Improved Numerical Integration for Analytical Photon Distribution Calculation in SPECT

by

Thomas Humphries

BMath, University of Waterloo, 2005

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the Department
of
Mathematics

© Thomas Humphries 2007
SIMON FRASER UNIVERSITY
Summer 2007

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without permission of the author, except for scholarly or other non-commercial use for which no further copyright permission need be requested.

APPROVAL

Name: Thomas Humphries

Degree: Master of Science

Title of Thesis: Improved Numerical Integration for Analytical Photon Distribution Calculation in SPECT

Examining Committee: Dr. J.F. Williams
Chair

Dr. Manfred Trummer
Senior Supervisor

Dr. Anna Celler
Senior Supervisor

Dr. Steven Ruuth
Internal/External Examiner

Date Approved: _____

Abstract

In this thesis we study an existing Analytic Photon Distribution (APD) method for SPECT imaging. The method uses multi-dimensional numerical integration to calculate photon propagation and detection probabilities for a SPECT camera system. This calculation is extremely time-consuming in the original implementation of the method. Fast computation of these probabilities is especially challenging for scattered photons, due to the large number of possible photon paths and the high dimension of the integration problem. The choice of integration method, however, can significantly improve the speed of the calculation. Replacing Romberg integration, which was used in the original implementation, with a Gaussian quadrature method allows us to greatly reduce the calculation time. The photon distributions generated using the improved quadrature method are in good agreement with those generated by the original program.

Keywords: Numerical Integration; Medical Imaging; SPECT; Gaussian Quadrature; Romberg's Method

Subject Terms: Numerical Integration; Tomography, Emission; Diagnostic Imaging

Acknowledgments

First and foremost, I would like to thank my two supervisors, Dr. Manfred Trummer and Dr. Anna Celler, for introducing me to the exciting area of research that is medical imaging. The field is diverse and challenging, and their expertise in the many aspects of medical imaging has been of great help to me. I would also like to thank Dr. Steve Ruuth for serving as my External Examiner, and Dr. J.F. Williams for chairing my committee, as well as the rest of the SFU Applied Mathematics faculty for providing a hospitable and challenging learning environment.

Special thanks to Dr. Sergey Shcherbinin of the Medical Imaging Research Group for the initial suggestions for improving the numerical quadrature in APD, to Eric Vandervoort for the use of several of his figures and his very patient and helpful advice when I was first getting my feet wet, to Dr. Jim Verner for reviewing my initial draft, and to Dr. Walter Gander for the use of his quadrature code in Section 2.1.

I would also like to thank the National Sciences and Engineering Research Council (NSERC) for supporting me in the second year of my MSc. program with a PGS-M scholarship, as well as Simon Fraser University for also supporting my education financially.

Finally, I would like to thank my parents, who first instilled in me the love of learning and have always supported me in my academic endeavours.

Contents

List of Figures

List of Tables

Chapter 1

Background

This thesis is fundamentally about finding an efficient and accurate method for a large numerical integration problem. Understanding the context of this problem, however, requires an understanding of both basic medical imaging concepts and of the specific calculation from which the problem arises. This background chapter is therefore divided into three sections. The first gives an overview of Single Photon Emission Computed Tomography (SPECT). The second discusses the Analytic Photon Distribution (APD) method and the numerical integration problem that arises from it. The final section gives an overview of the relevant numerical integration methods.

1.1 Overview of SPECT

SPECT is a commonly used functional medical imaging modality. A functional imaging modality gives information about aspects of the body's physiology, such as blood flow or cell metabolism. This is in contrast to modalities such as X-ray computed tomography (CT) or Magnetic Resonance Imaging (MRI) which primarily depict the anatomy of the patient.

The SPECT imaging process first requires administering a radiopharmaceutical to the patient. A radiopharmaceutical is a chemical compound which mimics some nutrient or other biological compound (glucose, for example), labeled with a radioactive isotope. After the radiopharmaceutical is administered to the patient, it is absorbed by some targeted organ in the body. The radiation emitted by the tracer can then be detected using a camera system. The camera acquires a set of discrete two-dimensional projections from multiple angles around the patient. A mathematical algorithm is then used to reconstruct these projections

into a three-dimensional image representing the distribution of the radiopharmaceutical in the body. Based on this information, the physician can make a diagnosis.

The two most common applications of SPECT are perfusion studies and detection of metastatic tumors. In a perfusion study, one uses the radiotracer to measure blood flow in a particular organ, such as the heart, lungs, or brain. A reduced amount of blood flow in some regions may indicate a health risk to the patient. In a tumor study, on the other hand, the physician looks for regions that show higher uptake of the radiotracer than surrounding tissue. Cells showing abnormally high uptake rates may be cancerous.

While a complete treatment of SPECT is not necessary in the discussion of this work, it is important to understand some of the fundamental concepts of the imaging process. In particular, the concepts of collimation, energy resolution and photon interaction with matter play an important part in the calculations done by the APD method, upon which this work is based. As such, we will briefly discuss these three elements of SPECT imaging. For a more thorough discussion of SPECT, see [AW04], particularly Chapters 2, 7 and 22.

1.1.1 Collimation

A gamma camera detects the photons emitted by the radiotracer using a flat scintillator crystal, usually sodium iodide (NaI). When a photon strikes the crystal, the energy from that photon creates a flash of light, which is detected by an array of photomultiplier tubes (PMT) located behind the crystal. These tubes produce an electrical current which is detected by the accompanying electronics, which then register this impact as an event. However, while the crystal and PMT array detect the location at which the impact occurred, they are not capable of determining the direction from which the photon originated. So, if photons are just allowed to strike the detector crystal at any angle, there is no hope of obtaining a meaningful reconstruction since every detected photon could have originated from anywhere.

To solve this problem, the SPECT camera system consists of another component known as a collimator. A collimator is a thick slab of heavy material (typically lead) which is perforated by an array of long, narrow channels. The collimator sits on top of the detector crystal surface and, in principle, ensures that every photon detected by the crystal must have originated from somewhere within the fairly narrow field of view of the collimator. A simple reconstruction method will then assume that every photon detected originated from somewhere along the straight line normal to the detector at the point where the photon was detected. More realistic methods will take into account that the field of view of each

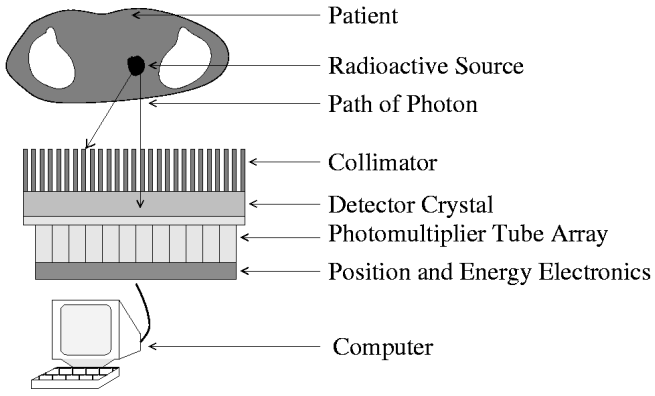

Figure 1.1: Schematic diagram of a typical SPECT camera system. The collimator prevents photons from being detected unless they follow a path that is almost normal to the detector surface. Taken from [Van04], with permission of the author.

collimator hole is actually a cone, defined by the angle of acceptance of the collimator. Figure 1.1 gives an illustration of a typical SPECT camera setup, including the collimator.

Collimation considerably reduces the number of photons detected by the camera system, by a factor of about 10^{-4}. As a result, SPECT requires a fairly long acquisition time in order to record a statistically significant number of events.

1.1.2 Energy Resolution

All photons originating from a radiotracer are emitted with a specific energy – or for some isotopes, one of several possible energies. Technetium-99m, for instance, emits photons with energy 140 keV, while Indium-111 emits photons with 171 or 245 keV. In principle, one would like to only record collisions with the detector crystal where the incident photon's energy matches with that of the radiotracer, to avoid detecting radiation from other sources. Unfortunately, in practice the NaI crystals that are most often used in a SPECT camera system are only able to measure the energy of the incident photon with limited accuracy – typically 5-10%. The measurement of the photon's energy is modeled as a Gaussian distribution centred at the true energy value.

As a result of the detector's limited energy resolution, the camera system has to accept photons detected within a certain range of energies (the energy window) as having been emitted by the radiotracer. For instance, with Technetium-99m one typically uses a 20%

energy window of 126-154 keV. While the use of an energy window is necessary in order to acquire sufficient data to create a meaningful image, it has the undesirable side-effect that photons from other sources, or scattered photons from the radiotracer (see next section) will be detected and contaminate the image.

1.1.3 Photon Interactions with Matter

As the emitted photon travels through the body, there are several ways in which it can interact with body tissue before reaching the camera detector. Sometimes the photon interacts with an atom and is completely absorbed, and thus never reaches the detector. This process is known as attenuation (or photoelectric absorption), and can distort both the quantitative and qualitative accuracy of an image, because fewer photons will be detected from sources deep within the body, giving the appearance of reduced activity in those areas.

A second type of photon interaction occurs when the incident photon strikes an electron and scatters off of it. This phenomenon is known as Compton (or incoherent) scattering. In addition to changing the direction of the photon, Compton scattering also causes it to lose energy, as some is transferred to the struck electron. The amount of energy lost depends on the magnitude of the scattering angle θ. The energy of the photon after scattering, E_f, can be determined by the formula:

$$E_f = \frac{E_0}{1 + \alpha(1 - \cos\theta)} \tag{1.1}$$

where E_0 is the initial energy of the incident photon, and

$$\alpha = \frac{E_0}{m_e c^2}, \tag{1.2}$$

is the ratio of the initial photon energy to the rest mass energy of the electron. The probability that a photon interacts with a free electron and scatters with angle θ is given by the Klein-Nishina cross-section:

$$\frac{d\sigma}{d\Omega}(\theta, \alpha) = \frac{r_0^2}{2}\left(1 + \cos^2\theta\right)\left(\frac{1}{1 + \alpha\left(1 - \cos\theta\right)}\right)^2\left(1 + \frac{\alpha^2(1 - \cos\theta)^2}{(1 + \alpha[1 - \cos\theta])(1 + \cos^2\theta)}\right) \tag{1.3}$$

where α is as defined in (1.2), and r_0 is the classical electron radius. Note that the Klein-Nishina cross-section gives the probability of photon interaction with a free electron. Since the majority of electrons are bound to an atom, a multiplicative correction known as the incoherent scattering function must be applied to the cross-section when using it in practice.

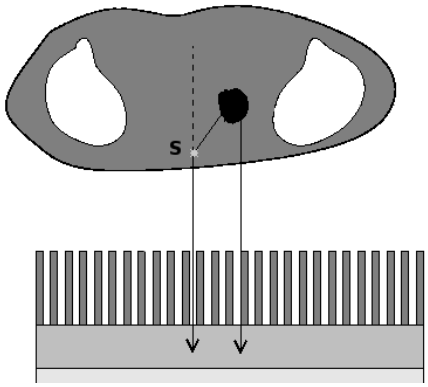

Figure 1.2: Illustration of a Compton-scattered photon. The photon on the left has scattered at site S and then been detected by the camera system. Without scatter correction, the reconstruction algorithm will assume that the photon originated from somewhere along the dashed line, when in fact no activity exists there.

This scattering function represents the probability that an electron, which acquired energy from a scattering photon, will ionize or escape from the atom. [Wel97]

A third type of photon interaction is known as Rayleigh (coherent) scattering, which occurs when the electromagnetic field of an atom deflects the incident photon. In this case the photon changes direction without losing any energy. Rayleigh scattering occurs mostly for low energy photons (50 keV or less), and is a much less prevalent effect in SPECT imaging than Compton scattering. It will not be discussed further in this work.

Compton scattering is a significant problem in SPECT imaging because it gives false spatial information about the distribution of activity in the patient. Although a scattered photon will lose energy, it can still have sufficient energy to be detected within the energy window of the camera system. For instance, a 140 keV photon scattering through an angle of 57° will have a final energy of 126 keV, and thus is still likely to be detected in a 20% energy window. A naïve reconstruction algorithm will then assume that the photon originated from somewhere in the region defined by the collimator's field of view, when in fact it came from elsewhere. Compton scattering thus results in a degraded image, where there will be reduced contrast around regions of high activity, as well the appearance of increased overall activity. There may also appear to be activity where none actually exists. Figure 1.2 gives an illustration of Compton scattering and how it can affect the reconstruction.

1.2 The APD method

In this section we describe the Analytic Photon Distribution (APD) method, with particular emphasis on calculations requiring numerical integration.

1.2.1 Overview of APD

The APD method calculates the point spread function (PSF) for a given distribution of activity, separated into primary (unscattered) and scattered components. The PSF is a two-dimensional function which represents the computed probabilities that a photon emitted by the radiotracer from some point in the patient volume is detected at any point on the detector. In other words, the PSF shows the expected distribution of detected photons on the camera head due to a point source, for a given acquisition angle. Unlike real SPECT data, the distribution calculated by APD is completely noiseless. APD takes into account the distribution of activity in the body (the activity map) as well as the distribution of body tissues (the attenuation map). The attenuation map is necessary for accurate PSF calculation, since denser tissues are more likely to scatter or attenuate photons. Both of these maps are represented as discretized volumes, consisting of cubical voxels. The surface of the detector is also modeled as a discrete 2-dimensional pixel grid. The matrix size used in APD is $64 \times 64 \times 64$ for the activity and attenuation maps, and 64×64 for the detector grid.

The fundamental idea behind APD is that the calculation of PSF probabilities relies on two different types of parameter. Some parameters, such as the matrix size, imaging geometry, and the energy window used for acquisition, are completely independent of the actual patient being imaged. Other parameters, such as the activity and attenuation maps, will be different for each patient. APD separates the PSF calculation into two components – one independent of the patient, and one which depends on patient-specific data. The first component can be precalculated and stored in a set of lookup tables, rather than recalculating it for every patient. Thus we can greatly reduce the amount of calculation required to do full PSF calculations for multiple patients.

The original APD method was developed in [Wel97], and explicitly calculates the PSF for every voxel containing activity. Doing so resulted in extremely long calculation times, which led to the development of the APDI method several years later [Van04]. APDI (standing for APD with Interpolation) only calculates the PSF for a subset of the voxels containing

activity, and then estimates the PSFs for the remaining voxels using trilinear interpolation. It was found that this modification to the program reduced computation time by a factor of 10 to 20. Furthermore, the differences from the exact APD calculation introduced by interpolation were small, relative to the statistical uncertainty already present in SPECT projection data. In this work, the distinction between APD and APDI is not important, as we focus on parts of the calculation that are identical in both methods.

Because APD gives a PSF which separates primary from scattered photons, it can be used to correct for scatter as part of a reconstruction algorithm. [Van04] describes how APD can be used in tandem with Ordered Subsets Expectation Maximization (OSEM – a well-known iterative image reconstruction algorithm) to produce a scatter-corrected image. This method is referred to as OSEM-APDI. The main problem with APD is that the calculation time is too long for it to be applied clinically – even after the APDI improvement. Photon scatter probabilities must be calculated for every source location and every possible scatter location, for both first-order scatter (photons that scatter only once before being detected) and second-order scatter (photons that scatter twice before being detected). To use APD in a reconstruction algorithm, we must also calculate the PSF for each of the angles of rotation at which the projection data was acquired – often 60 or 120 angles. The calculation of scatter probabilities requires a great deal of numerical integration, and this integration makes up the bulk of APD's calculation time.

1.2.2 PSF Calculations in APD

The simplest APD calculation is of the primary photon distribution function (PDF). A primary photon is one which travels straight from the source voxel to the detector pixel without scattering anywhere in the patient, as illustrated by the dash-dotted line in Figure 1.3. Primary photons give accurate information about the distribution of activity in the body, and so the objective of scatter correction is to keep as much information as possible from primary photons, while eliminating false information obtained from scattered photons.

The expected number of primary photons emitted from a source voxel $\vec{s}$ and detected in the detector pixel centred at $\vec{n}_c$ is given by [Wel97]:

$$PDF\left(\vec{s},\vec{n}_c\right) = QP_E(0)\exp\left(-\int_{\vec{s}}^{\vec{n}_c}\mu(x)dx\right)\int_{A_c} d^2n\, F(\xi)\frac{\cos\xi}{4\pi r_{ns}^2} \tag{1.4}$$

where:

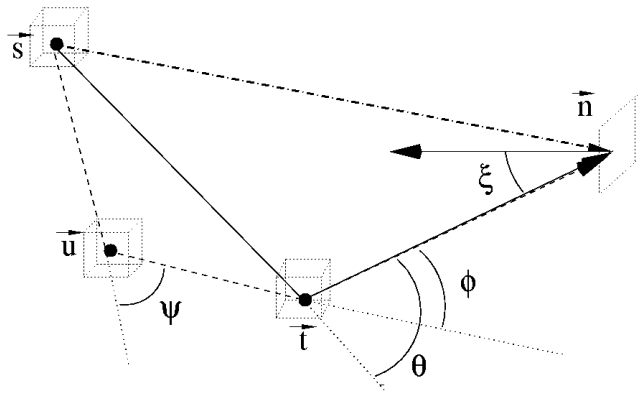

Figure 1.3: Schematic diagram showing possible photon paths from a source voxel centred at $\vec{s}$ to the detector pixel centred at $\vec{n}$. The dash-dotted line represents a primary photon, travelling directly from $\vec{s}$ to $\vec{n}$ without scattering. The first-order scattered photon (solid line) originates from $\vec{s}$ and scatters once at position $\vec{t}$ before being detected at $\vec{n}$. Finally, the dashed line represents a second-order scattered photon, which scatters at position $\vec{u}$ and position $\vec{t}$, before finally being detected at $\vec{n}$. Taken from [Van04], with permission of the author.

- Q is the activity at $\vec{s}$ – the number of photons emitted from that point during the acquisition,
- A_c is the area of the detector pixel
- $F(\xi)$ is the probability that a photon arriving at the collimator surface at an angle ξ with respect to the normal will pass through the collimator and strike the detector crystal,
- $P_E(\theta)$ is the probability that a photon that has Compton scattered through an angle θ will be detected in the energy window being used,
- r_{ns} is the distance between the points $\vec{n}$ and $\vec{s}$,
- $\frac{\cos\xi}{4\pi r_{ns}^2}\, d^2n$ is the solid angle of a sphere centred at $\vec{s}$ and subtended by the detector pixel element d^2n, and
- $\mu(y)$ represents the attenuation coefficient at position y along the path travelled by the photon. The attenuation coefficient of a particular tissue measures the likelihood that this tissue attenuates a photon with a given energy.

In this formula we can clearly see how the calculation is split into patient-dependent and independent components. The area integral over A_c depends only on the geometry of the camera system, while $P_E(\theta)$ depends only on the initial energy of the emitted photons and the energy window being used. In contrast, the path integral of the attenuation from $\vec{s}$ to $\vec{n}$ depends on the attenuation map for a specific patient, and the activity level Q depends on the activity map for that patient.

In APD, the area integral is calculated as a discrete function of source and detector position, and stored in a primary lookup table. The appropriate values can then simply be accessed from the table when calculating the PSF, rather than calculating them every time. This primary lookup table is quite small, as a primary photon can only be detected in a fairly small subset of detector pixels under the source, due to the small acceptance angle of the collimator. As a result, there are relatively few table entries to be calculated.

The PDF calculation is quite simple and does not require a significant amount of computation time. The calculation of PSFs for first and second-order scatter is much more complicated. The number of photons that originate from source voxel $\vec{s}$, scatter once, and are then detected in pixel $\vec{n}_c$ (as illustrated by the solid line in Figure 1.3) is given by the scatter distribution function (SDF) [Wel97]:

$$SDF_1\left(\vec{s},\vec{n}_c\right) = Q\sum_{i=1}^{I}\rho_e(\vec{t}_i)K_{st_in_c}\exp\left(-\int_{\vec{s}}^{\vec{t}_i}\mu(y)\,dy-\int_{\vec{t}_i}^{\vec{n}_c}\mu(y,\theta)\,dy\right), \tag{1.5}$$

where

$$K_{st_in_c} = \int_{A_i} d^2n \int_{V_k} d^3t \frac{d\sigma}{d\Omega}(\theta,\alpha)F(\xi')P_E(\theta)\frac{\cos(\xi')}{4\pi r_{ts_j}^2 r_{nt}^2} \tag{1.6}$$

and

- $\frac{d\sigma}{d\Omega}(\theta,\alpha)$ is the Klein-Nishina scattering cross-section (1.3),
- $\rho_e(\vec{t}_i)$ is the electron density in scattering voxel $\vec{t}_i$,
- θ is the Compton scattering angle,
- $\mu(y,\theta)$ is the attenuation at position y for a photon which has scattered through an angle θ, with $\mu(y) \equiv \mu(y,0)$[1], and

[1]Note that μ now depends on θ, since attenuation depends on the energy of the photon as well as the density of the body tissue. Since the photon's energy changes after scattering through an angle θ according to (1.1), the attenuation function must take this into account.

- α is as defined in (1.2).

All other symbols are defined as in (1.4). Note that since there are many voxels $\vec{t}_i$ where the photon can scatter before being detected at $\vec{n}_c$, SDF_1 must include a summation over all I voxels which could be scatter sites.

In (1.5), all patient-independent factors are collected into the term $K_{st_in_c}$, given in (1.6). This term requires integrating over the area of the detector pixel $\vec{t}_i$ as well as the volume of the scattering voxel $\vec{n}_c$, for a total of five dimensions. Since this term depends only on geometry (as well as the energy window, which is constant), symmetry allows APD to avoid having to calculate the factor $K_{st_in_c}$ for every combination of source, scattering and detector site. Instead, the first-order lookup table can be parameterized by five physical dimensions: four distances and one angle, as illustrated in Figure 1.4.

As with the primary lookup table, these $K_{st_in_c}$ factors are calculated for a discrete set of input values and stored as a lookup table. Unlike the primary lookup table, however, this table is quite large and expensive to calculate, even after taking advantage of symmetry as described above. For the level of discretization chosen for APD, the table consists of approximately eight million values, compared to about five thousand for the primary table. Roughly one quarter of these values are calculated directly using (1.6), while the rest are estimated using interpolation.

Finally, the APD method also performs a second-order scatter calculation, for photons that scatter twice prior to being detected, as illustrated by the dashed line in Figure 1.3. The photon first scatters through an angle ψ at voxel $\vec{u}$, then through angle ϕ at the second voxel $\vec{t}$, before being detected. In principle, this calculation would require an 8-dimensional integral – a volume integral over each of the two scattering sites, plus an area integral over the detector pixel. Such a computation would be extremely expensive, however. Instead, the APD method uses the $K_{st_in_c}$ factors (1.6) from the first-order calculation, as well as a number of approximations, in order to simplify the second-order calculation. As a result, calculating the second-order SDF requires only performing one additional volume integral. The formula for the second-order SDF is [Wel97]:

$$SDF_2\left(\vec{s},\vec{n}_c\right) = Q\sum_{i=1}^{I} K'_{st_in_c}\int_{\vec{s}}^{\vec{t}_i}\mu(y)\,dy\,\exp\left(-\int_{\vec{s}}^{\vec{t}_i}\mu(y)\,dy-\int_{\vec{t}_i}^{\vec{n}_c}\mu(y,\theta)\,dy\right), \qquad (1.7)$$

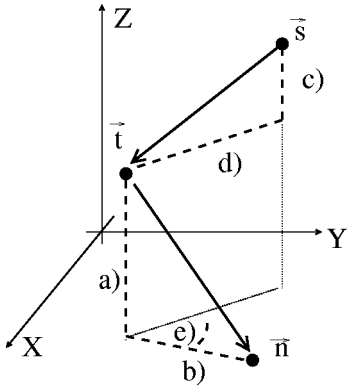

Figure 1.4: Diagram of the five-dimensional parameterization of the APD lookup tables. The XY plane represents the detector surface, with the Z-axis being normal to the detector. The point $\vec{s}$ represents the source voxel, while $\vec{t}$ is the last scattering voxel and $\vec{n}$ is the detector pixel. The five dimensions used to parameterize the table are the distances from $\vec{t}$ to $\vec{n}$ in the Z-direction (a) and XY-direction (b), from $\vec{s}$ to $\vec{t}$ in the Z-direction (c) and XY-direction (d), and finally the angle (e) between the vectors $\vec{s}-\vec{t}$ and $\vec{t}-\vec{n}$, projected into the XY plane. Taken from [Van04], with permission of the author.

where

$$K'_{st_in_c} = \frac{\rho_w}{\mu_w r_{st_i}} \sum_{j=1}^{J} K_{u_j t_i n_c} \times \int_{V_j} d^3u \left(\frac{d\sigma\left(\psi_i, \alpha\right)}{d\Omega} \right) \frac{1}{r_{us}^2} \frac{P_E\left(\psi_i, \phi_{ic}\right)}{P_E\left(\phi_{ic}\right)} \exp\left(\mu_w \left(r_{st_i} - r_{us} - r_{ut_i}\right)\right) \quad (1.8)$$

The terms ρ_w and μ_w are the electron density and attenuation coefficient for water, respectively. All other terms are as defined for the equations (1.4) and (1.5), with some small modifications where necessary. For example, the function P_E can now take two arguments, to account for the fact that the photon has lost energy by scattering through two angles. For the derivation of the second-order formula, see [Wel97], Section 3.2.2.

To calculate the factors $K'_{st_in_c}$, APD first evaluates the volume integral in (1.8) for a discrete set of values and stores it in an intermediate table. These factors are then multiplied with the appropriate factors $K_{u_j t_i n_c}$ (1.6) and summed as per the formula. The final $K'_{st_in_c}$ factors are stored in the second-order lookup table. The second-order lookup table can be parameterized in exactly the same way as the first, as specified in Figure 1.4.

Higher-order scattered photons are also present in acquired SPECT data, but are rarely

detected since the photons will usually lose too much energy to fall within the energy window. APD does not perform any higher-order scatter calculations, but rather makes a small adjustment to the second-order scatter contribution to compensate.

As the formulas show, the PSF calculations in APD consist of two types of integral. The first type are path integrals of the attenuation values through the patient's body. These integrals must be calculated for each individual patient, and therefore cannot be stored in a lookup table. The second type are the area and volume integrals that are part of the lookup table calculations.

1.2.3 Features of the Integrands

It is now useful to examine the integrands in the formulas (1.4), (1.6) and (1.8) in more detail to determine any important features. The primary photon integrand in (1.4) is the simplest to analyze. Rather than modelling the collimator as a discrete collection of holes through which photons can pass, APD models the collimator with the acceptance function $F(\xi)$, described in Section 3.1 of [Wel97]. The larger the angle ξ that the incident photon makes with the detector normal, the smaller the probability that the photon will pass through the collimator and be detected. In fact, if ξ exceeds the maximum acceptance angle of the collimator, $F(\xi)$ will be zero. As a result, $F(\xi)$ can be interpreted as forming a cone on the detector surface. The maximum probability of detection occurs when ξ is zero, at the point on the detector surface directly under the source. $F(\xi)$ then decreases radially until ξ exceeds the collimator acceptance angle, at which point it becomes zero. For a source point that is far away from the detector surface, the cone will be wide since the photon can be detected over a larger region on the detector, but as the source moves closer the cone becomes more narrow.

The area integral in (1.4) will be over some portion of this cone, and possibly the entire cone if it is sufficiently small. As a result, in some cases the integrand will be nonsmooth since the cone has discontinuities in the first derivative. This includes the point at the top of the cone, as well as the edge along the base where the integrand becomes zero. Figure 1.5 gives an illustration for two different source locations.

We now consider the first-order scatter integral (1.6). We can view this 5-dimensional integral as an iterated integral, consisting of two parts. First, consider holding the location of the scattering point fixed, and simply integrating over the area of the detector pixel. This integral will be similar to the one seen in the primary case, as the integrand has a roughly

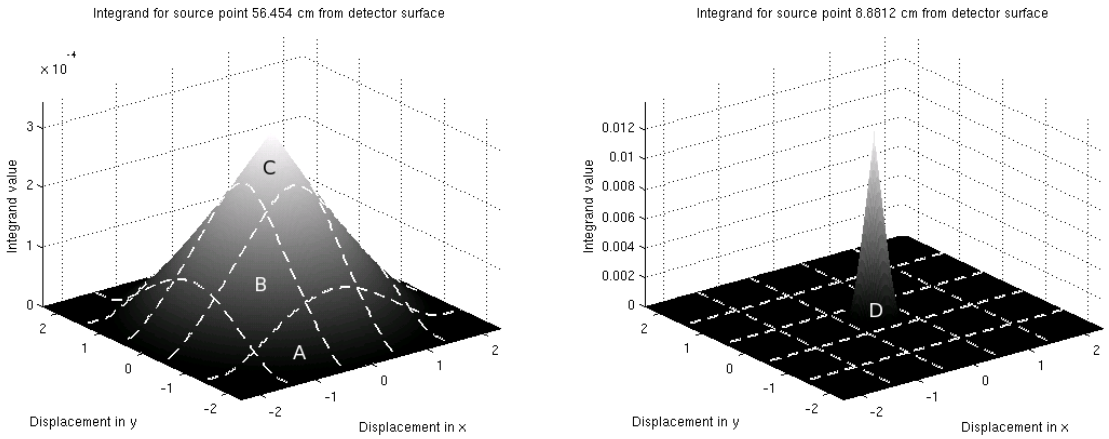

Figure 1.5: Surface plot of integrand for primary lookup table, for two different source locations. The detector pixel grid is overlaid on the plot, and the source voxel was located above the centre pixel. In the left figure, the source is far away from the detector, and the photon can be detected in several pixels. The integrand in pixel A is nonsmooth because it contains part of the edge of the cone. The integrand in pixel C (the pixel directly below the source) is also nonsmooth since it contains the tip of the cone. The integrand in pixel B is smooth, however. In the right figure, the source is quite close to the detector. As a result, the photon can only be detected in pixel D, directly below the source. The integrand is zero elsewhere.

conical shape. However, since the photon has been scattered, the integrand is no longer radially symmetric about the point directly below the scattering location. Photons on one side of the "cone" will have scattered through a larger angle than the photons on the other side, and thus will have lost more energy. Since the energy window is centred around the initial energy of the photon, these photons with less energy will have a lower probability of being detected. As a result, the "cone" will be slightly concave on one side. Figure 1.6 gives an example of this type of behaviour.

To evaluate the three-dimensional volume integral, we then consider varying the location of the scattering point within the scattering voxel. The function being integrated is now some portion of the volume of the conical function that was just discussed – the part contained within one detector pixel. As we vary the coordinates of the scattering point, this function will have fairly smooth behaviour compared to the integrand at the detector level. This is because the volume over the pixel area varies fairly smoothly as the cone-shaped function

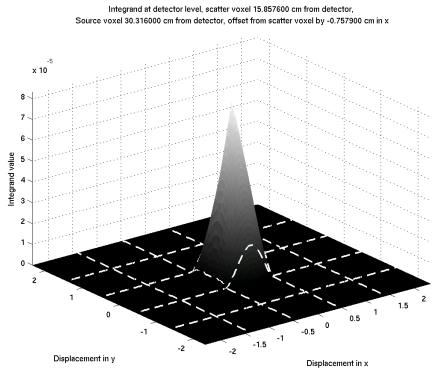

Figure 1.6: Surface plot of integrand for the first part of the first-order scatter integral, with pixel grid overlaid. The source point is 14.46 cm above the scattering voxel, and slightly offset in the horizontal plane. The photons on the near side of the cone have been scattered through a larger angle than the photons on the far side of the cone, and thus have a lower probability of detection. As a result, the cone is slightly concave on that side.

is translated on the detector surface (if the scatter point varies in the plane parallel to the surface) or as it broadens or narrows (as the distance of the scatter point from the detector surface varies). Figure 1.7 gives an example of how the volume function changes as the position of the scatter point shifts within the scattering voxel. The key point is that even though the function is initially not smooth, after integrating it over the two detector pixel dimensions, the resulting function has smoother behaviour.

The second-order integrand does not have a simple physical interpretation like these first two integrands, as it is essentially a scaling factor that is multiplied by first-order table elements to give second-order scattering probabilities. It is also more complicated to visualize this integrand as there are more parameters coming into play. Since the second-order calculation is not as significant as the primary and first-order calculations (since it contributes much less to the acquired projection data), we will not examine this integrand in detail as we did the other two.

1.3 Numerical Integration

As described in the previous section, APD calculations require multidimensional integration of a numerically calculated function over pixel areas and voxel volumes. The method that

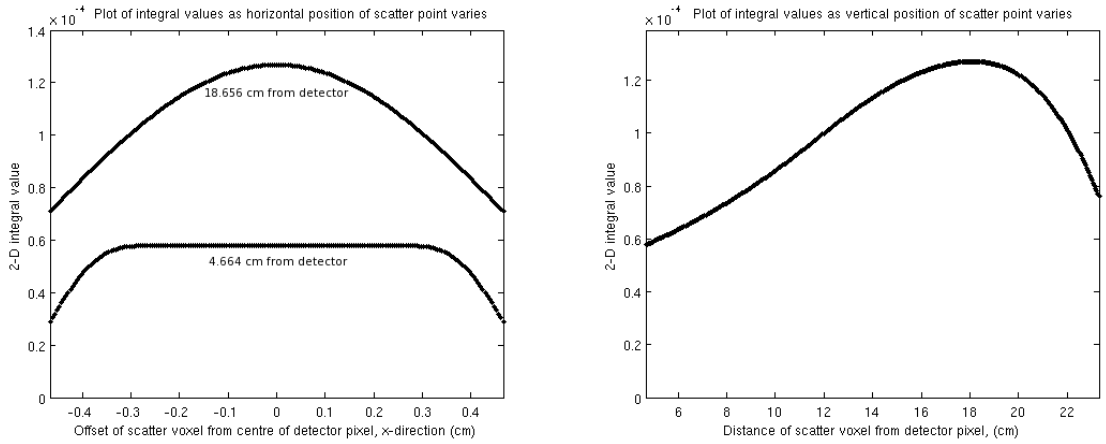

Figure 1.7: Plots showing how the integrand varies smoothly once we have integrated over the area of the detector pixel. In both cases, the scattering voxel is located directly above the detector pixel. The source voxel is 30.316 cm above the detector surface, and is offset from the scattering voxel by 7.579 cm in the x-direction. In the left figure, the y and z co-ordinates of the scattering point are held fixed, while it varies in the x-direction inside the scattering voxel. In the right figure, the x and y co-ordinates of the scattering point are held fixed, while we vary the distance from the detector surface. In this second figure the distance from the detector surface is ranging over many scattering voxels; over a single voxel there is only a slight change in the value of the integrand. These integrand values are approximate numerical values calculated by Gaussian quadrature (cf. Section 1.3.4)

was being used by the code was to evaluate the multidimensional integrals as iterated integrals using Romberg's method. In this thesis we will investigate the use of iterated Gaussian quadrature instead. A discussion of numerical integration, including these two methods, follows.

1.3.1 Interpolation

The interpolation problem is fundamental to many numerical integration methods. Given a set of points x_i, $i = 0 \ldots n$, and corresponding function values f_i, the goal of interpolation is to find a simple function $P(x)$ such that

$$P(x_i) = f_i \tag{1.9}$$

for all i. The function $P(x)$ can then be used, for instance, to estimate the value of f at other points close to the x_i.

One of the most basic interpolating functions is the Lagrange interpolating polynomial. If one defines polynomials L_i as follows:

$$L_i(x) = \sum_{\substack{k=0 \\ k \neq i}} \frac{x - x_k}{x_i - x_k} \tag{1.10}$$
$$= \begin{cases} 1 & \text{for } x = x_i \\ 0 & \text{for } x = x_j,\ j \neq i \end{cases}$$

then the following polynomial will have degree at most n and satisfy (1.9):

$$P(x) = \sum_{i=0}^{n} f_i L_i(x). \tag{1.11}$$

Furthermore, this polynomial is the unique polynomial of degree n or less satisfying this property.

Constructing the Lagrange polynomial using this definition is not computationally efficient, however. If one wishes to estimate the value of the function at a single point x using the Lagrange interpolant, then a better method is Neville's algorithm [SB02]. Neville's algorithm is based on the fact that the Lagrange polynomial can be constructed recursively. Suppose we have $n + 1$ data points (x_i, f_i), and let $P_{j,k}$ denote the Lagrange polynomial interpolating the points $x_j,\ x_{j+1} \ldots x_k,\ j \leq k$. Then, the Lagrange polynomials can be derived recursively by

$$P_j(x) = f_j \tag{1.12}$$

and

$$P_{j,k}(x) = \frac{(x - x_j)P_{j+1,k}(x) - (x - x_k)P_{j,k-1}(x)}{x_k - x_j}. \tag{1.13}$$

That is, the Lagrange polynomial with degree n can be constructed from two Lagrange polynomials of degree $n - 1$, each of which passes through all but one of the interpolation points. To evaluate the full Lagrange polynomial of degree n at a single point, then, one can construct it using a table, as illustrated in Table 1.1.

In cases where the Lagrange polynomial needs to be evaluated at several points, however, other methods which explicitly construct the polynomial may be preferable to Neville's.

$P_0(x)$					
$P_1(x)$	$P_{0,1}(x)$				
$P_2(x)$	$P_{1,2}(x)$	$P_{0,2}(x)$			
$P_3(x)$	$P_{2,3}(x)$	$P_{1,3}(x)$	$P_{0,3}(x)$		
$\vdots$	$\vdots$			$\ddots$	
$P_n(x)$	$P_{n-1,n}(x)$	$\dots$	$\dots$	$\dots$	$P_{0,n}(x)$

Table 1.1: Table showing how Neville's algorithm recursively constructs the Lagrange polynomial. The elements $P_i(x)$ in the leftmost column are simply the data values f_i that were provided. The table can be built column-wise from left to right or row-wise from top to bottom, using the recursion formula (1.13).

1.3.2 Basic Quadrature Methods

Suppose we want to evaluate the definite integral

$$I = \int_a^b f(x)\,dx \tag{1.14}$$

The most basic numerical quadrature methods are derived by replacing the integrand $f(x)$ with an interpolating function, and then exactly integrating the interpolant. The simplest of these methods is the trapezoid rule, which is obtained by integrating the Lagrange interpolant of degree 1 passing through both endpoints [Ral65]:

$$I = \frac{h}{2}\left[f(a) + f(b)\right] - \frac{h^3}{12}f''(\xi) \tag{1.15}$$

where $h = b - a$ is the spacing between the interpolation points, and ξ is some unknown point in the interval (a, b). To estimate I, we drop the final term which may be interpreted as an error in this estimate. The trapezoid rule is said to have degree of precision 1, since it will give the exact value of the integral for polynomials up to degree 1.

The trapezoid rule is obviously a fairly crude integration formula. One way of obtaining higher accuracy is to use more points and use a higher-order Lagrange interpolant. For instance, using three equally-spaced points and the second-order Lagrange interpolant gives Simpson's rule:

$$I = \frac{h}{3}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right] - \frac{h^5}{90}f^{(4)}(\xi), \tag{1.16}$$

which has degree of precision 3. Note that here the spacing between points is $h = \frac{1}{2}(b-a)$.

In general, however, constructing Lagrange interpolants of increasing order and integrating them is not practical as the number of points increases, since the interpolant tends to become highly oscillatory. A more effective method of improving accuracy is to split the interval of integration into smaller subintervals of equal length and use low-order quadrature formulas on each of those subintervals. Let x_0 be the left endpoint a and x_n be the right endpoint b, with equally spaced interpolation points x_i between them. The spacing between points is denoted by $h = \frac{1}{n}(b-a)$, as before. Then, applying this piecewise approach leads to composite integration formulas, such as the composite trapezoid rule:

$$I = \frac{h}{2}\left[f(x_0) + 2f(x_1) + 2f(x_2)\ldots + 2f(x_{n-1}) + f(x_n)\right] - \frac{b-a}{12}h^2 f''(\xi_n) \tag{1.17}$$

and the composite Simpson's rule:

$$I = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3)\ldots \\ \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] - \frac{b-a}{180}h^4 f^{(4)}(\xi_n) \tag{1.18}$$

These formulas both fall under the category of closed Newton-Cotes formulas; "closed" because they include the endpoints of the interval, and "Newton-Cotes" formulas because the evaluation points are all equally spaced. While these formulas are simple to apply, their accuracy is limited by the spacing h between points. To obtain higher accuracy one must use more points.

1.3.3 Romberg's Method

A more efficient way of improving accuracy is Romberg's method. Romberg's method combines the composite trapezoid rule (1.17) with a technique known as Richardson's extrapolation to improve the accuracy of the calculation by "cancelling off" error terms of increasing order. In particular, while (1.17) gives the composite trapezoid rule with the error term in closed form (obtained by evaluating $f''(x)$ at an unknown point ξ in the interval), one can also write it with the error term as an infinite series:

$$I = \frac{h}{2}\left[f(x_0) + 2f(x_1) + 2f(x_2)\ldots + 2f(x_{n-1}) + f(x_n)\right] + \sum_{k=1}^{\infty} a_{2k}h^{2k}. \tag{1.19}$$

(This form is derived using the Euler-Maclaurin sum formula – see [Hen82], pp. 282-285 for details.) The error term coefficients a_{2k} are linear combinations of the $2k-1^{\text{th}}$ derivative of

the integrand evaluated at the endpoints a and b. So, writing the composite trapezoid rule in this form implicitly assumes that f has any desired number of continuous derivatives [Hen82].

Provided that the form (1.19) holds, if one were to apply the trapezoid rule with N points having a spacing of h between them, then apply it again with $2N$ points (having a spacing of $\frac{h}{2}$), then the leading order error term a_2h^2 should be 4 times larger for the first approximation than the second. If we call the first approximation T_N and the second T_{2N}, it then follows that the expression

$$I \approx \frac{1}{3}\left(4T_{2N} - T_N\right) \tag{1.20}$$

should have a leading error term of order h^4, since the $\mathcal{O}\left(h^2\right)$ terms will cancel off. In fact, if we actually evaluate (1.20), we derive the composite Simpson's rule approximation (1.18), which does have an $\mathcal{O}\left(h^4\right)$ leading order error. We also note that the composite trapezoid rule (1.19) only contains even-powered error terms, so this one step allows us to go from having an $\mathcal{O}\left(h^2\right)$ error to an $\mathcal{O}\left(h^4\right)$ error.

Romberg's method is a generalization of this process that uses Richardson's extrapolation to successively cancel off error terms until the desired accuracy is reached. The Romberg approximation is usually calculated in a tabular fashion, as illustrated in Table 1.2.

Entries are calculated row-by-row. The entry $R_{0,k}$ is the composite trapezoid rule approximation with 2^k subintervals, and is calculated in an intelligent way so as to re-use as much data as possible from the previous application of the composite trapezoid rule:

$$R_{0,k} = \frac{1}{2}\left(R_{0,k-1} + h_{k-1}\sum_{i=1}^{2^{k-1}} f\left(a + (2i-1)\,h_k\right)\right) \tag{1.21}$$

where $h_k = \frac{b-a}{2^k}$ is the spacing between the points. Meanwhile, the entry $R_{j,k}$ is the result of applying Richardson's extrapolation to the previously calculated values $R_{j-1,k-1}$ and $R_{j-1,k}$ in order to cancel off the $\mathcal{O}\left(h_k^{2j}\right)$ error term:

$$R_{j,k} = \frac{1}{4^j - 1}\left(4^j R_{j-1,k} - R_{j-1,k-1}\right) \tag{1.22}$$

The final output of Romberg's method with m steps is the last diagonal entry $R_{m,m}$. As desired, for smooth integrands the Romberg's method approximations $R_{k,k}$ will usually converge to the true value I much faster than the trapezoid rule approximations $R_{0,k}$.

$R_{0,0}$					
$R_{0,1}$	$R_{1,1}$				
$R_{0,2}$	$R_{1,2}$	$R_{2,2}$			
$R_{0,3}$	$R_{1,3}$	$R_{2,3}$	$R_{3,3}$		
$\vdots$	$\vdots$			$\ddots$	
$R_{0,m}$	$R_{1,m}$	$\dots$	$\dots$	$\dots$	$R_{m,m}$

Table 1.2: Table of Romberg's method calculations

However, it is not always necessary to calculate the full table. Each column of the table should also converge to the true value faster than the first column, and so one could also just use the last entry of one of these columns as the final result.

Table 1.3 gives an example of ideal behaviour for Romberg's method. If f is not differentiable, however, then the error behaviour indicated by (1.19) is not applicable, and Romberg's method may not give the expected convergence. For instance, consider the "hat function" shown in Figure 1.8:

$$f(x) = \begin{cases} 1 - \frac{1}{0.15}\,|x - 0.6| & \text{if } |x - 0.6| < 0.15 \\ 0 & \text{otherwise} \end{cases} \tag{1.23}$$

The integral of $f(x)$ on the interval $[0, 1]$ is clearly 0.15 (since the area is a triangle with height 1 and base 0.3), but $f(x)$ is not differentiable on that interval due to the presence of several sharp corners. When we apply Romberg's method (Table 1.4), we find that the values obtained from extrapolation are actually worse than those obtained with the composite trapezoid rule, since the extrapolation assumes that the error term has a form which does not hold for non-differentiable functions.

When applying Romberg's method, one typically does not know *a priori* how many refinements need to be made. Instead, Romberg's method is often applied with some error tolerance specified by the user. One can compare successive Romberg approximations to get an estimate of the error in the integrand, and then continue the refinement if necessary. Romberg's method is particularly well suited to this kind of refinement, since the composite trapezoid rule can reuse the function evaluations done at the previous step in the next

n				
2	1.35914091422952			
3	0.88566061595228	0.72783384985986		
5	0.76059633244804	0.71890823794663	0.71831319715241	
9	0.72889017701469	0.71832145853691	0.71828233990960	0.71828185011209

Exact value: 0.71828182845905

Table 1.3: Romberg's method applied to the integral of x^2e^x on the interval $[0, 1]$. The exact value of the integral is $e - 2$. The leftmost column gives the total number of function evaluations used to obtain the trapezoid rule estimate in the second column. When comparing the table values to the exact one, we find that even the first entry obtained from Richardson's extrapolation, $R_{1,1}$, is slightly more accurate than the fourth composite trapezoid rule evaluation, $R_{0,3}$. The final answer $R_{3,3}$ is accurate to 7 decimal places, compared to only 1 decimal place for $R_{0,3}$.

refinement, as per (1.21).

Romberg's method was implemented in APD closely following the algorithm presented in Section 4.3 of [PTVF92]. Rather than explicitly building the Romberg approximations as illustrated in Table 1.2, this algorithm makes use of a polynomial interpolation scheme based on Neville's algorithm. Table 1.2 is in fact identical to the Neville's algorithm table (Table 1.1) that one obtains if one treats the set of composite trapezoid rule approximations $R_{0,k}$ as a function of relative step size squared, and then extrapolates to a step size of $h = 0$. (This derivation of Romberg's method is followed in [SB02], as opposed to the approach taken in this section). Rather than comparing previous estimates to determine the error, the algorithm in [PTVF92] uses the error estimate provided by the interpolation routine.

1.3.4 Gaussian Quadrature

In Section 1.3.2, we considered only Newton-Cotes quadrature formulas, where the spacing between the evaluation points (the abscissae) was equal. Given $n + 1$ abscissae on the interval, the only choice was which weight to use for each the function evaluation. With these $n + 1$ degrees of freedom, it was only possible to achieve a quadrature formula with

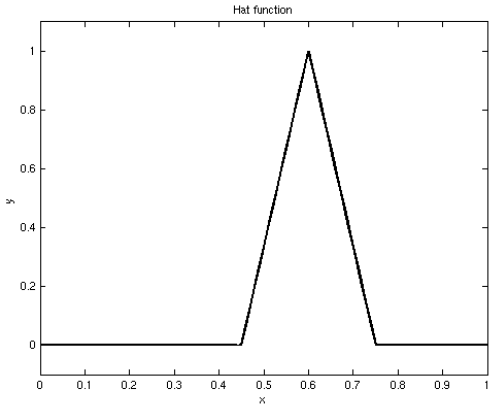

Figure 1.8: Plot of hat function given by (1.23)

degree of precision n or $n+1$.[2]

Suppose instead that the abscissae no longer need to be equally spaced, and can be chosen freely. We now have $2n+2$ degrees of freedom, as we can choose both the locations of the abscissae and how to weight the function values at those points. As might be expected, this allows us to achieve a degree of precision of $2n+1$. However, we still must determine how to choose the weights and abscissae to achieve this degree of precision.

One method of doing so is to generate a system of equations by enforcing that the quadrature formula be exact for polynomials up to degree $2n+1$; in particular, we would require that

$$\begin{aligned} \sum_{i=0}^{n} w_i x_i^p &= \int_a^b x^p \, dx \\ &= \frac{b^{p+1} - a^{p+1}}{p+1} \end{aligned} \tag{1.24}$$

for powers p up to $2n+1$. This would give a nonlinear system of equations for the weights w_i and abscissae x_i, which we could then solve to obtain the correct values. This algebraic approach is quite cumbersome, however, and it is preferable to take a more analytical approach to the problem, such as the one found in [SB02], Section 3.6.

Consider the special case of integrating a function on the interval $[-1, 1]$. The goal is to

[2]In particular, if n is odd then we can only achieve degree of precision n, but for even n the degree of precision is $n+1$. See [Ral65], pp. 116-117 for details.

0.0000					
0.1667	0.2222				
0.0833	0.0556	0.0444			
0.1458	0.1667	0.1741	0.1761		
0.1458	0.1458	0.1444	0.1440	0.1438	
0.1497	0.1510	0.1514	0.1515	0.1515	0.1515

Exact value: 0.1500

Table 1.4: Romberg's method applied to the hat function (1.23). Romberg's method provides no improvement to the convergence, since it assumes error behaviour that is only applicable for smooth functions.

find w_i and x_i such that

$$\int_{-1}^{1} p(x)\,dx = \sum_{i=0}^{n} w_i p(x_i) \tag{1.25}$$

for any polynomial $p(x)$ of degree $2n+1$ or lower. On the interval $[-1,1]$, there exists a well-known orthogonal basis for the space of polynomials, namely the Legendre polynomials [Hoc64]. The Legendre polynomials are a set of polynomials of increasing degree such that

$$\int_{-1}^{1} p_i(x)p_j(x)\,dx = \begin{cases} \frac{2}{2i+1} & \text{if } i=j \\ 0 & \text{otherwise} \end{cases} \tag{1.26}$$

The first few Legendre polynomials are $p_0(x)=1$, $p_1(x)=x$, $p_2(x)=x^2-\frac{1}{3}$. Subsequent terms are generated using a recursive relationship. The Legendre polynomial p_n has n real zeros on the interval $[-1,1]$.

Now, since $p(x)$ is a polynomial of degree $2n+1$ or lower, it can be written as

$$p(x) = p_{n+1}(x)q(x) + r(x) \tag{1.27}$$

where p_{n+1} is the $n+1^{\text{th}}$ Legendre polynomial, and $q(x)$ and $r(x)$ have degree n or lower. Furthermore, since the Legendre polynomials form a basis for the space of polynomials, we can write

$$q(x) = \sum_{k=0}^{n} \alpha_k p_k(x) \qquad\qquad r(x) = \sum_{k=0}^{n} \beta_k p_k(x) \tag{1.28}$$

for some real coefficients α_k, β_k. By (1.27) and the orthogonality of the Legendre polynomials (1.26), it thus follows that

$$\begin{aligned}\int_{-1}^{1} p(x)\,dx &= \int_{-1}^{1} p_{n+1}(x)q(x)\,dx + \int_{-1}^{1} r(x)\,dx \\ &= 0 + \int_{-1}^{1} r(x)p_0(x)\,dx \\ &= 2\beta_0 \end{aligned} \tag{1.29}$$

The left-hand side of (1.25) thus simplifies to $2\beta_0$. Now, consider the summation on the right-hand side. If we let the x_i be the $n+1$ zeros of p_{n+1}, substituting (1.27) into the summation gives

$$\begin{aligned}\sum_{i=0}^{n} w_i p(x_i) &= \sum_{i=0}^{n} w_i r(x_i) \\ &= \sum_{i=0}^{n} w_i \sum_{k=0}^{n} \beta_k p_k(x_i) \end{aligned} \tag{1.30}$$

We now force the w_i to satisfy a linear system of equations. In particular, we require

$$\sum_{i=0}^{n} w_i p_k(x_i) = \begin{cases} 2 & \text{for } k = 0 \\ 0 & \text{for } k = 1 \ldots n \end{cases} \tag{1.31}$$

It can be shown (see [SB02]) that this system has a unique solution. If the w_i satisfy this system, (1.30) then gives

$$\sum_{i=0}^{n} w_i p(x_i) = 2\beta_0 \tag{1.32}$$

From (1.29) and (1.32), it follows that choosing the x_i to be the roots of the Legendre polynomials, and w_i to satisfy the system (1.31) satisfies the desired condition (1.25). Furthermore, one can show that this choice of weights and abscissae is the only choice that satisfies this condition (cf. [SB02]). The values of x_i and w_i for increasing values of n have been extensively tabulated (see [Hoc64], for instance). As well, [PTVF92] provides algorithms for calculating these values.

This choice of weights and abscissae is the most widely-used type of Gaussian quadrature, known as Gauss-Legendre quadrature. Other types of Gaussian quadrature also exist. In particular, if the integrand is well-approximated by a polynomial times some function $W(x)$,

Number of points	Value	Absolute Error
2	0.71194177424227	6.34e-3
4	0.71828176830821	6.01e-8
8	0.71828182845890	1.47e-13

Table 1.5: Gauss-Legendre quadrature applied to the integral of x^2e^x on the interval $[0, 1]$. The accuracy obtained with only 4 function evaluations is comparable to the Romberg value $R_{3,3}$ in Table 1.3, which required 9 function evaluations as well as the extra computation for Richardson's extrapolation. With only 8 function evaluations we are nearly able to obtain machine precision.

then one would want a quadrature formula which exactly integrates

$$\int_a^b W(x)p(x)\,dx$$

for polynomials $p(x)$ up to degree $2n + 1$. The correct choice of weights and abscissae can be determined in the same way as was done for the Gauss-Legendre quadrature (which corresponds to $W(x) = 1$). The main difference is that rather than using roots of Legendre polynomials, one uses a class of polynomials that are orthogonal when integrated against the function $W(x)$. Some common choices include Gauss-Chebyshev quadrature ($W(x) = (1 - x^2)^{-1/2}$) or Gauss-Jacobi quadrature ($W(x) = (1 - x)^\alpha(1 + x)^\beta$, for some constant powers α, β) [PTVF92]. In this work we only consider the most commonly-used Gauss-Legendre type. Note that to apply Gauss-Legendre quadrature, we must first translate and scale the domain of integration to the interval $[-1, 1]$. The integral is then calculated using the Gauss-Legendre formula and then rescaled to the original domain of integration.

Gaussian quadrature is a very powerful method because it can achieve high accuracy with very few function evaluations, provided that the integrand is well-approximated by a polynomial. Table 1.5 gives the results of Gauss-Legendre quadrature for the integral of x^2e^x on $[0, 1]$. When we compare to Table 1.3, we see that Gauss-Legendre quadrature is far more accurate than the composite trapezoid rule, and is even able to achieve the same accuracy as the Romberg scheme with fewer function evaluations. As with Romberg's method, however, the effectiveness of Gaussian quadrature depends to a large extent on the behaviour of the integrand. For Romberg's method, we saw in Table 1.4 that the convergence will be poor if the integrand is not smooth. Similarly, Gauss-Legendre quadrature may not give good results if the integrand is not well-approximated by a polynomial. Table 1.6 shows the result

Number of points	Value	Absolute Error
8	0.17132858080136	2.13e-2
16	0.14145299219483	8.55e-3
32	0.14896390157660	1.04e-3

Table 1.6: Gauss-Legendre quadrature applied to the hat function (1.23). As with Romberg integration, the results are not very accurate. We achieve about the same accuracy as the Romberg scheme for a given number of function evaluations.

of Gauss-Legendre quadrature for the hat function (1.23). As was the case with Romberg integration, the results are poor compared to the accuracy that was achieved for the smooth integrand in Table 1.5.

One disadvantage of Gaussian quadrature is that unlike Romberg's method, it is not particularly well suited to refinement if the first estimate proves to be unsatisfactory. For instance, if one calculates a Gaussian quadrature approximation using 8 points, none of these points can be reused in a 16-point approximation, since all the weights and abscissae are different. A modified scheme known as Gauss-Kronrod quadrature does exist, where one can increase the number of points from n to $2n + 1$, and re-use all of the n points from the first approximation. The resulting quadrature scheme only has degree of precision $3n + 1$, however, rather than the accuracy of $4n + 1$ that one would obtain from the full Gaussian scheme. Furthermore, computation of the weights and abscissae is more complicated than for straightforward Gaussian quadrature. See [Lau97] for more details.

Chapter 2

Development of Improved Quadrature for APD

Calculation of the lookup tables in APD is very time-consuming in the original implementation, with the calculation of primary, first-order and second-order lookup tables taking nearly two weeks to complete. If the camera geometry and specified energy window do not change, then this is not a serious deficiency since the lookup tables only need to be calculated once. However, the calculation time does impose limitations on the type of studies for which APD is practical. A study which involved testing different energy window configurations, for instance, would require calculating a complete set of lookup tables for each one.

While the slow calculation time is due in large part to the number of lookup table elements that need to be calculated, it may also due to inefficient numerical integration. As discussed in the previous chapter, Romberg integration assumes that the integrand is a smooth function. However, since the integrands in APD sometimes have discontinuities in the first derivative (cf. Section 1.2.3), Romberg's method will be slow to converge for these cases. Furthermore, even for smooth functions, the results in Table 1.5 suggest that Gaussian quadrature may provide better accuracy with fewer function evaluations than Romberg's method. Thus, it seems likely that Gaussian quadrature would be preferable to Romberg integration for this application. Even though Gaussian quadrature does not perform very well for nonsmooth integrals either (as illustrated in Table 1.6), its performance is certainly on par with Romberg's method.

As such, it makes sense to consider Gaussian quadrature as an alternative to the Romberg

scheme that was used in the original implementation. However, before implementing Gaussian quadrature in APD, it is useful to run some simple numerical experiments to see how it compares to Romberg quadrature, as well as several other methods. We then discuss the implementation of Gaussian quadrature in APD, looking specifically at each of the three lookup tables.

2.1 Comparison of Numerical Quadrature Methods

In this section we compare several numerical quadrature routines in MATLAB with some integrands which mimic those found in APD. We include both the straightforward Romberg and Gaussian integration methods discussed in the previous chapter, as well as some more sophisticated integration methods that have already been implemented in MATLAB. Since the goal of this work is to improve the speed of the numerical quadrature in APD while maintaining comparable accuracy, we compare the computation times of these quadrature routines, while setting parameters to achieve roughly the same accuracy. The integration routines tested are the following:

- Romberg integration based on explicit construction of Table 1.2
- Gauss-Legendre quadrature
- Adaptive Simpson quadrature, using the MATLAB method `adaptsim` from [GG00]
- Adaptive Lobatto quadrature, using the method `adaptlob`, also found in [GG00]
- MATLAB's `quad` method, which is largely based on the `adaptsim` method.

The first two methods were run for a fixed number of function evaluations. The latter three are adaptive methods, which refine the integral until a desired tolerance is reached. These tolerances were set so as to achieve similar accuracy to that obtained with the first two methods. It is worth noting that Lobatto quadrature is very similar to Gaussian quadrature in that it allows (almost) free choice of abscissae and weights in order to maximize the degree of precision of the method. The key difference is that Lobatto quadrature requires that both endpoints of the interval be included as abscissae. As a result, the degree of precision of a Lobatto quadrature formula is two less than that of the corresponding Gaussian quadrature formula. Since these two methods are quite similar, we would expect that the method

Method	Tolerance	Average Rel. Error	Maximum Rel. Error	Average runtime (ms)	Average Function Evaluations
Romberg	–	0.00213	0.0475	2.15	33
Gaussian	–	0.00230	0.0394	**0.14**	**32**
`adaptsim`	1.0e-3	0.00239	0.8460	1.92	30.8
`adaptlob`	1.0e-2	0.00196	0.0366	1.44	52.8
`quad`	6.25e-5	0.00181	0.0387	1.57	25.1

Table 2.1: Comparison of quadrature methods for integrating hat functions

`adaptlob` will give some idea of the efficiency of Gaussian quadrature when used in an adaptive way (using Kronrod rules, as discussed at the end of Section 1.3.4).

In the first experiment, hat functions of the form (1.23) were generated with randomly generated heights, widths and peak locations on the interval $[0, 1]$. These functions mimic the first-derivative discontinuities present in APD integrands at the detector level, as described in Section 1.2.3. Since the true value of the integral can be determined easily, we can compare the numerically calculated values to the true value for each method to check its accuracy. The runtime of each method was also measured, as well as the total number of function evaluations used for each integral.

One thousand random hat functions were generated and numerically integrated. The following parameters were uniformly randomized:

- Location of the peak between 0.1 and 0.9
- Height of peak between 0.3 and 1.0
- Total width of function between 0.1 and 0.6

The results are summarized in Table 2.1.

Based on this experiment, Gaussian quadrature is definitely very competitive with the other methods. For functions of this type, it obtains comparable accuracy to Romberg's method with roughly the same number of function evaluations. The only method which requires significantly fewer function evaluations on average is `quad`. When comparing the actual runtimes, however, Gaussian quadrature outperforms all of the other methods by a considerable margin. It is roughly 10 times faster than the next-fastest method, and 15 times faster than Romberg's method. This dramatic difference in calculation time may not necessarily carry over to APD (which is coded in C rather than MATLAB, and which has

some additional calculational overhead); however, it does suggest that Gaussian integration is very computationally efficient. Furthermore, it also appears to be fairly reliable, as the maximum relative error that it attained in this experiment is on par with the best results. (Note that the max error for the `adaptsim` method is much larger than any of the others – in some pathological cases where the hat function is quite narrow, this method terminates despite having "missed" a large part of the integrand).

When comparing the methods for several smooth functions, we found that Gaussian quadrature is often able to achieve higher accuracy than the other methods, using fewer function evaluations. In addition, the calculation time is also significantly faster, as was the case in the first experiment.

It is important to note a key difference between the Gaussian quadrature method used here and the adaptive methods. The Gaussian quadrature method uses a fixed number of function evaluations, and as a result there is no guarantee of obtaining a desired error. In fact, without any other estimates of the integral, there is no way of even estimating what the error in the approximation is. The three adaptive methods, on the other hand, will refine the estimate until a desired error tolerance is reached, taking more function evaluations if necessary. While a fixed number of points was also used for Romberg integration in these experiments, Romberg's method is also usually implemented to refine the approximation until a desired tolerance is reached (cf. Section 1.3.3), which was the case in APD.

Given the large size of the integration problem in APD, however, a quadrature method that does this type of refinement may not be efficient. Calculating a full set of primary, first-order and second-order lookup tables in APD requires tens of billions of integration operations. If each one of these integration steps requires refinement, there will be several recursive calls for each integration step as well. These recursive steps add a significant amount of overhead to each integration call, especially if the integral is slow to converge. Thus, using Gaussian quadrature with a fixed number of points may be a preferable method for APD lookup table calculation. This method can be implemented with very little overhead, as it simply requires evaluating the function at the specified points, multiplying by the appropriate weights and then summing up the resulting values. Furthermore, the experimental results suggest that Gaussian quadrature with a fixed number of points can be trusted to give results with comparable accuracy to other quadrature methods, if enough points are used. So, if we can determine how many points are "enough", then we can be reasonably sure that Gaussian quadrature will provide accurate results in much faster time

than any of the adaptive methods. While this method does not give any idea of the error in the approximation, some uncertainty in the accuracy of the calculated probabilities is acceptable, since the probabilities in APD are not exact anyways, but the result of a number of simplifying assumptions.

The goal is then to develop a fast Gaussian quadrature routine for lookup table calculations in APD. This routine should give comparable results to those obtained with the Romberg scheme (which were experimentally validated in [Wel97] and [Van04]), but take significantly less time to compute. A runtime on the order of hours rather than days is desired. The main problem is finding the acceptable tradeoff between accuracy (i.e. the number of quadrature points to use) and runtime. We will take a somewhat experimental approach to this problem, and will also make use of some observations about the integrands from Section 1.2.3.

2.2 Primary Lookup Table

Primary lookup table calculation is not a time-consuming calculation in APD. Even using the Romberg scheme of the original implementation, the calculation only takes several seconds to complete. This is due in large part to the fact that the primary table contains only about five thousand elements, since a primary photon can only be detected in a small subset of detector pixels on the camera head due to the collimator. Furthermore, the integral is only two-dimensional, making each element fast to compute. Nonetheless, the primary lookup table is a good starting point to see how Gaussian quadrature compares to Romberg integration for the actual APD calculation. The primary lookup table calculation consists of evaluating the two-dimensional area integral in Equation (1.4).

In this section and all subsequent ones, we are computing lookup tables for Technetium-99m (initial energy of 140 keV), using an energy window of 130-150 keV. The physical parameters are set to model a Philips VXGP collimator. As mentioned in the previous chapter, APD uses $64 \times 64 \times 64$ activity and attenuation maps, and a 64×64 detector grid. All calculations were done on a machine with dual 3.6 GHz Pentium 4 processors and 2 GB of RAM.

Using 16 Gaussian quadrature points for each dimension gave a primary lookup table that showed fairly good agreement with the original table. Only about 3% of the non-zero entries differed by more than 2% relative difference. Furthermore, those entries that did

differ by more than 2% were all on the order of 10^{-7} or smaller, which is about 100 times smaller than the median value of the table. Creation of the new table was about ten times faster than the old one as well, although the difference was not noticeable in realtime.

Accuracy is especially important for the primary table, however, since primary photons are the most significant contribution to SPECT data. Since runtime was not an issue, we increased the number of quadrature points to 32 for each dimension to try to improve the accuracy further. Doing so gave a table that was virtually identical to the one obtained with Romberg integration. The only entries differing by more than 2% were on the order of 10^{-11} or smaller, i.e. essentially zero.

While the change to Gaussian quadrature has not improved the speed of the primary lookup table calculation appreciably, it does show that Gaussian quadrature is at least competitive with Romberg's method in the APD calculation itself.

2.3 First-Order Lookup Table

Due to the large number of elements and the higher dimension of the integration problem, it is not possible to use as many quadrature points for the first-order table as for the primary table. Even using just 16 points for every dimension results in a calculation time of several days. Halving this to 8 points for every dimension produces an acceptable runtime of several hours, but the table is not sufficiently accurate for cases where the scattering point is close to the detector surface. It is difficult to accurately approximate the integral with a small number of points in this case, since the integrand is only nonzero on a small part of the detector surface. Using 8 points does, however, provide results that match up well with the original table for values where the scattering point is far away from the detector surface. We therefore use this Gaussian quadrature scheme as a starting point for the development of our method.

The first-order lookup table (Equation 1.6) is separated into a number of smaller files, split up based on the distance from the scattering point to the detector (parameter (a) in Figure 1.4). We will go through each of these files and compare the values in the original APD lookup table (calculated using Romberg's method) to those in the new table using Gaussian quadrature. We will also compare to a Gaussian quadrature-based table using 16 points for every dimension. As mentioned, the runtime to create this table is much longer than the target runtime. That said, it does provide a good benchmark for the highest

accuracy that we can reasonably expect to obtain with Gaussian quadrature.

We first compare the Gaussian quadrature-based table using 8 points to these two benchmarks. In particular, we will look at the median and average relative difference between table entries, as well as how many of the entries differ by more than 5% relative difference. The relative difference is calculated as the absolute value difference between the entries, divided by the size of the entry in the original table. Since the lookup tables contain a large number of zero entries (anywhere from 45% to 80%, depending on the distance of the scattering point from the detector), we compute these statistics only for those entries that are non-zero, to avoid skewing the statistics.

Figure 2.1 plots these statistics, as a function of scatter-detector difference, when comparing against the original Romberg-calculated table. The plot is split into two parts, since the differences between the two tables are quite significant when the scattering point is close to the detector. We can see that for distances of 10 cm or less, the tables do not agree with one another very well at all. The majority of table entries differ by more than 5%, and the differences can even average more than 100% in some cases. Between 10 and 30 cm, there is some fluctuation in the statistics, but on the whole only about 5 to 10% of entries differ by more than 5%, and the average difference varies in about the same range. The median difference is always less than 1%. Finally, from 30cm onwards, the statistics are fairly consistent, with about 5% of table entries differing by more than 5%, and the average difference sitting around 2-3%. The median is about one to two-tenths of a percentage point.

The agreement between the 8-point and 16-point Gaussian quadrature-based tables is considerably better, though still poor for short distances. For distances of about 10 cm or more, however, we find that just over 1% of table entries differ by more than 5%, and on average entries only differ by 0.5% or less. There is a sharp increase in the number of elements differing and the size of the differences around 22 cm, which is difficult to explain. Certainly there does not seem to be any physical reason that would account for poorer agreement between table entries at that distance. Furthermore, examining the lookup table entries directly and comparing against nearby values (i.e., the entries for distances slightly less and slightly more than 22 cm) does not indicate any aberrant behaviour to the naked eye. This large discrepancy may be an artifact of the specific matrix and pixel size that was used in this calculation. In any event, the difference only reaches about 5% of entries differing by more than 5%, which was the norm when comparing against the Romberg table. Since the total number of lookup table entries is so great, it is unlikely that this "spike" in

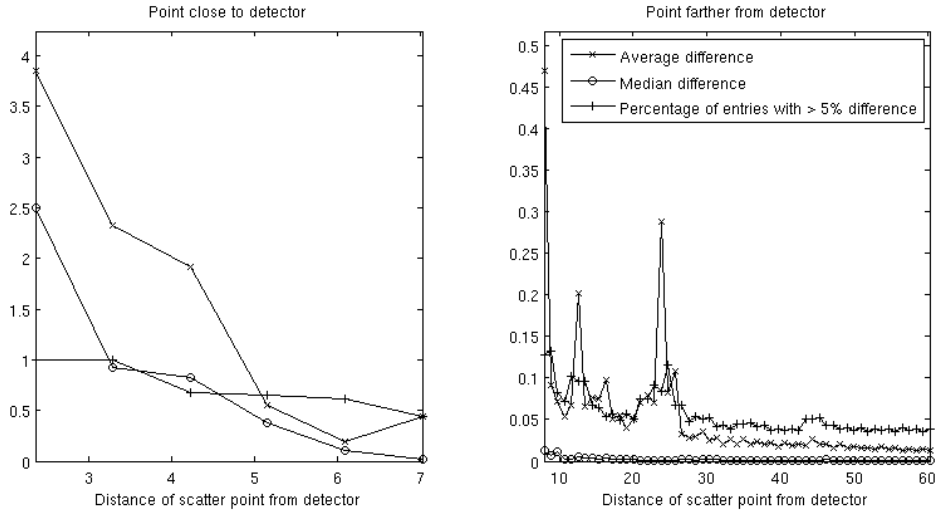

Figure 2.1: Plot of differences between first-order lookup tables calculated using 8-point Gaussian quadrature and Romberg's method, as a function of distance of the scattering point from the detector surface.

the difference between the table values for this distance will result in a noticeable effect on the final PSF calculation.

Figure 2.2 gives a plot of the statistics for 8-point vs. 16-point Gaussian quadrature. While it is clear that the 8-point and 16-point Gaussian quadrature-based tables agree with each other better than the 8-point Gaussian and Romberg tables, it is not entirely clear what the implications are. Without knowing the exact values of the integrals, it is not possible to say which of these tables is in fact the most accurate. It could be that the 16-point Gaussian table is the most accurate, in which case it is encouraging that the 8-point Gaussian table agrees with it quite well in most cases. It could also be that Romberg table is the most accurate, which would be less encouraging because the agreement is not as good. Or it could be that neither of the two benchmarks is very accurate, and the 8-point Gaussian table simply agrees better with the 16-point table.

What is very obvious, however, is that the Gaussian quadrature method is far faster than Romberg's method. The 8-point table takes only 4 hours to compute, a vast improvement over the runtime using Romberg's method, which was roughly 10 days. Since speed is the primary motivating factor in this application, this weighs more heavily than the fairly small

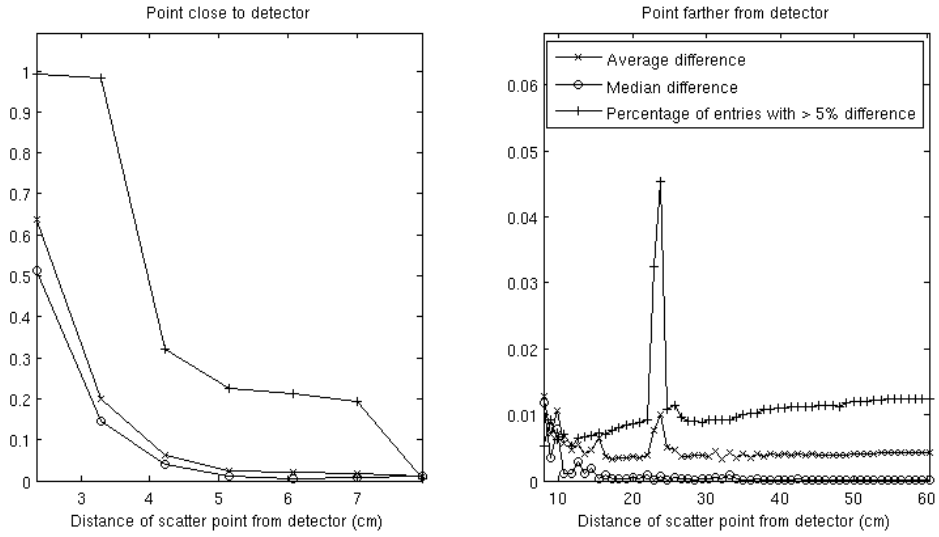

Figure 2.2: Plot of differences between first-order lookup tables calculated using 8-point Gaussian quadrature and 16-point Gaussian quadrature, as a function of distance of the scattering point from the detector surface.

deviations between the Romberg table and the Gaussian-quadrature based tables. Having accepted that Gaussian quadrature is preferable to Romberg integration, we will design our quadrature scheme with the intention of matching up with the 16-point Gaussian table as well as possible. For scatter-detector distances of about 10 cm or more, the agreement between the tables is already acceptable. So, the problem becomes getting better agreement for the shorter distances, without significantly increasing the runtime.

It may be possible to achieve a good tradeoff between speed and accuracy for these short distances by finding some value between 8 and 16 points per level, but this may not be efficient. Going from 8 to 12 points, for instance, would increase the runtime by $1.5^5 \approx 7.6$ times, since the integral is five-dimensional. Instead of changing the number of points at every level, it may make sense to use fewer points when the integrand is smooth, and more if it is not. In particular, as noted in Section 1.2.3, the integrand will have smoother behaviour when integrating over the voxel volume than over the pixel area. Thus, we may be able to use a small number of points at those last three levels of integration without losing too much accuracy.

So, we try a scheme that uses 16 points for the first two levels of integration (the detector

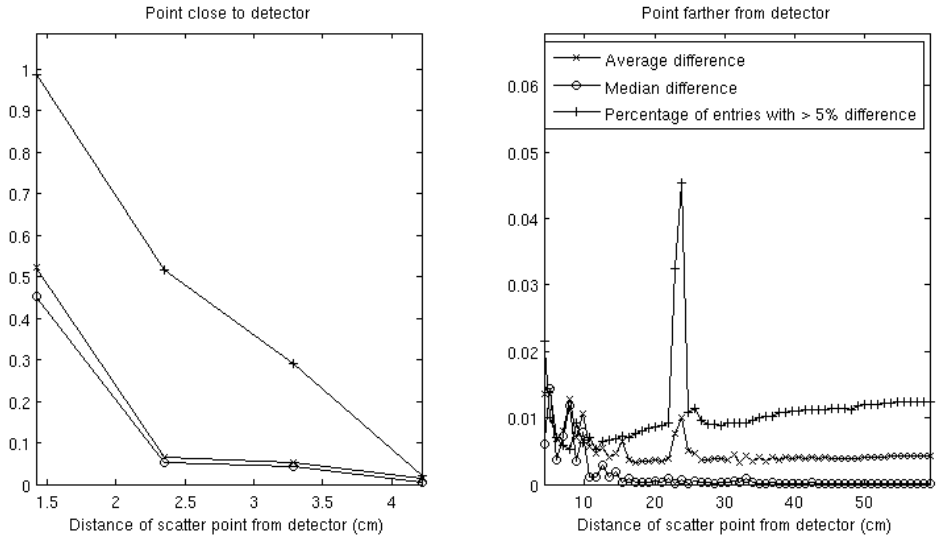

Figure 2.3: Plot of differences between first-order lookup tables calculated using the final quadrature scheme and 16-point Gaussian quadrature, as a function of distance of the scattering point from the detector surface. The final scheme uses 8 points at every level for scatter-detector distances greater than 10 cm, and switches to using 16 points for the lowest two dimensions when the scattering point is closer.

level) and 8 points for the last three. This will increase the runtime by a factor of four over the 8-point method, so we only want to use it when necessary, i.e. for small scatter-detector distances. For larger distances, we will use 8 points at every level. With the new scheme, we find that we now get good agreement with the 16-point table for short distances down to about 4 cm, as shown in Figure 2.3. The values for closer distances still do not match up very well, but this will probably have a negligible effect on the actual APD calculation itself. The reason is that these table entries will be rarely used, since there is seldom any scattering material that close to the detector surface in an actual SPECT scan. Thus, we simply accept that the integrals will be difficult to accurately calculate in these cases, and do not seek to improve the accuracy here any further. The final Gaussian quadrature scheme that we use for the first-order table will be 16-16-8-8-8 for scatter-detector distances of about 10 cm or less, and use 8 points at every level for larger distances. The full set of first-order lookup tables takes about 6 hours to calculate using this new scheme.

2.4 Second-Order Lookup table

The process to construct the second-order lookup table consists of two steps. Numerical integration takes place in the first step, where the volume integral component of the $K'_{st_i n_c}$ term (1.8) is computed. These values are stored in an intermediate table. The second step completes the calculation of the $K'_{st_i n_c}$ terms by multiplying values from this table and the first-order lookup table and summing them according to the formula. Unfortunately, the second step of this process is the most time-consuming, so improving the speed of the first part will not dramatically improve the overall runtime. In particular, the first step requires only about 6 hours to compute with Romberg's method, while the summation takes over 60 hours.

That said, there were some other concerns about the second-order scatter calculation in APD, aside from the time needed to create lookup tables. In [Van04], the photon distributions generated by APD were compared to simulated data from SimSET, a Monte Carlo-based program which simulates SPECT scans [HHG+93]. The author comments in Appendix A that

> "there are discrepancies in the individual components from primary photons, first and second order Compton scattered photons. In particular, there is a large systematic difference between SimSET and APD in the relative magnitude of the second order Compton scatter distribution ... For SimSET 74% of the total counts come from primary photons, 22% from first order Compton scatter, and 4% from second order. For APD the percentages are 78% primary, 20% first order, and 2% second order."

In other words, the amount of second-order scatter calculated by APD was significantly less than the amount calculated by SimSET. SimSET does make its own approximations and assumptions which may differ from those used in APD, so perfect agreement between the two methods is certainly not expected. (SimSET data also contains noise due to the random nature of the calculation, but even in the absence of this noise one would still expect some differences due to the differing assumptions). Nonetheless, the discrepancy in the amount of second-order scatter calculated is significant enough to be of some concern. Some possible explanations were suggested for the discrepancy in [Van04], but no satisfactory conclusion was reached as to the cause. One possible cause that was not mentioned is the presence of errors in the second-order lookup table. An examination of the intermediate table of volume

integrals reveals that some entries have negative values. Since the integrands in APD are always non-negative, this is clearly an error. Although the number of negative entries is not large (roughly 0.1% of non-zero table entries), the magnitudes of many of these negative entries is significant compared to the other entries in the table. Thus, they could reduce the size of the final second-order lookup table entries when included in the summation step. Furthermore, the presence of these negative entries may also indicate some problem with the integration method (or its implementation) that could make other table entries unreliable as well.

Conclusively tracking down the cause of these errors is difficult, due to the large number of calculations done for even a single one of the volume integrals. It is clear, however, that the negative values arise as a result of the extrapolation step in Romberg's algorithm. While the extrapolation should not produce any negative values in theory, it appears that numerical error results in some small negative values arising at the lowest dimension of integration. These negative values are then magnified in the subsequent two levels of integration, resulting in some large negative values in the end result. Switching to Gaussian quadrature will ensure non-negativity, since the method simply sums up a series of non-negative values with no extrapolation.

The completed second-order lookup table is parameterized in the exact same way as the first. To determine an appropriate Gaussian quadrature scheme, we will again compare lookup tables to both the original second-order table and a benchmark table generated with Gaussian quadrature. In this case, the benchmark table is generated using 32 points at each of the three dimensions in the initial integration step. As in the first-order case, we start with 8 points at every level.

The 8-point Gaussian table shows poor agreement with the Romberg-calculated table, particularly for scatter-detector distances of 10cm or less. After this point there is a fairly consistent trend of about 30% of entries differing by more than 5% relative difference, and the average relative difference being about 10%. Checking the signs of these differences reveals that in most cases, the entries in the Gaussian table are larger than those in the Romberg table, so we would expect that the Gaussian-calculated lookup table will indeed result in more second-order scatter when used to calculate the PSFs.

The 8-point Gaussian table does agree fairly well with the 32-point table, regardless of scatter-detector distance. (See Figure 2.4). In most cases about 4% of table entries differ by more than 5%, and the average relative difference is around 2%. Furthermore, calculating

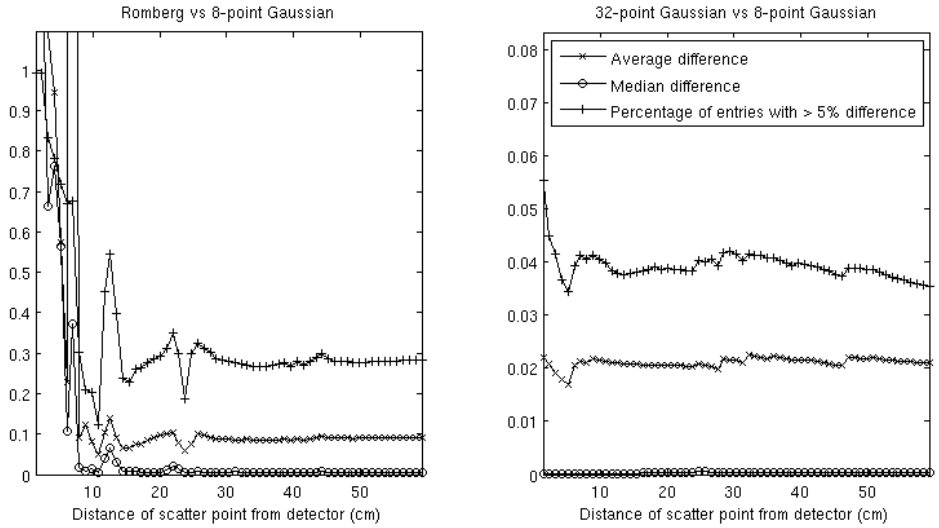

Figure 2.4: Plot of differences between second-order lookup tables calculated using 8-point Gaussian quadrature and using Romberg integration (left plot) and 32-point Gaussian quadrature (right plot). The agreement between the two tables calculated with Gaussian quadrature is fairly good, but the Romberg table differs significantly.

the table with only 8 points is very fast, taking only about 20 minutes. Increasing to 16 points at every level will increase this runtime by a factor of roughly 8, for a total of a few hours. Since this is still an acceptable runtime, we will go to 16 points to see if the accuracy is much improved.

Agreement between the 16-point and Romberg tables is also poor. As one would expect, however, the 16-point Gaussian table does agree better with the 32-point table than the 8-point table did. Entries now differ by less than 1% on average. Therefore, there does not seem to be any advantage to using 32 points to calculate the intermediate table, as it gives more or less the same results as with 16 points. Since the results are also fairly consistent for all scatter-detector distances, we will simply use a 16-point scheme at all three levels for the whole calculation, rather than switching midway through as was done for the first-order case. The time taken to calculate the intermediate table is just over 2 hours; almost a threefold improvement over Romberg's method. However, as mentioned this does not appreciably improve the runtime for second-order lookup table calculation, since the second step still takes several days.

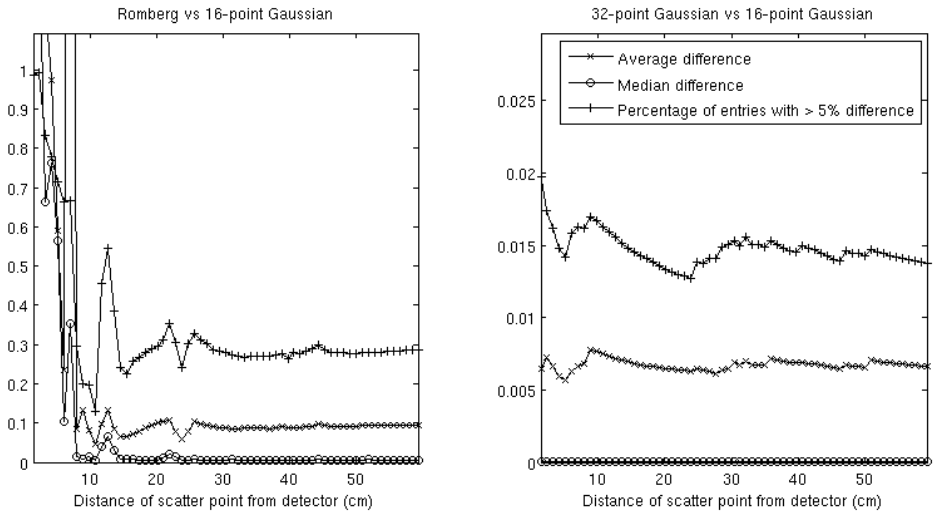

Figure 2.5: Plot of differences between second-order lookup tables calculated using 16-point Gaussian quadrature and using Romberg integration (left plot) and 32-point Gaussian quadrature (right plot). Agreement with the Romberg table is still poor, but the agreement with the 32-point Gaussian table is improved by using 16 points rather than 8.

Chapter 3

Experimental Validation

In the previous section the lookup tables were compared entry-by-entry to see how much of a discrepancy was being introduced by switching to Gaussian quadrature. While the agreement between the new and old tables was generally good, there were differences noted between them, particularly for the first and second-order scatter tables. On the whole these differences did not appear to be large, but it is difficult to judge how much of an effect they will have when incorporated into a full APD calculation. In particular, simply looking at average or median differences between table entries does not take into account that some lookup table entries may be accessed much more frequently than others when doing an actual PSF calculation, or that differences between the tables may average out during the calculation. Thus, in order to better assess the impact of the change of integration methods, we must compare actual APD PSF calculations using both sets of tables. In this section we first describe the parameters of the experiments, then the results.

3.1 Experimental Setup

We performed experiments with two different objects to compare the PSFs. In the first experiment a simple point source is placed in an elliptical water cylinder, off-centre near the edge of the cylinder. Placing the source off-centre means that the amount of scatter will change as the camera rotates, since there will be varying amounts of water between the detector and the source. Figure 3.1 shows the setup for this first experiment.

The second experiment uses the Mathematical Cardiac Torso (MCAT) phantom [PXT+97] to model a realistic human torso with activity in the heart. MCAT creates an attenuation

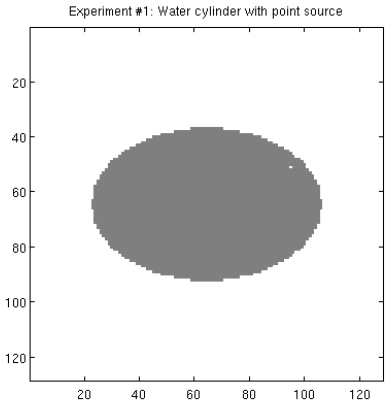

Figure 3.1: Water cylinder and point source used for first experiment. The white point inside the cylinder indicates the location of the source.

map modelling realistic tissue distributions for a human torso, including cardiac, bone, liver, muscle and lung tissue. It also permits the user to specify different levels of activity inside each one of these regions. A standard sized female torso was used for the experiment, with uniform activity in the left and right myocardia of the heart. Figure 3.2 shows three axial slices through the phantom used for this experiment.

In both experiments, we calculate the PSFs using both the old and new lookup tables. The activity and attenuation maps were also used to generate projection data in the SimSET [HHG+93] Monte Carlo simulator. In all cases the projections were acquired over 360° at 3° intervals, resulting in a total of 120 projections. Since both SimSET and APD allow us to separate the projection data into primary, first-order and second-order components, we were able to compare each of these components with one another to see how they differ. In particular, this allows us to assess

- How much of a difference is introduced into the actual calculated PSFs with the change in integration method for the lookup tables
- How the APD PSFs compare with data from SimSET. We are particularly interested in whether the amount of second-order scatter calculated is closer to that estimated by SimSET (cf. Section 2.4) when using the new lookup tables.

For both experiments, we compare the data in two ways. First, for each projection angle

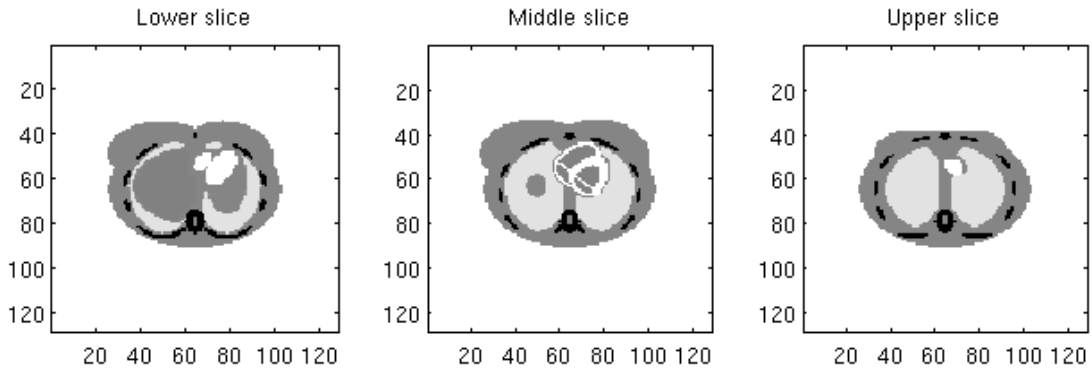

Figure 3.2: MCAT phantom used for the second experiment. The figure shows three axial slices through the phantom. Darker regions indicate more dense tissues. The activity in the left and right myocardia is shown in white.

we compare the PSFs by drawing profiles along lines through the detector bin with the most counts. This allows us to compare the spatial distribution of the counts in part of the PSF, as well as as the quantitative values. Secondly, for each projection angle, we summed up the total counts for primary, first-order and second-order scatter and calculated the ratio of each to the total number of counts. This gives an overall idea of how the calculated amounts of primary, first-order and second-order detected photons compare for the different projection angles. In this second case we also compared against the ratios given by SimSET.

3.2 Results and Discussion

In the first experiment, we look at the PSFs for two specific acquisition angles around the water cylinder – 90° and 270°. The 90° angle corresponds to having the detector on the right side of the cylinder shown in Figure 3.1, while the 270° angle is on the left side of the cylinder. In the first case, the source is fairly close to the detector with little attenuating material between them. As a result, we expect primary photons to make up the bulk of the detected photons, and for the photons to be detected in a fairly small region of the detector. In the second case, the detector is on the far side of the cylinder, with a large amount of water between it and the source. We therefore expect to see a much greater quantity of

scatter, and for the PSFs to be spread over a larger part of the detector surface.

Figure 3.3 shows the primary, first-order and second-order PSFs calculated using both versions of the lookup tables. Spatially the PSFs are virtually indistinguishable from one another. However, it is apparent from the scaling that there is a discrepancy in the magnitude of the first-order PSF. In Figure 3.4 we draw profiles through the dashed lines in Figure 3.3 to better see the difference in the values. Here we can clearly see that the peak value of the first-order PSF calculated with Gaussian quadrature is larger than the value calculated with Romberg integration. Quantitatively, the value in the pixel with the largest magnitude is about 18% higher. Overall, the quantity of first-order scatter calculated for this angle using the new lookup tables is about 4.5% more than that calculated with the old ones.

Some discrepancy in the first-order PSF is not unexpected for the 90° angle, since the source is close to the detector. Recall from Section 2.2 that it was for short scatter-detector distances that the two tables had the largest differences between them. 18% is a significant difference, however, and it is important to see whether the amount of first-order scatter calculated with the new lookup table is unusually high. Based on the data from SimSET, this does not appear to be the case. The ratio of first-order scattered photons to total photons detected for the 90° acquisition angle was 11.00% in SimSET, 10.02% with the new lookup tables, and 9.58% with the old lookup tables. So the quantities calculated with the new lookup table are actually in better agreement with those found in SimSET than those calculated with the old lookup table. It may be that the lookup table values calculated for short distances using Romberg's method were not accurate, and that APD was slightly underestimating the quantity of first-order scatter using those lookup tables.

Interestingly, the PSFs for second-order scatter essentially overlap, even though it was for second-order scatter that we observed the most significant differences between the old and new lookup tables. It is not immediately obvious why this is the case. Going back to the direct comparison of table entries (as was done in Section 2.4) we find that there are many more entries for which the new table value is larger than the old table value. However, in cases where the old table value is larger, the difference tends to be greater. Thus, it may be that the differences between the two tables end up cancelling out for the most part, resulting in two nearly identical PSFs.

The PSFs for the 270° angle (Figures 3.5 and 3.6) are characterized by a much wider distribution of photons, since the source is farther away from the detector now. As expected,

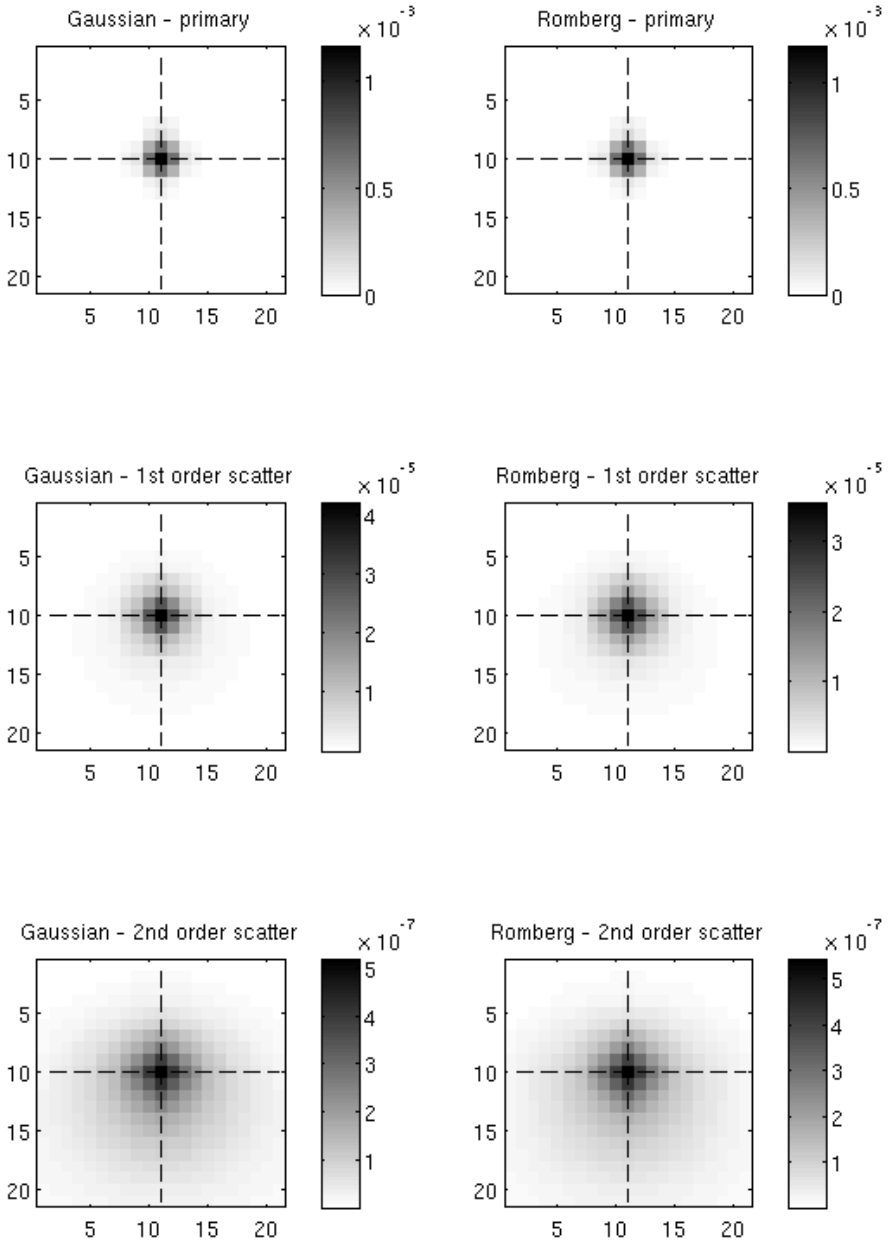

Figure 3.3: Primary, first-order and second-order PSFs for first experiment – 90° acquisition angle. The PSFs in the left column were calculated using the tables calculated with Romberg's method, and those in the right used the new Gaussian quadrature schemes. The figures are zoomed in on the region of most interest.

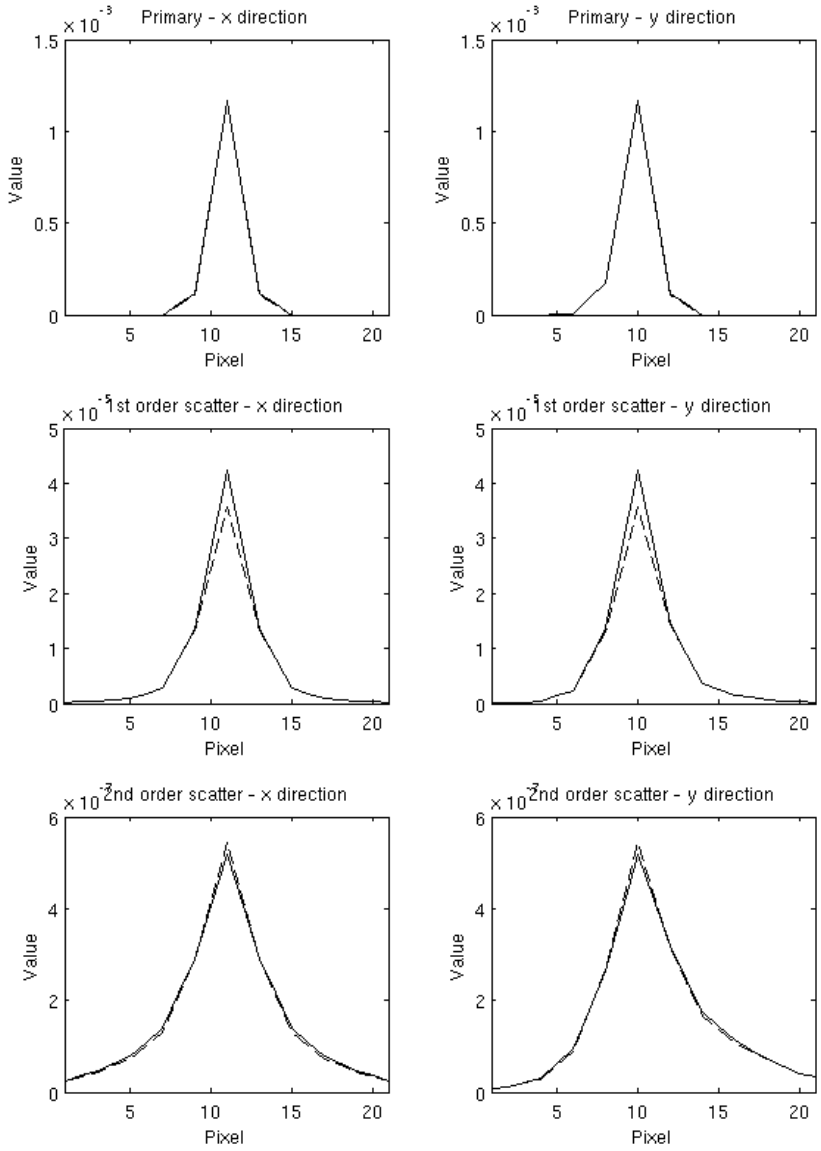

Figure 3.4: Corresponding profiles drawn along the dashed lines in Figure 3.3. The profiles drawn with a solid line are profiles for the Gaussian lookup tables; the dashed lines are for the Romberg lookup tables.

there is also significantly more scatter. The agreement between the PSFs is better in this case than for the 90° angle. There is still a noticeable discrepancy in the first-order PSF, but it is smaller than it was when the source was close to the detector. The peak value is now only about 6% higher with the new table, and the overall amount of first-order scatter calculated is less than 1% more.

Finally, in Figure 3.7, we compare the ratios of each type of detected photon to the total number of detected photons, for all 120 acquisition angles. The most obvious feature is that the amount of second-order scatter calculated by SimSET is still much higher than that calculated with APD, especially when second-order scatter becomes more prevalent when there is more attenuating material between the source and detector. The change to the lookup tables has not significantly increased the amount of second-order scatter calculated in APD. However, the ratios for the APD PSFs calculated using either set of lookup tables match up very well overall. The maximum difference in these ratios for any of the acquisition angles is 0.6%. Overall, the total ratios across all projections are:

- 71.9% primary, 22.5% first-order and 5.6% second-order in SimSET
- 74.9% primary, 22.2% first-order and 2.9% second-order in APD with new lookup tables
- 75.2% primary, 21.9% first-order and 2.9% second-order in APD with old lookup tables

So, the discrepancy between the amount of second-order scatter calculated by APD and SimSET still exists. We conclude that this discrepancy is not a result of the negative values in the intermediate second-order calculation (c.f. Section 2.4). There is some other difference in the way that SimSET and APD calculate second-order scatter that is causing this difference.

In the second experiment, there are now multiple voxels containing activity, and the attenuating medium is much more complex. We wish to see if either of these factors results in a significant difference between the PSFs calculated with the old and new lookup tables. As before, we look at the PSF for two acquisition angles – 90°, when the detector is on the right side of the phantom in Figure 3.2, and 270°, when it is on the left. Again, there should be more scatter for the 270° angle, since there is more attenuating material between the activity and the detector.

Figures 3.8 through 3.12 show results corresponding to Figures 3.3 to 3.7, for the second experiment. The main differences between the new and old results are the same as for

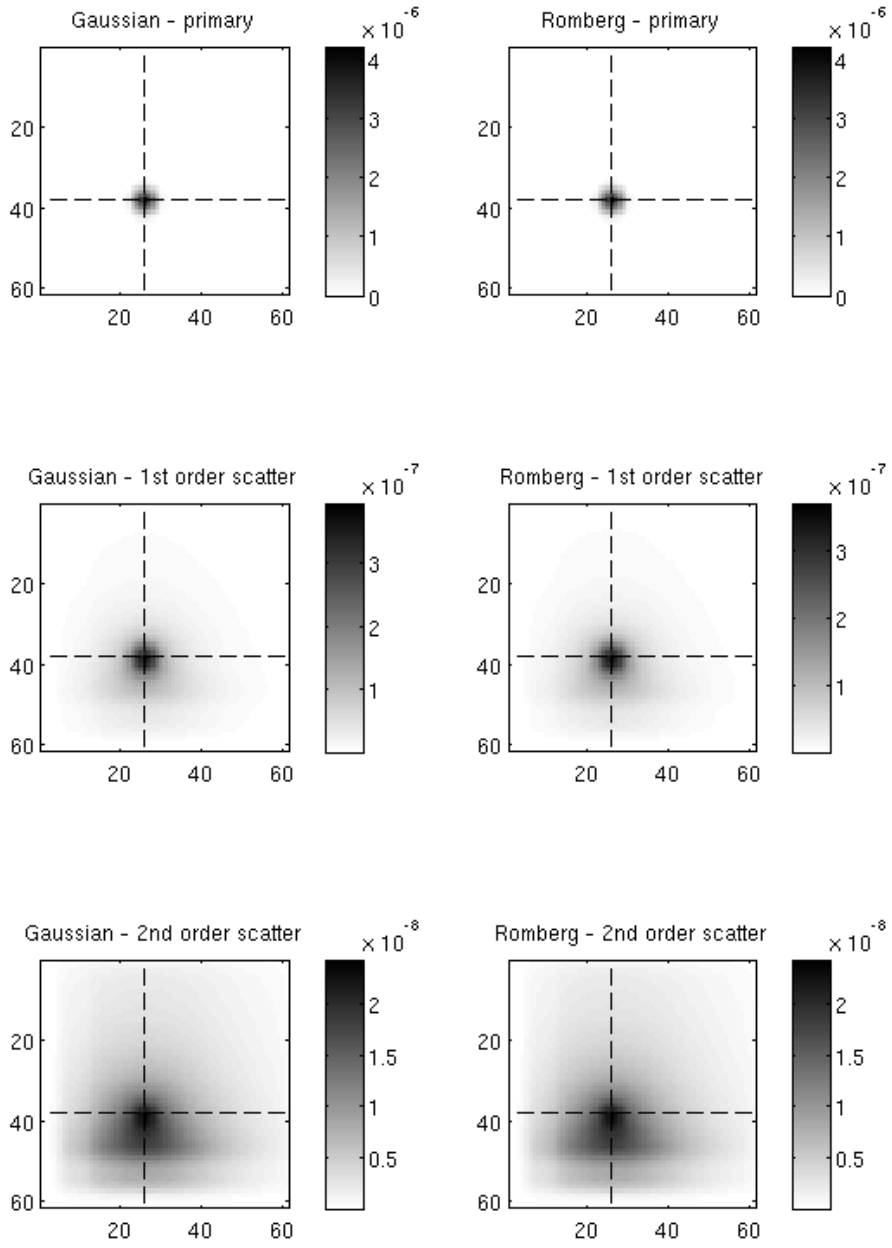

Figure 3.5: Primary, first-order and second-order PSFs for first experiment – 270° acquisition angle. The PSFs in the left column were calculated using the tables calculated with Romberg's method, and those in the right used the new Gaussian quadrature schemes. The figures are zoomed in on the region of most interest.

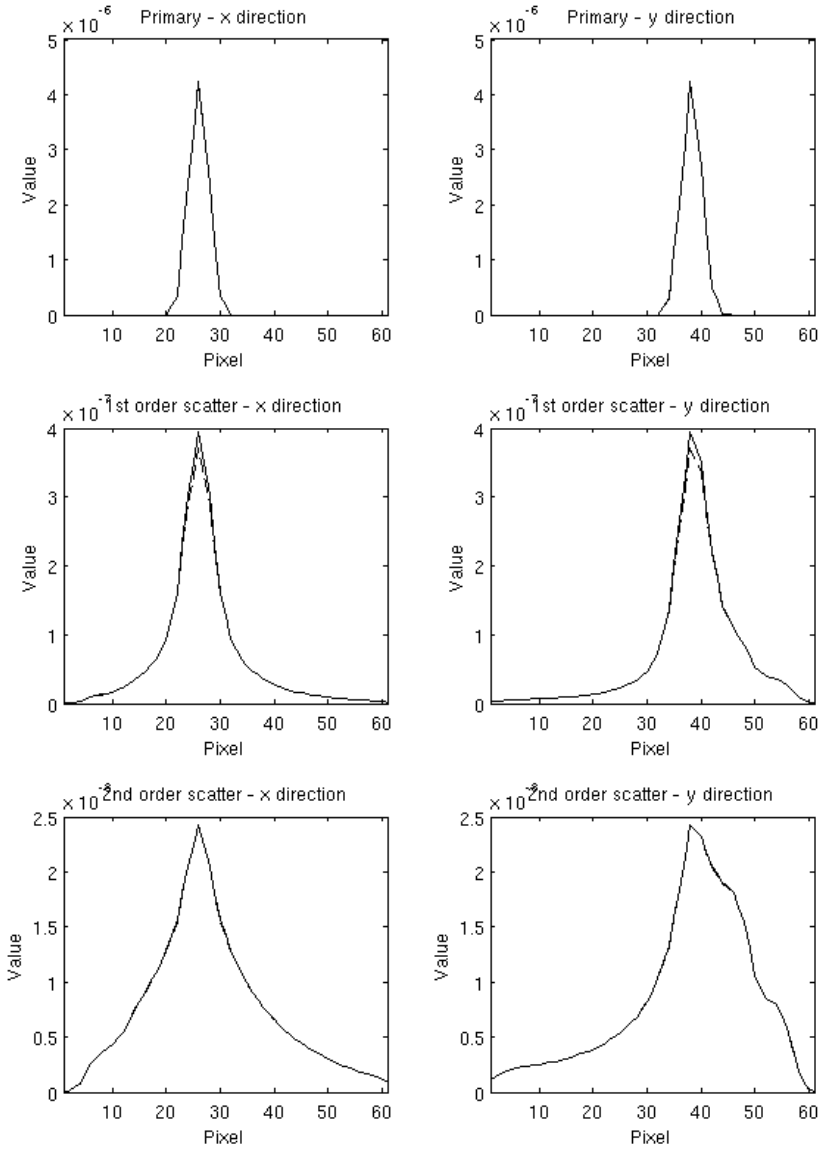

Figure 3.6: Corresponding profiles drawn along the dashed lines in Figure 3.5. The profiles drawn with a solid line are profiles for the Gaussian lookup tables; the dashed lines are for the Romberg lookup tables.

the first experiment. The most significant area of disagreement is still around the peak values for the first-order scatter PSF, where the values from the new lookup table are larger than those from the old table. The discrepancies are not as large as they were in the first experiment, however, and are usually in the range of 2-3%. Aside from this, there is very little difference in the PSFs calculated with the old and new lookup tables. As in in the first experiment, however, there is a significant difference in the amount of second-order scatter between APD and SimSET. Overall the ratio of second-order scatter is about 3.1% for the APD PSFs, and 4.95% for the SimSET data.

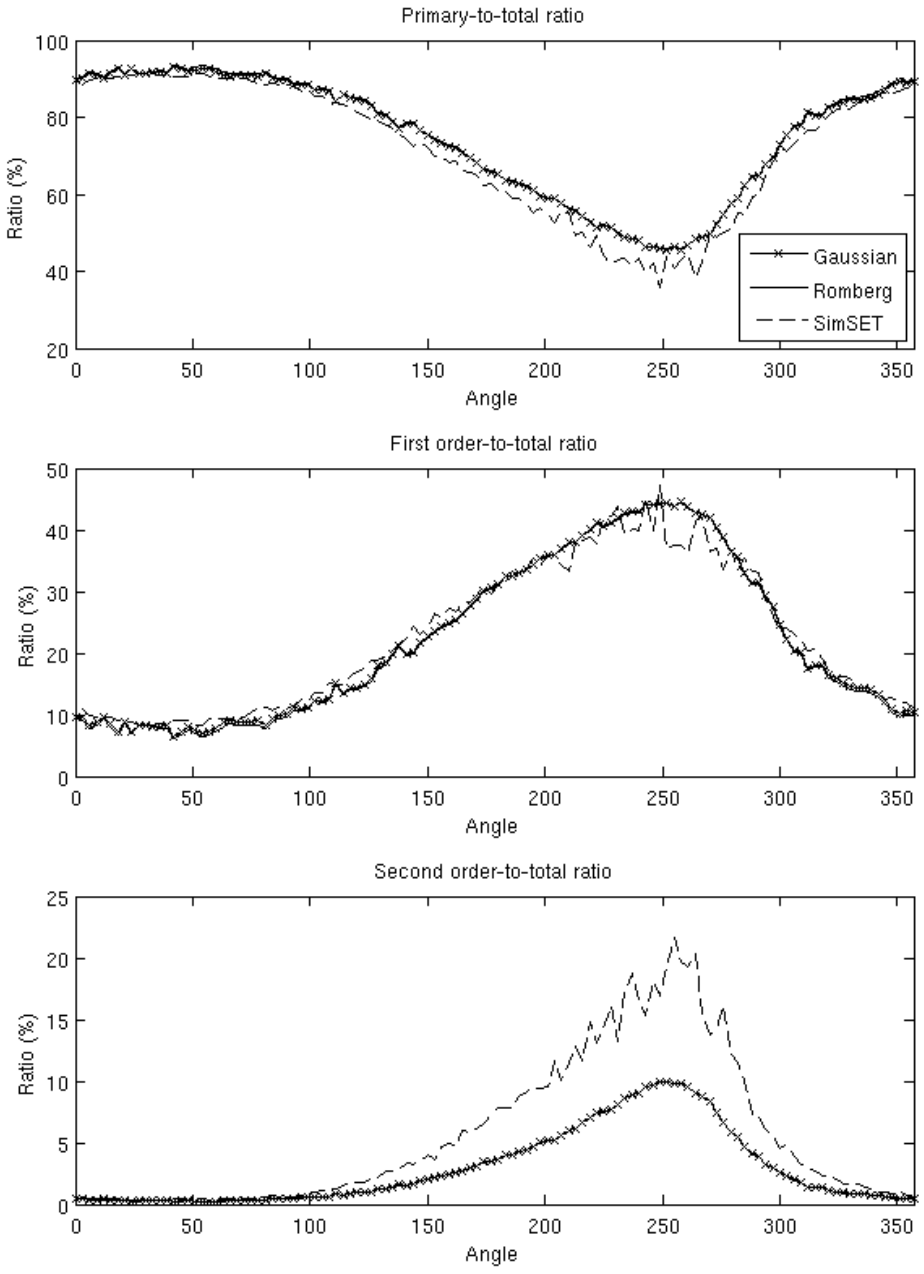

Figure 3.7: Ratios of primary, first-order and second-order photons detected to total photons detected for first experiment.

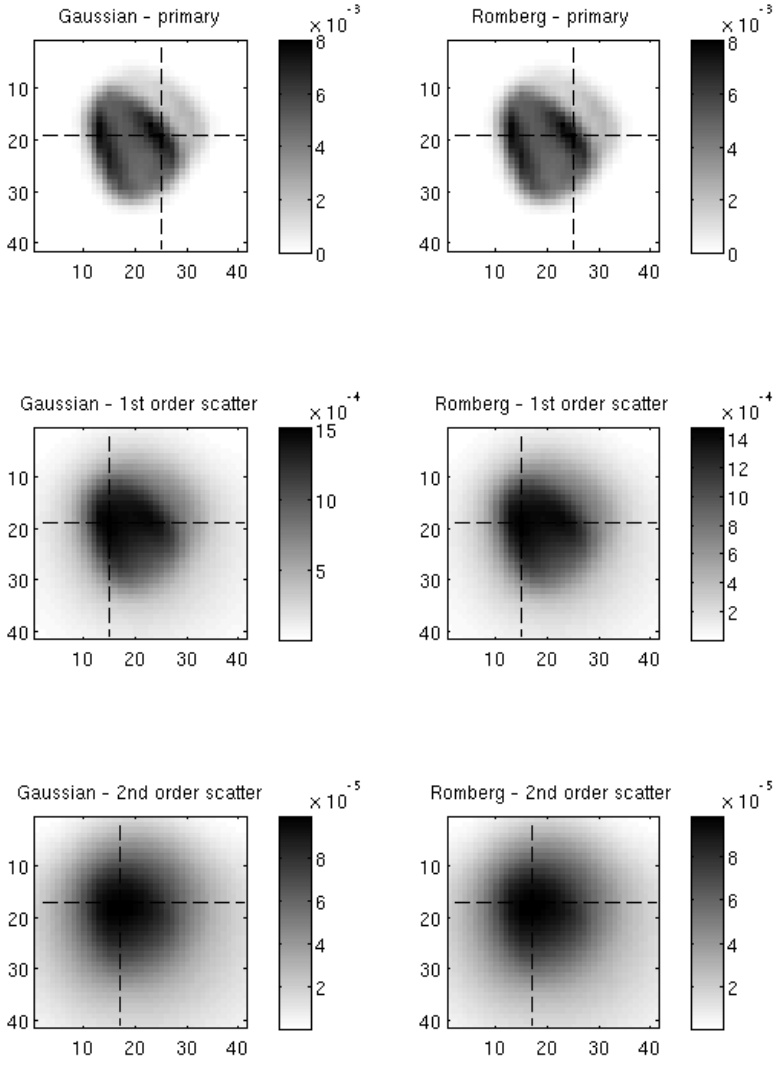

Figure 3.8: Primary, first-order and second-order PSFs for second experiment – 90° acquisition angle. The PSFs in the left column were calculated using the tables calculated with Romberg's method, and those in the right used the new Gaussian quadrature schemes. The figures are zoomed in on the region of most interest.

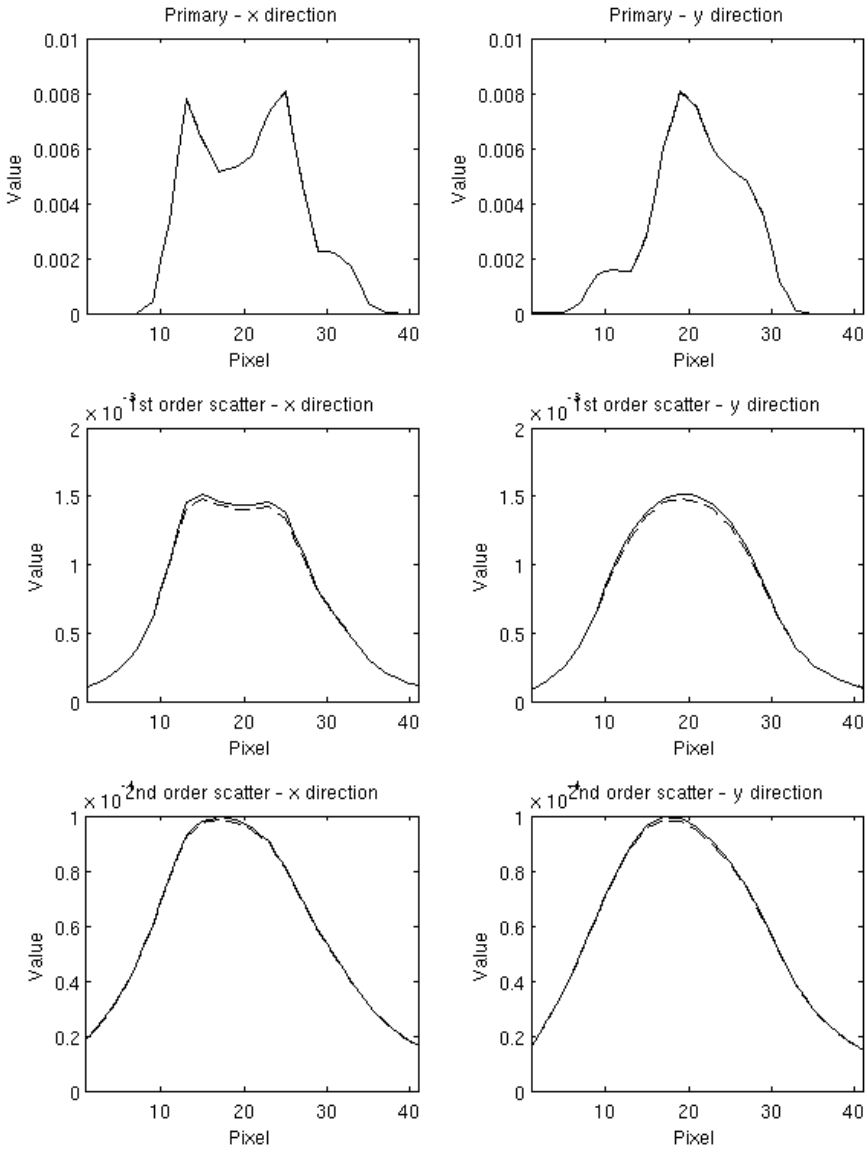

Figure 3.9: Corresponding profiles drawn along the dashed lines in Figure 3.8. The profiles drawn with a solid line are profiles for the Gaussian lookup tables; the dashed lines are for the Romberg lookup tables.

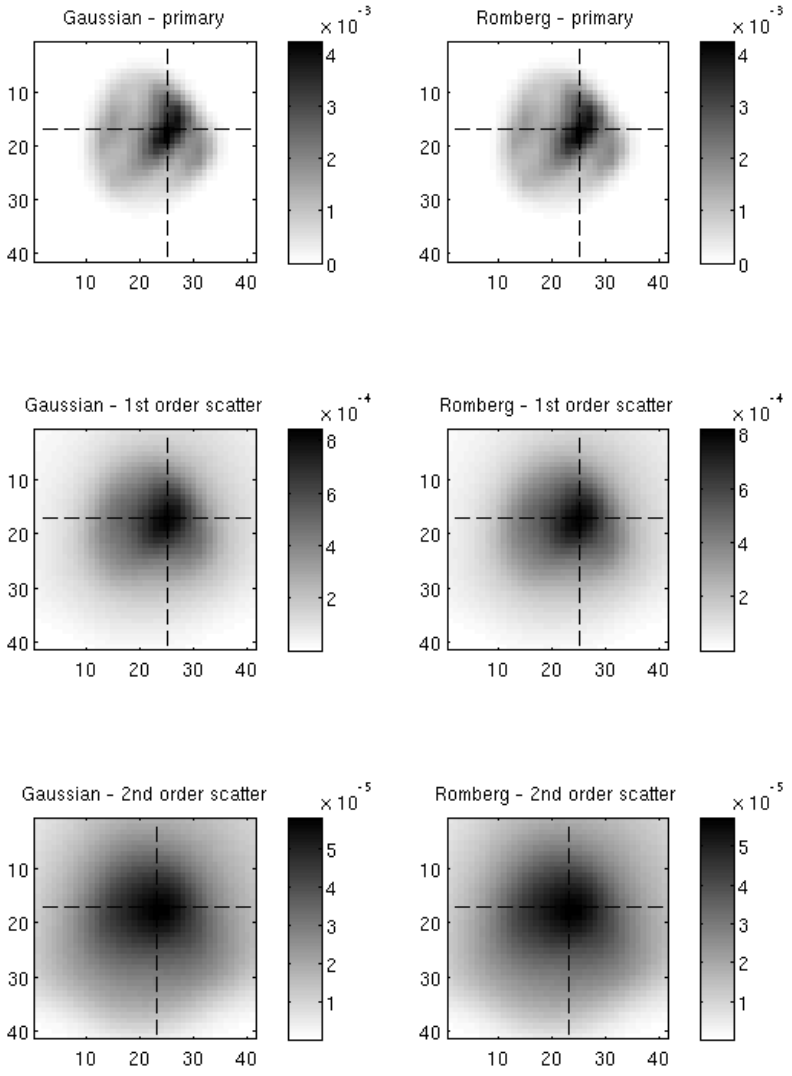

Figure 3.10: Primary, first-order and second-order PSFs for second experiment – 270° acquisition angle. The PSFs in the left column were calculated using the tables calculated with Romberg's method, and those in the right used the new Gaussian quadrature schemes. The figures are zoomed in on the region of most interest.

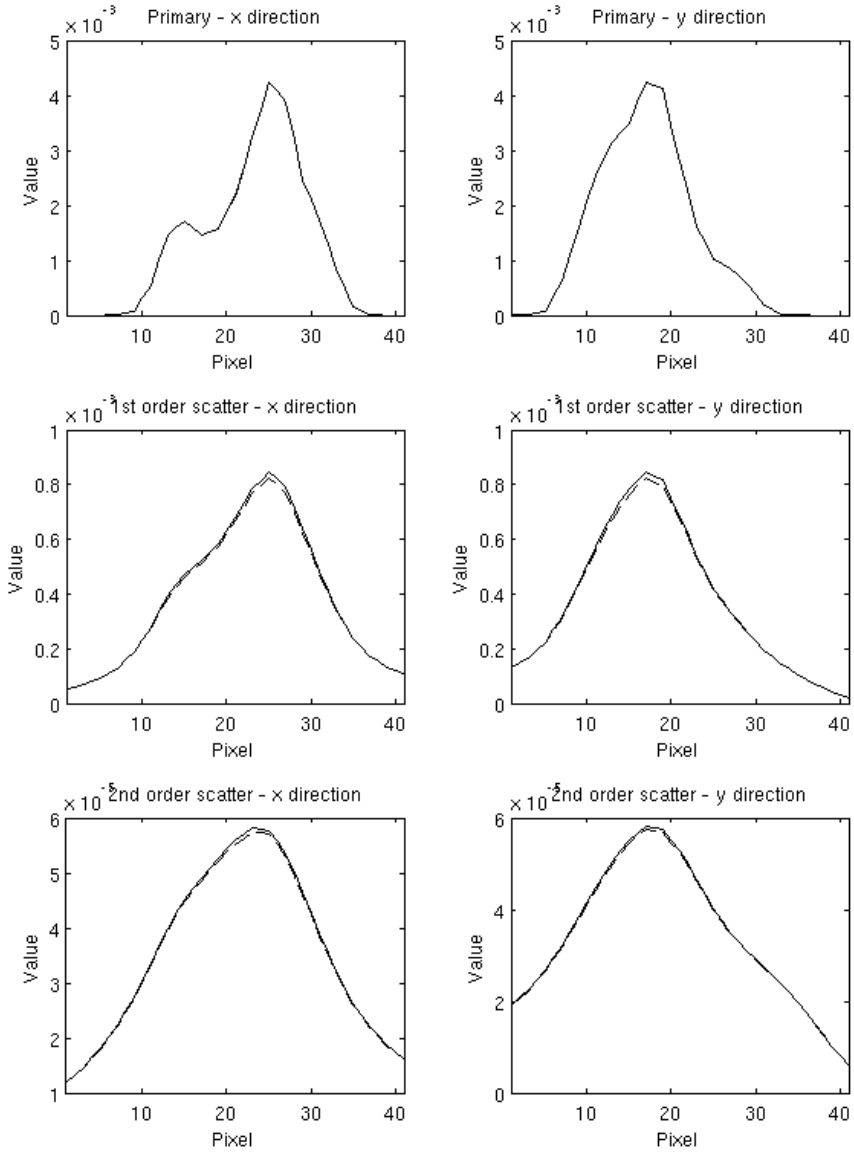

Figure 3.11: Corresponding profiles drawn along the dashed lines in Figure 3.10. The profiles drawn with a solid line are profiles for the Gaussian lookup tables; the dashed lines are for the Romberg lookup tables.

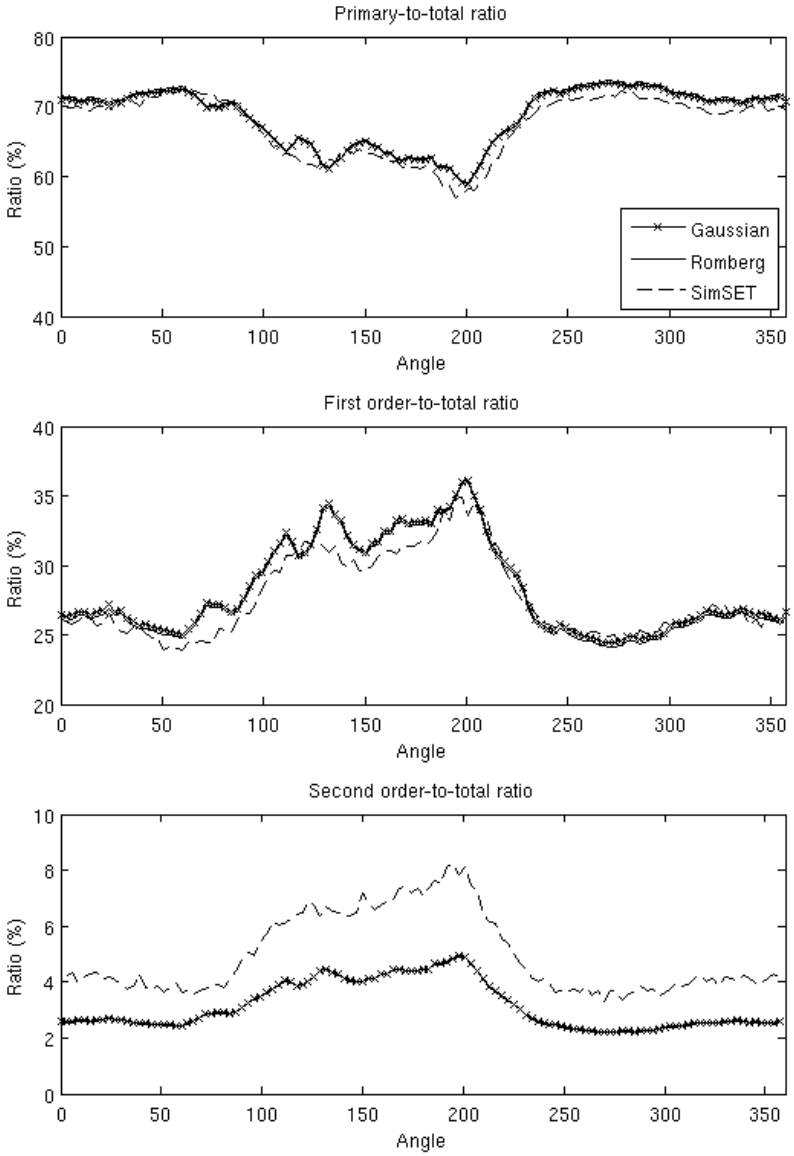

Figure 3.12: Ratios of primary, first-order and second-order photons detected to total photons detected for second experiment.

Chapter 4

Concluding Remarks and Future Work

4.1 Conclusions

In this thesis, we have replaced the Romberg integration method that was being used to evaluate integrals in APD with a Gaussian quadrature method. The primary advantage of this method is that it is significantly faster than Romberg's method. The time to create the first-order lookup table has been reduced from about 10 days (240 hours) to 6 hours, an improvement of 4000%. The time to create the second-order lookup table has not been significantly improved, however, since the bulk of that calculation does not involve integration. Nonetheless, the time to calculate a full set of APD lookup tables has now been reduced from nearly two weeks to about three days.

The change in integration method has resulted in some differences between the new and old lookup tables. In the case of the primary lookup table, these differences are small and the resulting PSFs calculated by APD are basically indistinguishable. For the first-order lookup table, the differences are larger, and are fairly significant for cases where the scattering point is close to the detector surface. As a result, the PSF for first-order scatter shows a higher quantity of scatter with the new tables. The difference in peak number of counts was observed to be as high as 18% for a point source in a water cylinder. However, the amount of first-order scatter calculated with the new lookup tables is actually in better agreement with Monte Carlo data for this case, so the amount of first-order scatter calculated is not

unusually high. For second-order scatter, the differences between the new and old tables are fairly significant. However, the actual calculated PSFs do not show large discrepancies with one another.

As a more general remark, we find that Gaussian quadrature (with a fixed number of points) has proven to be quite reliable and extremely computationally efficient for this application. In other applications where there is an extremely large number of integration operations required, Gaussian quadrature may be preferable to other methods which have significantly more computational overhead. One drawback of the method, however, is that it does not produce any error estimates. Thus, prior to adopting this kind of Gaussian quadrature method, one should do some experimentation to ensure how many points one should use to obtain sensible results.

4.2 Future Work

Some of the decisions made in the development of the quadrature method (such as the number of points to use, and the switching point from 16-16-8-8-8 to 8-8-8-8-8 in the first-order table calculation) may be dependent on the parameters that were used during the experimentation (cf. Section 2.2). It may be worth experimenting with some different sets of parameters to make sure that these choices are still sensible. It would also be interesting to see if the unusual spike in discrepancies around 22 cm mentioned in Section 2.3 is still present with a different pixel size.

While it is clear that the Gaussian quadrature scheme that has been implemented is much faster than Romberg's method, it is not clear whether or not it is more accurate. In light of some of the differences observed during the experiments, it is definitely of interest to know whether using Gaussian quadrature has also improved the overall accuracy of the APD calculation. Determining whether or not this is the case would require comparing against values that we know to be very accurate. One possibility would be to use results generated with more sophisticated quadrature methods (such as the adaptive methods mentioned in Section 2.1) as a basis for comparison.

From the point of view of APD itself, there are still areas where the speed of the calculation could be improved. The main bottleneck in lookup table calculation is now the summation that takes place in the construction of the second-order lookup table, and it may be worth examining this calculation to see if it can be accelerated as well. The main APD

method that calculates the PSFs can also be examined for potential speed improvements. Improving the speed of this calculation could help push APD towards being a more clinically viable method of scatter correction.

Finally, the discrepancy between the amount of second-order scatter calculated by SimSET versus APD is still a concern which warrants further investigation. We may wish to compare APD against other Monte Carlo based programs to see if the amounts of second-order scatter calculated are in better agreement with one another.

Bibliography

[AW04] John N. Aarsvold and Miles N. Wernick. *Emission Tomography: The Fundamentals of PET and SPECT*. Academic Press Inc., San Diego, 2004.

[GG00] Walter Gander and Walter Gautschi. Adaptive quadrature—revisited. *BIT*, 40(1):84–101, 2000.

[Hen82] Peter Henrici. *Essentials of numerical analysis with pocket calculator demonstrations*. John Wiley & Sons Inc., New York, 1982.

[HHG+93] R.L. Harrison, D.R. Haynor, S.B. Gillispie, S.D. Vannoy, M.S. Kaplan, and T.K. Lewellen. A public-domain simulation system for emission tomography photon tracking through heterogeneous attenuation using importance sampling. *J. Nuc. Med.*, 34(5), 1993.

[Hoc64] Urs W. Hochstrasser. Chapter 22: Orthogonal polynomials. In Milton Abramowitz and Irene A. Stegun, editors, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*, pages xiv+1046. For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1964.

[Lau97] Dirk P. Laurie. Calculation of Gauss-Kronrod quadrature rules. *Math. Comp.*, 66(219):1133–1145, 1997.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C*. Cambridge University Press, Cambridge, second edition, 1992.

[PXT+97] P.H. Pretorius, W. Xia, B.M.W. Tsui, T.S. Pan, and B.J. Villegas. Evaluation of right and left ventricular volume and ejection fraction using a mathematical cardiac torso phantom for gated pool spect. *J. Nuc. Med.*, 38(10), 1997.

[Ral65] Anthony Ralston. *A first course in numerical analysis*. McGraw-Hill Book Co., New York, 1965.

[SB02] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition, 2002. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.

[Van04] Eric Vandervoort. Implementation of an analytically based scatter correction in spect reconstructions. Master's thesis, University of British Columbia, January 2004.

[Wel97] R. Glenn Wells. *Analytical Calculation of Photon Distributions in SPECT Projections*. PhD thesis, University of British Columbia, August 1997.