

9. Analyse und Simulation

In diesem Kapitel sind die Details der Analyse von Laufzeitverteilungen und der Monte-Carlo Simulation zusammengefasst.

9.1. Vorverarbeitung von Laufzeitverteilungen

Bevor eine Laufzeitverteilung zur weiteren Analyse verwendet wurde, werden verschiedene Vorverarbeitungsschritte durchgeführt:

1. Durch den Dunkelstrom des Detektors und das restliche Umgebungslicht besitzt die Laufzeitverteilung einen zeitunabhängigen Untergrund. Dieser kann entfernt werden. Dazu wird in einem zeitlichen Bereich vor der Laufzeitverteilung ein mittleres Untergrundsignal bestimmt und dieses von der Laufzeitverteilung abgezogen.
2. Die Gesamtzahl der Photonen im Referenzkanal ändert sich während der Messung durch Driften der Laser und des Detektors. Es kann für jeden Referenzpuls ein Faktor bestimmt werden, mit dem dieser multipliziert wird, so dass die Gesamtphotonenzahl im Referenzpuls während der Messung konstant ist. Derselbe Faktor kann auch auf die Laufzeitverteilungen angewendet werden, um diese von den apparativ bedingten Schwankungen der totalen Photonenzahl zu befreien.
3. Mit dem Referenzkanal können Schwankungen des Zeitnullpunktes bestimmt und korrigiert werden. Hierzu wird die zeitliche Verschiebung jedes Referenzpulses ermittelt. Diese Verschiebung kann z.B. über das erste Moment des Referenzpulses quantifiziert werden ¹. Die Zeitach-

¹Eine andere Variante ist in [1] beschrieben.

9. Analyse und Simulation

se des Referenzpulses und der Laufzeitverteilung wird über numerische Interpolationsroutinen so verschoben, dass das erste Moment des Referenzpulses sich nicht ändert. Somit wird die Laufzeitverteilung von apparativen Zeitnullpunktsschwankungen bereinigt.

9.2. Anpassen von Laufzeitverteilungen

Um die optischen Eigenschaften des homogenen semi-infiniten Mediums zu bestimmen, wird die analytische Beschreibung der Messgrößen j (2.24) mit der Apparatefunktion gefaltet und an die gemessene Laufzeitverteilung $N_{mess,k}$ angepasst. Hierzu wird

$$\chi^2 = \frac{1}{k_r - k_l + 1 - f} \sum_{k=k_l}^{k_r} \frac{(N_{mess,k} - A j_{theo,k})^2}{\sigma_k^2} \quad (9.1)$$

minimiert. Die Grenzen, in denen die Laufzeitverteilung ausgewertet wird, sind durch k_r und k_l gegeben. Die Fitparameter sind die optischen Eigenschaften μ'_s , μ_a und ein zeitlicher Offset t_0 mit $t \rightarrow t - t_0$. Die Anzahl der Fitparameter ist somit $f = 3$. Da keine absoluten Photonflussdichten vorliegen, wird ein Amplitudenfaktor

$$A = \frac{\sum_{k=k_l}^{k_r} N_{mess,k}}{\sum_{k=k_l}^{k_r} j_{theo,k}} \quad (9.2)$$

eingeführt, der während der Fitroutine immer wieder optimiert wird. Die Fitroutine wurde in IDL² umgesetzt und nutzt die Powell-Methode [2].

9.3. Entfaltung von Laufzeitverteilungen

Um eine möglichst korrekte zeitabhängige Attenuation zu erhalten, müssen gemessene Laufzeitverteilungen von der Apparatefunktion 'bereinigt' werden. Wie in Abschnitt 2.3 beschrieben, sind die gemessenen Laufzeitverteilungen N_{mess} mit den interessierenden Laufzeitverteilungen N über eine Faltung mit der zeitlichen Apparatefunktion g_{resp} des Systems

$$N_{mess}(t) = N(t) \times g_{resp}(t) \quad (9.3)$$

²Research Systems Inc., Boulder, CO, USA

9.3. Entfaltung von Laufzeitverteilungen

verknüpft.

Als Entfaltung sei hier jedes Verfahren definiert, mit dem eine Schätzung von $N(t)$ aus $N_{mess}(t)$ abgeleitet werden kann. Zwei verbreitete Algorithmen, nämlich der Van Crittens Algorithmus und ein Algorithmus im Frequenzraum, wurden für die Entfaltung von gemessenen Laufzeitverteilungen getestet [2]. Diese haben für die hiesige Anwendung zwei entscheidende Nachteile: Erstens wird das Rauschen in den Kanälen der Laufzeitverteilung verändert. Zweitens müssen die Algorithmen für jede Laufzeitverteilung einer Messreihe angewendet werden und sind dementsprechend rechenintensiv.

Aus dieser Problemstellung wurde für diese Arbeit ein einfacher Algorithmus entwickelt, der diese Nachteile umgeht. Durch die Faltung mit der Apparatfunktion ist die interessierende Laufzeitverteilung verbreitert. Die prinzipielle Idee ist, dass diese Verbreiterung durch ein nichtlineares Umskalieren der Zeitachse ($t_{skal} \rightarrow t$) beseitigt werden kann. Die neue, gestauchte Zeitachse kann geschrieben werden als

$$t_{skal} = f_{skal}(t). \quad (9.4)$$

Diese Funktion hat zwei Eigenschaften. Erstens ist sie monoton, damit die Skalierung eindeutig ist. Zweitens ist sie so zu wählen dass, mit

$$N_{conv}(t) = N_{mess}(t_{skal}(t)) \times g_{resp}(t) \quad (9.5)$$

$$\chi^2 = \sum_{k=k_l}^{k_r} \frac{(N_{mess,k} - N_{conv,k})^2}{N_{mes}(t_k)} \quad (9.6)$$

minimiert wird. Die Funktion f_{skal} kann somit über eine nichtlineare Fitroutine ermittelt werden. Verwendet wurde sowohl ein Polynom dritten Grades

$$t_{skal} = P(0) + P(1)t + P(2)t^2 + P(3)t^3 \quad (9.7)$$

als auch

$$t_{skal} = P(0) + P(1)t + P(2)(1 - \exp(P(3)t)). \quad (9.8)$$

Von diesen beiden Funktionen wurde jeweils die verwendet, die zu einem kleineren χ^2 führte. Abbildung 9.1 zeigt das Ergebnis einer solchen Entfaltung.

9. Analyse und Simulation

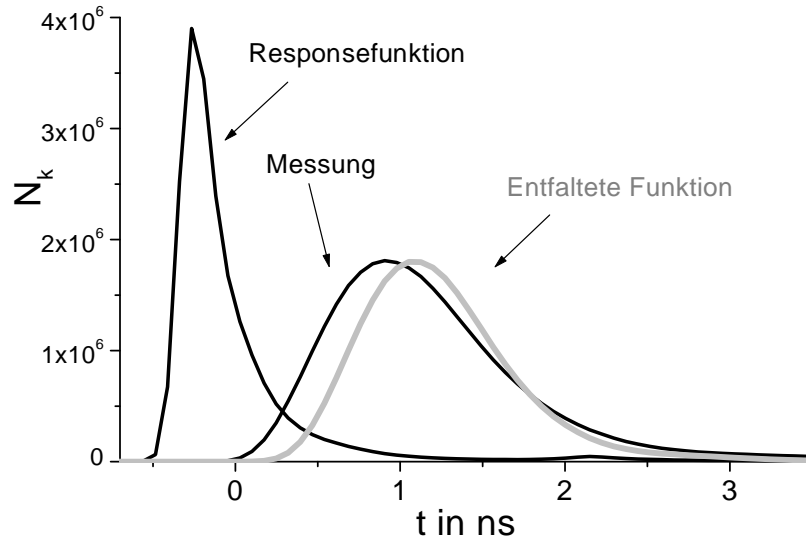


Abb. 9.1.: Eine mit dem klinischen Messplatz gemessene Laufzeitverteilung und die zugehörige zeitlichen Apparatefunktion. Die durch ein Skalieren der Zeitachse entfaltete Laufzeitverteilung (grau) ist ebenfalls dargestellt.

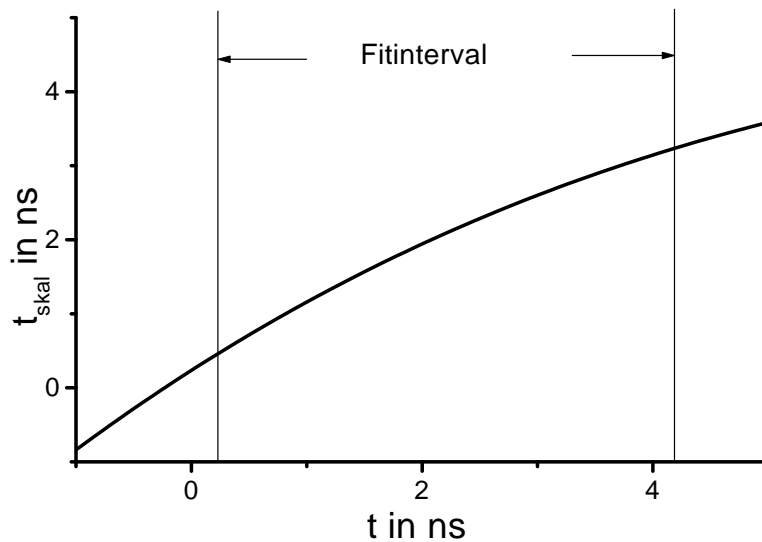


Abb. 9.2.: Die Skalierungsfunktion für die Entfaltung der Laufzeitverteilung in Abb. 9.1.

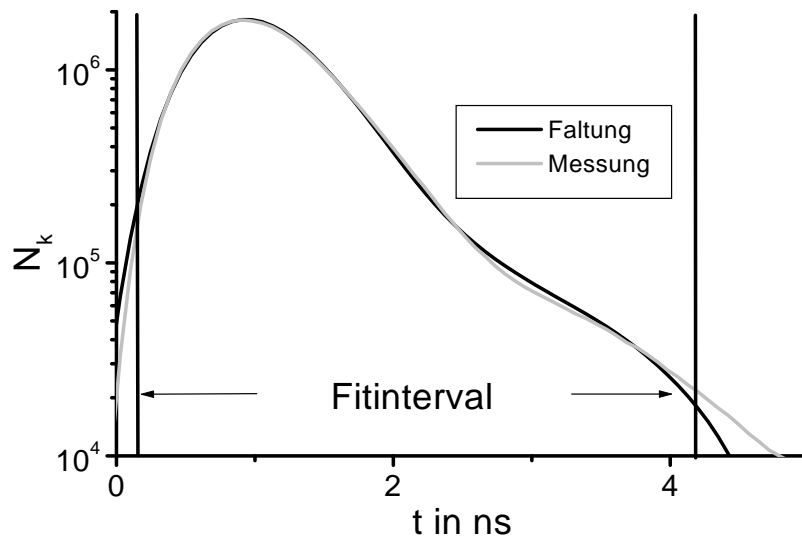


Abb. 9.3.: Die Qualität des Entfaltungsprozesses kann am besten gezeigt werden, indem die gemessene Laufzeitverteilung mit der gestauchten und gefalteten Laufzeitverteilung verglichen wird.

Die entfaltete Laufzeitverteilung ist schmaler als die gemessene Laufzeitverteilung. Die zugehörige Skalierungsfunktion ist in Abbildung 9.2 dargestellt. Die Qualität der Entfaltung ist durch einen Vergleich der gemessenen Laufzeitverteilung N_{mes} mit $N(t_{skal}) \times g_{resp}(t)$ möglich (Abbildung 9.3).

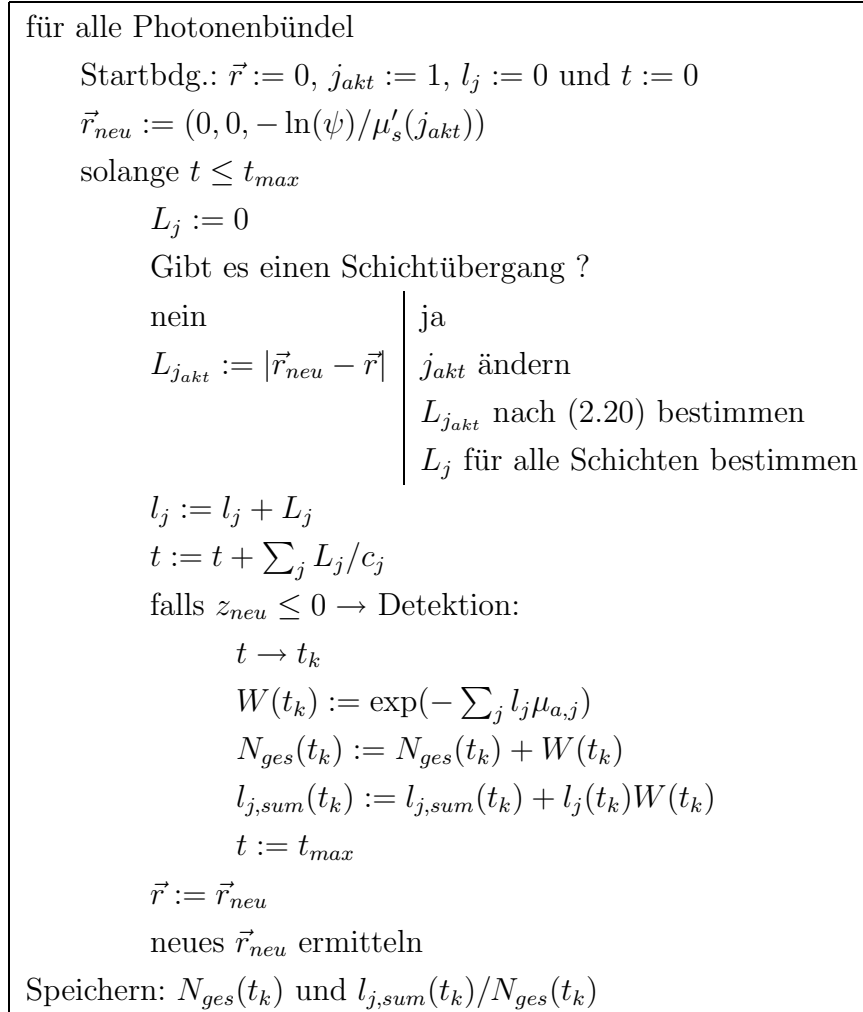
Für Echtzeit-Routinen ließe sich diese Entfaltungsmethode gut anwenden, da die Skalierungsfunktion in einer anfänglichen Kalibrationsphase ermittelt werden kann.

9.4. Der Code der Monte-Carlo Simulation

Für diese Arbeit wurde eine Varianz-Reduktions-Monte-Carlo Simulation entwickelt. In Abschnitt 2.4 wurden die grundlegenden Prinzipien einer solchen Simulation bereits beschrieben. Hier folgt die algorithmische Umsetzung und ein Funktionstest.

9.4.1. Algorithmus

In Tabelle 9.1 ist in einem stark vereinfachten Flussdiagramm der MC-Algorithmus dargestellt. Das streuende Medium wird über die Dicke, den Brechungsindex und die optischen Eigenschaften seiner Schichtkompartimente beschrieben. In der Simulation werden nacheinander eine vorgegebene



Tab. 9.1.: Ein Diagramm zum Code für die MC-Simulation. Ist der Index j angegeben ist immer 'für alle $j \in [1, j_{max}]$ ' gemeint.

Anzahl von Photonenbündeln in das Medium am Ort $\vec{r} = (0, 0, 0)$ injiziert. Die z -Achse des raumfesten Koordinatensystems zeigt senkrecht in das Medium. Für jedes Photonenbündel wird ein erster Streuort mit $\vec{r}_{neu} = (0, 0, -\log(\psi)/\mu'_s(j_{akt}))$ berechnet, wobei ψ eine Zufallszahl zwischen Null und

9.4. Der Code der Monte-Carlo Simulation

Eins ist. Die freie Weglänge bis zum Streueignis ist $L_{j_{akt}} = |\vec{r}_{neu} - \vec{r}|$, wobei j_{akt} das Kompartiment bezeichnet, in dem sich das Photonbündel gerade befindet. Für die weiteren Streueignisse wird jeweils \vec{r}_{neu} nach (2.13), (2.14) und (2.15) berechnet. Finden zwischen zwei Streueignissen ein oder mehrere Übergänge in andere Kompartimente statt, wird j_{akt} geändert. Weiterhin wird die Pfadlänge in den anderen Kompartimenten nach (2.20) korrigiert. Nach jedem Streueignis wird die bisher in den verschiedenen Kompartimenten zurückgelegte Weglänge mit $l_j = l_j + L_j$ für alle $j \in [1, j_{max}]$ berechnet. Zusätzlich wird die bereits verbrachte Flugzeit mit $t = t + \sum_j l_j / c_j$ berechnet. Wenn das Photon das Medium verlässt ($z_{neu} < 0$) wird es detektiert, d.h. seine radiale Position wird einem Detektor und seine Laufzeit t einem Zeitkanal k zugeordnet. Der Beitrag des Photonenbündels zur Laufzeitverteilung wird mit $W = \exp(-\sum_j \mu_{a,j} l_j)$ berechnet. Dieser Beitrag wird zu der bereits im k -ten Zeitkanal vorhandenen Photonenzahl addiert ($N(t_k) = N(t_k) + W(t_k)$). Die Summe der noch nicht normierten gewichteten Pfadlängen $l_{j,sum}(t_k) = l_{j,sum}(t_k) + l_j(t_k)W(t_k)$ wird ebenfalls berechnet. Falls das Photonbündel nach einer vorgegebene Zeit t_{max} den Detektor nicht erreicht hat, wird kein weiteres Streueignis bestimmt, sondern wie nach der Detektion ein neues Photonenbündel gestartet. Nachdem alle Photonenbündel abgearbeitet sind, werden $N(t_k)$ und $l_{j,sum}(t_k)/N(t_k)$ für verschiedene Optodenabstände gespeichert. Somit liegen die Laufzeitverteilungen und die zeitabhängigen mittlere Pfadlängen für mehrere Optodenabstände vor.

Die Simulationen liefen während der Arbeit auf einem DEC alpha Cluster. Der Quelltext ist in C geschrieben. Eine DOS-Version wurde ebenfalls entwickelt.

Die **Eingabe** erfolgt über eine Eingabedatei, die in Tabelle 9.2 dargestellt ist. In den ersten Zeilen der Eingabedatei werden die Namen für die Ausgabedateien angegeben. Die erste Datei beinhaltet die Laufzeitverteilungen und die zweite die Laufzeitverteilungen die sich ergeben, wenn das Medium nicht absorbiert ist. Für alle Schichten des Mediums befindet sich in der dritten Datei die mittleren zeitabhängigen partiellen Pfadlängen und in der vierten die zeitabhängigen $\langle l_j l_m \rangle$. Die nächsten drei Zahlen geben an, wie viel Photonenbündel registriert werden bis: 1. die Simulation beendet wird, 2. die vier Ausgabedateien abgelegt und 3. eine Ausgabe auf dem Bildschirm oder

9. Analyse und Simulation

```
/*Filenames
LZVMIT LZV.asc
LZVOHNE LZV0.asc
LJLENGTH lj.asc
LJSQR ljsqr.asc
/*Number of registered photons
/*Nrofmaxregphotons, how many bundles to save, howmany bundles to print to
screen
photonbundles 4e6 4e5 60000
/*Number of timewindows is 150, sampling window :
SAMPLINGWINDOW 25e-12
/*Number of radial positions, Width of rings in mm, position of first ring
in mm
RHODEF 19 3. 5.
/*Geometry
/*Number of layers regarded
NROFLAY 1
/* depth of the layers always declare 10 !
SCHICHTTIEFE 0 30 200 300 400 500 600 700 800 900 1000
/* Optical Parameters in 1/mm
MUSSTR 0.0 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
MUA 0.00 0.0356 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
REFRACTIVE 1 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4
```

Tab. 9.2.: Die Eingabedatei für die PC-Variante der Monte-Carlo Simulation.

in eine Logdatei erfolgt. 'Samplingwindow' ist die Kanalbreite der Laufzeitverteilung in Sekunden. 'Rhodef' beschreibt die Detektorgeometrie. Die erste Zahl steht für die Anzahl der Detektionsringe (in mm), die zweite für die Breite der Detektionsringe, und die dritte beschreibt die radiale Position des Detektionsringes mit dem kürzesten Quell-Detektorabstand. 'Nroflayer' gibt die Zahl der verwendeten Schichten an. Diese werden in den folgenden Zeilen bezüglich ihrer Tiefe, ihres Streukoeffizienten, ihres Absorptionskoeffizienten und des Brechungsindex beschrieben.

9.4.2. Test der MC-Simulation

Um die Simulation zu testen, wurde ein homogenes semi-infinites Medium simuliert und die berechnete Laufzeitverteilung mit der Lösung der Diffusionsgleichung (2.24) verglichen. Abbildung 9.4 zeigt die Laufzeitverteilung berechnet mit dem homogenen Modell aus Kapitel 4 ($\mu_a = 0.01 \text{ mm}^{-1}$, $\mu'_s = 1.0 \text{ mm}^{-1}$ und $\rho = 30 \text{ mm}$). Diese wurde mit der Lösung der Diffusionsgleichung (2.24) gefittet. Die Minimierung von χ^2 lieferte $t_0 = 4 \text{ ps}$, $\mu_a = 0.0097 \text{ mm}^{-1}$, $\mu'_s = 1.01 \text{ mm}^{-1}$. Dies ist in hinreichend guter Übereinstimmung mit der durchs Modell simulierten Verteilung. Im Gegensatz zum Fit einer gemessenen Laufzeitverteilung sind die gewichteten Residuen nicht gleichverteilt. Dies liegt an der nicht quantifizierbaren Standardabweichung der

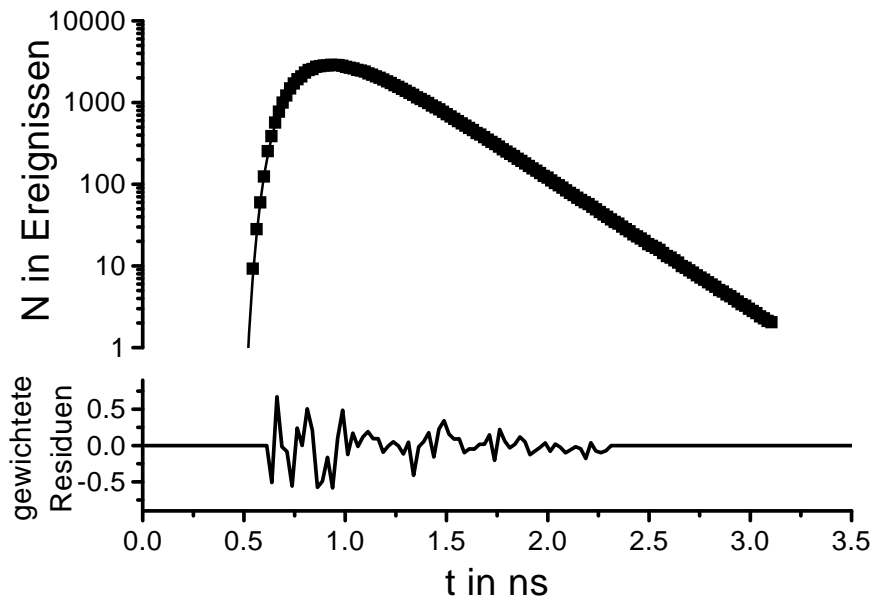


Abb. 9.4.: Eine mit MC simulierte Laufzeitverteilung ($\mu_a = 0.01$, mm^{-1} , $\mu'_s = 1.0 \text{ mm}^{-1}$, $\rho = 30$ bis 33 mm , quadratische Datenpunkte) und der Fit mit der Lösung der Diffusionsgleichung für ein homogenes semi-infinitytes Medium ($\rho = 31.5 \text{ mm}$, Linie).

Photonenzahl in den Zeitkanälen, der mit der VRMC-Simulation berechneten Laufzeitverteilung.

9.4.3. Rauschen der MC-Simulation

Da ein stochastischer Prozess die Grundlage der MC-Simulation ist, rauschen die berechneten Größen. Mit (2.5) kann vorhergesagt werden, wie viele Photonen notwendig sind, um die Standardabweichung der Photonenzahl in den Kanälen der Laufzeitverteilung unter einem bestimmten Wert zu halten. Es gibt bisher keinen Formalismus, mit dem die Standardabweichung in der Laufzeitverteilung für die verwendete VRMC Simulation abgeschätzt werden kann.

Für die Frage der korrekten Funktionsweise der Simulation ist ein Überprüfen der Statistik notwendig. Hierfür wurden zwei Laufzeitverteilungen für einen Optodenabstand von 30 mm für dasselbe homogen Medium berechnet. Der Absorptionskoeffizient wurde auf Null gesetzt. Da nun die Überlebens-

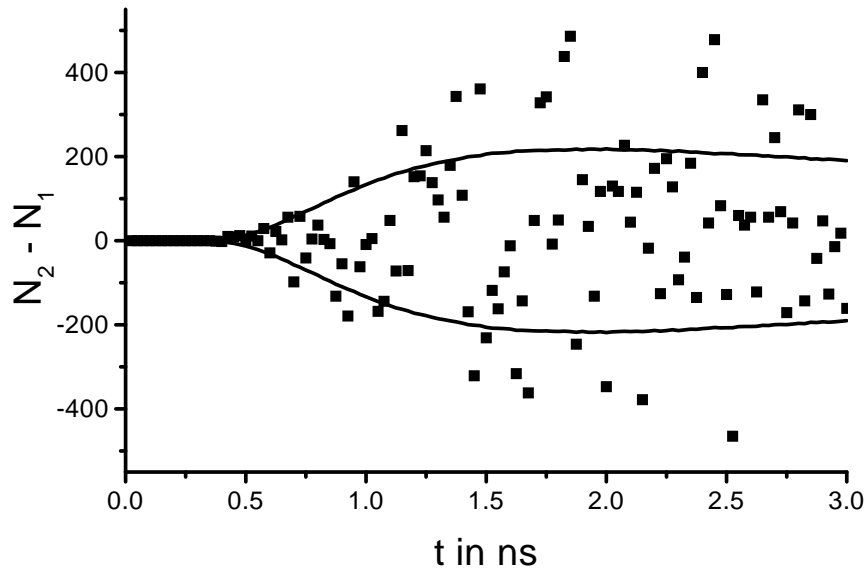


Abb. 9.5.: Die Differenz zweier simulierter Laufzeitverteilungen (Quadrate), die für das selbe Medium berechnet wurden. 66 % der Wert liegen indem Intervall $[-\sqrt{2}\sqrt{N(t_k)}, \sqrt{2}\sqrt{N(t_k)}]$ (Linie).

wahrscheinlichkeit W der Photonen immer eins beträgt, sollte das Photonenrauschen der Laufzeitverteilungen der Poissonstatistik gehorchen. Wichtig ist hierbei die Korrelationslänge der Zahlenkette die der Zufallsgenerator erzeugt. Hier wurde ein Zufallsgenerator genommen, der eine große Korrelationslänge besitzt [3, 4].

In Abbildung 9.5 ist die Differenz dieser beiden Laufzeitverteilungen und das Intervall $[-\sqrt{2}\sqrt{N(t_k)}, \sqrt{2}\sqrt{N(t_k)}]$ dargestellt. Da ca. 66 % der Werte in diesem Intervall liegen (88 innerhalb, 45 außerhalb), kann man schließen, dass die berechneten Laufzeitverteilungen der Poissonstatistik gehorchen.

9.5. Literatur

- [1] H. Danlewski. *Physikalische Grundlagen der zeitaufgelösten optischen Mammographie, einschliesslich Anwendungen*. Dissertation, Freie Universität Berlin, 1998.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Nu-*

merical Recipes in C. Cambridge University Press, 1994.

- [3] F. James. A review of pseudorandom number generators. *Comp. Phys. Comm.*, **60**:329–344, 1990.
- [4] G. Marsaglia und A. Zaman. *Towards a Universal Random Number Generator*. Florida State University FSU-SCRI-87-50, 1987.

