Characterizing Gram-Charlier Approximation Performance and Use Case Identification for Dose Metrics in Proton Therapy

Zappa van Ommen

Student number: 5453666

To obtain the degree of **Bachelor of Science**
at the **Delft University of Technology.**

Project duration: May 9th, 2025 - August 1st, 2025.

To be defended publicly on **August 19th, 2025 at 10:20.**

Supervisors:
Dr. ir. Danny Lathouwers
MSc. Jelte de Jong

Thesis Committee:
Dr. ir. Danny Lathouwers
Dr. ir. Martin Rohde

Faculty of Applied Sciences
Delft University of Technology
The Netherlands
August 1st, 2025

Abstract

This study investigated the Gram-Charlier (GC) series approximation of probability density functions (PDFs). We applied it to simulated polynomials, patient dose metrics, and voxel-level data from a phantom geometry. The goal was to assess GC use cases and identify predictors of fit quality or divergence. The results across a wide range of simulated polynomial PDFs showed that no single metric (e.g., coefficient sums, constant terms, or truncation errors) consistently predicted GC fit quality. However, a small machine learning classifier model combining multiple metrics achieved 95% precision and 94% recall in identifying good fits, indicating its potential for guiding GC applicability in proton therapy treatment planning.

Patient data showed selective GC success, with low-order truncations yielding accurate fits for certain dose metrics (e.g., Organs at Risk D_{mean} and D_{40}), particularly in one patient. Voxel analysis within the phantom, even at skewed dose probability distributions near the planning target volume (PTV) edge, showed visual and statistical agreement between GC fits and sampled histograms. This was characterized by the measured L^2 and tail errors for some voxels aligning with the expected errors of a realized normal distribution.

As expected, GC divergence was frequent at high orders and for PDFs with large higher-order cumulants. Small sample sizes, limited patient data, simplified 1D uncertainties, and numerical precision also constrained the study's generalizability. Tail fit quality metrics were application-specific, with tail mean absolute error per bin suitable for low-sample PDF histograms. Future work should explore larger datasets, improved numerical precision, 3D uncertainties, and classifier optimization to assess the GC expansion's clinical viability in proton therapy.

Contents

1 Introduction

1.1 Motivation

Conventional radiotherapy, although an effective treatment against cancer cells, can be damaging to healthy tissues surrounding a tumor (Marcus, 2025). In some applications, proton therapy provides a unique advantage; one that can ensure a more targeted radiation dose distribution, even for target volumes deep into sensitive organs such as the brain (National Cancer Institute, 2020). Despite this advancement over conventional methods, this highly localized dose delivery has inherent drawbacks. Due to the steeper gradients of radiation dose along the edges of the treatment volume, proton therapy is more susceptible to setup errors. These errors arise from factors such as uncertainty in patient setup, changes in anatomy, and uncertainty in tissue characterization. In order to build a robust treatment plan, accurate modeling of these uncertainties is crucial. This research dives into statistical tools to describe and measure these uncertainties in dose distributions.

1.2 Objective and Scope of the Study

The overarching research explores probabilistic estimation of voxel doses by means of Polynomial Chaos Expansion (PCE) and sampling. In this context, the PCE is used as a model to translate setup errors to resulting dose at a given position in the target volume. Sampling of the PCE is then used to build a histogram that coincides with the general shape of the dose PDF at this position, from which percentiles are estimated. The primary objective of this study is to determine whether a specific type of mathematical expansion called the Gram-Charlier (GC) series can be implemented to directly approximate the PDF of a polynomial response (such as the PCE) of a random variable. For example, a patient setup error can be modeled as a Gaussian random variable with mean 0 and standard deviation of 3 mm, the PCE will model how the setup error relates to received dose at any given position, and the GC series will analytically approximate the resulting dose distribution from the PCE; potentially acting as a more computationally efficient and precise substitute to sampling methods. In order to better understand the applicability of the approximation, this research will derive the series from its core components, analyze the quality and convergence of the GC series across multiple truncation orders of a range of polynomials, investigate practical metrics to determine truncation limits, assess the fit quality, and explore potential prediction models to determine usability before sampling.

2 Background

2.1 Proton Therapy

Proton therapy is a type of ionizing radiation therapy composed of a beam of high energy protons. As these particles enter human tissue, they begin to deposit energy. For protons (and other heavy ion beams) the peak of energy deposit (Bragg peak) may occur deep into the tissue, and the depth of this peak can be calibrated through changes in beam energy (Marcus, 2025). A key characteristic of proton therapy is that a much lower proportion of energy is deposited before the peak, and beyond the depth of the Bragg peak the energy deposit drops to zero very quickly. The unit for the dose received by a specified volume is called the Gray (Gy), and it measures the ionizing radiation energy absorbed per unit mass (Radiopaedia Contributors, 2024). The main purpose of particle therapy is to destroy cancer cells while minimizing damage to healthy surrounding tissue as much as possible. Compared to radiotherapy with photons (like X-rays or Gammas), the damage of proton therapy on surrounding healthy tissue can be much less significant, meaning cancer cells can be more accurately targeted.

2.2 Uncertainties in Treatment Planning

2.2.1 Sources of Uncertainties and Planning Techniques in Proton Therapy

When the decision has been made to treat a patient with proton therapy, a clinical target volume (CTV) has to be defined (Unkelbach et al., 2018). In essence, this is the volume that contains the entire visible tumor and microscopic extensions that are not visible on a computed tomography (CT) scan. However, this geometry is not sufficient for a treatment plan. This is due to the fact that there are uncertainties in both the imaging component (which is of finite resolution), and the delivery of the dose to the target volume. Each individual patient may require a unique ideal dose, the resolution of the CT scan will introduce margins, and in combination with inevitable setup errors there are guaranteed to be uncertainties in the realized dose distribution (Unkelbach et al., 2018). More specifically, CT scans use Hounsfield units for different tissue types to represent the attenuation of X-rays (Schaffner & Pedroni, 1998), these units are then converted to proton stopping power, but this conversion has uncertainty of its own. This is because X-rays interact

differently with tissue than protons do (Schaffner & Pedroni, 1998). Furthermore, setup errors are multidimensional and consist of a mix of patient misalignment and changes in patient geometry over the course of treatment. Of all the uncertainties discussed, setup errors are the primary focus of this study.

To account for all the present uncertainties, two different methods are commonly used in radiation therapy. One method includes defining a planning target volume (PTV). The PTV encompasses the entire CTV and adds a margin to account for all the uncertainties at once (van Herk, 2004), which is a common choice for photon therapy. However, because proton therapy involves highly localized energy deposition with steep dose gradients, robust treatment planning is preferred (Unkelbach et al., 2018). The sensitivity to setup errors means proton therapy requires a more precise and adaptive treatment plan, in contrast to the static PTV of photon therapy. In robust treatment planning, the delivered dose is computed for multiple discrete error scenarios and then various methods are used to optimize the plan based on the range of chosen scenarios and doses. Conventional methods may include minimizing dose to healthy tissue for worst-case scenarios, or maximizing the minimum dose to the CTV. However, in the context of our study, we investigate the less common probabilistic optimization. This method not only models the range of possible error scenarios, but optimizes based on the probabilities of the input uncertainties. This technique builds upon the principles of robust treatment planning in an effort to account for uncertainties more effectively.

2.2.2 Phantom Geometries

A phantom geometry can be used in treatment planning to simulate the anatomy of both the patient and the CTV (Kainz et al., 2019). This geometry is typically derived from a CT scan, which is discretized into a grid of voxels (small cubic segments), resulting in a finite-resolution three-dimensional representation of the patient and the CTV. Each voxel has a fixed coordinate and can then be given unique properties corresponding to the type of tissue in which it is found. This virtual (phantom) geometry may now be used to quantify voxel doses and other dose metrics. Voxels located within organs at risk (OARs) must be spared, while voxels within the CTV should receive maximum radiation dose (Lee et al., 2019). The phantoms allows us to test and simulate new treatment plans before implementing them (Greener et al., 2022), making them absolutely crucial to account for and minimize the effects of uncertainties on doses to OARs.

2.2.3 Uncertainty Quantification Pipeline

Below follows a brief summary of the treatment planning methodology placed within the broader context of uncertainty quantification in proton therapy.

- Initially, setup uncertainties are defined; for patient misalignment this is typically considered a continuous Gaussian uncertainty with mean 0 and some standard deviation (Unkelbach et al., 2018).
- Next, for each voxel it is determined how setup error relates to dose, in the scope of this research this has been done via PCE. This makes our voxel dose a random variable dependent on our previously defined Gaussian setup error. An example of a voxel's PCE is given in Figure 2.1 below.

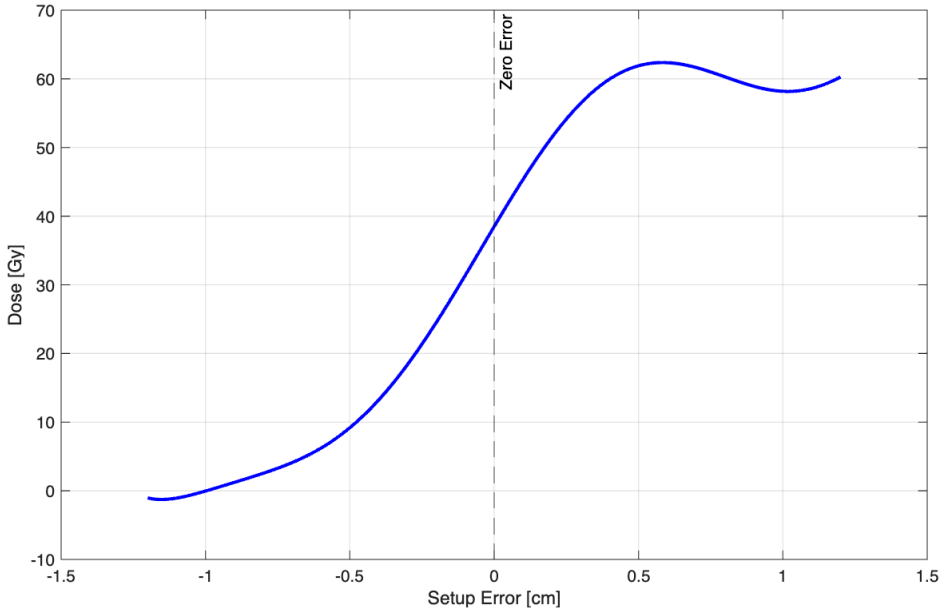


Figure 2.1: PCE of a randomly selected CTV edge voxel from a phantom geometry with 1D setup uncertainty.

- Lastly, we analyze the resulting voxel dose PDF of this random variable. In this emerging area of research into PCE methods, such distributions are commonly obtained through sampling, as shown in Figure 2.2. However, the purpose of this study is to investigate whether the dose distribution can instead be approximated analytically from the PCE coefficients using the GC series. From this distribution, it is then possible to extract relevant dose metrics, such as percentiles for

over/under dose, mean dose, max/min doses, standard deviations, and so on.

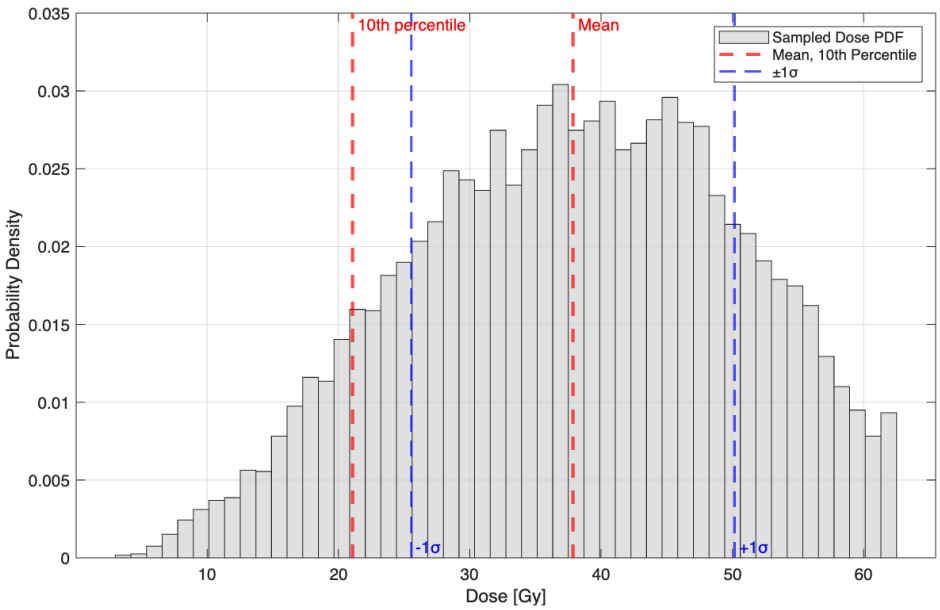


Figure 2.2: The sampled dose distribution from the PCE of Figure 2.1, which is to be approximated directly from the PCE coefficients with the GC series.

2.3 Mathematical Framework

This section will introduce and clarify all the mathematical tools that are central to the uncertainty quantification pipeline from Section 2.2.3

2.3.1 Probability Distributions, Moments, and Cumulants

PDFs describe the range of values a continuous random variable can take with their associated probability. For continuous variables, an outcome between values L_1 and L_2 is defined as:

$$\int_{L_1}^{L_2} p(L)\, dL = P(L_1 \leq L \leq L_2). \tag{2.1}$$

In this case $p(L)$ describes the PDF of a random variable L (for example voxel dose), and P is the total probability of finding a value within the defined

range. Consequently, the area under the PDF determines the probability, and what follows is another key characteristic feature of the PDF, which is that the total area must equal 1:

$$\int_{-\infty}^{\infty} p(L)\, dL = 1. \tag{2.2}$$

The last feature of a real PDF is that it must be non-negative everywhere, as negative probability densities have no meaningful significance in this context. Moments of a random variable are defined in terms of the expectation of the variable. Equation 2.3 shows the expectation operator $\mathbb{E}$ acting on random variable L raised to the nth power. This is then written in integral form, and lastly $\mu_{L,n}$ represents the nth moment of L.

$$\mathbb{E}[L^n] = \int_{-\infty}^{\infty} L^n\, p(L)\, dL = \mu_{L,n} \tag{2.3}$$

The first moment is simply the expectation value of our random variable, also known as the mean. The second moment is related to the variance, the third to skewness, and the fourth to kurtosis (peak sharpness of the PDF) (Jiu & Shi, 2022). Higher-order moments become increasingly more complex, and all these moments are intricately related to each other.

One way to determine moments of a PDF other than the formal definition, is through a generalized version of the moment generating function (MGF) (DeGroot & Schervish, 2012). As the name suggests, the MGF is used to find moments of a random variable, and is commonly defined as:

$$M_L(t) = \mathbb{E}[e^{tL}] = \int_{-\infty}^{\infty} e^{tL}\, p(L)\, dL, \tag{2.4}$$

where $M_L(t)$ is the MGF of random variable L, and t is simply a placeholder variable to find the nth moment of L, for which the method is depicted in Equation 2.5 below.

$$\frac{d^n}{dt^n} M_L(t)\bigg|_{t=0} = \mathbb{E}[L^n] = \mu_{L,n} \tag{2.5}$$

Evaluating the nth derivative of the MGF at $t = 0$ yields the nth moment of random variable L: $\mu_{L,n}$. The MGF however, may not exist for all distributions, therefore a more general function is defined through multiplication with the imaginary unit i: $M_L(it) = \Phi(t)$, the characteristic function (DeGroot & Schervish, 2012). This will be relevant in the derivation of the GC

series in Section 2.3.4, where the characteristic function is introduced and used to find the cumulants of our random variable. These are calculated by taking the logarithm of the characteristic function, and evaluating just like the moment generating function of Equation 2.5. Cumulants are useful descriptors of PDFs that characterize different attributes of a distribution, such as skewness and kurtosis. They also have other unique properties (such as all higher order cumulants being equal to 0 for a standard Gaussian) (Jiu & Shi, 2022) that make them essential for the GC derivation and other similar PDF-approximating expansions.

2.3.2 Polynomial Chaos Expansion

When faced with the problem of an input which is a random variable (with some distribution), and an output which is a complicated function of this input, one can imagine the resulting relationship to be complex. In the context of proton therapy, it is possible to imagine a slight shift in patient alignment to result in an non-linear shift in dose distribution in the patient. PCE is used to form a polynomial approximation of a function with a random variable as input (Jakeman et al., 2019). Depending on the distribution of the input uncertainty, different polynomial bases are used in the PCE, and though for this study the exact details and derivation of the PCE is beyond the scope, it is important to note that the shape of the function we are trying to build a PDF of is a higher-order polynomial stemming from the PCE. Moreover, as the setup errors in this study are assumed to be Gaussian, Hermite polynomials were chosen as the basis for our PCE coefficients. The nature of these polynomials is discussed next in Section 2.3.3.

2.3.3 Hermite Polynomials

The nth-probabilist's Hermite polynomials $He_n(x)$ are defined by applying the differential operator D_x, n times on a standardized normal Gaussian, $\phi(x)$, with mean $\mu = 0$ and standard deviation $\sigma = 1$ as depicted here:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad (-D_x)^n \phi(x) = He_n(x)\, \phi(x), \tag{2.6}$$

with e Euler's number (Brenn & Anfinsen, 2017). The reason Hermite polynomials are used in the PCE for a Gaussian random variable is because they form an orthonormal basis with respect to the standard normal kernel. It is the equivalent of choosing an orthonormal basis for a vector space, which will naturally result in extensive simplifications. As for the GC series, the

Hermite polynomials naturally arise during the derivation. Hence, it is important to recognize them in order to build a more compact and generalized formula.

2.3.4 Gram-Charlier Series

The GC series (Brenn & Anfinsen, 2017) is fully defined as:

$$f_Y(y) = \phi(y) \sum_{n=0}^{\infty} a_n He_n(y) \tag{2.7}$$

Where:

- $\phi(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$ is the standard normal probability density function (PDF).
- $He_n(y)$ are the probabilist's Hermite polynomials at order n.
- a_n are expansion coefficients.
- y are realizations of the random variable Y.

The PDF of the function we are investigating (f_Y) is written as a normal Gaussian (the kernel) multiplied by an infinite sum of Hermite polynomials weighted by coefficients a_n. In our context, Y is a polynomial $Y(X)$ (from the PCE), and X is Gaussian distributed. At its core, the GC Series is modeling a Gaussian with correction terms to arrive at a PDF that is non-Gaussian but assumed to be 'close-to Gaussian' (Capodaglio et al., 2021; de Kock, 2009). As discussed in Section 2.3.3, Hermite polynomials are the naturally chosen orthogonal basis for $\phi(y)$, and a_n are the normalized projections onto that basis. The coefficients a_n are defined as follows:

$$a_n = \frac{1}{n!} \int_{-\infty}^{\infty} f_Y(y) He_n(y)\, dy. \tag{2.8}$$

It may be clear that these coefficients and this series are similar in form to a Fourier series. The reason for this will be made clear during the derivation later in this section. It should be noted that in order to find the coefficients a_n, the function that is to be approximated must be known. However, since our goal is to determine the PDF, we need to work around this by making certain approximations. From these approximations, the general GC A-series will be determined. This version of the series expansion is used when moments or cumulants are known, but the PDF f_Y is unknown. The GC B-series uses the formal definition as seen in Equation 2.7 and 2.8 (Brenn & Anfinsen, 2017).

Gram-Charlier with unknown probability density function

As mentioned earlier, the PDF f_Y is unknown, therefore, we can not analytically determine coefficients a_n from the integral definition. In order to determine a PDF approximation we need to solve for the properties of the higher order correction terms, which are best described by the cumulants of our random variable Y. The reason cumulants are used is because they directly describe deviations from Gaussianity, that is: the third cumulant measures skewness, the fourth kurtosis, and higher orders measure subsequent properties (Capodaglio et al., 2021). This is essential in the derivation, in which we obtain the characteristic function of Y, rewrite it into cumulants, and finally transfer back from Fourier to normal space. To find the cumulants, we first need to establish the characteristic function of f_Y. From Section 2.3.1 and Equation 2.4, the characteristic function $\Phi_Y(t)$ is the expectation value of e^{itY}:

$$\Phi_Y(t) = \mathbb{E}[e^{itY}] = \int_{-\infty}^{\infty} e^{ity} f_Y(y)\, dy. \tag{2.9}$$

This is equivalent to the Fourier Transform of our PDF, which will be relevant later:

$$\mathcal{F}[f_Y(y)] = \int_{-\infty}^{\infty} f_Y(y) e^{ity}\, dy, \tag{2.10}$$

- with $\mathcal{F}$ the Fourier Transform operator.

Again, we do not know the PDF f_Y, which requires us to further rewrite our expression for the characteristic function. The expectation is a complex exponential, which can be written as a Taylor expansion around t=0:

$$\Phi_Y(t) = \mathbb{E}[e^{itY}] = \mathbb{E}[\sum_{n=0}^{\infty} \frac{(itY)^n}{n!}] = \sum_{n=0}^{\infty} \frac{(it)^n}{n!} \mathbb{E}[Y^n] = \sum_{n=0}^{\infty} \frac{(it)^n}{n!} \mu_n, \tag{2.11}$$

where μ_n are the moments of our random variable Y. This expansion will help us to bypass the fact that we do not know the PDF f_Y, while the equivalence to the Fourier Tranform will allow us to apply an inverse Fourier Transform to arrive at the useful final form of the GC A-series (the PDF in question) (Capodaglio et al., 2021). We could take the inverse Fourier Transform of the characteristic function in Equation 2.11, but it is a much more complex task than if we were to reformat further. This is where we introduce cumulants.

The cumulant generating function (CGF), K_Y, works exactly like the MGF in Equation 2.5, but is defined as the natural logarithm of the characteristic function ($\Phi_Y(t) = M_Y(it)$):

$$K_Y(t) = \ln(\Phi_Y(t)). \tag{2.12}$$

Then the nth cumulant $\kappa_{Y,n}$ is the coefficient corresponding to the nth order term in (it) from the Taylor series expansion of $\ln(\Phi_Y(t))$ around $t = 0$ (Jiu & Shi, 2022), resulting in:

$$K_Y(t) = \sum_{n=1}^{\infty} \kappa_{Y,n} \frac{(it)^n}{n!}. \tag{2.13}$$

Deriving the value of these cumulants through this formal definition is quite complex, and is more easily represented through partial Bell polynomials (Kim et al., 2021), which will be discussed briefly in Section 2.3.5.
Now that we have the CGF of random variable Y, the next key step is to introduce the CGF of a standard Gaussian distribution. This PDF $\phi(y)$ has characteristic function $\Phi_\phi(t)$, and CGF:

$$K_\phi(t) = \ln(\Phi_\phi(t)) = \sum_{n=1}^{\infty} \kappa_{\phi,n} \frac{(it)^n}{n!}. \tag{2.14}$$

Note that the cumulants of the standard Gaussian distribution $\kappa_{\phi,1}$ and $\kappa_{\phi,2}$ are 0 (mean) and 1 (variance) respectively, and all higher order cumulants are 0 (Brenn & Anfinsen, 2017). If Y is a standardized variable then $\kappa_{Y,1}$ and $\kappa_{Y,2}$ will be 0 and 1 too. Combining Equation 2.13 and Equation 2.14 through an exponential and a division we get:

$$\frac{\Phi_Y(t)}{\Phi_\phi(t)} = \frac{\exp \sum_{n=1}^{\infty} \kappa_{Y,n} \frac{(it)^n}{n!}}{\exp \sum_{n=1}^{\infty} \kappa_{\phi,n} \frac{(it)^n}{n!}}, \tag{2.15}$$

which leads to:

$$\Phi_Y(t) = \exp\left(\sum_{n=1}^{\infty} (\kappa_{Y,n} - \kappa_{\phi,n}) \frac{(it)^n}{n!} \right) \Phi_\phi(t). \tag{2.16}$$

Now we use Bell Polynomials to simplify the relation between our characteristic functions, so that we can easily take the inverse Fourier Transform to determine our final result. The property of these polynomials we can exploit is that:

$$\exp\left(\sum_{n=1}^{\infty} x_n \frac{(it)^n}{n!} \right) = \sum_{n=0}^{\infty} B_n(x_1, x_2, ..., x_n) \frac{(it)^n}{n!}, \tag{2.17}$$

with $B_0 = 1$. Rewriting this for Equation 2.16, we arrive at:

$$\Phi_Y(t) = \left(1 + \sum_{n=1}^{\infty} B_n(\kappa_{Y,1} - \kappa_{\phi,1}, ..., \kappa_{Y,n} - \kappa_{\phi,n}) \frac{(it)^n}{n!} \right) \Phi_\phi(t). \tag{2.18}$$

Remembering that the first two differences are equal and the rest of $\kappa_{\phi,n} = 0$, we can simplify further to:

$$\Phi_Y(t) = \left(1 + \sum_{n=1}^{\infty} B_n(0, 0, \kappa_{Y,3}, ..., \kappa_{Y,n}) \frac{(it)^n}{n!}\right) \Phi_\phi(t). \tag{2.19}$$

Taking the inverse Fourier Transform of Equation 2.19, of which we know that the inverse Fourier Transform of $\Phi_\phi(t)$ is $\phi(y)$:

$$\mathcal{F}^{-1}[\Phi_Y(t)] = f_Y(y) f_Y(y) = \mathcal{F}^{-1}\left[(1 + \sum_{n=1}^{\infty} B_n(0, 0, \kappa_{Y,3}, ..., \kappa_{Y,n}) \frac{(it)^n}{n!}) \Phi_\phi(t)\right], \tag{2.20}$$

and with properties of multiplication by $(it)^n$ in Fourier space equating to alternating sign derivatives in real space, we arrive at:

$$f_Y(y) = \left(1 + \sum_{n=1}^{\infty} B_n(0, 0, \kappa_{Y,3}, ..., \kappa_{Y,n}) \frac{(-1)^n D_y^n}{n!}\right) \phi(y). \tag{2.21}$$

This is one way of writing the GC A-series (the B-series refers to the formal definition of Equation 2.7), which will now be further simplified for our specific case application. First the sum will be taken from $n = 3$ as the first two terms are 0 when modeling a standardized random variable. Secondly, we can rewrite the derivative term as Hermite polynomials of the standardized random variable Y. We can then finally write the complete GC A-series in terms of cumulants, Bell and Hermite polynomials, and the standard Gaussian kernel:

$$f_Y(y) = \left(1 + \sum_{n=3}^{\infty} B_n(0, 0, \kappa_{Y,3}, ..., \kappa_{Y,n}) \frac{He_n(y)}{n!}\right) \phi(y). \tag{2.22}$$

2.3.5 From Moments of a Polynomial to Cumulants

In order to resolve Sections 2.3.1 to 2.3.4, we must remind ourselves that we are given a polynomial in the form of a PCE with a Hermite basis. These PCE coefficients of different order Hermite polynomials should first be reordered into a list of monomials of ascending orders. One may imagine going from PCE coefficients c_k (for $k = 0, 1, ..., n$) of the polynomial Z (which models the voxel dose as a function of the 1D setup error X):

$$Z(X) = c_0 He_0(X) + c_1 He_1(X) + c_2 He_2(X)..... + c_n He_n(X), \tag{2.23}$$

to monomial coefficients d_n:

$$Z(X) = d_1X^0 + d_2X^1..... + d_nX^n, \tag{2.24}$$

With Gaussian distributed $X \sim N(\mu, \sigma)$. The kth raw moment of random variable Z is $\mathbb{E}[Z(X)^k]$, or:

$$\mathbb{E}[Z(X)^k] = \mathbb{E}[(d_1X^0 + d_2X^1..... + d_nX^n)^k]. \tag{2.25}$$

It is possible to find an analytical solution for different order Gaussians (Papoulis & Pillai, 2002), $\mathbb{E}[X^k]$, hence we must first expand Equation 2.25 into a list of monomials of X. Once this is done, we can perform a multinomial expansion (a generalization of the binomial theorem) (Encyclopedia Britannica Contributors, 2025) to arrive at a form we can analytically solve; similar to Equation 2.24 but up to a higher order as a result of the expansion. The expectation operator is linear, and at this stage we can determine the first two raw moments, through which we normalize the polynomial:

$$Y = \frac{Z(X) - \mathbb{E}[Z(X)]}{\sqrt{\mathbb{E}[Z(X)^2] - \mathbb{E}[Z(X)]^2}} = \frac{Z - \mu_1}{\sqrt{\mu_2 - \mu_1^2}} = \frac{Z - \mu}{\sigma}. \tag{2.26}$$

As was required in Section 2.3.4, Y is now a standardized variable with $\mu_y = 0$ and $\sigma_y = 1$. After this conversion, higher order moments of this standardized variable can be found. Lastly, we must convert these moments into cumulants to complete the approximation. This can be done formally using the definition of the CGF, or through the use of Bell polynomials (Kim et al., 2021). Moments can be represented in terms of cumulants relatively simply with complete Bell polynomials:

$$\mu_n = B_n(\kappa_1, \kappa_2, \ldots, \kappa_n). \tag{2.27}$$

This is one application of Faà di Bruno's formulas, which also has an inverse relationship variant that will be more useful for our purposes (Chou et al., 2006):

$$\kappa_n = \sum_{\ell=1}^{n} \frac{(-1)^{\ell-1}}{(\ell-1)!} B_{n,\ell}(m_1, m_2, \ldots, m_{n-\ell+1}). \tag{2.28}$$

In Equation 2.28, $B_{n,l}$ indicates the *partial* Bell polynomial (Kim et al., 2021).

We now have access to all the tools necessary to build our GC approximation from PCE coefficients, or any polynomial of Gaussians in the form of Equation 2.24.

3 Methodology

3.1 Modeling Approach

3.1.1 Evaluation Metrics

In order to quantify how accurate the GC approximation is, for the purposes of possibly being able to identify indicators or characteristics that can predict use cases, we need to assert universal evaluation metrics. Since all histograms and PDFs will be normalized, comparing them through a set of fixed metrics will give a good summary of performance.

Relative L2 Norm

The L^2-norm (commonly referred to as the L2 norm) (Weisstein, 2025) is a measure of the magnitude of a function, defined in a certain interval, as follows:

$$L^2_{a,b} = \sqrt{\int_a^b |f(x)|^2 dx}, \tag{3.1}$$

where a and b are the bounds for determining the magnitude of a function $f(x)$. In our case we will use the relative L^2-norm between two functions (which means replacing $f(x)$ with the difference between two functions). Since all PDFs will be normalized before analysis, no further normalization is required to make comparisons. For analyzing the relative L^2 error of sampled data, we will use the bin height at bin centers of the histograms in comparison to the PDF function value at the bin centers. For all data and PDFs, the relative L^2 error will be defined for the center of our distributions, between ± 2 standard deviations ($\simeq 95\%$ of the data if close to Gaussian). This measure may give us a good insight how 'close' our approximations are within our chosen bounds (Capodaglio et al., 2021).

Tail Mean Absolute Error

Because of the nature of the context of this study, the tail behavior of these distributions is equally, if not *more* important to quantify. This is because tails correspond to under and over dosage, which are important for voxels inside and outside the CTV respectively. The chosen metric that was verified with many visual inspections was the tail mean absolute error per bin. As mentioned, all distributions are normalized, which means even absolute value differences between functions and bin heights can be a good universal indication of how precisely the densities are being represented by the PDF. The GC

function is evaluated at each bin center beyond ± 2 standard deviations, and if no bins are present, additional "phantom" bins may be introduced with height 0, of equal width and amount as in the other tail (with a minimum of 5). Then, the difference between the GC approximation and bin heights at all these bin centers is averaged, this is repeated individually for right ($\geq 2\sigma$) and left ($\leq -2\sigma$) tails and stored separately.

Freedman-Diaconis Binning

In order to standardize the binning method for all histograms used in this study, we opted to use the Freedman-Diaconis rule. This is important as the number of bins will affect the height of each bin, and subsequently the error when compared to the PDF approximation. The rule finds the recommended bin width through the following formula:

$$\Delta w \;=\; \frac{2IQR(Z)}{\sqrt[3]{n}}, \tag{3.2}$$

(Freedman & Diaconis, 1981) with IQR the inter-quartile range of sampled variable Z, Δw the bin width, and n the amount of samples. Now, the bin count can be easily computed by dividing the range of the histogram values by the bin width.

Acceptable Error Sizes

Because we are working with realized random variables in order to test our distribution's accuracy, this will always introduce some error, even if we theoretically have the exact PDF. One can imagine having a small sample space, of for example 100 data points, with a histogram of 10 bins. This would introduce high variance every time the distribution is sampled, with some bins peaking higher or lower than expected. This is a classic example of the central limit theorem. The more samples we can afford to use, the better our approximation will be. Therefore, it was necessary to test the previous evaluation metrics from this section on a sampled normal distribution with an exact standard Gaussian function as comparison. Since the polynomials of Section 3.1.3 will be sampled 100.000 times, and the patient data will consist of histograms of 1.000 samples, both of these sample sizes were tested on the above metrics. The results are given below in Table 3.1 and Table 3.2.

Metric	Mean	Standard Deviation
Relative L2 Error (Center)	0.10383	0.02183
Left Tail Mean Abs. Error per Bin	0.00627	0.00278
Right Tail Mean Abs. Error per Bin	0.00626	0.00278

Table 3.1: Error statistics from 100,000 standard Gaussians sampled 1,000 times each

Metric	Mean	Standard Deviation
Relative L2 Error (Center)	0.02379	0.00222
Left Tail Mean Abs. Error per Bin	0.00077	0.00015
Right Tail Mean Abs. Error per Bin	0.00077	0.00015

Table 3.2: Error statistics from 1,000 standard Gaussians sampled 100,000 times each

When evaluating our GC approximations on histograms of size $1,000$ we should refer to Table 3.1, and when our histograms are of size $100,000$, then we should refer to Table 3.2. These tables set the standard for the quality of fit for Gaussian-like distributions.

3.1.2 Probability Density Function Approximation Accuracy-Indicators

To investigate the convergence behaviour of the GC Series on our sampled polynomials, the metrics from Section 3.1.1 must be evaluated against different properties of the resulting PDF expressions. These properties may work as universal indicators that can help us identify use cases, to avoid having to sample altogether. Naturally, if we need to sample for every voxel to confirm the GC applicability, we are losing almost all of the benefits of our moment-based approximation methods.

Gram-Charlier Expansion-Coefficient Based Indicators

The PDF expression as determined by Equation 2.22 from each simulated polynomial at all orders (3-16), will be organized in terms of x-powers. After we have done this, we will investigate how properties of the coefficients may indicate to us when and to what extent the series begins to diverge. Such coefficient-based indicators that will be discussed in Section 4 are:

- Absolute sum of coefficients up to expansion order k versus error metrics.
- Size of the first coefficient (the constant term) versus error metrics.
- The absolute difference of the sum of coefficients between truncation orders.

The choice of these properties is based on the fact that the GC series contains monomials up to order k for an expansion of the same order. As high order polynomials have the tendency to grow very large very quickly, the coefficients on higher order terms must be small for the perturbations on the Gaussian kernel to remain small too. This is true within reasonable limits, as these polynomials will almost always be unstable beyond a certain domain. Secondly, it is important to note that the first coefficient term is 1 when the polynomial in question is perfectly Gaussian ($Z(X) = X$), and all other higher-order terms (from the higher-order cumulants) become 0. This is expected with a Gaussian kernel. However, this also means that for PDF approximations with good convergence quality we may expect this term to remain close to 1, as this type of expression would describe a Gaussian PDF with higher order correction terms. Lastly, because series approximation is built on (hermite) polynomials, when it starts to diverge, it usually does so very rapidly. This was the reason for specifically analyzing the change in coefficient size too.

Relative L2 Norm Based Indicators

Since we have already defined the relative L2 error in Section 3.1.1, we may use this to monitor changes in the PDF expression shape as indicators of when it will diverge from the corresponding sampled histogram. As we must assume that we do not have access to the histogram, we should make our comparisons with known functions such as the standard Gaussian, or previous truncations of the PDF in question. Indicators to be tested include:

- Relative L2 error between the PDF expression and standard Gaussian versus error metrics.
- Relative L2 error between derivatives of the PDF expression and derivatives of the standard Gaussian versus error metrics.
- Relative L2 error between consecutive truncation orders of the PDF expression versus error metrics.

Comparing both the normal L2 error of the PDF expression and the derivatives with that of a standard Gaussian may give a good indication of the PDF's proximity to Gaussianity, and consequently the usability of the GC expression as an accurate approximation. The derivatives were included to assess the curvature and 'waviness' of the approximation, which should also be related to deviations from Gaussianity. We should note that the integral form of the relative L2 error is being applied in this case, as we are comparing analytical functions. These will then be plotted against the relative L2 from the error metrics, which are discrete comparisons of the approximation to the sampled bin height. Like coefficient-based indicators, these errors may offer us a specific identification of use cases.

Machine Learning Classification Model

In the event that no single indicator can consistently predict GC Series use cases for generic non-Gaussians, it may be possible to use a combination of different indicators (features) to build a classifier model. In theory, given a certain threshold of acceptable error (such as those given in Table 3.2), a well trained classifier could very rapidly and accurately identify use cases. In a phantom geometry, this would be used to efficiently identify all the voxels eligible for analytical GC approximation methods.

3.2 Implementation

With a thorough understanding of the mathematical tools, it is now necessary to develop the methodology for implementing the theory and realizing the results.

3.2.1 Software and Tools

The software used in this study consists of Python and MATLAB (The MathWorks, Inc., 2024). Python (Foundation, 2024) will be used for all of the following:

- Simulating of polynomials as discussed in Section 3.2.2.
- Sampling of polynomials (both simulated and PCE).
- Building GC approximations of polynomials and patient samples.
- Error metrics.

- Processing and graphing results.
- Machine Learning Classification Model of GC use cases.

The full list of Python libraries along with all the required functions are visible in Appendix B.1. The Python functions have been built to accept a wide range of polynomials and expansion orders, and they work in a modular fashion.
MATLAB is used to visualize voxel locations within the predefined phantom geometry and to sample the PCE's of these voxels.

3.2.2 Simulating and Sampling of Test Polynomials

In order to investigate the general behavior of the GC series in approximating polynomials of Gaussians, 961 different polynomials will be sampled and fitted. These polynomials are of the form:

$$Z(X) = X + \alpha X^2 + \beta X^3 \tag{3.3}$$

With 31 values of α ranging between -0.3 and 0.3, 31 values of β between -0.1 and 0.1, and X as a standard normal distribution with $100,000$ samples. The coefficients were chosen due to their relative proximity to a standard normal distribution ($\alpha, \beta = 0$), while also deviating substantially. For each polynomial: the histogram, cumulants, PDF coefficients, order of the expansion, various error measures (discussed in Section 3.1.2), and bin count will be saved in a Python dictionary. Additionally, the errors will be calculated for GC series truncation orders 3 through 16, giving us 13.454 data points ($31 \times 31 \times 14$) in total.

3.2.3 Small Machine Learning Classification Model

With all the PDF expansions and error metrics data from Section 3.2.2, a relatively small classifier model can be built. The model in question will be a Random Forest Regressor model, commonly used in data science for complex relationships between input and output variables (IBM Corporation, 2024) Some key features used in the model include: coefficients of the polynomial, moments, and truncation order. The model was then tested on a smaller set of newly defined polynomials, with 20 α and 10 β values between -0.1 and 0.1.
As the exact details of machine learning models and their methods are slightly beyond the scope of the study, the more intricate details of this classifier has been omitted. This is because these brief preliminary results primarily act

as a proof-of-concept for future research, and is not the main focus of the study.

3.2.4 Gram-Charlier with Patient Data

Nature of the Patient Data

After the general features and behaviour of the GC approximation on perturbed Gaussians has been investigated, its accuracy in modeling general patient data histograms over entire treatment volumes will be analyzed. This broadening from voxel specific application to dose volume histogram may clarify other roles the GC series could play in treatment optimization. The patient data in question are a collection of 17 dose volume histograms from each of the following measures: Organs at Risk (OARs) $D_{40\%}$, D_{mean}, and $D_{0.03cc}$. This data was acquired from two neuro-oncological patients treated at HollandPTC (J. I. de Jong, 2025). The OARs $D_{40\%}$ data corresponds to the radiation dose absorbed by the 40% of the volume of the OARs receiving the highest dose. D_{mean} represents the mean radiation dose that the OARs receives, and lastly OARs $D_{0.03cc}$ represents the minimum dose that is received by the mostly irradiated 0.03cc part of the structure (Lee et al., 2019).The dose volume histograms were sampled over 1.000 treatment scenarios, and each sample indicates the associated dose type received by the given volume. These metrics may be used to analyze how effective and how damaging the treatment was.

Reason for Testing

To optimize the treatment plan, we should minimize the radiation exposure of the OARs while trying to maximize the exposure for the CTV. Testing the GC approximation on these measures is effectively a check of what these distributions look like in practice. For $D_{40\%}$ and $D_{0.03cc}$ it may not yet be feasible to use moment-based approximations because we need to estimate optimization gradients of these measures. These gradients tell us how our dosage is dependent on setup parameters. Specifically, we need the gradients with respect to the beam weights (the proton intensities), which is crucial for optimizing the treatment plan. For D_{mean} however, we *could* find gradients, and thus we may be able to apply moment-based approximation for treatment optimization. Knowing the possible applications to mean dose, and potential future application to percentiles (if we can accurately estimate gradients for these), we may be able to broaden the use of PCE in combination with moment-based methods for treatment planning in proton therapy.

The benefits of this may be computational speed as well as the precision that a continuous distribution offers over discrete histogram alternatives. It is for these reasons that it is relevant to analyze which measures may fall under GC use cases.

Test Metrics

For all samples of OARs $D_{40\%}$, $D_{0.03cc}$, and D_{mean}, we will use the Python functions from Appendix B.1 to determine the bin count, empirical moments, and lastly the GC approximation of different orders. For each of these truncations, the evaluation metrics from Section 3.1.1 will be recorded, and then the mean and standard deviation of these metrics will be determined across all 17 histograms. This way the error of the fits can be compared to Table 3.1 to assert the quality of the approximation for each truncation order. The results should give us some insight into which of the three data types may be candidates for moment-based methods with PCE.

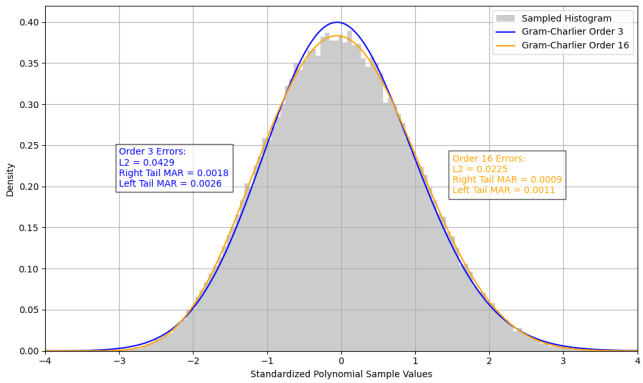
3.2.5 From Phantom Geometry to Gram-Charlier Approximation of Voxel Dose Distribution

As discussed in Section 2.2.2, a phantom geometry is one of the most important tools for simulating treatment plans and error scenarios in proton therapy. In combination with PCE (introduced in Section 2.3.2), a model of voxel dose dependence on setup errors is made. The geometry and treatment plan used in this study were designed and simulated at TU Delft Medical Physics and Technology (J. de Jong, 2025). The phantom is set up as follows: a volume of $45\,\text{mm} \times 45\,\text{mm} \times 45\,\text{mm}$ with $1\,\text{mm}^3$ voxels encompasses a centered spherical CTV of radius 9 mm. A treatment plan has been made with a 6 mm margin around the CTV, optimized to receive 60 Gy, while dosage to all tissue outside the CTV is minimalized (ideally 0 Gy). This makes the PTV a centered sphere of radius 15 mm. The PCE was constructed for each voxel up to the 7th order Hermite polynomial for a systematic setup uncertainty in the x-direction. This setup uncertainty was made to be Gaussian with $\mu = 0$ and $\sigma = 3$ mm. The proton beams were simulated to travel along the positive z-axis, so the setup shifts occur perpendicular to the beam direction. Five different voxels will be tested with the GC approximation as well as sampling to verify the fit quality. Four voxels will be chosen outside the edge of the PTV and one right on the border (where the steepest dose gradient is present).

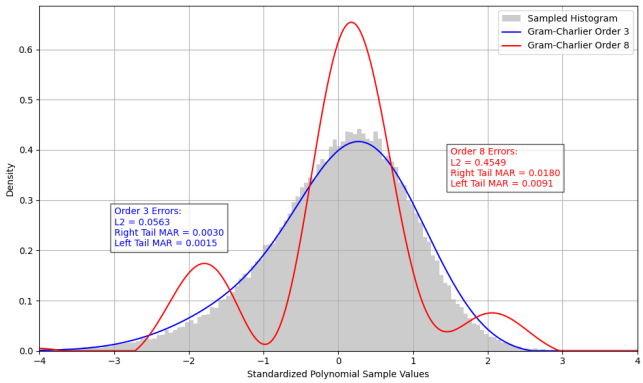
4 Results

4.1 Gram-Charlier on Simulated Polynomials of Gaussians

This section explores both the visual and quantitative fit quality of GC series approximations on polynomials defined in Section 3.2.2. First we will visually inspect two samples of the simulated test polynomials. Next, the fit quality based on error metrics from Section 3.1.1 is compared with indicators introduced in Section 3.1.2. Lastly, the results of the classifier defined in Section 3.2.2 are presented.



(a) $\alpha = 0.02$, β = -0.0133



(b) α = -0.1, $\beta = 0.02$

Figure 4.1: Sampled histogram of the standardized polynomial versus Gram-Charlier polynomial PDF fits for two different (α, β) parameter combinations and truncation orders.

When discussing L^2 and tail errors, the reader may refer to Figures 4.1 (a) and (b) as a guide for visually understanding "good" and "bad" fits, as defined in Table 3.2. Figure (a) shows a polynomial with a relatively low valued (α, β) pair, recall that with $\alpha, \beta = 0$ we model a perfect Gaussian. One can see that a higher order fit (in yellow) offers better performance than a low order (in blue). However, for larger (α, β) pairs, such as the one depicted in Figure (b), the divergence properties of the GC series are clearly present; with a low order visibly (and statistically) approximating the sampled polynomial much better than a higher order.

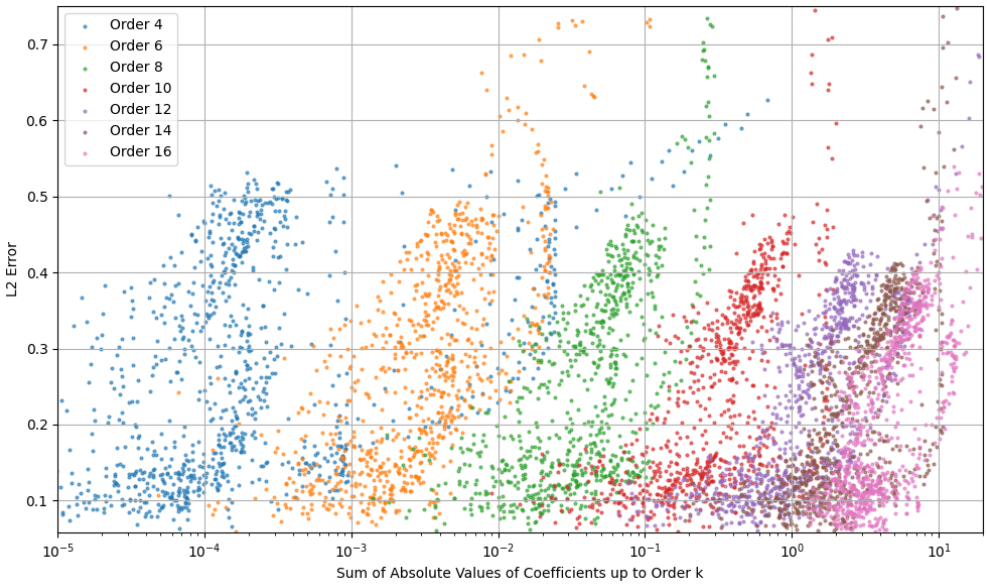


Figure 4.2: Sum of Gram-Charlier expansion coefficients of even order truncations between 4 and 16 versus the L^2 center error of the approximation and the sampled polynomial.

When analyzing Figure 4.2, we can see that for different truncation orders some clustering occurs at relatively low L^2 errors. For order 16 (visible in pink), the data is clustered on the right hand side and has a much more shallow gradient (on logarithmic scale) than order 4 on the left. This is to be expected, as with more terms there are more coefficients in the sum, and thus an order 16 expansion with some sum of coefficients should generally yield a lower error than that of order 4 at the same sum value, as the average size of the perturbation terms would be lower. The left and right tail mean error per bin showed almost identical relationships, and have been omitted in the entire section since the L^2 error was chosen as the primary indicator.

The large clusters shows that the correlation between sum of coefficients size of a GC expansion and its fit quality is poor.

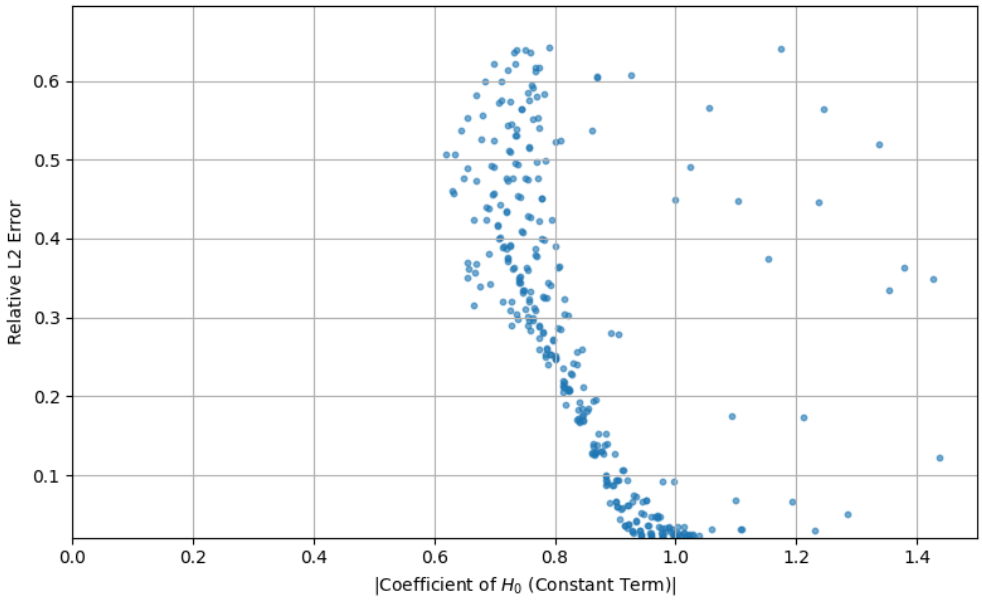
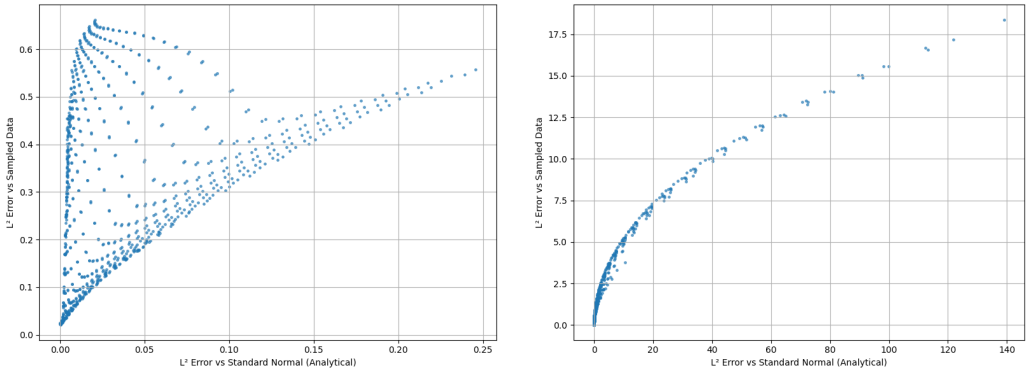


Figure 4.3: Relationship between the size of the constant term in the 4th order Gram-Charlier expansion and the L^2 center error compared with the sampled polynomial.

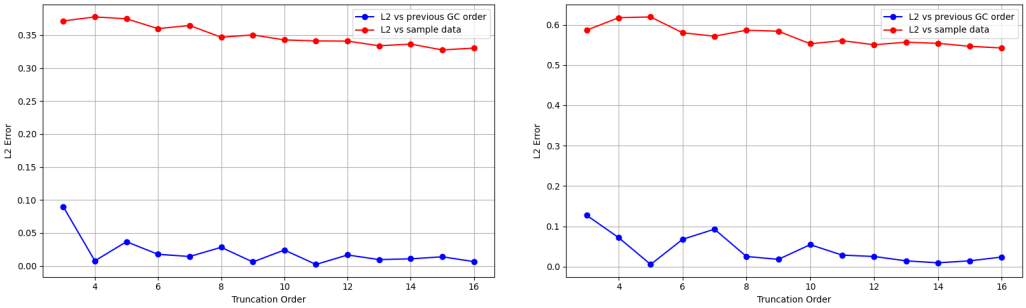
Just like Figure 4.2, the results for the size of the constant term indicator in Figure 4.3 coincided with the reason it was chosen in Section 3.1.2. The first coefficient term is 1 when the polynomial in question is perfectly Gaussian ($Z(X) = X$), and all higher-order terms become 0. We expected this first term to remain close to 1 for Gaussian-like distributions, which can be seen in the figure by the lowest error generally coinciding with data clustered around 1.0 on the horizontal axis. However, it is clear from the figure that this is not a perfect indication of L^2 error found from the sample (and by extension the tail errors). We can recall Table 3.2, which shows that for a histogram of 100.000 samples a perfect fit function should have a mean relative L^2 error of close to 0.02379. In Figure 4.3 there are a non-negligible number of points with constant term 1 and much higher L^2 error. The figures for higher order truncations showed the same relationships, and have therefore been omitted.



(a) Truncation Order 4 samples (b) Truncation Order 7 samples

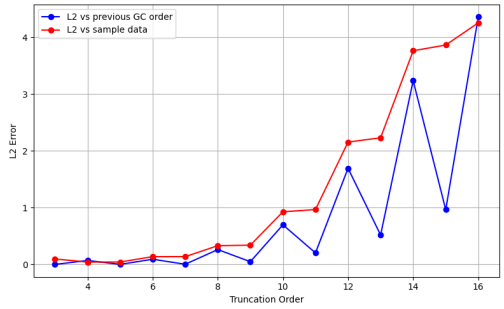
Figure 4.4: Comparison of the analytical L^2 center error between the Gram-Charlier approximation and a standard normal distribution versus the approximation error relative to the sampled polynomial.

Figures 4.4 (a) and (b) depict the results of comparing our analytical GC fit function with that of a standard normal, as a test of Gaussian-like shape and consequently an indication of fit quality. One can see from (a) that at lower L^2 errors the data points rise almost vertically. Although we can see from (b) that a larger analytical error results in a larger sampled data error, we are really only interested in correctly identifying good fits (≈ 0.02379). The shape of figure (a) suggests that in low error scenarios there are possibly a combination of other characteristics that may be dominant in defining the fit quality. One can also see that it appears that multiple curves from this initial rise converge together in the higher error values. These are split into what looks like distinct lines. This is almost certainly due to the α and β values chosen for our polynomials, and if many more values were sampled within our range then the space between these curves would likely be filled up. Just as Figures 4.3 and 4.2, the data appear too noisy for any individual lower order predictions of good fits.

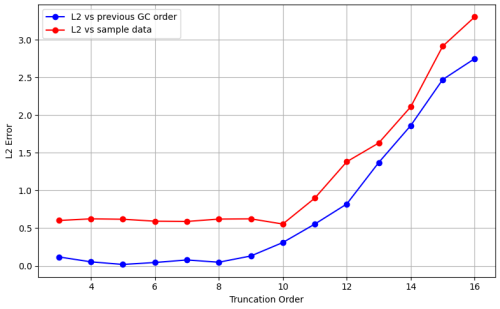


(a) α = -0.16, β = -0.04

(b) α = 0.28, β = -0.05



(c) α = 0.00, β = 0.04



(d) α = 0.28, β = -0.06

Figure 4.5: Comparison of the analytical L^2 center error between consecutive truncations of the Gram-Charlier expansion and the approximation error relative to the sampled polynomial.

Figure 4.5 depicts the use of the L^2 error between consecutive truncations of the GC expansions as an indicator of divergence of the fit. As may be clear from comparing figures (a) and (b) to (c) and (d), as long as the L^2 error between truncation orders is stable, so is the error to the sampled PDF. This relationship does not extend to predicting the starting value of the L^2 error of the sampled fit, only how it changes. The first point is always the relative error to a standard Gaussian, which was already discussed in Figure 4.4.

4.1.1 Brief Machine Learning Classification Model for Gram-Charlier Use Cases

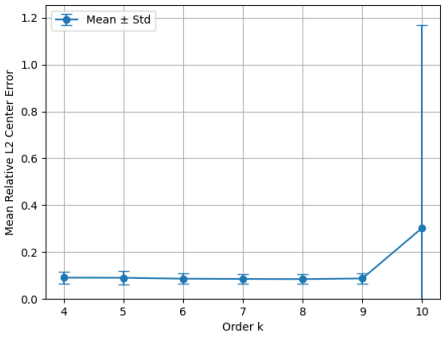
Class	Precision	Recall	Support
Bad Fit (0)	0.74	0.77	505
Good Fit (1)	0.95	0.94	2311
Accuracy	0.912 (on 2816 samples)		
Macro Avg	0.85	0.86	–

Table 4.1: Classification report for identifying good polynomial fits ($L^2 < 0.03$) of test data for machine learning model from Section 3.2.2.

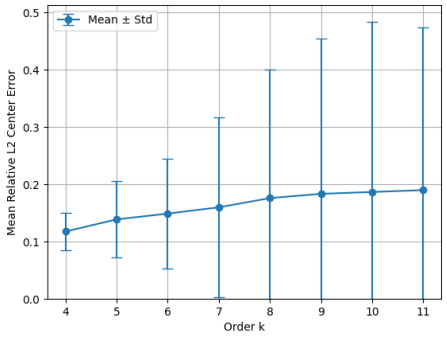
Table 4.1 is the result from applying the classification model on a small test sample, both defined in Section 3.2.2. The most relevant elements are the precision and recall of the good fit (class 1). The precision tells us what proportion of the fits that are classified as "good" are actually good fits. For our test this value was 95%. The recall refers to the proportion of the good fits in the data that are correctly labeled by the model. This was 94% for our test. The overall identification accuracy of the model was 91.2% on a sample of 2.816 different polynomials. The test data clearly had some bias, with 2.311 class 1 samples versus 505 class 0 (bad fit) samples.

4.2 Gram-Charlier on Patient Data Dose-Statistics

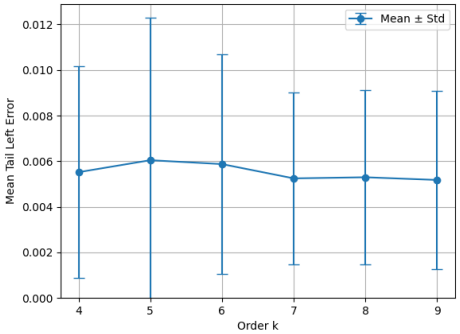
In this section, the realized dose-metrics samples of two oncological patients from Erasmus MC, Rotterdam were empirically fitted with the GC approximation and evaluated against the error measures from Section 3.1.1. This was done on a collection of 17 histograms with 1000 samples each, for both OARs D_{mean} and OARs $D_{40\%}$ error metrics. The mean error and standard deviation across all 17 histograms per metric is plotted against the GC truncation orders for each error type.



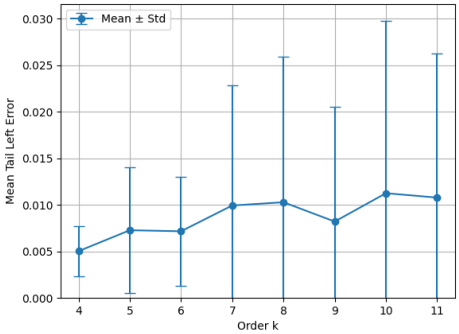
(a) Patient 1



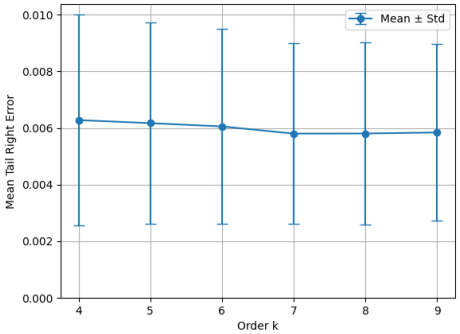
(b) Patient 2



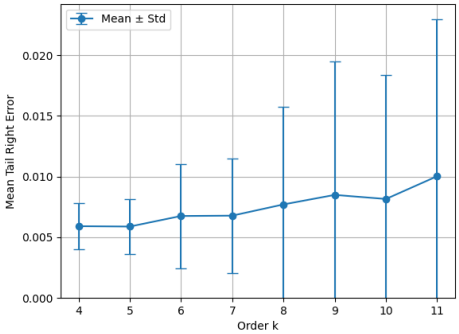
(c) Patient 1



(d) Patient 2



(e) Patient 1



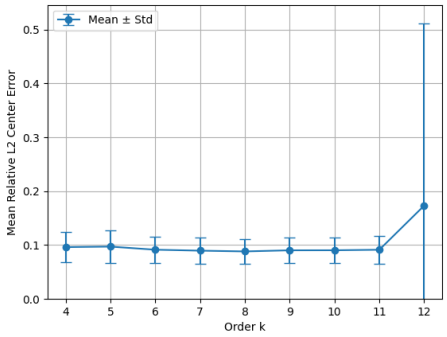
(f) Patient 2

Figure 4.6: Three error types for patient 1 and 2 of OARs D_{mean} dose metric, with mean and standard deviation over the patient population plotted across all relevant truncation orders.

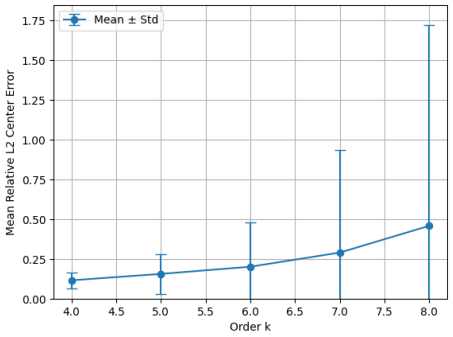
In Figure 4.6, we see the mean relative L^2 center error, along with the left and right tail errors across all 17 histograms for both patients. We should

refer to Table 3.1 when evaluating the quality of the fit, which must be in close proximity to: $L^2 = 0.10383$, and left/right tail mean absolute error per bin of 0.00627/0.00626 respectively.
For patient 1 in (a), the mean L^2 center error was stable and within the acceptable bounds up to order 9. After this truncation, the error and standard deviations of the histograms spike. This indicates that for the OAR D_{mean} dose, truncations up to order 9 provided effective models—matching or exceeding the accuracy of a standard Gaussian fit based on 1.000 samples. For patient 2 in figure (b) we see that only the first truncation is within the acceptable error, and the standard deviation grows quickly. This was primarily due to the approximation on one of the histograms having poor fit quality and diverging very rapidly in successive orders. However, for fourth-order truncations of OAR D_{mean} the data was accurate for both patients across the 17 histograms.

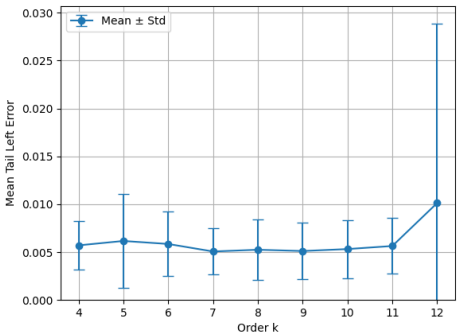
Figures (c) and (e) show a relatively stable tail performance of the GC series up to order 9 for patient 1, with the 9th order coinciding with the most accurate fit. Beyond the 9th order the tail error diverged too much to include in the plot. The tail performance was approximately equal and on average slightly better than that of the ideal case, though the standard deviation was substantially higher than presented in Table 3.1. Figures (d) and (f) also had good performance, but only for the lowest orders (4/5).



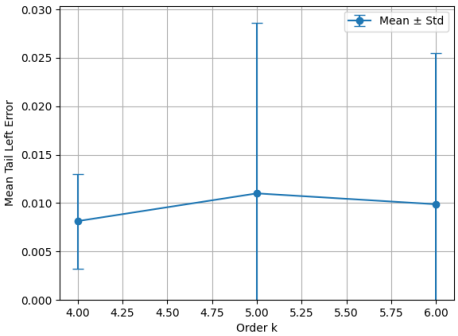
(a) Patient 1



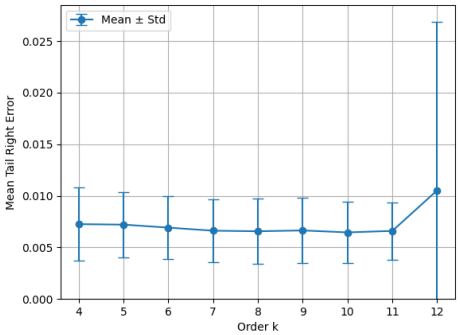
(b) Patient 2



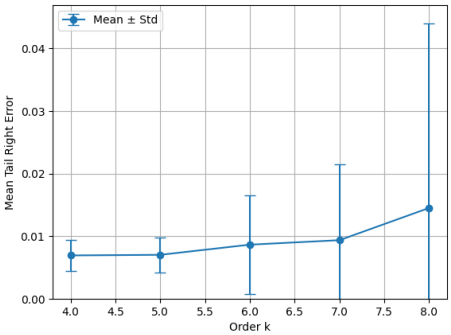
(c) Patient 1



(d) Patient 2



(e) Patient 1



(f) Patient 2

Figure 4.7: Three error types for patient 1 and 2 of OARs $D_{40\%}$ dose metric, with mean and standard deviation plotted across all relevant truncation orders.

Figure 4.7, which evaluates GC on OARs $D_{40\%}$, has very similar properties to the OARs D_{mean} data from Figure 4.6. The L^2 and tail error from

Figure 4.7 (a), (c), and (e) are all within the expected error up to a truncation order of 11. For Patient 2, we again see a worse overall performance, but sufficient at low truncation orders. There are many possible reasons for this difference between patient data: it could be related to CTV size or location, patient anatomy, or even the breathing and 'stillness' of the patient etc. Despite the limited samples, the overall fit quality was sufficient when compared to the standard Gaussian ideal scenario.

It should be noted that the OARs D0.03cc metric was omitted due to very poor fit quality, and other metrics such as CTV D98 and CTV D2 were tested but omitted because they only contained 1 histogram per patient.

4.3 Gram-Charlier for Voxels in a Phantom Geometry

The following section explores the GC series applied to the phantom geometry described in Section 3.2.4. We begin by identifying voxels around the edge of the CTV, followed by a deeper dive into one of the voxels at the CTV edge. The results of this voxel include the sampled dose distribution, a standardized version of the data with a GC fit, a mapping of the PCE of the voxel, and lastly a return to the original voxel dose distribution with an unstandardized GC fit. Also, the standardized GC approximations of the remaining voxels and their fit quality will be shown and summarized.

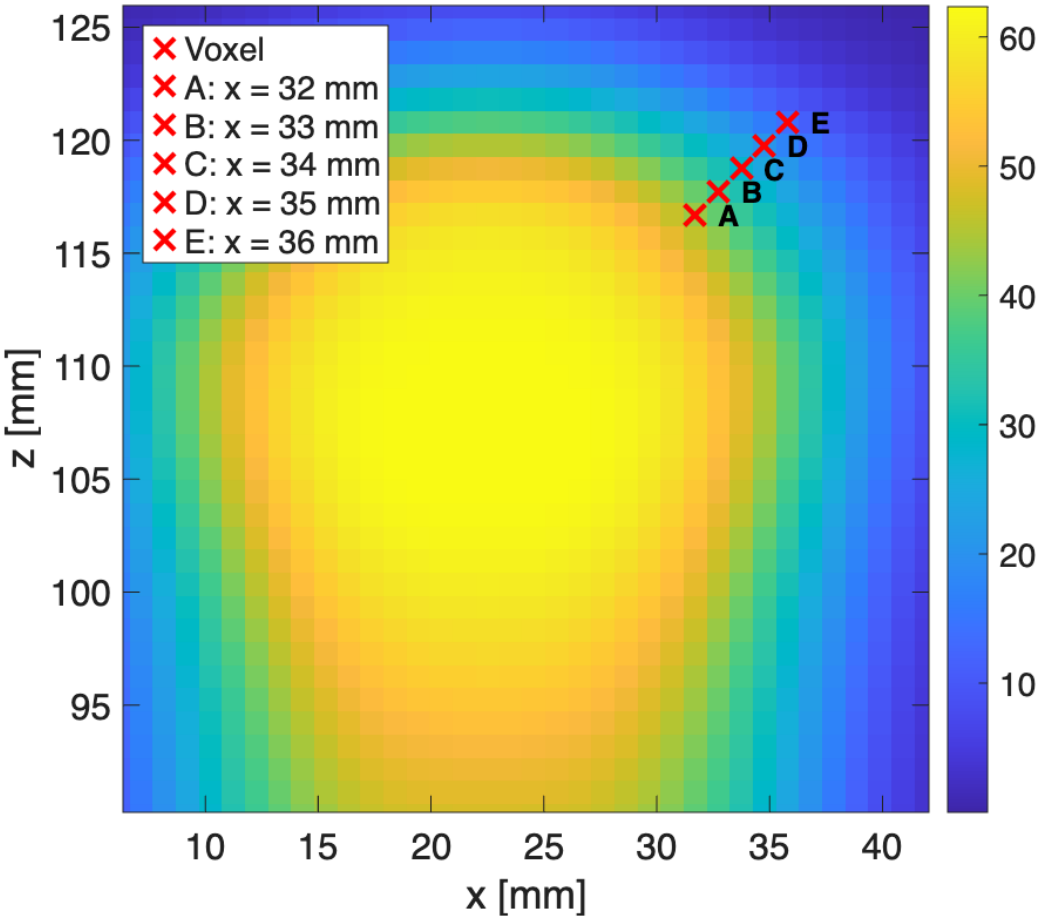


Figure 4.8: Slice of the phantom geometry with nominal dose distribution indicated by the legend on the right (in Gy). Markers A through E indicate voxels located near the edge of the planning target volume, selected to analyze the quality of the Gram-Charlier fit to their dose probability distributions.

A slice of the phantom geometry in Figure 4.8 shows the resulting nominal dose distribution of a treatment plan with a 1D setup error in the x-direction. It is clear that the simulated proton beams arrive in the positive z-direction, as can be seen by the higher dosage received in the bottom of the figure compared to the top, where the beam energies quickly drop off past their Bragg peaks. The voxels depicted in this figure will showcase setup uncertainty effects on both high dose voxels on the edge of the PTV (voxel A) and lower dose voxels outside the PTV (voxel E). The slice shown here is at $y = 34\,\text{mm}$, while each voxel is evaluated at equivalent x and y coordinates. For example, voxel A is evaluated at $x, y = 32\,\text{mm}$ in the phantom. The voxel locations plotted on their own slice are shown in Appendix A.1.

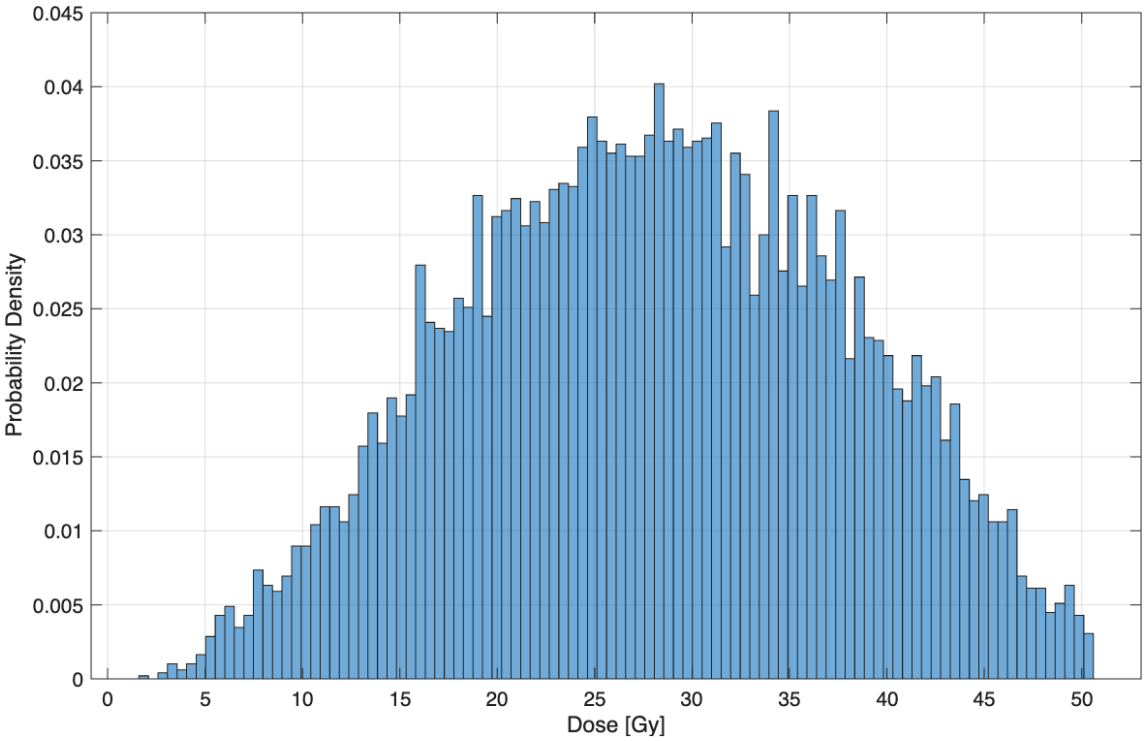


Figure 4.9: Histogram of dose distribution of voxel C with 10.000 samples. The Polynomial Chaos Expansion coefficients of this voxel are: [28.1076, 9.7793, -0.1392, -0.3447, -0.0047, 0.0081, 0.0006, -0.0001].

In Figure 4.9 we see a sampling of the PCE of voxel C, and we can immediately identify that the first PCE coefficient term coincides with the mean dose of the voxel. The sampled histogram looks similar to a normal distribution, with one important caveat: There were no samples above a little over 50 Gy. This is due to the nature of the phantom of Figure 4.8. A setup error in the x direction essentially shifts the entire 'heat map' of dosage along the x-axis. A maximum of 50 Gy means that anywhere along the x-axis for the fixed y and z coordinates of the chosen voxel, the dose does not exceed this maximum value. What is also clear from Figure 4.9 is that at 10.000 samples there is still plenty of noise present, of which some is still visible with 100.000 samples in Figure 4.10. Note that this initial raw sampling from the phantom was done with less samples to conserve time while analyzing many individual voxels.

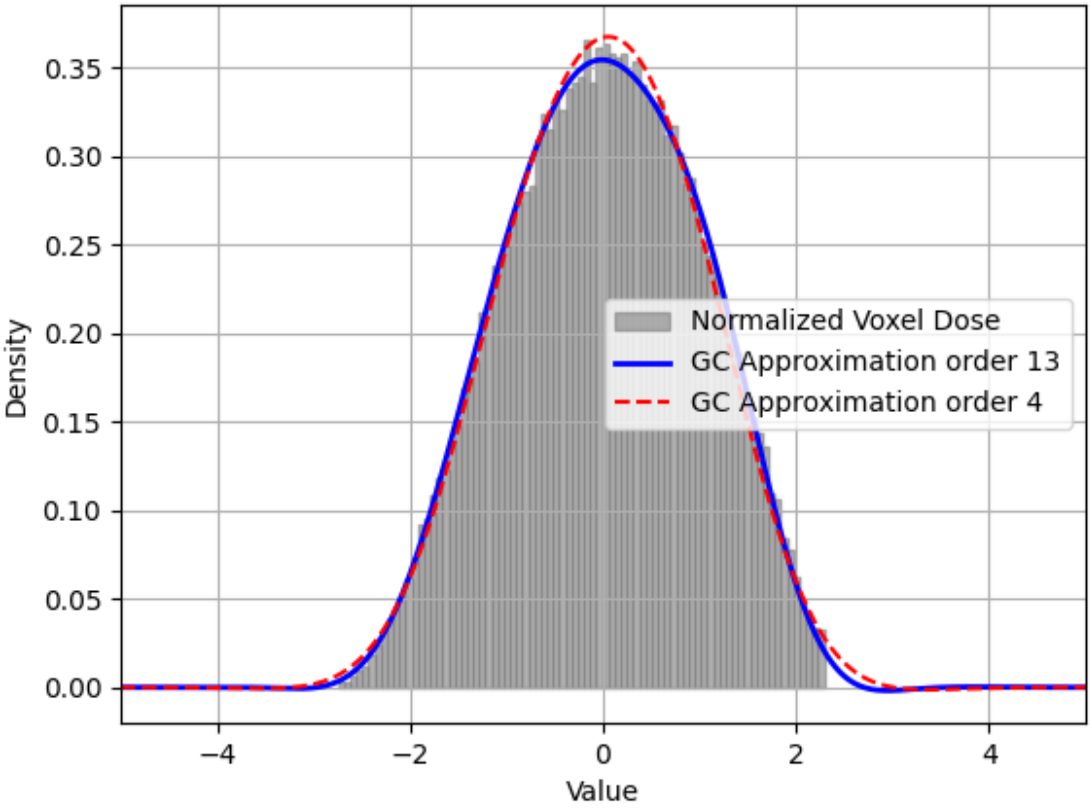


Figure 4.10: Standardized dose histogram of voxel C with 100.000 samples and fitted Gram-Charlier approximations.

After standardizing the PCE and fitting the GC approximation from its analytical moments, we can visually inspect the fit quality between a low and high order truncation. Figure 4.10 reaffirms that the fit is close to Gaussian, seen by the similarity between the 4th and 13th order GC approximations. The fit appears to be close to the sampled histogram, with the 13th order truncation modeling the left tail marginally better than the lower order. The cut-off for the right tail from the maximum dose of this voxel is still present, but now just at a standardized value.

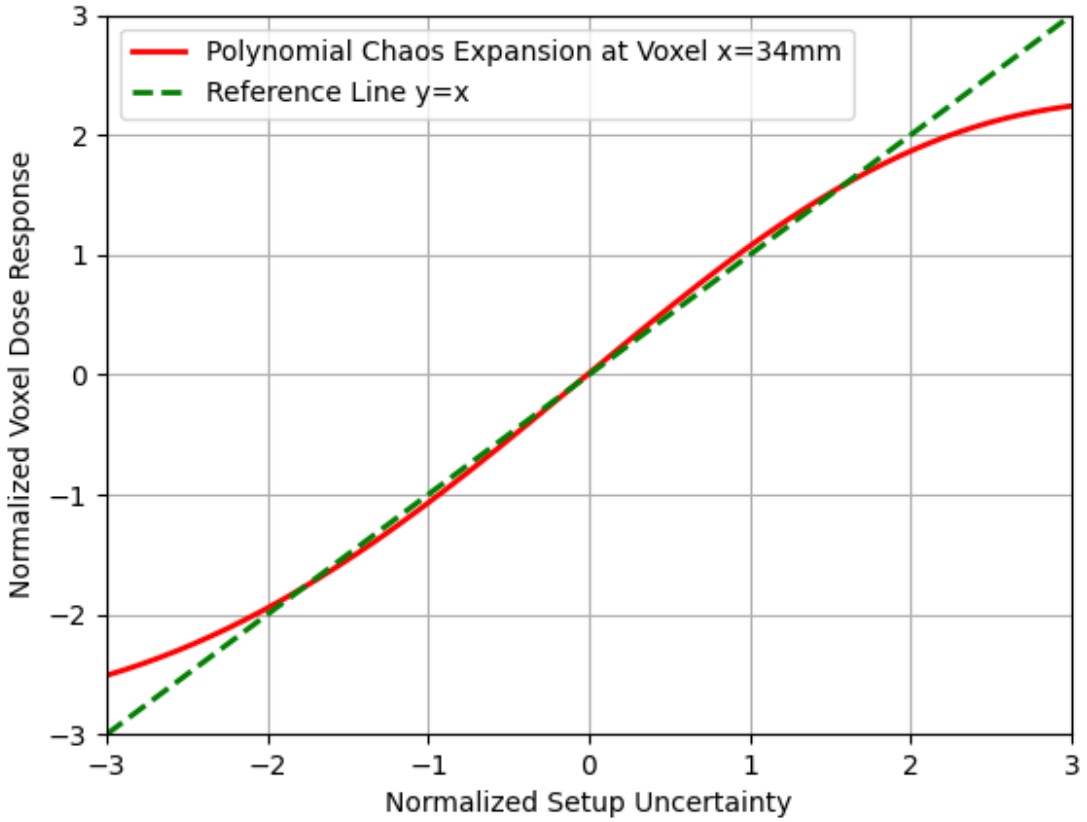


Figure 4.11: Standardized Polynomial Chaos Expansion of voxel C with a line corresponding to a perfectly Gaussian distribution as a reference. The standardized coefficients are: [0.0118, 1.1148, -0.0086, -0.0444, -0.0014, 0.0010, 0.0001, -0.0000].

Another way to verify Gaussianity of the polynomial random variable is by graphically examining the standardized PCE around 0. Figure 4.11 shows us that the PCE of voxel C is close to the perfectly linear response in green. In this standardized formulation, a line $Y = X$ would be the PCE of a standard Gaussian, which the GC series can exactly reproduce. The PCE of voxel C is close to Gaussian up to approximately ± 2.

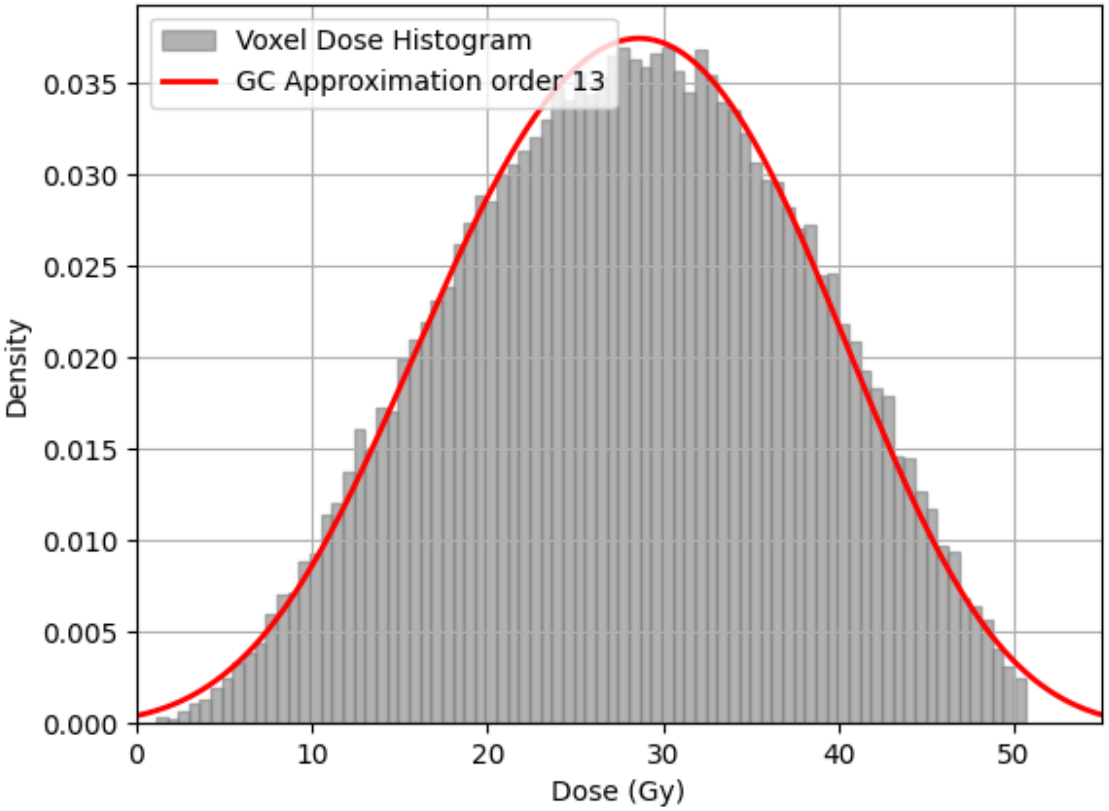
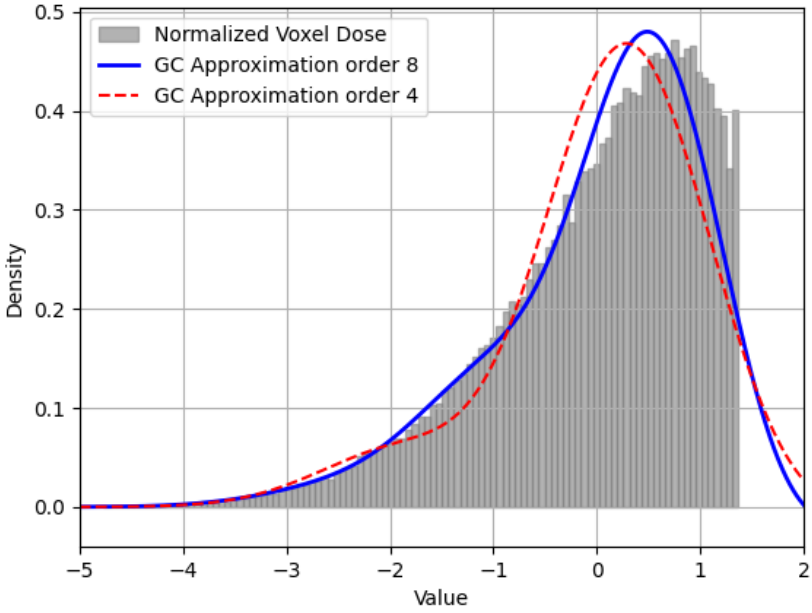
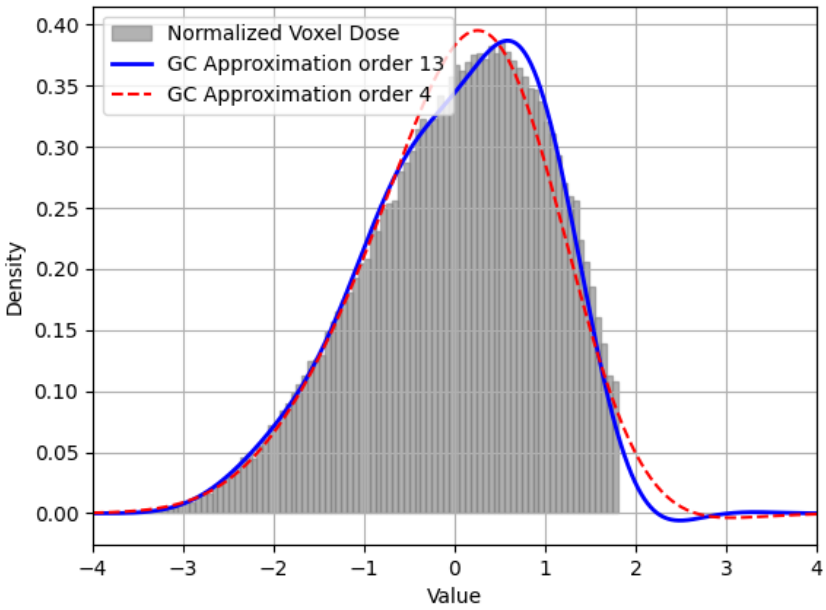


Figure 4.12: Histogram of dose distribution of voxel C with 100.000 samples and fitted Gram-Charlier approximation.

Once the GC fit quality is evaluated in a standardized form (for universal comparison), returning to approximation on the real dose distribution only requires replacing the standardized variable in the expression with its non-standard scaling (using the mean and standard deviation from already computed moments/cumulants). This is illustrated in Figure 4.12. In this figure, even with 100.000 samples, the random spread in the data is quite visible, which highlights the inherent problem with sampling as a PDF approximation method.



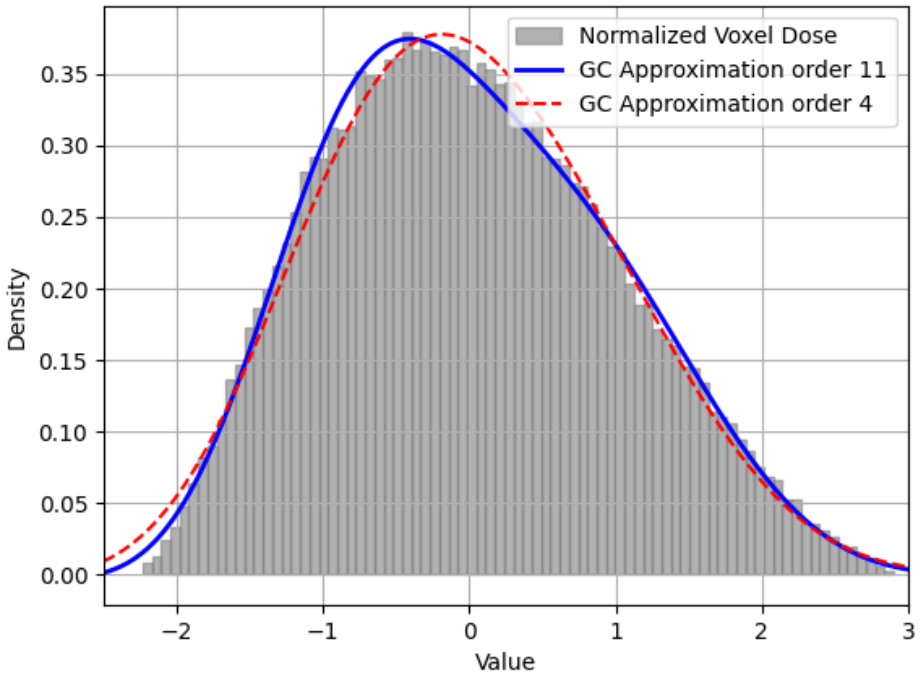
(a) Voxel A



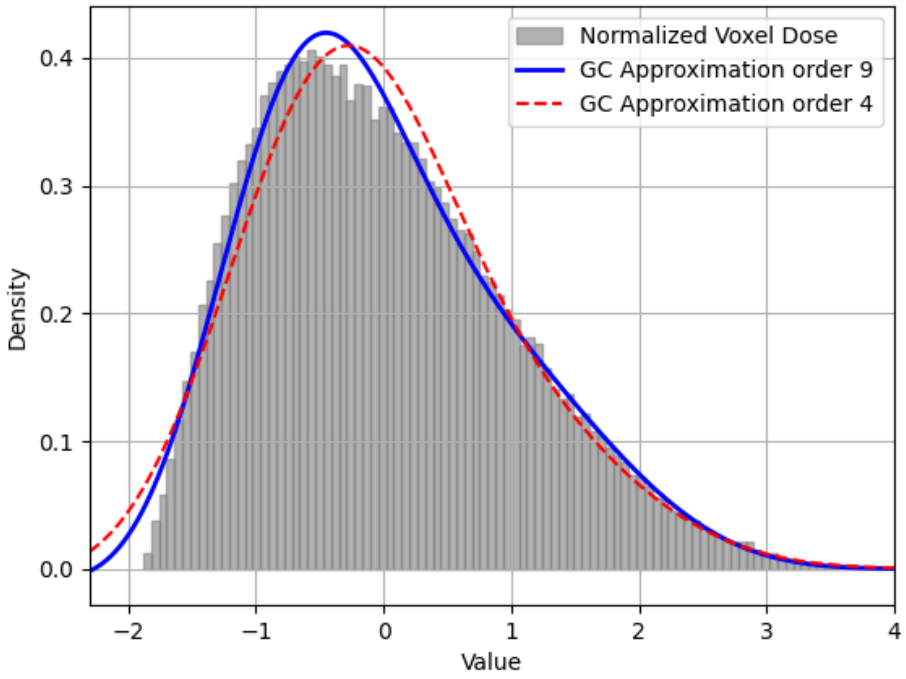
(b) Voxel B

Figure 4.13: Standardized dose histogram of voxels A and B with 100.000 samples and fitted Gram-Charlier approximations.

Figure 4.13 shows the standardized dose distribution with GC overlays for voxels A and B respectively. Voxels A and B are located inside the PTV, which is the reason they are skewed left; they are receiving their maximum dose with a high probability, and smaller doses with a lower probability. The left tails, representing the under-dose, show visibly promising alignment with the samples. The L^2 error (center alignment) breaks down with higher skewness, and the right tails do not approximate the sudden cut-offs accurately. Both (a) and (b) show a tendency of the GC approximation to become negative beyond the right cut-off, which is an unwanted property of GC series in highly non-Gaussian applications.



(a) Voxel D



(b) Voxel E

Figure 4.14: Standardized dose histogram of voxel D and E with 100.000 samples and fitted Gram-Charlier approximation

For voxels D and E in Figure 4.14, an opposite skewness is present to that of Figure 4.13. Voxel D is closer to the PTV edge, and thus yields a higher nominal dose; while voxel E receives a lower nominal dose and consequently has a larger right skewness. Similar to voxels A and B, the L^2 center error appears lowest for the more Gaussian (less skewed) voxel D, and it had an overall better fit quality. It is clear that visually the right tails for both voxel D and E are a good fit, and if these voxels represent those of an OAR, then the right tails are the most important feature, as they help quantify a near-maximum dose.

Voxel (Location)	Closest Order	L^2 Center Error	Left Tail Error	Right Tail Error
A (x = 32 mm)	8	0.1586	0.0013	0.1410
B (x = 33 mm)	13	0.0499	0.0014	0.0389
C (x = 34 mm)	13	0.0258	0.0017	0.0044
D (x = 35 mm)	11	0.0378	0.0090	0.0033
E (x = 36 mm)	9	0.0645	0.0264	0.0018

Table 4.2: Gram-Charlier fit quality for selected voxels: lowest-error truncation order and associated metrics from Section 3.1.1.

Table 4.2 shows that the overall fit quality appears to be better for the more Gaussian-looking voxels B,C, and D. The L^2 center error is very high for voxel A, and adequate (but still above the Table 3.2 ideal error) for voxel B. Though their left tail errors are also larger than the ideal Gaussian fit errors, visually they appear to approximate the sampled histogram very closely, and are almost equally accurate for both voxels. Likewise, the opposite right tail errors of the further out voxels D and E were smaller than their left errors, which corroborates the visual inspection from Figure 4.14. Again, the center error was better for voxel D, closer to the PTV edge. It should be noted that for each of the voxels, the order plotted and presented in Table 4.2 was the truncation with the best fit. The GC approximations on voxels that were further from a standard Gaussian diverged earlier, resulting in lower order approximations.

5 Discussion

5.1 Analysis of Gram-Charlier on Simulated Polynomials of Gaussians

This section will discuss the results presented in Section 4.1.

5.1.1 Implications

As Figures Figure 4.2—4.5 show, there is quite a large spread in the data, suggesting our chosen indicators do not have a straightforward predictable relationship to the L^2 error of our polynomial fit. Despite this, there are relevant and important implications to consider. First, of all the chosen indicators that were evaluated, no single one (including tests that were not presented in the results) could consistently be used to identify and predict for which polynomials the GC expansion would yield a precise fit. Likewise, the divergence behaviour of the GC series at higher orders was clear and apparent, which is a known characteristic of the series (Blinnikov & Moessner, 1998). Large higher order cumulants (embedded in the coefficients) also consistently resulted in high error and poor fit quality. These findings were mostly expected (Brenn & Anfinsen, 2017)(de Kock, 2009)(Capodaglio et al., 2021), and the machine learning model shows a promising solution to the first classification problem. If a classifier can be trained to effectively predict all good GC use cases of polynomials of Gaussians for a given threshold, then it has the potential to be a very useful tool in proton therapy treatment planning. This could eliminate sampling of many of the most relevant voxels in a phantom geometry, and offer more precise analytical methods for percentile calculations.

5.1.2 Limitations and Sources of Error

The greatest limitation of this study was time and consequently sample size. Only 961 different polynomials were tested, and only up to the third order Gaussian (X^3). To achieve a fully comprehensive understanding of the GC series behaviour with this specific application in the context of PCEs, the sample size should include higher order terms and many more coefficient values ($\alpha, \beta, ...$). Python's floating point errors may introduce some uncertainty too, though this was likely negligible as analytical moments were computed with precision of approximately 15 decimal places. Finally, the classifier model should be seen purely as a proof of concept. The model was built on

fewer than 14.000 samples, and tested on an even smaller data set. The test data was chosen to be different from the training data, but naturally similarities were present. Lastly, the data was biased, with a majority of samples falling under one class. This can heavily skew the precision and recall values of the classifier.

5.1.3 Possible Future Work

Future work should include larger sample sizes, with a complete set of coefficients representing the range of (almost) all possible combinations that arise from the PCEs of phantom geometries. It would be both relevant and interesting to realize a machine learning model trained on an extensive amount of data, and test its performance on general polynomials and PCEs of a treatment plan. Lastly, different expansions such as the Edgeworth series could be investigated as a substitute for GC. They may have overall better convergence properties, but may sacrifice some simplicity and interpretability.

5.2 Analysis of Gram-Charlier on Patient Data Dose-Statistics

This section will discuss the results presented in Section 4.2.

5.2.1 Implications

The results of the patient data dose-metrics suggest that some percentiles may follow Gaussian-like distributions, and can therefore be approximated with the GC series. In this case, as long as a PCE can be built for a given metric, it will be possible to implement GC as a substitute for sampling. This has the possibility of improving both modeling accuracy and computational efficiency if implemented effectively.

5.2.2 Limitations and Sources of Error

The patient data that was accessible in this research was lacking in both sample size and variety. Though error metrics were consistent with that of a Gaussian of 1,000 samples, this is insufficient for reliably building PDFs. This is because the GC approximations were built from empirical moments, and 1,000 samples is a very limited sample space to model tail behaviour and higher order deviations from Gaussianity. Likewise, 17 histograms is a relatively small data pool, and half of the metrics had to be omitted as

they only included 1 histogram per patient. The visible performance difference between fits on patient 1 versus patient 2 could have many different origins, and the limited access to treatment plan specifics limits our ability to pinpoint the error sources; whether that be patient related or GC related.

5.2.3 Possible Future Work

It would be valuable to extend the tests on patient dose-metrics to include both larger sample sizes and PCE based methods. Building a larger database of GC use cases (or possibly other PDF approximation methods) in proton therapy can only serve to better our understanding of modeling uncertainties in treatment planning.

5.3 Analysis of Gram-Charlier for Voxels in a Phantom Geometry

This section will discuss the results presented in Section 4.3.

5.3.1 Implications

Visually and statistically strong GC fits of voxel dose distributions around the PTV edge of a phantom geometry could imply real value for moment-based methods in this context. Despite the relevant tail error metrics being not quite as strong as an ideal case, the error was still small in terms of standard probability density units. The cause for the figures having such sharp cut-offs, rather than gradual changes, is because of the setup uncertainty chosen. σ was $\pm 3\,\text{mm}$, which means a significant percentage of the data will lie close to the maximum dose along the x-coordinates of the chosen voxel. Though this study was done on a simplified example of a phantom geometry with 1D uncertainty, good fit quality of the standardized PDFs implies applicability to real voxel dose distributions that may extend to 3D errors. All results in Section 4.3 show that even for voxels with a heavily skewed distribution, it is possible to approximate and extract relevant PDF features such as a tail or center mass.

5.3.2 Limitations and Sources of Error

The limitations of this study include: simplified setup uncertainty, a perfectly spherical CTV, and sub-optimal error metric for tail behaviour. For the sake of simplicity, a 1D setup error was chosen which acted only along

the x-axis of the phantom. In combination with a perfectly spherical CTV, this limits the scope of our conclusions to similar setups, and warrants more investigation for GC applied to the intricate 3D uncertainties of asymmetric CTVs. Lastly, this study focused on being able to predict GC use cases, and the tail mean absolute error per bin was a good indicator for low sample histograms that corroborated with visual verification methods. It was also chosen because it did not put too much weight on outliers, but for larger sample sizes it may be more effective and relevant to analyze tail mass error. Tail mass error of the standardized histograms could directly be converted into total dose error for analyzing how much dosage is missed by the approximation.

The main source of uncertainty in the analysis of voxel PCEs was the decimal place precision. The coefficients were precise up to only 4 decimal places, which for 7th order Gaussian terms is likely not enough to avoid early divergence in higher order cumulants.

5.3.3 Possible Future Work

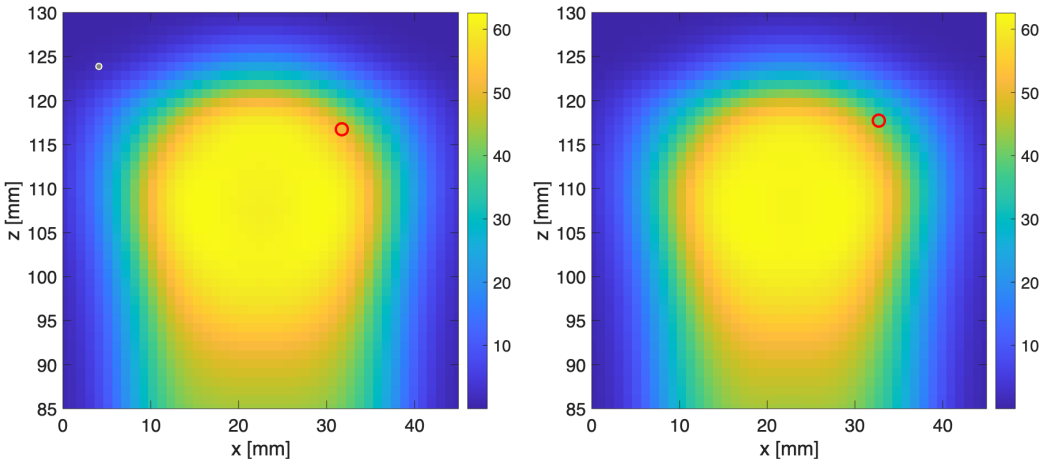
In order to improve on the results, the application of the GC series should be expanded to a larger variety of more complex phantom geometries and a wider range of voxels. The PCEs should be developed to an optimal precision while taking into account computational cost, and the tail mass error should be introduced and tested for very large sampled histograms ($>> 10^5$). For this extension it will be important define a cutoff for the GC expression (since it is unbounded), and program built-in safeties for when the expression becomes negative. Finally, the computational speed of the moment-based method should be tested against traditional sampling. If moment-based methods turn out to be more computationally intensive even when optimized, it will be necessary to evaluate the difference in dose error between traditional and novel techniques, and weigh their benefits.

6 Conclusion

The aim of this study was to better understand the applicability of the GC series in proton therapy, as accurate estimation of voxel-level dose PDFs is essential for uncertainty quantification. We evaluated the GC series using generated polynomials, dose metrics from patient data, and voxel-level dose PDFs in a phantom geometry. The study focused on identifying use cases for the GC series and to explore predictors of fit quality and divergence, based on both simulated and real-world data. The results of individual indicators across a range of simulated polynomials of Gaussians showed that no single metric (e.g., expansion coefficient sums, constant term values, or relative truncation errors) consistently predicted the quality of GC fits. However, a machine learning classifier that combined multiple metrics showed promise: the proof-of-concept model achieved high precision (95%) and recall (94%) in identifying good GC fits, highlighting its potential for aiding use case prediction in proton therapy treatment planning. Patient data shows selective GC applicability: the GC approximations remained within ideal bounds for certain metrics (e.g., OAR D_{mean} and $D_{40\%}$) at low truncation orders, particularly in one of the two patients. The voxel analysis in the phantom geometry with a 1D setup error supported visual and statistical validity. Even for skewed voxel-level dose PDFs near the edge of the PTV, the relevant tail approximation was close to the sampled histogram. When solving GC approximations directly from PCE coefficients for voxel-level dose PDFs, lower orders were typically more accurate for modeling tail behaviour of highly non-Gaussian PDFs. Optimal truncation order was between $8 - 13$ for the chosen voxels, after which the series approximation would quickly diverge.

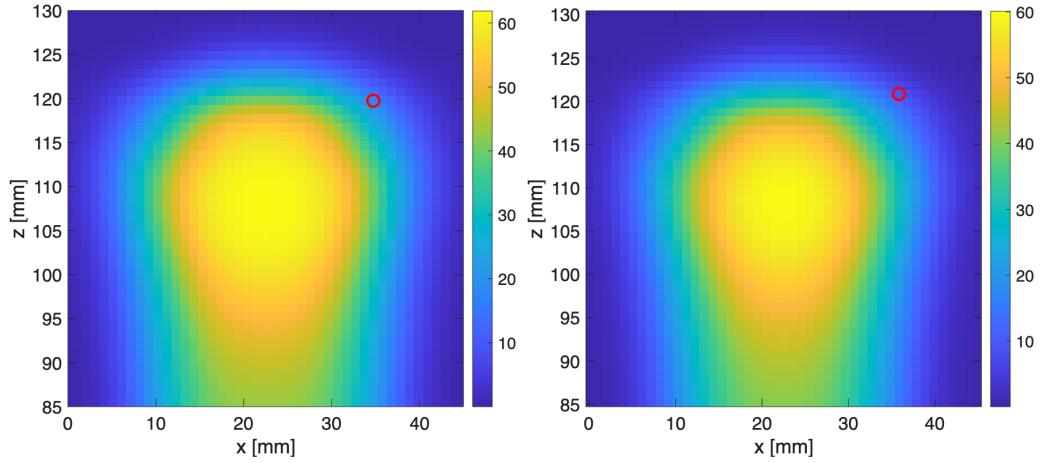
Naturally, the GC approximation did not always converge to the desired PDF, and sample sizes constrained the generalizability of the results. Divergence at higher truncation orders was frequent and expected, especially for PDFs with large higher-order cumulants. Furthermore, small sample sizes, limited patient data, simplified 1D setup uncertainties, and limited coefficient precision reduced the scope and reliability of the conclusions. The chosen tail error metrics and fit quality remained application-dependent, with the tail mean absolute error per bin a suitable metric for low sample sizes, but future work should consider tail mass error for high-fidelity PDF modeling of voxel dose. Other future work necessary to expand the results to clinical feasibility may include: expanding data sets, improving moment precision, testing 3D uncertainties, optimizing classifier features, and evaluating computational performance. These critical next steps may aid in establishing whether the GC series PDF approximation has real practical utility in proton therapy.

A Additional Figures



(a) Voxel A ($x, y = 32\,\mathrm{mm}$)

(b) Voxel B ($x, y = 33\,\mathrm{mm}$)



(c) Voxel C ($x, y = 35\,\mathrm{mm}$)

(d) Voxel D ($x, y = 36\,\mathrm{mm}$)

Figure A.1: The four remaining voxels from Figure 4.8 plotted on their own slice in the phantom geometry.

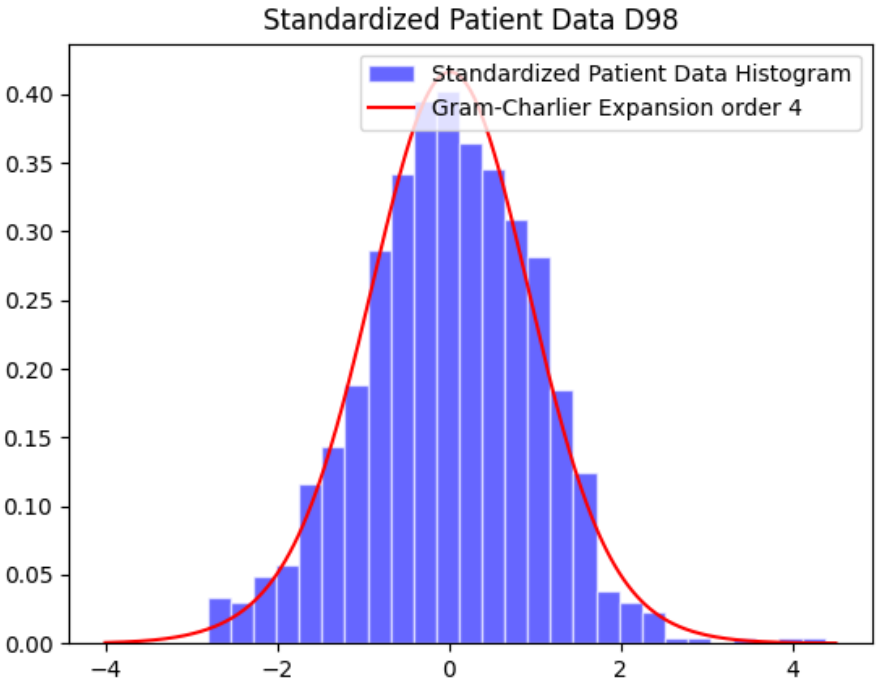


Figure A.2: Standardized $D_{98\%}$ Patient Data from Patient 1, with GC approximation fitted on the only $D_{98\%}$ histogram of 1.000 samples.

B Code Snippets

```
import numpy as np
from sympy import *
import math
from sympy import symbols, expand, Poly, Symbol
from sympy import symbols,bell,simplify
import sympy.polys.orthopolys
from math import comb

def expanded_polynomial_as_list(coeffs, k): #This expands the polynomial and returns an ordered list
    # of increasing power in the form: [(power, coefficient), (next power, coefficent)]
    X = symbols('X')
    Y = sum(a * X ** i for i, a in enumerate(coeffs))
    Yk = expand(Y ** k)
    poly = Poly(Yk, X)
    terms = [(exp[0], coeff) for exp, coeff in poly.terms()]
    return sorted(terms)
#This expands the polynomial and returns an ordered list
    # of increasing power in the form: [(power, coefficient), (next power, coefficent)]

def moment_of_polynomial(polynomial, mu, sigma):
    exp_Yk = 0
    for p in range(len(polynomial)):
        n = polynomial[p][0]
        coeff = polynomial[p][1]
        expected_value = 0
        for r in range(n + 1):
            if r % 2 == 0:  # only even moments are non-zero for Gaussian
                expected_value += (
                    math.factorial(n) * (sigma ** r) * (mu ** (n - r))
                    / ((2 ** (r / 2)) * math.factorial(n - r) * math.factorial(r // 2))
                )
        exp_Yk += expected_value * coeff
    return exp_Yk
```

```
34 #the moment of a polynomial of different order gaussians
       with mean mu and std sigma
35
36
37 def standardize_Z(Z,mu,sigma):
38     Z_squared= expanded_polynomial_as_list(Z,2)
39     Z_one= expanded_polynomial_as_list(Z,1)
40     Var_Z = moment_of_polynomial(Z_squared,mu,sigma)-
          moment_of_polynomial(Z_one,mu,sigma)**2
41     Mu_Z = moment_of_polynomial(Z_one,mu,sigma)
42     if np.isclose(Var_Z, 0):
43         print("Warning: Variance of the polynomial is
              effectively zero     standardization may fail
              .")
44     Y=Z.copy()
45     Y[0]=Y[0]-Mu_Z
46     Y = np.array(Y)/(np.sqrt(float(Var_Z))) #divide all
          elements by std of Z
47     return Y
48 #standardize the polynomial response (which is just a
      list of coefficients of ascending orders of gaussians
      )
49
50 def n_cumulants_from_moments(moments_of_Y): #function to
       make list of n cumulants from all moments of Y[]
51     t, n = symbols('t n') #initiate symbols
52     My = Function('My') #moment generating function
53     Ky = ln(My(t)) #cumulant generating function, ln of
          MGF as a function of t
54     all_cumulants = [] #initiate empty list to store all
           the cumulants
55     moments_dictionary = {Derivative(My(t), (t, k)):
          moments_of_Y[k] for k in range(len(moments_of_Y))
          } #a dictionary to convert the derivatives from
          Sympy into the equivalent moments of Y
56     for i in range(len(moments_of_Y)):
57         kn = Ky.diff(t, i) #nth cumulant
58         all_cumulants += [kn.subs(moments_dictionary)] #
              add nth cumulant to list of all cumulants
59     return all_cumulants
60 #function to make list of n cumulants from all moments
      of Y[]
61 import math
```

```
62
63 def cumulants_from_moments_fast(standardized_moments):
64     """
65     Compute cumulants from central moments for a
          standardized variable (mean=0, var=1).
66     Input:
67         central_moments: list or array of central
              moments [mu1=0, mu2=1, mu3, ..., mu_n]
68                          length n
69     Output:
70         cumulants: list of cumulants [k1=0, k2=1, k3,
              ..., k_n]
71     """
72     n = len(standardized_moments)
73     cumulants = [0, 1]  # first cumulant = mean=0,
          second cumulant = variance=1
74
75     for i in range(3, n + 1):
76         k = standardized_moments[i - 1]  # i-th central
              moment (index i-1)
77         for m in range(1, i):
78             k -= math.comb(i - 1, m - 1) * cumulants[m -
                  1] * standardized_moments[i - m - 1]
79         cumulants.append(k)
80     return cumulants
81
82
83 def full_bell_polynomial(n,ku): #up to order n of
       cumulant list ku = ()
84     bell_polynomial = sum(bell(n,k,ku[1:]) for k in
          range(1,n+1)) #Computes the complete Bell
          polynomial B_n from cumulants ku. Takes ku[0] = 0
           (ignored).
85     return bell_polynomial
86 #full bell polynomial up to order n of cumulant list ku
       = () each order n corresponds to calculating until
       the nth cumulant
87
88 def nth_hermite_polynomial(n,x): #nth order bell
       polynomial with variable x is output
89     Hen=sympy.polys.orthopolys.hermite_prob_poly(n,x)
90     return Hen
91
```

```
92
93  def momentslist(Y,k,mu,sigma):
94      moment_list = []
95      for r in range(k+1):
96          Y_k=expanded_polynomial_as_list(Y,r)
97          moment_list += [moment_of_polynomial(Y_k,mu,
                sigma)]
98      return(moment_list)
99
100 def empirical_moments(samples, k):
101     return [np.mean(samples ** i) for i in range(k + 1)]
102
103 def standardize_moments(raw_moments):
104     mu = raw_moments[1]
105     var = raw_moments[2] - mu**2
106     standardized = []
107
108     for n in range(len(raw_moments)):
109         moment = 0
110         for j in range(n + 1):
111             coeff = math.comb(n, j)
112             moment += coeff * ((-mu) ** (n - j)) *
                    raw_moments[j]
113         moment /= var ** (n / 2)
114         standardized.append(moment)
115     return standardized
116
117 def pdfgenerator(cumulants,order):
118     x = symbols("x")
119     sums = 0
120     fy=0
121     adjusted_cumulants = cumulants.copy()
122     adjusted_cumulants[2] = adjusted_cumulants[2]-1 #
        these are the lists of cumulant differences from
        the expansion
123     for i in range(3,order+1):
124         sums += full_bell_polynomial(i,
                adjusted_cumulants)*nth_hermite_polynomial(i,
                x)/(math.factorial(i))
125     fy=(1+sums)*sympy.exp(-(x**2)/2)/(math.sqrt(2*math.
        pi))
126     return(fy)
127
```

```
128 def pdf_components(cumulants, max_order):
129     x = symbols("x")
130     adjusted = cumulants.copy()
131     adjusted[2] = adjusted[2] - 1  # adjust variance
132 
133     hermite_terms = []
134     for i in range(3, max_order + 1):
135         coeff = full_bell_polynomial(i, adjusted) / math
                .factorial(i)
136         hermite = nth_hermite_polynomial(i, x)
137         hermite_terms.append(coeff * hermite)
138 
139     return hermite_terms  # List of correction terms
```

Listing B.1: Python functions for GC approximations

Bibliography

Blinnikov, S., & Moessner, R. (1998). Expansions for nearly gaussian distributions. *Astronomy and Astrophysics Supplement Series*, *130*(1), 193–205. https://doi.org/10.1051/aas:1998221

Brenn, T., & Anfinsen, S. N. (2017). A revisit of the gram–charlier and edgeworth series expansions [Preprint available via UiT Munin repository]. *Manuscript submitted at University of Tromsø*. https://munin.uit.no/handle/10037/11261

Capodaglio, G., Gunzburger, M., & Wynn, H. P. (2021). Approximation of probability density functions for pdes with random parameters using truncated series expansions. *Vietnam Journal of Mathematics*, *49*(3), 685–711. https://doi.org/10.1007/s10013-020-00465-5

Chou, W. S., Hsu, L. C., & Shiue, P. J. S. (2006). Application of faà di bruno's formula in characterization of inverse relations. *Journal of Computational and Applied Mathematics*, *190*, 151–169. https://doi.org/10.1016/j.cam.2004.12.041

de Jong, J. (2025). Personal communication [PhD Candidate at TU Delft, Faculty of Medical Physics and Technology].

de Jong, J. I. (2025). Personal communication [PhD Candidate at Erasmus Medical Center in Rotterdam].

de Kock, M. B. (2009, December). Gram–charlier and edgeworth expansions for non-gaussian correlations in femtoscopy [University of Stellenbosch]. https://indico.cern.ch/event/74919/contributions/1246262/attachments/1047650/1493312/Heavy_Ions_2009_Michiel_de_Kock_Edgeworth.pdf

DeGroot, M. H., & Schervish, M. J. (2012). *Probability and statistics* (4th) [See page 304]. Pearson. https://educationexclusive.com/upload/pdf/Probability%20and%20Statistics%20by%20Morris%20H.%20DeGroot%20and%20Mark%20J.%20Schervish.pdf

Encyclopedia Britannica Contributors. (2025). Multinomial theorem [Accessed 21 July 2025].

Foundation, P. S. (2024). Python language reference, version 3.12 [Accessed 21 July 2025].

Freedman, D., & Diaconis, P. (1981). On the histogram as a density estimator: L^2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, *57*(4), 453–476. https://doi.org/10.1007/BF01025868

Greener, B., McDonough, J. E., Farahani, R., Pankuch, M., Wu, H., Li, H.-M., Lin, L., Kooy, H. M., Flampouri, S., & Seco, J. (2022). Evaluation of anthropomorphic phantoms for geometric accuracy in proton

therapy. *Journal of Applied Clinical Medical Physics*, *23*(9), e13782. https://doi.org/10.1002/acm2.13782

IBM Corporation. (2024). Random forest [Accessed 21 July 2025].

Jakeman, J. D., Franzelin, F., Narayan, A., Eldred, M., & Pflüger, D. (2019). Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, *351*, 643–666. https://doi.org/10.1016/j.cma.2019.03.009

Jiu, L., & Shi, D.-Y. H. (2022). Moments and cumulants on identities for bernoulli and euler numbers. *Mathematical Reports*, *24*(4), 547–556. https://imar.ro/journals/Mathematical_Reports/Pdfs/2022/4/4.pdf

Kainz, W., Neufeld, E., Bolch, W. E., Graff, C. G., Kim, C. H., Kuster, N., Lloyd, B., Morrison, T., Segars, W. P., Yeom, Y. S., Zankl, M., Xu, X. G., & Tsui, B. M. W. (2019). Advances in computational human phantoms and their applications in biomedical engineering – a topical review. *IEEE Transactions on Radiation and Plasma Medical Sciences*, *3*(1), 1–23. https://doi.org/10.1109/TRPMS.2018.2883437

Kim, T., Kim, D. S., Kwon, J., et al. (2021). Some properties of degenerate complete and partial bell polynomials. *Advances in Difference Equations*, *2021*(1), 304. https://doi.org/10.1186/s13662-021-03460-3

Lee, S., Stewart, J., Lee, Y., Myrehaug, S., Sahgal, A., Ruschin, M., & Tseng, C. L. (2019). Improved dosimetric accuracy with semi-automatic contour propagation of organs-at-risk in glioblastoma patients undergoing chemoradiation. *Journal of Applied Clinical Medical Physics*, *20*(12), 45–53. https://doi.org/10.1002/acm2.12758

Marcus, K. J. (2025). Radiation therapy [Accessed 3 July 2025].

National Cancer Institute. (2020). Proton therapy: A safer option? [Accessed 3 July 2025].

Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables and stochastic processes* (4th) [See p. 148 for Gaussian moments]. McGraw-Hill.

Radiopaedia Contributors. (2024). Gray (si unit) [Accessed 21 July 2025].

Schaffner, B., & Pedroni, E. (1998). The precision of proton range calculations in proton radiotherapy treatment planning: Experimental verification of the relation between ct-hu and proton stopping power. *Physics in Medicine and Biology*, *43*(6), 1579–1592. https://doi.org/10.1088/0031-9155/43/6/016

The MathWorks, Inc. (2024). *Matlab version 9.14 (r2024a)*. The MathWorks, Inc. Natick, Massachusetts. https://www.mathworks.com/products/matlab.html

Unkelbach, J., Alber, M., Bangert, R., Bortfeld, T., Boye, D., Chan, T. C., Deistler, M., Fredriksson, A., Gorissen, B. L., Graeff, C., Liu, W., Shih, H. A., Toftegaard, J., Trovò, M., & Ziegenhein, P. (2018). Ro-

bust radiotherapy planning. *Physics in Medicine and Biology*, *63*(22), 22TR02. https://doi.org/10.1088/1361-6560/aae659

van Herk, M. (2004). Errors and margins in radiotherapy. *Seminars in Radiation Oncology*, *14*(1), 52–64. https://doi.org/10.1053/j.semradonc.2003.10.003

Weisstein, E. W. (2025). L^2-norm [From MathWorld – A Wolfram Web Resource. Accessed 21 July 2025].