distribution function of $p(r)$ is given by

$$P(r) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{r} e^{-\frac{1}{2}t^2} dt = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{r}{\sqrt{2}}\right)\right). \tag{2.37}$$

If we invert this function, we get $r = \sqrt{2}\,\operatorname{erf}^{-1}(2u - 1)$ and the variable $r$ will be normally distributed. The inverse of the error function can be calculated by a series expansion. In practice, normally distributed random variables are generated by less computational intensive methods. One popular method is the Box-Muller-Transform (Box & Muller 1958) that generates two normally distributed random variables $r_1, r_2$ from two uniformly distributed variables $u_1, u_2$ at once.

The advantage of the direct method is that only one input random variable $u$ has to be drawn for each output random variable $r$. Thus, it has a maximum efficiency. The disadvantage is that this method only works if the cumulative distribution function $P(r)$ is invertible. This is only the case in 1-dimensional sampling problems. Practically, the analytical or numerical calculation of $P(r)$ can be difficult, especially if the probability function $p(r)$ has poles or other pathologies.

### Rejection Sampling

The rejection sampling method (von Neumann 1951) can be also utilized on multi-dimensional probability distribution functions $p(\vec{r})$. In the first stage the random variable $\vec{r}$ is drawn by some arbitrary random number generator. In the second stage, the algorithm decides according to its probability $p(\vec{r})$, whether to keep $\vec{r}$ or reject it and draw another random number. Suppose the maximum probability $p_{\max} = \max p(\vec{r})$ is known or can be calculated and $u$ is an uniformly distributed random variable in the interval $(0, 1)$. Then the random variable $\vec{r}$ is accepted, if

$$u \leq \frac{p(\vec{r})}{p_{\max}}. \tag{2.38}$$

The algorithm is repeated until some variable $\vec{r}$ is eventually accepted. This is also the main drawback of the method: first, for each random variable $\vec{r}$ an additional random number $u$ has to be drawn for the decision process (2.38). Second, if $p(\vec{r})$ is very spiky, a lot of trial numbers have to be drawn until some $\vec{r}$ is finally accepted. Thus, the efficiency of the method depends mostly on the shape of $p(\vec{r})$ and can be very poor.

### Markov Chain Monte Carlo

Markov chains describe systems that perform transitions from one state to another. The next state (and/or its probability) depends only on the current state and not on the whole system.

In the Markov Chain Monte Carlo (MCMC) sampling algorithm, the current value of the sampling variable $\vec{r}_c$ is interpreted as a state of a Markov Chain. Then, the probability of a state transition to the variable $\vec{r}$ depends only on the probabilities $p(\vec{r}_c)$ and $p(\vec{r})$. One of the most important MCMC sampling algorithms is the Metropolis-Hasting algorithm (Metropolis et al. 1953, Hastings 1970). In this algorithm, the transition probability is given by

$$p_t(\vec{r}_c \to \vec{r}) = \frac{p(\vec{r})Q(\vec{r}_c, \vec{r})}{p(\vec{r}_c)Q(\vec{r}, \vec{r}_c)}. \tag{2.39}$$

In this equation, $Q(\vec{r}_c, \vec{r})$ is the so-called proposal density function that is used to generate a proposal variable $\vec{r}$. That is $\vec{r}$ is sampled from $Q$. The practical use of this function is to avoid too large steps in the variable space from one random variable to another. Consider the random walk as an example. Here, $Q$ could be the normal distribution, which practically limits the spatial distance $\|\vec{r} - \vec{r}_c\|_2$ to be smaller than $2\sigma$ in 96% of the cases.

Given the transition probability function $p_t$, the following steps are executed in the MCMC algorithm:

1. Sample the proposal variable $\vec{r}_p$ from the proposal density function $Q(\vec{r}_c, \vec{r}_p)$.

2. Calculate the transition probability $p_t(\vec{r}_c \rightarrow \vec{r}_p)$ according to (2.39).

3. Draw an uniformly distributed random number $u \in (0, 1)$.

4. If $u < p_t$, accept transition ($\vec{r}$ gets $\vec{r}_p$). Else, reject transition and keep current state ($\vec{r}$ gets $\vec{r}_c$).

In principle, the Metropolis-Hastings algorithm is similar to the rejection method. However, there are important differences: first, the new random variable $\vec{r}$ is set after only one round of acceptance or rejection. This fact makes the Metropolis-Hastings algorithm much more efficient. Second, there is no need to determine the maximum probability in the variable space. Especially when sampling variables from a high-dimensional space, it is not always possible to determine this maximum (in a limited time).

It should be noted however, that random variables from Markov Chain algorithms are not statistically independent anymore but are highly correlated (Bishop 2006, section 11.2). Also, the distribution of the generated random numbers only converges against $p(\vec{r})$ for a large number of samples.

### 2.3.2 Monte Carlo dose calculation

One important application of Monte Carlo simulations is the dose calculation in radiation therapy. It is regarded as one of the most accurate dose calculation techniques in radiation therapy. In terms of accuracy, only a novel technique that solves the Boltzmann transport equations is able to compete with Monte Carlo simulations (Vassiliev et al. 2010). Other established dose calculation methods are based on analytic models and their accuracy is limited in cases with strong tissue inhomogeneities (Scholz et al. 2003, Krieger & Sauer 2005). Depending on the type of radiation therapy the tracks of either photons, electrons or protons are initially simulated through a virtual representation of the treatment machine. The emerging particles are eventually transported through the patient and their interactions with the tissue are simulated. If a particle undergoes an inelastic interaction, its transferred energy to the medium is recorded and is finally converted into the irradiation dose.

The physical and mathematical theory behind Monte Carlo dose calculations is very complex and its complete description will go beyond the scope of this work. Therefore, just the basic principles of the theory are mentioned in the following. If the reader is interested in more details, an excellent introduction into the topic is given by the Monte Carlo book of Bielajew (2001).

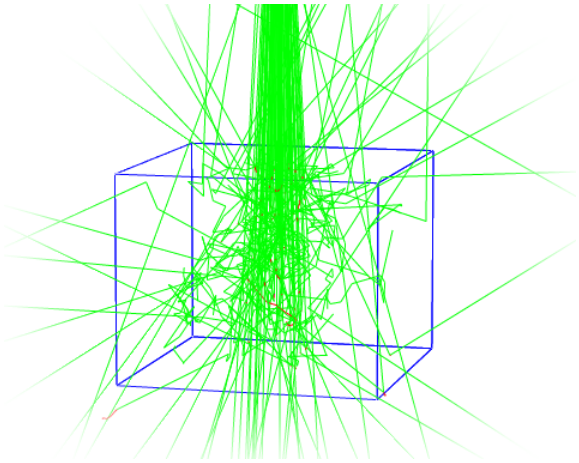


**Figure 2.6:** Example of a Monte Carlo simulation of a high energy photon beam imping on a water phantom. Photons are depicted green, electrons red. Courtesy of Georg Altenstein, created with Geant4 (GEANT4 Collaboration 2003).

### Simulating particle-medium interactions in photon therapy

In radiation therapy with high energy photons, the particle interactions with the patient can be practically limited to interactions of photons and electrons with matter. Most of the particles emerging from a typical MeV linear accelerator are photons that were initially generated in form of bremsstrahlung events, where electrons are stopped in a target of tungsten or other high-Z materials. When these photons enter the patient, they interact with the tissue by one of the interaction channels:

**Photoelectric effect** The dominant interaction for low energy photons is the photoelectric effect. Photons are absorbed by an electron of an atom which then is emitted by the atom as a consequence. The maximum kinetic energy of the emitted electron is given by the energy of the photon minus the binding energy of the electron.

**Inelastic scattering** Also known as Compton scattering, this process removes an electron from the atom. The initial energy of the photon is split up into the energy of the resulting photon, the kinetic energy of the outgoing electron and the work required to remove the electron from the atomic shell. In the medium energy range, where this effect is dominant (about 100 keV - 10 MeV, depending on the material), the binding energy of the electrons is negligible and therefore the electrons can be considered as free.

**Pair production** If the energy of the incoming photon exceeds about twice the rest energy of an electron ($\approx$ 1022 MeV) the process of pair production gains importance. In this process the photon decays into an electron-positron pair. Due to the conservation of momentum, pair production can only occur in the proximity of a nucleus which can absorb the momentum of the photon. At high energies

($> 10\,\mathrm{MeV}$, depending on the material) this process is the dominant channel of the photon-medium interactions.

**Elastic scattering** Elastic scattering of photons is also known as Rayleigh scattering. The energy of the scattered photon remains the same. Rayleigh scattering often occurs if the wavelength of the photon is large in comparison to the size of the molecules in the medium. Rayleigh scattering occurs in the atmosphere and is the cause of the blue color of the sky. Although the cross section of elastic scattering is at least one order of magnitude less than the photoelectric cross section it is still important for an accurate Monte Carlo dose calculation (Bielajew 2001).

In a typical Monte Carlo simulation, a list stores all particles, their position, direction, energy and charge that are currently inside the simulated geometry. Initially, photons from the particle source will be inserted into the list. Later, also secondary particles that are produced by the interaction events will be added to the list. With this list of particles, the Monte Carlo simulation of photons in the medium would then work as follows (simplified):

1. Draw a photon from the list and get its energy, position and direction. If its energy is smaller than a predefined threshold, this particle is "absorbed". That is, the particle is discarded and its energy is scored as a dose contribution to the current geometry.

2. The distance $z$ to the next interaction point is sampled from the distribution $p(z) = e^{-\mu(E)z}$, where $\mu(E)$ is the energy dependent attenuation coefficient of the material in the current geometry. If the medium is made up of compounds (which is mostly the case), its mass attenuation coefficient $\mu/\rho$ is calculated first. With the mass density $\rho$, the distance to the next interaction is sampled according to $p(z) = e^{-\left(\frac{\mu}{\rho}\right)\rho z}$. The photon is discarded if the patient geometry is left.

3. One of the upper four interaction channels is sampled according to their relative importance $\mu_i(E)$.

4. Finally, the directions and energies of the outgoing particles are sampled from the differential cross section of the current interaction channel. The resulting particles and their phase space properties are stored into the list of particles.

This algorithm is repeated until the list of particles is empty. Then, either new particles from the source are inserted and the algorithm starts over, or the calculation is halted.

All these processes, except from Rayleigh scattering, create secondary electrons which will themselves interact with the medium. Additionally, positrons are created by pair production events. Therefore, also electron and positron interactions have to be modeled in a Monte Carlo dose calculation algorithm. These are:

**Electron-electron/positron scattering** Electron-electron scattering, also called Möller scattering, are events in which incident electrons collide with atomic hull electrons.

Similar to this interaction is the electron-positron (Bhaba) scattering. However, this type has one additional reaction channel, in which the electron-positron pair annihilates to a photon which then decays back into an electron-positron pair. Therefore, the cross section of the Bhaba interaction is larger than the cross section of the Möller scattering.

**Bremsstrahlung** Each change in velocity of a charged particle results in the creation of radiation. Particularly when electrons are decelerated by the electro-magnetic field of a nucleus, so-called bremsstrahlung photons are emitted by these electrons.

**Positron annihilation** Another mode is the positron annihilation, where a positron recombines with an electron and creates a photon pair. This interaction typically follows the pair production, where positrons were created.

The simulation of electrons is similar to the photon transport algorithm. There is however one important difference. The electron processes can be divided into hard and soft events. Hard events are scattering processes with a large deflection angle. These are handled by the same transport logic as in the photon case. Soft events are small angle scattering processes. Usually, many of these soft events are statistically combined and handled separately by a multiple scattering algorithm. That is, the path between two hard events is subdivided into many sub-steps. Along these steps, a continuous slowing down approximation is made from one sub step to another and a small scattering angle is sampled from a theoretical probability distribution. This distribution can be derived by multiple scattering theories such as the Molière theory (Molière 1947, Molière 1948), which is used e.g. by the Geant4 toolkit (GEANT4 Collaboration 2003).

One of the main issues in Monte Carlo dose calculations is the virtual representation of the patient. Theoretically, the material composition for each point in the patient has to be known. Because this is not possible, a computer tomography (CT) image of the patient is created prior to the treatment. This CT image is a three-dimensional voxelized structure. Each voxel of the image represents the attenuation $\mu$ of the x-rays in this point. The CT value of each voxel is stored in Hounsfield Units (HU), which are defined by the relative attenuation of x-rays in a material compared to the attenuation in water:

$$\text{CTvalue} := \left( \frac{\mu}{\mu_{\text{water}}} - 1 \right) \cdot 1000\,\text{HU}. \tag{2.40}$$

Because the attenuation coefficient is normally the only information available for each patient, the actual tissue composition and materials inside the patient have to be "guessed" from these data. One method was proposed in a paper by Schneider et al. (2000), which provides a conversion table to convert HU values into material compositions and mass densities. Thus, the accuracy of the Monte Carlo dose calculation depends to a great extend on this conversion, which includes the type and the number of included materials (du Plessis et al. 1998, Verhaegen & Devic 2005). Another issue is the voxel size dependency of the MC simulation: the larger the size of one voxel is, the more different tissues are combined in one voxel and an additional error may occur (Smedt et al. 2005).

### Estimating dose uncertainties by batch statistics

The result of Monte Carlo calculations comes always with an error due to the stochastic nature of the calculation. For many applications, not only the mean value of the physical quantity but also its uncertainty has to be known. For example in radiotherapy, the relative dose uncertainty in high-dose voxels should be small enough (e.g. less than 2 %) in order to be clinically accepted (Chetty et al. 2007). One of the simplest but very versatile and effective methods to estimate the uncertainty of a calculation is called *batch processing*.

Batch processing splits up a calculation into a certain number of independent calculations, called batches. In order to provide statistically independent results, the random number generator in each batch has to be initialized with a different seed (the initial state of the random number generator). In the case of a MC dose calculation this means that each batch simulates the same number of particles. Suppose $N$ are the number of particles to be simulated and there are $M$ batches, then each batch simulates $N_b = N/M$ particles. The dose $\bar{d}_i$ in each voxel and its uncertainty $\Delta\bar{d}_i$ then are given by the arithmetic mean and the standard error of the mean

$$\bar{d}_i = \frac{1}{M}\sum_{j=1}^{M} d_i^j \tag{2.41}$$

$$\Delta\bar{d}_i = \sqrt{\frac{\sum_{j=1}^{M}(d_i^j - \bar{d}_i)^2}{M(M-1)}}. \tag{2.42}$$

Obviously, this calculation gives only reliable results if the number of batches $M$ is "large enough". Bielajew (2001, p. 57) recommends at least 30 batches for reasonable estimates of $\Delta d_i$. Our configuration of VMC$^{++}$ uses 50 batches for the uncertainty calculation.

### Combining multiple MC runs

Theoretically, the dose in each voxel is a linear function of the number of simulated particles. The more particles are simulated, the more energy is deposited inside the patient and the dose increases.

For practical reasons, this behavior is often changed in Monte Carlo simulations and the dose distribution is eventually normalized by the number of simulated primary particles. With this modification, the magnitude of the simulated dose distribution does not depend on the number of particles but only its statistical uncertainty. The simulated particles then do not act as physical particles anymore but have a certain statistical weight that increases or decreases their effect. Therefore, these particles are often called *histories* instead to emphasize their role in the simulation. To change the dose scaling, these Monte Carlo dose calculation programs allow to specify the monitor units for each particle source instead.

With this modification, the combination of $M$ multiple independent runs is straight forward: if each simulation is calculated with the same number of particles, the combination is given by their arithmetic mean

$$d'_i = \frac{1}{M} \sum_{j=1}^{M} d_i^j. \tag{2.43}$$

If the number of particles $N_j$ for each run $j$ is different, the resulting dose in each voxel $d'_i$ is given by the weighted mean

$$d_i(N) = \sum_{j=1}^{M} \frac{N_j}{N} d_i(N_j), \quad \text{with } N := \sum_{j=1}^{M} N_j, \tag{2.44}$$

where $d_i(N_j)$ is the calculated dose at voxel $i$ by run $j$. If we apply error propagation to this formula, we can estimate the statistical error $\Delta d_i(N)$ of the dose in each voxel:

$$\Delta d_i(N) = \sqrt{\sum_{j=1}^{M} \left( \frac{N_j}{N} \Delta d_i(N_j) \right)^2} \tag{2.45}$$

Here, $\Delta d_i(N_j)$ is the dose uncertainty in voxel $i$ of run $j$ that is estimated with e.g. batch processing (as explained in the previous section). From this equation, an important relation between the number of particles and the dose uncertainty can be derived. Suppose, the dose calculation is split up into $N$ separate calculations. If we apply previous formula to this case, we get

$$\Delta d_i(N) = \frac{1}{N} \sqrt{\sum_{j=1}^{N} \left( \Delta d_i\,(1) \right)^2} = \frac{\Delta d_i(1)}{\sqrt{N}}. \tag{2.46}$$

Strictly speaking, this calculation is not correct because there are no statistics for simulations with only one particle. This equation holds however if we think of $N$ separate runs with an equal amount of particles instead. Still, we can conclude the important and well known general relation

$$\Delta d(N) \sim N^{-1/2}. \tag{2.47}$$

If this relation holds for the dose uncertainty in each voxel then it holds for the mean uncertainty of a total dose distribution as well. With this proportionality, we can estimate how many particles $N$ are required in order to get a specific mean dose uncertainty $\sigma$ if a previous test run with $N_{\text{ref}}$ particles and a dose uncertainty $\sigma_{\text{ref}}$ was carried out:

$$N = N_{\text{ref}} \left( \frac{\sigma_{\text{ref}}}{\sigma} \right)^2 \tag{2.48}$$

When calculating dose distributions with MC algorithms, the treatment planner should not be forced to specify the number of particles/histories. Instead, the desired uncertainty of the final dose is usually specified (Chetty et al. 2007). Therefore, a MC dose algorithm may perform an initial dose calculation with a small number of particles. Then, the remaining number of particles can be approximated with the upper equation in order to achieve the requested mean dose uncertainty.

### Smoothing dose distributions

Depending on the final uncertainty of a MC simulation, the visualization of the dose distribution as dose plots and isodose lines might appear noisy or distorted. With noisy dose data, local dose features like hot and cold spots can often be hard to identify as they cannot be distinguished from the noise. One method to reduce the noise was presented in the previous section, which was the calculation of additional particles. Due to possible time restrictions, this may not always be a practical solution, because the reduction of the uncertainty by a factor of 1/2 requires four times as many particles.

One fast alternative to reduce noise is the smoothing of the dose distribution. There are a variety of smoothing algorithms for dose distributions and their performance was compared in (Naqa et al. 2005). In this work we used an anisotropic diffusion based filter whenever smoothing was required. Anisotropic diffusion is a standard image denoising technique and was first described by Perona & Malik (1990). The advantage of this method in comparison to standard filters is that it tries to preserve edges and steep gradients. This is achieved by suppressing diffusion at pixels/voxels with a strong gradient and increase diffusion at homogeneous image data. A diffusion coefficient is assigned to each pixel, which depends on the size of gradients around the pixel: small gradients result in large diffusion, large gradients in small diffusion coefficients. Mathematically, diffusion can be described by the differential equation

$$\frac{\partial}{\partial t} d(\vec{x},t) = \operatorname{div}\left(c(\vec{x},t)\vec{\nabla} d(\vec{x},t)\right). \tag{2.49}$$

Here, $c(\vec{x},t)$ is the time and spatial dependent diffusion coefficient that controls the amount of diffusion at each point $\vec{x}$ in space. The image (dose distribution) is represented by the function $d(\vec{x},t)$. In the original paper, the two gradient-based bell-shaped functions for the diffusion coefficient were proposed:

$$c_1(\vec{x},t) := e^{-(\|\vec{\nabla} d(\vec{x},t)\|_2/K)^2} \tag{2.50}$$

$$c_2(\vec{x},t) := \frac{1}{1+(\|\vec{\nabla} d(\vec{x},t)\|_2/K)^2} \tag{2.51}$$

The width $K$ of the bell curves acts as a threshold for the gradients. If the size of the image/dose gradient $\|\vec{\nabla} d(\vec{x},t)\|_2$ is larger than $K$, the diffusion coefficients gets small and diffusion will be suppressed.

The anisotropic diffusion technique was adapted by Miao et al. (2003) to incorporate also the local dose uncertainty $\Delta d(\vec{x})$ into the diffusion process. Herewith, the gradient threshold $K$ is a function of the spatial dose uncertainty and gets $K(\vec{x}) = k\Delta d(\vec{x})$. The interpretation of the modified diffusion is simple: voxels with large uncertainty get a large threshold value $K$. That is, the dose gradient in this voxel has to be relatively large in order to suppress smoothing. If the dose uncertainty is already small, diffusion will be reduced.

The diffusion process (2.49) is iteratively computed by simulating diffusion in discrete time steps $\Delta t$. Details about the numerical diffusion calculation can be found in the original paper. In the current implementation we are simulating 4 iterations of diffusion with a threshold of $k = 1.75$, as proposed in (Miao et al. 2003). The time constant was set to a lower value $\Delta t = 3/44$ because we also incorporated gradient contributions from diagonals into the calculation. Our C++ implementation is based on the MATLAB®code by Lopes (2007), who proposed this time constant if gradients from diagonal voxels are included into the calculation. For the calculation of the diffusion coefficient we chose the Cauchy-Lorentz function $c_2(\vec{x}, t)$ as recommended in (Miao et al. 2003).

### Objective function estimation from uncertain dose distributions

In the IMRT optimization cycle (section 2.2.3) objective function values are calculated for each occurring dose distribution to quantify the quality of the treatment plan. If the underlying dose distribution is however noisy – as it is the case in MC dose calculations – its objective function value has an uncertainty.

Let $\vec{d}$ be a dose distribution; each entry of the vector corresponds to the dose of a voxel in the patient. In addition, let $\Delta\vec{d}$ be the associated dose uncertainty in each voxel. In section 2.2.1, the objective function was defined as the sum over the dose differences to the prescribed doses in each voxel $i$:

$$f(\vec{d}) := \sum_i p_i(d_i)\,(d_i - d_i^p)^2\,. \tag{2.52}$$

Suppose there is an underlying true dose distribution without any statistical error. If this dose distribution is near to the optimal solution, many dose values $d_i$ will be at close distance to their prescribed dose values $d_i^p$. If we add noise to the dose distribution, the distance to $d_i^p$ will increase in many voxels and therefore the value of a quadratic objective function will increase. In most cases, this objective function value is increasing with increasing dose uncertainties and vice versa. In the following, we will describe possible methods to estimate the objective function value of the "true noise-free dose distribution" from noisy dose data. In MC-based treatment plan optimization, these objective function values are highly interesting. These values allow to distinguish if an objective function is decreasing because of an improved fluence map or if the decrease is just an artifact due to a reduced dose uncertainty.

**1. Gaussian error propagation** The naive approach of approximating the uncertainty of the objective function is given by the classical error propagation mechanism. If we assume that the penalty parameters $p_i$ and dose prescriptions $d_i^p$ are constant for small variations of $d_i$, then the error propagation gives the following relation:

$$\Delta_G f(\vec{d})^2 \approx \sum_i \left(\frac{\partial f}{\partial d_i}\Delta d_i\right)^2 = \sum_i \left(2p_i\,(d_i)\,[d_i - d_i^p]\,\Delta d_i\right)^2 \tag{2.53}$$

The Gaussian error propagation is derived from a first order Taylor expansion. That is, it is only accurate if the objective function is "linear enough" at the current dose distribution $\vec{d}$. However, if this dose distribution is near to the optimal solution, this is not the case anymore due to the following reason: the mean dose of one organ will be approximately its prescribed dose and the dose values $d_i$ inside this organ are scattered around the prescribed dose. Many dose values will be close to the minimum of the parabola $(d_i - d_i^p)^2$. In this region, a first order Taylor expansion results in large errors and therefore underestimates the variation of $f$ induced by the uncertainties $\Delta d_i$.

Because the objective function value of the noise-free dose distribution $f(\vec{d^t})$ is always smaller than the objective function value of the noisy dose $f(\vec{d})$, we approximate

$$f(\vec{d^t}) \approx f(\vec{d}) - \sqrt{\Delta_G f(\vec{d})^2}. \tag{2.54}$$

**2. Objective function approximation by dose smoothing** Another method to evaluate the objective function of the "true dose distribution" $f(\vec{d^t})$ is to apply a smoothing filter on the noisy dose distribution (see section 2.3.2). Since the smoothing reduces the variance of the dose distribution but still preserves edges and strong gradients, the resulting dose distribution $\vec{d^s}$ is a better approximation of $\vec{d^t}$.

With the smoothed dose distribution, the objective function of the noise free dose distribution is given by the approximation

$$f(\vec{d^t}) \approx f(\vec{d^s}). \tag{2.55}$$

**3. Objective function approximation with noise simulation** A third method to approximate the objective function of the underlying "true dose distribution" is the simulation of noise. If $\vec{d^t}$ is again this true dose distribution, a noisy dose distribution can be simulated, by randomly adding or subtracting $\Delta d_i$ on each dose element $d_i^t$. The objective function value of this noisy dose distribution will be larger than $f(\vec{d^t})$ due to the reason described above.

However, the true dose distribution is unknown. Therefore we can only analyze the impact on the objective function if noise is added on the already noisy dose distribution $\vec{d}$. Let $\mathcal{N}(x)$ be a function that randomly returns $x$ or $-x$ and let $\vec{\mathcal{N}}(\vec{x})$ be its multidimensional variant. Then the objective function of the current dose distribution $\vec{d}$ plus noise is given by $f(\vec{d} + \vec{\mathcal{N}}(\Delta\vec{d}))$. If we finally assume that the absolute increase of the objective function is constant (independent of the underlying noise), that is

$$f(\vec{d^t} + \vec{\mathcal{N}}(\Delta\vec{d})) - f(\vec{d^t}) \approx f(\vec{d} + \vec{\mathcal{N}}(\Delta\vec{d})) - f(\vec{d}), \tag{2.56}$$

and also assume that the objective function values of the noisy dose distribution and the objective function of the artificially created noisy dose distribution are the same

$(f(\vec{d^t} + \vec{\mathcal{N}}(\Delta d)) = f(\vec{d})\ )$, we get the approximation for the objective function of the true dose:

$$f(\vec{d^t}) \approx 2f(\vec{d}) - f(\vec{d} + \vec{\mathcal{N}}(\Delta \vec{d})) \tag{2.57}$$

The assumptions made in this model are quite crude. Still, it will be shown in section 3.4 of the results chapter, that this method of noise simulation creates the best guess of $f(\vec{d^t})$ from all these three methods.

#### Dose calculation with VMC$^{++}$

All dose calculations performed in this work were accomplished with VMC$^{++}$ (Kawrakow & Fippel 2000, Kawrakow 2001). It is one of the new generation Monte Carlo packages for treatment planning that involve a variety of variance reduction techniques (VRT) in order to increase the efficiency of the computation. With VMC$^{++}$, a typical dose calculation can be carried out in only a few minutes. In comparison, general purpose Monte Carlo codes like Geant4 (GEANT4 Collaboration 2003) or EGSnrc (Kawrakow & Rogers 2001) usually take several hours or even up to days for a dose distribution with the same mean uncertainty.

The efficiency of a Monte Carlo algorithm is defined by

$$\epsilon := (\sigma^2 t)^{-1}, \tag{2.58}$$

where $\sigma$ is the statistical uncertainty of the variable of interest and $t$ is the calculation time required to achieve this accuracy (Bielajew 2001). The efficiency of a certain MC-algorithm/implementation is a constant, because the calculation time $t$ is linear in the number of calculated particles/histories $N$ and the uncertainty $\sigma$ is proportional to $N^{-1/2}$ (see section 2.3.2). The aim of VRTs is to increase the efficiency by either reducing directly the computation time or decreasing the uncertainty of the calculation. Some of the VRTs – utilized in VMC$^{++}$ – are:

**STOPS** **S**imultaneous **t**ransport **o**f **p**article **s**ets (Kawrakow & Fippel 2000, Kawrakow 2001) speeds up the calculation by transporting particles in sets of the same energy and charge. The runtime reduction is achieved by calculating material independent quantities as "interpolation indices, azimuthal scattering angels, distances to discrete interaction, etc." (Kawrakow 2001) once for all particles in one set. Especially the runtime for multiple scattering simulation of electron tracks can be substantially reduced. Other quantities that are material dependent, such as multiple scattering angles and stopping powers, are calculated separately for each particle of a set.

**Photon splitting + Russian roulette** When a photon undergoes an interaction, the resulting photon (either scattered or created) is split into $N_{split}$ "subphotons", each carrying now a statistical weight of $1/N_{split}$. The range to the next interaction point of the original photon is partitioned onto the subphotons. Herewith, the

number of required random numbers per particle can be substantially reduced. Often combined with photon splitting is "Russian roulette". At some point, the statistical weight of particles can become very small due the splitting technique. When a sub-threshold weight photon interacts with the medium by e.g. the Compton process, the resulting secondary photon is only kept with a certain probability $\alpha$; else it is discarded. Because the energy in the system must be conserved, the particle's weight is multiplied with $1/\alpha$.

**Quasi-random numbers** Quasi-random number generators are created with the aim to distribute the sequence of random numbers as evenly as possible over the space of interest. Therefore, the calculation of the desired quantity (the dose) converges faster compared to the same simulation with a pseudo-random number generator (standard random number generator). With quasi-random numbers the dose uncertainty $\sigma$ gets smaller in comparison when calculating the dose distribution with standard random numbers, even if both simulations are carried out with the same number of particles. Thus, the efficiency increases.

The combination of these variance reduction techniques results in an efficiency increase about a factor of 5–10 (Kawrakow & Fippel 2000).

The VMC$^{++}$ package can be either used as a stand-alone program or as a library that can be linked against user programs. We decided to use the second option to combine the treatment plan optimization code with the Monte Carlo engine. In this mode, the user program passes the patient CT as a mass-density cube. Because the CT image is stored originally in Hounsfield units, it is first converted into mass-densities with Schneider's method (Schneider et al. 2000). The VMC$^{++}$ code then internally converts the mass density data into material compositions. The dose can be scored either as dose-to-water or dose-to-medium. In all calculation made for this work, we followed the advice of Ma & Li (2011) and used the dose-to-medium scoring technique.

### The particle source model

The VMC$^{++}$ Monte Carlo software package can be extended by external plug-ins. Typical plug-ins are user geometries and special particle sources. We implemented a basic particle source for a direct simulation of an IMRT fluence map. Currently, the source creates mono-energetic photons of 2 MeV. The particles emerge from a virtual photon source with a distance $d_{\mathrm{sad}}$ from the beam isocenter. The size of the virtual photon source can be adjusted by changing the variance parameter of the Gaussian-shaped source.

The initial particles (photons) are sampled from the source according to the fluence map $\vec{w}$ and the primary fluence $\psi(x, y)$. The unnormalized sampling probability of particles from beamlet $j$ can be expressed by the equation

$$p_j = w_j \bar{\psi}_j, \tag{2.59}$$

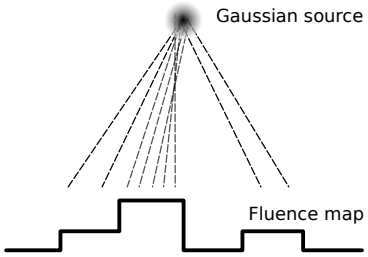



**Figure 2.7:** Illustration of the simple particle source model for fluence map simulation

where $\bar{\psi}_j$ is the mean primary fluence at beamlet $j$ and $w_j$ is its fluence weight. The particles are sampled with the Metropolis algorithm (Metropolis et al. 1953, Hastings 1970), which uses a Markov chain to model the transition probability between two beamlets. The reasons for this choice from sampling algorithms are the following:

- Sampling probabilities do not have to be normalized. Thus we can sample even from multi-dimensional probability distributions, where a normalization constant is hard to calculate. For example, we could even extend the source model to sample not even an average fluence per beamlet but the true fluence distribution, which is a 2-dimensional sampling problem.
- Fewer random numbers are wasted than in the rejection method.
- It is fast because only little computation is required.

A potential drawback of the metropolis algorithm is that the number of sampled particles per beamlet converges only for a large total particle number against the probability distribution, defined by the fluence map. That is, the dose calculation of a fluence map $\vec{w}$ gets an additional uncertainty due to a possibly incorrect fluence map sampling, when simulating too few particles. This uncertainty is reduced, if one ensures that the number of simulated particles is significantly larger than the number of beamlets. Typical numbers of simulated particles with VMC$^{++}$ are at least $2 \cdot 10^5$ and go up to $5 \cdot 10^7$ so that this premise is true in most cases.

## 2.4 A reference FMO algorithm for MC based dose calculations

The Monte Carlo dose calculation algorithm is implemented into the standard IMRT optimization framework. VMC$^{++}$ is modified to allow a separate dose scoring for each particle source and thus to calculate the dose influence matrix based on the patient CT and the beam geometry. The number of particles in the simulation has to be set in the input script. These number of particles then are equally partitioned onto all beamlets, i.e. the same number of particles are simulated from each beamlet. The resulting dose influence matrix is stored in two KonRad-compatible *.dij files, one for the dose values and one for the dose uncertainties.

To reduce the number of dose elements to store in the dose influence matrix, we use a dose-dependent sampling strategy (Thieke et al. 2002). This method determines, whether a dose value is inserted into the dose influence matrix or not. First, the dose values of voxels are stored that lie in a cylinder of radius $r_1$, which is centered around the central axis of the beamlet. On the other hand, all dose values from voxels that lie outside a cylinder of radius $r_2 > r_1$ are discarded. For each voxel $i$ that lies in between both cylinders, a probability $p_i$ is calculated that determines if the value has to be stored or not:

$$p_i := d_i / d(r_1) \tag{2.60}$$

In this equation $d(r_1)$ is the dose at the point where the line from the central axis to the current voxel $i$ intersects the cylinder of radius $r_1$. The dose value $d_i$ is stored if a random number $r \in [0, 1]$ is smaller than $p_i$. To conserve energy, each dose value is divided by $p_i$ before storage. For our 2 MeV mono-energetic photon source (see section 2.3.2), the radii $r_1 = 3\,\text{cm}$ and $r_2 = 7\,\text{cm}$ are a good compromise between accuracy and memory requirement.

We will see later that the dose influence matrix requires a minimal statistical accuracy in order to get stable optimization results. To characterize the accuracy of the dose influence matrix, we introduce the average dose uncertainty per beamlet $\bar{\sigma}_{\text{bix}}$. It is defined by the arithmetic mean of the mean uncertainties of the dose distribution of each beamlet:

$$\bar{\sigma}_{\text{bix}} := N_{\text{bix}}^{-1} \sum_j \bar{\sigma}_j, \quad \text{with} \tag{2.61}$$

$$\bar{\sigma}_j := \sqrt{\frac{1}{N_{50}^j} \sum_{D_{ij} > 50\%\, d_j^{\max}} \left( \frac{\Delta D_{ij}}{d_j^{\max}} \right)^2} \tag{2.62}$$

In these equations, $d_j^{\max}$ is the maximum dose value of beamlet $j$ and $\Delta D_{ij}$ represents the dose uncertainty in voxel $i$ from beamlet $j$. Also, $N_{50}^j$ is the number of voxels whose dose value is larger than $0.5 \cdot d_{\max}^j$ and $N_{\text{bix}}$ is the total number of beamlets. This average dose uncertainty per beamlet agrees with the general definition of the mean uncertainty of a dose distribution (2.70).

The precalculated matrix is then used by KonRad or our optimization framework for inverse treatment planning. To estimate the error of the resulting dose distribution, we apply the error propagation law to (2.23):

$$(\Delta d_i)^2 = \sum_j (\Delta D_{ij} w_j)^2 \tag{2.63}$$

The uncertainty of the objective function is estimated with the simulated noise method, which was presented in section 2.3.2.

### 2.4.1 Why the reference FMO algorithm is inefficient

The precalculation of the dose influence matrix is elegant, because it decouples the dose calculation from the optimization. When dealing with MC-based dose calculations, there are however some problems with this approach:

- Each dose value has its uncertainty. Thus, the objective function is uncertain too. In order to limit the convergence error of the optimization, the error of the objective function has to be small enough. But before entering the optimization loop, it cannot be estimated how many particles are required for the calculation of the dose influence matrix to get a defined uncertainty of the objective function in the end.
- In a forward MC dose calculation, the number of particles simulated by each beamlet is proportional to its beamlet weight. Especially beamlets that were closed by the optimizer (weights are zero) are completely ignored in the simulation. This sampling strategy is optimal in terms of uncertainty reduction, because the dose error is dominated by dose contributions of beamlets with large weight $w_j$ (as implied by (2.63)). When precalculating the dose influence matrix however, the number of particles per beamlet is either constant or adjusted to get a fixed dose uncertainty per beamlet (Jeraj & Keall 1999, Bergman et al. 2006, Siebers 2008) because the optimal fluence map is not previously known. Thus, compared to the forward dose calculation, too many particles are simulated from small weighted beamlets and are therefore wasted. As a result, the standard FMO approach with a precalculated dose influence matrix requires much more particles and thus computation time than necessary.

### 2.4.2 Efficiency of a MC based optimization algorithm

To quantify, how a MC based optimization algorithm performs in comparison to a forward MC dose calculation, we define the optimization efficiency similarly as in (Laub et al. 2000). Suppose the resulting dose distribution – which was created by an optimization algorithm – has a mean dose uncertainty $\bar{\sigma}_{\text{opt}}$ after simulating $N_{\text{opt}}$ particles.

Now suppose that the simulation of the same fluence map with a forward MC dose calculation requires $N_{\mathrm{fw}}$ particles to achieve the same mean dose uncertainty. Then the optimization efficiency $\varepsilon$ is defined as the quotient of the particle numbers

$$\varepsilon := \frac{N_{\mathrm{fw}}}{N_{\mathrm{opt}}}. \tag{2.64}$$

This efficiency value can be even calculated if the mean dose uncertainties of the forward dose calculation $\bar{\sigma}_{\mathrm{fw}}$ and the dose result of the optimization $\bar{\sigma}_{\mathrm{opt}}$ differ. Because the mean dose uncertainty decreases with $N^{-1/2}$ (see equation (2.48)), we can estimate the number of particles to simulate in order to achieve another mean dose uncertainty. The number of particles in a forward simulation that are required to achieve the same dose uncertainty as the optimization is given by:

$$N'_{\mathrm{fw}} = N_{\mathrm{fw}} \left( \frac{\bar{\sigma}_{\mathrm{fw}}}{\bar{\sigma}_{\mathrm{opt}}} \right)^2 \tag{2.65}$$

Hence, the general form of the optimization efficiency gets

$$\boxed{\varepsilon = \frac{N_{\mathrm{fw}}\, \bar{\sigma}^2_{\mathrm{fw}}}{N_{\mathrm{opt}}\, \bar{\sigma}^2_{\mathrm{opt}}}}. \tag{2.66}$$

Because each particle has the same statistical weight in the forward dose calculation and contributes maximally to the decrease of the calculation uncertainty, the forward dose calculation by MC simulation is the reference in terms of efficiency. Therefore optimization efficiency values will be always smaller than 1. For the standard IMRT cycle with a precalculated dose influence matrix, efficiency values of only 0.1–0.3 where reported (Laub et al. 2000). This implies, that the reference fluence map optimization method requires up to ten times more particles than a forward dose calculation of the optimal fluence map.

## 2.5 A hybrid sequential algorithm for Monte Carlo based plan optimization

In this section, we introduce an alternative to the reference algorithm that tries to avoid the explained efficiency problems. The aim of this algorithm is to achieve clinically relevant calculation times of only a few minutes in combination with the fast VMC$^{++}$ code. This is done by minimizing the number of MC dose calculations and additionally decreasing the dose uncertainty for each voxel in every iteration of the optimization. The first aspect is addressed in section 2.5.1, where we calculate an optimized search direction that allows the algorithm to walk to the optimum in only a few iterations. The second aspect makes a dose recalculation after the optimization phase dispensable as the resulting dose distribution will be indistinguishable from a forward calculated MC dose

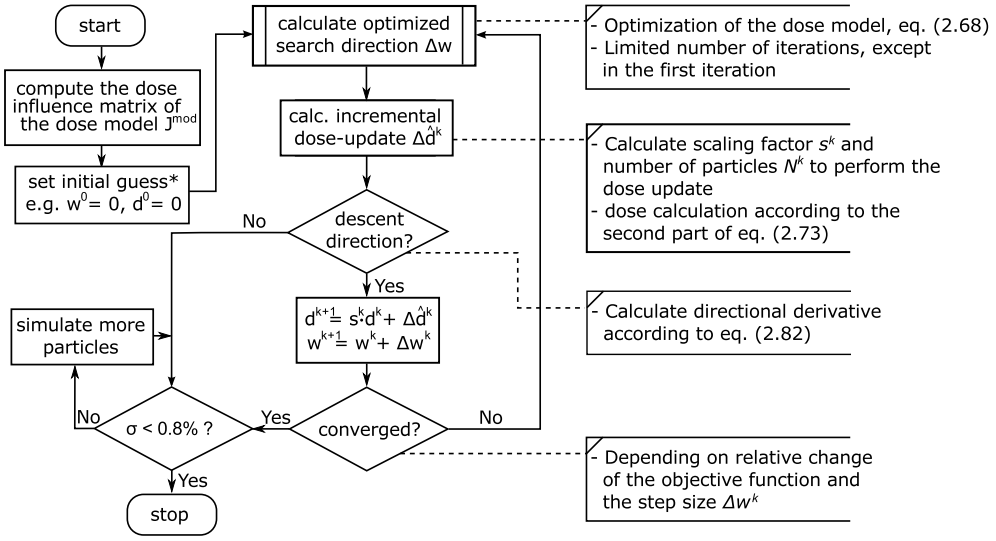



**Figure 2.8:** Flow diagram of the hybrid optimization algorithm. Explanations to the steps of the algorithm can be found on the right side of the figure. *The initial guess of the fluence weights can be either zero or a result of a previous fluence map optimization using the dose model matrix $J^{\mathrm{mod}}$.

distribution. Also, this dose update, which is executed in every iteration, is designed to reuse already simulated particles as much as possible – i.e. it minimizes the "waste" of particles. We call this strategy the *incremental dose update* and examine it in section 2.5.2. An overview of the algorithm is shown in figure 2.8.

### 2.5.1 Optimized search direction

In (Laub et al. 2000), the calculation of the gradients $\vec{\nabla} f_w(\vec{w}^k) = \vec{\nabla}_d f(\vec{d})^T J_d$ is approximated by calculating the dose influence matrix $J_d$ with a less accurate pencil beam algorithm. Therefore, a time consuming computation of this matrix with a MC dose algorithm is avoided. The dose calculation itself, which is required for the evaluation of the objective function, is purely Monte Carlo-based throughout the optimization. The change of the fluence map $\Delta\vec{w}^k$ is determined by a conjugate gradient calculation (Hestenes & Stiefel 1952) in each iteration. Still, if the dose differences between the pencil beam algorithm and the MC result are large, this optimization algorithm needs a relatively high number of iterations to converge and therefore requires also a large number of MC simulations.

Here, we propose a sequential optimization approach for the calculation of the search direction that allows a faster convergence, without knowing the accurate dose influence matrix. Inspired by Newton's method/Sequential Quadratic Programming, the search direction $\Delta\vec{w}^k$ is a solution of a minimization problem in each iteration $k$: first, the

dose is modeled around the current dose distribution $\vec{d}(\vec{w}^k)$ with a linear hybrid dose approximation:

$$\vec{d}^{\,\mathrm{mod}}_{w^k}(\Delta\vec{w}^k) := \vec{d}(\vec{w}^k) + J_d^{\mathrm{mod}}\Delta\vec{w}^k \approx \vec{d}(\vec{w}^k + \Delta\vec{w}^k) \tag{2.67}$$

Here, the dose influence matrix $J_d^{\mathrm{mod}}$ is calculated with a simpler dose calculation algorithm, for example with pencil beams. Then, the search direction $\Delta\vec{w}^k$ is given by minimizing the objective function (2.24) of this dose model – i.e. by solving the following equation:

$$\boxed{\min_{\Delta w^k} f(\vec{d}^{\,\mathrm{mod}}_{w^k}(\Delta\vec{w}^k)) := \min \sum_i p_i \left( \left[ J_d^{\mathrm{mod}}\Delta\vec{w}^k \right]_i + d(\vec{w}^k)_i - d_i^p \right)^2 \text{ s.t. } \vec{w}^k + \Delta\vec{w}^k \geq 0} \tag{2.68}$$

This optimization tries to answer the following question: given the current MC-based dose distribution $\vec{d}(\vec{w}^k)$, what dose distribution should be added or subtracted in order to get an optimal dose distribution with respect to the objective function? This additional dose distribution is estimated with the model dose distribution $J_d^{\mathrm{mod}}\Delta\vec{w}^k$. If the Monte Carlo generated dose distribution $\vec{d}(\vec{w}^k)$ is already optimal, no additional dose is required and the resulting search direction $\Delta\vec{w}^k$ will be zero. Then, the optimization stops.

For the optimization of the dose model we use the limited-memory BFGS algorithm presented in section 2.1.3. This optimization can be halted after a certain number of iterations because optimality is not necessary in order to find a good search direction. Due to the super-linear convergence of the BFGS method (Broyden 1970, Fletcher 1970, Goldfarb 1970, Shanno 1970, Liu & Nocedal 1989) we let the optimization run for maximum 30 iterations.

With the calculated search direction $\Delta\vec{w}_k$ the fluence map is update according to $\vec{w}^{k+1} = \vec{w}^k + \alpha\Delta\vec{w}^k$. Here, the step-length $\alpha$ is determined by a line search that minimizes $f(\vec{d}(\vec{w}^k)+\alpha\vec{d}(\Delta\vec{w}^k))$, subject to $\vec{w}^k+\alpha\Delta\vec{w}^k \geq 0$. It should be noted that the information from the dose model is only used to predict a good search direction. The dose calculation for the current fluence map is always done by Monte Carlo simulation at the end of each iteration. Because $\Delta\vec{w}^k$ is predicted by a less accurate dose model, it cannot be guaranteed that it will be a descent direction of $f$ i.e. that the directional derivative $\vec{\nabla} f(\vec{w}^k)^T\Delta\vec{w}^k$ is negative. If it is not negative, the optimization algorithm has to be halted to avoid loops in the optimization. However, if the dose model (2.67) does not differ severely from the Monte Carlo-simulated dose distribution, this case might only occur in the terminal phase of the optimization.

Formally it can be shown, that our method of the optimized search direction can be transformed into the hybrid optimization algorithm of Siebers et al. (2007) if the line search parameter is fixed at $\alpha = 1$. Sieber's method repeatedly optimizes a pencil beam dose distribution plus a correction term, which includes the differences of the pencil beam dose distribution and the MC dose distribution from the previous run. In this work a more classical optimization approach is chosen, where the change of the fluence

weights is determined by a search direction. The advantage of this approach is that many optimization concepts can be directly applied to our method. These concepts include for example the line search and the downhill direction check.

### 2.5.2 Efficient incremental dose update

A Monte Carlo dose calculation samples the particles according to their probability distribution. When simulating fluence maps directly this implies that the number of simulated particles $N_j$ of a beamlet is proportional to its fluence weight:

$$N_j = \varrho\, \bar{\psi}_j w_j \tag{2.69}$$

Here, $\bar{\psi}_j$ is the mean primary photon fluence of beamlet $j$, $w_j$ is its fluence map value, and $\varrho$ is some constant which can be interpreted as the density of particles per unit weight. To be efficient, an optimization technique for Monte Carlo dose engines should utilize this optimal sampling strategy that minimizes the number of required particles to simulate, given a desired mean uncertainty of the dose distribution. As a measure for the mean dose uncertainty we use the definition of Kawrakow (2001):

$$\bar{\sigma} := \sqrt{\frac{1}{N_{50}} \sum_{d_i > 50\%\, d_{\max}} \left(\frac{\Delta d_i}{d_{\max}}\right)^2} \tag{2.70}$$

Here, $\Delta d_i$ is the absolute dose uncertainty in voxel $i$ and $N_{50}$ is the number of voxels whose dose values $d_i$ are larger than 50 % of $d_{\max}$.

In each iteration $k$, the optimized search direction (section 2.5.1) returns a proposed change of the fluence map; hence the fluence map is updated according to $\vec{w}^{k+1} = \vec{w}^k + \Delta\vec{w}^k$. To keep efficiency high, the following strategy is pursued: As proposed by Laub et al. (2000), the changes of fluence are handled by the algorithm by simulating additional particles, i.e. only the dose of the fluence update $\vec{d}(\Delta\vec{w}^k)$ is calculated. This technique reuses previously simulated particles and fewer particles are therefore required in each iteration. Assuming $\Delta\vec{w}^k \geq 0$, the number of particles required for the dose update calculation is $\Delta N^k = \varrho \sum_j \bar{\psi}_j \Delta w_j^k$. The particle density $\varrho$ has to be the same as in (2.69). Let $\tilde{w}_j := \bar{\psi}_j w_j$ be the effective beamlet weight and $N^k = \sum_j N_j^k$ be the total number of particles simulated for the dose distribution $\vec{d}(\vec{w}_k)$ of the current iteration, then we can rewrite this relation as

$$\Delta N^k = N^k \frac{\sum_j \Delta\tilde{w}_j^k}{\sum_j \tilde{w}_j^k}. \tag{2.71}$$

The final dose distribution for the current iteration is the sum of the previous dose distribution and the dose update:

$$d(\vec{w}^{k+1}, N^k + \Delta N^k) = \vec{d}(\vec{w}^k, N^k) + \vec{d}(\Delta\vec{w}^k, \Delta N^k) \tag{2.72}$$

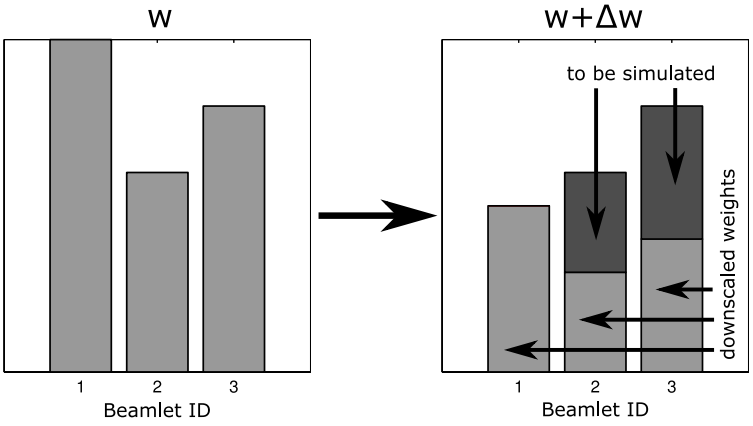



**Figure 2.9:** Simulating negative weight updates by downscaling and compensation

This presented dose update will only maintain high efficiency if the fluence update $\Delta\vec{w}^k$ is positive, because it is not possible to remove particles from a previous simulation. To take account of negative changes in the fluence map, particles from such beamlets are simulated with a negative statistical weight to remove dose. The sampling probability for these particles then has to be $|\Delta w_j^k|$. Therefore, (2.71) changes to $\Delta N^k = N^k \left(\sum_j |\Delta\tilde{w}_j^k|\right)\left(\sum_j \tilde{w}_j^k\right)^{-1}$. With this strategy however, too many particles are simulated for beamlets with a negative weight change in combination with the previous calculation. Thus, a fraction of particles from these beamlets are wasted and the particle efficiency decreases. This implies that the density of particles per beamlet is not constant anymore but is increased in these beamlets. To overcome this limitation and maintain high efficiency, the current dose $\vec{d}(\vec{w}^k)$ is scaled down by a factor $s^k$ and the dose update has to compensate for the scaling (see figure 2.9). With this strategy the dose update formula gets its final form

$$\boxed{\vec{d}(\vec{w}^{k+1}, N^k + \Delta N^k) = s^k \cdot \vec{d}(\vec{w}^k, N^k) + \vec{d}\left(\left[1 - s^k\right]\vec{w}^k + \Delta\vec{w}^k, \Delta N^k\right), \quad s^k \in (0,1]}. \tag{2.73}$$

Due to the down-scaling, the particle density per unit weight increases by $1/s^k$ and gets $\varrho = N^k \left(s^k \sum_j \tilde{w}_j^k\right)^{-1}$. Depending on the amount of the scaling $s^k$, the mean dose uncertainty decreases with each dose update during the optimization. To take account for the scaling, the number of particles for the dose calculation of the compensated weight-change (second part of (2.73)) gets:

$$\boxed{\Delta N^k = N^k \frac{\sum_j \left|(1-s^k)\tilde{w}_j^k + \Delta\tilde{w}_j^k\right|}{s^k \sum_j \tilde{w}_j^k}} \tag{2.74}$$

### Specifying the amount of down-scaling

It seems intuitive, that the scaling factor is determined by the beamlet weight that gets the largest relative (negative) change. However, there are situations, when the optimizer puts one (or more) weight down to zero i.e. if $w_i^k + \Delta w_i^k = 0$. In this case, the scaling factor $s^k$ would be zero; the result of the previous dose calculation could not be reused. Fortunately, because of the nature of the optimization algorithm, this only happens for beamlets that had a relatively small weight before and thus had only minor contribution to the total dose distribution. Therefore, we will concentrate only on beamlets with "large" weight and negative change. Up to now, only a few particles were simulated for beamlets with small weight. Therefore we can ignore if some of these few particles are wasted. Hence, the decision about $s^k$ depends only on the subset of all beamlets defined by

$$\hat{B}^k := \left\{ j | w_j^k \geq \delta w_{\max} \wedge \Delta w_j^k < 0 \right\}, \quad w_{\max} := \max_j(w_j^k), \quad \delta \in (0,1), \tag{2.75}$$

where $w_{\max}$ is the largest element of the fluence map. In our implementation we us a cut-off value of $\delta = 0.4$. Given the subset $\hat{B}^k$ we get for the scaling factor

$$\boxed{s^k = \min_{j \in \hat{B}^k} \left( \frac{w_j^k + \Delta w_j^k}{w_j^k} \right)}. \tag{2.76}$$

If $\hat{B}$ is empty, e.g. if the weight change is positive for all beamlets, $s$ will be set to one.

### Limits for the scaling factor

Generally it is advisable to define minimum and maximum bounds for the scaling factor $s$ to have control over the number of particles to simulate. The minimum bound $s_{\min}$ limits the number of particles for the dose update, because a very small value of $s$ implies a much increased particle density per weight. Similarly, a maximum bound $s_{\max}$ can be utilized to reduce the mean dose uncertainty in every iteration. Therefore, this upper bound is determined by the minimum number of particles and the lower bound is determined by the maximum number of particles we want to simulate for the dose update calculation. Given the minimum number of particles $\Delta N_{\min}$ and maximum number of particles $\Delta N_{\max}$ acceptable for the dose update, the bounds for $s$ can be derived as follows:

**1)** $s_{\min}$**:** We have to enforce, that $\Delta N \leq \Delta N_{\max}$:

$$\begin{aligned}
\Delta N &= N\frac{\sum_j |(1-s)\tilde{w}_j + \Delta\tilde{w}_j|}{s\sum_j \tilde{w}_j} \\
&\leq N\frac{\sum_j (1-s)\,|\tilde{w}_j| + \sum_j |\Delta\tilde{w}_j|}{s\sum_j \tilde{w}_j} \\
&\overset{\tilde{w}\geq 0}{=} \frac{N}{s}\left[\frac{\sum_j |\Delta\tilde{w}_j|}{\sum_j \tilde{w}_j} + 1\right] - N \\
&=: \Delta N_{\max}
\end{aligned}$$

$$\Rightarrow \boxed{s_{\min} = \left[\frac{\sum_j |\Delta\tilde{w}_j|}{\sum_j \tilde{w}_j} + 1\right] \cdot \left[\frac{\Delta N_{\max}}{N} + 1\right]^{-1}} \tag{2.77}$$

**2)** $s_{\max}$**:** We have to enforce, that $\Delta N \geq \Delta N_{min}$:

$$\begin{aligned}
\Delta N &= N\frac{\sum_j |(1-s)\tilde{w}_j + \Delta\tilde{w}_j|}{s\sum_j \tilde{w}_j} \\
&\geq N\frac{\sum_j (1-s)\tilde{w}_j + \sum_j \Delta\tilde{w}_j}{s\sum_j \tilde{w}_j} \\
&= \frac{N}{s}\left[\frac{\sum_j \Delta\tilde{w}_j}{\sum_j \tilde{w}_j} + 1\right] - N \\
&=: \Delta N_{\min}
\end{aligned}$$

$$\Rightarrow \boxed{s_{\max} = \left[\frac{\sum_j \Delta\tilde{w}_j}{\sum_j \tilde{w}_j} + 1\right] \cdot \left[\frac{\Delta N_{\min}}{N} + 1\right]^{-1}} \tag{2.78}$$

In our implementation we use the fixed values $\Delta N_{\max}/N = 1.5$ and $\Delta N_{\min}/N = 0.05$ as particle limits.

### 2.5.3 Line search with the incremental dose update approach

Because FMO is a box-constraint optimization problem with a lower bound of $\vec{l} = 0$, the adequate line search (section 2.1.4) minimizes $f(P(\vec{w} + \alpha\Delta\vec{w}))$. Here $P$ is the projection operator that was already defined in that section. It ensures that the objective function is evaluated only in the allowed region $\Omega := \{\vec{w} \in \mathbb{R}^n | \vec{w} \geq 0\}$. Due to the definition of the optimized search direction, the possibly new weights $\vec{w} + \Delta\vec{w}$ are feasible too. Because $\vec{w}$ is feasible and the solution space $\Omega$ is convex, each vector on the straight line between $\vec{w}$ and $\vec{w} + \Delta\vec{w}$ is also feasible; that is, $(\vec{w} + \alpha\Delta\vec{w}) \in \Omega, \alpha \in [0, 1]$ and $P$ gets

the identity operator. This property can be exploited by a linear decomposition of the dose:

$$f(\vec{d}(P(\vec{w}+\alpha\Delta\vec{w}))) \overset{\alpha\in[0,1]}{=} f(\vec{d}(\vec{w}+\alpha\Delta\vec{w})) = f(\vec{d}(\vec{w})+\alpha\vec{d}(\Delta\vec{w})) \tag{2.79}$$

In this case the evaluation comes basically for free because both dose distributions $\vec{d}(\vec{w})$ and $\vec{d}(\Delta\vec{w})$ were calculated before. Because of the nature of the weight update calculation there is a high chance that $P$ is not linear any more for $\alpha > 1$ due to a violation of the lower bound constraint. As a result, the upper decomposition cannot be performed and the line search evaluation is very expensive. Therefore we limit the line search parameter $\alpha \in [0,1]$.

In the case of the dose down-scaling, the upper evaluation has to be altered again because a modified update dose $\hat{\Delta}\vec{d}^k := \vec{d}(\Delta\vec{w}^k + [1-s^k]\vec{w}^k)$ is simulated instead of $\vec{d}(\Delta\vec{w}^k)$. If we exploit the linearity of the dose with respect to the fluence map, we can write

$$\begin{aligned}
\vec{d}(\vec{w}^k+\alpha\Delta\vec{w}^k) &= \vec{d}(\vec{w}^k)+\alpha\vec{d}(\Delta\vec{w}^k) \\
&= \vec{d}(\vec{w}^k)-\alpha[1-s^k]\vec{d}(\vec{w}^k)+\alpha[1-s^k]\vec{d}(\vec{w}^k)+\alpha\vec{d}(\Delta\vec{w}^k) \\
&= (1+\alpha[s^k-1])\vec{d}(\vec{w}^k)+\alpha\vec{d}(\Delta\vec{w}^k+[1-s^k]\vec{w}^k) \\
&= \underbrace{(1+\alpha[s^k-1])}_{s_{\text{eff}}}\vec{d}^k+\alpha\hat{\Delta}\vec{d}^k.
\end{aligned} \tag{2.80}$$

This result also allows a computational cheap evaluation of the line search for the modified dose update. The factor $(1+\alpha[s-1])$ can be understood as the effective down-scaling factor $s_{\text{eff}}$, which is in the range $[s,1]$ for $\alpha \in [0,1]$.

As a line search strategy we use the Armijo-backtracking approach (algorithm 3) with $\alpha_{\max} = 1$. Due to the down-scaling of the dose distribution by the factor $s^k$ for the incremental dose update, the Armijo-condition (2.15) changes to

$$\boxed{f\left(\left[1+\alpha(s^k-1)\right]\vec{d}^k+\alpha\hat{\Delta}\vec{d}^k\right) \le f(\vec{d}^k)+\alpha c_1\vec{\nabla}f_d(\vec{d}^k)^\top\left(\hat{\Delta}\vec{d}^k+[s^k-1]\vec{d}^k\right)}. \tag{2.81}$$

This line search is specifically necessary in the terminal phase of the optimization in which the systematic errors of the dose model start to affect the calculation of the optimized search direction.

### 2.5.4 Details of the algorithm

In addition to standard convergence criteria – i.e. change of the objective function and evaluation of the step size – two additional checks are introduced: As mentioned in section 2.5.1, the search direction $\Delta\vec{w}$ may not be a descent direction. The directional derivative of the objective function at the current dose $\vec{d}^k$ along the search direction $\Delta\vec{d}(\Delta\vec{w}^k)$ is given by

$$\begin{aligned}
\vec{\nabla}_{\Delta\vec{w}^k} f(\vec{w}^k) &= \vec{\nabla}_d f(\vec{d}^k)^\top \vec{d}(\Delta\vec{w}^k) \\
&\approx \vec{\nabla}_d f(\vec{d}^k)^\top\left(\Delta\hat{\vec{d}}-[1-s^k]\vec{d}^k\right),
\end{aligned} \tag{2.82}$$

where $\Delta\hat{\vec{d}} = \vec{d}\left(\left[1 - s^k\right]\vec{w}^k + \Delta\vec{w}^k, \Delta N^k\right)$ is the dose distribution of the modified dose update calculation and $s^k$ is the down-scaling parameter (see second part of (2.73)). To avoid loops in the optimization, the algorithm has to be halted if this relation is not negative. A second additional check determines how much the dose model (2.67) can reduce the objective function value. If the predicted reduction using the dose model is too small, convergence is reached. The reason for this check is the following: First, a further (unnecessary) MC simulation can be avoided. Second, in addition to the method of simulated noise (see section 2.3.2) this check provides a hint whether the reduction of $f$ is made by the change of weights $\Delta\vec{w}$ or if it is just an artifact due to the decreased dose uncertainty.

The complete optimization algorithm, which is outlined in figure 2.8, works as follows: First, the dose influence matrix $J_d^{\text{mod}}$ is calculated with a model dose calculation algorithm. Details to possible algorithms are given in the next section. Then, a starting solution is generated by optimizing the corresponding dose distribution. Second, the initial MC dose distribution $\vec{d}^0$ of this starting solution is calculated with a mean relative dose uncertainty of $\bar{\sigma}_{50} = 1.5\%$ (as defined in (2.70)). Third, the optimization loop is executed until convergence. Finally, if the mean dose uncertainty is still larger than $\bar{\sigma}_{50} = 0.8\%$, additional particles will be simulated to decrease the dose uncertainty to this level. The number of additionally required particles are estimated with equation (2.48).

## 2.6 Dose models for the hybrid optimization algorithm

The calculation of the search direction $\Delta\vec{w}$ in the hybrid algorithm (section 2.5.1) requires an approximation of dose influence matrix $J_d$, which stores the normalized dose contribution of each beamlet to the patient. This matrix represents the Jacobian of the dose function with respect to the fluence map and is required for calculating the gradients of the hybrid dose model (2.67) during the optimization. An ideal dose model would be very accurate and also fast to calculate. Unfortunately, there is no such algorithm and a trade-off between accuracy and calculation time has to be made.

In this work we chose two different models, the *macroscopic pencil beam* algorithm and the *geometric kernel approximation*.

### 2.6.1 Macroscopic pencil beam

In our research version of the KonRad inverse planning system this method is also known as the external pencil beam algorithm (Bortfeld et al. 1993, Nill et al. 2004, Siggel et al. 2011). It uses a precalculated water-dose distribution caused by a quadratic field of the size of one beamlet. To remove most of the dependency of the dose on the source-surface-

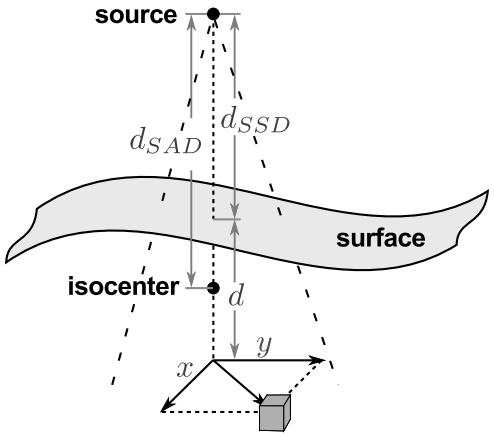



**Figure 2.10:** Scheme for the dose calculation with the macroscopic pencil beam algorithm.

distance (SSD), this precomputed pencil kernel is stored in a non-divergent geometry. It can be calculated by either simulating a parallel beam or converting the dose distribution from a divergent beam into a parallel geometry.

To calculate a beamlet's dose distribution from arbitrary beam angles, SSDs and patient geometries, the following equation is used:

$$D(x,y,d) = \left(\frac{d_{SAD}}{d+d_{SSD}}\right)^2 D^{||}\left(x\frac{d_{SAD}}{d+d_{SSD}}, y\frac{d_{SAD}}{d+d_{SSD}}, d_{rad}\,(x,y,d)\right) \tag{2.83}$$

The first part of this equation is the distance-square-dependency of radiation from point sources and takes account for the divergent beam geometry. Here, $d$ is the depth of the calculation point inside the patient, $d_{SAD}$ is the distance from the particle source to the beam's isocenter and $d_{SSD}$ is the source-surface-distance. The sum $d_{SSD}+d$ corresponds to the geometric distance from the source to the calculation point. The term $D^{||}$ is the precalculated dose in a parallel geometry at a lateral distance $x$ and $y$ from the beam center. To incorporate a homogeneity correction into the algorithm, the radiological depth $d_{rad}$ is used in the upper equation. It is defined by the path integral from the source $S$ to the calculation point $P$

$$d_{rad} := \int_S^P \rho(x)dx, \tag{2.84}$$

where $\rho(x)$ is the relative electron density (in relation to the electron density of water) at position $x$. In order to calculate this integral, ray-tracing (Siddon 1985, Fox et al. 2006) has to be performed that returns all voxels and intersection lengths on that path. To reduce artifacts due to the voxel discretization of the precalculated pencil beam, the dose values $D^{||}$ are calculated by a trilinear interpolation.

The external pencil beam is precalculated once with the VMC$^{++}$ program using the mentioned basic 2 MeV particle source. To simulate a parallel beam, the source-isocenter-

distance is set to 100 m. To support different leaf-widths, pencil beams of 5 mm and 10 mm beamlet sizes are generated. In order to avoid noise artifacts, the large number of $5 \cdot 10^6$ particle histories were simulated that results in a mean dose uncertainty of 0.255 % (10 mm beamlet size) and 0.133 % (5 mm beamlet size). The resulting dose cubes were further post-processed by averaging over the symmetric quadrants of the dose distributions. The voxel size of the external pencil beam is 1 mm in x and y direction (perpendicular to the beam) and 3 mm along the z axis (parallel to the beam).

These precalculated pencil beams are eventually used by KonRad to calculate and store the dose model matrix $J_d^{pb}$.

### 2.6.2 Geometric kernel approximation

When applying only one treatment beam to a patient or a phantom, the dose at a voxel inside the beam consists mainly of the primary dose deposition by a single beamlet that geometrically lies between the voxel and the irradiation source. A relatively small part of the dose comes from other beamlets due to photon and electron scattering. Thus, when having $N$ different beam directions, the main dose in a voxel comes from maximum $N$ beamlets as long as the voxel is geometrically inside the beams-eye-view of each beam.

We can use this property, to create a rough approximation of the dose influence matrix. Let $d_i^k$ be the dose in voxel $i$ caused by the $k$-th beam. Then the dose matrix $J_d^{gk}$ is given by:

$$J_{ij}^d = \begin{cases} 0, \text{ if voxel } i \text{ does not lie geometrically “under” beamlet } j \\ d_i^k, \text{ if voxel } i \text{ is “under” beamlet } j \text{ that belongs to the beam } k \end{cases} \tag{2.85}$$

The sparsity of the resulting dose matrix (few non-zero elements) offers a great advantage in computation speed during optimization compared to the pencil beam method. In order to create this dose influence matrix, a Monte Carlo simulation for each beam direction has to be performed with a normalized fluence profile. That is each beamlet weight $w_j$ is set to one. Because the matrix $J_d^{gk}$ is only used to estimate a good search direction $\Delta\vec{w}$, a high precision calculation is not necessary. Hence, the number of simulated particles can be relatively small compared to the number of particles required for the dose calculation during the optimization.

An advantage of this method is that the magnitude of the voxel's dose is estimated very well because of the accuracy of the Monte Carlo method. However, it should be noted that this approximation removes all scattering information and that the dose values in the matrix are overestimated. As a result, synergistic dose scattering summation cannot be exploited and the dose of voxels outside the beams cannot be optimized properly.

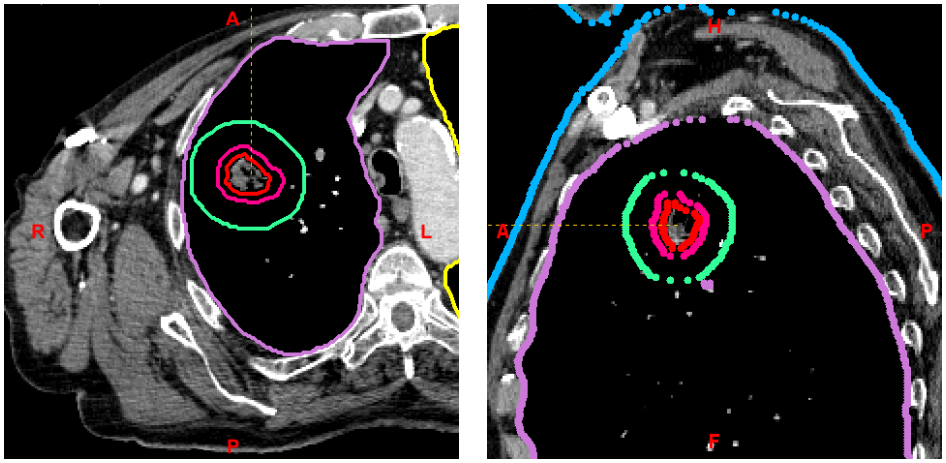
## 2.7 Clinical evaluation of optimization results, patient cases

We demonstrate the hybrid optimization algorithm on different patient geometries with both dose models. To quantify the performance of this algorithm, all plans are additionally optimized with the reference FMO method. In this case, the dose influence matrix is precalculated by the Monte Carlo dose engine using a constant number of particles per beamlet. For a fair comparison, both optimization methods use the same dose prescriptions and penalties.

Because the full FMO method represents the benchmark for the optimization, the dose distributions from both methods are compared by inspecting dose-volume histograms (DVHs), dose slices, mean doses $\bar{d}$, median dose values $d_{50}$ and maximum doses $d_{\max}$ for each organ of interest. From the mathematical point of view the final objective function values of the optimization algorithms are compared, which serve as a measure of plan quality. The fulfillment of the clinical constraints is checked for all plans.

### 2.7.1 Lung

The most challenging case for the hybrid optimization algorithm are lung treatment plans. Due to strong tissue heterogeneities and air cavities, the dose distributions calculated by pencil beam and the Monte Carlo algorithms can differ by a large amount and it has been shown that pencil beam dose algorithms significantly overestimate the tumor and lung dose at these treatment sites (Scholz et al. 2003, Krieger & Sauer 2005). Therefore a larger number of iterations during the optimization can be expected when using the pencil beam dose model.




(a) (b)

**Figure 2.11:** Transversal and sagittal isocentric slices of the lung case.

This particular example is a 10-beam non-coplanar irradiation setup, where the target is centered in the right lung and is therefore surrounded by low-density tissue. The spherical PTV (magenta) has a diameter of 29 mm and consist of the GTV (red) plus a 4 mm margin. In order to avoid hot spots inside the healthy lung tissue, a shell (green) with a margin of 14 mm around the PTV is added. In the optimization, this shell is handled as an organ at risk with a maximum dose of 30 Gy. The definition of the prescription dose for the PTV was difficult due to the following reason: in the original plan (which would be calculated with a pencil beam algorithm), GTV and PTV have both the same dose prescription of 57 Gy. The PTV margin consists of low density lung material. If PTV and GTV would be irradiated with a homogeneous photon fluence, the tumor (GTV) absorbs a higher dose than the PTV margin due to its higher mass density (if we use dose-to-medium scoring). Thus, in order to provide the same dose at the PTV margin, the weights from fluence elements at the field boundaries have to be increased. However, this increase of the beamlet weights causes hot spots behind the beam entry in the patient. Also, the dose in the PTV margin is only the average dose in each voxel whereas potential microscopic tumors will likely absorb a higher local dose. The same happens when the GTV moves inside the PTV margin because of breathing. Due to these issues, a slightly reduced dose prescription for the PTV of 54 Gy was chosen.

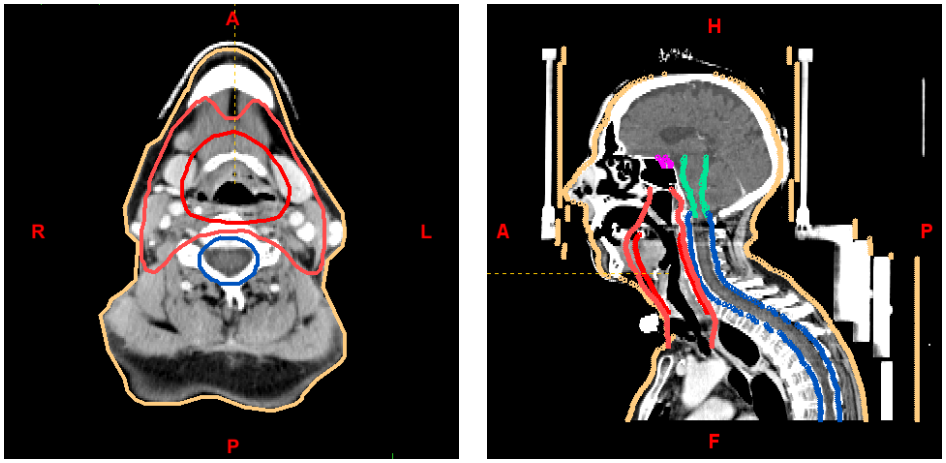
**Table 2.1:** Clinical goals for the lung treatment plan.

| Volume | Clinical Goal |
|---|---|
| GTV | $d_{50} = 57\,\mathrm{Gy}$ |
| PTV\GTV | $d_{50} = 54\,\mathrm{Gy}$ |

### 2.7.2 Nasopharynx

The irradiation of the nasopharynx is a typical head-and-neck IMRT treatment plan with 9 coplanar beams. The plan contains an integrated boost concept with a high dose to the tumor of the nasopharynx and a lower dose to the surrounding lymph-drains. The target volume including the boost and the lymph-drains is $689\,\mathrm{cm}^3$. The clinical constraints are given in table 2.2. The most important organs at risk are the spinal cord and the parotids. As the spinal cord is a serial organ and dose hot spots have to be avoided, a safety margin of 5 mm around the spinal cord acts as an organ at risk in the optimization.

The large target volume requires also large irradiation fields. Thus, a relatively high number of particles can be expected in order to achieve a relative mean dose uncertainty of 0.8% as in the other treatment plans. Due to the air cavities in the head-and-neck site, deviations between pencil beam and Monte Carlo dose calculation can be expected. Therefore the optimization algorithm should require more iterations than e.g. in a prostate case.




(a) (b)

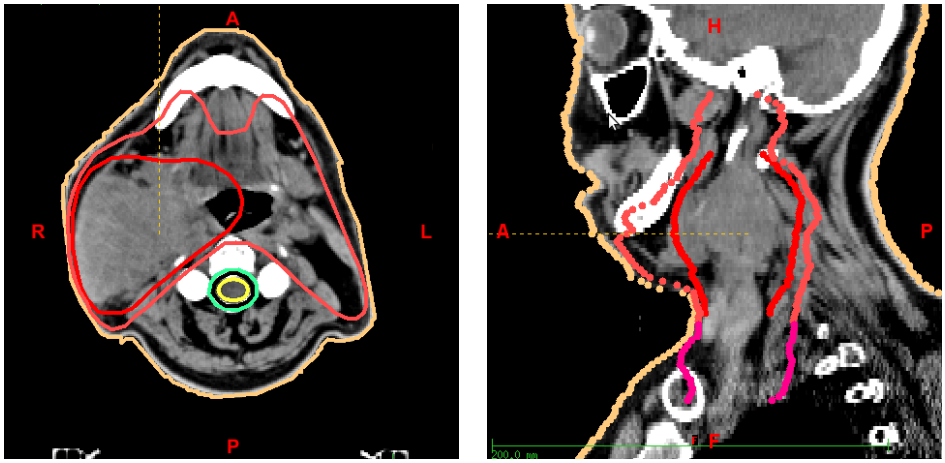
**Figure 2.12:** Transversal and sagittal isocentric slices of the nasopharynx case.

**Table 2.2:** Clinical goals for the nasopharynx treatment plan.

| Volume | Clinical Goal |
|---|---|
| Boost | $d_{50} = 66\,\mathrm{Gy}$ |
| Lymph drains | $d_{50} = 54\,\mathrm{Gy}$ |
| Spinal cord 5 mm | $d_{\max} < 40\,\mathrm{Gy}$ |
| Parotids | $\bar{d} < 26\,\mathrm{Gy}$ and $d_{50} < 30\,\mathrm{Gy}$ |

### 2.7.3 Larynx

This case is another coplanar 9-beam IMRT head-and-neck treatment plan. The tumor of the larynx is mainly located on the right side of the head. Again, an integrated boost concept is used in which a lower dose is applied to the lymph drains and a high dose to the boost. The target is comparably large and has a total volume of $1180\,\mathrm{cm}^3$. Critical organs are the spinal cord and the left parotis. The brain stem is located above the target volumes and therefore does not lie inside the beams. The shortest distance of the spinal cord to the target volume is about 13 mm. As in the nasopharynx case, a 5 mm safety margin is created around the spinal cord, which acts as an organ at risk for the plan optimization. The left parotis partially overlaps with the target volume. Because the priority of the treatment is a homogeneous dose coverage of the target, a helping volume is created that consists only of the non-overlapping part of the left parotis. Only this smaller volume is considered as an organ at risk during the optimization. The right parotis is affected by the tumor and is therefore ignored by the optimizer.




(a) (b)

**Figure 2.13:** Transversal and sagittal isocentric slices of the larynx case.

**Table 2.3:** Clinical goals for the larynx treatment plan.

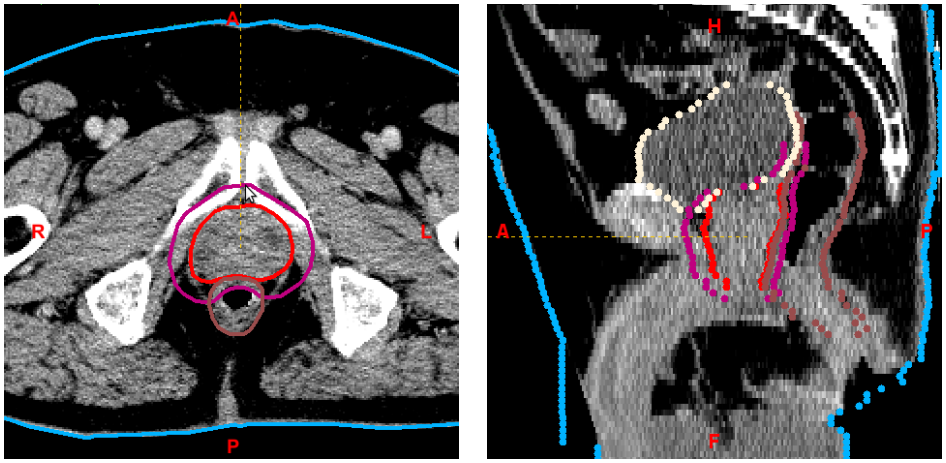
| Volume | Clinical Goal |
|---|---|
| Boost | $d_{50} = 70.6\,\mathrm{Gy}$ |
| Lymph drains | $d_{50} = 57.6\,\mathrm{Gy}$ |
| Lymph drains caud. | $d_{50} = 57.6\,\mathrm{Gy}$ |
| Spinal cord 5 mm | $d_{\max} < 40\,\mathrm{Gy}$ |
| Left parotis | $\bar{d} < 26\,\mathrm{Gy}$ and $d_{50} < 30\,\mathrm{Gy}$ |

### 2.7.4 Prostate

The prostate case is a 5-beam photon treatment plan. The directions of the incident beams are chosen in order to avoid the irradiation of the femoral heads. The CTV includes the prostate and the seminal vesicles and has a volume of $106\,\mathrm{cm}^3$. Adjacent organs at risk are the rectum and the bladder. Their clinical goals are given in table 2.4.

In addition to these clinical goals, a large maximum dose penalty was set for exceeding the dose in the normal tissue to avoid dose hot spots behind the entrance of the beams. To guide the optimizer, additional dose-volume constraints were set for the rectum: less than 52% should get a dose of more than 13 Gy, less than 33% a dose of more than 24 Gy and less than 15% of more than 30 Gy. These values should however not be understood as clinical constraints but more as a help for the optimization algorithm.

Fast convergence can be expected for the prostate case due to the high tissue homogeneity in this treatment site. The dose distributions of the pencil beam and MC algorithm should not differ much. Therefore, the hybrid optimization algorithm will likely require only a few iterations with the pencil beam dose model.




(a) (b)

**Figure 2.14:** Transversal and sagittal isocentric slices of the prostate case.

**Table 2.4:** Clinical goals for the prostate treatment plan.

| Volume | Clinical Goal |
|---|---|
| CTV | $d_{50} = 66\,\mathrm{Gy}$ |
| Rectum | $\bar{d} < 25\,\mathrm{Gy}$ |
| Bladder | $\bar{d} < 25\,\mathrm{Gy}$ |

## 2.8 Algorithmic performance, efficiency

In addition to the clinical plan quality evaluation, algorithmic properties such as runtimes, number of iterations and particle efficiencies are measured for each patient case. The particle efficiency values are evaluated for the hybrid optimization method with both dose models (section 2.5.1 and 2.6) and for the reference FMO algorithm (section 2.4) according to its definition (2.66).

For the reference FMO method we define two different efficiency values: the first can be understood as the number of particles required for the computation of the dose influence matrix in order to achieve a resulting dose distribution with the reference mean dose uncertainty $\bar{\sigma}_{\mathrm{fw}}$. With (2.66), the efficiency is calculated as

$$\varepsilon_{\mathrm{fmo}} = \left(N_{\mathrm{fw}}\bar{\sigma}_{\mathrm{fw}}^2\right)\left(N_{\mathrm{fmo}}\bar{\sigma}_{\mathrm{dc}}^2\right)^{-1}. \tag{2.86}$$

Here, $N_{\mathrm{fmo}}$ is the total number of particles used for the calculation of the dose influence matrix $J_d^{\mathrm{mc}}$ and $\bar{\sigma}_{\mathrm{dc}}$ is the resulting mean dose uncertainty when calculating the dose of the optimized fluence map $\vec{w}_{\mathrm{opt}}$ via $\vec{d} = J_d^{\mathrm{mc}}\vec{w}_{\mathrm{opt}}$. Accordingly, $N_{\mathrm{fw}}$ is the number of particles used for the forward MC dose calculation and $\bar{\sigma}_{\mathrm{fw}}$ is its mean dose uncertainty.

Practically this efficiency cannot be achieved because the average uncertainty of the resulting dose distribution depends not only on the uncertainty of the dose matrix but also on the final fluence map vector. However, this is not known in advance. Therefore, we calculate the second efficiency $\varepsilon^*_{\mathrm{fmo}}$ that can be understood as an effective efficiency and is defined by the number of particles required to achieve the desired mean dose uncertainty in each beamlet of the dose influence matrix:

$$\varepsilon^*_{\mathrm{fmo}} = \frac{N_{\mathrm{fw}}\,\bar{\sigma}^2_{\mathrm{fw}}}{N_{\mathrm{fmo}}\,\bar{\sigma}^2_{\mathrm{bix}}} \tag{2.87}$$

In this equation, $\bar{\sigma}_{\mathrm{bix}}$ is the average dose uncertainty per beamlet of the dose influence matrix as defined by (2.61). This more realistic efficiency value is typically smaller than $\varepsilon_{\mathrm{fmo}}$ because the average beamlet dose uncertainty $\bar{\sigma}_{\mathrm{bix}}$ is larger than the mean dose uncertainty of the final dose distribution $\bar{\sigma}_{\mathrm{dc}}$ (Siebers 2008).

## 2.9 Evaluation of the macroscopic pencil beam dose model

The hybrid sequential MC optimization algorithm requires an alternative dose calculation engine in order to calculate search directions in the optimization space. The performance of this optimization algorithm depends on the accuracy of the alternative algorithm. One alternative dose calculation algorithm was chosen to be the macroscopic pencil beam. The better the agreement between pencil beam and MC is, the fewer iterations during optimization have to be performed and the "more optimal" will be the final solution. In order to quantify the accuracy of the macroscopic pencil beam method, we will compare the dose differences between the macroscopic pencil beam algorithm and the MC simulation. These calculations include water and slab phantom simulations and the dose calculations of the presented patient cases. An analysis of the geometric kernel approximation in terms of dose calculation accuracy will not be given in this work since the limits of this approach are obvious.

**Water phantom** We simulated narrow square beams of 5 mm and 10 mm field side-length, impinging on a water-phantom with a source-surface-distance (SSD) of 88 cm. The voxel size for the calculation was chosen to be 1 mm$^3$. The dose distributions of the MC simulation and the pencil beam calculation are compared by inspecting depth dose curves and lateral dose profiles.

**Slab phantom** To increase the difficulty for both dose calculation algorithms, a bone and an air slab is inserted into the water phantom. Both inserts have the same size, a height of 3 cm and a length and width of 2 cm × 15 cm. Both slabs are centered on the beam axis, the bone slab first at a depth from 3 cm to 6 cm and the air slab at a depth from 9 cm to 12 cm. This setup is illustrated in figure 2.15. We calculated the

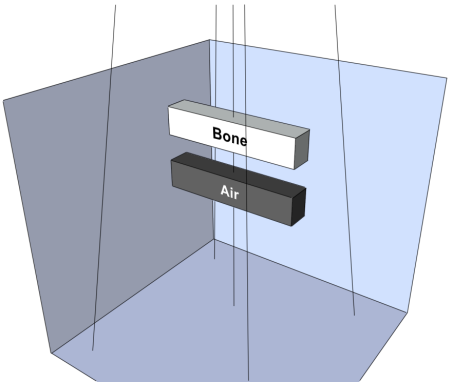



**Figure 2.15:** Sketch of the slab phantom.

dose distributions of a narrow $1\times1\,\mathrm{cm}^2$ square field and a broad $7\times7\,\mathrm{cm}^2$ beam with a SSD of 90 cm. The dose-to-medium technique was used for dose scoring during the MC simulation. The dose differences between both algorithm are quantified by comparing again lateral dose profiles and depth dose curves.

**Realistic patient cases** In addition to the single-beam water phantom calculations, we compare the pencil beam dose calculation with the MC simulation on realistic patient cases. These cases, which we have already been presented in section 2.7, were first optimized with the standard IMRT optimization method (see section 2.2.3), based on pencil beam dose distributions. Then, the dose distributions from the resulting fluence maps were recalculated by MC simulation at a mean dose uncertainty of 0.8 %. For these calculations we chose a voxel size of $(2.6\,\mathrm{mm})^3$ and use the dose-to-medium technique for MC dose scoring. The differences in dose are quantified by comparing dose-volume-histograms and by calculating the average dose differences of voxels with $d > 0.5 \cdot d_{\max}$.

# 3 Results

The hybrid sequential algorithm for inverse treatment planning was implemented into a small program written in C++. This program features the basic functionality of an inverse treatment planning system. In addition to an interface to the VMC$^{++}$ framework and the implementation of the hybrid optimization algorithm it includes the conventional dose influence matrix based optimization and dose calculation. It also allows the inspection of previously calculated dose distributions, calculates and displays important indicators of dose distributions and calculates the associated dose volume histograms (DVH). A small graphical user interface (GUI) was written, offering the typical tools for inverse planning as e.g. displays for DVHs, CT data, dose data, isodose lines and also controls for adjusting penalties, dose and DVH constraints. This program can be optionally started without any GUI for the use in scripts for example by setting an environment variable. A screenshot of the program running on Linux is shown in figure 3.1. Most of the following results and data in this chapter are generated by this software. Only the calculation of the beamlet positions and the generation of the pencil beam dose matrix is carried out with KonRad.


**Figure 3.1:** Screenshot of the graphical user interface for the hybrid MC treatment plan optimizer.

## 3.1 Macroscopic pencil beam vs. Monte Carlo

In order to quantify the accuracy of the pencil beam dose model, we compared pencil beam dose calculations against their MC simulation counterpart. The conceptional weaknesses of pencil beam algorithms are commonly known and similar results were published before (Scholz et al. 2003, Krieger & Sauer 2005). Still, knowing the differences between dose distributions from MC simulations and pencil beam calculation are crucial for understanding the functioning of the hybrid optimization algorithm.

### 3.1.1 Water phantom

First, we compared the resulting dose distribution from the precalculated macroscopic pencil beam (section 3.1.2) algorithm with the results of the Monte Carlo dose calculation on a water phantom. The results of the calculation are illustrated in figure 3.2. The figure shows the depth dose curves and the lateral dose profiles of both cases (5 mm and 10 mm field side-length), which are taken at the depths of 15 mm, 50 mm, 100 mm and 200 mm.

The dose calculations of both algorithms of the 10 mm beam agree very well and the relative dose differences of the two algorithms are less than one percent. In the case of the 5 mm square beam however, the differences between both algorithms become visible. In the entrance region of the first 5–10 cm, the pencil beam algorithm overestimates the dose up to 4 %. The differences become smaller with increasing depth as the pencil beam also overestimates the attenuation of the incoming photons.

The reason for these differences is not completely understood as the macroscopic pencil kernel is derived from MC simulations. Even on homogeneous water phantoms, the macroscopic pencil beam uses approximations. In theory, a separate pencil beam has to be calculated for each SSD to take account for the changing field sizes of the beam, measured at the entry of the beam into the patient. These different sizes reflect in slightly different scattering characteristics that can be expressed with phantom scatter ratios. This fact is however ignored by the macroscopic pencil beam method in the KonRad inverse planning system. Another source of error is the voxel size dependence of the Monte Carlo method since the dose depositions in a voxel are averaged over the volume of the voxel. On small fields, this averaging causes artifacts especially on the steep dose gradients at the edges of the irradiation field (Chetty et al. 2007). In contrast, the pencil beam algorithm evaluates the dose at a point – more specifically at the center of each voxel – instead of averaging the dose over the voxel's volume.

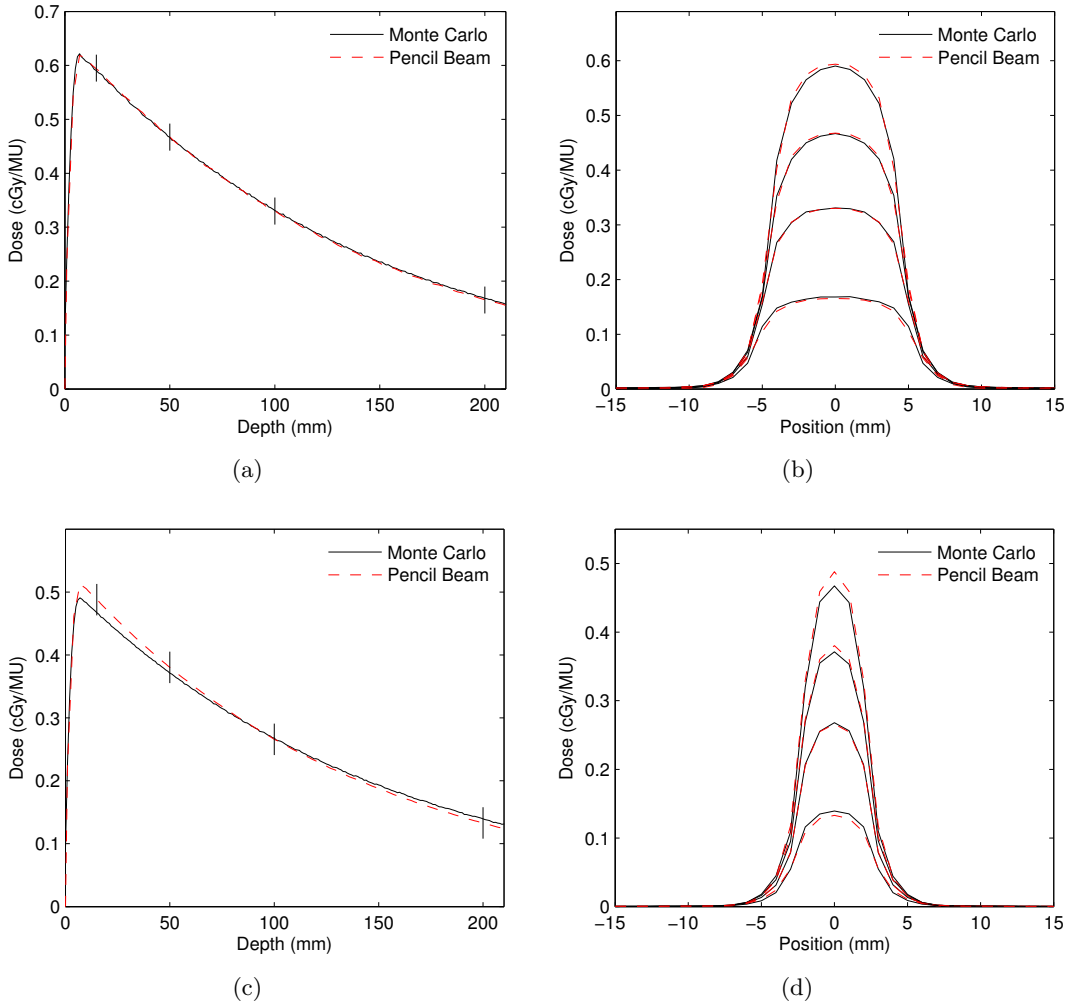



**Figure 3.2:** Dose distributions in the water phantom at a SSD of 88 cm. (a) and (b) show depth doses and dose profiles from a $1 \times 1\,\mathrm{cm}^2$ beam, (c) and (d) from a $0.5 \times 0.5\,\mathrm{cm}^2$ beam. The dose profiles are taken at the depths 15 mm, 50 mm, 100 mm and 200 mm.

### 3.1.2 Slab phantom

The results of the dose calculations of the bone-air phantom (depicted in figure 3.3) demonstrate the limitations of the pencil beam algorithm. Especially the dose to air is dramatically overestimated by this algorithm. More subtle, the different scattering processes inside the bone insert compared to water lead to lateral dose discrepancies, which can be seen in the plot of the lateral dose profiles. This issue is less pronounced for the large beam, where the width of the slab insert is smaller than the side length of the beam. In this case, also laterally scattered high energy particles from the surrounding water contribute dose to the bone slab and the differences between pencil beam algorithm and MC simulation become smaller. The same argumentation holds for the air insert. From a clinical perspective, air doses are uninteresting because they do not affect the patient. On the other hand, lung tissue is very similar to air and the dose to the lung has to be considered during the optimization. However, behind an air cavity, a new dose build-up takes place due to the range of the secondary electrons. This can be clearly seen in the depth dose curve of the MC dose calculation (figure 3.3(a), black line). Depending on the photon energy, the dose needs about 1–2 cm depth to reach its maximum.

Therefore, small tumors inside a lung lobe are effected by the build-up effect, which is completely neglected by the pencil beam algorithm. As a consequence, the dose to the tumor can be substantially overestimated by a pencil beam algorithm. These findings are not new and were published before by e.g. Krieger & Sauer (2005) or Scholz et al. (2003). Aside from the differences in the air and the bone slabs, the pencil beam generated dose distributions can be seen as a good approximation of the true dose distributions. The macroscopic pencil beam is therefore a good candidate for the dose model (2.67) in the hybrid optimization algorithm.

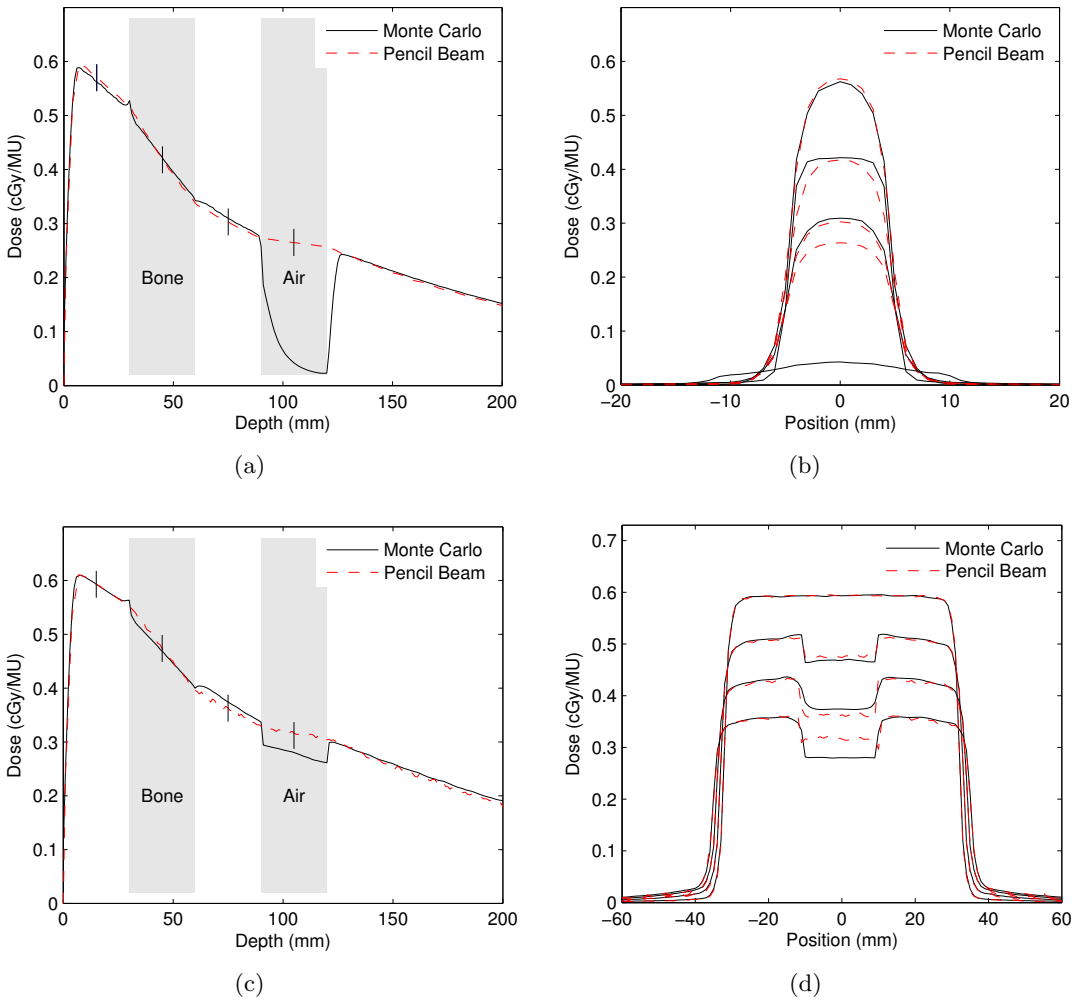


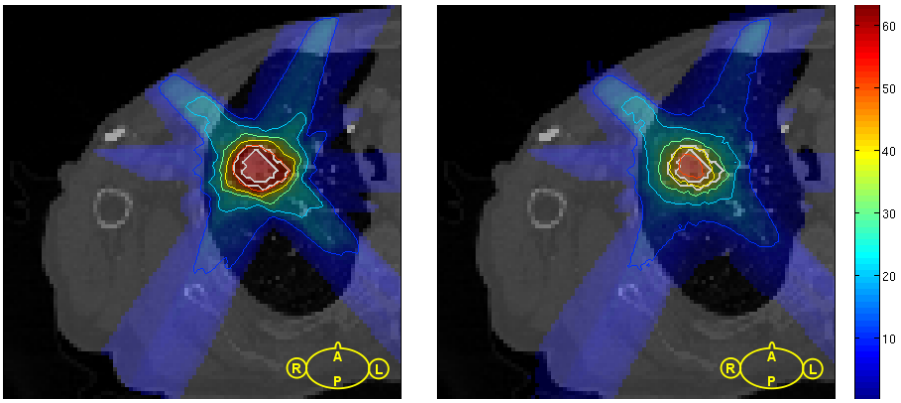

**Figure 3.3:** Dose distributions (dose-to-medium) in the slab phantom. (a)-(b) are depth doses and lateral dose profiles from a narrow $1 \times 1\,\mathrm{cm}^2$ beam; (c) and (d) from a broad $7 \times 7\,\mathrm{cm}^2$ beam. The dose profiles are taken at the depths 15 mm, 45 mm, 75 mm and 105 mm.

### 3.1.3 Patient cases

In principle, the Monte Carlo simulations of the optimized pencil beam treatment plans act as a starting point in the hybrid optimization algorithm with the pencil beam dose model. The magnitude of the dose differences between both dose calculation algorithms reflect the level of difficulty for the sequential hybrid optimization algorithm and thus its convergence rate. Also, it is a measure if a MC dose calculation is even required for the optimization. The results of the dose calculations of the patient plans are illustrated by the dose-volume histograms (DVH) in figure 3.5.

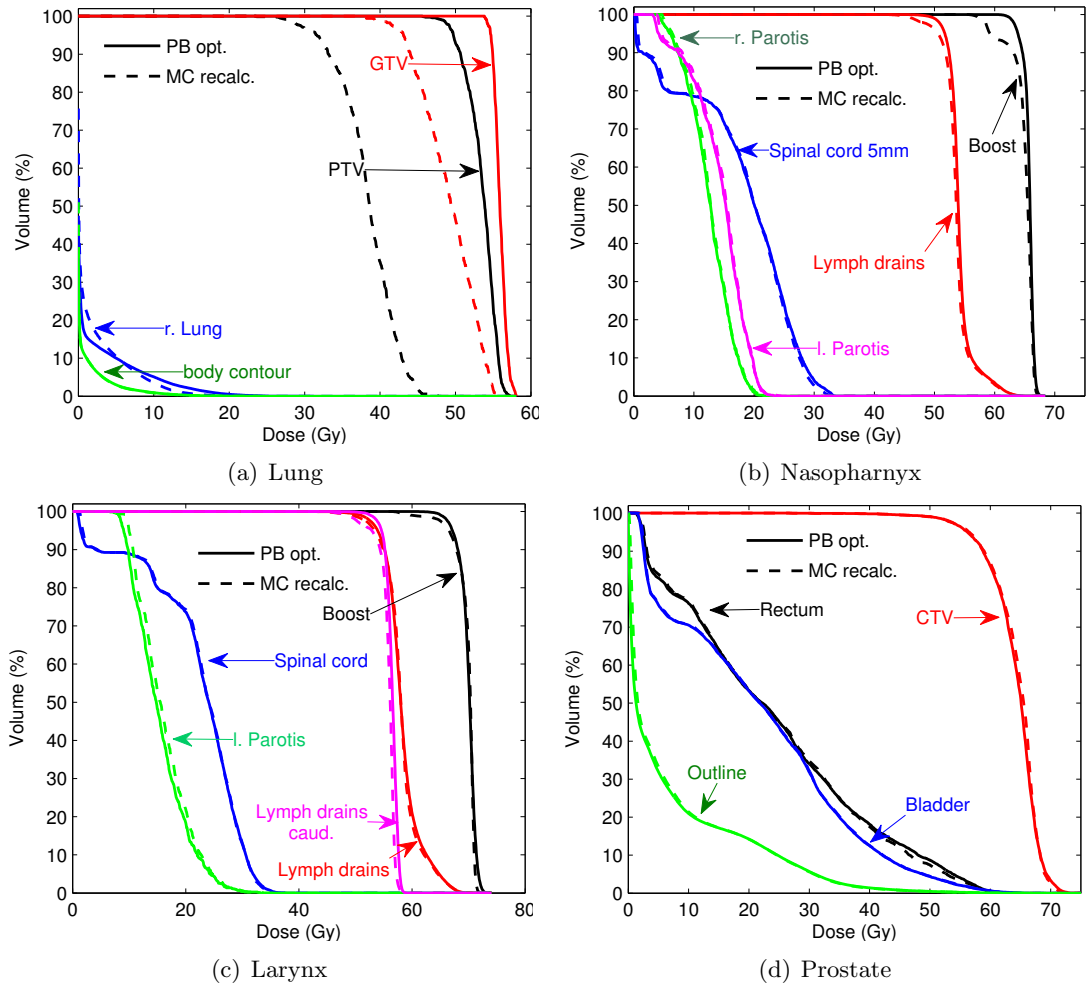
The lung case shows the largest discrepancies between both dose calculation algorithms. These differences are practically limited to voxels inside the right lung. In addition to the DVH, both dose distributions are illustrated as transversal slices in figure 3.4, which points out these dose differences. The MC recalculation reveals a significant underdosage of the GTV and the PTV and a worse dose homogeneity in the target. This underdosage cannot be compensated by a simple upscaling of the fluence weights. The average relative dose difference between voxels with a dose larger than half the maximum dose is 25.9 %. This example demonstrates without any doubt, why pencil beam algorithms are insufficient for a treatment planning of small lung tumors and why accurate dose calculation algorithms as e.g. MC simulations have to be incorporated into the optimization.

The deviations in the other cases are considerably smaller. From the DVH of the nasopharynx treatment plan (figure 3.5(b)), an underdosage of parts of the boost and the lymph drains can be identified. It should be noted that the DVHs include voxels of air




(a) PB optimization (b) MC recalculation

**Figure 3.4:** Transversal slices through the dose distributions of the lung treatment plan. The left image shows the result of a plan optimization based on a pencil beam dose calculation algorithm. The right image shows the dose distribution of a plan recalculation with a MC algorithm. The dose values in the legend are given in Gy.

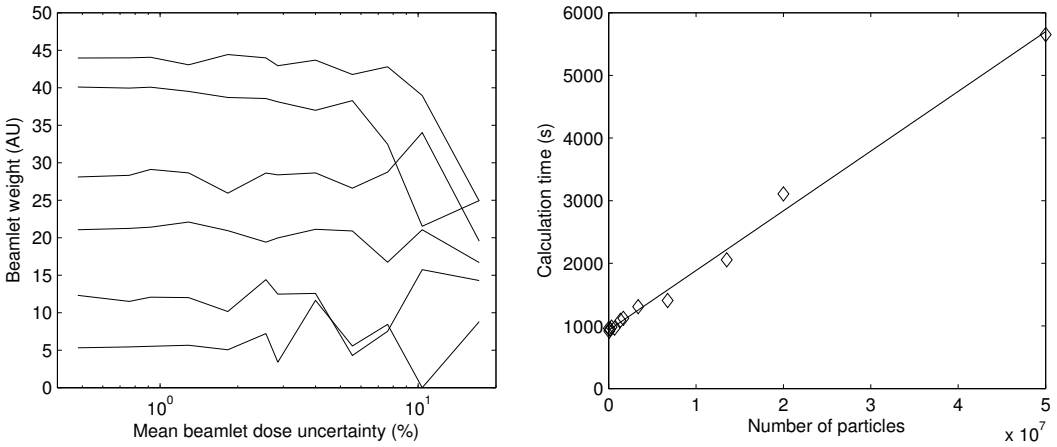



(a) Lung (b) Nasopharnyx

(c) Larynx (d) Prostate

**Figure 3.5:** Dose-volume histograms of four optimized treatment plans using pencil beams compared to dose distributions of the MC dose recalculation.

and that the underdosed voxels lie inside or behind air cavities like the throat. The calculated average relative dose differences of voxels with $d > 0.5\,d_{\max}$ is 2.24 %. The differences are even smaller in the larynx case. Here a mean relative dose difference of only 1.70,% of the voxels with doses larger than half of the maximum dose was calculated. The DVHs of the pencil beam optimization and the MC recalculation are almost identical. In principle, a treatment plan optimization with MC dose calculation algorithms is not necessary in this case. The prostate case shows the smallest deviation between planned and recalculated dose distribution of 1.61 % in average. It should be noted that the noise of 0.8 % with respect to the maximum dose has a main share on the differences to the pencil beam doses in the larynx and the prostate case.




(a) Fluence map development (b) Runtimes for the dose matrix computation

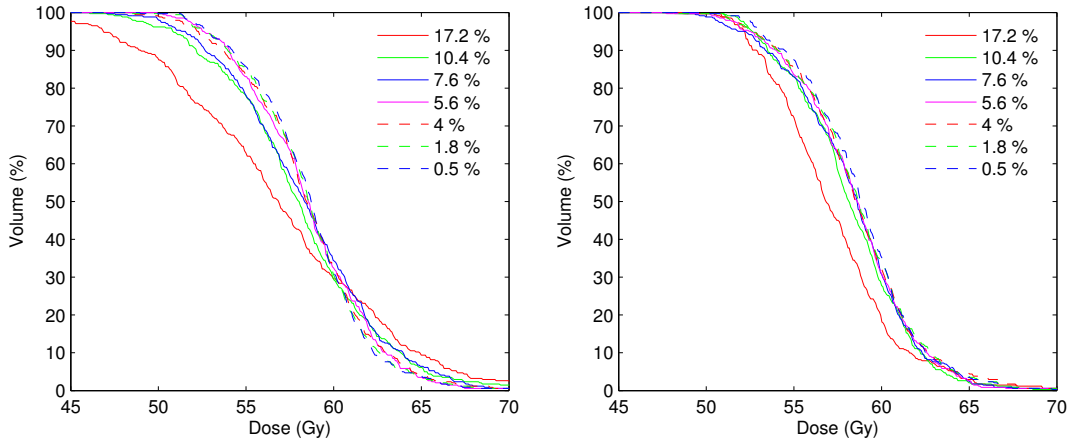
**Figure 3.6:** (a) - Values of optimized beamlet weights depending on the uncertainty of the dose influence matrix. The presented six beamlet weights are arbitrarily chosen. In this particular example, the optimization stabilizes after the mean beamlet dose uncertainty falls below 2 %. (b) - Runtimes for the calculation of the dose influence matrix at different uncertainty levels of this treatment plan with 169 beamlets. The runtime follows a linear trend but shows a significant offset of about 930 seconds.

## 3.2 Stability of the reference FMO algorithm

Apart from its poor efficiency, one large issue of the reference FMO algorithm is that the result of the optimization depends on the statistical accuracy of the dose influence matrix (dose uncertainty). In order to get stable optimization results and therefore limit the convergence error of the optimization, a certain minimum accuracy is necessary, which is however different for each treatment plan.

To demonstrate this behavior, we took the lung case as an example (see section 2.7.1) and calculated its dose influence matrix at different uncertainty levels – varying from 0.5 % – 17 % mean beamlet dose uncertainty. The created dose influence matrices then were used for the FMO optimization with the limited-memory BFGS algorithm. The optimization was stopped if the iteration limit of 100 was exceeded. To analyze the impact of the different uncertainties on the treatment plan quality, the dose distribution of each optimized fluence map was recalculated at a mean dose uncertainty of 0.8 %.

Figure 3.6(a) illustrates how some of the optimized fluence weights change with the uncertainty of the dose influence matrix. Obviously, the optimization in this example becomes unstable if the mean dose uncertainty exceeds 2 %. From 5 % uncertainty the beamlet weights undergo dramatic changes. For example the weight of one particular beamlet (not shown in the plot) reduces from 24.3 (at 17 % uncertainty) down to 3.3 (at 0.5 %). That is, the importance of this beamlet is completely overrated by the low




(a) Optimized at varying dose uncertainty.

(b) Recalculated at 0.8% uncertainty.

**Figure 3.7:** Impact of varying uncertainty of the dose influence matrix on the DVH. The figure shows the DVH of a lung GTV. (a) shows the result of the optimization depending on the dose uncertainty. (b) shows the results of the high precision dose recalculation from the optimized fluence maps.

uncertainty optimization. The optimization stabilizes only if the mean beamlet dose uncertainty is reduced to 1–2 %. The effect of the different dose uncertainty levels on the optimized dose distributions and thus on the plan quality is presented by the dose volume histograms (DVH) in figure 3.7. The left plot (figure 3.7(a) shows the DVH of the GTV as a direct result of the optimization. It can be clearly seen, that the increase in noise results in a degradation of target dose homogeneity. This effect is however only an artifact of the noisy dose distribution. The true DVHs are shown in the figure on the right, which are the result of the high precision recalculation at 0.8 % mean dose uncertainty. According to these DVHs of the GTV, the optimization seems to stabilize at 4 % mean beamlet dose uncertainty and lower. These results are however heavily case specific and are demonstrated only to emphasize the problem of the right choice of the uncertainty.

Finally, one important issue of the FMO algorithm in connection with the VMC$^{++}$ Monte Carlo package will be sketched out here. Naturally, the time for the dose influence matrix calculation depends linearly on the number of particles: more particles result in a lower dose uncertainty but require more calculation time. Figure 3.6(b) acknowledges this linear dependence for the example case. Remarkably, the simulation of a dose influence matrix with only a few particles ($\sim$ 200000) takes more than 15 minutes (y-axis offset of 930 s in figure 3.6(b)). Given that the treatment plan consists of 169 beamlets, this results in a dose calculation delay of 5.5 seconds per beamlet. The program for this calculation was explicitly written in order to avoid a reinitialization of the VMC interface (setup of patient specific data and particle source) prior to the dose calculation of each beamlet. A further investigation into the VMC$^{++}$ code revealed that the batch statistics

implementation in VMC$^{++}$ is responsible for this delay, as it has to iterate 50 times (the number of batches in VMC$^{++}$) over the dose cube of each beamlet according to (2.41). This calculation is independent of the number of simulated particles and has to be even done if only a few particles are simulated.

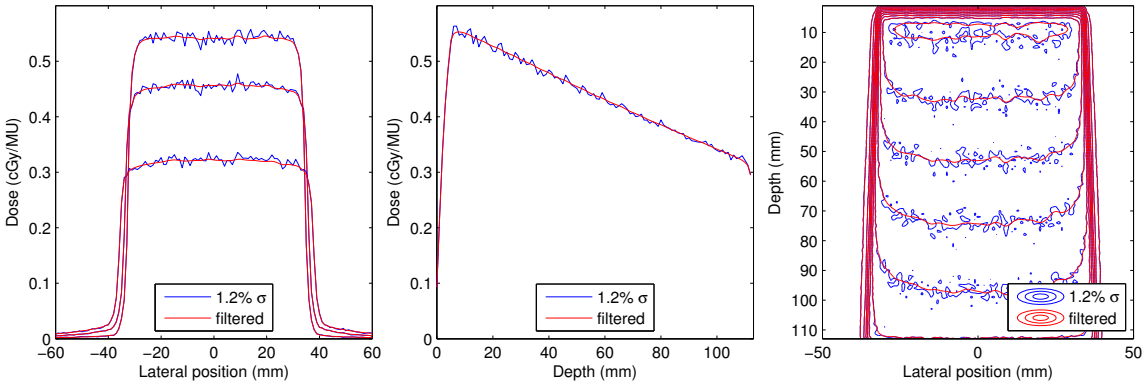
## 3.3 Anisotropic filtering

To demonstrate the effect of dose smoothing, we applied the anisotropic filtering algorithm (section 2.3.2) to dose distributions of a water-phantom treatment plan and a realistic prostate treatment plan. In the case of the water phantom, the size of the irradiation beam is $7\times7\,\mathrm{cm}^2$ (in the beam's isocenter plane). The MC dose calculation was carried out at a relative mean dose uncertainty of 1.2 %. The prostate case is a pre-optimized standard 5-beam IMRT treatment plan. Its dose distribution was calculated at a mean relative dose uncertainty of 1.8 %. For a "gold standard" comparison, the same treatment plan was additionally calculated at an uncertainty of 0.5 %. Anisotropic diffusion for dose smoothing was finally applied to the noisy dose distributions with their corresponding uncertainty distributions. The standard parameters of $k = 1.75$, $N_{iter} = 4$ and $\Delta t = 3/44$ were used for the diffusion algorithm.

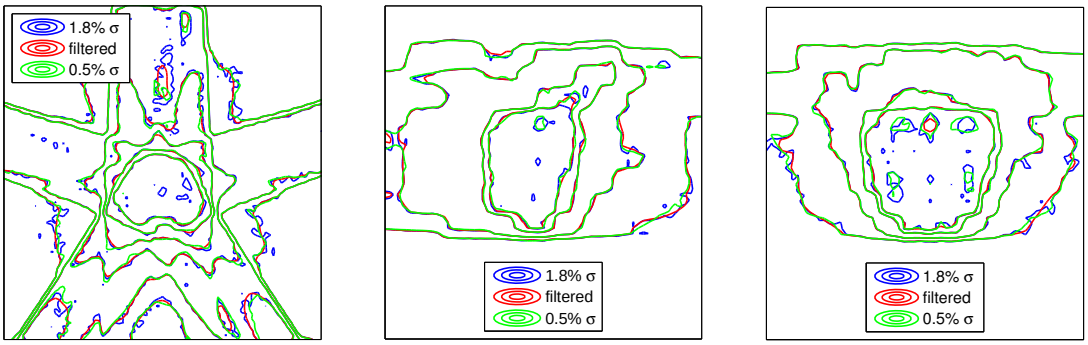
The outcome of the dose smoothing is shown in figure 3.8. The isodose lines in the original dose distribution of the single-beam water phantom case are noisy and distorted (figure 3.8(c), blue line). The isodose lines of its filtered dose distribution (red lines) are much smoother and resemble more a realistic dose distribution. Due to the edge-preserving character of this algorithm, the gradients at the field edges are not effected by a possible washout. This is backed also by figure 3.8(a) with the lateral dose profiles, in which doses at the field edges of the original and the denoised dose distribution match very well. In the region of constant dose however, diffusion increases so that the spikes caused by the noise are almost completely flattened.

The original dose distribution of the prostate treatment plan seems less noisy compared to the single-beam scenario, despite the increased uncertainty level. Nevertheless, the noise has a significant impact on the isodose lines. It creates the impression that the treatment plan with the noisy dose distribution includes a lot of small hot spot islands (blue isodose lines). The recalculation at high precision (green lines) reveals however, that these islands are mostly created by the noise and are not a feature of the treatment plan. Denoising with the anisotropic filtering algorithm removes these "hot spots" and also smooths all other isodose lines (red). Its result agrees very well with the high precision calculation. Still, this denoising technique – and many other noise filters – decrease maximum doses and spikes, even if these are not caused by noise. Therefore, also some of the true spots of increased dose that also appear in the accurate calculation are removed by the filter. This can be particularly recognized in the frontal slice through the dose distribution.

Due to these reasons, dose filtering is a great tool for post-processing dose distributions for e.g. the presentation in the treatment planning software. In a forward dose calculation, this denoising allows a significant reduction of particle histories (Miao et al. 2003). During the optimization process however, intermediate dose smoothing removes small local features as hot and cold spots from a treatment plan. If this technique was applied after each iteration of the hybrid optimization algorithm with an incremental dose update, these small effects would accumulate over time and the optimization algorithm would have no opportunity to correct for hot or cold spots. Therefore, denoising during the optimization should be avoided.




(a) Lateral dose profiles (b) Depth dose curve (c) Isodoses




(d) Transversal slice (e) Sagittal slice (f) Frontal slice

**Figure 3.8:** Results of the anisotropic filtering: (a)-(c) Dose distribution of a single beam in water, noisy (1.2 % mean dose uncertainty, blue line) and smoothed dose (red line). (d)-(f) Dose calculation of a prostate treatment plan, noisy (blue line, 1.8 % mean dose uncertainty), filtered (red line) and high precision calculation (0.5 % mean dose uncertainty, green line).

## 3.4 Uncertainty estimation of the objective function

The value of the objective function depends to a great extend on the accuracy/uncertainty of the dose distribution. As a rule of thumb, the larger the dose uncertainty is, the larger will be the value of a quadratic objective function. We evaluated the three strategies for the estimation of the objective function value of the "underlying noise-free" dose distribution that were already discussed in the methods in section 2.3.2. For four different patient cases (2 prostate, 1 head and neck, 1 lung) we calculated the objective function for an arbitrary beamlet configuration at different mean dose uncertainties. For each dose distribution, the estimation of the true objective function $f(\vec{d}^t)$ was calculated with all three methods.

Figure 3.9 illustrates the measured and estimated objective function values. The dependency of the objective function from the dose uncertainty is clearly visible (red solid line). As expected, the objective function value decreases with the dose uncertainty. The relative decrease in the objective function depends however on the patient case, the simulated fluence map and the settings of dose constraints and penalties. Particularly in case (a), the decrease is enormous and $f$ changes about 40% when reducing the mean dose uncertainty from 2% to 0.4%.

The objective function estimation with the error propagation method is poor (dashed blue graph). The first order error propagation significantly underestimates the error in the objective function and the approximation of $f(\vec{d}^t)$ is much to high.

Although smoothing with the anisotropic diffusion filter creates "visually appealing" dose distributions, it reduces maximum dose values and small local dose features. Thus, the resulting dose distributions are often "too good to be true" and their associated objective function value is too small. The estimation of $f(\vec{d}^t)$ using smoothing is depicted in figure 3.9 by the magenta dash-dotted line. Because the strength of smoothing decreases with reduced noise, this method still converges to the objective function values of the noise-free dose distribution when reducing the dose uncertainty.

The best prediction of the objective function of the noise-free dose distribution provides the method of noise simulation. Although the assumptions in this model are very crude, the predicted values are practically constant and in good agreement with the extrapolated guess to zero uncertainty. This method is represented with the green dash-dotted line in the figure.

Due to these findings, the last method of simulated noise is the best candidate if an estimation of the error of the objective function or the objective function value of the noise-free dose distribution is required. First, it gives consistent results over a large range of dose uncertainties. Therefore it can be used to determine the reason of an objective function decrease (better treatment plan or reduced dose uncertainty). Second, the computational overhead for this calculation is small so that the computation of the actual objective function value and its error estimation can be carried out simultaneously.

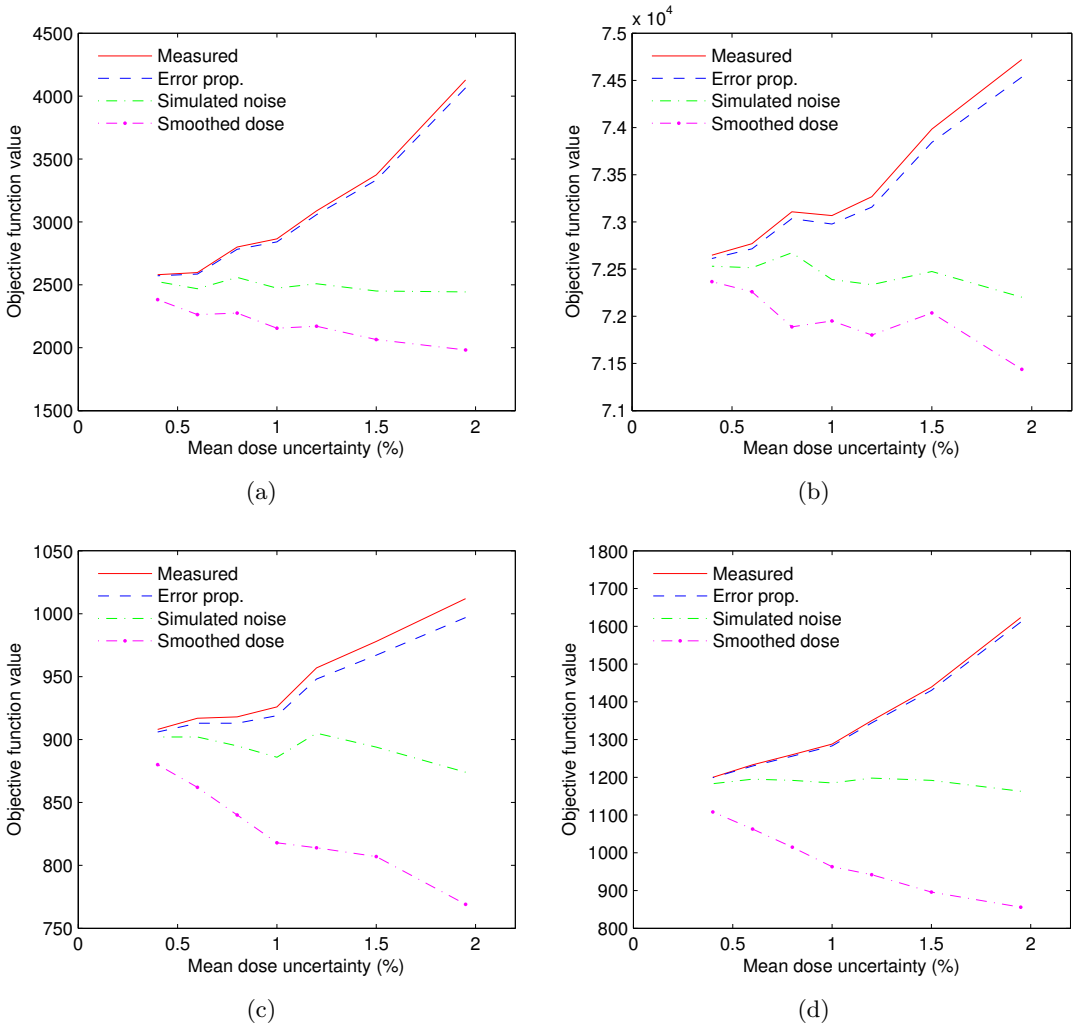
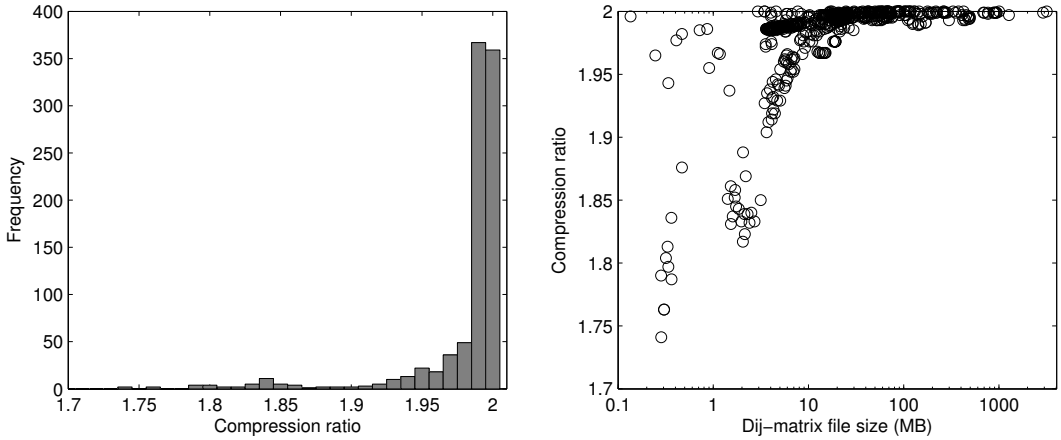



**Figure 3.9:** Objective function change with increasing dose uncertainty for 4 different patient cases (a)–(d): the solid red line represents the objective function of the MC calculated noisy dose distribution at different mean dose uncertainties. The objective function of the underlying noise-free "true dose distribution" $f(\vec{d}^t)$ can be estimated by extrapolating this line to zero uncertainty. The dashed blue line is the estimation of $f(\vec{d}^t)$ with the gaussian error propagation method. This method underestimates the error in $f$ by far. The green dash-dotted line is the estimation of $f(\vec{d}^t)$ with the simulated noise method. It shows a good agreement with the extrapolated value. The dash-dotted magenta line shows the estimation of $f(\vec{d}^t)$ with the smoothing method. This method severely underestimates the objective function value.

## 3.5 Dose influence matrix compression

### 3.5.1 Compression ratios

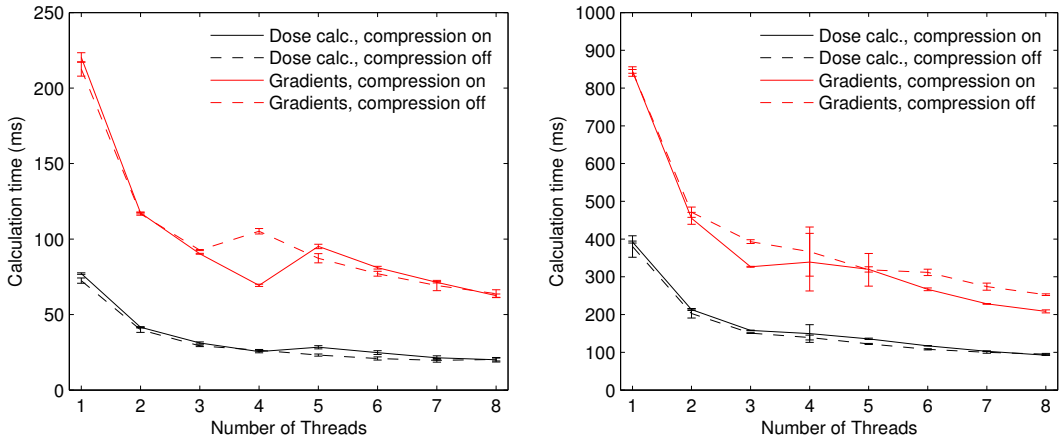
With a small script, each dose influence matrix file (*.dij) on our computer was analyzed and the possible compression ratios were calculated. These compression factors are given as the ratio of the file sizes prior compression to the file sizes after compression. In total 930 files were processed, with a file size from 50 KB to 3.2 GB. These files are stored in an uncompressed format on the hard disc drive. The resulting compression ratios are presented in figure 3.10. The histogram shows that the theoretical maximum compression ratio of 2 can be achieved in most cases. This is because the index differences between two consecutive voxels with a dose value larger than zero is less than 192 for the most part (as already shown in figure 2.5). These small index gaps are encoded with an one byte integer instead of using four bytes in the uncompressed case. Because the storage of the dose value of a voxel takes additional two bytes, only 3 bytes are required for the tuple (voxel-index, dose-value) with the compression method compared to the 6 bytes without compression. Interestingly, the probability of a high compression ratio increases with the file size according to figure 3.10(b). A compression ratio larger than 1.95 could be generally achieved on dij-files greater than 20 MB.




(a) Occurence of compression ratios

(b) File size vs. compression ratio

**Figure 3.10:** Results of the dose influence matrix compression from 930 different files.




(a) Small case: 150 MB

(b) Larger case: 656 MB

**Figure 3.11:** Runtimes of the dose calculation and the gradient computation with and without dose matrix compression for two different patient cases.

### 3.5.2 Impact on runtime performance

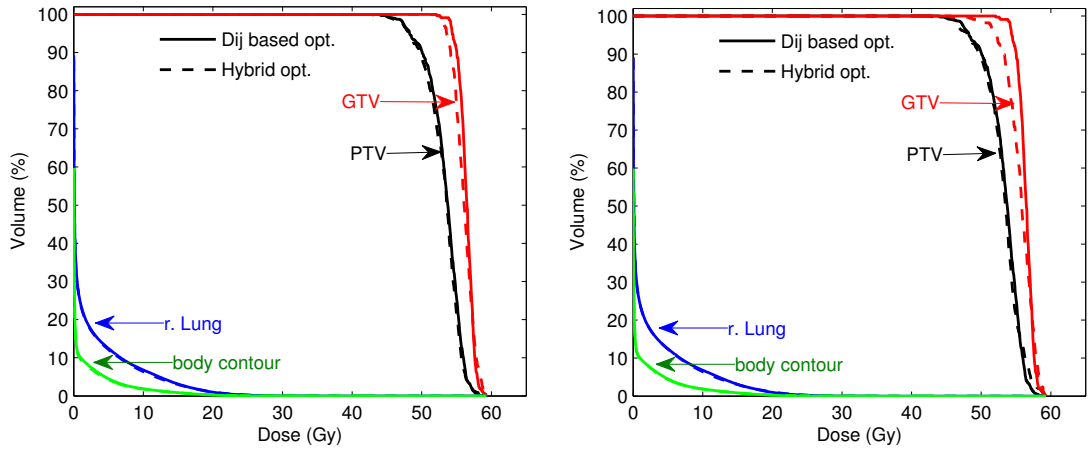
In addition to the pure compressibility analysis, the impact of the dose matrix compression on the calculation time was examined. Because the dose influence matrix $J_d$ is stored in its compressed form to reduce memory usage, the voxel indices have to be decoded on-the-fly for the dose calculation $\vec{d} = J_d\vec{w}$ and the calculation of the gradient of the objective function $\vec{\nabla}_w f(\vec{w}) = J_d^\top \vec{\nabla}_d f(\vec{d}(\vec{w}))$. Therefore, we measured the runtimes of the dose and the gradient calculation for two patient cases with and without compression. The first patient case consists of only $78 \times 55 \times 94$ voxels and 463 beamlets from 5 beams. The file size of the uncompressed dose matrices is 150 MB. The second case consists of $215 \times 149 \times 174$ voxels with dose contributions from 454 beamlets and 10 beams. Due to the increased number of voxels, the uncompressed dose matrices of the second case require 656 MB. Because the compression scheme is designed for parallel code, the runtimes were measured for a different number of calculation threads, varying from 1 to 8. Each measurement was repeated 5 times. These measurements were executed on an Intel®Core™i7-860 CPU (2.8 GHz, 4 calculations cores, hyper-threading) with a system memory of 4 GB. The results of this measurements are presented in figure 3.11. No specific runtime optimizations were used for the implementation of both calculations as e.g. compiler intrinsics (SSE, SSE2) or data partitioning schemes.

Aside from the gap at 5–6 threads in the first patient case, the runtime for the dose calculation is about 5 % slower with compression enabled. This performance regression is however negligible given that there is the huge saving of computer memory that allows much larger patient cases. A general trend for the calculation of the objective function's gradient cannot be observed. In the first patient case, the runtimes are quite on a par

with each other. Interestingly, there is a large difference between the 4 thread measurements with an advantage for the calculation with compressed data. This behavior could be reproduced on all subsequent measurements, that is this result is not an outlier. Apart from that, the calculation with the uncompressed data has a small advantage in runtime. The calculation of the gradients in the second patient case behaves differently. Here, the calculation with data compression decreases the runtime up to 20 %. Given that the decompression adds complexity and CPU instructions, the cause of this behavior has to be the reduced data size that probably results in fewer cache misses. Attention should be drawn to the large error bars at the 4-thread measurement. While using 4 threads, the runtimes were very unstable and varied from 260 ms to 400 ms. Other parallel code tests with the OpenMP directives (Dagum & Menon 1998) would reproduce an unstable behavior in some cases. We suspect the outdated operating system (Linux kernel 2.6.31, OpenSuse 11.2) and the OpenMP library as a source for this issue since it was one of the first Linux distributions that delivered OpenMP support. Still, the computation with data compression can be regarded on average as equivalent with respect to the runtime. Therefore, this presented compression method is an excellent choice when the memory requirements are critical.

## 3.6 Comparison of the optimization algorithms - patient cases

We optimized IMRT treatment plans at different body sites with the reference FMO algorithm and the hybrid optimization algorithm with the pencil beam dose model (Hybrid/PB) and the geometric kernel approximation (Hybrid/GK). The patient cases have been presented in section 2.7. In order to vary the complexity of the optimization, treatment plans with both 5 mm and 10 mm square beamlets were generated. KonRad was utilized to calculate spot positions and to generate the dose influence matrix for the pencil beam dose model. The dose distributions were calculated at a voxel size of $(2.62\,\mathrm{mm})^3$. The reference FMO algorithm was ultimately stopped after 100 iterations. For a fair comparison, all three optimization algorithms used the same dose prescriptions and penalties.




(a) Pencil beam dose model

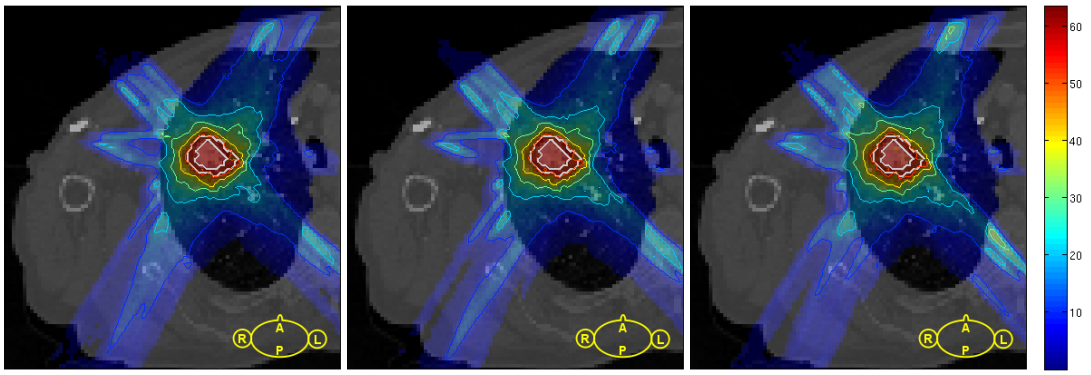
(b) Geometric kernel

**Figure 3.12:** Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm from the Lung case with 5 mm square beamlet size. The left image represents the hybrid algorithm with the pencil beam dose model, the right uses the geometric kernel approximation.

### 3.6.1 Lung – 5 mm × 5 mm square beamlets

For the hybrid optimization with the pencil beam dose model, this patient case is the most challenging one due to the large differences between pencil beam and Monte Carlo dose distributions. The plan consists of 454 beamlets in total. The hybrid optimization with the pencil beam dose model took 8 iterations to converge. Although the geometric kernel approximation provides a better estimation of the dose values in the lung tissue, the optimization still took 7 iterations. The approximate minimum of the objective function, calculated with the reference algorithm, is $f_{\min} = 782$. The Hybrid/PB algorithm stopped at an objective function value of $f = 924$. Using the Hybrid/GK algorithm, a value of $f = 1130$ could be reached. Thus, both hybrid algorithms cannot reach the mathematical minimum of the optimization problem, but the Hybrid/PB generates a better solution from mathematical point of view.

The clinical differences of the resulting dose distributions of the three algorithms are presented in the dose-volume histograms (DVHs, figure 3.12) and the transversal dose slices (figure 3.13). In addition, table 3.1 shows mean doses, median doses (50% of all voxels have a dose larger than this value) and maximum doses for the target volumes, the right lung and the body contour. The dose slices reveal that the hybrid algorithm tends to increased doses behind the beams' entrance into the patient. The increased dose comes mainly from a boosting of beamlets that aim at the PTV margin. The maximum doses inside the patient contour support this finding. The maximum dose of the hybrid algorithm is about 10 Gy higher than the maximum dose of the reference algorithm



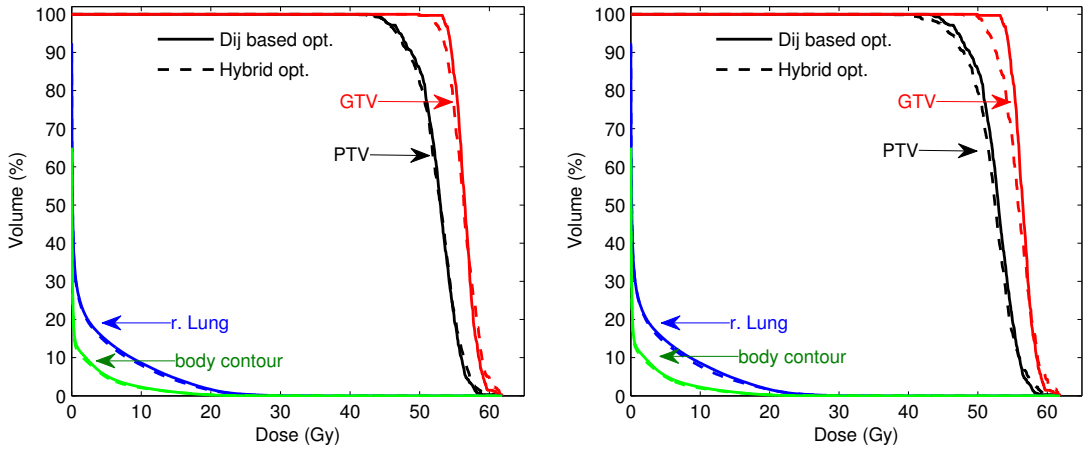

(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

**Figure 3.13:** Lung, 5 mm square beamlets: The figure compares doses in a transversal slice from the reference FMO algorithm (a) against doses from the hybrid algorithm with the pencil beam dose model (b) and the geometric kernel approximation (c). GTV and PTV are highlighted white. Isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy and 51.3 Gy (95 % of PTV dose prescription) are indicated. The dose values in the color legend are given in Gy.

**Table 3.1:** Median, mean and maximum dose values of the Lung case with 5 mm square beamlets. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| GTV | 56.5 | 56.2 | 55.9 | 56.4 | 56.2 | 55.8 | 58.7 | 61.5 | 61.5 |
| PTV | 53.7 | 53.6 | 53.5 | 53.3 | 53.1 | 53.3 | 59.3 | 59.3 | 65.0 |
| Right lung | 0.0 | 0.1 | 0.1 | 1.8 | 1.9 | 1.9 | 34.2 | 37.9 | 33.6 |
| Body contour | 0.0 | 0.0 | 0.0 | 0.7 | 0.7 | 0.7 | 33.6 | 44.5 | 43.7 |

(44 Gy vs. 34 Gy). The DVH of the Hybrid/PB algorithm does not differ significantly from the reference DVH. Only the dose homogeneity of the target is slightly worse. In the case of the Hybrid/GK algorithm, a further degradation of the target dose coverage can be observed.




(a) Pencil beam dose model

(b) Geometric kernel

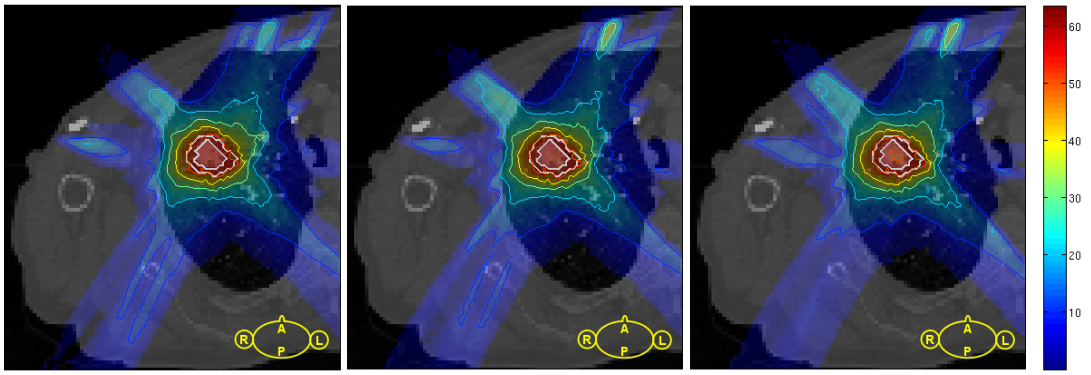
**Figure 3.14:** Lung, 10 mm square beamlets: Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm. The left image represents the hybrid algorithm with the pencil beam dose model, the right uses the geometric kernel approximation.

### 3.6.2 Lung – 10 mm × 10 mm square beamlets

The results of the lung case with 169 square beamlets with 10 mm side length are similar to the previous lung case. With the Hybrid/PB algorithm, the optimization took 6 iterations. The optimization stopped because the search direction $\Delta\vec{w}$ was no longer pointing downhill in the optimization space. The final objective function value was $f = 1340$. The Hybrid/GK stopped even after only 3 iterations at a higher objective function value of about $f = 1640$. The absolute minimum of this optimization problem of $f_{\min} = 1220$ was evaluated with the reference FMO method. Again, from a mathematically point of view, the Hybrid/PB algorithm outclasses the Hybrid/GK algorithm.

**Table 3.2:** Median, mean and maximum dose values of the Lung case with 10 mm square beamlets. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

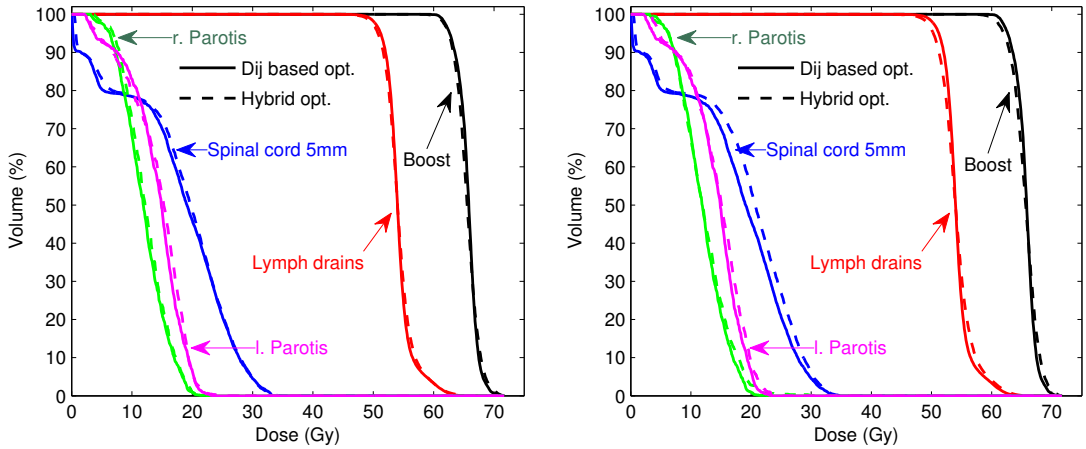
| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| GTV | 56.5 | 56.3 | 56.0 | 56.6 | 56.5 | 56.2 | 62.8 | 64.3 | 71.4 |
| PTV | 52.9 | 52.8 | 52.5 | 52.7 | 52.7 | 52.3 | 61.1 | 60.6 | 63.0 |
| Right lung | 0.0 | 0.1 | 0.1 | 2.2 | 2.2 | 2.2 | 34.5 | 35.7 | 39.3 |
| Body contour | 0.0 | 0.0 | 0.0 | 0.8 | 0.8 | 0.8 | 29.9 | 38.4 | 40.0 |




(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

**Figure 3.15:** Lung, 10 mm square beamlets: The figure compares doses in a transversal slice from the reference FMO algorithm (a) against doses from the hybrid algorithm with the pencil beam dose model (b) and the geometric kernel approximation (c). GTV and PTV are highlighted white. Indicated are the isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy and 51.3 Gy (95 % of PTV dose prescription). The dose values in the color legend are given in Gy.

Clinically, both hybrid optimization algorithm show the same characteristics as in the previous lung case. Figure 3.15 reveals, that the dose behind the beams' entrance into the patient is elevated in the hybrid cases, specifically in voxels inside the beam coming from the 30° gantry position. The 95 % isodose line wraps optimally around the PTV in all three cases. The Hybrid/GK features however a cold spot inside the GTV. Mean and median doses are similar in the dose distributions of the three optimization methods. The hybrid algorithm however increases maximum doses. Inside the body contour, the maximum dose is increased about 10.1 Gy by the Hybird/GK algorithm and about 8.5 Gy by the Hybrid/PB method. The DVHs in figure 3.14, which compare the results of both hybrid algorithm against the outcome of the reference algorithm, support these data. In the case of the Hybrid/GK algorithm, the dose homogeneity inside the GTV is clearly worse, the $D_{95}$ dose is reduced by 2.6 Gy. Using the pencil beam dose model, the target dose coverage decreases only slightly against the reference algorithm such that the $D_{95}$ dose is reduced only by 0.9 Gy.




(a) Pencil beam dose model

(b) Geometric kernel

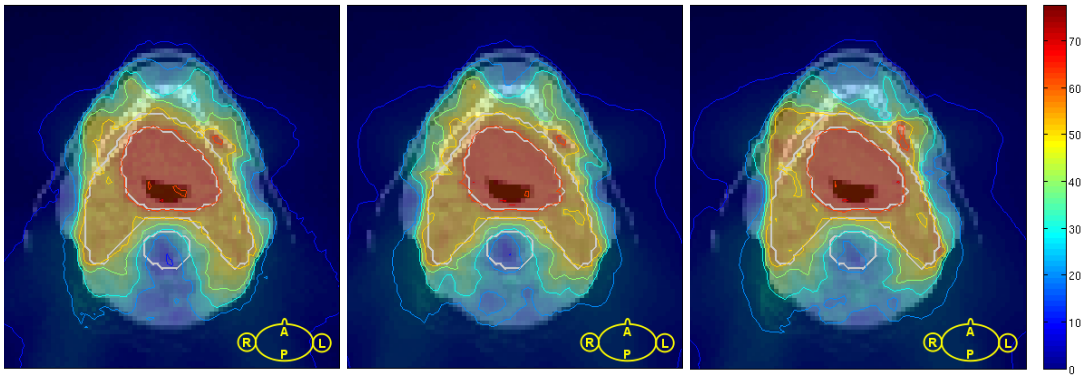
**Figure 3.16:** Nasopharynx, 5 mm square beamlets: Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm. The left image represents the hybrid algorithm with the pencil beam dose model, the right uses the geometric kernel approximation.

### 3.6.3 Nasopharynx – 5 mm × 5 mm square beamlets

The similarities in the dose distributions of the pencil beam optimized treatment plan and its Monte Carlo dose recalculation suggest that this case should be much simpler to optimize than the Lung case. On the other hand, with 5614 beamlets in total the dimensionality of this optimization problem is huge, which makes the optimization more difficult.

The reference algorithm with the precalculated MC dose influence matrix converged at an objective function value of $f_{\min} = 590$. This is the benchmark for the hybrid algorithms. Using the Hybrid/PB method, a final objective function of $f = 703$ was achieved after 8 iterations. The main improvement over the MC recalculation after the initial pencil beam optimization was obtained after the first iteration, which reduced the objective function from 1122 to 780. The Hybrid/GK algorithm stopped after 6 iterations at an objective function value of $f = 910$. Thus, from a mathematical point of view, the Hybrid/PB optimization performs better than the Hybrid/GK algorithm.

Looking at the dose data (table 3.3, figure 3.17 and figure 3.16), no significant differences between all three algorithms can be found. Still, the Hybrid/GK algorithm tends to produce higher maximum doses, which are specifically higher in the right parotis. Also, the doses to the spinal cord are increased when optimized with the hybrid algorithms. Particularly with the Hybrid/GK algorithm, the spinal cord doses increase, as can be seen in the DVH (figure 3.16(b)). It should be noted, that the allowed maximum dose



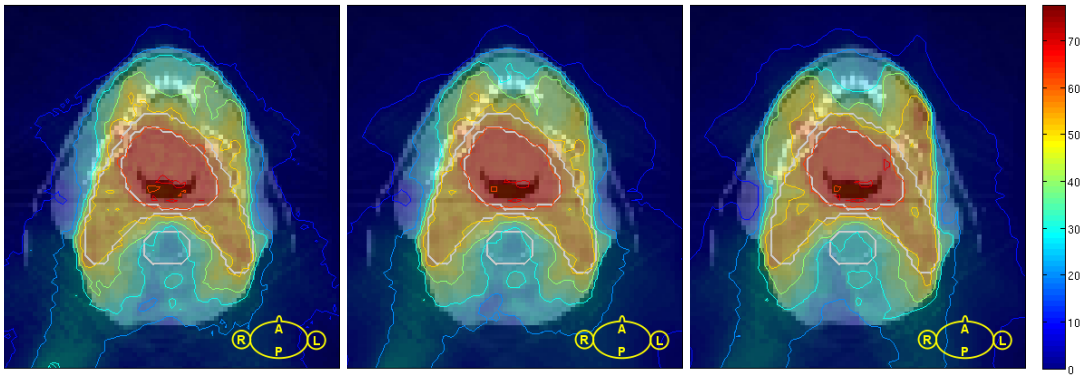

(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

**Figure 3.17:** Nasopharynx, 5 mm square beamlets: The figure compares doses in a transversal slice from the reference FMO algorithm (a) against doses from the hybrid algorithm with the pencil beam dose model (b) and the geometric kernel approximation (c). The lymph drains, the boost and the 5 mm margin around the spinal cord are highlighted white. Indicated are the isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy, 51.3 Gy (95 % of lymph drain dose prescription), 62.7 Gy (95 % of boost dose prescription) and 70 Gy. The dose values in the color legend are given in Gy.

**Table 3.3:** Median, mean and maximum dose values of the nasopharynx case with 5 mm square beamlets. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| Boost | 65.8 | 65.6 | 65.7 | 65.7 | 65.6 | 65.7 | 71.7 | 73.8 | 74.8 |
| Lymph drains | 54.0 | 54.0 | 54.0 | 54.3 | 54.3 | 54.3 | 66.8 | 68.8 | 68.6 |
| Spinal cord 5 mm | 19.1 | 19.7 | 20.7 | 17.4 | 17.8 | 18.6 | 33.8 | 34.8 | 35.5 |
| Left parotis | 14.9 | 15.3 | 15.1 | 14.1 | 14.3 | 14.5 | 23.6 | 24.8 | 25.2 |
| Right parotis | 11.9 | 12.2 | 11.9 | 12.0 | 12.4 | 12.3 | 22.8 | 21.9 | 30.3 |

to the spinal cord was set to 33 Gy in the optimization. Since this value is exceeded and penalized in a few voxels only, the incentive of the optimizer to reduce the spinal cord dose is small. In all cases, there is a small hot spot at the left jaw close to the lymph drain target. This hot spot is even more pronounced in the treatment plan of the Hybrid/GK algorithm. Nevertheless, the clinical constraints could be fulfilled in all three cases.



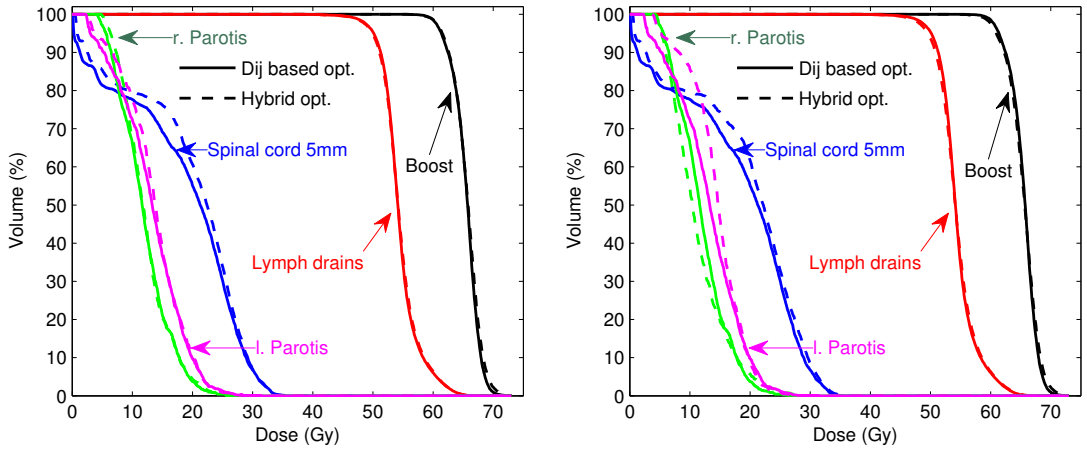

(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

**Figure 3.18:** Nasopharynx, 10 mm square beamlets: The figure compares doses in a transversal slice from the reference FMO algorithm (a) against doses from the hybrid algorithm with the pencil beam dose model (b) and the geometric kernel approximation (c). The lymph drains, the boost and the 5 mm margin around the spinal cord are highlighted white. Indicated are the isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy, 51.3 Gy (95 % of lymph drain dose prescription), 62.7 Gy (95 % of boost dose prescription) and 70 Gy. The dose values in the color legend are given in Gy.

### 3.6.4 Nasopharynx – 10 mm × 10 mm square beamlets

The outcome of the optimization of the nasopharynx case with 10 mm square beamlets (1541 in total) follows the trend of the previous cases. The Hybrid/GK stopped with the highest objective function value of $f = 1330$ after 4 iterations. This value could be reduced with the Hybrid/PB optimization algorithm down to $f = 1187$ after 4 iterations. The absolute minimum objective function value of $f_{\min} = 1105$ was determined with the reference algorithm. Penalty factors and dose prescriptions are identical to the 5 mm square beamlet case.

From the clinical perspective, a clear plan degradation compared to the 5 mm beamlet case can be observed. All plans feature a higher dose to the normal tissue, specifically the dose distribution of the Hybrid/GK. With the high dose region to the left side of the head, this treatment plan would probably not be applied to a patient. Besides, also the 51.3 Gy isodose line (95 % lymph drain prescription dose) is not very target conformal and the dent of this isodose level at the right target lobe leads to an underdosage of the target. The reference optimization algorithm and the Hybrid/PB algorithm could achieve a significantly better dose coverage of the lymph drains. The dose to the boost volume is similar in all cases. The maximum dose in the boost is however increased about 3.5 Gy by the hybrid algorithms. As in the 5 mm beamlet case, the dose to the spinal cord is increased when optimized with one of the hybrid algorithms. This higher dose is however not penalized in most of the voxels due to the allowed maximum dose of 33 Gy. The dose to the left parotis is increased mainly in the treatment plan created




(a) Pencil beam dose model

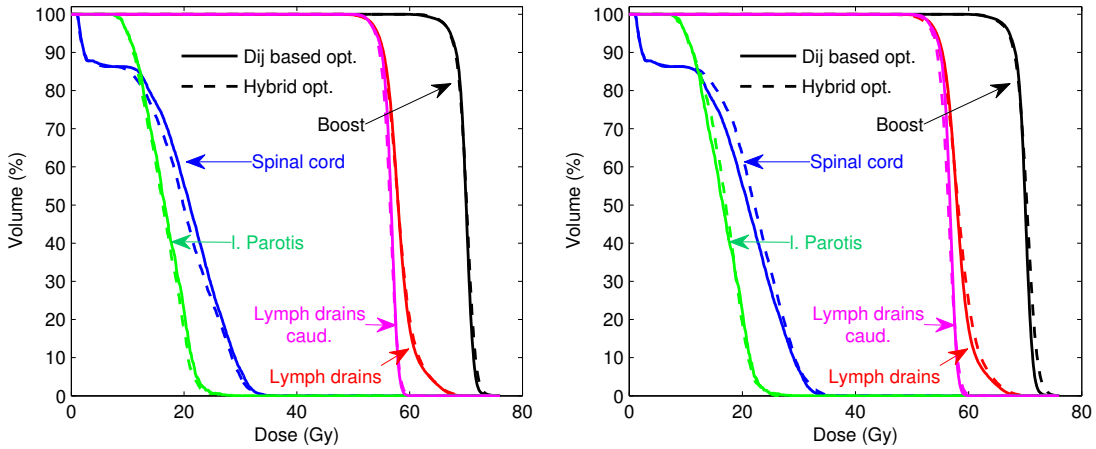
(b) Geometric kernel

**Figure 3.19:** Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm from the nasopharynx case with 10 mm square beamlet size. The left image represents the hybrid algorithm with the pencil beam dose model, the right uses the geometric kernel approximation.

**Table 3.4:** Median, mean and maximum dose values of the nasopharynx case with 10 mm square beamlets. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| Boost | 65.8 | 65.8 | 65.7 | 65.6 | 65.7 | 65.5 | 72.9 | 76.3 | 76.5 |
| Lymph drains | 54.1 | 54.2 | 54.1 | 54.5 | 54.5 | 54.5 | 68.2 | 67.8 | 68.2 |
| Spinal cord 5 mm | 21.2 | 22.4 | 22.2 | 18.8 | 19.6 | 19.5 | 35.5 | 36.1 | 35.9 |
| Left parotis | 13.1 | 13.8 | 14.7 | 13.1 | 13.7 | 14.5 | 31.3 | 30.6 | 28.2 |
| Right parotis | 11.5 | 11.8 | 10.4 | 11.8 | 12.2 | 11.3 | 26.4 | 25.6 | 29.7 |

by the Hybrid/GK algorithm, although its maximum dose could be slightly reduced with this method. All three treatment plans fulfill the clinical constraints (see section 2.7.2). Still, due to the increased dose to the normal tissue, the plan of the Hybrid/GK algorithm would probably not be clinically accepted.




(a) Pencil beam dose model

(b) Geometric kernel

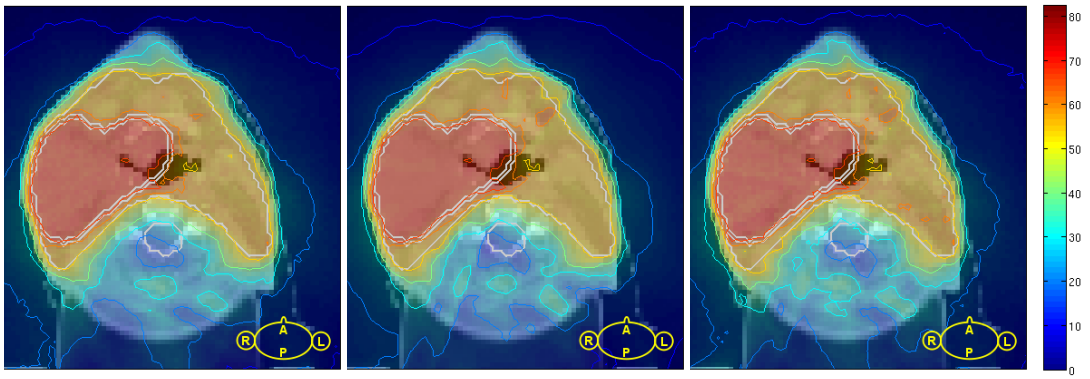
**Figure 3.20:** Larynx, 5 mm square beamlets: Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm. The left image represents the hybrid algorithm with the pencil beam dose model, the right uses the geometric kernel approximation.

### 3.6.5 Larynx – 5 mm × 5 mm square beamlets

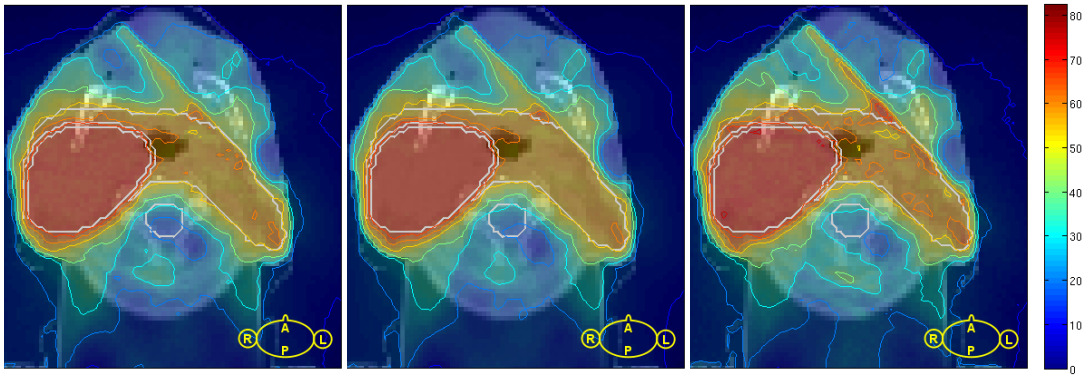
With the 5 mm beamlet side length resulting in not less than 7371 beamlets, this treatment plan is the most complex case to optimize. For the Hybrid/PB optimization algorithm however, this case is comparably easy due to the strong similarities between the pencil beam and MC dose distributions (see section section 3.1.3). The algorithm took 7 iterations to converge. A final objective function value of $f = 1880$ could be achieved. The Hybrid/GK algorithm converged after 7 iterations too with an objective function value of $f = 2060$. The reference objective function value $f_{\min} = 1720$ was determined again with the reference FMO algorithm that was stopped after 100 iterations. Like in

**Table 3.5:** Median, mean and maximum dose values of the larynx case with 5 mm beamlets. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| Boost | 70.3 | 70.1 | 70.3 | 70.1 | 70.0 | 70.2 | 75.2 | 76.9 | 79.0 |
| Lymph drains | 58.0 | 58.0 | 58.1 | 58.3 | 58.3 | 58.5 | 71.8 | 74.8 | 75.4 |
| Lymph drains caud. | 56.8 | 56.5 | 56.4 | 56.6 | 56.5 | 56.4 | 59.6 | 61.7 | 65.8 |
| Spinal cord 5 mm | 20.5 | 20.1 | 22.3 | 18.9 | 19.0 | 20.7 | 35.7 | 36.5 | 37.2 |
| Left parotis | 15.8 | 16.2 | 17.0 | 15.8 | 16.2 | 16.8 | 27.9 | 28.2 | 26.0 |




(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

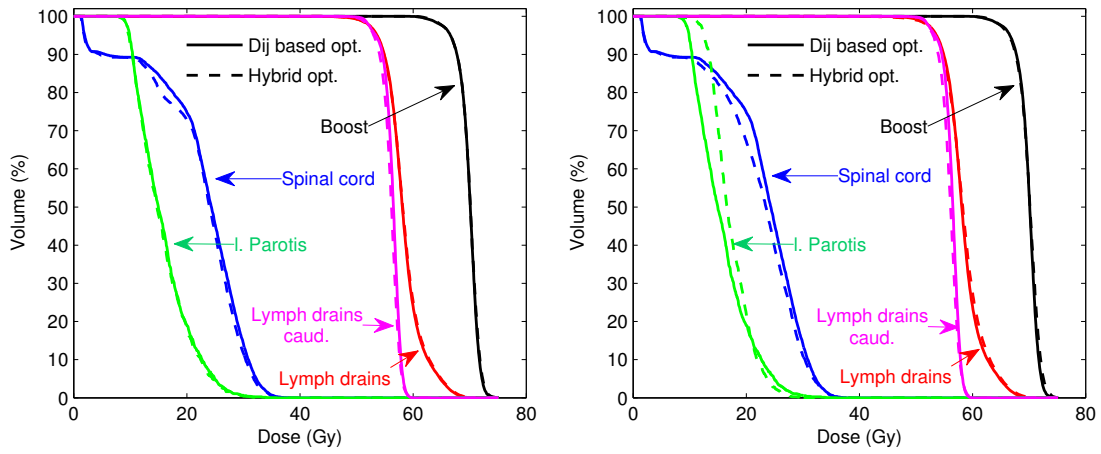



(d) Full FMO (e) Hybrid / PB (f) Hybrid / GK

**Figure 3.21:** Larynx, 5 mm square beamlets: The figure compares doses in 2 transversal slices from the reference FMO algorithm (a,d) against doses from the hybrid algorithm with the pencil beam dose model (b,e) and the geometric kernel approximation (c,f). The lymph drains, the boost and the 5 mm margin around the spinal cord are highlighted white. Indicated are the isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy, 54.7 Gy (95 % of lymph drain dose prescription), 60 Gy, 67.0 Gy (95 % of boost dose prescription) and 75 Gy. The dose values in the color legend are given in Gy.

all patient cases before, the Hybrid/PB outperforms the Hybrid/GK algorithm from a mathematical point of view but neither can achieve the absolute optimum.

If we inspect the dose distributions, some differences between the 3 treatment plans can be observed: the upper dose slice in figure 3.21 shows 2 spots with elevated doses at the anterior left side of the lymph drain target in both hybrid plans. The target dose conformity for the lymph drains and the boost is good and very similar in all three generated treatment plans. All plans seem to favor one particular beamlet, which results in a high dose "needle" from the 320° gantry position, as can bee seen in the lower dose slices (figure 3.21 (d)-(f)). This region of high dose is even more boosted in the Hybrid/GK treatment plan. The reference FMO algorithm and the Hybrid/PB




(a) Pencil beam dose model

(b) Geometric kernel

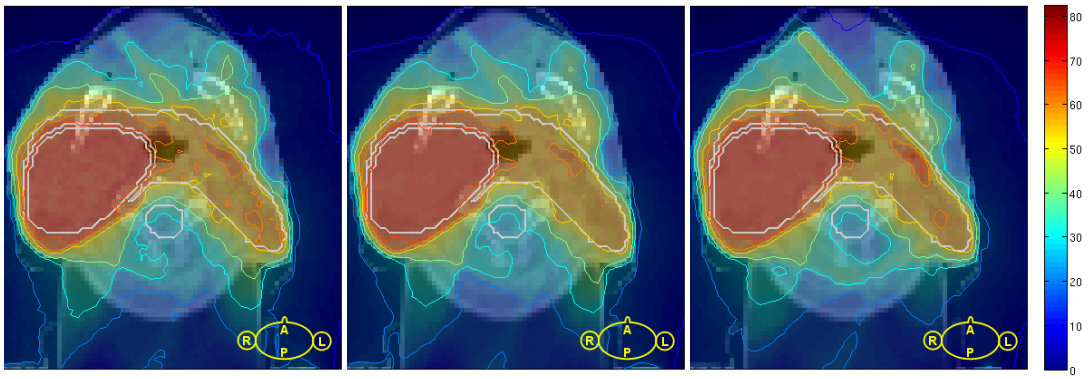
**Figure 3.22:** Larynx, 10 mm square beamlets: Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm. The left image represents the Hybrid/PB algorithm, the right uses the Hybrid/GK method.

algorithm could also achieve a lower dose to the posterior side of the brain. The dose-volume histograms (figure 3.20) do not differ much from each other. The Hybrid/PB shows a better sparing of the spinal cord compared to the reference algorithm. Using the Hybrid/GK algorithm this is contrary. The target dose homogeneity is somewhat worse using the Hybrid/GK method. This decreased homogeneity originates from many high dose islands, which can be identified in the illustration of the transversal dose slices. Again, both hybrid algorithms tend to increase maximum doses. The increase in dose is however moderate in this case and basically limited to the target volumes (table 3.5). The clinical constraints are fulfilled in all cases.

### 3.6.6 Larynx – 10 mm × 10 mm square beamlets

We get similar results if the beamlet size is increased to 10 mm × 10 mm. The complexity of the optimization then reduces to 2004 beamlets. As in all other cases, the reference algorithm could achieve the smallest objective function value of $f_{\min} = 2450$. The Hybrid/PB algorithm stopped at an objective function value of about $f = 2540$ after 4 iterations and the Hybrid/GK algorithm achieved an objective function value of $f = 2750$ after 9 iterations.

If we look at the dose distributions, it can be seen that the same beamlet as in the 5 mm beamlet case is boosted and creates a needle of higher dose. This beamlet gets even more boosted by the Hybrid/GK algorithm. The increase of the beamlet's weight results in a hot spot of about 68 Gy inside the lymph drain target. The 95 % isodose



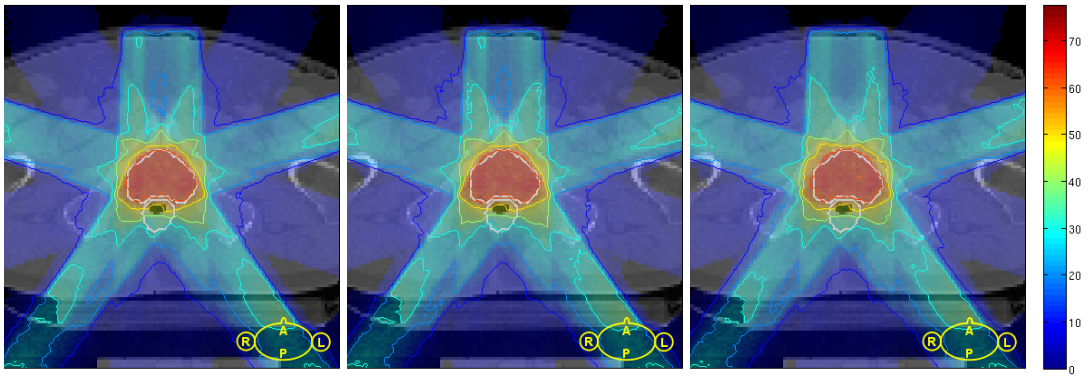

(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

**Figure 3.23:** Larynx, 10 mm square beamlets: This figure compares doses in a transversal slice from the reference FMO algorithm (a) against doses from the hybrid algorithm with the pencil beam dose model (b) and the geometric kernel approximation (c). The lymph drains, the boost and the 5 mm margin around the spinal cord are highlighted white. Indicated are the isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy, 54.7 Gy (95 % of lymph drain dose prescription), 60 Gy, 67.0 Gy (95 % of boost dose prescription) and 75 Gy. The dose values in the color legend are given in Gy.

**Table 3.6:** Median, mean and maximum dose values of the larynx case with 10 mm beamlets. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| Boost | 70.2 | 70.1 | 70.2 | 70.0 | 70.0 | 70.0 | 75.3 | 77.3 | 77.0 |
| Lymph drains | 58.1 | 58.2 | 58.3 | 58.5 | 58.6 | 58.6 | 71.8 | 76.0 | 73.3 |
| Lymph drains caud. | 56.5 | 56.2 | 56.1 | 56.4 | 56.1 | 56.0 | 59.9 | 62.4 | 60.8 |
| Spinal cord 5 mm | 24.2 | 24.4 | 24.0 | 22.2 | 22.3 | 22.0 | 38.0 | 38.4 | 38.7 |
| Left parotis | 14.3 | 14.7 | 16.6 | 15.2 | 15.7 | 17.3 | 33.0 | 32.6 | 29.6 |

lines (of the boost and the lymph drains) wrap around the target volumes very well and look similar in all cases. The average dose to the left parotis is similar in the reference and the Hybrid/PB algorithm and is clearly lower than in the treatment plan of the Hybrid/GK method (see DVH in figure 3.22). This increase in mean parotis dose is however compensated with an about 3 Gy lower maximum dose. Both hybrid method show a slightly better sparing of the spinal cord. Again, the hybrid algorithm tends to a general increase of the maximum doses in most of the volumes. Remarkably, in this patient case this behavior is more distinct in the Hybrid/PB algorithm. Due to the similarities in the dose distributions of the reference and the Hybrid/PB algorithm, both treatment plans can be considered as clinically equivalent. In addition, all three dose distributions fulfill the clinical constraints.



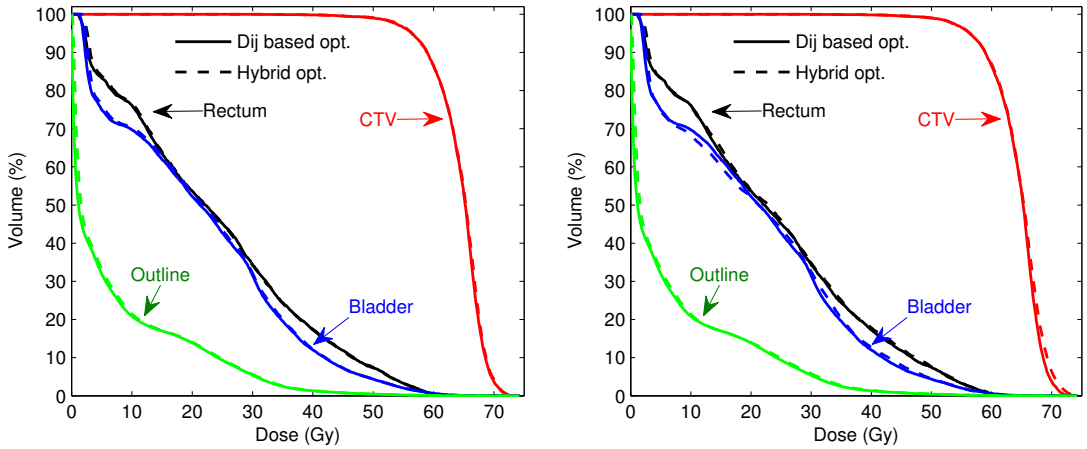

(a) Full FMO (b) Hybrid / PB (c) Hybrid / GK

**Figure 3.24:** Prostate: The figure compares doses in a transversal slice from the reference FMO algorithm (a) against doses from the hybrid algorithm with the pencil beam dose model (b) and the geometric kernel approximation (c). The CTV and the rectum are highlighted white. Indicated are the isodose levels 10 Gy, 20 Gy, 30 Gy, 40 Gy, 50 Gy and 62.7 Gy (95 % of the CTV dose prescription). The dose values in the color legend are given in Gy.

### 3.6.7 Prostate – 10 mm × 10 mm square beamlets

The final case is the prostate treatment plan. Because the MC dose recalculation and the initial pencil beam optimization lead to very similar dose distributions and the number of incident beams is relatively small, this treatment plan is a comparably easy case for the hybrid optimizer. This plan was only selected to demonstrate the versatility of the hybrid algorithm. Therefore we have limited the optimization to the 10 mm square beamlet case. The treatment plan consists of 282 beamlets in total. Due to the low complexity of the optimization problem, the reference FMO algorithm using the limited-memory BFGS method (section 2.1.3) converged after only 23 iterations with an objective function value of about $f_{\min} = 71330$. The Hybrid/PB algorithm reached its final objective function value $f = 72870$ after only 2 iterations and the Hybrid/GK algorithm stopped at about $f = 75450$ after 3 iterations.

The dose distributions of the three algorithms are presented in the transversal dose slices (figure 3.24), the DVHs (figure 3.25) and in table 3.7. The symmetry of the patient setup is recognized by all three optimization algorithms. All three algorithms achieve a dose sparing of the rectum by reducing the weights of the central beamlets of the beam impinging from 12 o'clock. The DVHs are very similar to each other but the Hybrid/GK algorithm produces a slightly worse target dose homogeneity. The maximum doses get increased by the hybrid algorithms of about 1–2 Gy, specifically by the Hybrid/GK algorithm. Despite the small differences in the dose distributions, all three treatment plans can be considered as clinically equivalent and the clinical constraints can be fulfilled in all cases.




(a) Pencil beam dose model

(b) Geometric kernel

**Figure 3.25:** Resulting DVHs from the hybrid optimization algorithm against the results of the full FMO algorithm from the prostate case. The left image represents the hybrid algorithm with the pencil beam dose model, the right uses the geometric kernel approximation.

**Table 3.7:** Median, mean and maximum dose values of the prostate case. Comparison of the full FMO algorithm (Full) against the hybrid algorithm with the pencil beam dose model (PB) and the geometric kernel approximation (GK).

| | $d_{50}$ (Gy) | | | $\bar{d}$ (Gy) | | | $d_{\max}$ (Gy) | | |
|---|---|---|---|---|---|---|---|---|---|
| Volume | Full | PB | GK | Full | PB | GK | Full | PB | GK |
| CTV | 65.2 | 65.4 | 65.4 | 64.4 | 64.5 | 64.7 | 74.2 | 74.0 | 76.0 |
| Rectum | 22.2 | 22.0 | 22.9 | 23.6 | 23.7 | 23.9 | 62.4 | 63.4 | 63.3 |
| Bladder | 21.3 | 21.8 | 21.3 | 21.5 | 21.9 | 21.6 | 66.8 | 66.6 | 67.0 |
| Outline | 1.1 | 1.6 | 1.6 | 6.7 | 7.0 | 7.0 | 66.9 | 68.3 | 70.0 |

## 3.7 Efficiency

With the number of simulated particles during optimization, the efficiencies of both hybrid optimization algorithms and of the reference method were calculated for all clinical cases as explained in section 2.7. In addition, the runtimes of both hybrid algorithms were measured for all patient cases.

The efficiency values of the Hybrid/PB and the reference FMO method are shown in the last three columns of table 3.8. Also included are the number of simulated particles by each method and the resulting mean relative uncertainties of the associated dose distributions. The differences in the number of particles from case to case can

be explained by the varying target volumes, since bigger targets require larger beam areas and thus more beamlets. The efficiency of the Hybrid/PB algorithm is generally very high. Except for the lung cases, the efficiency values are larger than 90 %. In the prostate case, an efficiency of even 99 % could be achieved. These values imply that only few additional particles are required in the hybrid algorithm compared to a forward MC dose calculation at the same dose uncertainty. In the lung cases, the efficiency reduces to 79 % when using $(5\,\mathrm{mm})^2$ beamlets and to 84 % with $(10\,\mathrm{mm})^2$ beamlets respectively. The reason for this decrease are simulated particles from beamlets, whose weights were set to zero during optimization. In such a case, the incremental dose update (section 2.5.2) cannot prevent the waste of particles.

The efficiency of the dose influence matrix based reference algorithm is lower in all cases. The value $\varepsilon_{\mathrm{fmo}}$ can be understood as an upper efficiency limit for this method. Practically, this efficiency cannot be achieved, because the uncertainty of the resulting dose distribution is fluence map dependent and can therefore hardly be estimated from the average beamlet dose uncertainty. The very small effective efficiencies $\varepsilon^*_{\mathrm{fmo}}$ of about 10 % to 30 % are a more realistic efficiency measure of the reference method. This value assumes that the dose influence matrix is calculated at an uncertainty of 0.8 %, which is the same value as the desired uncertainty of the final dose distributions. A low efficiency value of about 15 % implies that more than six times as much particles have to be simulated with the reference method as for the forward dose calculation. These effective efficiency values for the reference method are in a good agreement with the reported values of Laub et al. (2000).

Similar results are achieved with the Hybrid/GK algorithm (see table 3.9). The calculated efficiency values are however generally smaller than the efficiencies of the Hybrid/PB algorithm (average efficiency about 83 %). The reason is, that the pencil beam

**Table 3.8:** Particle efficiencies for the Hybrid/PB optimization and the reference full FMO method. The efficiency values are calculated according to equations (2.66) and (2.87) by recalculating the dose distributions from the fluence map result of the Hybrid/PB optimization. The number of particles $N$ are given in millions ($10^6$).

| | forward MC | | Hybrid/PB opt. | | full MC-FMO opt. | | | efficiencies (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| case | $N_{\mathrm{fw}}$ | $\bar{\sigma}_{\mathrm{fw}}$ (%) | $N_{\mathrm{hyb}}$ | $\bar{\sigma}_{\mathrm{hyb}}$ (%) | $N_{\mathrm{fmo}}$ | $\bar{\sigma}_{\mathrm{dc}}$ (%) | $\bar{\sigma}_{\mathrm{bix}}$ (%) | $\boldsymbol{\varepsilon_{\mathrm{hpb}}}$ | $\varepsilon_{\mathrm{fmo}}$ | $\varepsilon^*_{\mathrm{fmo}}$ |
| Lung† | 2.93 | 0.799 | 3.82 | 0.787 | 100 | 0.172 | 0.285 | 79.1 | 63.1 | 23.0 |
| Lung‡ | 3.16 | 0.780 | 4.51 | 0.711 | 50 | 0.255 | 0.476 | 84.3 | 59.2 | 17.0 |
| Nasoph.† | 20.09 | 0.796 | 23.11 | 0.774 | 100 | 0.420 | 1.003 | 91.9 | 72.2 | 12.7 |
| Nasoph.‡ | 18.39 | 0.800 | 19.97 | 0.788 | 50 | 0.613 | 1.435 | 94.9 | 62.6 | 11.4 |
| Larynx† | 25.30 | 0.796 | 27.04 | 0.794 | 100 | 0.490 | 1.220 | 94.1 | 66.8 | 10.8 |
| Larynx‡ | 24.32 | 0.799 | 24.45 | 0.802 | 100 | 0.495 | 1.210 | 98.7 | 63.4 | 10.6 |
| Prostate‡ | 8.26 | 0.799 | 8.10 | 0.810 | 50 | 0.400 | 0.586 | 99.2 | 65.9 | 30.7 |

†5 mm square beamlets, ‡10 mm square beamlets, *effective efficiencies of the full FMO-MC method (see section 2.7)

dose model of the Hybrid/PB algorithm represents the true dose distributions much better than the geometric kernel approximation, because particle scattering is taken into account. As a result, the hybrid optimization algorithm can estimate better search directions and has to waste fewer particles.

## 3.8 Runtime

The runtimes of both algorithms are shown in table 3.10. The computations were done on a desktop computer with an Intel®Core™i7-860 CPU at 2.8 GHz and 4 GB system memory. Seven calculation threads were used for the optimization and the dose calculation. The task of the remaining extra thread was the visual representation of the dose distribution in the graphical user interface. It should be noted, that the runtimes do not include the calculation of the dose influence matrix of the dose model. This calculation can be done very fast and has to be done only once per patient. The table presents the calculation times $t_{MC}$ for the MC dose calculations, the times $t_{opt}$ spent for optimizing the dose model (2.68) and the total times $t_{total}$ given as the sum of both.

As a general trend, the time for the dose model optimization is significantly smaller in the Hybrid/GK model. Due to the ignored scattering doses, the resulting dose influence matrix has much fewer dose entries than the pencil beam generated dose matrix. Therefore, dose calculation and the computation of the gradient of the objective function can be executed much faster. A second trend is the increased MC calculation time with the Hybrid/GK optimization although the total number of simulated particles are practically the same. The main reason is a higher number of particles per iteration due to larger fluence weight changes in each iteration. In all cases, the Hybrid/PB algorithm

**Table 3.9:** Particle efficiencies for the Hybrid/GK optimization and the reference full FMO method. The efficiency values are calculated according to equations (2.66) and (2.87) by recalculating the dose distributions from the fluence map result of the Hybrid/GK optimization. The number of particles $N$ are given in millions ($10^6$).

| | forward MC | | Hybrid/GK opt. | | full MC-FMO opt. | | | efficiencies (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| case | $N_{\mathrm{fw}}$ | $\bar{\sigma}_{\mathrm{fw}}$ (%) | $N_{\mathrm{hyb}}$ | $\bar{\sigma}_{\mathrm{hyb}}$ (%) | $N_{\mathrm{fmo}}$ | $\bar{\sigma}_{\mathrm{dc}}$ (%) | $\bar{\sigma}_{\mathrm{bix}}$ (%) | $\boldsymbol{\varepsilon_{\mathbf{hgk}}}$ | $\varepsilon_{\mathrm{fmo}}$ | $\varepsilon^{*}_{\mathrm{fmo}}$ |
| Lung† | 2.80 | 0.791 | 5.54 | 0.671 | 100 | 0.160 | 0.285 | 70.0 | 68.4 | 21.5 |
| Lung‡ | 2.58 | 0.798 | 3.32 | 0.777 | 50 | 0.230 | 0.476 | 82.1 | 61.9 | 14.5 |
| Nasoph.† | 19.63 | 0.799 | 25.55 | 0.763 | 100 | 0.406 | 1.003 | 84.1 | 76.1 | 12.4 |
| Nasoph.‡ | 18.65 | 0.797 | 21.99 | 0.778 | 50 | 0.627 | 1.435 | 89.0 | 60.4 | 11.5 |
| Larynx† | 24.56 | 0.800 | 31.88 | 0.794 | 100 | 0.476 | 1.220 | 78.2 | 69.3 | 10.6 |
| Larynx‡ | 25.11 | 0.796 | 33.28 | 0.757 | 100 | 0.513 | 1.210 | 83.5 | 60.5 | 10.9 |
| Prostate‡ | 8.13 | 0.798 | 8.70 | 0.788 | 50 | 0.400 | 0.586 | 95.9 | 64.7 | 30.2 |

†5 mm square beamlets, ‡10 mm square beamlets, *effective efficiencies of the full FMO-MC method (see section 2.7)

**Table 3.10:** Runtimes of the hybrid algorithms.

| | $t_{\mathrm{MC}}$ (s) | | $t_{\mathrm{opt}}$ (s) | | $t_{\mathrm{total}}$ (min) | |
|---|---|---|---|---|---|---|
| case | H/PB | H/GK | H/PB | H/GK | H/PB | H/GK |
| Lung† | 139 | 179 | 141 | 36 | 4:40 | 3:35 |
| Lung‡ | 168 | 83 | 85 | 16 | 4:14 | 1:39 |
| Nasoph.† | 579 | 703 | 652 | 75 | 20:32 | 12:58 |
| Nasoph.‡ | 364 | 645 | 131 | 43 | 8:15 | 11:28 |
| Larynx† | 716 | 956 | 959 | 117 | 27:55 | 17:53 |
| Larynx‡ | 409 | 1066 | 174 | 81 | 9:44 | 19:07 |
| Prostate‡ | 94 | 93 | 6 | 3 | 1:30 | 1:36 |

†5 mm square beamlets, ‡10 mm square beamlets

needs fewer particles until convergence, resulting first in a higher mean dose uncertainty. Only with a final simulation of additional particles, the desired mean dose uncertainty of less than 0.8 % is ultimately achieved.

Although the Hybrid/GK algorithm tends to shorter calculation times due to the significantly smaller size of the dose influence matrix, there is no clear winner with regard to the runtimes. In the larynx and the nasopharynx case optimized with 10 mm beamlets, the situation is reversed and this algorithm takes more time. The shortest runtime of 1:30 min could be achieved with the optimization of the prostate plan. This time should not be considered as a benchmark, since prostate plans are normally optimized with pencil beam algorithms that are precise enough at that body site. More interesting are the runtimes for the lung cases of about 4 minutes. These short runtimes are ideal in a clinical environment, where a great number of treatment plans have to be created every day. The relatively large runtimes of 20–27 minutes for the optimization of the head-and-neck cases may be at the limit of what is clinically acceptable.

In contrast, the calculation of the dose influence matrix for the reference algorithm took generally several hours, depending on the number of beamlets and the number of simulated particles. This calculation was however not parallelized due to technical limitations with the VMC$^{++}$ framework in combination with dose influence matrices. This fact should be considered when comparing runtimes of the reference algorithm with the hybrid optimization algorithm.

# 4 Discussion and conclusion

## 4.1 Sequential hybrid optimization

We developed an efficient algorithm for the optimization of IMRT treatment plans based on Monte Carlo (MC) simulated dose distributions. The aim for this development was to increase the efficiency of current MC based optimization algorithms to reduce their calculation times in order to get the high accuracy of MC algorithms into clinical inverse planning. The increase in computation efficiency is achieved by two proposals: the optimized search direction uses information of an alternative, faster dose calculation algorithm to converge in as few iterations as possible to the optimum. This allows keeping the number of MC simulations small and thus reducing its computational overhead. As the calculation of the optimized search direction uses an only slightly modified objective function, established optimization techniques can be extended to include this algorithm with only minor adjustments. The second idea, called the efficient incremental dose update, reuses already simulated particles from dose calculations in previous iterations of the optimization. This strategy allows a significant reduction of required particles.

### 4.1.1 Quality of optimized treatment plans

This algorithm was tested amongst treatment scenarios at different body sites. The treatment plans include a lung case, two head-and-neck cases and one prostate plan. The complexity of the algorithmic inverse planning was varied by simulating beamlet sizes of 5 mm and 10 mm side length. The algorithm converged in less than 10 iterations in all cases; in most of the cases the convergence was even reached after less than 5 iterations. We evaluated the impact of two alternative dose calculation models for the hybrid optimization algorithm on the treatment plan quality and the computational performance, which are the macroscopic pencil beam (Hybrid/PB, section 2.6.1) and the geometric kernel approximation (Hybrid/GK, section 2.6.2). In all cases, the optimized search direction leads to better treatment plans when using the pencil beam based hybrid dose model. Its advantage is that it featured smaller maximum doses in organs at risk and also achieved more homogeneous dose coverage of the target volumes. Compared to the reference fluence map optimization (FMO) method for MC dose algorithms, especially the Hybrid/PB variant showed good results which are partially indistinguishable from the dose distributions of the reference algorithm. It should however not be concealed, that the hybrid algorithm tends to increase maximum doses that can result in dose hot

spots in healthy tissue. Particularly in the lung case, the maximum dose in the normal tissue increased from 30 Gy to 40 Gy with the hybrid algorithm. These high dose regions are often caused by a strong increase of the weights of single beamlets as a result of contradictory dose information between the dose model and the dose recalculation with MC. This behavior could be suppressed, by adding a regularization term to the objective function that penalizes fluence maps with a high variance (Alber & Nüsslin 2000, Kessen et al. 2000, Webb 2001).

### 4.1.2 Uncertainty estimation of the objective function

The main idea of the incremental dose update is to decrease the uncertainty of the dose distribution in each iteration of the optimization. We demonstrated with four examples that a decreasing average dose uncertainty can lead to a reduction of its objective function value. This fact raises however a new challenge for the hybrid optimization algorithm, since the algorithm has to determine, whether an objective function is decreasing due to an improved treatment plan quality or only on account of a decreased dose uncertainty. Therefore we evaluated three different methods for the estimation of the objective function uncertainty. These are the first order Gaussian error propagation, the error estimation by dose smoothing and the simulated noise method. The error propagation method severely underestimates the error of the objective function since a first order approximation does not take account for the quadratic structure of the objective function, specifically on nearly optimal dose distributions. The diffusion based dose smoothing removes cold and hot spots and reduces dose fluctuations in the target. This leads to an overestimation of the objective function error. Only the method of simulated noise could calculate objective function values that practically do not depend on the noise level and which correspond to an objective function value of zero dose uncertainty. Therefore, the latter method is used in the hybrid optimization algorithm for the calculation of the “true objective function value”. If this value is increasing, the optimization will be stopped in order to prevent plan quality degradation.

It should be critically noted, that the strength of the change of the objective function value depends to a great extend on the set of dose constraints and penalties. If for example the minimum and maximum dose constraint of the target volume differs, noise might not be penalized as the dose values in the target could lie in the allowed dose region. Although very unlikely, the noise could also ”improve” a dose distribution, resulting in a lower objective function value. Thus, the described methods indeed allow an estimation of the variability of the objective function but the deduction of the objective function value of the noise-free dose distribution is simply a heuristic.

### 4.1.3 Efficiency and runtime

The efficiency of the algorithm, which is a measure of particle waste, was generally higher than 80 %. An even higher efficiency of more than 90 % could be measured for the head-and-neck cases and the prostate treatment plan. Compared to the efficiency of the reference algorithm, which could achieve effective efficiencies of maximum 30 %, this is a great improvement which eventually leads to a significant reduction of calculation time. When comparing the impact of the dose models on the efficiency of the hybrid optimization algorithm, it reveals that the use of the pencil beam dose model results in a higher efficiency value in all patient cases. This was the case even in the optimization of the lung, where it is known that the pencil beam doses differ significantly from MC calculated dose distributions (Scholz et al. 2003, Krieger & Sauer 2005). The calculation times of both hybrid algorithms are similar and vary between 1.5–28 minutes. These low runtimes are achieved to a great extend by the VMC$^{++}$ framework for fast clinical Monte Carlo dose calculation (Kawrakow 2001). The Hybrid/GK method allows much shorter times for the calculation of the search direction due to the sparsity of the dose model matrix. Because more particles have to be simulated per iteration due to larger beamlet weight changes, this advantage in computation time is reduced by a longer time for the MC simulation. Compared to the runtimes of the reference FMO algorithm of several hours, great time savings were achieved with the hybrid algorithm.

### 4.1.4 Limitations

In theory, the sequential hybrid algorithm could be classified as a "pseudo optimization algorithm" as the calculation of the search direction cannot assure a decrease of the objective function. When a dose distribution has to be tuned only slightly to be optimal, the dose model (as e.g. the pencil beam algorithm) may predict an improvement of the objective function whereas the MC recalculation results in the opposite. This is the point where the optimization has to stop. The minimum of the objective function – and thus optimality in a mathematical sense – cannot be reached. Accordingly, the convergence error depends on the accuracy of the dose model. We were able to show that the pencil beam dose model leads to objective function values close to the optimum. These values are generally smaller than the objective function values gained with the geometric kernel approximation. It is still questionable, if other sufficiently fast dose calculation models could reduce the convergence error even more.

Currently, the optimization does not take into account if the calculated fluence maps can be realized with a multi-leaf collimator. A further leaf-sequencing step is therefore required which may result in a degradation of the treatment plan quality (Siebers et al. 2002). From a technical point of view, depending on the sequencing algorithm, further MC dose calculations could be required, which would lead to an increase of the total optimization time. One possibility to avoid treatment plan degradation and additional MC simulations is to incorporate the leaf sequencing directly into the optimization cycle (Fippel et al. 2000, Siebers et al. 2002).

### 4.1.5 Similar methods

During the last decade, other publications have also tried to answer the question of how Monte Carlo dose calculations can be incorporated into an inverse planning framework in an efficient manner. The works of Laub et al. (2000) and Siebers et al. (2007) propose similar hybrid optimization methods, which use pencil beam dose calculation algorithms in addition to the accurate Monte Carlo simulation. In (Laub et al. 2000), the gradients of the objective function are approximated by using a pencil beam based dose influence matrix instead of a MC simulated dose matrix. Inside the optimization, a conjugate gradient search direction (based on the gradient approximation) leads to a reduction of the objective function. As in our work, the dose is calculated by Monte Carlo simulation after each iteration in an incremental fashion. In contrast to our work, the dose update handles negative fluence weight changes by simulating negative dose instead of downscaling the previous dose distribution to keep efficiency high. Although quasi-Newton methods – as e.g. the BFGS algorithm – are currently hyped in the optimization community, the conjugate gradient method shows similar performance on many problems (Nocedal & Wright 1999). According to our experience with the limited-memory BFGS technique, at least 20–40 iterations are required in a standard FMO approach in order to get good optimization results (the actual number depends on the treatment plan complexity and can be much higher). The same applies for the conjugate gradients, so that the hybrid algorithm of Laub et al. (2000) still requires a relatively large number of iterations and thus MC simulations. Laub et al. (2000) reduce the number of iterations by pre-optimizing a pencil beam based treatment plan, which then acts as a starting solution for the hybrid algorithm. If the dose distributions calculated by the MC and pencil beam algorithm strongly differ, then it is most likely that the starting solution is far from optimal. In this case, this algorithm still requires a comparable large amount of iterations.

A different approach is published in (Siebers et al. 2007), which is also a sequential optimization algorithm (i.e. each iteration is an optimization). In this work, the difference between an optimized pencil beam dose distribution and a recalculated Monte Carlo dose distribution is calculated and stored as a difference cube after each iteration. In the following iteration of the optimization, the pencil beam dose distributions are corrected by adding the difference cube from the previous iteration on the pencil beam generated dose distribution. Although this method seems different from our hybrid sequential optimization approach, it can be mathematically shown that it leads to the same sequence of fluence maps as our optimized search direction strategy, as long as we omit the line search. The published numbers of required iterations for convergence (Siebers et al. 2007) are similar to our method, which confirms this hypothesis. Contrary to our work, this method does not include an incremental dose update for efficiency improvement, although Siebers makes the educated guess that it might improve calculation times. Our more formal optimization approach, which includes the calculation of a search direction, the minimization of the objective function along the search direction (line search) and the final update of the beamlet weights, offers advantages as standard

optimization concepts can be directly applied in order to improve convergence. The additional line search is computationally cheap and crucial for preventing too long steps, which lead to a premature stop of the optimization. In summary, our new sequential hybrid optimization approach combines the fast convergence of the work of Siebers et al. (2007) with an incremental dose update to reduce calculation time, similar to the work of Laub et al. (2000). Its development was focused on maximum efficiency to keep the number of simulated particles as small as possible.

## 4.2 Reference FMO algorithm

From the perspective of treatment plan quality the reference FMO algorithm is still the gold standard. This could be specifically seen at the maximum doses in the organs at risk, which were smaller in most of the cases compared to the hybrid algorithm. Another advantage of the algorithm is that it can be included into a classical treatment plan optimization concept because of the strict separation between dose calculation and optimization, allowing a very modular design.

Two important issues of this algorithm have to be addressed. First, it is unclear, at which statistical accuracy the dose influence matrix has to be calculated in order to limit the convergence error. The minimal accuracy is not only patient specific but some tests revealed, that is also depends on the actual setting of penalty parameters. The optimization of the presented lung case stabilized if the mean dose uncertainty was reduced to less than 2 %. A change of the penalty parameters of the objective function on the same patient resulted in an unstable optimization over a large range of dose uncertainties and stabilized only after the mean uncertainty of the dose influence matrix was reduced to less than 1 %. In contrast to our work, Siebers (2008) published that an average dose error of 10 % per beam is required, in order to keep the convergence error small. This conclusion was however drawn for his hybrid optimization algorithm on prostate cases only and may be not applicable to a dose influence matrix based optimization approach.

In our work, the reference method was utilized to define the mathematical optimum of each treatment plan and to control the accuracy of the hybrid optimization algorithm. To avoid convergence errors, the dose influence matrix was calculated at a sufficient high accuracy of less than 1 % per beamlet. This high accuracy comes at the price of very long calculation times.

We demonstrated in section 3.2, that the sequential simulation with VMC$^{++}$ of the dose distribution of each beamlet leads to an offset in the calculation time. This comparably long time prevents a fast dose matrix computation with only a few particles. The offset is independent of the number of particles and it is a consequence of the repetitive batch statistic calculation for each beamlet. A possible solution is the use of a different scoring technique instead as e.g. the history-by-history scoring, whose runtime depends on the number of simulated particles.

The second issue is the poor efficiency of the reference method. Due to the decoupling of dose calculation and optimization, information about the importance of each beamlet cannot be exploited during the dose calculation. Thus, the dose of each beamlet has to be simulated with an equal amount of particles or at the same statistical uncertainty. It is inevitable that this strategy requires more particles than a forward Monte Carlo dose calculation. In this work and in the publication of Laub et al. (2000) the low effective efficiencies of 10 %–30 % were calculated.

## 4.3 Dose influence matrix compression

The method of the dose influence matrix compression is a by-product of the development of the hybrid optimization algorithm. It originated from the need to include multiple dose influence matrices into the algorithm in order to test the algorithm for correctness. The use of this technique is however not limited to the hybrid optimization algorithm. All gradient based optimization algorithms in radiation therapy require information about the dose contribution of each beamlet or irradiation segment. If these data are precalculated, the presented compression technique offers one method that halves the usage of memory while keeping the computational overhead minimal. Our measurements could not determine any significant performance regression in the dose calculation and the computation of the gradients of the objective function when enabling the dose matrix compression. On the contrary, the calculation of the gradients could lead to a runtime reduction of up to 20 %, if the dose influence matrix was sufficiently large. The theoretical maximum compression rate of 2 could be achieved in most of the cases, specifically in the case of large dose matrices. Because this algorithm adds only few lines of code (32 lines of C code for the decompression), existing optimization algorithms can be easily adapted for a dose influence matrix decompression “on-the-fly”.

## 4.4 Clinical relevance

The aim of the work was to combine the high accuracy of Monte Carlo dose calculations with an optimization framework that allows calculation times of less than 30 minutes on standard computer hardware. As shown, the inclusion of MC algorithms into the inverse planning helps to avoid underdosage of tumors in proximity to low density tissue, as it is the case in lung tumors or in head-and-neck treatment plans. The results show that this goal could be achieved even at high statistical dose accuracies of 0.8 %. Thus, this algorithm offers a relevant option for clinical use. Most importantly, the algorithm does not interfere with the current clinical workflow but can be seen as a post-processing step after the conventional optimization phase. That is, the treatment plan is first optimized based on an inaccurate dose model (e.g. pencil beams) using the standard tools. In this step, the dose constraints and penalty factors are adjusted until an acceptable treatment

plan is found. After that, the hybrid algorithm can use this solution for its initial guess and its hybrid dose model will be based on the previously generated dose influence matrix. The advantage in doing so is that a repetitive Monte Carlo-based optimization can be avoided as the found constraints and penalties are most likely still a good choice.

There is still some work left, before a clinical application can be considered. First, the particle source used in this work is extremely simplified and has to be expanded to include a realistic energy spectrum. At least the fluence of primary photons and the beam penumbra have to be tuned to match phantom measurements. Better would be a complete simulation of the treatment head. As discussed above, a leaf sequencing step has to be implemented that converts abstract fluence maps into a series of deliverable fields or leaf trajectories. Third, the impact of the CT-to-material conversion process on the dose calculation error has to be improved and the material conversion table must be adapted and commissioned to a specific CT scanner. It was shown, that errors during the CT conversion can lead to dose errors up to 10 % for 6 MV photon beams (Verhaegen & Devic 2005).

The inclusion of Monte Carlo dose algorithms into the inverse planning raises new challenges for the definition of treatment plan parameters. In treatment plans for lung tumors, the PTV includes a relatively large margin around the tumor to account for tumor motion, which is caused by the breathing of the patient. If the planned dose to the PTV margin was the same as the CTV dose (tumor dose), the fluence of beamlets aiming on that margin would have to be severely increased as the energy absorption of the tissue inside the margin is small. This increase in fluence leads to two problems: first, the dose to the normal tissue increases as a consequence. Second, if the tumor moves inside the margin, it also absorbs a higher dose than planned (it is debatable if tumor overdosage is a problem from a medical point of view). This issue might be reduced with the dose-to-water scoring method instead of a dose-to-medium scoring. There is still a large debate of which scoring technique should be clinically used as there are no official recommendations (Chetty et al. 2007, Ma & Li 2011). The advantage of using dose-to-water scoring, is that the clinical experience of the last decades was gained from such data. Also, current equipment for dosimetry reports dose-to-water. On the other hand, dose-to-medium scoring is consistent with reality as it is a physical measure for the energy absorption in the patient. Probably the most straight forward solution to reduce these kinds of problems is the inclusion of organ motion into the optimization and the treatment delivery (Keall et al. 2004, Chetty et al. 2007).

The question, whether a treatment plan has to be optimized on Monte Carlo based dose distributions, can only be answered after a Monte Carlo dose recalculation of the treatment plan is done. If differences between planned dose and MC dose distribution are too large (if e.g. the failure rate in the gamma test (Low et al. 1998) is unacceptable), a MC based inverse planning should be considered.

## 4.5 Conclusion and outlook

We developed a fluence map optimization algorithm for Monte Carlo-based inverse treatment planning in radiation therapy. This algorithm combines the high dose calculation accuracy of Monte Carlo (MC) simulations with the high efficiency of less accurate but faster dose calculation engines. Our hybrid method can be understood as a pseudo optimization algorithm as each search direction is determined by a heuristic, in which the change of the fluence weights is estimated by predicting dose changes with an alternative dose calculation algorithm. For all patient cases we could demonstrate rapid convergence of less than 10 iterations; this is a result of the sequential optimization approach. The particle efficiency of the optimization algorithm – i.e. the statistical impact of each simulated particle – could be improved to about 80–95 % by reusing particles from previous iterations with an incremental dose update. This increase in efficiency significantly reduces the number of required particles during optimization up to 1/10 of the number of simulated particles with a reference optimization algorithm. Thus, clinically relevant calculation times of only a few minutes for the treatment plan optimization, including all MC simulations, can be achieved in combination with the fast VMC$^{++}$ package. Due to these relatively small runtimes, this algorithm is a good candidate for clinical treatment planning of e.g. small lung tumors, where non MC-based dose calculation algorithms are known to be inaccurate. Due to the heuristic character of the hybrid optimization algorithm, optimality in a mathematical sense cannot be guaranteed. Compared to the (slow) reference optimization method, the resulting treatment plans have a tendency of slightly increased maximum doses and small hot spots. Still, the quality of the generated treatment plans are similar to the optimal treatment plans, which were created with the reference method. The best results in terms of plan quality and efficiency could be achieved with the hybrid MC/pencil beam dose model for the calculation of the search directions.

The sequential hybrid optimization method still offers some room for improvement. After each iteration, the MC dose calculation gives accurate results that differ from the prediction of the hybrid dose model. The differences between both dose distributions could be used to iteratively adapt the dose model. If the dose influence matrix of the dose model could be improved – that is, if the differences between dose model matrix and the unknown MC-based dose matrix could be reduced, the hybrid algorithm would even show better convergence (faster convergence, more optimal solution). A correction factor based method was tested, however it could only reduce the final objective function values marginally (the results of this test are not presented in this work). Machine learning methods, such as e.g. the Bayesian linear regression method (Bishop 2006, pp. 152), seem to be more promising. These methods make predictions based on observed data and prior knowledge and lead to good results if the prior knowledge is modeled correctly. In this particular problem, the great challenge for machine learning and other algorithms lies in the huge number of free parameters to be inferred (#beamlets $\times$ #voxels). This implies that a complete dose influence matrix (or at least some parts of it) has

to be derived from only a small number of observed variables, which are given as a series of dose deviation cubes (#voxels $\times$ #iterations). This leads to a heavily underdetermined system of equations which can only produce reasonable results, if as much prior knowledge as possible about the physical and spatial structure of dose distributions is included into this machine learning algorithm. Another challenge is the adaptation of our method to direct aperture optimization (DAO). This type of treatment plan optimization tries to find an optimal set of directly deliverable apertures/fields. This optimization problem is however not convex and requires different optimization techniques (Shepard et al. 2002, Romeijn et al. 2005, Men et al. 2007). The use of a hybrid dose model in the DAO is conceivable, which could also be realized by a sequential optimization approach. It remains open for future research, how an incremental dose update strategy can be combined with the way a DAO algorithm explores its search space. The presented strategies for efficient MC-based optimization are not only an approach for the conventional fluence map optimization but they may be also a foundation for further developments in the Monte Carlo-based inverse planning.

# Acknowledgments

During my work at the German Cancer Research Center and the writing of my thesis, I had great support and contribution from many people. I want to thank

- my supervisor, my first referee and our group leader Prof. Uwe Oelfke, who always provides a great atmosphere in the group,
- Prof. Wolfgang Schlegel for giving me the opportunity to work in his department,
- my second referee Prof. Peter Bachert,
- my other supervisor Dr. Simeon Nill for his help in all respects,
- Dr. Iwan Kawrakow for all his hints and help with the VMC$^{++}$ package and for his constructive feedback about this project,
- Prof. Hans Georg Bock for giving an excellent lecture about algorithmic optimization,
- my combatants during the optimization lecture Dr. Mark Bangert and Corijn Kamerling for the very helpful discussions about optimization problems and algorithms,
- the whole group E0401 at the DKFZ for a great time together,
- the current and former crew of my office "the submarine", foremost Florian Kroupal, Kerstin Hofmann, Dr. Siri Jetter and Marcus Zuber,
- my parents for their support and
- Sofia for countless hours of proofreading and always cheering me up in difficult times.

Thank You!

# List of Figures

# List of Tables

# Bibliography


Alber M & Nüsslin F 2000 Intensity modulated photon beams subject to a minimal surface smoothing constraint *Phys. Med. Biol.* **45**, N49–N52.

Bergman A M, Bush K, Milette M, Popescu I A, Otto K & Duzenli C 2006 Direct aperture optimization for IMRT using Monte Carlo generated beamlets *Med. Phys.* **33**, 3666–78.

Bielajew A F 2001 *Fundamentals of the Monte Carlo method for neutral and charged particle transport*. `http://www-personal.umich.edu/~bielajew/`.

Bishop C M 2006 *Pattern recognition and machine learning* Springer.

Bortfeld T 2006 IMRT: a review and preview *Phys. Med. Biol.* **51**, R363–79.

Bortfeld T, Bürkelbach J, Boesecke R & Schlegel W 1990 Methods of image reconstruction from projections applied to conformation radiotherapy *Phys. Med. Biol.* **35**, 1423–34.

Bortfeld T, Schlegel W & Rhein B 1993 Decomposition of pencil beam kernels for fast dose calculations in three-dimensional treatment planning *Med. Phys.* **20**, 311–18.

Bortfeld T, Stein J & Preiser K 1997 Clinically relevant intensity modulation optimization using physical criteria *in* 'Proceedings of the XIIth ICCR, Salt Lake City' pp. 1–4.

Box G E P & Muller M E 1958 A Note on the Generation of Random Normal Deviates *Ann. Math. Statist.* **29**, 610–1.

Brahme A 1988 Optimization of stationary and moving beam radiation therapy techniques *Radiother. Oncol.* **12**, 129–40.

Brahme A, Roos J E & Lax I 1982 Solution of an integral equation encountered in rotation therapy *Physics in Medicine and Biology* **27**, 1221–9.

Broyden C G 1970 The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations *IMA J. Appl. Math.* **6**, 76 –90.

Chetty I J, Curran B, Cygler J E, DeMarco J J, Ezzell G, Faddegon B A, Kawrakow I, Keall P J, Liu H, Ma C C, Rogers D W O, Seuntjens J, Sheikh-Bagheri D & Siebers J V 2007 Report of the AAPM Task Group No. 105: Issues associated with clinical implementation of Monte Carlo-based photon and electron external beam treatment planning *Med. Phys.* **34**, 4818–53.

Dagum L & Menon R 1998 OpenMP: an industry standard API for shared-memory programming *IEEE Computational Science and Engineering* **5**, 46–55.

Devroye L 1986 *Non-uniform random variate generation* Vol. 4 Springer-Verlag New York.

du Plessis F C P, Willemse C A, Lötter M G & Goedhals L 1998 The indirect use of CT numbers to establish material properties needed for Monte Carlo calculation of dose distributions in patients *Med. Phys.* **25**, 1195–201.

Fippel M, Alber M, Birkner M, Laub W, Nüsslin F, Kawrakow I et al. 2000 Inverse treatment planning for radiation therapy based on fast Monte-Carlo dose calculation *in* 'Monte Carlo 2000 Conference, Lisbon' pp. 217–22.

Fletcher R 1970 A new approach to variable metric algorithms *Comput. J.* **13**, 317–22.

Fox C, Romeijn H E & Dempsey J F 2006 Fast voxel and polygon ray-tracing algorithms in intensity modulated radiation therapy treatment planning *Med. Phys.* pp. 1364–71.

GEANT4 Collaboration 2003 GEANT4: A simulation toolkit *Nucl. Instrum. Meth. A* **506**, 0.

Gill P, Murray W, Saunders M & Wright M 1984 User's guide for npsol (version 2. 1): a fortran package for nonlinear programming Technical report Stanford Univ., CA (USA). Systems Optimization Lab.

Goldfarb D 1970 A Family of Variable-Metric Methods Derived by Variational Means *Math. Comput.* **24**, 23–6.

Hamacher H & Küfer K 2002 Inverse radiation therapy planning — a multiple objective optimization approach *Discrete Appl. Math.* **118**, 145–61.

Hårdemark B, Liander A, Rehbinder H & Löf J 2003 Direct machine parameter optimization with RayMachine in Pinnacle *Ray-Search White Paper* .

Hastings W K 1970 Monte Carlo sampling methods using Markov chains and their applications *Biometrika* **57**, 97–109.

Hestenes M R & Stiefel E 1952 Methods of Conjugate Gradients for Solving Linear Systems *J. Res. Nat. Bur. Stand.* **49**, 409–36.

ICRU 1993 *ICRU Report 50—Prescribing, Recording and Reporting Photon Beam Therapy* International Commission on Radiation Units and Measurements, Bethesda, MD.

Jeraj R & Keall P 1999 Monte Carlo-based inverse treatment planning *Phys. Med. Biol.* **44**, 1885–96.

Kawrakow I 1996 3D electron dose calculation using a Voxel based Monte Carlo algorithm (VMC) *Med. Phys.* **23**, 445–57.

Kawrakow I 2001 VMC$^{++}$, electron and photon Monte Carlo calculations optimized for radiation treatment planning *in* 'Proceedings of the Monte Carlo 2000 Conference' Springer Berlin/Heidelberg Lisbon, Portugal pp. 229–36.

Kawrakow I & Fippel M 2000 VMC$^{++}$, a MC algorithm optimized for electron and photon beam dose calculations for RTP *in* 'Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society' Vol. 2 IEEE pp. 1490–3.

Kawrakow I & Rogers D W O 2001 The EGSnrc code system: Monte Carlo simulation of electron and photon transport.

Keall P J, Siebers J V, Joshi S & Mohan R 2004 Monte Carlo as a four-dimensional radiotherapy treatment-planning tool to account for respiratory motion *Phys. Med. Biol.* **49**, 3639–48.

Kelley C T 1999 *Iterative methods for optimization* SIAM.

Kessen A, Grosser K & Bortfeld T 2000 Simplification of IMRT intensity maps by means of 1-D and 2-D median-filtering during the iterative calculation *in* 'Proceedings of the XIIIth International Conference on the Use of Computers in Radiation Therapy. Heidelberg, Germany: Medical Physics Publishing' pp. 545–7.

Krieger T & Sauer O A 2005 Monte Carlo- versus pencil-beam-/collapsed-cone-dose calculation in a heterogeneous multi-layer phantom *Phys. Med. Biol.* **50**, 859–68.

Laub W, Alber M, Birkner M & Nüsslin F 2000 Monte Carlo dose computation for IMRT optimization *Phys. Med. Biol.* **45**, 1741–54.

Liu D C & Nocedal J 1989 On the limited memory BFGS method for large scale optimization *Math. Program.* **45**, 503–28.

Lopes D 2007 Anisotropic Diffusion (Perona & Malik). `http://www.mathworks.com/matlabcentral/fileexchange/14995`.

Low D A, Harms W B, Mutic S & Purdy J A 1998 A technique for the quantitative evaluation of dose distributions *Med. Phys.* **25**, 656–61.

Ma C & Li J 2011 Dose specification for radiation therapy: dose to water or dose to medium? *Phys. Med. Biol.* **56**, 3073–89.

Mackie T, Holmes T, Swerdloff S, Reckwerdt P, Deasy J, Yang J, Paliwal B & Kinsella T 1993 Tomotherapy: a new concept for the delivery of dynamic conformal radiotherapy *Med. Phys.* **20**, 1709–19.

Men C, Romeijn H E, Taşkın Z C & Dempsey J F 2007 An exact approach to direct aperture optimization in IMRT treatment planning *Phys. Med. Biol.* **52**, 7333–52.

Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H & Teller E 1953 Equation of state calculations by fast computing machines *J. Chem. Phys.* **21**, 1087–92.

Miao B, Jeraj R, Bao S & Mackie T R 2003 Adaptive anisotropic diffusion filtering of Monte Carlo dose distributions *Phys. Med. Biol.* **48**, 2767–81.

Molière G 1947 Theorie der Streuung schneller geladener Teilchen I. Einzelstreuung am abgeschirmten Coulomb-Feld *Z. Naturforsch.* **2a**, 133–45.

Molière G 1948 Theorie der Streuung schneller geladener Teilchen II. Mehrfach-und Vielfachstreuung *Z. Naturforsch.* **3a**, 78–97.

Monz M, Küfer K H, Bortfeld T R & Thieke C 2008 Pareto navigation—algorithmic foundation of interactive multi-criteria IMRT planning *Phys. Med. Biol.* **53**, 985–98.

Naqa I E, Kawrakow I, Fippel M, Siebers J V, Lindsay P E, Wickerhauser M V, Vicic M, Zakarian K, Kauffmann N & Deasy J O 2005 A comparison of Monte Carlo dose calculation denoising techniques *Phys. Med. Biol.* **50**, 909–22.

Németh G & Schlegel W 1987 Radiation Therapy of Intrathoracic Paraaortic Lymph Node Metastases: Three-dimensional treatment planning *Acta Oncol.* **26**, 203–6.

Niemierko A 1999 A generalized concept of equivalent uniform dose (EUD) *Med. Phys.* **26**, 1100.

Nill S, Bortfeld T & Oelfke U 2004 Inverse planning of intensity modulated proton therapy *Z. Med. Phys.* **14**, 35–40.

Nocedal J 1980 Updating quasi-Newton matrices with limited storage *Math. Comput.* **35**, 773–82.

Nocedal J & Wright S 1999 *Numerical Optimization* Springer.

Oelfke U & Bortfeld T 2001 Inverse planning for photon and proton beams *Med. Dosim.* **26**, 113–24.

Perona P & Malik J 1990 Scale-space and edge detection using anisotropic diffusion *IEEE T. Pattern Anal.* **12**, 629–39.

Pflugfelder D, Wilkens J J, Nill S & Oelfke U 2008 A comparison of three optimization algorithms for intensity modulated radiation therapy *Z. Med. Phys.* **18**, 111–9.

Preiser K, Bortfeld T, Hartwig K, Schlegel W & Stein J 1998 Inverse radiotherapy planning for intensity modulated photon fields *Radiologe* **38**, 228–34.

Romeijn H E, Ahuja R K, Dempsey J F & Kumar A 2005 A Column Generation Approach to Radiation Therapy Treatment Planning Using Aperture Modulation *SIAM J. Optimiz.* **15**, 838–62.

Schneider W, Bortfeld T & Schlegel W 2000 Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions *Phys. Med. Biol.* **45**, 459–78.

Scholz C, Nill S & Oelfke U 2003 Comparison of IMRT optimization based on a pencil beam and a superposition algorithm *Med. Phys.* **30**, 1909–13.

Sempau J, Wilderman S J & Bielajew A F 2000 DPM, a fast, accurate Monte Carlo code optimized for photon and electron radiotherapy treatment planning dose calculations *Phys. Med. Biol.* **45**, 2263–91.

Shanno D F 1970 Conditioning of Quasi-Newton Methods for Function Minimization *Math. Comput.* **24**, 647–56.

Shepard D M, Earl M A, Li X A, Naqvi S & Yu C 2002 Direct aperture optimization: A turnkey solution for step-and-shoot IMRT *Med. Phys.* **29**, 1007–18.

Sherman J & Morrison W J 1950 Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix *Ann. Math. Stat.* **21**, 124–7.

Siddon R L 1985 Fast calculation of the exact radiological path for a three-dimensional CT array *Med. Phys.* **12**, 252–5.

Siebers J V 2008 The effect of statistical noise on IMRT plan quality and convergence for MC-based and MC-correction-based optimized treatment plans *J. Phys.: Conf. Ser* p. 012020.

Siebers J V, Kawrakow I & Ramakrishnan V 2007 Performance of a hybrid MC dose algorithm for IMRT optimization dose evaluation *Med. Phys.* **34**, 2853–63.

Siebers J V, Lauterbach M, Keall P J & Mohan R 2002 Incorporating multi-leaf collimator leaf sequencing into iterative IMRT optimization *Med. Phys.* **29**, 952–9.

Siggel M, Ziegenhein P, Nill S & Oelfke U 2011 Boosting runtime-performance of photon pencil beam algorithms for radiotherapy treatment planning *Phys. Medica* . doi:10.1016/j.ejmp.2011.10.004.

Smedt B D, Vanderstraeten B, Reynaert N, Neve W D & Thierens H 2005 Investigation of geometrical and scoring grid resolution for Monte Carlo dose calculations for IMRT *Phys. Med. Biol.* **50**, 4005–19.

Spirou S V & Chui C 1998 A gradient inverse planning algorithm with dose-volume constraints *Med. Phys.* **25**, 321–33.

Thieke C, Nill S, Oelfke U & Bortfeld T 2002 Acceleration of intensity-modulated radiotherapy dose calculation by importance sampling of the calculation matrices *Med. Phys.* **29**, 676–81.

Vassiliev O N, Wareing T A, McGhee J, Failla G, Salehpour M R & Mourtada F 2010 Validation of a new grid-based Boltzmann equation solver for dose calculation in radiotherapy with photon beams *Phys. Med. Biol.* **55**, 581–98.

Verhaegen F & Devic S 2005 Sensitivity study for CT image use in Monte Carlo treatment planning *Phys. Med. Biol.* **50**, 937–46.

von Neumann J 1951 Various techniques used in connection with random digits *Nat. Bureau Standards* **12**, 36–8.

Webb S 1989 Optimisation of conformal radiotherapy dose distribution by simulated annealing *Phys. Med. Biol.* **34**, 1349–70.

Webb S 2001 A simple method to control aspects of fluence modulation in IMRT planning *Phys. Med. Biol.* **46**, 187–95.

Woodbury M 1950 Inverting modified matrices *Memorandum report* **42**, 106.

Xia P & Verhey L J 1998 Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments *Med. Phys.* **25**, 1424–34.

Zhu C, Byrd R, Lu P & Nocedal J 1997 L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization *ACM T. Math. Software* **23**, 550–60.