Modelling Light Transport Through Biological Tissue Using the Simplified Spherical Harmonics Approximation

Michael Chu

School of Physics

University of Exeter

Submitted to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Physics, September 2010.

Declaration

This thesis is available for library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(Signature) ....

Abstract

Optical Tomography is a medical imaging modality that can be used to non-invasively image functional changes within the body. As near-infrared light is highly scattered by biological tissue, the process of image reconstruction is ill-posed and, in general is also under-determined. As such, model based iterative image reconstruction methods are used. These methods require an accurate model of light propagation through tissue, also known as the forward model.

The diffusion approximation (DA) to the radiative transport equation is one of the most widely used forward models. It is based on the assumption that scattering events dominate over absorption events resulting in a diffuse light distribution. This is valid in cases with low absorption coefficients or large geometries (greater than a few scattering lengths). In many cases, however, such as in small animal imaging where the source-detector separation is small, this assumption is not valid and so a higher-ordered approximation is required.

In this thesis, a three-dimensional frequency domain forward model based on the simplified spherical harmonics (SP_N) approximation to the radiative transport equation is introduced. By comparison with a Monte-Carlo model, the SP_N approximation is shown to be more accurate than the DA, especially in regions near to the sources and detectors and the increase in accuracy is greater in cases with stronger absorption. This is particularly important for bioluminescent imaging of small animals which involve both small geometries and strong absorption. Due to the asymptotic nature of the

SP_N approximation, the highest ordered model was not necessarily the most accurate, but all models with N>1 were more accurate than the DA.

The SP_N based forward model has also been implemented into an image reconstruction algorithm. Despite the fact that the SP_N approximation does not combine the scattering coefficient and anisotropy factor into a single variable, as is the case in the DA, it was found that it is not possible to reconstruct them uniquely. The SP_N based models were shown to be able to reconstruct optical maps with greater accuracy than the DA. However, due to the increased number of unknowns to be recovered, the SP_7 based reconstructed images contained significant artefact and cross-talk.

Finally, a SP_N-Diffusion hybrid model was developed in which the SP_N model was used in the regions near to the source and the DA elsewhere. This model provides the increase of accuracy of the SP_N models in the regions where the DA is insufficient, whilst retaining the computational efficiency of the DA. It was shown that the hybrid model leads to increased accuracy not only in the regions solved using the SP_N model, but also in the DA based regions where as in a pure DA model, the errors near the source were propagated throughout the domain. It is also shown that the hybrid model can be solved in half the time of the full SP_N model.

Acknowledgements

I would first like to thank my supervisor Dr. Hamid Dehghani for giving me the opportunity to carry out this PhD. Without his continuous support, encouragement and guidance, this thesis would not have been possible.

I would also like to thank my friends and colleagues at the Biomedical physics group for creating an enjoyable research environment and many interesting conversations over cups of tea. Thanks to Matthew Eames, James Bell, Elly Martin, Natalie Garret, Jessica Mansfield, John Hale, Julius Apweiler, Sarah Carr, Amanda Lindsay, Rachel Palfrey, Sterenn Le Nen Davey as well as my friends from Birmingham and Cardiff and anyone I may have forgotten.

My parents and family have always been supportive and encouraging to me over the years, and I am extremely grateful for them. Finally, I would like to thank Anne-Marie for her love and patience and for keeping me sane.

Thank you all

Mike

List of publications

Conference Papers

M. K. Chu, A. D. Klose and H. Dehghani (2008). Light Transport in Soft Tissue Based on Simplified Spherical Harmonics Approximation to Radiative Transport Equation. Biomedical Optics Topical Meeting. St Petersburg, FL, Optical Society of America.

M. K. Chu, A. D. Klose, I. B. Styles, K. Vishwanath and H. Dehghani (2009a). Frequency domain 3D simplified spherical harmonics approximation: development, validation, and implication in bioluminescence imaging. Optical Tomography and Spectroscopy of Tissue VIII, San Jose, CA, USA, SPIE.

M. K. Chu and H. Dehghani (2010). Diffuse Optical Tomography based on Simplified Spherical Harmonics Approximation. Biomedical Optics Topical Meeting, Miami, Fl, Optical Society of America.

Journal papers

M. K. Chu, K. Vishwanath, A. D. Klose and H. Dehghani (2009). "Light transport in biological tissue using three-dimensional frequency-domain simplified spherical harmonics equations." Physics in Medicine and Biology **54**(8): 2493-2509.

M. K. Chu and H. Dehghani (2009). "Image reconstruction in diffuse optical tomography based on simplified spherical harmonics approximation." Opt. Express **17**(26): 24208-24223.

List of Figures

List of Tables

Table of Contents

Chapter 1 - Introduction

1.1 Background

The development of three-dimensional (3D) medical imaging techniques have revolutionised the way illnesses and diseases are both diagnosed and treated. The ability to see inside the body with techniques such as X-ray computed tomography (CT) or magnetic resonance imaging (MRI), have allowed physicians to both identify problems as they develop and also to directly observe the effects of various treatments.

Imaging contrast is provided by the interactions between the probing radiation and tissues within the body. In the case of CT imaging, for example, incident X-rays undergo Compton type scattering effects with free-electrons in the tissue (Wolbarst 1999). As such, CT images indicate the electron densities of different tissue types. MRI imaging, however, probes the tissue with a strong magnetic field, causing the magnetic moments of protons within the body to align with it. A radio-frequency field is then applied, causing the alignment of the protons to change. When this field is switched off, the protons will re-align with the magnetic field, emitting a detectable signal (Wolbarst 1999). The contrast in an MRI image is therefore determined by the varying relaxation times of protons in different tissues.

Diffuse Optical tomography (DOT) is a relatively new imaging modality that uses low-energy near-infrared light to probe the body. The use of optical techniques in medical diagnosis and research can be dated as far back as 1929, when Cutler et al used transillumination as an aid in detecting breast legions

(Cutler 1929). This technique involved illuminating one side of the breast and observing shadows cast by large lesions, Figure 1.1. In recent years, however, significant advances have been made in the areas of light sources, light detection, photon transport theory and numerical modelling, allowing optical imaging to develop into a 3D, functional imaging tool.

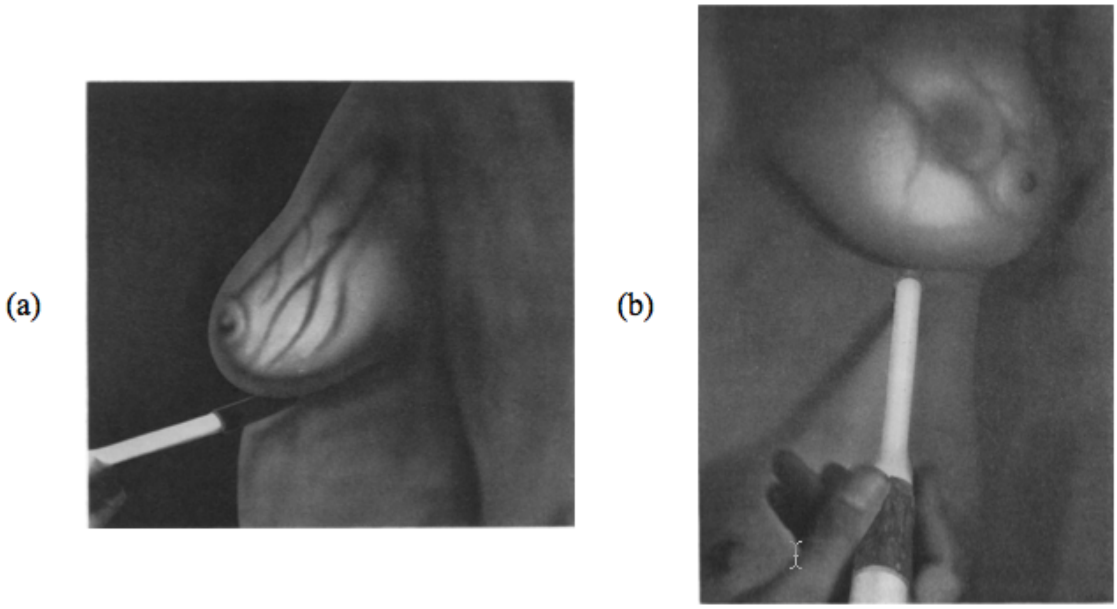


Figure 1.1 The appearance of (a) a healthy breast and (b) the shadow cast by a solid tumour on transillumination (Cutler 1929)

Unlike X-rays which pass through the body almost linearly, near infrared (NIR) light is highly scattered in biological tissue. NIR photons can also be absorbed by the sample, but this process is much less probable than scattering events. Contrast in DOT is provided by the differing absorption and scattering properties of various chromophores, such as water, oxy-haemoglobin and deoxy-haemoglobin and, as such, is able to provide functional information such as blood volume and oxygenation levels (Gibson et al. 2005).

One of the current applications of DOT is in the detection of breast cancer. Current screening methods involve x-ray mammography for women over the age of fifty. There is no routine screening as mammography of pre-menopausal breasts is less effective due to high absorption in dense breasts . As tumours are known to result in increased vascularisation (Rice and Quinn 2002), they are detectable using optical methods.

DOT imaging has also been shown to be beneficial in imaging the haemodynamic response to brain function (Bluestone et al. 2001). This type of imaging is particularly relevant to neonates, as they can be susceptible to brain injuries that can lead to permanent damage and disabilities. Also, as the heads of newborn infants are small, light can penetrate into deeper regions of the brain.

In recent years, the application of optical methods to functional imaging in small animals with endogenous contrast has grown considerably (Nziachristos 2006). Targeted bioluminescent or fluorescent markers can be injected into the body to be activated or absorbed only by specific cell types or functional states. This allows imaging of physiology at the molecular level. Imaging in small animals, however, introduces specific challenges for optical methods. The small tissue volumes encountered, for example, limit the accuracy of currently used light propagation models. Small animal studies, however, are still a useful tool for research and pre-clinical studies.

DOT has several advantages over traditional imaging modalities. One of the most important of these is that, unlike x-ray CT or radionuclide imaging, the probing radiation is non-ionising. This means that repetitive imaging can be performed with no risk of complications. The equipment required is also relatively inexpensive when compared to modalities such as MRI. The equipment is also compact and can be fitted into a single trolley allowing imaging at the bedside with fast acquisition times. However, due to the highly scattering nature of near-infrared light in tissue, spatial resolution is low. As scattering increases with depth, the resolution decreases. At the centre of the domain, it is estimated that objects no smaller than 10% of the domains external diameter can be detectable (Pogue et al. 2006).

1.2 Aims & Motivation

This thesis discusses the development of a model of light propagation in tissue, an important problem in the field of DOT. At present, the Diffusion Approximation (DA) to the Radiative Transport Equation (RTE) is widely used. Whilst accurate in many human tissues for large geometries, the DA is known to be inadequate in many cases. In the case of small animal imaging, for example, the separation between light sources and detectors is small. As a result, photons may not have undergone sufficient scatters to create a diffuse distribution before reaching the detector. In bioluminescent imaging of small animals, the light source takes the form of a light-emitting protein that has a peak emission at a wavelength corresponding to strong absorption by haemoglobin. Both of these factors mean that the diffusion approximation is not valid. The aim of this thesis is to develop a model for light propagation through soft tissue that can overcome the limitations of the diffusion approximation, particularly those encountered in the bioluminescent imaging of small animals.

The model developed in this thesis makes use of the Simplified Spherical Harmonics (SP_N) Approximation to the Radiative Transport Equation, which is discussed in detail in Chapter 2. This model has delivered an increase in

accuracy over the commonly used Diffusion Approximation whilst incurring lower computational costs than other higher-ordered approximations.
The structure of this thesis is as follows:

Chapter 2 provides an introduction to the principles behind tissue optics, and in particular the optical properties encountered in small animals.

Chapter 3 introduces the instrumentation required for optical tomography and discusses some of the current applications.

Chapter 4 discusses some of the existing medical imaging modalities that are currently used in both clinical and research environments.

Chapter 5 introduces the forward problem in which the propagation of light through soft tissue is modelled.

Chapter 6 introduces the inverse problem of the image reconstruction process.

Chapter 7 introduces a new model of light propagation based on the Simplified Spherical Harmonics approximation to the radiative transport equation. The implementation and validation of the approximation is discussed.

Chapter 8 investigates the use of the SP_N approximation in image reconstruction. The sensitivities to optical parameters are tested and a series of reconstructions are performed.

Chapter 9 introduces a SP_N – Diffusion hybrid model. This hybrid model uses the SP_N approximation in the regions near to sources and detectors where the DA is known to be inaccurate. In the remaining regions of the domain, i.e. beyond a few scattering lengths from the sources and detectors, the diffusion approximation can be used. The hybrid model aims to overcome the inaccuracies of the diffusion approximation whilst minimising the increase in computational load introduced with the use of higher-ordered models.

Chapter 10 provides a conclusion to the thesis and a discussion of possible directions for future work.

Chapter 2 - Tissue Optics

2.1 Introduction

The propagation of light through the body is governed by the optical properties of the various tissues encountered. Light with wavelengths between 600-900nm is capable of passing deep into tissue due to a window of low absorption, Figure 2.1. The varying interactions between the probing light and common chromophores such as oxy and deoxy-haemaglobin and water provide contrast in DOT images. The changes in contrast due to changes in the relative concentrations of these chromophores allows the reconstruction of functional information such as blood volume and oxygenation levels (Gibson et al. 2005).

In this section, the basic principles behind the interactions of near infrared (NIR) light and tissue will be introduced.

2.2 Scattering

Unlike X-rays which pass through the body almost linearly, near infrared (NIR) light is highly scattered in biological tissue. It has been empirically shown that the scattering spectrum fits well with Mie scattering theory which which describes the scattering of light by a series of spherical particles (van Staveren et al. 1991; Mourant et al. 1997), Figure 2.2. The probability that a photon will undergo a scattering event whilst travelling through a unit length of a medium is given by the scattering coefficient

$$\mu_s = \rho_s \sigma_s \tag{2.1}$$

where ρ_s is the volume density, with units of cm^{-3} of scattering particles and σ_s is the scattering cross section of the particles with units of cm^2. The mean free path between scattering events, which is the average distance travelled by a photon before being scattered, is the inverse of the scattering coefficient.

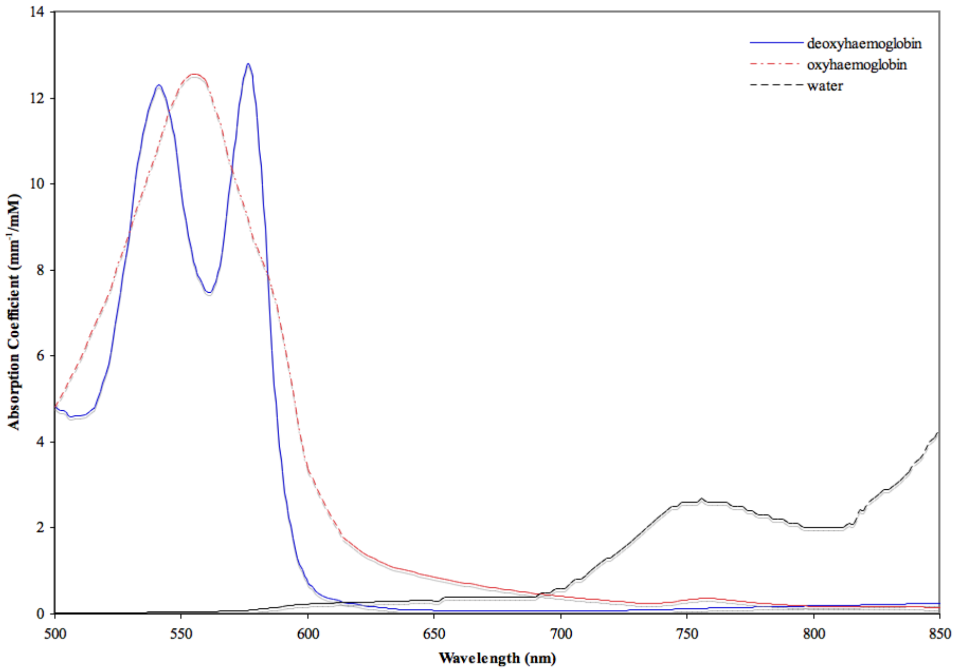


Figure 2.1 Absorption coefficients for oxyhaemoglobin, deoxyhaemoglobin and water at varying wavelengths based on data from Prahl (Prahl 2001)

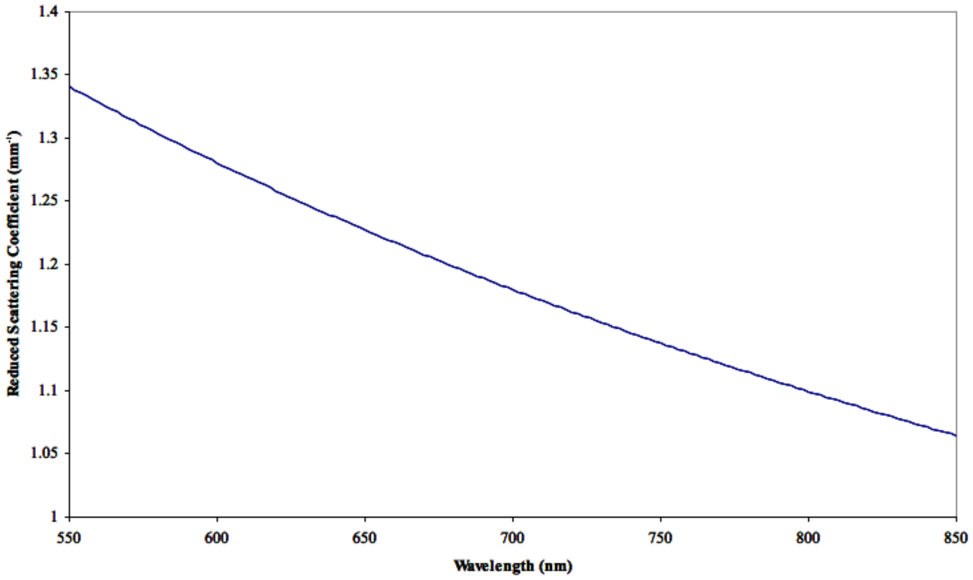


Figure 2.2 Spectrum of reduced scattering coefficient based on data from Prahl (Prahl 2001).

The probability of a photon being scattered from a direction $\hat{s}$ to a new direction $\hat{s}'$, given by the phase function $f(\hat{\mathbf{s}},\hat{\mathbf{s}}')$, can be thought of as a probability distribution and therefore normalised such that

$$\int_{4\pi} f(\hat{\mathbf{s}},\hat{\mathbf{s}}')d\omega = 1 \tag{2.2}$$

where $d\omega$ is a differential solid angle in the direction of $\hat{s}$. The choice of phase function is an important one as it determines how well the model can describe the nature of scattering events. The most basic assumption would be that scattering is wholly isotropic and thus

$$f(\hat{s},\hat{s}') = \frac{1}{4\pi} \tag{2.3}$$

meaning scattering in all directions is equally probable. In reality, however, scattering in biological tissue tends to be biased towards a particular direction and is therefore anisotropic. In this case, the Henyey-Greenstein phase function proves to be an accurate description and is commonly applied (Kienle et al. 2001; Sharma and Banerjee 2003)

$$f(\cos\theta) = \frac{1-g^2}{4\pi\left(1+g^2-2g\cos\theta\right)^{3/2}} \tag{2.4}$$

where the anisotropy factor, g, is defined as the average cosine of the scattering angle θ. In general, human tissue tends to forward scattering with typical values of g ranging between 0.8 and 0.9 (Jacques et al. 1987; Yoon et al. 1987; Cheong et al. 1990).

When modelling light propagation with the diffusion approximation, which will be discussed in chapter 5, the reduced scattering coefficient is commonly applied and is defined by $\mu_s{}' = (1-g)\mu_s$. According to Mie theory, the reduced scattering coefficient is dependent upon the wavelength of the photon and is given by

$$\mu_s' = a\lambda^{-b} \tag{2.5}$$

The scatter amplitude and scatter power are defined by a and b respectively and both depend on the scattering cross section and number density of particles. Larger scatterers generally have lower a and b values whilst smaller scatterers tend to have higher a and b values (Dehghani et al. 2008). These parameters are likely to vary with tissue composition and various cellular structures/densities (Wang et al. 2006b) and so variations in reduced scatter coefficient may indicate variations in tissue structure.

2.3 Absorption

In the absence of scattering, the attenuation coefficient can be calculated using the Beer-Lambert Law. This law states that the intensity of a beam of light travelling through an attenuator, Figure 2.3, will reduce exponentially depending on the width of the attenuator, equation 2.6.

$$I = I_0 e^{-\mu_a x} \tag{2.6}$$

where I is the measured intensity, I_0 is the incident intensity, x is the thickness of the attenuator and μ_a is the absorption coefficient, also with units of mm^{-1}. By plotting the ratio of incident and exciting intensities against the attenuator thickness, as shown in Figure 2.4, the absorption coefficient will be equal to the rate of decay.

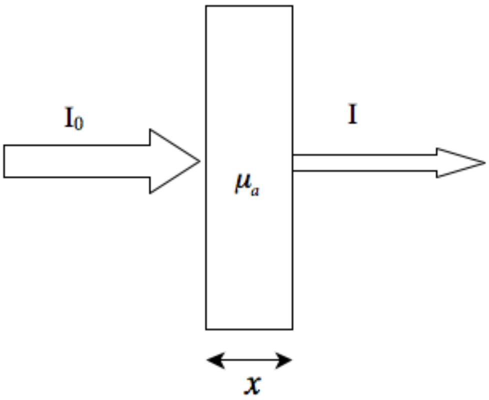


Figure 2.3 Attenuation of an incident light beam through a non-scattering slab.

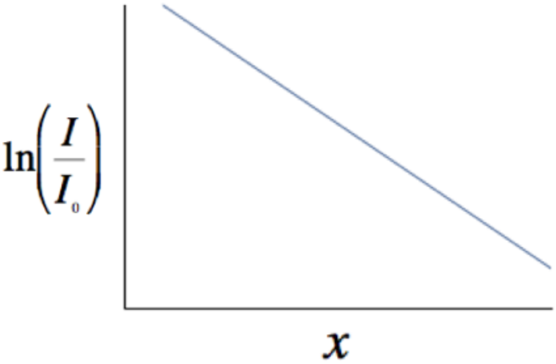


Figure 2.4 The gradient of a plot of logged ratio of transmitted and incident intensities against the sample thickness determines the absorption coefficient.

In biological tissues, a number of chromophores with varying absorption spectra, see Figure 2.1, will be present in varying concentrations. The relationship between the tissue's total absorption coefficient and its chromophore composition is defined by

$$\mu_a(\lambda) = \sum_n \varepsilon_i(\lambda) c_i \tag{2.7}$$

where n is the number of chromophores present in the tissue, $\varepsilon_i(\lambda)$ is the extinction coefficient of a particular chromophore at wavelength λ and c is the concentration of the chromophore. The product of the extinction coefficient and concentration gives the absorption coefficient of a particular chromophore at a given wavelength.

In highly scattering media, the Beer-Lambert law must be modified to include the effects of scattering. When a photon undergoes scattering as it passes through a tissue, its path length is increased. The actual distance that a photon travels through a tissue, the differential path length (DP), is related to the thickness of the tissue, x, by

$$DP = x \cdot DPF \tag{2.8}$$

where DPF is the differential path length factor. The DPF for a tissue can be determined by measuring the mean time of flight. To do this, a picosecond pulse of light is generated and split. One part of the light is led directly to the detector. The remaining part of the pulse passes through a tissue sample with an accurately measured thickness. By measuring the time difference between the two pulses arriving at the detector, the differential path length can be calculated as

$$DPF = \frac{vt}{d} \tag{2.9}$$

where v is the speed of light in the tissue, t is the time difference between the arrival of two pulses. The attenuation, A, can then be determined using the modified Beer-Lambert Law

$$A = \ln\left(\frac{I_0}{I}\right) = \mu_a \cdot x \cdot DPF + G \tag{2.10}$$

where G is a constant relating to the scattering losses, dependent on the geometry of the measurement system and the scattering coefficient of the tissue being studied. The value of G, however, is not known and so an absolute value of absorption coefficient cannot be measured. If it is assumed that G remains constant throughout a series of measurements, it is possible to determine changes in absorption coefficient using

$$A_1 - A_0 = \left(\mu_a^1 - \mu_a^0\right) \cdot x \cdot DPF \qquad (2.11)$$

where A_0 and μ_a^0 are the attenuation and absorption coefficients in an initial state and A_1 and μ_a^1 are the attenuation and absorption coefficients in a perturbed state.

2.3 Refractive Index

The speed of light in tissue is determined by the refractive index as

$$v = \frac{c}{n} \qquad (2.12)$$

where *c* is the speed of light in free space in ms^{-1} and *n* is the refractive index of the tissue. Accurate measurements of refractive index are difficult to make *in vivo*, and as such, the absolute values of refractive index are still under investigation. Bolin, *et al* found the refractive index of a range of tissues to lie in the 1.38-1.44 range, independent of species (Bolin et al. 1989). Many models, however, use a value between 1.33-1.6 (Brooksby et al. 2003). The lower values are justified by the high water content of many tissues, which would suggest its refractive index is close to that of water.

2.4 Measuring Optical Properties

The actual optical properties of a tissue can be determined either directly or indirectly. Direct methods basically measure the attenuation of light through thin slabs and use the Beer-Lambert law to determine the total attenuation coefficient μ_t. The accuracy of this method, however, is limited by the inability to separate direct, un-scattered light and scattered light that has scattered back on to the measurement axis.

Alternatively, the optical properties can be determined indirectly by comparing measurements of transilluminated light to those calculated using a photon transport model, typically a time-resolved Monte Carlo model (Peters and et al. 1990; Simpson and et al. 1998). Many measurements of extinction coefficients of tissue samples have been performed in vitro. It is not known, however, whether these properties would be the same in vivo.

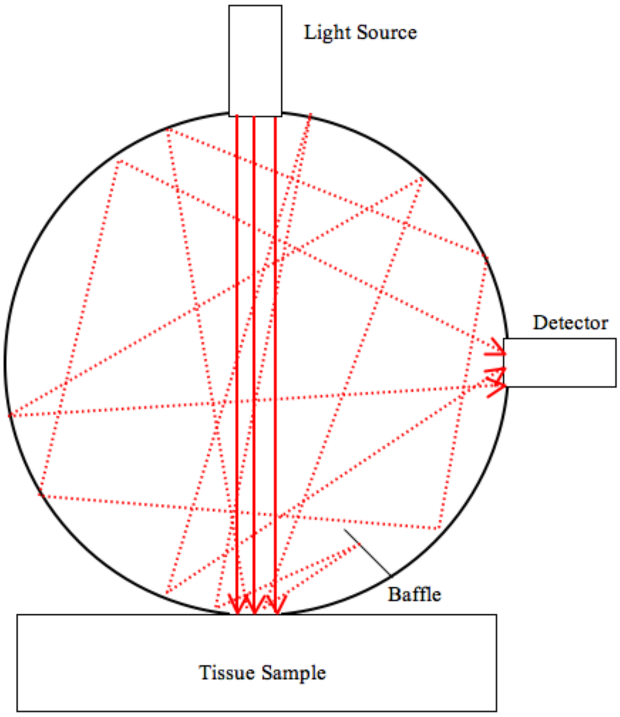


Figure 2.5 Schematic diagram of an Integrating Sphere used to measure bulk optical properties

Bulk optical properties of tissues can be measured in vivo with the use of an integrating sphere. The integrating sphere consists of a spherical chamber, which has a highly reflective inner surface, and a window that allows collection of light, Figure 2.5. As the collected light is reflected around the sphere, the distribution becomes diffuse and can be measured with a single optic fibre. Depending on whether the integrating sphere is placed on the same side of the sample as the light source or on the opposite side, measurements can be made of either reflectance or transmission (Jacquez and Kuppenheim 1955; Edwards et al. 1961). Alternatively, a double integrating sphere system can be used which places the sample between two spheres (Pickering et al. 1993). This allows simultaneous measurements of reflectance and transmittance meaning the effects of external conditions, e.g. heating, on reflectance and transmission can be measured. Measurements of reflectance and transmission can then be converted into values of scattering coefficient, absorption coefficient and anisotropy factor using the inverse adding-doubling method (Prahl et al. 1993).

A full review of tissue optical properties was compiled by Cheong et al (Cheong et al. 1990). The website of Scott Prahl also provides a useful resource with spectra of several tissue types, contrast agents and chromophores (Prahl 2001). In general, the absorption coefficient of most biological tissues tends to be somewhere in the range of 0.01-0.5cm^{-1} at wavelengths greater than 625nm with scattering coefficients ranging from 10-200cm^{-1}. Scattering in tissue is generally forward peaked, with typical anisotropy factors ranging between 0.5 and 0.95, depending on the type of tissue.

As the optical properties of a tissue are dependent on the wavelength of incident light, the choice of light source is important. Wavelengths are generally chosen from a window between 650 nm and 900 nm as below 650 nm, absorption due to haemoglobin prevents light of lower wavelength from penetrating the tissue whilst absorption due to water prevents the use of higher wavelengths, Figure 2.1.

2.5 Optical Contrast Between Healthy & Diseased States

In order for optical methods to be clinically useful, they must be capable of differentiating between normal and diseased states. Contrast in optical imaging is provided by differing concentrations of chromophores, such as oxy and deoxyhaemoglobin and water, that each have unique absorption spectra. In order for diseased states to be detectable using optical methods, they must result in a change in optical properties.

In the brain, a number of conditions can lead to changes in optical properties. Haematomas or haemorrhages are caused by ruptured blood vessels leading to a pooling of blood and, therefore, an increase in the total blood volume (Grubb et al. 1977). Cerebral ischemia is classified as a reduction in the blood flow to parts of the brain caused by haemorrhages, constrictions or obstructions to blood vessels. The result is a lack of sufficient oxygen supply to areas beyond the obstruction leading to an increase in deoxyhaemoglobin and a decrease in oxyhaemoglobin concentrations (Wolf et al. 1997; Culver et al. 2003b; Abookasis et al. 2009). The total volume of blood can also be seen to decrease. Optical methods may also be useful in detecting the degeneration of brain tissues, such as demyelination of nerves, as such changes lead to changes in scattering properties (Radhakrishnan et al. 2005; Senapati et al. 2005).

The optical properties of both healthy and diseased breast tissue has been widely investigated (Peters and et al. 1990; Tromberg et al. 1997; Ghosh et al. 2001; Durduran and et al. 2002). It has been shown that optical contrast exists between healthy and diseased states as tumour containing tissues tend to have increased absorption and scatter coefficients (Tromberg et al. 1997; Chernomordik et al. 2001).

Optical imaging studies of small animals have been conducted using both endogenous and exogenous contrast. In recent years, studies utilising endogenous contrast have been mainly focussed on blood oximetry to investigate both cerebral ischemia and haemodynamic responses in the brain (Hielscher 2005). In both of these applications, contrast is provided by varying concentrations of oxyhaemoglobin, deoxyhaemoglobin and total blood volume. In these cases, it has been shown that, rather than first reconstructing for μ_a and μ_s at multiple wavelengths, it is more efficient to directly reconstruct the chromophores concentrations (Corlu et al. 2003; Li et al. 2004).

Contrast in small animal imaging can also be provided by exogenous agents such as fluorescent or bioluminescent markers. Fluorescent markers emit light when excited by an external light field but at a wavelength higher than that of the excitation field. Alternatively, bioluminescent markers, such as luciferase, can be used which emit light when they encounter certain molecular environments. The main drawback of using bioluminescent luciferase is that it has a peak emission at approximately 570nm (Li et al. 2004) which corresponds to a region of strong absorption making the commonly used diffusion approximation, which will be discussed in Chapter 5, invalid.

Chapter 3 - Diffuse Optical Tomography Instrumentation & Applications

3.1 Introduction

The equipment required to build a DOT system is, in principle, well established (Gibson et al. 2005). An imaging system generally consists of a light source, which is commonly guided to the boundary of the imaging domain by a series of optic fibres. However, a number of systems exist that utilise a non-contact light source which allows the imaging of objects with arbitrary shapes (Schulz et al. 2003; Yu et al. 2003; Joshi et al. 2006). Measurements can then be made of either the transilluminated or reflected light using a number of detectors. These measurement can then be used to reconstruct 3D maps of the optical property distribution within the domain, or in the case of multi-wavelength systems, chromophore concentrations can be reconstructed.

This chapter describes the types of imaging system that can be used and some of the methods of improving image contrast or accuracy. Some of the existing small animal imaging systems are then discussed.

3.2 Imaging modes

DOT imaging systems can be classified into one of three source operation modes. In a continuous wave system, the target is illuminated continuously with a light source of constant intensity. In a frequency domain system, the target is also continuously illuminated, but the intensity of light is modulated at radio frequencies. In a time-domain system, light is delivered in a short pulse of the order of several pico-seconds. Each of these three operating modes will be discussed in more detail below.

3.2.1 Continuous Wave

Continuous wave systems provide a relatively simple and inexpensive technique for implementing optical tomography. A beam of light of constant intensity is transmitted between two points on the surface of the subject and the intensity of the exiting light is detected, Figure 3.1a. This basic technique has been in use, in more primitive forms, since the nineteenth century when it was used to study the head, testes and breast by simply observing trans-illuminated light (Bright 1831; Curling 1843; Cutler 1929). With increasing understanding of the interactions between light and tissue, the technique has more recently been used to observe haemodynamic and oxygenation changes in superficial tissues and of outer regions of the brain by detecting reflected diffuse light. The University of Washington in St Louis, for example, have developed a continuous wave system that has been used to image the adult visual cortex (Zeff et al. 2007a).

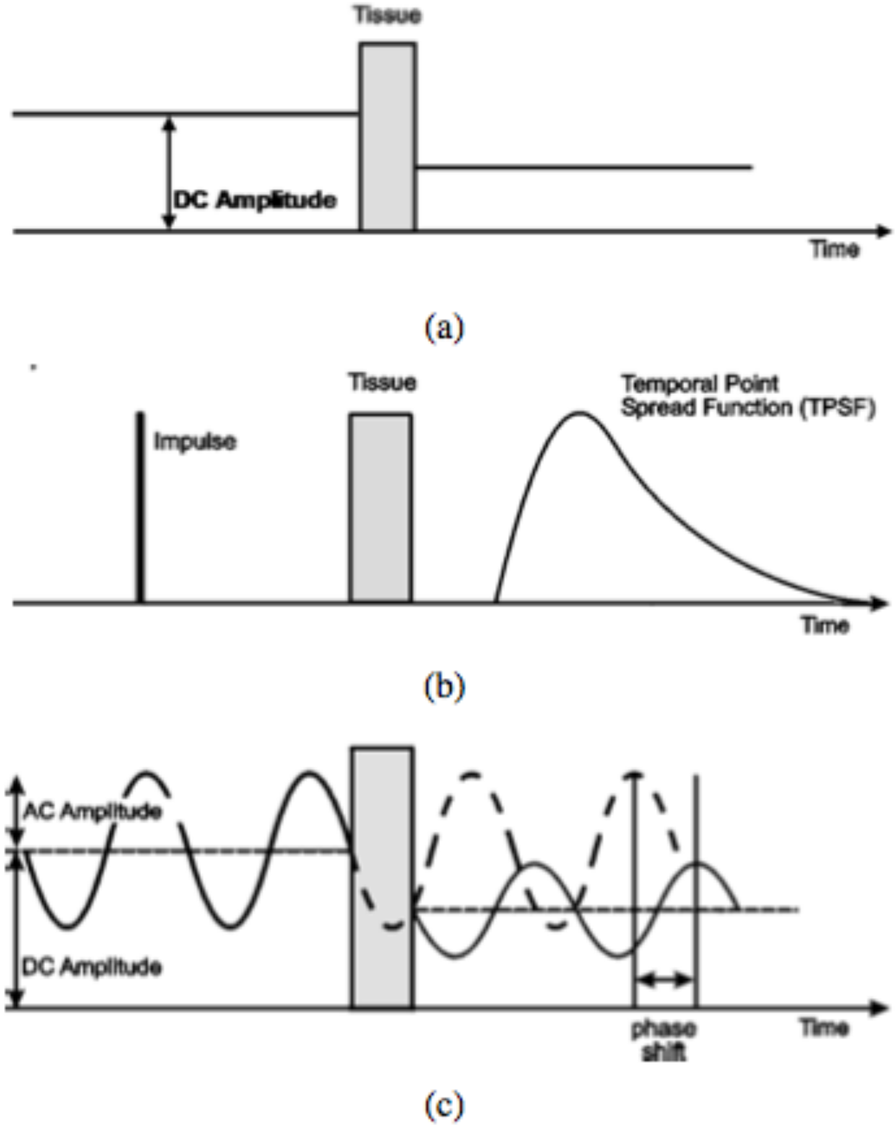
Whilst continuous wave systems can provide useful information for relatively low cost, they suffer from the problem of non-uniqueness. As previously mentioned, chapter 2, light can interact with tissue in either scattering events or absorption events. An increase in either the scattering coefficient alone, the absorption coefficient alone or a smaller increase in both the absorption and scattering coefficients could lead to exactly the same reduction in the intensity of light measured at the boundary, resulting in a non-unique measurement. It is therefore, not possible to reconstruct distinct maps of absorption and scattering coefficient using intensity measurements of a given wavelength alone (Arridge and Lionheart 1998). However, Culver *et al* have developed a hybrid frequency-domain/continuous wave system that has the benefits of a continuous wave system, i.e. ease of measurement and low cost, but the frequency domain data can be used to separate absorption and scattering effects (Culver et al. 2003a).

Continuous wave systems are also very sensitive to variations in the surface coupling, meaning measurements could be significantly different depending on the amount of contact between the source fibre and the skin. The effects of varying surface coupling can be accounted for by including coupling coefficients as unknowns in the image reconstruction process (Boas et al. 2001).

Another disadvantage to CW systems is that they are inherently more sensitive to the regions directly below the source/detector fibres due to the banana-shaped photon density measurement function (PDMF) (Arridge 1995b). The PDMF maps the sensitivity of boundary measurements to a point within the domain and is effectively a measurement of the probability of a detected photon having passed through that point. Sensitivity mapping will be discussed in more detail in chapter 6.

3.2.2 Time Domain

Time domain systems utilise light sources, e.g. lasers, that are capable of producing very short pulses of light. Some of the photons will pass through the domain to the detector directly, without scattering and are known as ballistic photons. Some photons will also be weakly scattered in a forward direction, following a similar path to the ballistic photons. The remainder of the photons will undergo many scattering events and are known as diffuse photons. As these different categories of photons travel through the medium with a wide range of path lengths, the time to reach the detector, the time of flight, will vary. As a result, the initial short pulse will be broadened by the time it is detected, Figure 3.1*b*. For a input pulse of the order of pico-seconds, this Temporal Point Spread Function (TPSF) can extend over several nano-seconds after passing through several centimetres of tissue (Hiraoka et al. 1993; Gibson et al. 2005). The shape of the TPSF, if measured accurately enough, can provide information about the attenuation properties of the medium. By integrating the TPSF, the intensity can be calculated which gives the same information as a continuous wave system. The mean time of flight of photons, i.e. the time it takes for a photon to travel between the source and detector, can then be used to estimate the photon's path length and distinguish between absorption and scattering effects.



(a)

(b)

(c)

Figure 3.1 Signal measured at the boundary for (a) a continuous wave source, (b) a pulsed, time-domain source, and (c) an intensity modulated, frequency domain, source (Hebden et al. 1997).

Further information can be obtained using time-gating techniques to distinguish between early and late arrival photons (Ntziachristos et al. 2005; Niedre et al. 2008; Leblond et al. 2009). The early arriving photons pass almost straight through the medium and are affected predominantly by absorption whereas the late photons are affected by both absorption and scatter. As well as improving spatial resolution, the use of early photons can reduce the ill-posed nature of the image reconstruction problem. This technique, however, is limited by the number of photons which have a sufficiently short path-length and therefore short flight times. Another disadvantage of taking measurements in the time-domain is that high temporal resolution equipment required can be very expensive.

Time domain systems tend to use time-correlated single photon counting (TCSPC) methods to record a histogram of the time of flight of individual photons relative to a reference pulse (Eda et al. 1999; Arridge et al. 2000a). Whilst these methods are slow, they generally have a high efficiency making them suitable for situations where the input beam travels through a thick medium resulting in a low intensity output beam (Becker 2005).

3.2.3 Frequency domain

Time domain data can also be expressed in the frequency domain, using Fourier transforms to switch between the two. The frequency-domain data can also be measured directly by measuring the exiting amplitude and phase shifts of an amplitude-modulated source beam, Figure 3.1c.

The quality of frequency domain data is dependent upon the choice of source modulation frequency. A review of phase measurements by *Chance et al* (Chance et al. 1998) showed that, for noise free data, the measurable phase shift increases with modulation frequency. In the presence of noise, however, the signal-to-noise ratio starts to decrease at higher frequencies. *Pogue et al* (Pogue et al. 1997) found the optimal modulation frequency to lie within the range of 50-250 MHz. In small animal imaging, however, the resultant phase shift is much smaller so a higher frequency is necessary (Kim et al. 2008).

The equipment necessary for frequency domain measurements is significantly less complex, and therefore, less expensive than that needed for time domain systems making it advantageous (Pogue and Patterson 1994; Pogue and et al. 1995; McBride et al. 2001; Orlova et al. 2008). Frequency domain systems are also relatively easy to develop and are capable of processing data quickly making them very well suited to situations where the output beam has a reasonably high intensity and data is required quickly.

3.3 Spectroscopy

The aim of a DOT imaging system is to provide a map of the optical properties within the sample tissue. If data is taken at a range of wavelengths, it is then possible to derive the various chromophore concentrations in a post-processing step by performing a least squares fit to the Beer's Law, equation 2.6. It is assumed that oxy-haemoglobin, deoxy-haemaglobin and water are the main sources of absorption and their molar absorption spectra have been measured experimentally (Prahl 2001; Srinivasan et al. 2003). The scatter amplitude and scatter power can be calculated from Mie theory, equation 2.5.

If data is obtained for multiple wavelengths simultaneously, it is possible to directly reconstruct for chromophore concentrations and scatter properties directly (Corlu et al. 2003; Srinivasan et al. 2006). This reduces the problem of reconstructing 14 unknowns, for example (based on reconstructing both $\mu_a$and μ_s at 7 wavelengths), to that of just 5 unknowns (oxy-haemoglobin, deoxy-haemoglobin, water and scatter power and amplitude) at each spatial location within the volume (Dehghani et al. 2008). The use of multiple wavelengths has also been shown to make the reconstruction process less sensitive to noise, which can potentially provide better contrast and reduce cross-talk between parameters (Srinivasan et al. 2005).

3.4 Molecular Imaging

Whilst the absorption and scattering properties of a tissue can provide functional information on a macroscopic level, the use of exogenous contrast agents can allow functional information on a molecular level. Molecular imaging using small animals has now become a growing field of interest (Schulz et al. 2004; Nziachristos 2006; Kumar et al. 2008; Zinn et al. 2008; Okawa and Yamada 2010).

Fluorescent imaging, for example, makes use of fluorescent probes that attach to a specific biological target or activate only when they encounter a specific chemical environment. When these markers are illuminated by an external light source at a specific excitation wavelength λ_x, they re-emit light with a emission wavelength λ_m. The fluorescence imaging problem is generally modelled in two stages. Firstly, the light field from the external light source is commonly modelled using the diffusion approximation to the radiative transport equation, see Chapter 5. The result of this diffusion equation is then used as a source term in a second diffusion equation which models the distribution of fluorescent light. Whilst the majority of models to date have utilised the diffusion approximation, a few transport based models have been developed (Klose and Larsen 2003; Klose et al. 2005). Fluorescence tomography has been applied with promising results in a number of studies (Chang et al. 1997; Lee and Sevick-Muraca 2002; Kepshire et al. 2008) including observations of physiology such as protease activity (Ntziachristos et al. 2002a) and tumor localisation (Chen et al. 2003). Davis *et al* have also combined a fluorescence imaging system with an MRI system to allow image-guided fluorescence tomography (Davis et al. 2008).

An alternative to fluorescent agents would be to use bioluminescent markers. As in the case of fluorescence imaging, a light emitting protein is attached to a specific cell type or biological target. The most commonly used light emitting protein is luciferase, which has a peak emission at approximately 560nm, Figure 3.2a. This corresponds to a region of strong absorption due to haemoglobin (Figure 3.2b), and so the DA is no longer valid as the condition $\mu_a >> \mu_s'$ does not apply. As such, there is a need for higher ordered approximations to the RTE. The strong absorption also limits bioluminescent studies to relatively superficial regions and more sensitive and expensive detectors are required (Hielscher 2005).

The location of the bioluminescent source in 3D is complicated by the fact that emission spectrum is dependent on the depth of the source due to the wavelength dependent nature of tissue optical properties (Chaudhari et al. 2005). It has been shown, however, that the use of multi-spectral data can greatly improve the accuracy of image reconstruction (Cong and Wang 2005). By labelling luciferase labelled HeLa cells into immuno-deficient mice, bioluminescent images were used to study the growth and metastasis of lung tumours (Edinger et al. 1999). It has also been shown that bioluminescence imaging can result in sensitivities comparible to PET (Iyer et al. 2002; Kuo et al. 2007).

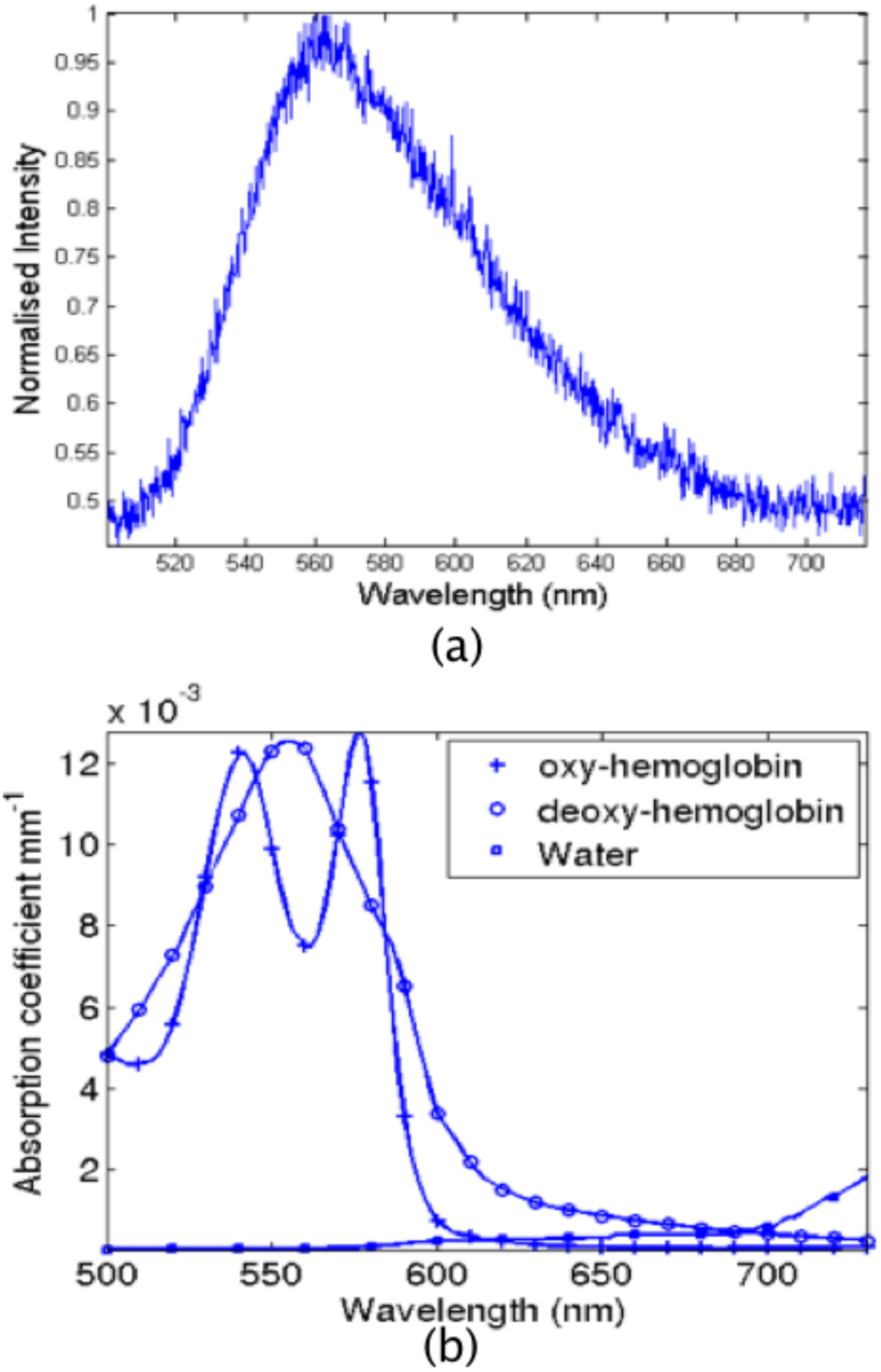


Figure 3.2 (a) Measured emission spectrum of luciferase, (b) absorption spectra of oxy and deoxy-haemoglobin and water.

3.7 Applications

3.7.1 Breast Imaging

The current method for breast cancer screening is based upon traditional x-ray mammography. The NHS system currently provides women over the age of fifty with regular mammograms every three years. Below the age of fifty, however, there is no routine screening system as mammograms are less effective in pre-menopausal breasts as their high density results in a low contrast between tumours and healthy tissue (Warren 2001). X-ray mammography involves the use of ionising radiation and as such the risks may outweigh the benefits, especially in younger women.

Tumours are known to be associated with an increased vascularisation (Rice and Quinn 2002). The subsequent changes in levels of Hb and HbO means that they can be imaged using optical methods. One of the main concerns of using DOT as a screening system is the limited spatial resolution that it can provide. As tumors grow beyond 1cm in diameter, the associated mortality rates rise rapidly (Webb et al. 2004). Any reliable screening method should therefore be able to detect tumors smaller than 1cm. Whilst optical tomography may not yet be ready to replace x-ray mammography, it may well prove to be useful as a complementary screening method which could also be used in younger women.

A number of DOT breast imaging systems have been developed in the last ten years. The Dartmouth group, for example, have built a frequency-domain system based on 16 sources (with a modulation frequency of 100 MHz) and 16 detectors arranged in a circular geometry (Pogue et al. 1997).

The system has also been coupled to a MRI scanner to allow simultaneous MR and DOT images allowing MRI-guided DOT (Brooksby et al. 2004).

One of the advantages of the circular source-detector arrangement as used by the Dartmouth group is that a large volume of tissue is sampled allowing good 3D reconstructions. However, due transmission through large volumes of tissue is relatively low and so measured intensities will be small.

The University of Pensylvannia have developed an alternative system that uses a compressed slab geometry (Choe et al. 2005). The patient's breast is compressed to a thickness of 5.5 to 7.5cm within a box that is then filled with a matching solution to create the slab geometry. The hybrid system uses six different light frequencies, four of which are intensity modulated, to allow simultaneous detection of continuous wave and frequency domain data (Konecky et al. 2008).

3.7.2 Imaging the neo-natal brain

Brain imaging in pre-term or term infants can lead to permanent disability or even death (Gibson et al. 2005). Conditions such as intraventricular haemorrhage, periventricular leucomalacia or hypoxic-ischaemic brain injury result in a disruption in the blood and therefore oxygen supply to areas of the brain. Whilst portable ultrasound systems are very effective at detecting haemorrhages, they cannot provide functional information about hypoxic ischaemic injuries (Hebden 2003). Near infrared spectroscopy has been applied to monitor changes in cerebral oxygenation and haemodynamics for over 30 years (Jobsis 1977), but does not provide any spatial information. Instead, optical tomography can be used to give functional information throughout the brain.

The first optical tomographic system for neonatal brain imaging was developed by Benaron et al (Hintz et al. 1998; Hintz et al. 1999; Benaron et al. 2000). They imaged two neonates, one with a stroke and one a control. Significant differences were seen between the two images at locations that agreed with CT images. This system, however, consisted of just one detector meaning measurement at each collection fibre were collected sequentially. Imaging times were in the order of several hours. More recent imaging systems utilize multiple channels to reduce scan times.

The UCL group, for example have built a 32 channel time-domain system that use custom made helmets to couple source and detector fibres to the infant head (Hebden et al. 2002). The system has been used to image the brain of a premature infant with a cerebral haemorrhage. The system has also been used for differential imaging, removing the need for calibration measurements. Large changes in absorption were measured as the partial pressures of oxygen and carbon dioxide were varied via a ventilator (Hebden et al. 2004).

Zeff *et al* developed a high-density optical tomography system that measures reflected light with a wide range of source-detector separations, where the different source-detector separations provide information at various depths (Zeff et al. 2007b). This system has been used to image neonatal brain. In response to a visual stimulus, increases in oxy and total haemoglobin were observed, along with decreases in deoxyhaemoglobin (Liao et al.).

3.7.3 Small animal imaging

In recent years, interest has grown in the optical imaging of small animals (Culver et al. 2003b; Bluestone et al. 2004a; Bluestone et al. 2004b; Ntziachristos et al. 2004), as progress in transgenic manipulation of small animals has allowed the development of models for human diseases (Hielscher 2005). Imaging can be performed using both endogenous and exogenous contrast with either bioluminescent or fluorescent markers.

As mentioned in Section 2.5, many of the studies utilising endogenous contrast have focussed on blood oxymetry in relation to ischemia and functional response. For example, Bluestone *et al* monitored the response to various degrees of hypercapnia and saw variations in oxy and deoxyhaemoglobin concentrations as the levels in inhaled CO_2 were varied. The same group also studied the effects of unilateral carotid artery occlusions, Figure 3.3.

Siegel *et al* investigated the neuro-response in rat brains to electrical stimulation of the paw in a comparative study with fMRI (Siegel et al. 2003). As shown in Figure 3.4, they observed increases in oxyhaemoglobin and deoxyhaemoglobin during stimulation. They found that the DOT system was able to provide temporal information on the haemodynamic response that was similar to that of fMRI.

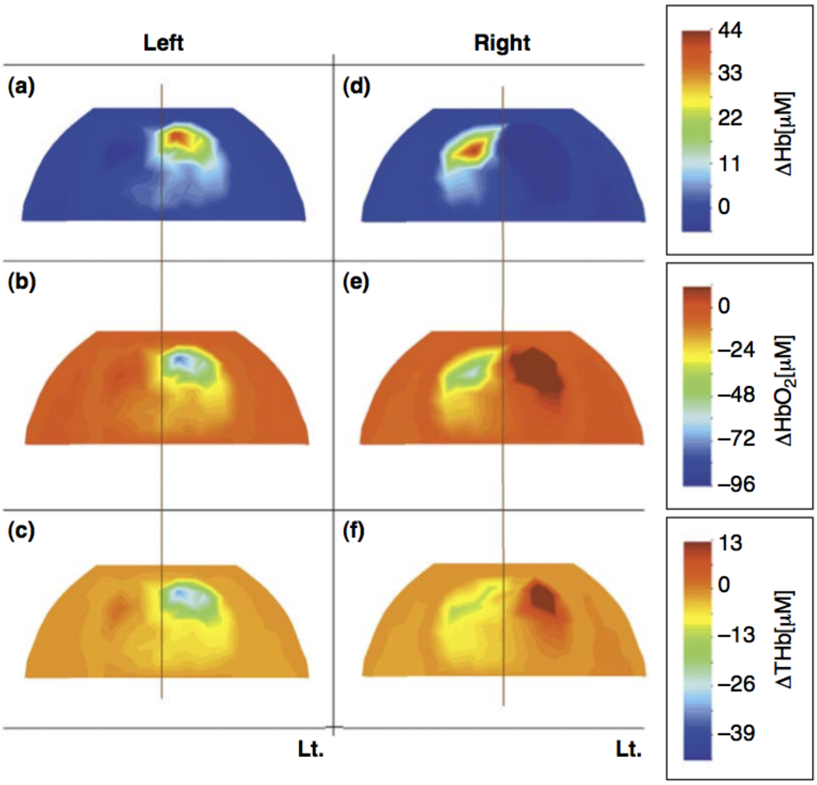


Figure 3.3 Coronal cross sections of rat brain showing changes in (a)-(c) oxy, deoxy and total haemoglobin after induced unilateral carotid artery occlusion on the left and (d)-(f) right sides (Hielscher 2005)

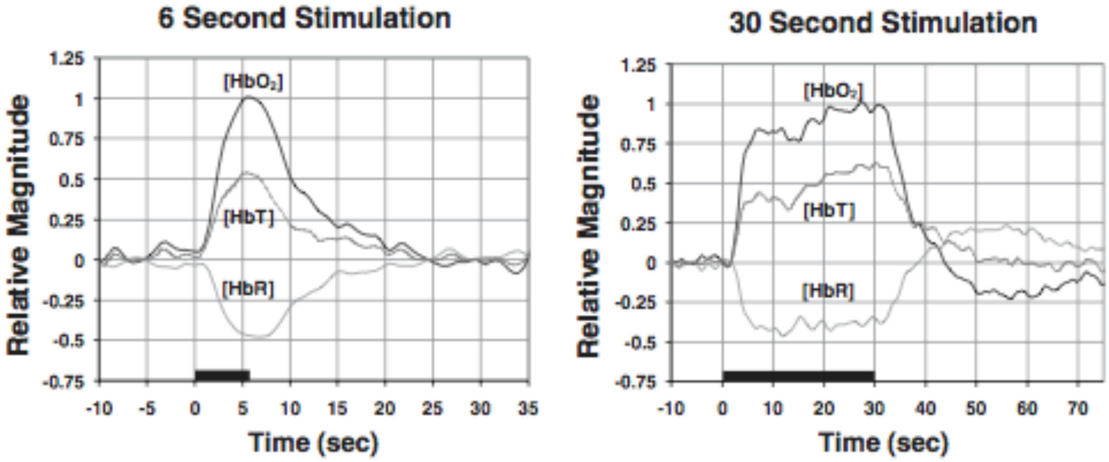


Figure 3.4 Temporal response of oxy (HbO$_2$), deoxy (HbR) and total (HbT) haemoglobin in response to electrical stimulus. The black bars represent the duration of the stimulation (Siegel et al. 2003)

Bioluminescence tomography promises to be an extremely useful tool in cancer research. Figure 3.5, for example, shows an image of a mouse in which human breast cancer cells were implanted. The cells were expressing the luciferase gene, emitting light which was used to determine the size of the tumour. This corresponded well with physical measurements (Zinn et al. 2008). Wang *et al* similarly injected a mouse with human prostate cancer cells expressing luciferase in order to observe the development of metastases (Wang et al. 2006a).

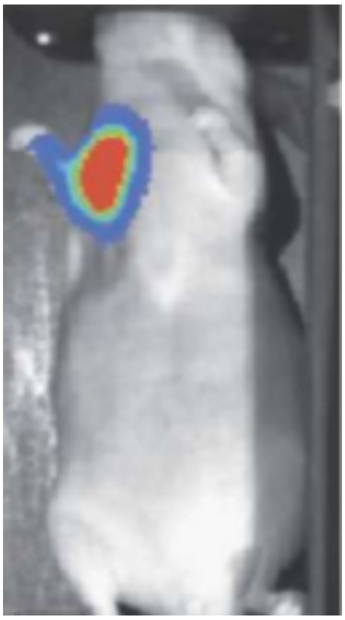


Figure 3.5 Image of a breast tumour in a mouse, expressing the luciferase gene. Light emission was found to be proportional to the tumour size (Zinn et al. 2008)

Chen *et al* used a frequency domain imaging system to reconstruct the distribution of a fluorescent marker which targets the enhanced tumour glycolysis. In doing so, they were able to locate tumours up to 2cm below the surface of the mouse with an accuracy of 2-3mm (Chen et al. 2003). Ntziachristos *et al* used fluorescent beacons to obtain three dimensional images of a protease in gliomas. They were able to identify the location of the fluorophores with high degrees of accuracy in deep tissues (Ntziachristos et al. 2002a).

The results of the studies discussed suggest that molecular tomographic imaging could be extremely beneficial in studying biological function. It could be used to observe physiology, develop models of cancer development or monitor the efficacy of drugs and therapies.

Chapter 4 - Current Medical Imaging Methods

4.1 Introduction

Optical tomography is a relatively new imaging technique and is not yet in regular clinical use. This chapter will describe some of the existing imaging methods, both anatomical and functional, that are currently used. Their advantages and disadvantages will be discussed.

4.2 X-ray Radiography and Computed Tomography

X-ray radiography was one of the earliest forms of medical imaging having been in use since as early as 1895 (Glasser 1993). Traditional x-ray radiography involves placing a film screen behind the patient to detect transmitted rays, resulting in a 2D map of the anatomy within the field. As x-ray absorption, via the photoelectric effect, is proportional to electron density, bone and other dense materials provide good contrast while soft tissue is almost transparent to x-rays.

In many cases, a 3D information is required, which is not possible with standard radiography. Early attempts at 3D imaging began with focal plane tomography in which a x-ray source and detector were moved in opposite directions either side of the patient, with each point within the patient being mapped onto the detector. Depending on the speed of motion, only one plane within the patient would remain in focus. Signals from the rest of the patient are then smeared across the detector.

More modern Computed Tomography (CT) uses a fan beam of x-rays which is rotated around the patient. Unlike conventional radiography, CT measurements do not directly result in a useful image. Instead, a series of projections is recorded at all angles around the patient. These projections can then be used to reconstruct a map of attenuation coefficients throughout the patient using the back-projection method where the attenuation profile recorded at each angle is projected back across the imaging space and summed (Buzug 2008). The basic back-projection method, however, results in a 'foggy' image containing strong artefacts, Figure 4.1. Reconstructed images can be greatly improved by using filtered back-projection, where the attenuation profiles for each angle are convolved with a filter function in fourier space, and then summed as before.

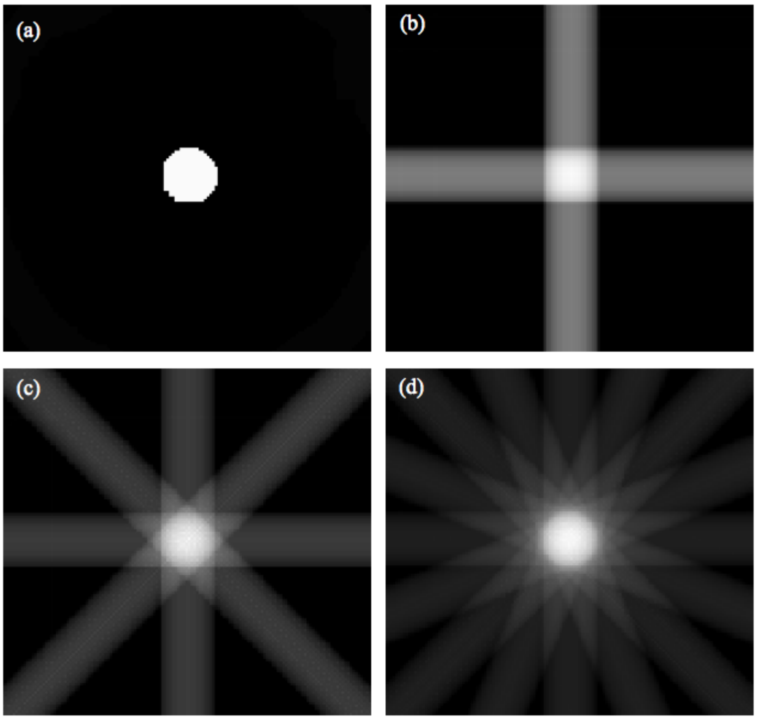


Figure 4.1 The reconstruction of an object (a) via backprojection results in star shaped artefact and a general fogging as projections (b)-(d) are summed

CT is primarily used to provide high resolution anatomical information, Figure 4.2. Some physiological information is possible, however, by monitoring the distribution of contrast agents with time. The development of desktop micro CT systems has led to in vivo, molecular imaging studies of small animals at microscopic scales (Holdsworth and Thornton 2002; Ritman 2002; Sasov 2002).

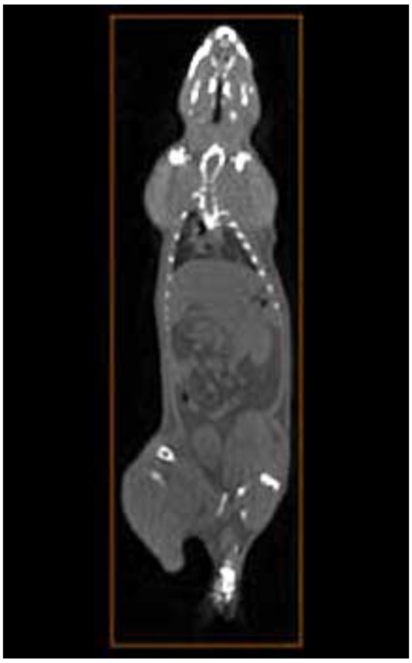


Figure 4.2 Example CT image of a small animal

One of the main strengths of x-ray CT imaging over optical tomography, is the ability to capture high resolution anatomical images with short acquisition times. The ionising nature of x-rays, however, is a major disadvantage as there is the probability of causing radiation related complications, which is not a problem with the near-infrared light used in optical tomography. Differential imaging of contrast agents, however, can only provide relative changes in physiological states. DOT, however, has the ability to provide absolute values of biological conditions such as oxygen saturation, haemoblogin concentrations or water content.

4.3 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) provides an alternative imaging modality. The principle of magnetic resonance imaging relies on the spinning nature of certain nuclei within the patient. When in pairs, sub-atomic particles tend to spin at the same rate but in opposite directions. For nuclei with even mass numbers, there is an even number of nucleons and so half spin in one direction, and half spin in the other direction; the nucleus therefore has no net spin. Nuclei with odd mass numbers, such as hydrogen, however, do have a net spin and are known as active nuclei.

When a strong external magnetic field, B_0, is applied to the active nuclei, their spins will attempt to align with the field. Most of the nuclei will align parallel with B_0, but a smaller number will align anti-parallel, resulting in a net magnetization parallel to the external magnetic field, Figure 4.3. The individual protons within the nuclei also precess around B_0 at the Larmor frequency, ω_0,which is given by

$$\omega_0 = B_0\lambda \tag{4.1}$$

where $B_0$is the external magnetic field strength and $\lambda$is the gyro-magnetic ratio.

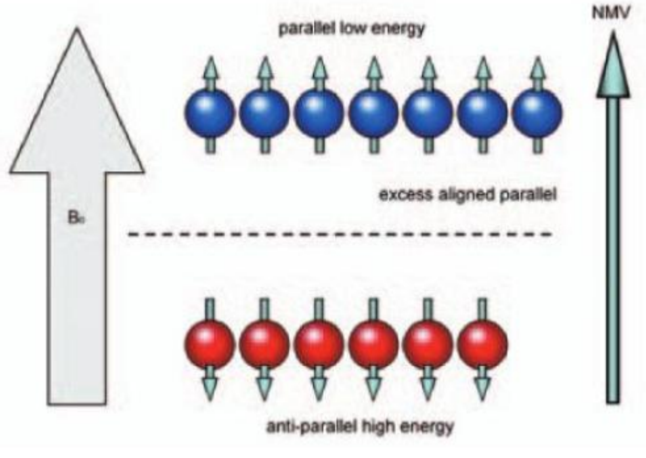


Figure 4.3 Alignment of nuclei with external magnetic field B_0 results in a net magnetization vector parallel to B_0 (Westbrook and Roth 2005)

If a radio frequency (RF) pulse with frequency ω_0 is applied perpendicular to B_0, the nuclei resonate and can gain energy causing some of the parallel spins to flip to an anti-parallel spin. This has the effect of moving the net magnetization out of alignment with B_0 and will eventually become perpendicular. When the RF pulse is turned off, the nuclei begin to return to their original state, known as relaxation, which can be detected as a voltage in a surrounding coil (Westbrook and Roth 2005).

If the external magnetic field B_0 has a gradient, i.e., its strength varies through the patient, the Larmor frequency will vary for each plane. Therefore, by applying a RF pulse with a single frequency, only one plane is activated. By collating data from each of the planes, a 3D dataset can be constructed.

The magnetic properties of haemoglobin can vary, depending on whether it is oxygenated or de-oxygenated. Oxy-haemoglobin results in a magnetic field aligned with B_0, while deoxy-haemoglobin results in a field that is anti-parallel to B_0. The perturbations in the magnetic field due to deoxy-haemoglobin is known as the blood oxygen level dependent (BOLD) signal. Functional MRI (fMRI) can be used to observe physiological function by studying the BOLD signal. The fMRI signal is often overlaid onto a structural MRI image to provide context, as shown in Figure 4.4.

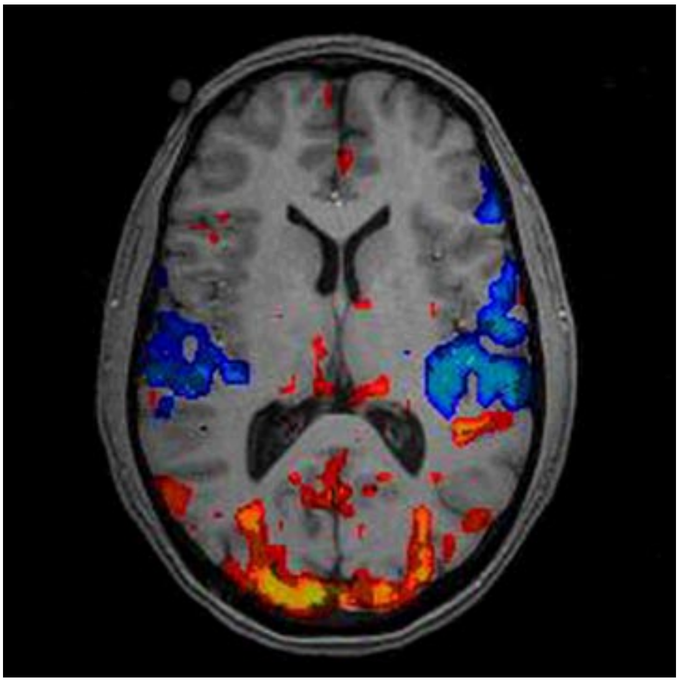


Figure 4.4 Example fMRI image overlaid on structural MRI. Red regions indicate active regions i.e. greater oxygen consumption and blue regions indicate less active regions

4.4 Radionuclide Imaging

In radionuclide imaging, clinical information is obtained by observing the distribution of a function or cell specific pharmaceutical within the body. Before being ingested or injected into the patient, the pharmaceutical is tagged with a gamma-emitting radionuclide which can be detected by an external gamma camera (Sharp et al. 2005). The detected radiation can then be used to produce either a planar image of the radiopharmaceutical distribution or a reconstructed, 3D distribution map.

Technetium-99m is a commonly used radionuclide in radionuclide imaging. It is an ideal tracer as it has a short half-life of approximately 6 hours, minimising the radiation dose to the patient. The short half-life, however, means that it is difficult to transport. Instead, molybdenum-99 generators are produced which decay into Technetium-99m with a half-life of around 67 hours. One of the major disadvantages of radionuclide imaging is the need to purchase molybdenum generators on a regular basis, which are produced in just a small number of facilities worldwide. When one of these facilities becomes inoperable, as in the case at the time of writing, the global demand for molybdenum can cause drastic shortages (Perkins and Vivian 2009).

4.4.1 Single Photon Emission Computed Tomography

Single photon emission computed tomography (SPECT) is a method of producing 3D images and has been in use since the late 1960s. The SPECT process is very similar to standard planar radionuclide imaging in that a gamma camera is used to record the emitted radiation. In SPECT, however, the gamma camera head is rotated around the subject to record projections at all angles. Modern gamma cameras are now double-headed allowing pairs of projections to be recorded simultaneously, halving the acquisition time. Image reconstruction methods, such as the filtered back projection approach used in CT, can then be used to reconstruct 3D images.

4.4.2 Positron Emission Tomography

Positron emission tomography (PET) differs from SPECT in that the radionuclide used does not directly emit gamma rays, but emits positrons instead. As the emitted positrons move away from the source, they are likely to meet an electron at which point they will be annihilated, emitting a pair of 511keV gamma rays in exactly opposite directions. A ring of detectors surrounding the patient is used to detect only coincident gamma rays that originate from a single positron annihilation. Images are again reconstructed by using filtered back-projection methods. The distance that a positron can travel before being annihilated limits the spatial resolution of PET images. An example PET image is given in

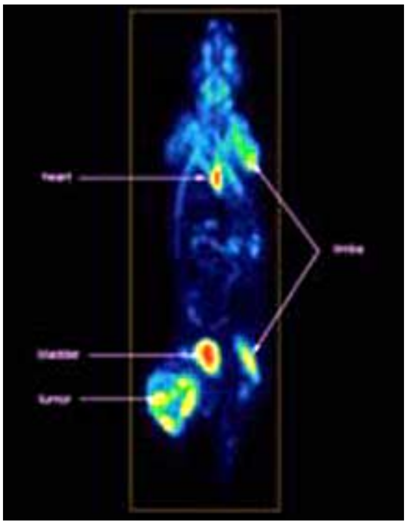


Figure 4.5 Example whole body PET scan of the same animal as in Figure 4.2

The positron source used in PET is commonly used to label fludeoxyglucose (FDG), a type of sugar, and is therefore preferentially absorbed by regions of increased metabolism. As it is known that cancer cells have a higher metabolic rate than healthy cells, the distribution of FDG can then be used to identify tumours.

4.5 Multi-modality Imaging

At present, there is no one imaging modality that can provide the perfect combination of high resolution structural imaging as well as functional information simultaneously. As such, multi-modality imaging systems are often sought.

The poor resolution of PET imaging, for example, makes their images almost un-readable to the untrained eye, Figure 4.6. As such, the majority of PET imaging is now performed using dual modality PET-CT scanners. As the patient is passed through both scanners, both structural and functional imaging is obtained for the same patient geometry and the two images can be later fused. The CT data can also be used to perform attenuation correction on the PET data. The synergy of these two imaging modalities has led to improved diagnosis (Tsukamoto and Ochi 2006).

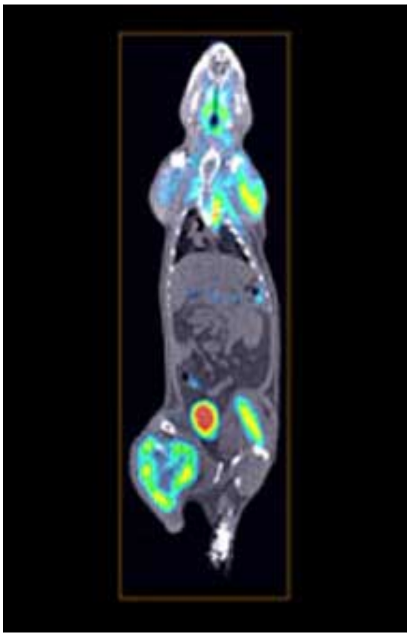


Figure 4.6 A fused PET/CT image resulting from overlaying Figure 4.5 onto Figure 4.2

A number of MRI – optical tomography systems have also been developed. In these systems, the MRI data not only provides structural information, but it can also be used to improve the accuracy of the optical data (Davis et al. 2008; Thayer et al. 2010). The use of such structural information in image reconstruction is discussed further in chapter 6. A number of Optical-radionuclide hybrid systems have also been developed for small animal molecular imaging (Huber et al. 2001; Luker et al. 2002; Celentano et al. 2003; Ray et al. 2003; Alexandrakis et al. 2005; Proud et al. 2005).

To assist in the development of multi-modality imaging, several mouse atlases, such as the digimouse (Dogdas et al. 2007), have been developed. The digimouse was constructed using CT and cryosection data. The various organs were then segmented and labelled with optical properties and x-ray attenuation coefficients, Figure 4.7. This allows image reconstruction for any of the imaging modalities to be performed using the same model. A comparison of the imaging modalities performed using the digimouse model is shown in Figure 4.8.

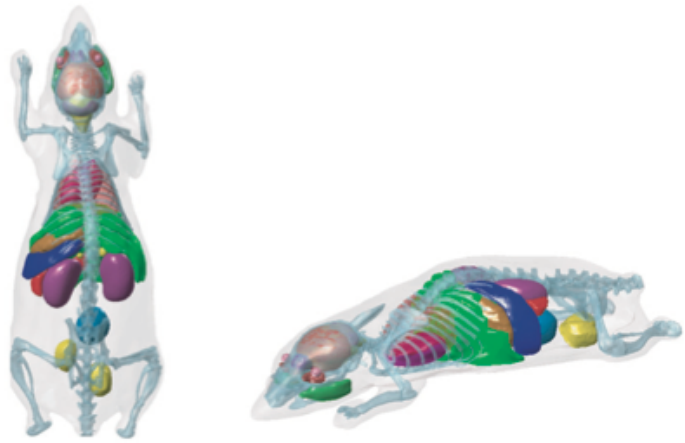


Figure 4.7 The Digimouse model, an atlas of the mouse showing segmented organs such as the brain, eyes, lungs, liver, stomach etc… labelled with optical absorption and scatter properties and x-ray attenuation coefficients (Dogdas et al. 2007)

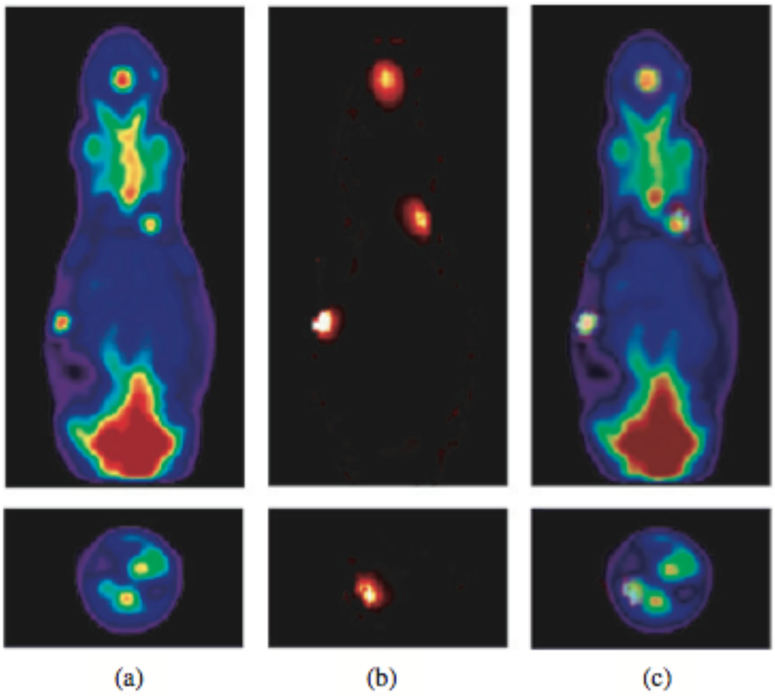


Figure 4.8 Comparison of (a) PET and (b) bioluminescence reconstructions based on a data set simulated using the digimouse. (c) shows a fusion of the PET and bioluminescence reconstructions

4.5 Discussion

Imaging modalities can be broadly classified into one of two groups: structural and functional. Functional modalities such as CT and MRI are capable of providing high spatial resolution images of anatomy. MRI can give good soft-tissue contrast but it requires very expensive hardware. The strong magnetic field also means that it is unsuitable for patients with pacemakers and its use is therefore limited. X-ray CT is less expensive than MRI, but it uses ionising radiation which can pose a risk to the patient as well as hospital staff. The hardware required for both MRI and CT imaging is very large and immobile. This means that imaging can only be performed in a fixed location, which makes it unsuitable for continuous monitoring of patients.

Functional imaging modalities provide the clinician with information about the physiology within the imaged region. This is often achieved by imaging the uptake of some form of tracer which, in the case of SPECT and PET, takes the form of a possibly damaging radionuclide. As well as being potentially harmful, the radionuclide tracers introduce a logistical problem as supplies can be variable. For example, in recent years, there has been a shortage in the supply of Tc^{99}, a commonly used radionuclide. As a result, many potentially vital scans were delayed or cancelled. DOT allows similar functional information to be obtained but without the need for radioactive tracers. Maps of optical properties or chromophore concentrations provide information on physiology and can be obtained by tomographic measurements alone. However, the use of fluorescent or bioluminescent agents allows optical imaging at the molecular scale.

As with MRI and CT, radionuclide imaging is limited to a fixed location the detectors used are large. This is not the case with DOT. The hardware required for an optical imaging system is relatively simple and it is possible to be mounted onto a transportable trolley. This makes it possible to perform bedside imaging and, as it uses non-ionising radiation, it could be used for regular monitoring.

As DOT suffers from limited spatial resolution, it is unlikely to become a standalone replacement for existing imaging modalities. However, as it is relatively inexpensive and can provide absolute functional information with a transportable imaging system, it is likely to be a useful addition to the existing range of imaging systems.

Chapter 5 - The Forward Problem

5.1 Introduction

The process of recovering images from DOT boundary measurements consists of two problems, Figure 5.1. The forward problem aims to calculate the boundary data that would be measured for a given set of optical properties within a known geometry and measurement system. Conversely, the inverse problem aims to calculate the optical property distribution within a known geometry given a set of boundary measurements. Due to the non-linear nature of the forward problem, see chapter 6, the inverse problem is generally solved by iteratively solving the forward model with the aim of minimizing the discrepancy between the resultant predicted data and experimental data. As such, model based image reconstruction algorithms are heavily dependent on an accurate forward model.

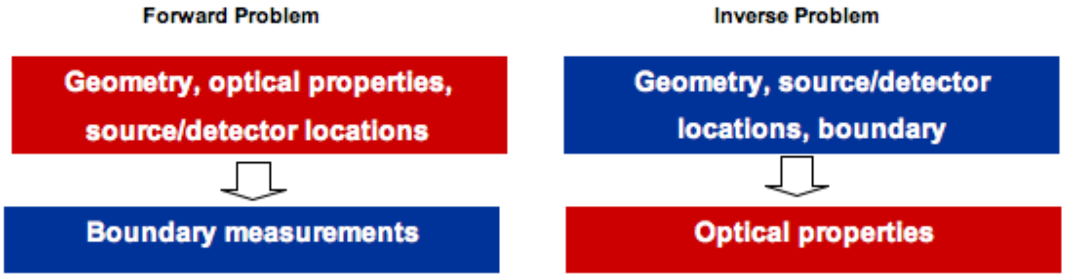

Forward Problem
Geometry, optical properties, source/detector locations
Boundary measurements
Inverse Problem
Geometry, source/detector locations, boundary
Optical properties

Figure 5.1 Illustration of the forward and inverse problems. Known information is listed in the upper boxes whilst the data to be recovered is listed in the lower boxes.

This chapter will focus on the forward problem with the aim of developing a model for accurate light propagation in tissue which will provide an improvement over existing models in cases where the Diffusion Approximation is not valid. The chapter consists of two parts, the first introducing modelling of photon transport through soft tissue, and the second discussing numerical techniques used to solve the partial differential equations introduced in part 1. The inverse problem is discussed later in chapter 6.

5.1 Modelling Photon Transport in Soft Tissue

5.1.1 The Radiative Transport Equation

By modelling light as particles, the propagation of light through biological tissue can be accurately described by the Radiative Transport Equation (RTE), a linearized version of the Boltzmann equation (Case and Zweifel 1967). Transport based problems have been extensively studied in a wide range of fields such as neutron transport in nuclear reactor problems and radiative transfer in stellar atmospheres and many other problems. By treating light as particles, however, the RTE neglects any wave effects and so light with wavelengths much smaller than the size of the subject of interest must be used. The RTE also assumes that all particles travel at the same speed and thus the refractive index of the subject must be homogenous (Case and Zweifel 1967; Gibson et al. 2005). It is possible, however, to extend the RTE to include spatially varying refractive index (Premaratne et al. 2005).

The RTE for frequency domain problems is

$$\begin{aligned}\frac{i\omega}{c} I(\mathbf{r},\omega,\hat{\mathbf{s}}) + \hat{\mathbf{s}}\cdot \nabla I(\mathbf{r},\omega,\hat{\mathbf{s}}) + (\mu_a + \mu_s) I(\mathbf{r},\omega,\hat{\mathbf{s}}) \\ = \mu_s \int_{4\pi} f(\hat{s},\hat{s}') I(\mathbf{r},\omega,\hat{\mathbf{s}}') d^2\hat{\mathbf{s}}' + q(\mathbf{r},\omega,\hat{\mathbf{s}})\end{aligned} \tag{5.1}$$

where $I(\mathbf{r},\omega,\hat{\mathbf{s}})$ is the radiance at point **r**, frequency **ω** and in the direction of unit vector **ŝ.** The μ_a and μ_s parameters are the absorption and scattering coefficients respectively, and c is the speed of light in the medium, which is dependent on the refractive index, often assumed to be homogeneous within the medium. The $f(\hat{\mathbf{s}},\hat{\mathbf{s}}')$term is the scattering phase function, which characterizes the intensity of a beam which is scattered from direction $\hat{\mathbf{s}}'$into direction $\hat{\mathbf{s}}$ (Arridge and Hebden 1997). Equation 5.1 is most accurately solved in conjunction with the vacuum boundary condition, which simply dictates that all photons arriving at the external boundary are completely absorbed and lost. The RTE for the time domain can be easily obtained by replacing the $i\omega/c$ terms with $\partial/\partial t$, equation 5.2

$$\begin{aligned}\frac{\partial}{\partial t}I(\mathbf{r},t,\hat{\mathbf{s}}) + \hat{\mathbf{s}}\cdot\nabla I(\mathbf{r},t,\hat{\mathbf{s}}) + (\mu_a+\mu_s)I(\mathbf{r},t,\hat{\mathbf{s}})\\ = \mu_s\int_{4\pi} f(\hat{s},\hat{s}')I(\mathbf{r},t,\hat{\mathbf{s}}')d^2\hat{\mathbf{s}}' + q(\mathbf{r},t,\hat{\mathbf{s}})\end{aligned} \tag{5.2}$$

The RTE can essentially be thought of as a balance equation between the mechanisms that lead to an increase in photon density and those that decrease photon density. A very small number of photons will pass through the medium undergoing no or few scattering events, resulting in the streaming term $\hat{\mathbf{s}}\cdot\nabla I(\mathbf{r},\hat{\mathbf{s}},\omega)$. The majority of photons will undergo multiple scattering events and follow a random walk until they are either absorbed or pass through the boundary of the medium$(\mu_a+\mu_s)I(\mathbf{r},\omega,\hat{\mathbf{s}})$. Scattering can also lead to increases in photon density if scattered photons are re-directed back towards the position under question, expressed as$\mu_s\int_{4\pi} f(\hat{s},\hat{s}')I(\mathbf{r},\omega,\hat{\mathbf{s}}')d^2\hat{\mathbf{s}}'$. The remaining term, $\frac{i\omega}{c}I(\mathbf{r},\omega,\hat{\mathbf{s}})$, accounts for variations in photon density due to the modulation of the external light source. These events can be schematically generalized as shown in Figure 5.2.

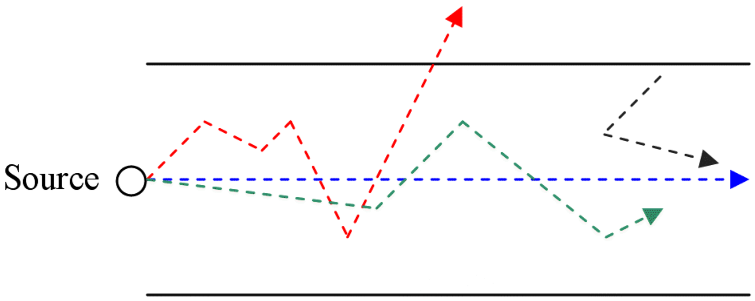


Figure 5.2 Schematic of photon-tissue interaction mechanisms. Blue – Small number of streaming photons pass through un-scattered; Red – scattering causes photon to cross boundary, leaving the domain; Green – photon is absorbed; Black – photons scattered back towar

The phase function, $f(\hat{\mathbf{s}},\hat{\mathbf{s}}')$, which describes the amount of light scattered from the direction of unit vector $\hat{s}$ to $\hat{s}'$, can be thought of as a probability distribution and therefore normalised such that

$$\int_{4\pi} f(\hat{\mathbf{s}},\hat{\mathbf{s}}')d\omega = 1 \tag{5.3}$$

The phase function then becomes the probability that a photon will be scattered from one direction to another, as discussed in chapter 2.

5.1.1 Approximations to the Radiative Transport Equation

One of the complexities in solving the radiative transport equation lies in the integration over the angular terms. The calculations can be simplified by discretising the angular terms. One of the methods of doing this, known as the discrete ordinates (S_N) approximation, Figure 5.3b, is to divide all of the possible angles into a finite number of components (eg 360 components, each of 1°) (Case and Zweifel 1967). Although the S_N approximation does simplify the calculation of the radiative transport equation, a system of N(N+2) equations (Klose and Larsen 2006), where N is the number of discrete ordinates) is required, leading to significant increases in computation time.

For example, a three dimensional model with 20,000 unknowns and 360 angular components would require 7.2×10^{6} unknowns to be calculated.

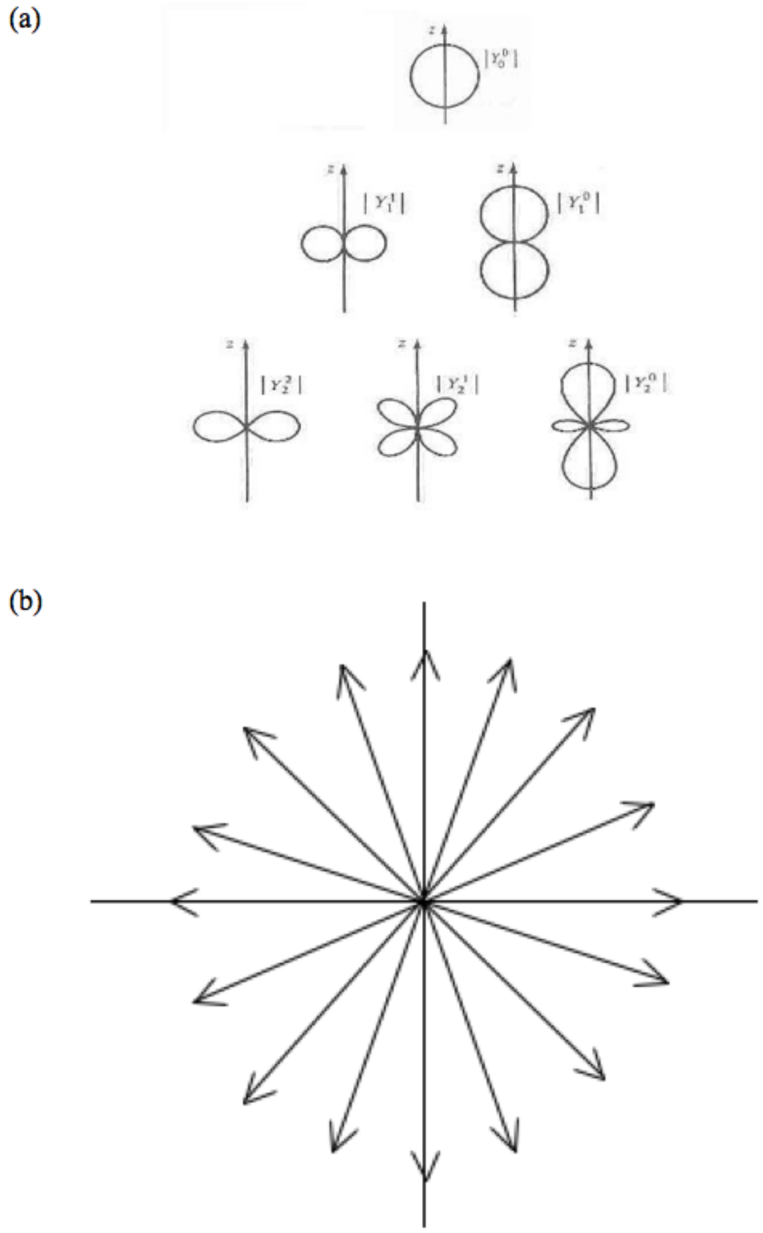


Figure 5.3 Angular terms can be represented using (a) spherical harmonics of order Y_l^m or (b) discrete ordinates method

The spherical harmonics (P_N) approximations provide an alternative method of angular discretisation. Spherical harmonics, Y_l^m, are the angular solution to the Laplace equation in spherical coordinates and could be used to represent directional terms as represented in Figure 5.3a. The solution of the radiative transfer equation could be simplified by expanding it into a set of spherical harmonics. The radiance and source terms would then become:

$$I(\mathbf{r},\hat{\mathbf{s}},t) = \sum_{l}^{\infty}\sum_{m=-l}^{l}\left(\frac{2l+1}{4\pi}\right)^{1/2}\psi_{l,m}(\mathbf{r},t)Y_l^m(\hat{\mathbf{s}}) \tag{5.6}$$

$$q(\mathbf{r},\hat{\mathbf{s}},t) = \sum_{l}^{\infty}\sum_{m=-l}^{l}\left(\frac{2l+1}{4\pi}\right)^{1/2}q_{l,m}(\mathbf{r},t)Y_l^m(\hat{\mathbf{s}}) \tag{5.7}$$

where $Y_l^m(\hat{s})$ is a spherical harmonic function of degree l and order m. Expanding to a higher order would give a better representation of the angular distribution but would result in additional terms to be solved. The number of equations required by a P_N approximation, however, grows with $(N+1)^2$ where N is the number of Legendre Polynomials (Case and Zweifel 1967)

5.1.3 The Diffusion Approximation

The diffusion approximation (DA), a P_1 approximation, makes the assumption that the light distribution within the region of interest is isotropic. This tends to be true when scattering effects within a region of interest are dominant over absorption effects i.e. $\mu_s' >> \mu_a$, where $\mu_s' = \mu_s(1-g)$.

The diffusion approximation can be derived from the radiative transport equation by defining the radiance within the medium as an isotropic fluence rate, $\Phi(\mathbf{r},t)$, and a small forward flux, $J(\mathbf{r},t)$.

$$\Phi(\mathbf{r},t) = \iint_{4\pi} I(\mathbf{r},\hat{\mathbf{s}}',t)d\hat{\mathbf{s}} \tag{5.8}$$

$$J(\mathbf{r},t) = \iint_{4\pi} I(\mathbf{r},\hat{\mathbf{s}}',t)\mathbf{r}d\hat{\mathbf{s}} \tag{5.9}$$

The radiance can then be expanded into first order spherical harmonics. Integrating over all solid angle then gives

$$I(\mathbf{r},t,\hat{\mathbf{s}}) = \frac{1}{4\pi}\Phi(\mathbf{r},t) + \frac{3}{4\pi}J(\mathbf{r},t)\cdot\hat{\mathbf{s}} \tag{5.10}$$

By substituting the fluence rate, $\Phi(\mathbf{r},t)$, and the forward flux, $J(\mathbf{r},t)$ into equation 5.10 and integrating over all solid angles, a continuity equation for the photon flux can be reached:

$$\frac{1}{c}\frac{\partial\Phi(\mathbf{r},t,\hat{\mathbf{s}})}{\partial t} + \nabla\cdot J(\mathbf{r},t,\hat{\mathbf{s}}) = -\mu_a\Phi(\mathbf{r},t,\hat{\mathbf{s}}) + S_0(\mathbf{r},t) \tag{5.11}$$

By multiplying by $\hat{\mathbf{s}}$ and integrating over all angles leads to:

$$\frac{1}{c}\frac{\partial J(\mathbf{r},t)}{\partial t} = -\frac{1}{3}\nabla\Phi(\mathbf{r},t) - \frac{1}{3\kappa}J(\mathbf{r},t) \tag{5.12}$$

where $\kappa = 1/[3(\mu_a + \mu_s^{'})]$ is the diffusion coefficient; $\mu_s^{'}$ is the reduced scattering coefficient and $\mu_s^{'} = (1-g)\mu_s$, where g is the anisotropic factor (the average cosine of all scattering angles, ≈0.9 in tissue). Since the time derivative of the flux is, under most conditions, significantly less than collision frequency, equation 5.12 can be simplified, leading to Fick's law:

$$J(\mathbf{r},t) = -\kappa\nabla\Phi(\mathbf{r},t) \tag{5.13}$$

Substituting equation 5.13 into equation 5.11 then leads to the diffusion equation :

$$\frac{1}{c}\frac{\partial\Phi(\mathbf{r},t)}{\partial t} - \nabla\cdot\kappa\nabla\Phi(\mathbf{r},t) + \mu_a\Phi(\mathbf{r},t) = S_0(\mathbf{r},t) \tag{5.14}$$

in the time domain, or

$$\frac{i\omega}{c}\Phi(\mathbf{r},\omega) - \nabla\cdot\kappa\nabla\Phi(\mathbf{r},\omega) + \mu_a\Phi(\mathbf{r},\omega) = S_0(\mathbf{r},\omega) \tag{5.15}$$

in the frequency domain where ω is the source modulation frequency. The DA is currently the most widely used light propagation models. It can be accurately applied when the condition $\mu_a << \mu_s'$ is true and is therefore commonly applied to problems with large tissue geometries such as the human head or breast (Ntziachristos et al. 2000; Dehghani et al. 2003c; Choe et al. 2005; Enfield et al. 2007).

5.1.4 Beyond the Diffusion Approximation

Whilst the DA can be accurately applied to many biological tissues, there are several applications in which the DA is not accurate. In small geometries of less than a few scattering lengths, for example, the light distribution within the region of interest will be forward biased due to the presence of the light source. This effect is also present in large geometries in the regions near to the source. A recent paper by Tarvainen *et al* (Tarvainen et al. 2005), proposed a radiative-transfer – diffusion hybrid model for these problems. This hybrid uses the RTE to model the light distribution in the vicinity of the source and the DA elsewhere. A Dirichlet boundary condition is then used to model the interface between the radiative-transport and diffusion regions. This method has been found to produce accurate results with little increase in computation time. However, as this method uses analytical solutions of the radiative transfer and diffusion equations, it is only capable of handling basic geometries.

The accuracy of the DA is also known to be limited in regions of low or zero scattering. Such conditions are likely to be encountered when imaging the brain in the clear cerebrospinal fluid (CSF). A possible solution to this problem was suggested by Firbank *et al* (Firbank et al. 1996) and later by Arridge *et al* (Arridge et al. 2000b), which used a radiosity-diffusion hybrid model which uses the diffusion equation in highly scattering regions and radiosity equations in non-scattering regions (Riley et al. 2000). The beam intensity used in the non-scattering regions is calculated from the outgoing radiance from the scattering region and is then used as the source term for the next scattering region.

The search for higher ordered light propagation models is particularly important for whole body imaging of small animals. As well as the problem of small geometries, the presence of highly vascularised organs, such as the liver or heart, within the field of view result in regions of strong absorption at wavelength between 500 – 650 nm. The use of bioluminescent markers in small animals is becoming increasingly widespread and introduce further limitations to the DA, due to signal emission at these lower wavelengths where the absorption of deoxy-haemoglobin is very high.

In Bioluminsecent imaging, a light emitting protein is attached to a pharmaceutical which is absorbed by specific types of cells. This allows the monitoring of disease progression or drug effectiveness. The most commonly used light emitting protein is luciferase, which has a peak emission at approximately 560nm, see Chapter 3. This corresponds to a region of strong absorption due to haemoglobin and so the DA is no longer valid as the condition $\mu_a << \mu_s^{'}$ does not apply. As such, there is a need for higher ordered approximations to the RTE. Klose *et al*, for example, have proposed a method that utilises the SP_3 approximation (see chapter 6) for the forward model. The measurements of the exiting bioluminescent light are then co-registered with CT & MRI data which enables the use of prior information (Klose and Beattie 2008; Klose et al. 2010). In this study, the forward model was solved using the Finite Difference Method and a blocking off method was used to enable curved boundaries to be modelled. An Finite Element Method based solver, however, would be beneficial for solving arbitrary complex geometries.

5.2 Solution Methods

The solution of the RTE is a highly complex problem which has been the subject of much research over the past 10 years. This section introduces some of the analytical and numerical methods of solving the RTE and some of the practical considerations relating to them.

5.2.1 Analytical Models

An analytical solution to the RTE initially appears to be an attractive approach. In reality, analytical solutions exist only for very simple geometries e.g. one-dimensional and so are of limited use in image reconstruction. However, analytical solutions for a range of geometries, such as the infinite-space, infinite slab or finite cylinders, have been published for the DA. Solutions also exist for simple geometries containing a single spherical inclusion (Boas et al. 1994). Existing solutions can, however, be extended to more complex geometries such as layered slabs (Martelli et al. 2002). Such models have been used as forward models in image systems involving geometries that can be approximated as layered slabs or infinite half-spaces (Culver et al. 2003c; Li et al. 2004).

5.2.2 Stochastic Models

Light propagation through tissue can be accurately modelled by using statistical methods. The Monte Carlo (MC) method, for example, models the path of individual photons as they propagate through the object of interest.

Figure 5.4 outlines the basic steps of a typical MC model, and example of which has been used for model verification in Chapter 7. Initially, a grid of 'bins' in which photons are permitted to land is defined across the geometry being modelled. The optical properties, and therefore photon transport rules, are defined at each of these bins a series of probability density functions (PDFs). Roulette probabilities are also defined to determine the likelihood of a photons being absorbed or scattered.

Photons begin their walk through the geometry at the designated source point and are allocated a 'weight'. A random number generator is then used to sample the optical property PDFs to determine the direction and length of the initial step. The photon's new location is then calculated. If it falls outside the boundary of the geometry, the path length is truncated so that it lands on the boundary. If the angle of incidence to the boundary is below the critical angle, the new direction is calculated and the photon is moved along this angle for the remainder of the initial step length. If the angle of incidence is lower than the critical angle, however, Fresnel's law is used to determine the reflectivity coefficient. This is then used to determine the reduction in the photon's 'weight' and the amount recorded as exitance.

At the new photon location, the photon weight is reduced to account for absorption. The weight reduction is then added to the fluence in the bin on which the photon has landed. If at any point the weight of the photon falls below a pre-determined threshold, a random number generator is used to decide whether the photon continues on its path or is totally absorbed. When the weight falls to zero, either due to absorption or scattering across the boundary, the photon path is ended and a new photon is launched (Wang et al. 1995). A good introduction to Monte Carlo methods is provided by the

Oregon Medical Laser Center website (Prahl 2007) which gives solutions to problems of increasing complexity.

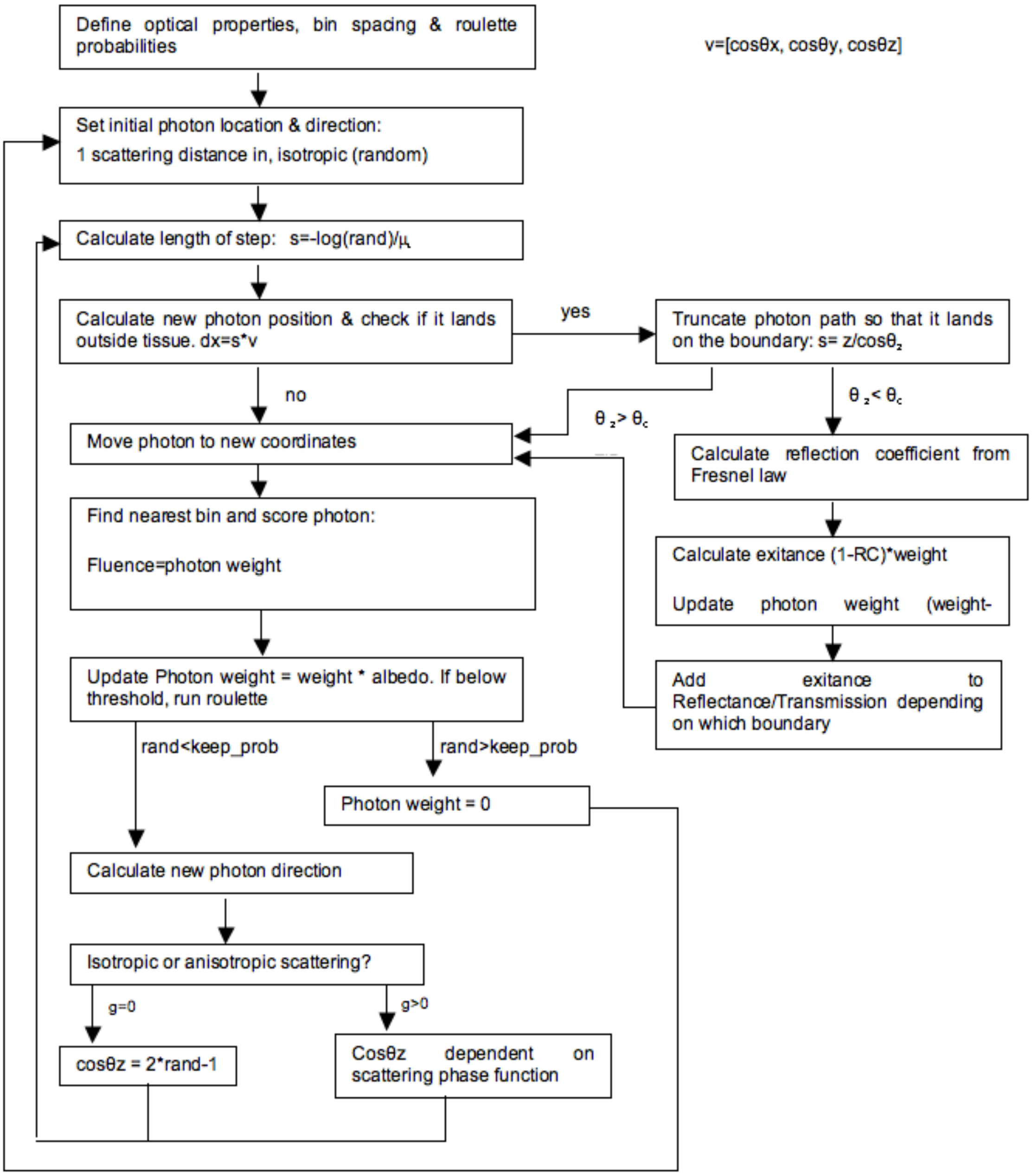
With sufficiently large number of photons, MC models prove to be highly accurate. The requirement of large numbers of photons, however, leads to lengthy computation times. As such, MC models are have been commonly used as validation for faster numerical models in the absence of experimental data (Dehghani et al. 2003a) or for problems in which standard models are inadequate, e.g. non-diffuse problems (Boas et al. 2002). A number of recent reports, however, have introduced parallelised Monte Carlo algorithms that can run on the computer's GPU (Fang and Boas 2009; Ren et al. 2010). Even with a low cost-graphics card, this has improved computation times by over 300 compared with traditional CPU based calculations.

5.2.3 Numerical Methods

A number of numerical solution techniques exist in which the spatial terms of the forward model are discretised in order to simplify the problem and are commonly used in engineering problems. In this section, some of the methods more commonly used in optical imaging will be discussed. The methods are described in terms of the RTE but are equally applicable to the DA.

5.2.3.1 Finite Difference Method

The finite difference method (FDM) approximates the solution to the RTE by replacing derivative terms with difference quotients which are approximately equivalent. The difference quotients can be defined in terms of either the forward difference, equation 5.16(a), backwards difference, equation 5.16(b) or the central difference, equation 5.16(c).



Define optical properties, bin spacing & roulette probabilities
v=[cosθx, cosθy, cosθz]
Set initial photon location & direction:
1 scattering distance in, isotropic (random)
Calculate length of step: s=-log(rand)/μt
Calculate new photon position & check if it lands outside tissue. dx=s*v
yes
Truncate photon path so that it lands on the boundary: s= z/cosθz
no
θz> θc
θz< θc
Move photon to new coordinates
Calculate reflection coefficient from Fresnel law
Find nearest bin and score photon:
Fluence=photon weight
Calculate exitance (1-RC)*weight
Update photon weight (weight-
Add exitance to Reflectance/Transmission depending on which boundary
Update Photon weight = weight * albedo. If below threshold, run roulette
rand<keep_prob
rand>keep_prob
Photon weight = 0
Calculate new photon direction
Isotropic or anisotropic scattering?
g=0
g>0
cosθz = 2*rand-1
Cosθz dependent on scattering phase function

Figure 5.4 Schematic flowchart detailing the general process used in Monte Carlo algorithms for a model with scattering coefficient μ_s, absorption coefficient μ_a anisotropy coefficient g where v defines the direction of travel of a photon.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{5.16a}$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h} \tag{5.16b}$$

$$f'(x) \approx \frac{f(x+\tfrac{1}{2}h) - f(x-\tfrac{1}{2}h)}{h} \tag{5.16c}$$

Where f(x) is the solution at spatial location x and h is the uniform spatial distance. The FDM requires the domain of interest to be discretised into a uniform grid of points as demonstrated in Figure 5.5. The RTE can then be solved at each point of the grid using one of the above difference equations in place of the derivative terms. An approximate solution to the RTE over the whole domain can then be found by collating each of the difference equations into a set of matrices of size number of nodes × number of nodes. As each node is dependent only on its neighbouring nodes, the matrices involved tend to be sparse and banded.

The requirement of a uniform grid for the FDM limits its ability to accurately model complex geometries. In its basic form, the FDM is limited to rectangular or cubic geometries. The advantages of the FDM are that it is relatively easy to implement. The FDM has been used to solve the forward problem in both transport and diffusion based problems (Hielscher et al. 1998; Hielscher et al. 1999; Culver et al. 2003a).

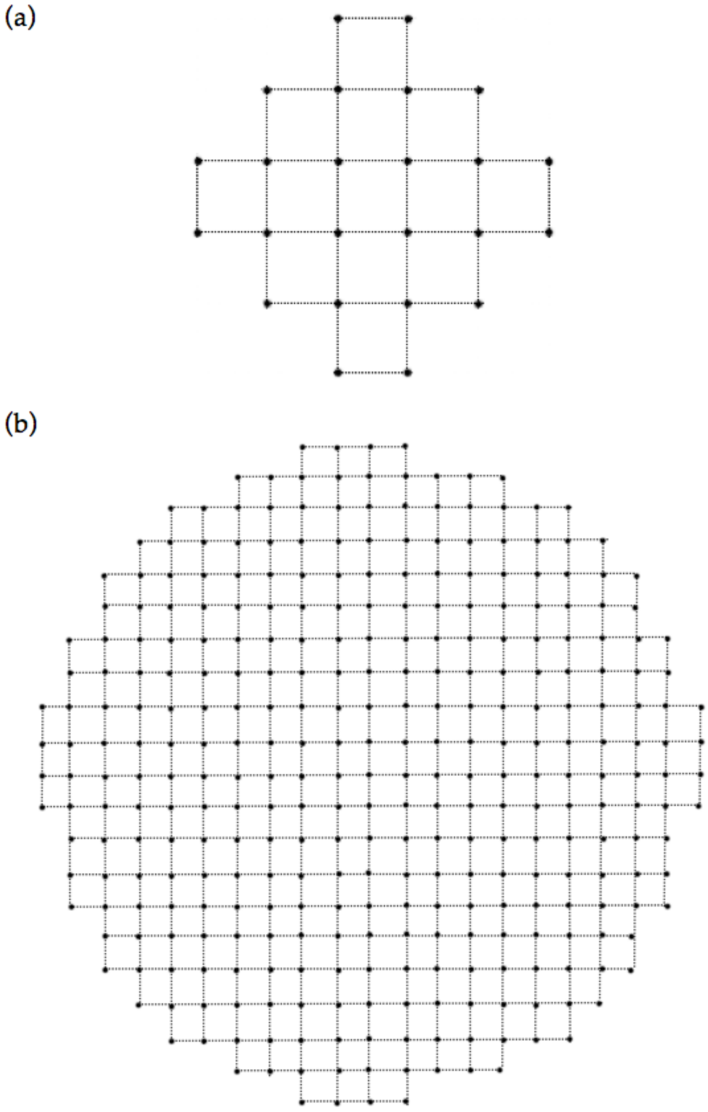


Figure 5.5 Example of a FDM discretisation of a circle in both (a) coarse and (b) fine resolutions

5.2.3.3 Finite Element Method (FEM)

The finite element method can be used to solve the forward problem over a domain Ω, by discretising it into a series of D elements connected by V vertex nodes, Figure 5.6. The fluence, $\Phi(\mathbf{r})$ is then approximated by the piecewise continuous polynomial function

$$\Phi^h(r,w) = \sum_i^V \Phi_i u_i(r)\Omega^h \tag{5.17}$$

where Ω^h is a finite dimensional subspace spanned by a shape function $u_i(r)$. The shape functions describe how the function $\Phi(\mathbf{r})$ varies over each element and can take several forms, Figure 5.7. A simple piecewise constant shape function, for example, assumes that there is no variation in the function Φ_i over the elements. This, however, results in a discontinuous field $\Phi^h(\mathbf{r})$. Alternatively, a piecewise linear shape function allows the function to vary linearly between nodes of an element. Higher ordered elements can also be used which have the effect of smoothing the solution (Schweiger and Arridge 2003).

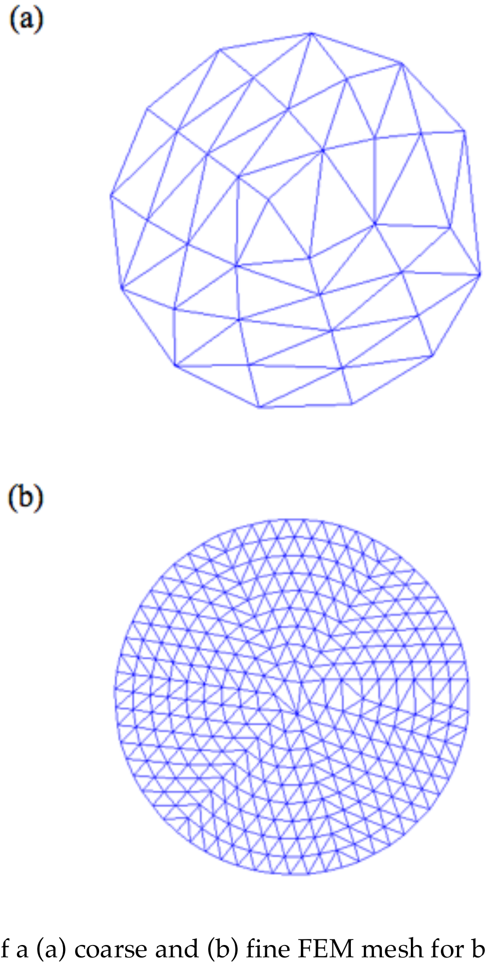


Figure 5.6 Example of a (a) coarse and (b) fine FEM mesh for basic circular geometry

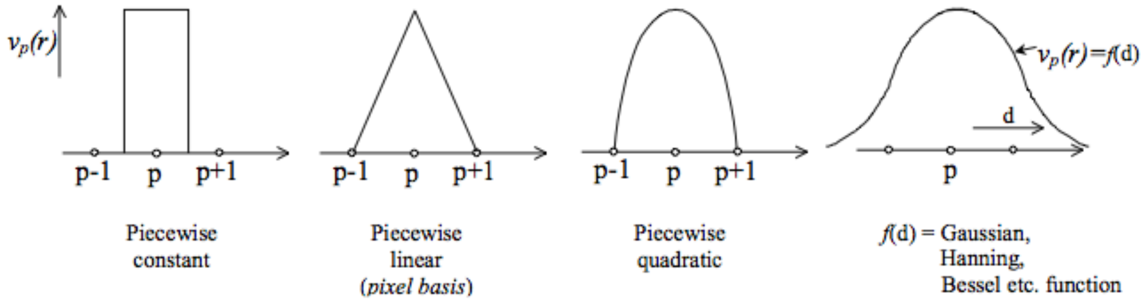


Figure 5.7 Representation of the various types of FEM shape functions (Hillman 2002)

The FEM formulation of the Diffusion Approximation, for example, can be found by multiplying equation 5.14 (for time domain) or equation 5.15 (for frequency domain) by a test function u_j and integrating over Ω

$$\int u_j \left[\frac{1}{c} \frac{\partial \Phi(\mathbf{r},t)}{\partial t} - \nabla \cdot D \nabla \Phi(\mathbf{r},t) + \mu_a \Phi(\mathbf{r},t) \right] d\Omega = \int u_j S_0(\mathbf{r},t) d\Omega \tag{5.18}$$

Integrating by parts and re-arranging then leads to the matrix equation

$$(K + C)\Phi + B\Phi = Q \tag{5.19}$$

where the elements of the matrices K, C, B, and Q are defined as

$$K_{i,j} = \int \kappa(r) \nabla u_j(r) \cdot \nabla u_i(r) d\Omega; \tag{5.20}$$

$$C_{i,j} = \int \mu_a(r) u_j(r) \cdot u_i(r) d\Omega; \tag{5.21}$$

$$B_{i,j} = \int \frac{i\omega}{c} u_j(r) \cdot u_i(r) d\Omega; \tag{5.22}$$

$$Q_j(r,\omega) = \int u_j(r) q_0(r,\omega) d\Omega \tag{5.23}$$

The K, C and B matrices are V by V elements in size where V is the number of nodes in the FE mesh. These matrices tend to be sparse and banded as entries are non-zero only for nodes that are part of the same element.

Whilst the FEM elements can take any shape, they generally take the form of triangles for two dimensional problems or tetrahedrals for three dimensional problems. The use of triangular elements enables the FEM to model complex geometries accurately. Unlike in the FDM, the FEM mesh does not require the nodes to be equally spaced. The accuracy of the mesh in regions in which rapid changes are expected can be improved by increasing mesh resolution in these areas whilst a coarse mesh is used elsewhere for computational efficiency.

Due to its ability to efficiently and accurately model arbitrary geometries, the FEM has been widely used in the forward problem of Optical Tomography (Arridge et al. 1993; Arridge and Schweiger 1995; Jiang and Paulsen 1995; Gao et al. 1998; Jiang 1998; Dehghani et al. 2003c).

5.2.3.3 Boundary Element Method

The boundary element method (BEM), is a method of solving PDEs that have been formulated in the form of boundary integrals. The main advantage of the BEM over the FEM or FDM is that it requires only the boundary of the domain to be discretised, Figure 5.8. This allows higher mesh resolutions with no additional computational cost. The lack of internal elements, however, limits the BEM to problems involving homogeneous distributions of optical properties, or a very small number of homogeneous regions.

One of the drawbacks of the BEM is that it tends to involve fully populated system matrices. The FEM and FDM, by contrast, involve sparse, banded matrices. For high mesh resolutions, the storage requirements of BEM matrices can rapidly increase as can computation times.

The BEM has been applied to region-based DA problems in which the imaging volume is divided into piecewise constant regions determined using prior information (Zacharopoulos et al. 2006; Srinivasan et al. 2007).

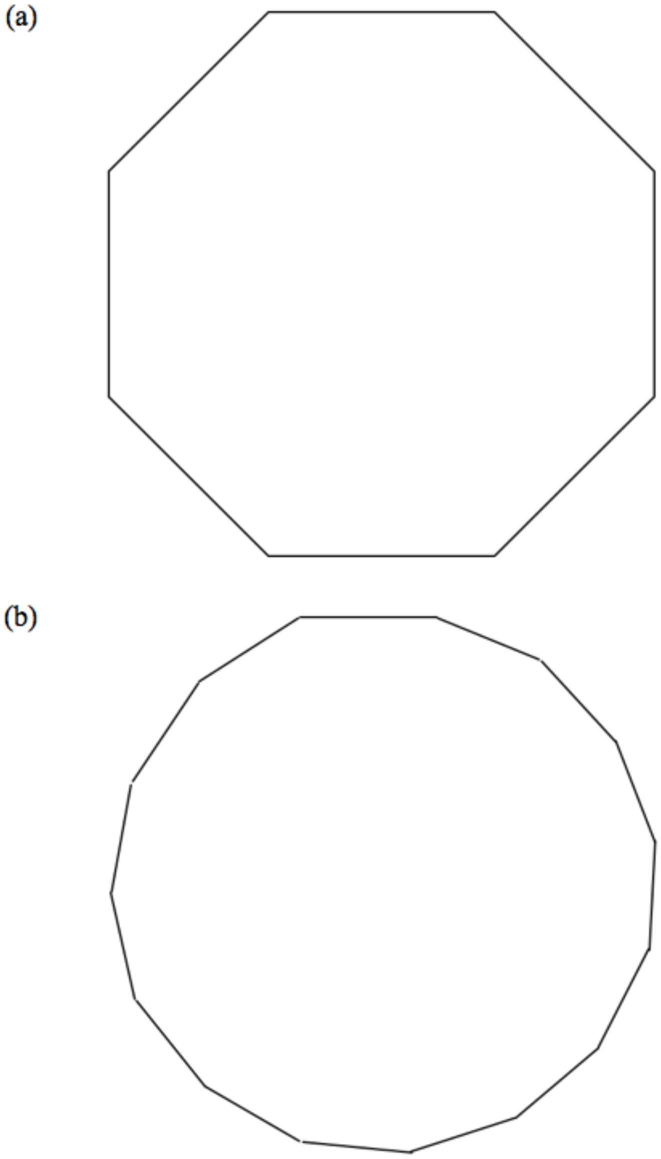


Figure 5.8 Example of a (a) coarse and (b) fine BEM mesh for basic circular geometry.

5.3 Existing Models

The majority of light propagation models are currently based on the DA which has been solved using a range of numerical methods. The NIRFAST package, for example, contains a FEM based frequency domain DA solver originally developed by Dehghani *et al* (www.nirfast.org). The source is modelled as a distributed Gaussian which aims to model the intensity profile at the end of an optic fibre. The DA, however, assumes that the source is spherically isotropic and good matches between theory and experimental data have been found when the source is moved one scattering length into the domain. The boundary conditions are modeled by a Robin type condition. The TOAST package, which also uses the FEM, was originally developed at UCL to solve the time-domain DA, although can now do frequency domain too (Schweiger et al. 1995; Schweiger 2008).

The University of Pennsylvania group have used the FDM to solve the frequency domain DA in a study of 3D breast imaging (Culver et al. 2003a). By compressing the breast with a pair of parallel plates, a slab geometry was created, which makes the FDM a suitable solution method.

Srinivasan has developed a BEM method to solve the frequency domain DA for image-guided optical tomography. Based on prior anatomical data, the domain is divided into a number of regions in which the optical properties are assumed to be homogeneous. In doing so, the DA can be expressed as a modified Helmholtz equation that can be solved using the BEM (Brebbia and Dominguez 1992). The model uses a point source placed one scattering distance inside the domain and a robin type boundary condition.

Elisee *et al* have developed a combined FEM/BEM method for layered media. A volume mesh is created for a region of interest which is then solved using the FEM. In the remainder of the geometry, optical properties are assumed to be homogeneous and are solved using the BEM. This approach has been shown to agree well with analytical solutions (Elisee et al. 2010).

In order to overcome the limitations of the DA, Dorn, developed a model based on the time dependent RTE which was solved using the finite difference method and the discrete ordinates method (Dorn 1998). In numerical studies, using artificial data, the model was shown to be capable of reconstructing scattering and absorption inhomogeneities. However, this model did not allow for the presence of void-like regions.

Heilscher et al (Hielscher et al. 1998) also developed a discrete ordinates, finite difference approximation to the RTE called DANTSYS. This was used to highlight the limitations of the DA in large heterogeneous media. It was found that when the conditions for diffusion theory (mu_a<<mu_s) were violated, the DA tended to overestimate absorption effects. This was also seen in regions in void like regions with low absorption and scatter, such as in the CSF surrounding the brain. It was also noted that, in large domains, small discrepancies between diffusion theory and transport theory could accumulate resulting in significant differences at large distances from the source.

More recently, Klose *et al* developed a similar model that again used the upwind FDM and discrete ordinates (32 discrete ordinates per calculation) method to solve the RTE in two dimensions (Klose et al. 2002). Unlike the Dorn model, this was validated using experimental data. Measurements of fluence from homogeneous phantoms were shown to agree well with those predicted by the forward model.

A higher ordered FEM-Pn approximation to the RTE in two dimensions, was developed by Aydin et al (Aydin et al. 2002; Aydin et al. 2004). This model was used to investigate light propagation using a very simple model of the head. The effects of anisotropic scattering and void-like regions were also investigated. Large differences of up to 100% were seen between the DA and RTE, even at large source-detector separations. It was also noted that, while the choice of anisotropy factor was found to be important in absorbing homogeneous media, it was almost negligible in heterogeneous media containing void-like structures.

Wright et al (Wright et al. 2006) presented a forward model and corresponding inverse model based on the PN approximation for N = 1, 3, 5 & 7. The model was applied to problems involving low levels of scatter in which the diffusion approximation was invalid. It was shown that the higher order models resulted in more accurate reconstructions of target volumes as well as lower levels of artefact. This work, however, was limited to solving two dimensional problems. The expansion of the 2D PN equations to 3D problems is non-trivial; a much larger set of equations is required.

5.4 Discussion

In this chapter, the forward problem of optical tomographic imaging has been introduced, and some of the existing models have been discussed. Whilst a full description of light transport in tissue can be achieved using the Radiative Transport Equation, solutions are computationally expensive. As such, the Diffusion Approximation to the RTE is commonly used and is known to be accurate in large geometries in which scattering effects are dominant over absorption effects.

The forward model can be solved in a number of ways. Most typically, numerical models such as the FEM or FDM are used. While Monte Carlo methods provide an accurate model of photon transport, their long computation times generally limit them to validation studies.

Chapter 6 - The Inverse Problem and Image Reconstruction

6.1 Introduction

The aim of the inverse problem in imaging is to calculate the distribution of optical properties within a known geometry given a set of boundary measurements. In x-ray CT imaging, the boundary measurements are a set of projections measured over a range of angles. Although x-rays undergo some scattering in tissue, it can be assumed that they predominately pass through in straight lines. This greatly simplifies the inverse problem and allows simple methods such as filtered back projection to be used (Curry et al. 1990).

In frequency domain optical tomographic imaging systems, boundary data consists of a number of amplitude and phase measurements of trans-illuminated light. Unlike the case of x-ray imaging, photons at the wavelengths used in optical imaging undergo heavy scattering and take random paths through tissue. This means that simple back-projection methods cannot be used and, in general, model based iterative image reconstruction (MOBIIR) techniques are used (Hielscher 1997). These MOBIIR methods involve making an initial guess at the optical property distribution and using a forward model to estimate the boundary measurements. The aim of the inverse problem is then to minimize the difference between the measured data Φ^M and data calculated by the forward model Φ^C. This chapter will discuss the solution of the inverse problem and introduce an image reconstruction algorithm based on the simplified spherical harmonics approximation.

6.2 Perturbation Method

If we have an estimate of the optical properties, μ_0, that is close to the true optical properties throughout the domain, μ the non-linear forward problem, $I = F(\mu)$, can be linearised. As a result, the measured data, I, will be close to the predicted data I_0 (Arridge 1999). The forward model can then be expanded via a Taylor expansion

$$\begin{aligned} I(\mu(r),\omega) = I(\mu_0(r),\omega) + F'(\mu_0)[\mu(r) - \mu_0(r)] + \ldots \\ F''(\mu_0)[\mu(r) - \mu_0(r)]^2 \ldots \end{aligned} \tag{6.1}$$

where F' and F'' are the first and second derivatives of the forward model with respect to the optical properties (commonly referred to as the Jacbian and Hessian respectively)(Hielscher et al. 1999), $\mu_0(r)$ is the initial estimate of the optical properties and $\mu(r)$ is the measured optical properties.

The perturbation approach involves neglecting terms of the second order and higher. The problem then becomes

$$I(\mu(r),\omega) = I(\mu_0(r),\omega) + J[\mu(r) - \mu_0(r)] \tag{6.2}$$

which can be rearranged as

$$\Delta I = J\Delta\mu \tag{6.3}$$

where ΔI is the difference between modelled and measured data, J is the Jacobian matrix and $\Delta\mu$ is the optical property update vector. The optical property update vector can then be found by inverting the Jacobian matrix, i.e. $\Delta\mu = J^{-1}\Delta I$. The optical properties of the domain can then be found by iteratively solving the inverse problem to minimise ΔI with optical properties for each iteration taking the values $\mu_i = \mu_{i-1} + \Delta\mu$.

6.3 Calculating the Jacobian Matrix

6.3.1 Perturbation Method

The rows of the Jacobian matrix represent the sensitivity in boundary data for a source-detector pair after an infinitesimal change in the optical properties and is known as a photon density measurement function (Arridge 1995a). The Jacobian for an optical property μ has the structure

$$J = \begin{bmatrix} \frac{\delta \ln I_1}{\delta \mu_1} & \frac{\delta \ln I_1}{\delta \mu_2} & \cdots & \frac{\delta \ln I_1}{\delta \mu_{NN}} \\ \frac{\delta \theta_1}{\delta \mu_1} & \frac{\delta \theta_1}{\delta \mu_2} & \cdots & \frac{\delta \theta_1}{\delta \mu_{NN}} \\ \frac{\delta \ln I_2}{\delta \mu_1} & \frac{\delta \ln I_2}{\delta \mu_2} & \cdots & \frac{\delta \ln I_2}{\delta \mu_{NN}} \\ \frac{\delta \theta_2}{\delta \mu_1} & \frac{\delta \theta_2}{\delta \mu_2} & \cdots & \frac{\delta \theta_2}{\delta \mu_{NN}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta \ln I_{NM}}{\delta \mu_1} & \frac{\delta \ln I_{NM}}{\delta \mu_2} & \cdots & \frac{\delta \ln I_{NM}}{\delta \mu_{NN}} \\ \frac{\delta \theta_{NM}}{\delta \mu_1} & \frac{\delta \theta_{NM}}{\delta \mu_2} & \cdots & \frac{\delta \theta_{NM}}{\delta \mu_{NN}} \end{bmatrix} \tag{6.4}$$

where $\delta \ln I_j / \delta \mu_r$ are sub-matrices that define changes in the *j*th measurement of log amplitude due to a change in optical properties at point r, and $\delta \theta_j / \delta \mu_r$ define changes in the *j*th measurement of phase due to changes in optical properties at point r. The total Jacobian consists of two kernels

$$J = [J_{\mu_a} \quad J_{\kappa}] \tag{6.5}$$

where J_{μ_a} and J_{κ} are the Jacobians due to changes in the absorption coefficient and diffusion coefficient respectively, and have the form

$$J_{\kappa} = \begin{bmatrix} \frac{\delta \log I}{\delta \kappa} & ; & \frac{\delta \theta}{\delta \kappa} \end{bmatrix} \tag{6.6a}$$

$$J_{\mu_a} = \left[\frac{\delta \log I}{\delta \mu_a} \quad ; \quad \frac{\delta \theta}{\delta \mu_a} \right] \tag{6.6b}$$

The perturbation method of building the Jacobian involves perturbing the optical properties, either μ_a or μ_s' (and therefore κ) , by a small fraction (<0.1%) and measuring the changes in boundary measurements, in relation to a set of reference data, for each source-detector pair. Each of these measurements represents an element in the Jacobian matrix. This method, however, is computationally inefficient. For a FEM mesh with 1000 nodes and 16 sources, the forward model would need to be solved 32000 times (number of nodes × number of sources × number of optical properties).

6.3.2 Adjoint Method

Assuming a domain with a single source and a single detector located on the boundary, Figure 6.1, the light distribution that would result is known as the *Direct field.* The principle of reciprocity states that if the source were located at point n and measurements were taken at the detector at point j, an identical measurement would be made if the detector were placed at point n and the source at point j. The field measured when the source and detector locations are interchanged is known as the *Adjoint field.* This holds true, regardless of the initial source position.

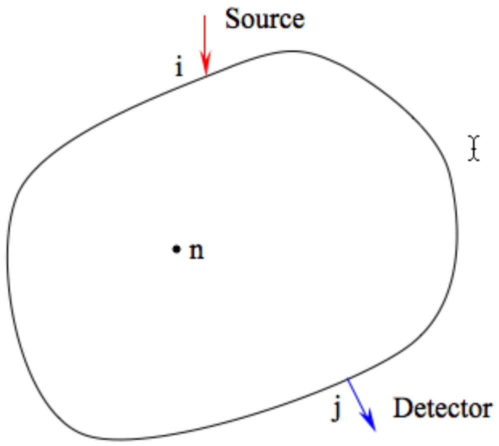


Figure 6.1 A domain with a single source and detector with an internal point of interest n

Now if the source was located at position i, and the detector at position j, the sensitivity to any changes to the optical properties would be affected by a combination of both the direct and adjoint fields. The probability of a photon travelling from the source to point n is given by P_{direct} and the probability of a photon travelling from an adjoint source at point j (the detector) to point n is given by P_{adj}. As the probability of a photon travelling from point n to the detector at point j is also given by P_{adj}, the probability of measuring a photon that has passed through point n, and therefore the sensitivity to changes at point n, is then given by $P_{direct} \times P_{adj}$. The reciprocity theorem can then be used to determine the Jacobian entries (based on the DA) to be

$$J_{\mu_a}(i,j) = \Phi_i \times \Phi_j^{Adj} \tag{6.7}$$

for absorption coefficient, and

$$J_{\kappa}(i,j) = \nabla\Phi_i \times \nabla\Phi_j^{Adj} \tag{6.8}$$

for the diffusion coefficient. The implication of the reciprocity approach is that, rather than solving the forward model once for every node in the mesh but only recording the change at the position of the detector, the forward model needs to be solved just twice, once for the direct field and once with an adjoint source located at the detector (Arridge and Schweiger 1995). Using the same example as for the perturbation method, a 1000 node mesh with 16 sources and detectors would require just 32 solutions of the forward model, 16 for the direct field and 16 for the adjoint field.

6.3.3 Normalisation & Regularisation

The magnitude of the entries in the Jacobian matrix are dependent on the data type and optical property to which they refer. The entries with greater sensitivities will tend to over power those with lower sensitivities, which negatively affects the reconstructed image (Eames and Dehghani 2008). To overcome this problem, the Jacobian can be normalised. One approach is to normalise the by either the rows or columns of the Jacobian to their minimum, maximum or mean values. This ensures that each row or column has an equal contribution to the update vector. This works well when maximum sensitivity is required from all regions of the model, but it can lead to artefact. A more effective approach is to normalise by optical properties. This method ensures that both μ_a and μ_s' have equal contributions and can be well separated.

Due to the highly scattering nature of light in tissue, the maximum sensitivity occurs in the regions nearest the sources and detectors, and becomes very small with increasing distance (Yalavarthy et al. 2006). As the sensitivity in these regions is so small, they have very little effect on the update vector, but their presence in the Jacobian increases the computational time required to solve the inverse problem. However, it has been shown that these regions can be removed with little effect on the results, but greatly improving computation times (Eames et al. 2007).

6.4 Tikhonov Minimization

The goal of the inverse problem is the recovery of the optical properties, μ, by minimizing the difference between the measured data Φ^M and data calculated by the forward model Φ^C using a modified Tikhonov minimization approach given by (Dehghani et al. 2008)

$$\chi^2 = \overset{\min}{\underset{\mu}{}} \left\{ \sum_{i=1}^{NM} \left(\Phi^M - \Phi^C\right)^2 + \lambda \sum_{j=1}^{NN} \left(\mu_j - \mu_0\right)^2 \right\} \tag{6.9}$$

where NM is the number of measurements, NN is the number of nodes in the FEM, μ_0 is an initial estimate of the optical properties and λ is the Tikhonov regularization parameter. By considering only the first order terms and applying the Levenberg-Marquardt procedure, this leads to an update vector

$$\delta\mu = \left(J^T J + \bar{\lambda} I\right)^{-1} J^T \delta\Phi \tag{6.10}$$

where $\bar{\lambda} = 2\lambda$ and is systematically reduced after each iteration.

Image reconstruction is an ill-posed problem, which means that either no true solution exists, there is no unique solution or the solution is not continuously dependent on the data. The Jacobian matrix tends to be under-determined as there are fewer measurements than unknowns to be solved. The Jacobian also tends to be ill-conditioned. Combined with the fact that the data is contaminated by noise, a small change in optical properties can lead to large changes in boundary data, meaning the update vector may not be accurate. This effect can be reduced by regularising the Jacobian before inversion with the Tikhonov regularisation parameter. The best value for the regularisation parameter can be found by plotting the difference between measured and calculated data against the difference in optical properties to get an 'L-curve'. The best value for the regularization parameter is the one that produces the vertex of the L-curve (Calvetti et al. 2000).

6.5 Inclusion of Prior Anatomical Information

One of the significant advances in optical tomography has been the inclusion of prior anatomical information, generally in the form of a structural MRI image, into the image reconstruction process. A multimodality imaging system can provide high-resolution, functional images. Schweiger and Arridge (Schweiger and Arridge 1999) incorporated prior information using a two step reconstruction process. The tissue properties were first reconstructed for a number of non-overlapping regions that were defined using the prior anatomical data. A full reconstruction onto the FEM basis was then performed, based on the results of the region-based reconstruction. Ntziachristos et al (Ntziachristos et al. 2002b) used MRI information to segment the imaging domain into a small number of regions pertaining to different tissue types. Each node within a region was then assumed to have the same optical properties. This greatly reduced the number of unknowns to be calculated whilst the number of measurements remains the same. This results in an over-determined system which can be solved to a higher degree of accuracy.

In general, the use of spatial priors involves one of two methods, soft priors or hard priors. These two methods will be discussed in the following sections

6.5.1 Soft Priors

The soft priors method, spatial prior information is incorporated into the Tikhonov minimization as a penalty term resulting in a Laplacian type regularization matrix, equation 6.9.

$$\delta\mu = \left(J^T J + \alpha L^T L\right)^{-1} J^T \delta\Phi \tag{6.11}$$

where the constant α balances the effect of the prior with the model-data mismatch (Brooksby et al. 2005). The prior MRI data is used to define a number of regions within the FEM mesh relating to different tissue types. The L matrix relates each node of the model to all other nodes. Therefore, for a given node i, it's relationship to another node j can be defined as (Brooksby et al. 2005)

$$L_{i,j} = \begin{cases} 0 & \text{if i and j are in different regions} \\ -1/N & \text{if i and j are in the same region} \\ 1 & \text{if i = j} \end{cases} \tag{6.12}$$

where N is the number of nodes within the FEM mesh. The $L^T L$ regularization term acts as a smoothing operator within each region by averaging the updates within the region.

6.5.3 Hard Priors

The hard priors approach involves reducing the number of unknown parameters to a number of regions segmented from the prior anatomical information. Within each of the regions, it is assumed that the optical properties are homogeneous. The number of unknown parameters can then be reduced by multiplying the Jacobian matrix by an *a-priori* matrix such that

$$J^* = JK \tag{6.13}$$

where K is a NN×NR matrix with values

$$K = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,NR} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,NR} \\ \vdots & \vdots & \ddots & \vdots \\ k_{NN,1} & k_{NN,2} & \cdots & k_{NN,NR} \end{bmatrix} \tag{6.14}$$

where

$$k_{i,j} = \begin{cases} 1, & i \in R_n \\ 0, & i \notin R_n \end{cases} \tag{6.15}$$

and R_n represents region *n* (Dehghani et al. 2003b). Applying the *a-priori* matrix reduces the Jacobian from dimensions NM×NN to NM×NR, where NR is the number of regions segmented from the structural information. In the case of a 1000 node mesh, the number of parameters to be determined can be reduced from 2000 (μ_a and μ_s at each node) with 32 boundary measurements (phase & amplitude at 16 detectors) to just 4 (μ_a and μ_s in two regions) with the same number of measurements. This reduction in parameters, greatly improves the computational efficiency of the Jacobian inversion and results in an over-determined problem. Even though the number of parameters is reduced, however, the problem may still be ill-posed (Arridge 1999), and so the optical properties are calculated using the Levenburg-Marquardt approach (Yalavarthy et al. 2007).

6.6 Conjugate Gradients Method

One of the drawbacks of the Levenberg-Marquardt algorithm is that it requires the Jacobian to be repeatedly created and inverted. As the size of the problem becomes large, the creation and inversion of the Jacobian can be computationally expensive . An alternative approach is to use gradient methods, such as the conjugate gradients, in which the gradient of the objective function is calculated using the adjoint method, where the objective function is defined as

$$\Phi(\varsigma) = \sum_{j=1}^{S}\sum_{i=1}^{M_j}\frac{\left(y_{j,i} - y_{j,i}^{c}\right)^2}{2\sigma_{j,i}^2} \tag{6.16}$$

where the first sum is over all sources and the second is over all measurement positions, $y_{j,i}$ is a measurement due to source j at detector i, $y_{j,i}^{c}$ is the calculated data and $\sigma_{j,i}$ is the standard deviation. The gradient of the objective function with respect to the optical properties is given by

$$\frac{d\Phi}{d\mu} = \sum_{j}^{S}\sum_{i}^{M_j}\frac{d\Phi}{dy_{j,i}^{c}}\frac{\partial y_{j,i}^{c}}{\partial\mu} \tag{6.17}$$

Once the gradient has been calculated, an iterative line minimization is performed in the direction of the gradient. This step involves repeated calculations of the forward model in which the optical properties are varied. Once the minimum along the line has been found, the gradient is re-calculated and another line minimization is performed along a new direction. This is repeated until the objective function is minimised (Hielscher et al. 1999).

6.7 Summary

The image reconstruction problem in optical tomography is ill-posed and under-determined. In general, images are reconstructed by iteratively solving the forward model whilst using methods such as Tikhonov minimization to reduce the difference between measured and model data.

The sensitivity of boundary measurements to changes in optical properties throughout the imaging domain can be mapped using the Jacobian matrix. The sensitivity profile between a source detector pair tends to take on a banana shape, with strong sensitivity near the source and detector with a lower, but broader sensitivity at depth.

The under-determined nature of the inverse problem can be reduced with the use of prior information. Prior anatomical information can be used to determine regions of similar tissue type. By assuming the optical properties within these regions will be homogeneous, the number of unknowns to be determined is significantly reduced.

The development of an image reconstruction algorithm based on the SP_N approximation will be introduced in chapter 8.

Chapter 7 – The Simplified Spherical Harmonics Approximation

7.1 Introduction

The P_N and S_N approximations, discussed in section 5.3, provide an approximation to the RTE by discretising the angular dependence of the angular flux. One of the drawbacks common to both the P_N and S_N approximations is that it they can be computationally expensive to solve; the set of coupled equations to be solved increases with $(N+1)^2$ for the P_N approximation and $N(N+2)$ for the S_N approximation. As such, it is clear that a photon transport model that could achieve a higher level of accuracy than the DA whilst being computationally efficient would be desirable.

Gelbard (1968), who was working on neutron transport problems, developed a heuristic simplification of the P_N equations which he called the simplified P_N (SP_N) approximation. His derivation involved the simple assumption that the differential operators occuring in the 1D P_N equations with their 3D counterparts. In the case of the P_1 (diffusion) approximation this expansion is exact which means that the P_1 and SP_1 approximations are equivalent. At higher orders, however, this equivalence does not apply. Whereas the P_N and S_N approximations converge on the true RTE solution as $N \to \infty$, the SP_N approximations asymptotically approach the RTE solution.

The theoretical foundation for Gelbard's derivation, however, was weak and, as such, not widely adopted. In the 1990s however, Larsen et al (Larsen 1993; Larsen et al. 1996) developed a much more theoretically sound derivation based on an asymptotic analysis, although this method did not include the appropriate boundary conditions. It was found that the SP_N

approximation could provide greater accuracy than the DA whilst requiring less computation time than the S_N. Although the SP_N approximations do not converge on the RTE, it was found that they also overcame the ray effect artifacts that are common to the S_N.

The 1990s also produced a derivation of the SP_N equations and their boundary conditions based on a variational analysis and it was shown that the SP_N approximation produced better results than the P_1 or diffusion approximation (Tomasevic and Larsen 1996; Brantley and Larsen 2000). All of the derivations, however, resulted in a similar set of equations as found by Gelbard in 1960s.

The SP_N approximation has previously been applied to a range of particle and heat transfer problems (Kotiluoto 2001) (Ciolini 2002; Larsen 2002). The problems found in the nuclear industry, however, have some distinct differences to those found in tissue optics. Neutron transport models, for example, generally consider systems that are driven by internal sources. In tissue optics, however, the system is generally driven by an external source on the surface of the tissue.

Neutron transport models generally consider systems driven by internal sources whereas tissue optics involve external sources. Light transport models, in general, involve a partially reflective boundary condition as some photons are reflected back into the tissue. This reflection does not occur in neutron transport problems and so different boundary conditions are required.

Josef (1996) developed an SP_N model for electron transport problems which generally involve more forward-peaked scattering which suggests that the SP_N approximations are appropriate for the anisotropic scattering found in tissue optics. The SP_N approximation was first used to model light propagation in biological tissue by Klose *et al (2006).* In this work, the mathematical framework for the approximation was developed for continuous wave problems and models were demonstrated for N=1,3,5 & 7. The resulting equations were implemented using a FDM and were compared against a validated S_N model. Whilst this publication presented only 2D data, it was noted that one of the advantageous features of the SP_N equations is that the same set of equations apply to both 2D & 3D problems. The SP_3 model was later applied to the problem image reconstruction in bioluminescence problems (Klose and Beattie 2008).

More recently, the SP_N approximation has been applied to time-domain problems {Dominguez, 2010 #423}. An FEM solution was developed which resulted in fluence distributions that closely reproduce Monte Carlo simulations. It was also shown that the SP_N model was better able to determine changes in absorption of small insertions in comparison to the diffusion approximation. Liemert *et al* (2010) have developed an analytical solution to the SP_N equations and, again, found that they result in better agreement with Monte Carlo data than the diffusion equation.

In this thesis, the SP_N equations for *frequency domain* tissue optics problems have been derived for the first time, section 7.2. They are also implemented using the FEM which allows a much greater degree of flexibility in modeling complex geometries compared to the FDM which has been used in previous studies. Although similar studies have been presented in the past, such as Wright et al (2006), the use of the SP_N approximation results in significant advantages. Firstly, the computational cost of implementing the SP_N approximation is significantly lower than for the PN approximation. The

SP_N approximation results in a set of diffusion-like equations which can be easily solved using existing diffusion solvers whereas the P_N equations are far more complex. The number of equations to be solved is also significantly less for the SP_N approximation, particularly at high N, further improving the computational efficiency.

In this chapter, the SP_N equations will be derived using Gelbard's original method. The implementation of the SP_N approximation using the FEM will be discussed and the validation against Monte Carlo data and an existing diffusion based model is presented.

7.2 Deriving the SP$_N$ Equations

The derivation of the SP_N equations for applications in optical imaging was first shown by Klose and Larsen for continuous wave problems(Klose and Larsen 2006). In this work, the SP_N equations have been implemented for frequency domain problems (Chu et al. 2008). More recently, the time domain SP_N equations have been shown (DomÌnguez and BÈrubÈ-LauziËre 2010).

Gelbard's derivation of the SP_N equations first requires the RTE to be expressed for a planar geometry

$$\omega\frac{\partial\psi}{\partial x}(x,\omega)+\mu_t(x)\psi(x,\omega)=\int_{-1}^{1}\mu_s(x,\omega,\omega')\psi(x,\omega')d\omega'+\frac{Q(x)}{2} \tag{7.1}$$

where

$$\mu_s(x,\omega,\omega')=\sum_{n=0}^{\infty}\frac{2n+1}{2}\mu_s(x)g^nP_n(\omega)P_n(\omega') \tag{7.2}$$

To obtain the 1D P$_N$ approximation, for $n \geq 0$, the Legendre moments of the radiance are defined as

$$\phi_n(x) = \int_{-1}^{1} P_n(\omega)\psi(x,\omega)d\omega \tag{7.3}$$

and the nth order absorption coefficients are defined as

$$\mu_{an}(x) = \mu_a(x) + \mu_s(x)(1 - g^n) \tag{7.4}$$

The operator $\int_{-1}^{1} P_n(\omega)(\cdot)d\omega$ is then applied to equation 7.1 for $n \geq 0$, resulting in

$$\frac{n+1}{2n+1}\frac{d\phi_{n+1}}{dx}(x) + \frac{n}{2n+1}\frac{d\phi_{n-1}}{dx}(x) + \mu_{an}(x)\phi_n(x) = \delta_{n0}Q(x) \tag{7.5}$$

with $\phi_{-1} = 0$. To close this system with the P$_N$ approximation, we select an odd positive integer N and take

$$\psi(x,\omega) \approx \sum_{n=0}^{N} \frac{2n+1}{2}\phi_n(x)P_n(\omega) \tag{7.6a}$$

which is equivalent to setting

$$\phi_n(x) = 0 \;,\; N < n < \infty. \tag{7.6b}$$

Equation 7.6(b) is then substituted back into equation 7.5 with the requirement that the result holds for $0 \leq n \leq N$. The N+1 unknown functions $\phi_n(x)$ results in a closed system of N+1 first order differential equations. Algebraically eliminating the odd-ordered moments of equation 7.5 gives

$$\phi_n(x) = -\frac{1}{\mu_{an}}\frac{d}{dx}\left[\frac{n+1}{2n+1}\phi_{n+1}(x) + \frac{n}{2n+1}\phi_{n-1}(x)\right] \tag{7.7}$$

for $1 \leq n \leq N$ and n odd. This result can then be used to eliminate the odd-ordered moments in the rest of equation 7.5 results in the following system of equations, constituting the P$_N$ approximation for planar geometries

$$
\begin{aligned}
&-\left(\frac{n+1}{2n+1}\right)\frac{d}{dx}\frac{1}{\mu_{a,n+1}}\frac{d}{dx}\left[\left(\frac{n+2}{2n+3}\right)\phi_{n+2}+\left(\frac{n+1}{2n+3}\right)\phi_n\right] \\
&-\left(\frac{n}{2n+1}\right)\frac{d}{dx}\frac{1}{\mu_{a,n-1}}\frac{d}{dx}\left[\left(\frac{n}{2n-1}\right)\phi_n+\left(\frac{n-1}{2n-1}\right)\phi_{n-2}\right]+\mu_{an}\phi_n=Q
\end{aligned}
\tag{7.8}
$$

The SP$_N$ equations can then be found by replacing the 1D diffusion terms

$\frac{d}{dx}\frac{1}{\mu_{an}(x)}\frac{d}{dx}\phi(x)$ with their 3D counterparts

$$
\nabla\cdot\frac{1}{\mu_{an}(x)}\nabla\phi(x).
\tag{7.9}
$$

Equation 7.8 becomes

$$
\begin{aligned}
&-\left(\frac{n+1}{2n+1}\right)\nabla\cdot\frac{1}{\mu_{a,n+1}}\nabla\left[\left(\frac{n+2}{2n+3}\right)\phi_{n+2}+\left(\frac{n+1}{2n+3}\right)\phi_n\right] \\
&-\left(\frac{n}{2n+1}\right)\nabla\cdot\frac{1}{\mu_{a,n-1}}\nabla\left[\left(\frac{n}{2n-1}\right)\phi_n+\left(\frac{n-1}{2n-1}\right)\phi_{n-2}\right]+\mu_{an}\phi_n=Q
\end{aligned}
\tag{7.10}
$$

For convenience, these equations can be re-written in terms of the composite moments of fluence ϕ:

$$\varphi_1=\phi_0+2\phi_2 \tag{7.11a}$$

$$\varphi_2=3\phi_2+4\phi_4 \tag{7.11b}$$

$$\varphi_3=5\phi_4+6\phi_6 \tag{7.11c}$$

$$\varphi_4=7\phi_6 \tag{7.11d}$$

The SP$_N$ equations for N=7 can then be found by expanding equation 7.10 into the following set of coupled equations.

$$
\begin{aligned}
&-\nabla.\frac{1}{3\mu_{a1}}\nabla\varphi_1+\left(\mu_a+\frac{i\omega}{c}\right)\varphi_1= \\
&Q+\frac{2}{3}\left(\mu_a+\frac{i\omega}{c}\right)\varphi_2-\frac{8}{15}\left(\mu_a+\frac{i\omega}{c}\right)\varphi_3+\frac{16}{35}\left(\mu_a+\frac{i\omega}{c}\right)\varphi_4
\end{aligned}
\tag{7.12a}
$$

$$\begin{aligned}&-\nabla.\frac{1}{7\mu_{a3}}\nabla\varphi_2+\left(\frac{4}{9}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{5}{9}\mu_{a2}\right)\varphi_2=\\&-\frac{2}{3}Q+\frac{2}{3}\left(\mu_a+\frac{i\omega}{c}\right)\varphi_1+\left(\frac{16}{45}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{4}{9}\mu_{a2}\right)\varphi_3\\&-\left(\frac{32}{105}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{8}{21}\mu_{a2}\right)\varphi_4\end{aligned}\tag{7.12b}$$

$$\begin{aligned}&-\nabla.\frac{1}{11\mu_{a5}}\nabla\varphi_3+\left(\frac{64}{225}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{16}{45}\mu_{a2}+\frac{9}{25}\mu_{a4}\right)\varphi_3=\\&\frac{8}{15}Q-\frac{8}{15}\left(\mu_a+\frac{i\omega}{c}\right)\varphi_1+\left(\frac{16}{45}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{4}{9}\mu_{a2}\right)\varphi_2\\&+\left(\frac{128}{525}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{32}{105}\mu_{a2}+\frac{54}{175}\mu_{a4}\right)\varphi_4\end{aligned}\tag{7.12c}$$

$$\begin{aligned}&-\nabla.\frac{1}{15\mu_{a7}}\nabla\varphi_4+\left(\frac{256}{1225}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{64}{245}\mu_{a2}+\frac{324}{1225}\mu_{a4}+\frac{13}{49}\mu_{a6}\right)\varphi_4=\\&-\frac{16}{35}Q+\frac{16}{35}\left(\mu_a+\frac{i\omega}{c}\right)\varphi_1-\left(\frac{32}{105}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{8}{21}\mu_{a2}\right)\varphi_2\\&+\left(\frac{128}{525}\left(\mu_a+\frac{i\omega}{c}\right)+\frac{32}{105}\mu_{a2}+\frac{54}{175}\mu_{a4}\right)\varphi_3\end{aligned}\tag{7.12d}$$

where, Q is the source term, ω is the modulation frequency, μ_s is the scattering coefficient and μ_{an} is the nth order absorption coefficient given by

$$\mu_{an}(x)=\mu_t(x)-\mu_s(x)g^n\tag{7.13}$$

The total fluence *Φ* is then given by

$$\phi=\varphi_1-\frac{2}{3}\varphi_2+\frac{8}{15}\varphi_3-\frac{16}{35}\varphi_4\tag{7.14}$$

The SP_1 approximation can then be found by setting $\phi_6=\phi_4=\phi_2=0$ and solving the first equation. The SP_3 approximation can be found by solving the first two equations with $\phi_6=\phi_4=0$ and the SP_5 approximation can be found by setting $\phi_6=0$ and solving the first three equations. It can then be seen that the SP_1 case simplifies to the diffusion equation as

$$-\nabla.\frac{1}{3\mu_{a1}}\nabla\varphi_1+\left(\mu_a+\frac{i\omega}{c}\right)\varphi_1=Q \tag{7.15}$$

where $\mu_{a1}=\mu_a+\mu_s(1-g)$ and $\varphi_1=\phi$ which leads to the diffusion equation

$$-\nabla.\frac{1}{3\left(\mu_a+\mu_s'\right)}\nabla\varphi_1+\left(\mu_a+\frac{i\omega}{c}\right)\varphi_1=Q \tag{7.16}$$

The resulting equations are not the same as the multi-dimensional P_N approximation. The full P_N approximation for 3D problems involves expanding the angular terms into higher spherical harmonic functions which involve two scalar angles as apposed to one. The resulting equations have a much higher complexity with many more unknowns for a given N (Klose and Larsen 2006).

7.2.1 Boundary Conditions

Depending on the angle of incidence, photons attempting to cross the boundary $\partial\Omega$, at $r\in\partial\Omega$, will either pass through at a refracted angle defined by Snell's law or will be reflected back into the domain. The probability of a photon being reflected is given by

$$R(\cos\theta')=\begin{cases}\frac{1}{2}\left(\frac{n_m\cos\theta''-n_0\cos\theta'}{n_m\cos\theta''+n_0\cos\theta'}\right)^2+\frac{1}{2}\left(\frac{n_m\cos\theta'-n_0\cos\theta''}{n_m\cos\theta'+n_0\cos\theta''}\right)^2 & \theta'<\theta_c\\ & \theta'\geq\theta_c\end{cases} \tag{7.17}$$

where θ' is the internal angle, θ'' is the refracted angle and θ_c is the critical angle.

In the derivation of the boundary conditions, several angular moments of the reflectivity are encountered which are defined as

$$R_n=\int_0^1 R(\cos\theta')\omega^n d\omega \tag{7.18}$$

The boundary conditions for SP$_7$ are then found to be

$$\left(\frac{1}{2}+A_1\right)\varphi_1+\left(\frac{1+B_1}{3\mu_{a1}}\right)\mathbf{n}\cdot\varphi_1=\left(\frac{1}{8}+C_1\right)\varphi_2+\left(\frac{D_1}{\mu_{a3}}\right)\mathbf{n}\cdot\varphi_2+\left(-\frac{1}{16}+E_1\right)\varphi_3+\left(\frac{F_1}{\mu_{a5}}\right)\mathbf{n}\cdot\varphi_3$$
$$+\left(\frac{5}{128}+G_1\right)\varphi_4+\left(\frac{H_1}{\mu_{a7}}\right)\mathbf{n}\cdot\varphi_4+\int_{\Omega\cdot n<0}S(\Omega)2|\Omega\cdot\mathbf{n}|d\Omega \tag{7.19a}$$

$$\left(\frac{7}{24}+A_2\right)\varphi_2+\left(\frac{1+B_2}{7\mu_{a3}}\right)\mathbf{n}\cdot\varphi_2=\left(\frac{1}{8}+C_2\right)\varphi_1+\left(\frac{D_2}{\mu_{a1}}\right)\mathbf{n}\cdot\varphi_1+\left(\frac{41}{384}+E_2\right)\varphi_3+\left(\frac{F_2}{\mu_{a5}}\right)\mathbf{n}\cdot\varphi_3$$
$$+\left(-\frac{1}{16}+G_2\right)\varphi_4+\left(\frac{H_2}{\mu_{a7}}\right)\mathbf{n}\cdot\varphi_4+\int_{\Omega\cdot n<0}S(\Omega)\left(5|\Omega\cdot\mathbf{n}|^3-3|\Omega\cdot\mathbf{n}|\right)d\Omega \tag{7.19b}$$

$$\left(\frac{407}{1920}+A_3\right)\varphi_3+\left(\frac{1+B_3}{11\mu_{a4}}\right)\mathbf{n}\cdot\varphi_3=\left(-\frac{1}{16}+C_3\right)\varphi_1+\left(\frac{D_3}{\mu_{a1}}\right)\mathbf{n}\cdot\varphi_1+\left(\frac{41}{384}+E_3\right)\varphi_2+\left(\frac{F_3}{\mu_{a3}}\right)\mathbf{n}\cdot\varphi_2$$
$$+\left(\frac{233}{2560}+G_3\right)\varphi_4+\left(\frac{H_3}{\mu_{a7}}\right)\mathbf{n}\cdot\varphi_4$$
$$+\int_{\Omega\cdot n<0}S(\Omega)\left(\frac{63}{4}|\Omega\cdot\mathbf{n}|^5-\frac{35}{2}|\Omega\cdot\mathbf{n}|^3+\frac{15}{4}|\Omega\cdot\mathbf{n}|\right)d\Omega \tag{7.19c}$$

$$\left(\frac{3023}{17920}+A_4\right)\varphi_4+\left(\frac{1+B_4}{15\mu_{a7}}\right)\mathbf{n}\cdot\varphi_4=\left(\frac{5}{128}+C_4\right)\varphi_1+\left(\frac{D_4}{\mu_{a1}}\right)\mathbf{n}\cdot\varphi_1+\left(-\frac{1}{16}+E_4\right)\varphi_2$$
$$+\left(\frac{F_4}{\mu_{a3}}\right)\mathbf{n}\cdot\varphi_2+\left(\frac{233}{2560}+G_4\right)\varphi_3+\left(\frac{H_4}{\mu_{a5}}\right)\mathbf{n}\cdot\varphi_3$$
$$+\int_{\Omega\cdot n<0}S(\Omega)\left(\frac{429}{8}|\Omega\cdot\mathbf{n}|^7-\frac{693}{8}|\Omega\cdot\mathbf{n}|^5+\frac{315}{8}|\Omega\cdot\mathbf{n}|^3-\frac{35}{8|\Omega\cdot\mathbf{n}|}\right)d\Omega \tag{7.19d}$$

The A$_n$……H$_n$ coefficients used in these equations are related to various orders of reflectivity and are defined in Appendix A.

An attractive feature of the SP$_N$ method is that the result consists of a set of diffusion-like equations. These equations can be easily solved using any existing DA solvers with very little modification. An important characteristic of the SP$_N$ method, however, is that the solution is asymptotic to the RTE solution whereas the P_N and S_N solutions converge as $N \rightarrow \infty$ (Klose and Larsen 2006). The result of this behaviour is that the most accurate SP_N solution is not always found using the highest ordered approximation. It can be shown, however, that the SP_N method with N>1 always leads to an improvement over the standard diffusion approximation (Chu et al. 2008).

7.3 Implementation of the SP_N Approximation

The composite moments of fluence, φ_n, as defined in equation 7.11, can be calculated with a set of coupled diffusion equations, equation 7.12(a)-(d). These can then be solved using a number of well-established numerical methods as discussed in Chapter 5.

In cases involving homogeneous optical properties, or a small number of homogeneous regions, the boundary element method (BEM) can be utilized. The BEM is an attractive option as only the surface of the imaging volume needs to be discretised. The internal fluence is then calculated based on measurements of light passing through the surface mesh. In cases of simple geometries, the BEM is computationally efficient. BEM calculations, however, tend to involve fully populated matrices, and as such, the memory requirements grow with the square of the number of elements in the problem. The BEM has been applied to region-based problems in which the imaging volume is divided into piecewise constant regions determined using prior information (Zacharopoulos et al. 2006; Srinivasan et al. 2007).

Inhomogeneous problems can be solved using the finite difference (FDM) (Hielscher et al. 1998; Klose et al. 2002) or finite elements (FEM) (Arridge et al. 1993; Jiang and Paulsen 1995; Schweiger et al. 1995; Gao et al. 1998; Jiang 1998; Dehghani et al. 2003a) methods. The FDM involves replacing the differential terms in a PDE with equivalent difference quotients. Whilst it is relatively easy to implement, the FDM is limited in the complexity of geometries that it can accurately model. In its basic form, the FDM is limited to problems consisting of rectangular or cubic geometries which is of limited use in clinical applications. By comparison, the FEM makes use of triangular or tetrahedral elements, for 2D and 3D problems respectively, thus providing a much more flexible solver as complex geometries can be easily modelled.

To date, the SP_N approximation has only been implemented using the Finite Difference Method for continuous wave systems (Klose and Larsen 2006). In this thesis, the frequency domain SP_N approximation has been implemented into NIRFAST (Dehghani et al. 2008) package, an FEM based diffusion solver.

7.3.1 FEM Implementation

As discussed previously, the finite element discretisation of a domain Ω can be obtained by subdividing the domain into D elements joined at V vertex nodes. The fluence, $\Phi(\mathbf{r})$ is then approximated by the piecewise continuous polynomial function

$$\Phi^h(r,w) = \sum_i^V \Phi_i u_i(r)\Omega^h \tag{7.20}$$

where Ω^h is a finite dimensional subspace spanned by basis functions $u_i(r)$. The basis functions describe how the function $\Phi(\mathbf{r})$ varies over each element and can take several forms e.g. piecewise constant, piecewise linear, piecewise quadratic etc. In this case, a piecewise linear basis function was used, in which case, Φ_i takes the form of a continuous linear function.

The FEM then reduces the problem of solving for Φ^h to one of sparse matrix inversion. In NIRFAST, this is done using a bi-conjugate gradients iterative solver. Equation 7.12(a)-(d) can then be represented in matrix form.

$$(K_1 + C + F_1)\varphi_1 = Q + (C_1 + G_1)\varphi_2 - (C_2 + H_1)\varphi_3 + (C_3 + I_1)\varphi_4 \tag{7.21a}$$

$$\begin{aligned}(K_3 + C_4 + C_5 + F_2)\varphi_2 = -\frac{2}{3}Q + (C_1 + G_2)\varphi_1 + (C_6 + C_7 + H_2)\varphi_3 \\ - (C_8 + C_9 + I_2)\varphi_4\end{aligned} \tag{7.21b}$$

$$\begin{aligned}(K_5 + C_{10} + C_{11} + C_{12} + F_3)\varphi_3 = \frac{8}{15}Q - (C_2 + G_3)\varphi_1 + (C_6 + C_7 + H_3)\varphi_2 \\ + (C_{13} + C_{14} + C_{15} + I_3)\varphi_4\end{aligned} \tag{7.21c}$$

$$\begin{aligned}(K_7 + C_{16} + C_{17} + C_{18} + C_{19} + F_4)\varphi_4 = -\frac{16}{35}Q + (C_3 + G_4)\varphi_1 - (C_8 + C_9 + H_4)\varphi_2 \\ + (C_{13} + C_{14} + C_{15} + I_4)\varphi_3\end{aligned} \tag{7.21d}$$

where the K_n and C_n matrices are given by

$$K_{n,ij} = \int_\Omega \kappa_n(r)\nabla u_i(r)\cdot\nabla u_j(r)d^n r \tag{7.22a}$$

$$C_{n,ij} = \int_\Omega \left(\mu_{an}(r) + \frac{i\omega}{c}\right)u_i(r)u_j(r)d^n r \tag{7.22b}$$

The boundary terms, given by matrices F_n, G_n, H_n and I_n have entries of the form

$$F_{ij} = \oint_\Omega u_i(r)u_j(r)d^{n-1}r \tag{7.23}$$

And the source vector Q has terms

$$q_i = \int_\Omega u_i(r)q(r)d^n r \tag{7.24}$$

Where the i, j indices refer to each node of the FEM mesh with $i, j = 1....n_m$ where n_m is the total number of nodes. The K, C, F, G, H and I matrices are all of $n_m \times n_m$ in size.

Equations 7.21(a)-(d) can then be rearranged into a single matrix equation of the form:

$$\begin{bmatrix} M_{11} & \cdots & \cdots & M_{14} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ M_{41} & \cdots & \cdots & M_{44} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \end{bmatrix} = \begin{bmatrix} Q_1 \\ -\frac{2}{3}Q_2 \\ \frac{8}{15}Q_3 \\ -\frac{16}{35}Q_4 \end{bmatrix} \tag{7.25}$$

where the M_{ij} matrices, as defined below, are also of size $n_m \times n_m$, and represent combinations of the system matrices. This allows the solution for composite moments $\varphi_1, \varphi_2, \varphi_3$ and φ_4 to be found simultaneously with just one matrix inversion. The large M matrix is often referred to as the Mass matrix.

$$M_{11} = K_1 + C + F_1, \tag{7.26a}$$

$$M_{12} = -(C_1 + G_1), \tag{7.26b}$$

$$M_{13} = C_2 + H_1, \tag{7.26c}$$

$$M_{14} = -(C_3 + I_1), \tag{7.26d}$$

$$M_{21} = K_3 + C_4 + C_5 + F_2, \tag{7.26e}$$

$$M_{22} = -(C_1 + G_2), \tag{7.26f}$$

$$M_{23} = -(C_6 + C_7 + H_2), \tag{7.26g}$$

$$M_{24} = C_8 + C_9 + I_2, \tag{7.26h}$$

$$M_{31} = K_5 + C_{10} + C_{11} + C_{12} + F_3, \tag{7.26i}$$

$$M_{32} = C_2 + G_3, \tag{7.26j}$$

$$M_{33} = -(C_6 + C_7 + H_3), \tag{7.26k}$$

$$M_{34} = -(C_{13} + C_{14} + C_{15} + I_3), \tag{7.26l}$$

$$M_{41} = K_7 + C_{16} + C_{17} + C_{18} + C_{19} + F_4 \tag{7.26m}$$

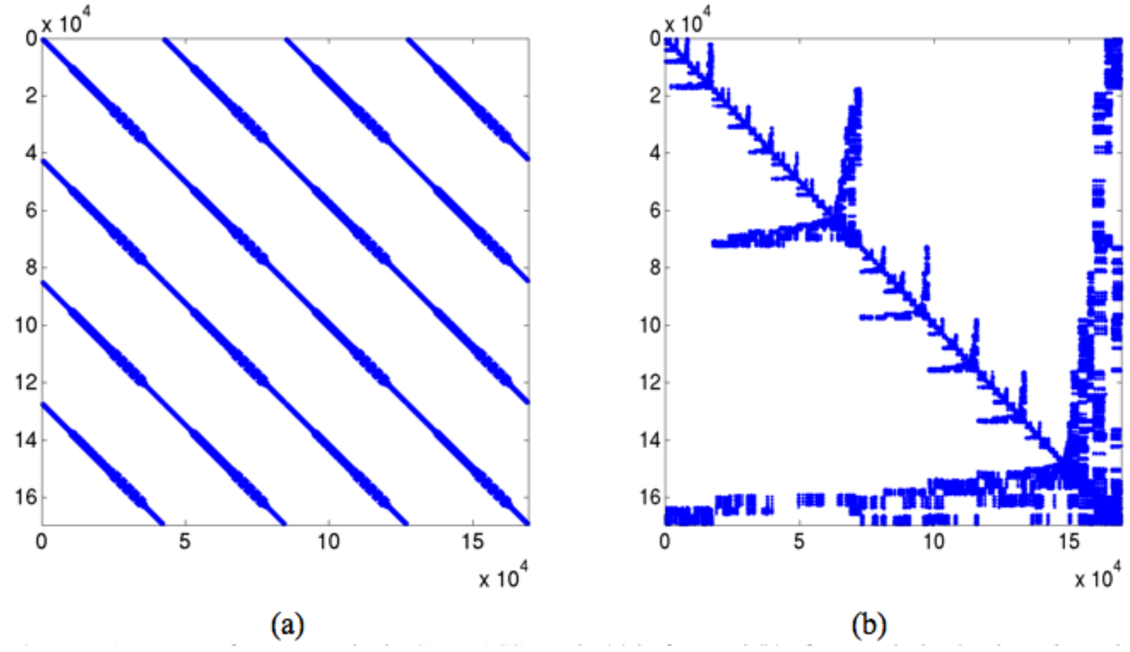
$$M_{42} = -(C_3 + G_4), \tag{7.26n}$$

$$M_{43} = C_8 + C_9 + H_4 \tag{7.26o}$$

$$M_{44} = -(C_{13} + C_{14} + C_{15} + I_4) \tag{7.26p}$$

7.3.2 Optimization through matrix re-ordering

The computational cost associated with solving sparse matrix problems depends not only on their size and sparsity but also on the distribution of the non-zero elements within the matrix. To minimize the fill-in phenomenon due to the decomposition of the Mass matrix in equation 7.25, the rows of the matrix are reordered ('optimized') using Symmetric Approximate Minimum Degree Permutation (AMD) so that produces a matrix with a sparser Cholesky factor as compared to the original matrix. Typical distributions of the non-zero elements in the Mass matrix before and after reordering are shown in Figure 7.1, using MATLAB's implementation of functions for sparse matrix operations (symamd) (Amestoy et al. 2004).



(a)

(b)

Figure 7.1 Pattern of non-zeros in the SP7 MASS matrix (a) before and (b) after reordering/optimzation. The x and y axes correspond to the length of the Mass matrix, which is 169,544 (nm × 4, the number of composite moments of flux).

To illustrate the benefits of matrix optimization, equation 7.23 was solved both with and without re-ordering for a Mass matrix consisting of 1785×1785 elements. For comparison, the matrix inversion was performed using the Gaussian elimination method as well as the bi-conjugate gradient stabilized iterative method. The computation times for each method are shown in Figure 7.2. It can be seen that, when using the Gaussian elimination method, optimization of the Mass matrix has resulted in a 56% reduction in computation time. Optimization also led to a 42% decrease in computation time when using the iterative matrix inversion method.

In the case of a small Mass matrix, there is little difference in computation time between the Gaussian elimination method and the iterative BCGS method. This is not the case with larger Mass matrices, however. Equation 7.25 was solved again with a Mass matrix, consisting of 236250×236250 elements, to determine the benefits of optimization in larger problems encountered when using higher mesh resolutions for better numerical accuracy or in the case of the SP_N approximation where the Mass matrix is $4n_m \times 4n_m$ in size. Even before optimization, it can be seen that the Gaussian elimination is less efficient than the BCGS method, taking approximately 5% longer. After optimization, the computation time for the Gaussian elimination actually increased by 15%. It is with the iterative method, however, that the benefits of optimization are greatest. The solution of the optimized system was dramatically quicker, an 85% improvement.

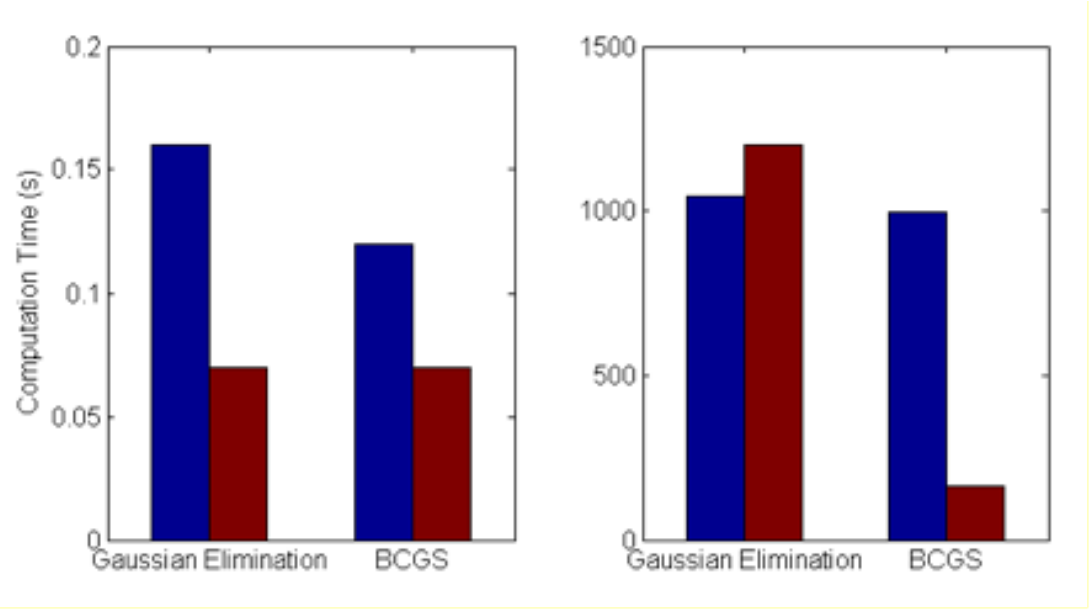


Figure 7.2 The computational time required to solve equation 7.25 using Gaussian Elimination and Biconjugate gradient stabilized (BCGS) methods for both a small (1785 ×1785 nodes) and large (236250 × 23650 nodes) Mass matrix. Blue bars show times before optimization, red bars show times for optimized matrices.

7.3.3 Running the forward model

In order to solve a forward problem, the *femdata_spn* function, which is provided in Appendix B, is used, where n=1,3,5 or 7. The function requires two input variables. The first, the FEM mesh, is stored as a structured variable containing information such as node coordinates and element linking, source/detector locations, optical properties, region number and an index to determine whether or not a node lies on the boundary. The second variable is the modulation frequency. In frequency domain problems, this is the frequency at which the intensity of the light source is modulated. Continuous Wave problems can be solved by simply setting the frequency to zero. The solver then outputs a structured variable containing boundary data for both phase and amplitude measurements as well as the internal fluence. The process of running the forward model is documented in Figure 7.3.

As higher order models are used, the size of the FEM problem increases correspondingly. This leads to an increase in the computational load required to solve the problem. The computation time required to solve a 3D problem are listed in Table 7.1. The models were run on a 3.0 GHz 64 bit Linux system with 8GB of physical memory using Matlab R2007b. The SP5 and SP7 models required the use of swap space and as such the computation time was considerably slower than expected.

Table 7.1 The computation times of the SPN models using an FEM mesh consisting of 42,386 nodes forming 235,869 tetrahedral elements and a Monte Carlo (MC) model simulating 10^6 photons.

Forward Model	Computation Time (minutes)
SP_1	4.2
SP_3	12.6
SP_5	31.7
SP_7	69.7
MC	28

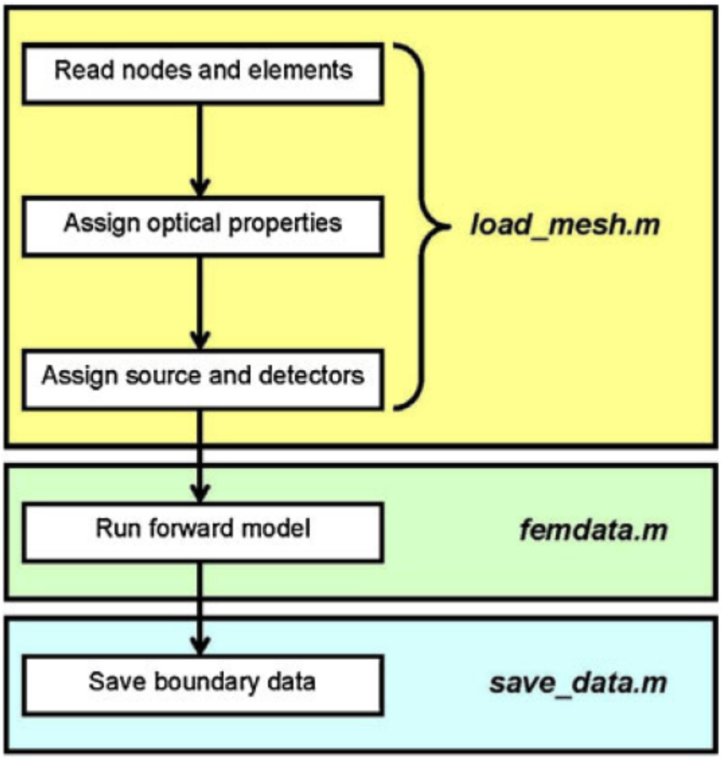


Figure 7.3 Flowchart outlining the process of running the forward model (Dehghani et al. 2008). A separate femdata routine exists for each of the SP_N orders.

7.4 Validation

7.4.1 Investigation of Mesh Resolution

The accuracy of the FEM is heavily dependent on mesh resolution as the numerical solution approaches the true solution as the element size approaches zero. In order for quantitative studies to performed, a numerically accurate mesh is vital. The choice of mesh resolution, however, tends to be based on a compromise between numerical accuracy and the computational expense of solving the problem.

In order to determine the optimal mesh resolution, a 1×1cm mesh with a single source located at the centre of one edge, Figure 7.4, was constructed. The resolution of the mesh was varied so that node spacing ranged from 0.3mm in the coarsest mesh down to 0.03mm in the finest mesh. After each increase in mesh resolution, the percentage difference in fluence data along a cross section, as indicated in Figure 7.4, was calculated and plotted in Figure 7.5.

It was determined that the optimal node spacing for quantitative studies would be approximately 0.11mm. The increase in accuracy of a higher resolution mesh would be small at just 1.5%, but would incur a significant increase in computation time.

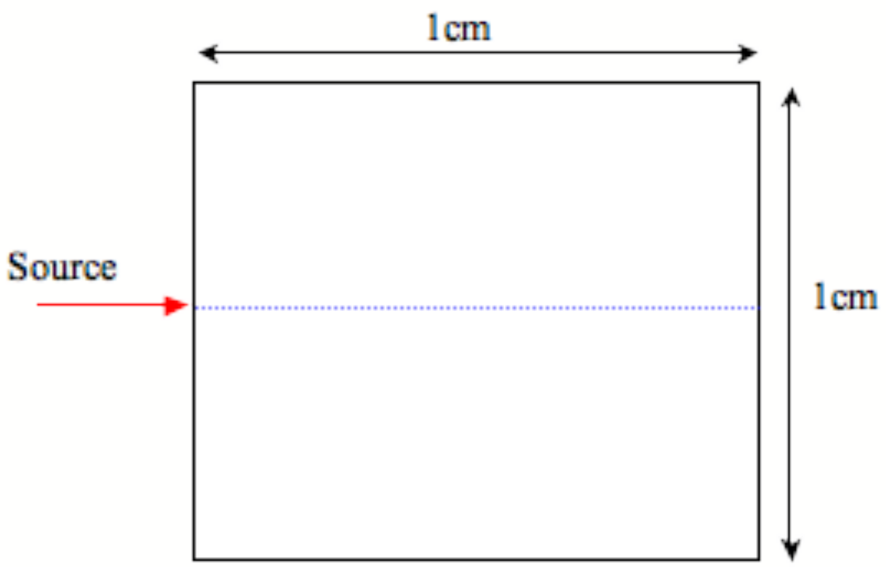


Figure 7.4 A 1cm × 1cm square geometry with μ_a = 0.001 mm^{-1}, μ_s = 1 mm^{-1} and a single source was used to test the effects of mesh resolution on forward model accuracy. A sample of the fluence was taken along the blue dashed line and compared to data obtained with higher mesh resolution.

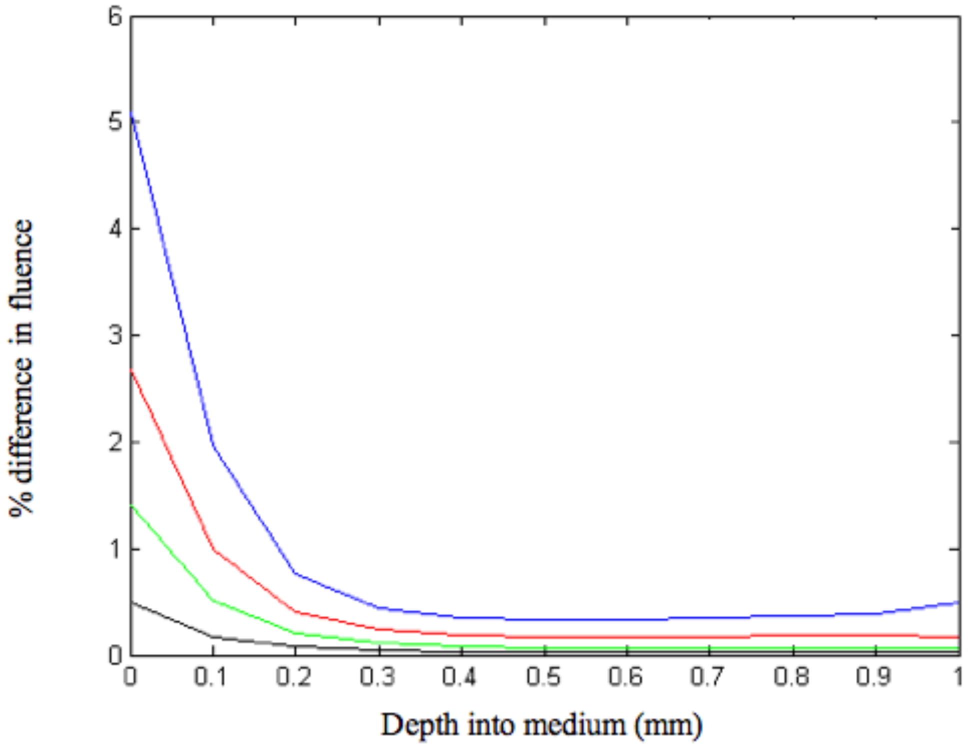


Figure 7.5 Percentage difference in fluence data between varying node spacings (resolution). Blue line: 0.3-0.17mm; red line: 0.17-11mm; green line 0.11-0.08mm; black line: 0.08-0.03mm

7.4.2 Monte Carlo Modelling

Monte Carlo (MC) models are capable of providing highly accurate results, provided that a sufficiently large number of photons are recorded. This requirement for large numbers of photons, however, leads to prohibitively lengthy computation times. Whilst the MC method may be of limited use as part of an image reconstruction algorithm, they are commonly used as 'gold standards' by which other forward solvers are tested.

An existing Monte Carlo model to simulate propagation of light in a semi-infinite turbid medium was extended to photon propagation in a stratified medium as described previously (Wang et al. 1995; Vishwanath et al. 2002). Briefly, the turbid-medium was modelled as having layers of finite thickness (along the *z*-axis) with specified transport coefficients and refractive indices in each layer. Photons were incident normally at the top face of the turbid medium. For every scattering event where the calculated step-size (along a particular direction) caused a photon to cross a boundary, the photon was first propagated to the point where its trajectory intersected the boundary via a shortened step-size. The angle of incidence with respect to ± *z*-axis (depending on the direction of photon travel) was computed and used to determine if the photon suffered total internal reflection (from Snell's law). If the photon was internally reflected, then the *z*-component of the photon's travel direction was reversed and the photon completed the remainder of the step in the same layer, otherwise the reflection coefficient from Fresnel's equations was computed and compared against a uniformly generated random number. For every sampling of the random number that was less than the reflection coefficient, the photon underwent total internal reflection, otherwise it was transmitted to the next layer (or escaped from the domain). On transmission into a different layer the final spatial location of the photon was calculated by propagating the photon by a distance that was adjusted in length (to account for the difference in transport coefficients between the two layers) and its direction corrected to consider refraction. All photons emanating from the top-layer of the turbid medium were spatially and temporally binned to calculate the reflectance from a turbid medium. Photons passing the surface were recorded in a series of concentric rings centred on the source. The transmittance is then divided by the area of the collection ring and by the number of injected photons so that the resulting measurement in terms of a probability per unit area. The MC data is obtained in the time domain with a pulse width of 10ns. The resulting temporal data was then Fourier transformed to give frequency-domain estimates of the amplitude and phase shift as a function of distance, in order to match the type of data used in our work.

7.4.3 Validating against MC data

In the following sections, we compare results from the developed SP_N model (N = 1, 3, 5 and 7) and MC model for a simple 3D slab of either homogenous or layered optical properties. For the MC results, the number of simulated photons was 1×10^6, with a total execution time of approximately 30 minutes (Table 7.1). All calculations were performed on a dual core 3.0 GHz 64bit Linux system with 8GB physical memory.

The validity of the diffusion approximation, and hence the SP_1 model in large medium problems and in cases where scattering dominates absorption have been previously shown demonstrating its validity by use of phantom data (Dehghani et al. 2003c) as well as Monte Carlo data (Dehghani et al.

2003a). The work presented here, therefore, concentrated on small volume domains and in cases of strong absorption, whereby the diffusion approximation is less valid.

The model domain used was a 3D slab of width and length of 40 mm (x coordinates) with a thickness of 20 mm (y coordinates) and depth of 30 mm (z coordinates), Figure 7.6. The source was placed at the center of the top most surface boundary and boundary reflectance measurements were made at 1 mm distances from 3 to 10 mm away from source. The source was modeled as a point source on the surface of the external boundary to most appropriately match the MC model, which used a collimated source located on the surface. The FEM mesh used in this study contained 42,386 nodes corresponding to 235,869 linear tetrahedral elements. The resolution of the mesh was increased in the areas surrounding the source in order to increase the numerical accuracy of the solution whilst minimizing the computational resources required.

Three individual cases were considered using both MC and SP_N models. In the first case, a homogenous medium with optical properties of $\mu_a = 0.001$ mm^{-1}, $\mu_s = 2.0$ mm^{-1}, g = 0.5 and n = 1.37. The computation times for the SP_1, SP_3, SP_5 and SP_7 calculations as well as the Monte Carlo computation time are shown in Table 7.1. The calculated amplitude and phase of the reflectance boundary data, from the 3D model (Figure 7.6) and the semi-infinite MC model are shown in Figure 7.7. The calculated 3D fluence from SP_7 and SP_1 were also used to calculate the percentage error map (error = $100\times(\Phi_{SP1}-\Phi_{SP7})/\Phi_{SP7}$), with the 2D cross-section at y = 0mm shown in Figure 7.7(c). As shown, the largest errors are seen near the source and detectors with errors as high as 60%.

Figure 7.8(a) shows the percentage error between amplitude data calculated by the SP_N approximations and the MC data at each of the detector positions. It can be seen that the SP1 approximation underestimates the amplitude at all distances from the detector. The higher order models, however, initially overestimate the amplitude but underestimate at distances greater than 4 cm from the source. In all models, however, it can be seen that the percentage error increases with distance from the source. This is due to the propagation of error throughout the domain and has been previously reported in other studies (Hielscher et al. 1998). The mismatch between SP_N and MC phase data is displayed in Figure 7.8(b).

The average percentage error and RMS mismatch in phase data for each SP_N order is listed in Table 7.2. In this case, the SP_5 approximation provides the lowest degree of error in both amplitude (-4.1%) and phase (0.7°) with respect to the MC data. The SP_3 data provides the next best fit with an average error of -5% in amplitude data compared to -5.7% for the SP_7 data. The SP_3 and SP_7 data, however, provide the same degree of accuracy in phase. The SP_1 or diffusion approximation results in the poorest match in both sets of data with an average -13.1% error in amplitude and RMS phase mismatch of 1.6°, approximately twice as large as the errors seen in the higher ordered data.

It should be noted that as the derivation of the SP_N approximation removes the requirement for the modeled data to approach the true solution at increasing orders, the highest ordered approximation will not necessarily result in the most accurate solution. As such, the most accurate ordered approximation will be dependent upon the system being modeled.

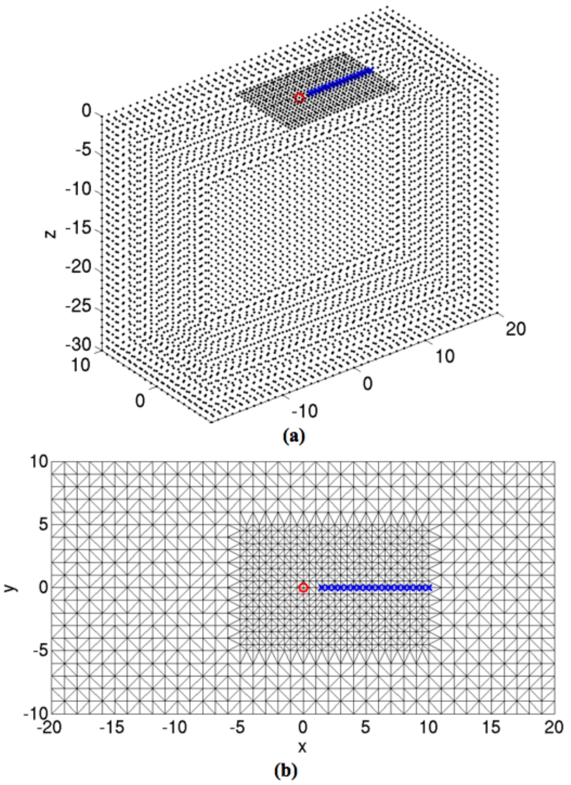
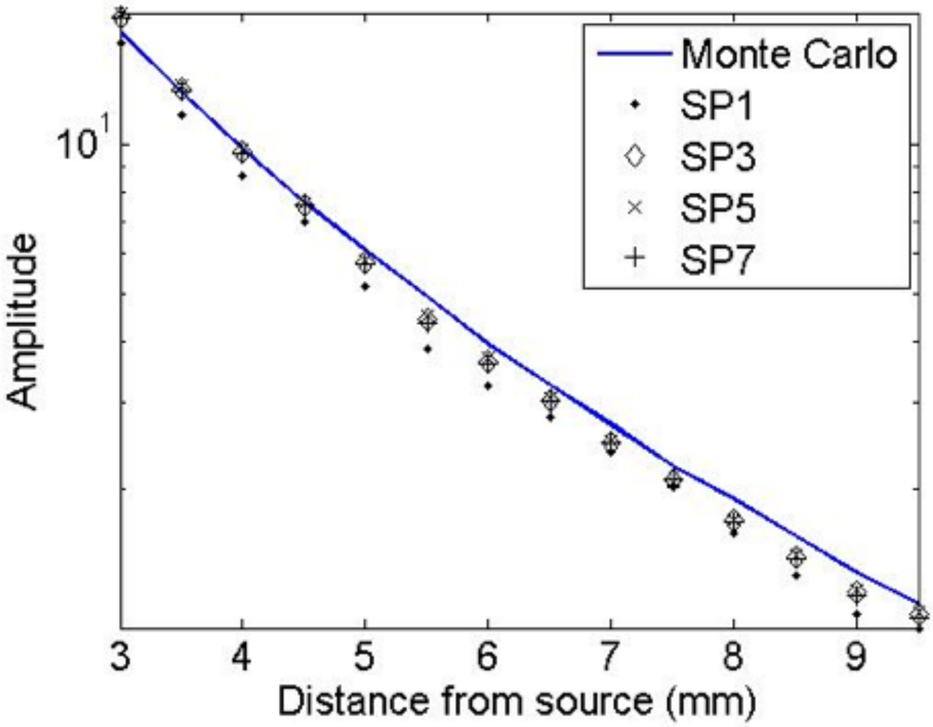
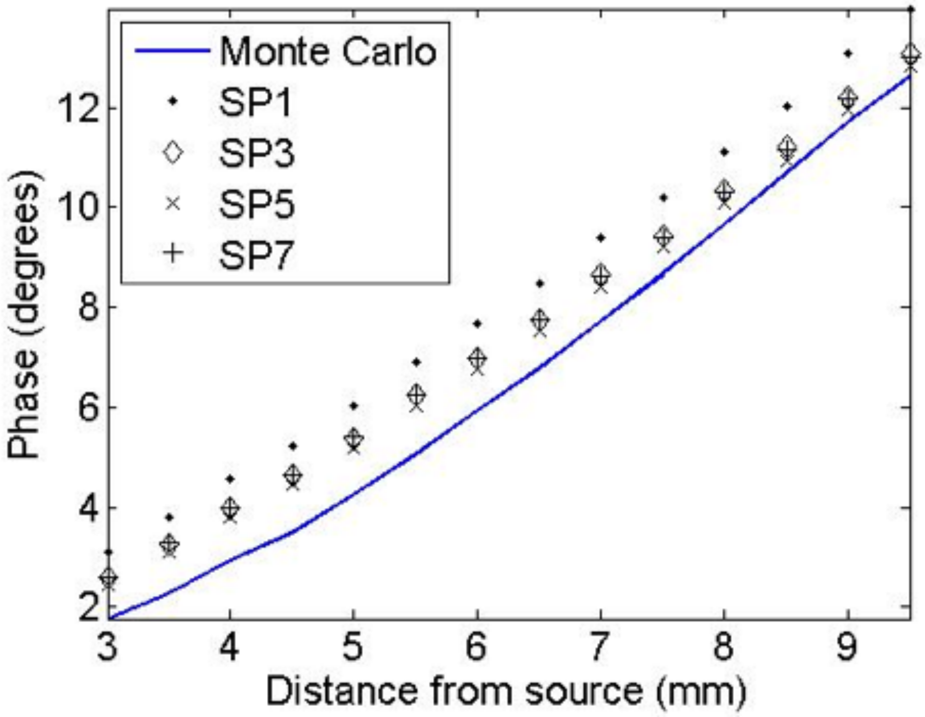


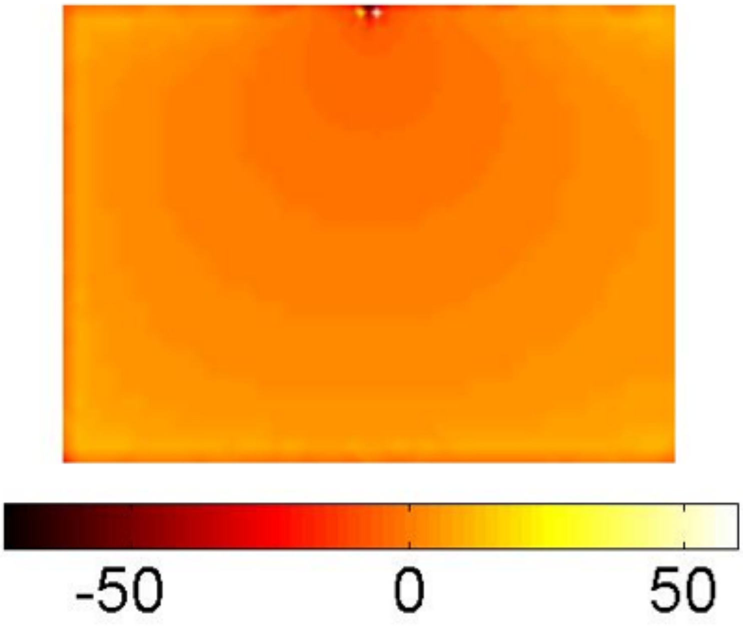
Figure 7.6 3D mesh used for the calculation of data using the SPN models (a) 3D view of the surface nodes and (b) 2D representation of the top surface. Red circle represents location of the source and blue cross represent location of the detectors.


Amplitude
10^1
3
4
5
6
7
8
9
Distance from source (mm)
Monte Carlo
SP1
SP3
SP5
SP7

(a)


Phase (degrees)
2
4
6
8
10
12
3
4
5
6
7
8
9
Distance from source (mm)
Monte Carlo
SP1
SP3
SP5
SP7

(b)



(c)

Figure 7.7 The calculated boundary data (a) log amplitude and (b) phase for SP_N and Monte Carlo model for uniform optical properties of $\mu_a = 0.001$ mm^{-1}, $\mu_s = 2.0$ mm^{-1}, g = 0.5 and n_m = 1.37. (c) is the cross sectional percentage error between SP7 and SP1 for the calculated fluence at y = 0 mm corresponding to the plane at which the source is placed.

Table 7.2: Summary of errors in SP_N data relative to MC data

	Average % Error in Amplitude	RMS mismatch in phase data (°)
SP_1	-13.1	1.6
SP_3	-5.0	0.9
SP_5	-4.1	0.7
SP_7	-5.7	0.9

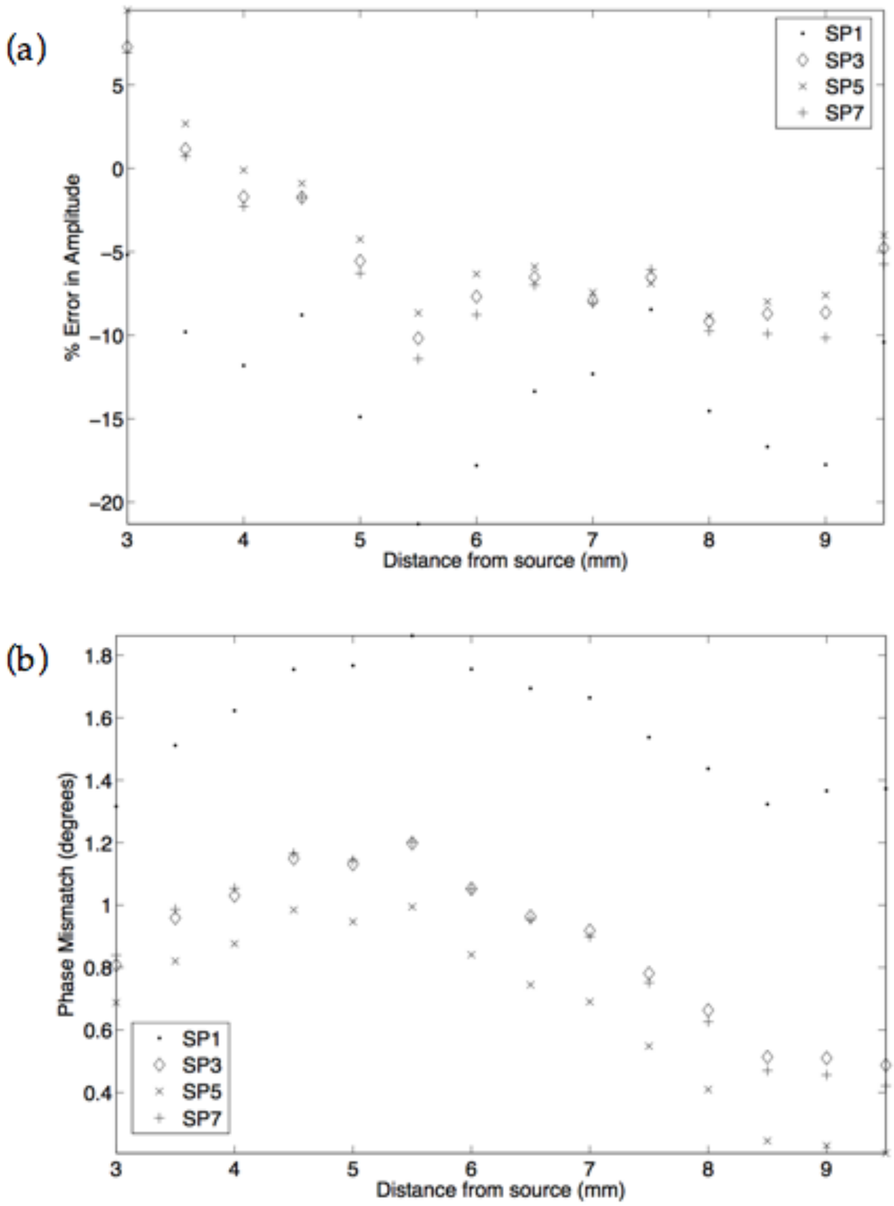
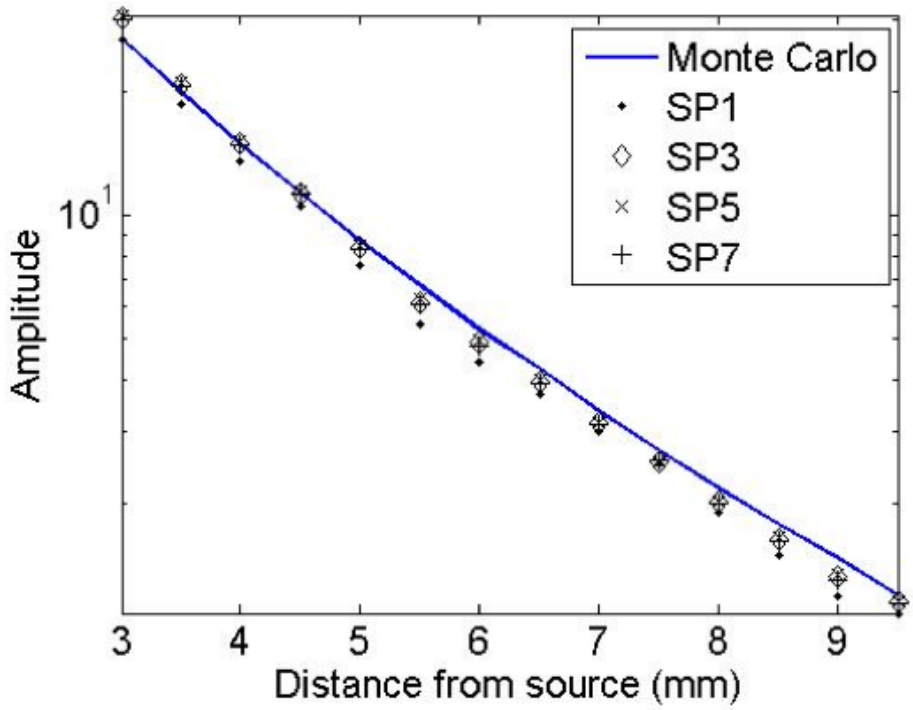


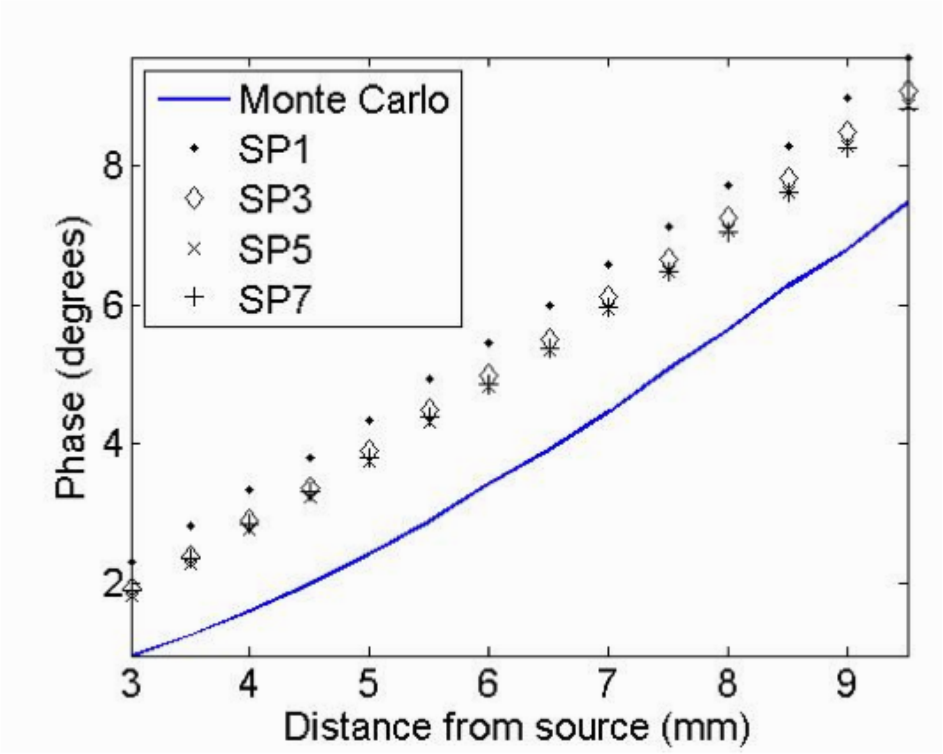
Figure 7.8: (a) Percentage error beween SPN and MC data with increasing distance from the source for the results displayed in Figure 7.7 (b) the mismatch in SPN and MC phase data

Using a second homogenous model with optical properties of μ_a = 0.01 mm^{-1}, μ_s = 2.0 mm^{-1}, g = 0.5 and n = 1.37 the boundary measurements and internal fluence were also calculated and compared, Figure 7.9. The percentage error in amplitude and phase mismatch at each detector position is plotted in Figure 7.10. As in the previous case, the SP$_5$ approximation provides the closest match to the MC data with average errors of -2.8% and

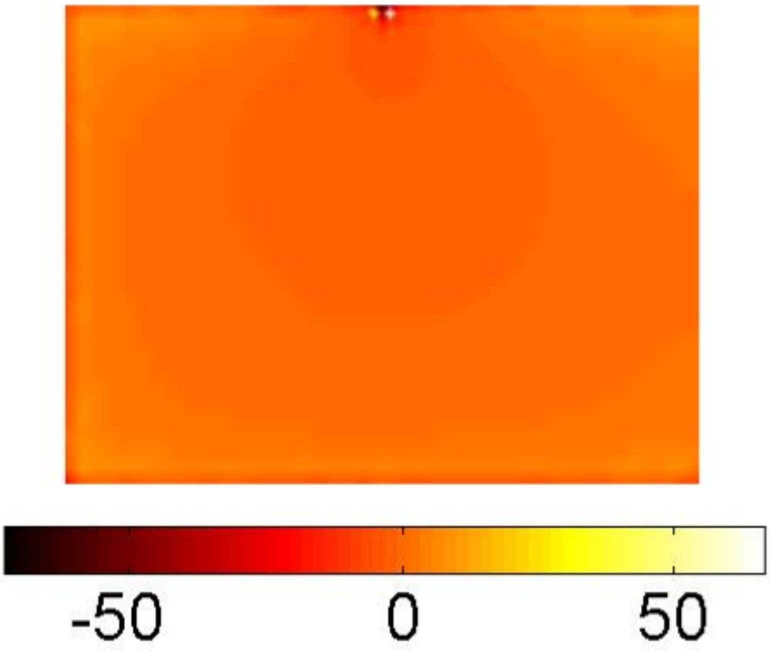
1.3^o in amplitude and phase data respectively. The SP$_3$ approximation, again, provides a better match to the MC amplitude data than SP$_7$ but, in this case, a slightly poorer match in phase data. The SP$_1$ approximation again shows a significantly larger error of -11.4% in amplitude data and the largest phase mismatch at 2^o. As shown in Figure 7.9c, errors of up to 60% are again seen between SP$_1$ and SP$_7$ fluence data, specifically near the source.


Amplitude
10^1
3
4
5
6
7
8
9
Distance from source (mm)
Monte Carlo
SP1
SP3
SP5
SP7

(a)


Phase (degrees)
2
4
6
8
3
4
5
6
7
8
9
Distance from source (mm)
Monte Carlo
SP1
SP3
SP5
SP7

(b)



(c)

Figure 7.9 Same as Figure 7.7, but for uniform optical properties of μa = 0.01 mm-1, μs = 2.0 mm-1, g = 0.5 and nm = 1.37

Table 7.3: Summary of error in SP_N data relative to MC data for case 2

	Average % Error in Amplitude	RMS mismatch in phase data (°)
SP_1	-11.4	2.0
SP_3	-3.7	1.6
SP_5	-2.8	1.3
SP_7	-4.8	1.4

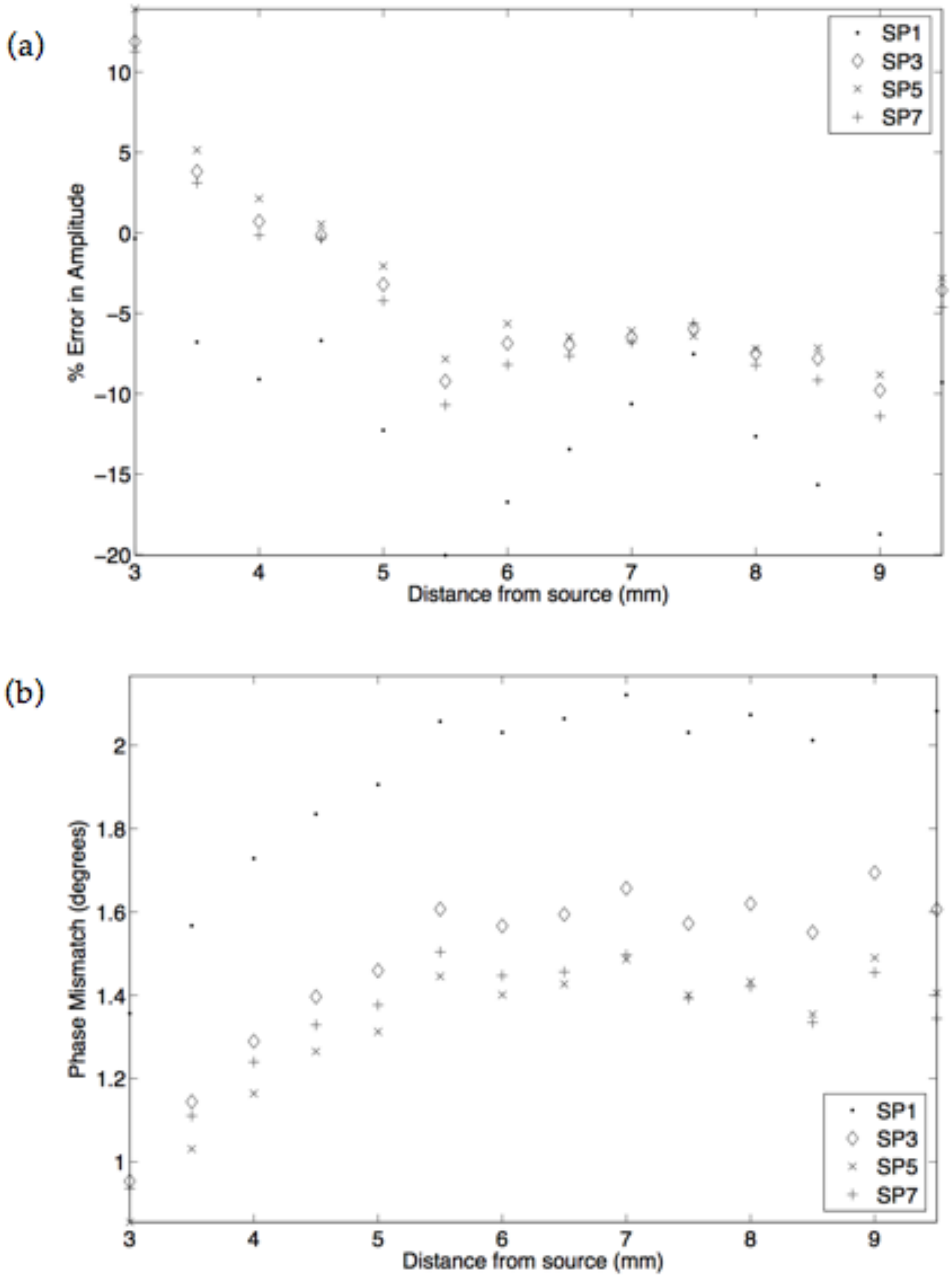


Figure 7.10: (a) Percentage error beween SPN and MC data with increasing distance from the source for the results displayed in Figure 7.9, (b)the mismatch in SPN and MC phase data

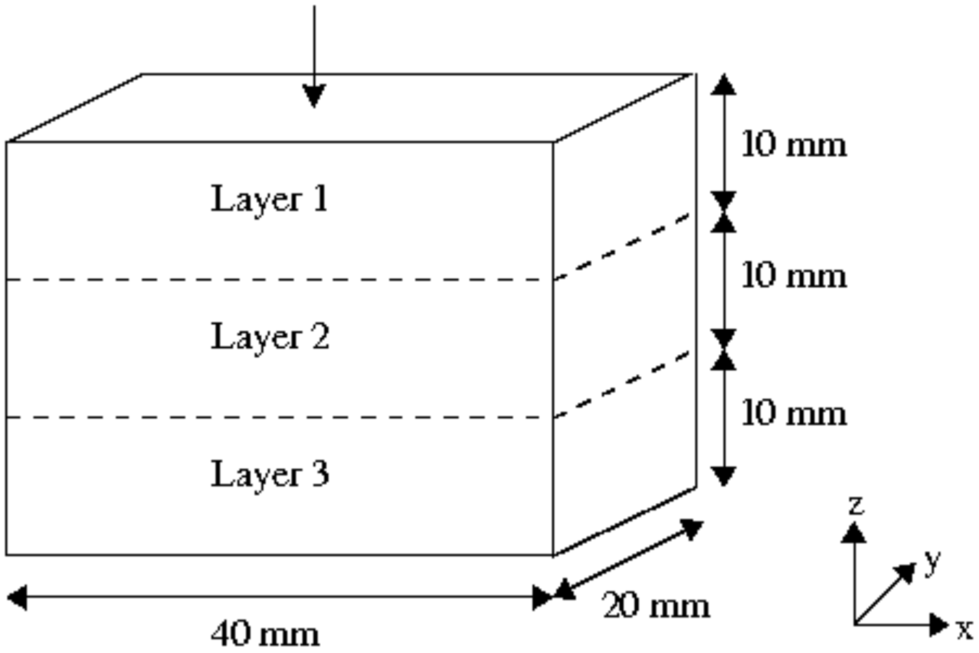
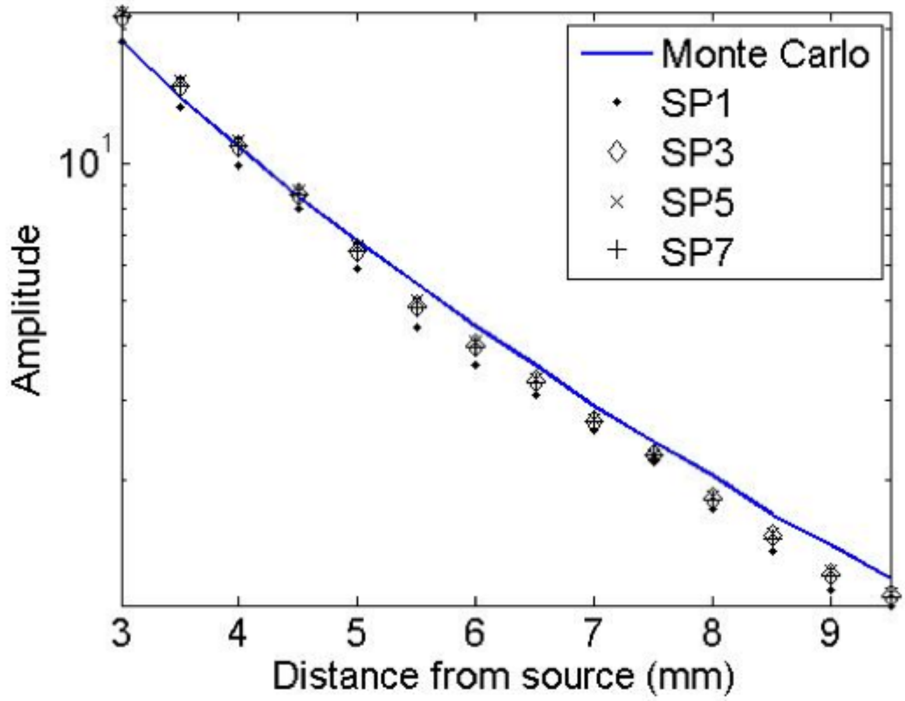
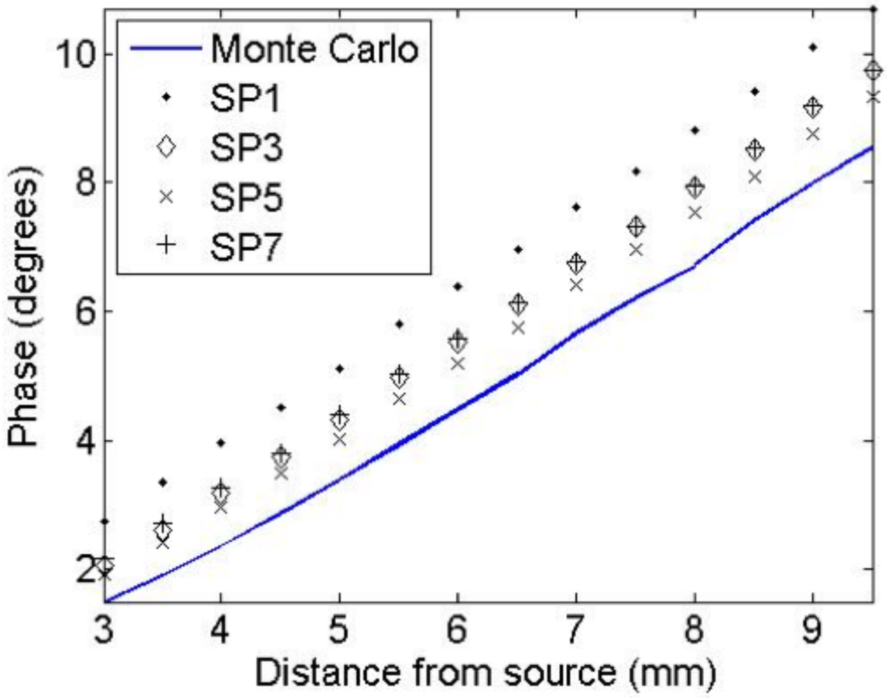


Figure 7.11 Schematic of the 3D layered model. Layer 1 and layer 3 have optical properties of μa = 0.001 mm-1, μs = 1.0 mm-1, g = 0 and nm = 1.37 and layer 2 has optical properties of μa = 0.2 mm-1, μs = 2.0 mm-1, g = 0.5 and nm = 1.37. The arrow indicates the location of the point source on the boundary

In order to investigate the effects of a layered heterogeneous model of varied optical properties, the same 3D geometry as previous cases where used, but containing 3 layers, as shown in Figure 7.11. In this case, the model represents a 3 layered case, whereby each layer has a thickness of 10 mm with the middle layer having a much stronger absorption (contrast of 200 in absorption) and scatter (contrast of 2 in scatter and increased anisotropic factor, g) as compared to the top and bottom layer. The results from both the SP_N and MC solution are shown in Figure 7.12. The SP_5 and SP_7 model provides the best match with MC for both the amplitude and phase. The cross section of the percentage error between SP_1 and SP_7, Figure 7.12c, again shows the largest error near the sources as well as the regions with and beyond the strong absorbing layer (layer 2). In this case, errors of up to 85% are seen with layer 2, indicating that the solution of SP_1 to be dramatically different to that of SP_5 and SP_7.


Monte Carlo
SP1
SP3
SP5
SP7
Amplitude
10^1
3
4
5
6
7
8
9
Distance from source (mm)

(a)


Monte Carlo
SP1
SP3
SP5
SP7
Phase (degrees)
10
8
6
4
2
3
4
5
6
7
8
9
Distance from source (mm)

(b)

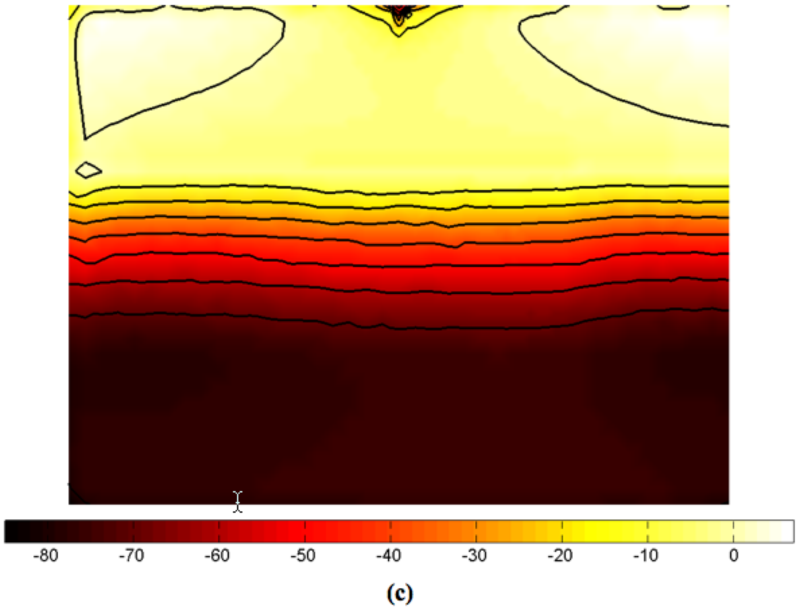


Figure 7.12 Same as Figure 7.7, but for the layered model as shown in Figure 7.11.

7.5 Discussion

In this chapter, the SP_N equations have been derived and presented for frequency domain problems. These equations are then implemented using the FEM for both 2D and 3D problems. The SP_N approximation was then validated using Monte Carlo data using a slab geometry for a range of optical properties.

The calculated boundary data using MC and developed 3D frequency domain modified simplified spherical harmonics expansion for 3 cases have been compared. It should be noted that the MC model does not provide an exact replication of the conditions modelled by the SP_N-FEM model. Whereas the SP_N data was obtained using a diffuse source, modelled by positioning it one scattering distance into the domain, the MC model uses a collimated source positioned on the boundary. As such, a small degree of mismatch is to be expected. As all of the SP_N data was obtained with the same source conditions, however, the discrepancies will be constant between all orders. To avoid artificially biasing the results towards one of the SP_N orders, however, in all of the results presented, the data has not been scaled, nor any offset added, therefore providing the best means of comparison between different models and methods.

In the first case, whereby the scattering properties were much higher than absorption (μ_a = 0.001 mm^{-1}, μ_s = 2.0 mm^{-1}, g = 0.5 and n_m = 1.37), it was found that the SP_5 approximation provided the best match to the MC data with an average error in amplitude of -4.1% and an RMS phase mismatch of 0.7°. The SP_1 approximation provided the poorest fit to the MC data with an average error of -13.1%, more than double that of the next best SP_N order (SP_7). The error in amplitude data appeared to increase with distance from the source as small errors are amplified. Conversely, the error in phase data, which is more dependent upon scattering properties, reduced with increasing distance from the source. This is expected as photons reaching the furthest detectors have undergone more scattering events resulting in an effectively diffuse distribution. In order to analyse the degree of error between N = 1 and N = 7, the percentage error between the total fluence calculated is shown for a cross-section of the 3D model at y = 0 mm (directly under the source), Figure 7.7(c). It is seen that the largest error (as large as 60%) is seen near the source, indicating that the suitability of the SP_N method lies within regions of close proximity to the source.

In the second case, whereby the absorbing properties were increased (μ_a = 0.01 mm^{-1}, μ_s = 2.0 mm^{-1}, g = 0.5 and n_m = 1.37), the SP_5 approximation again results in the closest match to the MC data in both phase and amplitude data. The SP_3 and SP_7 approximations provide a similar level of accuracy in both sets of data. The SP_1 data, however, results in a significantly higher error in amplitude data at 11.4%, again, more than double the largest error found in the higher ordered models. Whilst the SP_1 phase data also provides the poorest match, the relative difference is not as significant as in amplitude.

As in the previous case, the errors in amplitude data increased with distance from the source. The phase data, however, showed the opposite trend to the previous case with errors increasing further from the source. This is most likely due to fewer photons reaching these areas, due to the increased absorption coefficient, resulting in poorer statistics in the MC data.

In order to evaluate the effect of layered models with much higher absorption, data using a layered MC and SP_N model are shown in Figure 7.12. As with previous cases, the calculated data using $N = 1$ provides the least accurate solution as compared to MC. Figure 7.12c shows the calculated error between the total fluence as obtained using $N = 1$ and $N = 7$. It is seen that the magnitude of error is largest not only near the source, but also at the interface of two boundaries whereby the absorption coefficient increases. This is of importance given that some biological tissue, such as the liver which has a much higher blood content, will have a much greater absorption and appropriate modelling of such a situation is of paramount importance in small animal imaging. Similar results showing the effect of increased absorption within a medium have been shown in 2D (Klose and Larsen 2006).

It is important to point out, that due to the asymptotic nature of the problem, no single order (N) will provide the most accurate solution. However, it is demonstrated here that the use of higher order approximations ($N>1$) provides a much more accurate solution, specifically near source locations, which may be of importance when dealing with imaging of small volumes. Whilst it was seen that the SP_N data matched the MC data more closely than the diffusion data, they did not match perfectly. In order to make a like for like comparison, the data in all three cases has been shown without scaling or offsetting and as such, small differences between the two independent methods are unsurprising.

The amplitude and phase data in all three cases was studied only for the first 10mm away from the source only. This has led to errors between the $SP_{N>1}$ and SP_1 data in cases even when scattering dominates over absorption since the light distribution is still forwardly biased near the source. As stated earlier, the validity of the diffusion approximation has been previously demonstrated for larger source/detector separation using both phantom data (Dehghani et al. 2003c) as well as Monte Carlo data (Dehghani et al. 2003a), and this work has concentrated in situation where the diffusion approximation is known to be inaccurate.

The timings for the SP_N and Monte Carlo models are displayed in Table 7.1. It can be seen that the run-time of the Monte Carlo model was just seven times that of the SP_1 model and was in fact quicker than the SP_5 and SP_7 models which was unexpected. Due to the very large number of nodes in the FEM mesh, however, the Mass matrices were correspondingly large and their inversions required large amounts of memory. In the SP_5 and SP_7 cases, the memory requirements exceeded the physical memory available and therefore had to use 'swap space' which drastically increased the computation times. Also, the Monte Carlo data was obtained based on the paths of just 10^6 photons. While this was enough to obtain smooth data, this is a relatively small number of photons for Monte Carlo studies. A more quantitative study may use over ten times this number of photons leading to significantly longer computation times. In studies using meshes of lower resolutions where the use of swap space was not required, it was found that the SP_5 and SP_7 models took approximately $5t_d$ and $8t_d$ respectively where t_d is the computation time of the SP_1 model.

7.6 Conclusions

It is well known that the accuracy of the diffusion approximation is limited in regions of strong absorption and within regions before the NIR light becomes diffuse (i.e. near source locations). In such cases higher ordered approximations, such as the P_N and S_N methods are required and whilst these methods have already been applied to tissue optics with promising results (Boas et al. 1994; Aydin et al. 2002), they both introduce a prohibitively large number of unknowns making them computationally expensive to solve. The SP_N approximation has the advantage of providing an increase in accuracy over the diffusion approximation whilst requiring fewer unknowns.

In this study, a three-dimensional frequency domain light transport model based on the Simplified Spherical Harmonics approximation has been developed. The model was used to calculate data for three individual cases with the results being compared to data obtained using a Monte Carlo model. In each of the three cases presented, the SP_N approximations with $N > 1$ were shown to provide an increase in accuracy over the SP_1 approximation in both the phase and amplitude of the boundary data. Comparisons of the fluence data from the SP_1 and SP_7 models showed that there is a significant difference in the regions surrounding the source where the light distribution is forwardly biased. The final case highlighted the limitations of the diffusion approximation in handling regions of strong absorption. Whilst the fluence data showed good agreement in the initial diffuse region, major differences of up to 85% were evident when the absorption was increased.

It was observed that the most accurate solution to a given problem was not necessarily provided by the highest ordered approximation, which has also been noted in previous studies into SP_N methods (Klose and Larsen 2006). This is due to the fact that unlike other methods which converge on the RTE solution as $N \rightarrow \infty$, the SP_N approximation only asymptotically approaches the RTE solution as N increases.

The effect of anisotropy has been previously demonstrated to be significant in highly absorbing media and must be modelled accurately using techniques presented here (Klose and Larsen 2006). The impact of modelling anisotropic scattering on image reconstruction is a subject of ongoing studies.

The SP_N methods would be of particular importance in imaging systems that utilize wavelengths approaching the visible region where absorption due to both oxygenised and deoxygenised haemoglobin is strong. As the SP_N methods require fewer equations they are a more favourable forward model for image reconstruction.

The Monte Carlo model used in this study only calculated intensity data at a series of detectors located on the surface meaning comparisons of internal fluence were not possible. Modifications to the Monte Carlo model to include the recording of internal fluence data are left for future work.

Chapter 8 - Image Reconstruction Using the SP_N Approximation

8.1 Introduction

As discussed in chapter 6, the aim of the inverse problem is to calculate the distribution of optical properties within a domain from a set of boundary measurements. In frequency domain DOT, this boundary data consists of amplitude and phase measurements of the exiting light at a number of detector locations.

The inverse problem is generally solved using model based iterative image reconstruction (MOBIIR) techniques which involve estimating the optical property distribution and iteratively running the forward model with the aim of minimising the difference between the measured data Φ^M and data calculated by the forward model Φ^C.

This chapter will discuss the solution of the inverse problem and introduce an image reconstruction algorithm based on the simplified spherical harmonics approximation. In sections 8.2 – 8.4, the Jacobian matrix is constructed using the perturbation method to ensure accuracy. In the remaining sections, the reciprocity approach, which has been previously validated using the perturbation method for the DA (Arridge and Schweiger 1995), was used.

8.2 Mapping Sensitivity to Changes in Optical Properties

In the commonly used DA, the scattering coefficient and anisotropy factor are combined into the reduced scattering coefficient. As such, image reconstruction using the diffusion approximation generally aims to recover maps of absorption coefficient and reduced scatter coefficient throughout the imaging domain. There has been some discussion, however, over whether or not it would be possible during image reconstruction to separate between the scattering coefficient and anisotropy. As the implementation of the SP_N approximation, as discussed in chapter 7, does not combine the two values, it may be possible to reconstruct them individually.

The sensitivity of the SP_7 models to changes in optical properties were studied using a circular geometry of 43 mm radius with 16 equally spaced and collocated sources and detectors as shown in Figure 8.1. The Jacobian was constructed using the perturbation method before being mapped onto the mesh coordinates to produce a map of sensitivity to changes in optical properties. The reference data was calculated using homogeneous optical properties of μ_a = 0.01 mm^{-1}, μ_s = 10 mm^{-1}, g = 0.9 and refractive index (n) = 1.33 using a modulation frequency of 100MHz. The sources were modelled as point sources located at 1 mm inside the boundary, to correspond with one reduced scattering distance as in DA.

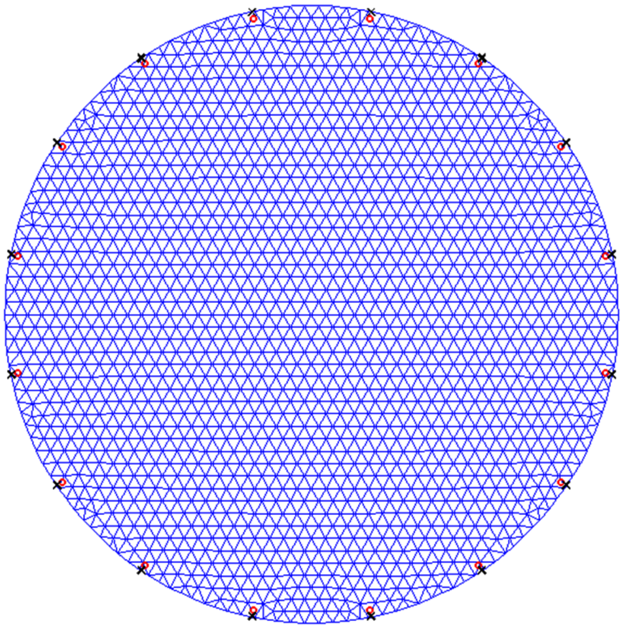


Figure 8.1 FEM mesh, of 43mm radius, containing 1785 nodes and 3418 linear triangular elements. Circles represent location of sources, crosses represent location of detectors.

The total sensitivity of log amplitude for all sources and detectors pairs to changes in absorption coefficient is mapped in Figure 8.2a. Areas of high sensitivity exist in the regions surrounding the sources and detectors with a much lower background sensitivity decreasing, as expected, towards the centre of the domain. The negative magnitudes of sensitivity confirm that increases in absorption lead to reductions in measurements of intensity. Figure 8.2b shows the sensitivity of same data-type to scattering coefficient and of similar trend to the previous case. The sensitivity scale for scattering is again negative but has a much lower magnitude than that of absorption. Figure 8.2c shows that the sensitivity of intensity measurements to changes in anisotropy factor and is again highly localized to the sources and detectors. The magnitude of sensitivity of g is several orders lower than that of both absorption and scattering.

Figure 8.2a-b show that increases in both the absorption and scattering coefficients lead to decreases in intensity measurements. This suggests that it is not possible to distinguish between the two effects using intensity data alone. A decrease in intensity, for example, could be caused by either a small increase in absorption or a large increase in scattering.

Figure 8.3a shows the sensitivity of phase measurements to changes in absorption coefficient. It is evident that the sensitivity is greater in the regions surrounding the sources and detectors as compared to the regions in between them. Additionally, the sensitivity although increasing to a maximum at a few scattering depths inside, it then decreases towards the centre of the domain, which is consistent with previous findings (Arridge 1995a). The scattering sensitivity map, Figure 8.3b, shows regions of very high sensitivity near the sources and detectors with much lower sensitivity elsewhere. The magnitude of scattering sensitivity is again much lower than that of absorption but with a positive scale. Sensitivity of phase measurements to changes in anisotropy factor is again limited to the regions surrounding the sources and detectors with a much lower magnitude as compared to absorption and scatter.

The results show that using log amplitude data alone, it is not possible to separate changes due to perturbations in absorption and scatter properties since both result in the same effect (i.e. increase in either absorption or scatter results in a decrease in log amplitude measurements). With the introduction of phase data, however, it can be seen that changes in amplitude and scatter have opposite effects on such data and so it is possible to distinguish between perturbations in absorption or scatter. However, since scattering and anisotropy factor have opposite sensitivities for both data-types, an increase in one parameter can be compensated by a decrease in the other and vice-versa. Therefore the problem appears non-unique for reconstruction of scatter and anisotropy factor.

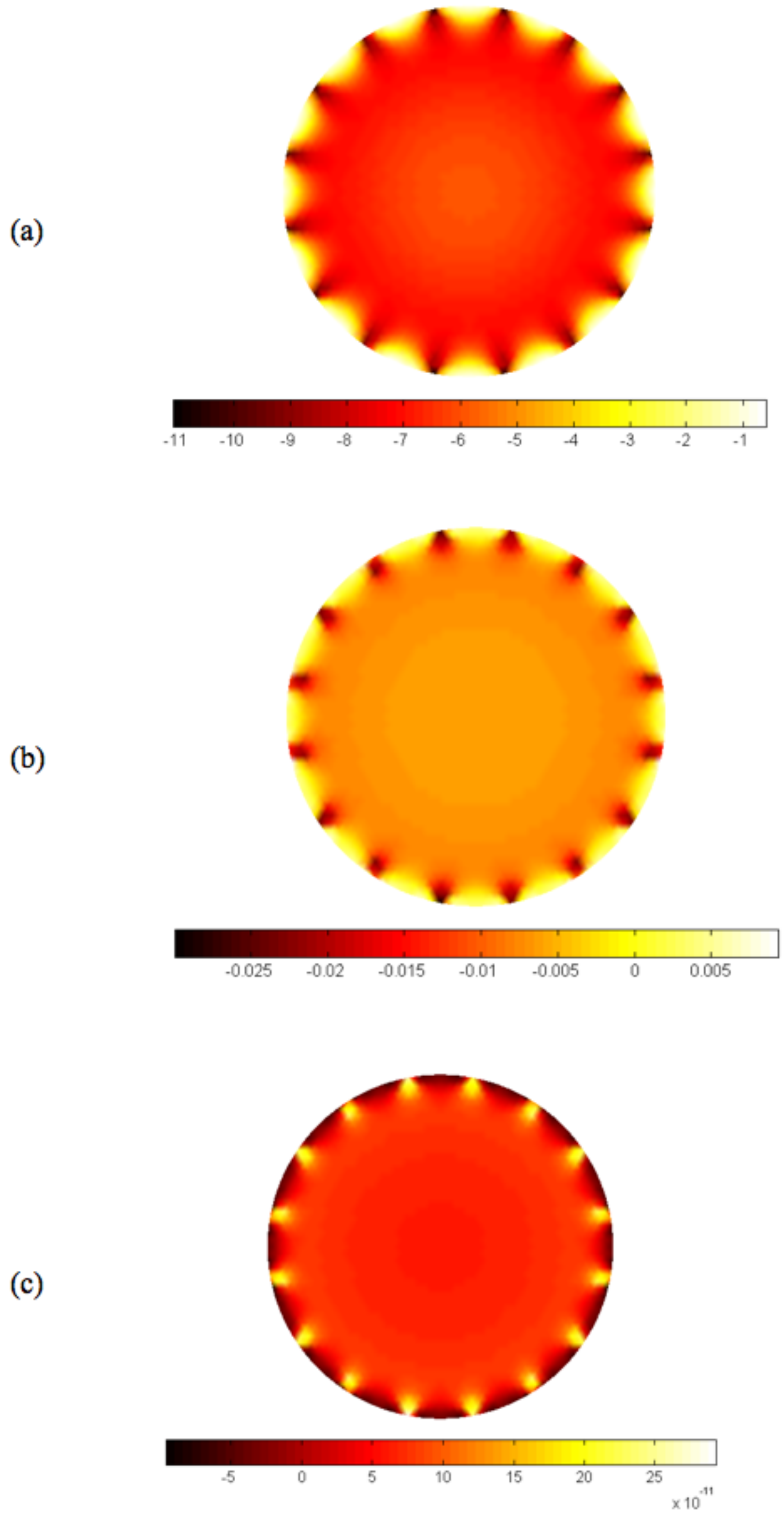


Figure 8.2 Maps of sensitivity of log Amplitude data to changes in a) absorption, b) scattering and c) anisotropy

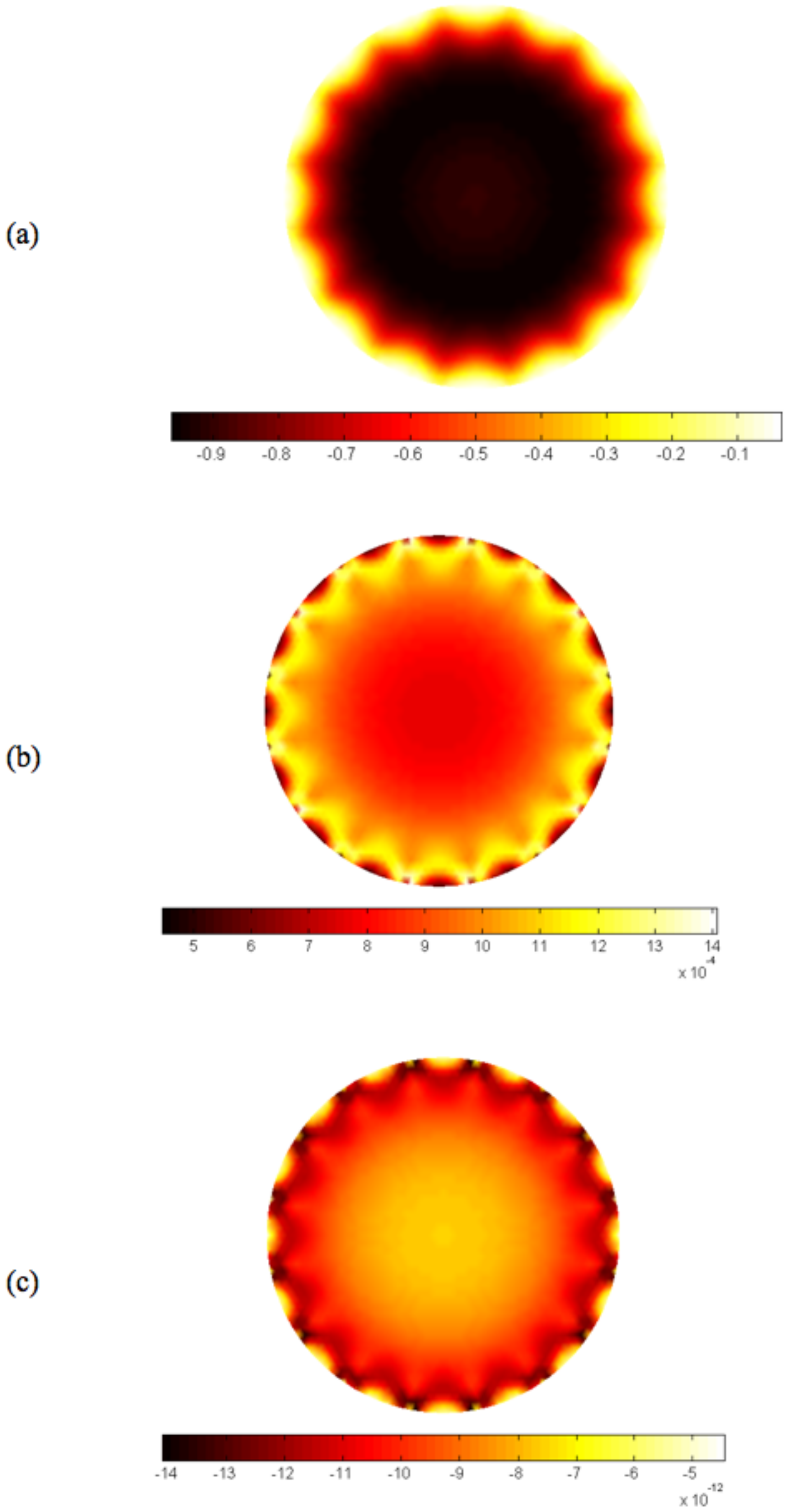


Figure 8.3 Maps of sensitivity of phase data to changes in a) absorption, b) scattering and c) anisotropy

8.3 Uniqueness of boundary data

Next the uniqueness of the imaging problem is analyzed. The problem of non-uniqueness arises when more than one set of optical property distributions lead to an identical set of boundary data, in which case, the simultaneous recovery of multiple optical properties, such as the absorption and scattering coefficients, is not possible (Arridge and Lionheart 1998).

The previously used circular mesh, Figure 8.1, was used to test the uniqueness of the SP_7 equations. A circular anomaly, with radius 10 mm, was inserted into the mesh as shown in Figure 8.4. The optical properties within the anomaly were perturbed with either the absorption coefficient ranging between 0.0125mm^{-1} to 0.0375mm^{-1}, scattering coefficient ranging between 10mm^{-1} to 30mm^{-1} and anisotropy factor ranged between 0 and 1 and corresponding boundary data for all possible combinations were calculated. These wide range of values are chosen in order to allow a comprehensive study of their effect on problem uniqueness. Reference data was also calculated using the optical properties of μ_a = 0.025 mm^{-1}, μ_s = 20 mm^{-1} and g = 0.9. The absolute error (L1 norm) for each data-type between the reference and each set of perturbed data was the defined as

$$\delta = \left|\sum\left(\phi_{ref} - \phi_{\mu}\right)\right| \tag{8.1}$$

where ϕ_{ref} is the reference homogeneous unperturbed boundary data (either phase or log amplitude) and ϕ_{μ} is the perturbed data. Figure 8.5a shows the map of error (equation 8.1) in amplitude data for each combination of μ_a and μ_s. A large band of optical property combinations exists for which the error between the reference data and perturbed data falls to zero. These equivalent combinations lead to identical amplitude data at the boundary and, as such, amplitude data alone is insufficient to recover both absorption and scattering properties simultaneously. By introducing phase measurements, however, the number of non-unique solutions can be minimized. Figure 8.5b shows the error map of phase measurements for the same range of optical properties. In this case, the optical property combinations leading to identical boundary data lie within a much more localized region. By combining the two data types, however, Figure 8.5c, the solution can converge on a smaller range of optical properties that lead to the perturbed boundary data.

The study was then repeated with varying absorption coefficients and anisotropy factors. Figure 8.6a shows that a large band of non-unique optical property combinations exists for amplitude measurements. The errors in phase measurements, Figure 8.6b, show an even larger range of equivalent combinations. When combining the two data types, however, it can be seen that the number of μ_a and g combinations leading to the reference data actually falls within a smaller narrow region, Figure 8.6c.

The map of error in amplitude measurements with varying μ_s and g, are also shown in Figure 8.7. Figure 8.7a, shows a large band of equivalent combinations of the two parameters for log amplitude data. The error in phase data actually shows two large regions of non-unique optical property combinations. The band of equivalent combinations in the amplitude error map is parallel to one of the bands in the phase error map. As such, even when combining the two data types, a large band of equivalent combinations still exists, Figure 8.7c, and therefore, even with both phase and amplitude data, it is not possible to distinguish between changes due to μ_s and g.

By direct visual comparison between Figure 8.6c and Figure 8.7c, it is evident that there exists a large range of absorption, scatter and anisotropy factor that would give rise to the same boundary measurements, even when considering both data-types, and hence indicating the non-uniqueness of the problem to reconstruct all three parameters.

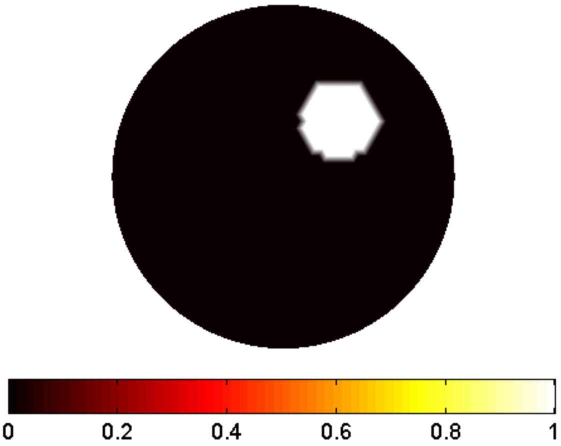


Figure 8.4 Circular mesh with a single inclusion of 10mm radius used to test uniqueness of boundary measurements.

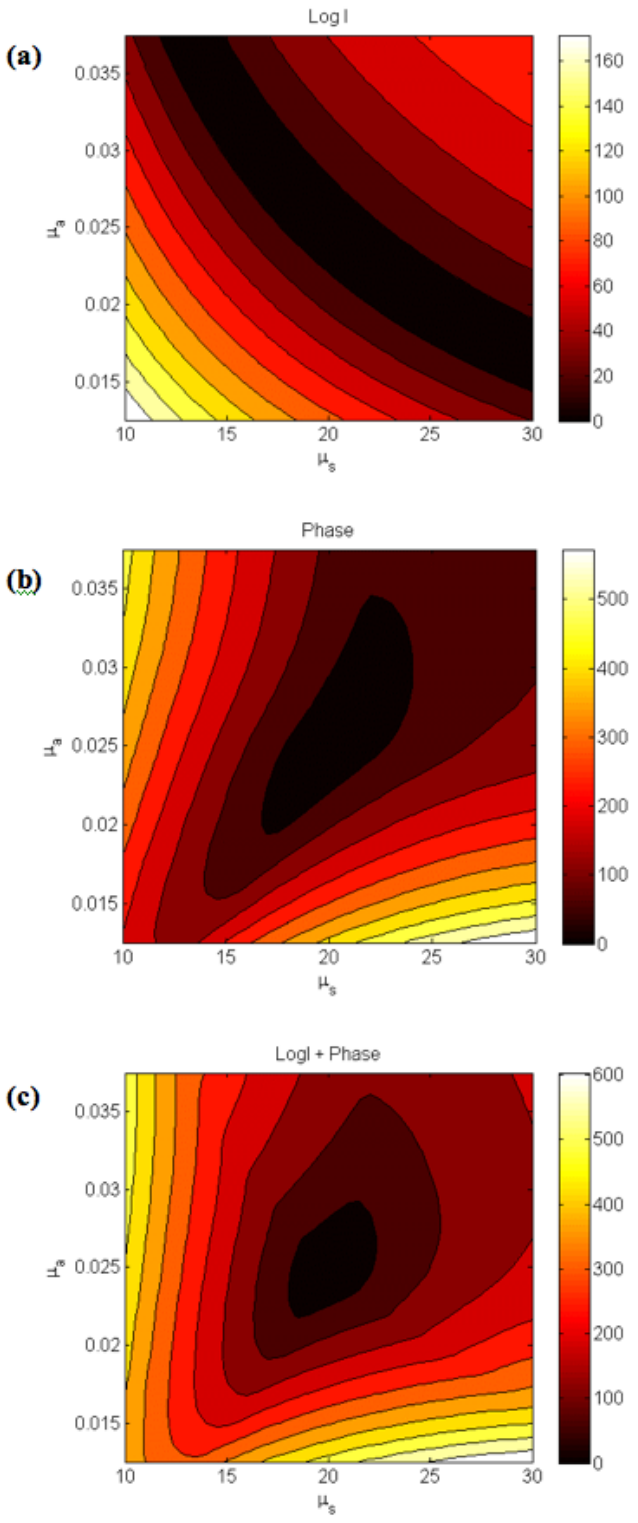


Figure 8.5 Error maps of SP7 data with varying optical properties with arbitrary units of error. (a) map of log(Amplitude) with varying *μ*a (y axis) and *μ*s (x axis), (b) same as (a) but for phase and (c) error map of combined log(Amplitude) and Phase data.

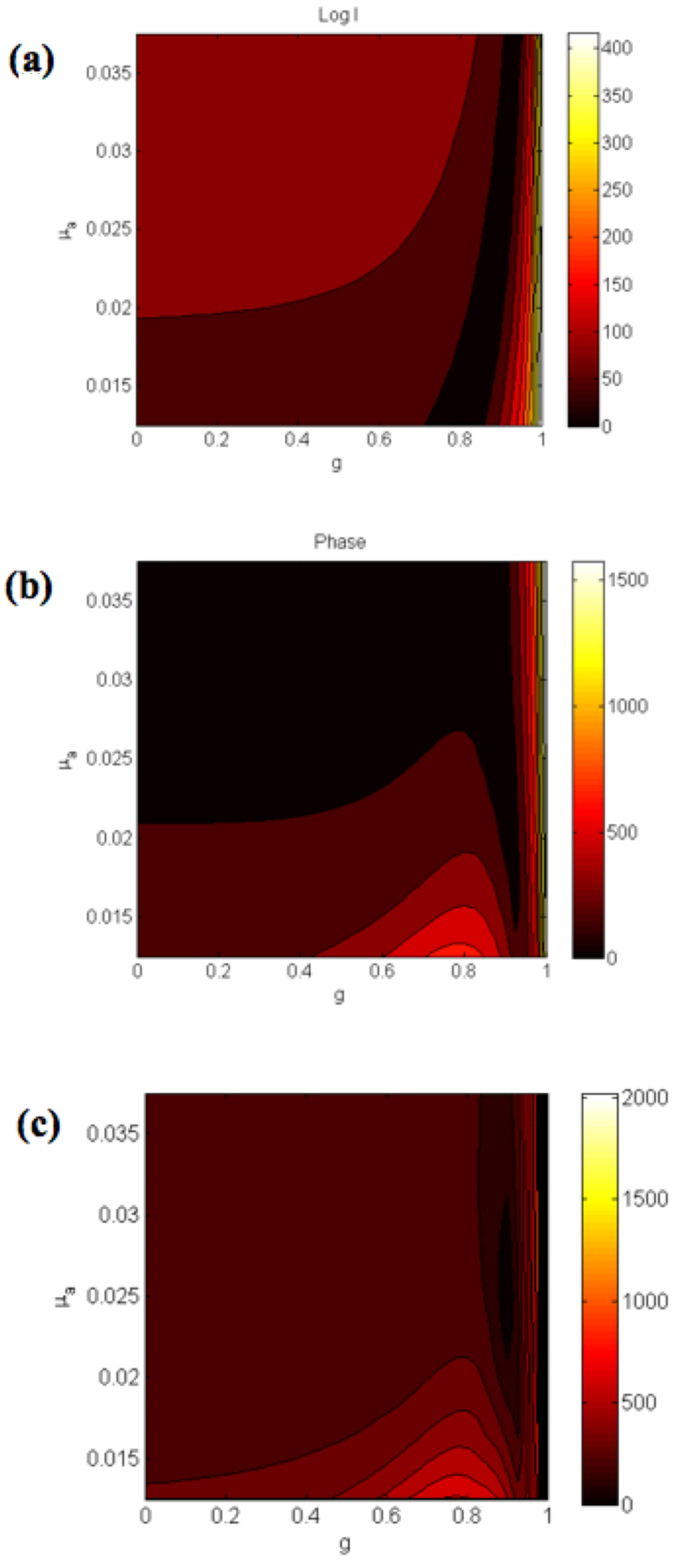


Figure 8.6 Same as Figure 8.5, but for (a) map of log(Amplitude) data with varying g and μa, (b) same as (a) but for phase, (c) sum of (a) and (b)

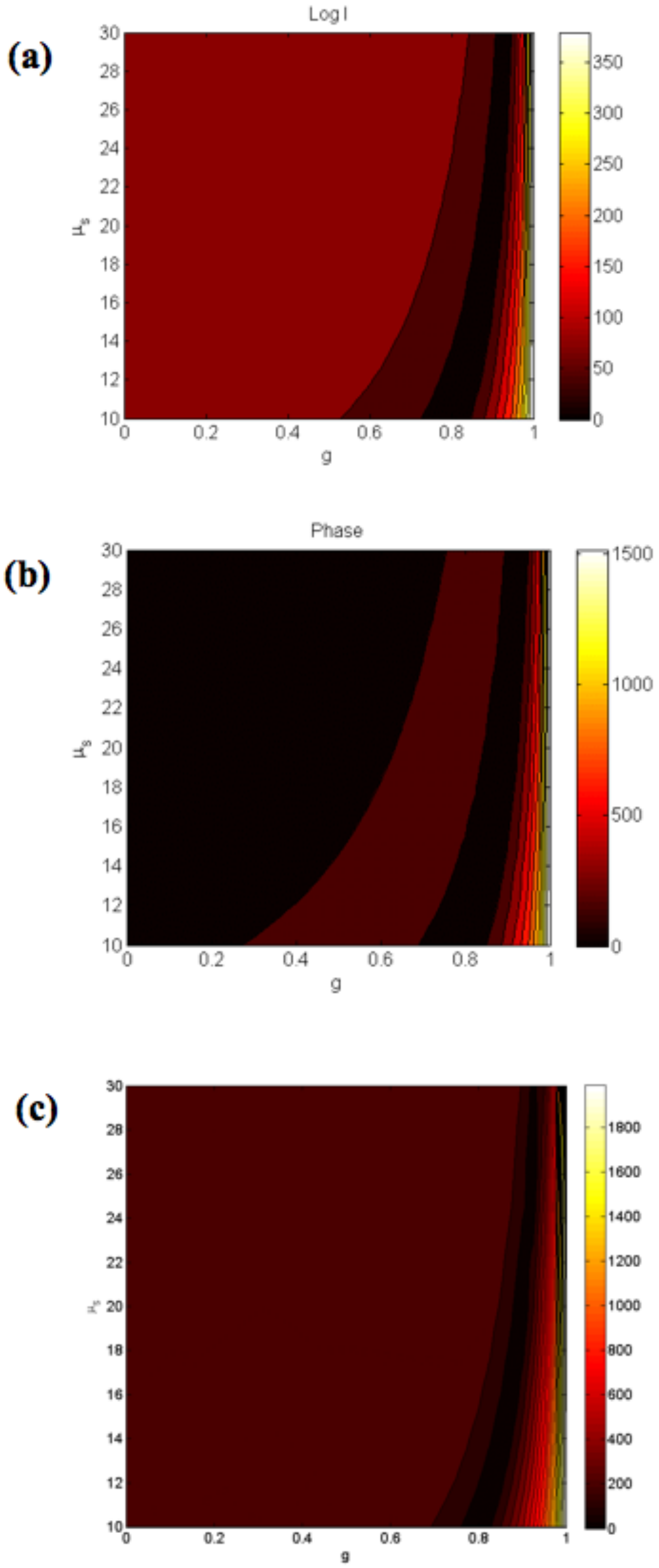


Figure 8.7 : Same as Figure 8.6, but for (a) map of log(Amplitude) data with varying g and *μ*s, (b) same as (a) but for phase, (c) sum of (a) and (b).

8.4 Uniqueness with the introduction of higher order terms

The use of higher ordered SP_N approximations introduces a number of new variables, μ_{an}, which are defined as in equation 7.4. These higher ordered absorption coefficients are dependent on both μ_s and g and as such, may make it possible to separate the effects due to changes in the two parameters, if it is assumed that the composite moments of the boundary data can be measured.

To test the uniqueness of the higher absorption orders, the values of μ_{an} for n=1:4 (as available for SP_5) were calculated using optical properties of μ_a=0.01 mm^{-1}, μ_s = 20 mm^{-1} and g = 0.5 as a reference. The values of μ_{an} were then re-calculated with values of the scattering coefficient ranging from 1mm^{-1} to 40mm^{-1} and the anisotropy factor ranging from 0.1 to 0.9.

Figure 8.8a shows the L1-norm between the reference value of μ_{a1} and those calculated for each combination of scattering coefficient and anisotropy factor. There is a large band of optical property combinations for which the residual falls to zero and therefore result in the same value of μ_{a1}. Figure 8.8b shows the same differences between values of μ_{a2}. There is still a large number of scattering coefficient / anisotropy factor combinations that lead to the same value in this case, although slightly smaller than for μ_{a1}. Figure 8.8c-d show similar plots for μ_{a3} and μ_{a4} respectively. It can be seen that as higher order moments of absorption are considered, the number of scattering coefficient / anisotropy factor combinations leading to the same value reduces, but does not converge to a particular combination. As such, the scattering coefficient and anisotropy factors cannot be simultaneously identified.

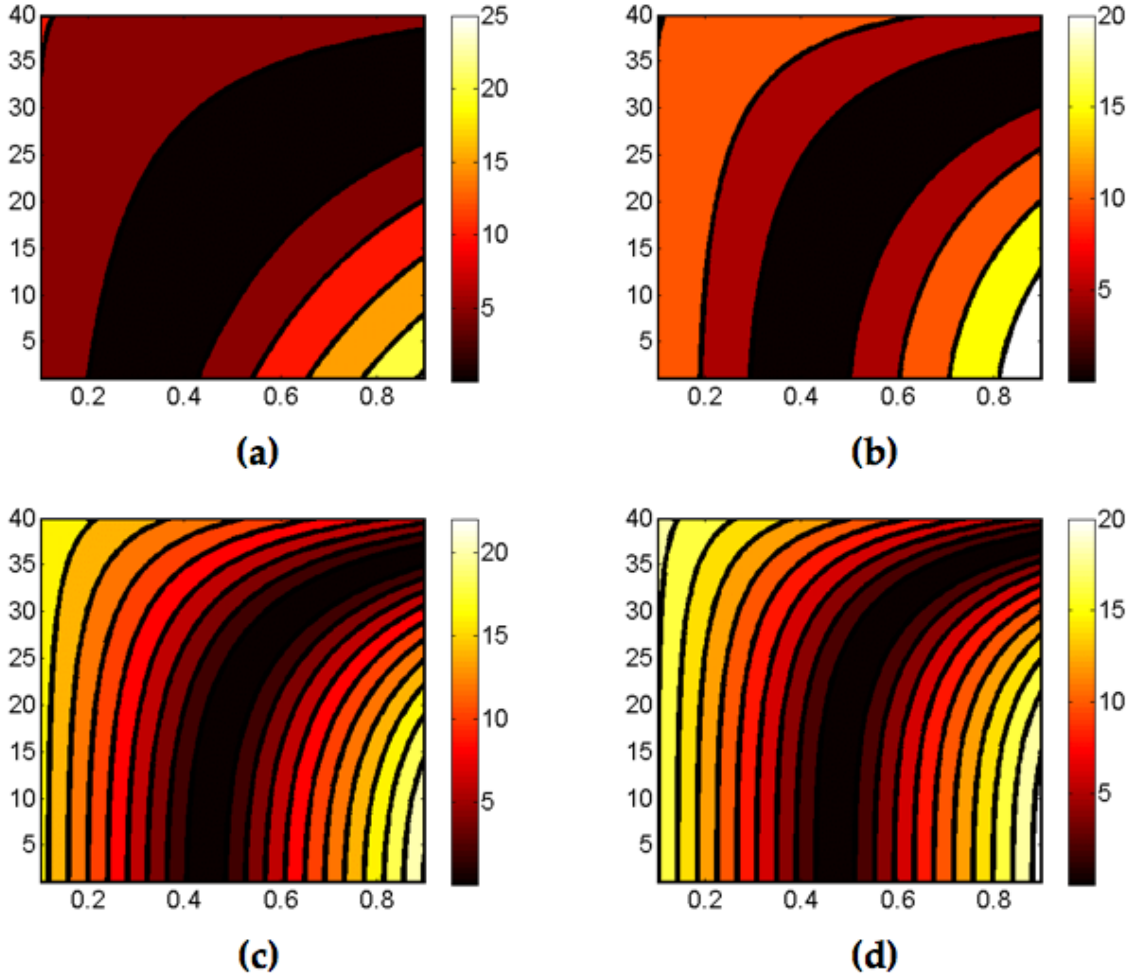


Figure 8.8 Maps of residuals between (a) μa1, (b) μa2, (c) μa3 and (d) μa4 for a range of scattering coefficients and anisotropy factors as compared to a set of reference values. The range of anisotropy factors is listed on the x-axes whilst the y-axes represent the range of scattering coefficients.

8.5 Multi-parameter image reconstruction using the SP$_N$ approximation

Image reconstruction algorithms were developed for N=1, 3, 5 and 7. Depending on the order, N, the SP$_N$ based forward models calculate a number of complex moments of fluence as defined by equations 7.11(a)-(d). The development of the image reconstruction algorithms was based on the assumption that these complex moments of fluence were physically measurable. The boundary data then included phase and amplitude values for each of these moments. When considering all of the moments of fluence, the number of optical properties to be reconstructed also increases to include the various orders of the absorption coefficient, as defined by equation 7.13, and various combinations of these to create diffusion-like terms. The optical properties to be calculated for each order are listed in Table 8.1. In order to reconstruct all of these optical parameters, the inverse problem, $J\mu = \varphi$, then takes the form

$$\begin{bmatrix} J^{\varphi_1}_{\kappa_1} & J^{\varphi_1}_{\mu_{a1}} & 0 & 0 & 0 & 0 & J^{\varphi_2}_{\mu_{a4}} & J^{\varphi_3}_{\mu_{a5}} & 0 \\ 0 & 0 & J^{\varphi_2}_{\kappa_3} & J^{\varphi_2}_{\mu_{a2}} & 0 & 0 & J^{\varphi_1}_{\mu_{a4}} & 0 & J^{\varphi_3}_{\mu_{a6}} \\ 0 & 0 & 0 & 0 & J^{\varphi_3}_{\kappa_5} & J^{\varphi_3}_{\mu_{a3}} & 0 & J^{\varphi_1}_{\mu_{a5}} & J^{\varphi_2}_{\mu_{a6}} \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \mu_{a1} \\ \kappa_3 \\ \mu_{a2} \\ \kappa_5 \\ \mu_{a3} \\ \mu_{a4} \\ \mu_5 \\ \mu_{a6} \end{bmatrix} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \end{bmatrix} \tag{8.2}$$

where φ_n are measurements of the complex moments of fluence, the kernels $J^{\varphi_n}_{\mu}$ are individual Jacobian matrices mapping the sensitivity of measurement φ_n to changes in optical property μ with the form

$$J^{\varphi_1}_{\mu_{a1}} = \begin{bmatrix} \dfrac{\delta \log \varphi_1}{\delta \mu_{a1}} & ; & \dfrac{\delta \theta_1}{\delta \mu_{a1}} \end{bmatrix} \tag{8.3}$$

The terms $\kappa_1 = \frac{1}{3\mu_{a1}}$, $\kappa_3 = \frac{1}{7\mu_{a3}}$ and $\kappa_5 = \frac{1}{11\mu_{a5}}$ are analogous to the diffusion coefficient in the DA. For ease of notation, the terms μ_{an} do not strictly refer to the orders of absorption coefficient defined in equation 7.13, but various combinations and fractions of these orders. The multiple entries per row of the Jacobian account for the inter-dependency of the fluence moments.

For computational efficiency, the individual Jacobian kernels were constructed using the reciprocal method, discussed in Chapter 6, where

$$J_{\kappa_1}^{\varphi_1}(i,j) = \nabla\varphi_{1i} \times \nabla\varphi_{1j} \tag{8.4a}$$

$$J_{\kappa 3}^{\varphi_2}(i,j) = \nabla\varphi_{2i} \times \nabla\varphi_{2j} \tag{8.4b}$$

$$J_{\kappa_5}^{\varphi_3}(i,j) = \nabla\varphi_{3i} \times \nabla\varphi_{3j} \tag{8.4c}$$

$$J_{\mu_{a1}}^{\varphi_1}(i,j) = \varphi_{1i} \times \varphi_{1j} \tag{8.4d}$$

$$J_{\mu_{a2}}^{\varphi_2}(i,j) = \varphi_{2i} \times \varphi_{2j} \tag{8.4e}$$

$$J_{\mu_{a3}}^{\varphi_3}(i,j) = \varphi_{3i} \times \varphi_{3j} \tag{8.4f}$$

The Jacobians were also separately built using the perturbation method as a means of validation. Prior to inversion, the Jacobians were normalised by optical properties (Dehghani et al. 2008).

Test boundary data was generated using the same circular geometry, Figure 8.1, with μ_a = 0.01 mm^{-1}, μ_s = 10 mm^{-1}, g = 0.9 and n = 1.33. A highly absorbing anomaly, with μ_a = 0.02 mm^{-1} and a highly scattering anomaly with μ_s = 20 mm^{-1}, were inserted into the mesh as shown in Figure 8.9. Both anomalies have the same anisotropic factor and refractive index as the background.

Each of the SP_N reconstruction algorithms (for N = 1, 3, 5 and 7) was capable of reconstructing different optical properties, depending on the order N. In order to test the capability of each of the SP_N models, the optical properties listed in Table 8.1 were reconstructed, and from these, values of μ_a and μ_s were extracted. For the inverse model, the same mesh as the forward model was used without added noise, often known as the inverse crime. This was done intentionally to ensure that any errors in the reconstructed optical maps were *only* due to the complexity of the higher ordered approximations. Each of the algorithms were allowed to continue until there was less than a 0.1% change in successive iterations. The initial regularization parameter λ (equation 6.10) for all algorithms was set to 10 times the maximum diagonal of the matrix J^TJ and was reduced by a factor of $10^{0.25}$ at each iteration (Dehghani et al. 2008).

Figure 8.10a shows the reconstructed images generated using an SP_1 based reconstruction algorithm and SP_1 forward data. The location of the two anomalies has been accurately reconstructed. The background optical properties have been accurately recovered although the values of the two anomalies have been over-estimated. The reconstructed image shows a maximum absorption coefficient of 0.0249 mm^{-1} and a maximum scattering coefficient of 23.9 mm^{-1}. The SP_1 reconstruction required 32 iterations at 3.1 second per iteration to recover the two relevant unknowns.

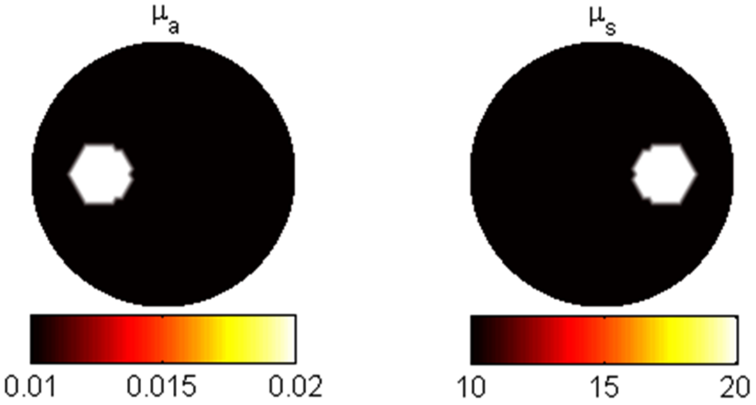


Figure 8.9 FEM model with one highly absorbing target and one highly scattering target.

Table 8.1 Reconstructed values of SP_N image reconstruction algorithms where the diffusion terms $\kappa_n = 1/(A\mu_{an})$ where A is a constant.

Reconstruction model	Forward data used	Unknown parameters
SP_1	SP_1	κ_1, μ_a
SP_3	SP_3	$\kappa_1, \kappa_3, \mu_{a1}, \mu_{a2}$
SP_5	SP_5	$\kappa_1, \kappa_3, \kappa_5, \mu_{a1}, \mu_{a2}, \mu_{a4}$
SP_7	SP_7	$\kappa_1, \kappa_3, \kappa_5, \kappa_7, \mu_{a1}, \mu_{a2}, \mu_{a4}, \mu_{a6}$

The SP_3 reconstruction (using SP_3 data) Figure 8.10b has also accurately recovered the location and shape of the two anomalies. The recovered absorption and scattering coefficients have underestimated the target values, returning 0.0189mm^{-1} and 17.3mm^{-1} respectively. Unlike the SP$_1$ image, the SP$_3$ reconstruction shows signs of cross talk between the absorption and scattering images. The SP$_3$ reconstruction required 13 iterations at 7.3 second per iteration to recover the four relevant unknowns.

Reconstruction using the SP$_5$ model (using SP$_5$ data) is shown in Figure 8.10c. In this case, the optical properties have again been overestimated with a maximum absorption coefficient of 0.0224mm^{-1} and a maximum scattering coefficient of 21.4mm^{-1}. The cross talk between the absorption and scattering images is still present. The SP$_5$ reconstruction required 34 iterations at 54 second per iteration to recover the six relevant unknowns.

The optical maps generated by the SP$_7$ model (using SP$_7$ data), Figure 8.10d contains artifact throughout the domain. The location of the highly absorbing target has been recovered with poor size and contrast accuracy. The reconstruction failed to recover the highly scattering target. It is likely that the failure of the SP$_7$ reconstruction is due to the highly under-determined nature of the problem. The SP$_7$ model requires 8 unique variables to be calculated at each node, i.e. 14280 unknowns, based on just 1920 boundary measurements. The SP$_7$ reconstruction required 20 iterations at 52 second per iteration to recover the eight relevant unknowns.

It should be noted that Figure 8.10(a)-(d) are plotted with varying scales and, as such, the size of the recovered anomolies may appear to vary. If a fixed scale were to be used on all figures, the artifact that appears in the lower orders would be artificially surpressed.

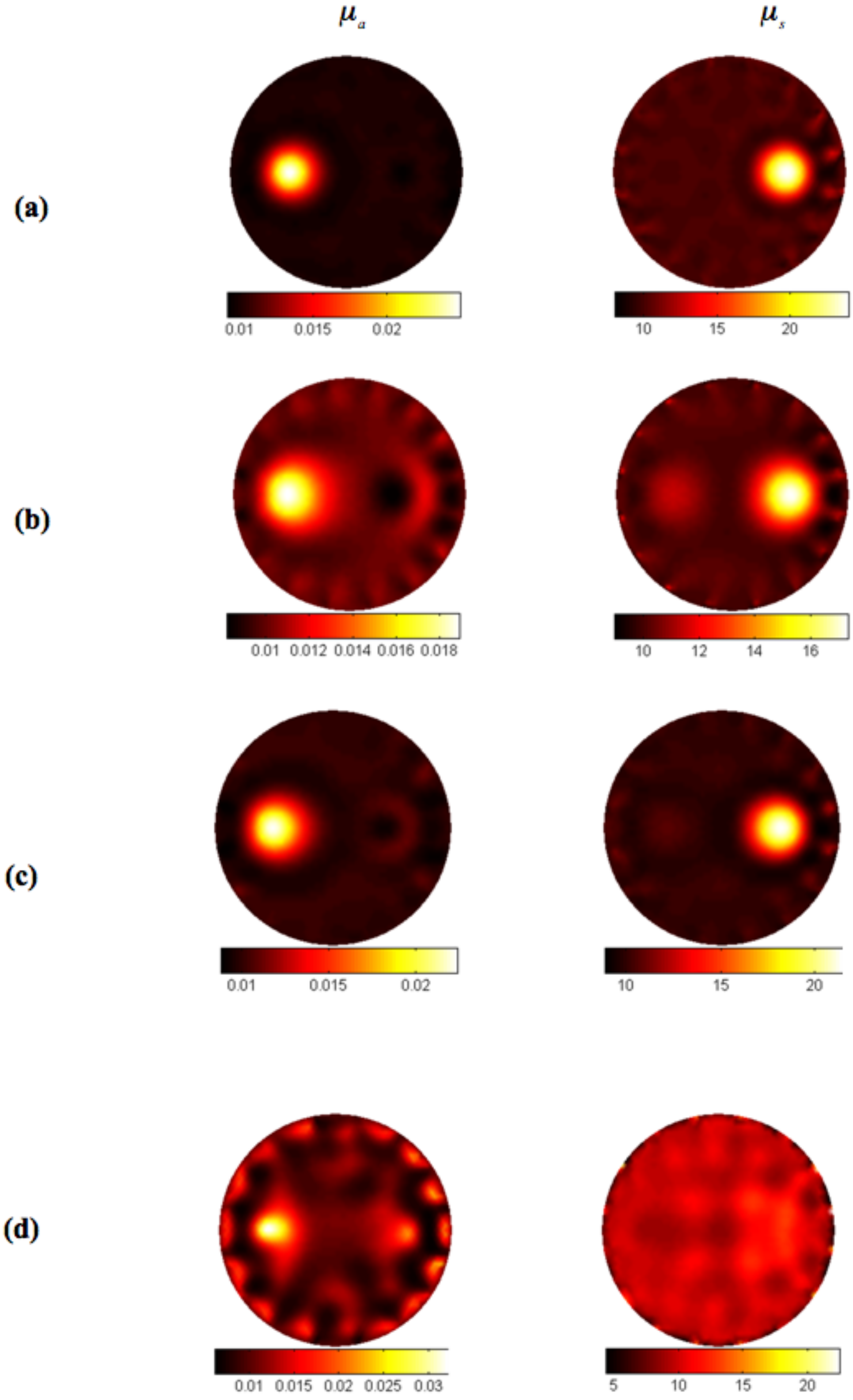


Figure 8.10 Reconstructed optical maps using a) SP1, b) SP3, c) SP5, d) SP7 based reconstruction algorithms

As mentioned above, the data shown in Figure 8.10 was generated by committing the inverse crime. To make the problem less trivial, the reconstructions were repeated using forward data obtained with a mesh containing 20% more nodes than the reconstruction mesh, Figure 8.11. It can be seen that these reconstructed optical property maps contain a greater degree of noise, particularly in the regions close to the boundaries.

To quantitatively analyse the reconstructions, a cross section through the reconstructed optical property maps was taken along the y=0 axis. Figure 8.12 shows the cross section through the μ_a map. The SP_1 approximation overestimates the value of the highly absorbing insertion by approximately 25% with a peak value of 0.025 mm^{-1}. The size of the insertion, however, is underestimated with a FWHM of 13.7mm compared to the target value of 20mm. The SP_3 approximation, however, most closely recovers the insertion with a peak value of 0.019 mm^{-1} and closely recovers the size of the anomaly with a FWHM of 17.4mm. The SP_5 and SP_7 approximations shows a similar level of accuracy with peak values of 0.0185mm^{-1} and 0.0184mm^{-1} respectively. The SP_7 approximation, however, underestimates the size of the insertion at 13.8mm and shows a highly absorbing region outside of the target. There is also a significant level of cross talk where the highly scattering insertion appears in the absorption image.

Figure 8.13 shows the cross section through the scattering image. The SP_1 approximation again overestimates the target value by approximately 20% whereas the SP_3, SP_5, and SP_7 approximations underestimate the target by 13%, 10% and 21% respectively. The higher ordered reconstructions also show signs of cross talk with the highly absorbing insertion.

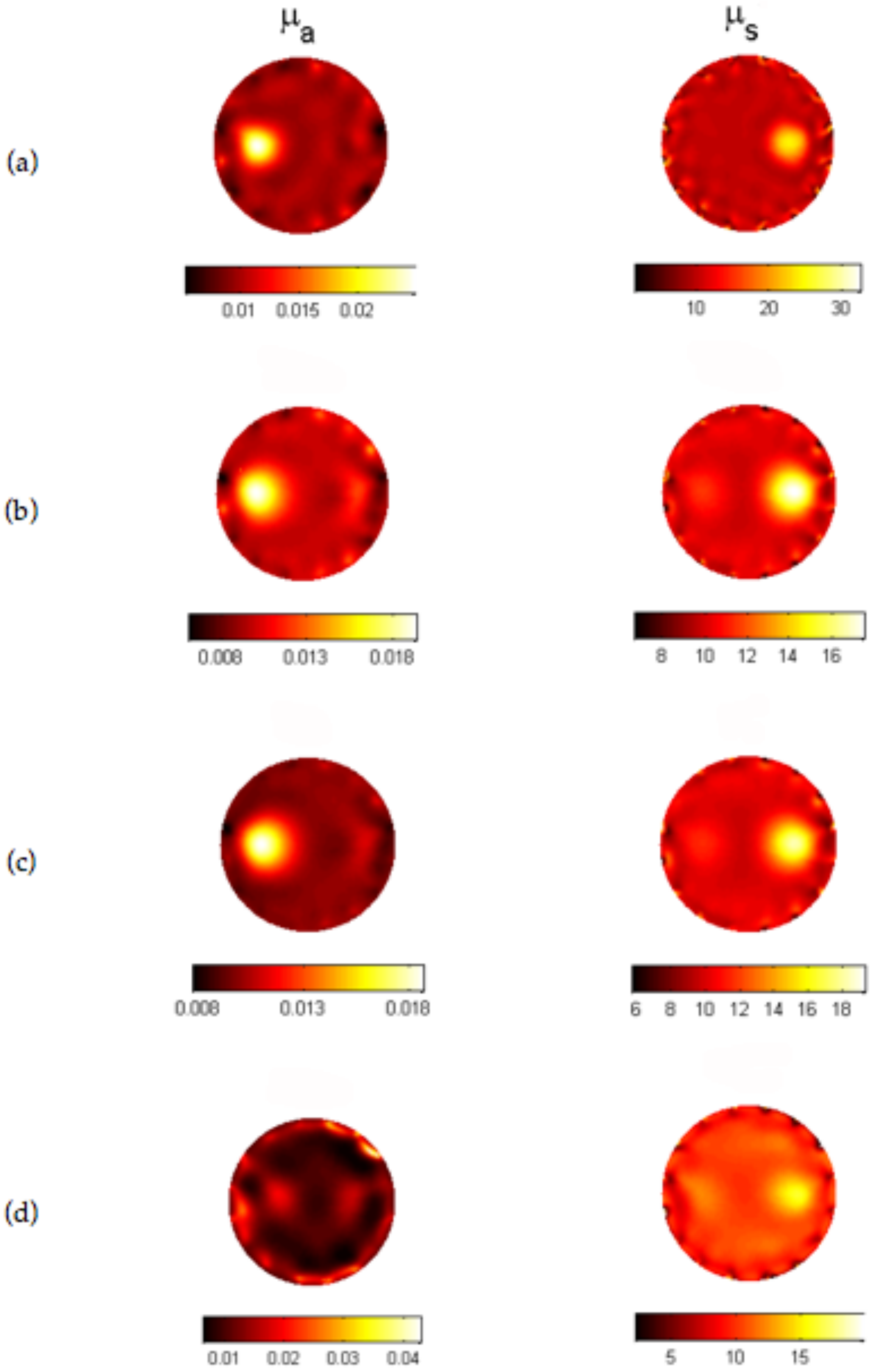


Figure 8.11: As Figure 8.10 with forward data obtained with a mesh containing 20% more nodes than in reconstruction mesh

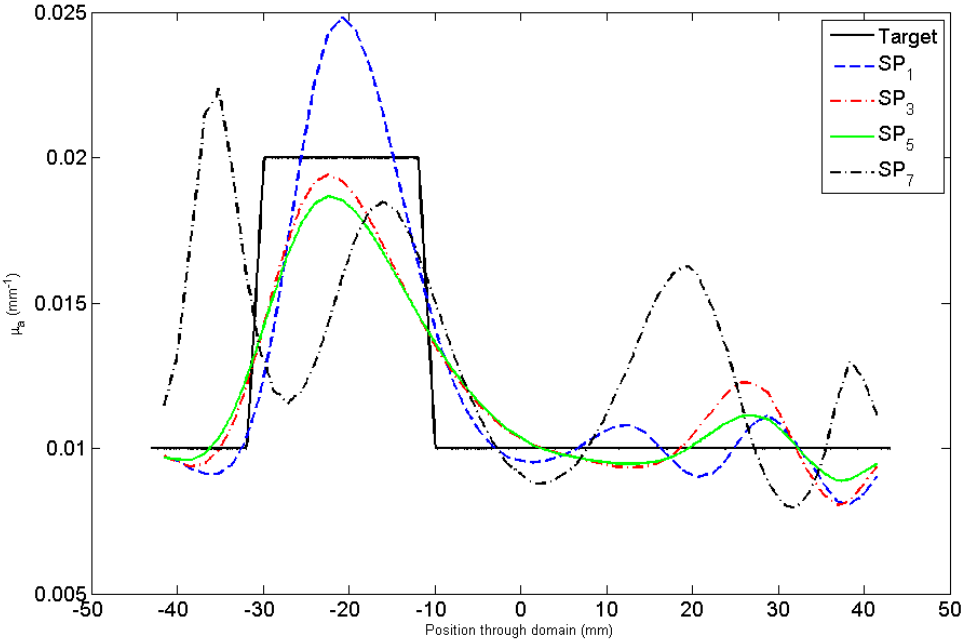


Figure 8.12: Cross section through reconstructed μ_a maps shown in Figure 8.11.

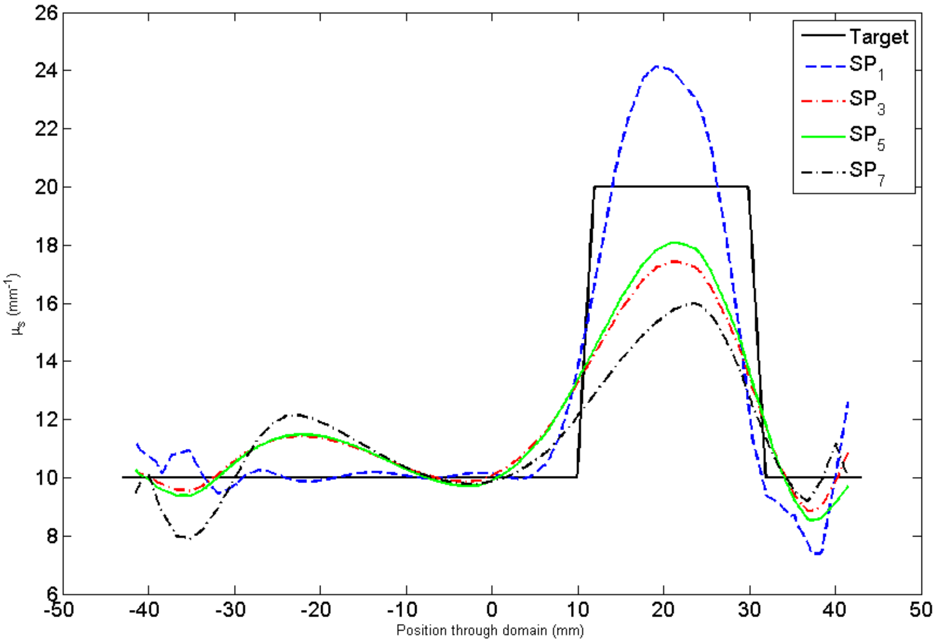


Figure 8.13: As Figure 8.12 for μ_s

The uniformity of the background regions in the absorption and scattering images was calculated as

$$\text{Uniformity} = 100\% \times \frac{(\text{max value - min value})}{(\text{max value + min value})}$$

and is listed in Table 8.2. The SP_5 approximation results in the most uniform background in the absorption image, indicating good suppression of noise, whereas the SP_3 approximation results in the best uniformity in the scattering image. Due to the underdetermined nature of the SP_7 problem, the resulting image has poor uniformity in both the absorption and scattering images.

Table 8.2: The uniformity of the background regions in the reconstructed optical property maps

	Uniformity (%)	
	μ_a	μ_s
SP_1	16	18
SP_3	20	12
SP_5	11	14
SP_7	34	20

8.6 Multi-parameter image reconstruction and hard priori information

The previous study was repeated using region-based reconstruction for the SP_N methods with N=1, 3 and 5. The domain was divided into 3 regions, one for each of the anomalies shown in Figure 8.9 and a third which encompassed the remainder of the domain. This information was then included in the reconstruction using the hard priors method discussed in Chapter 6. The SP_7 model has been omitted from further studies as the increase in accuracy over the SP_5 model has been shown to be very small and neglectable (Klose and Larsen 2006).

It can be seen (Figure 8.14) that the use of prior information, whereby instead of reconstruction the unknown parameters at each spatial variable, homogeneous values are estimated for each unknown region (in this case 3) has enabled the SP_5 reconstruction (when using SP_5 forward data) to accurately recover both targets. This is due to the vast reduction in the number of unknowns from 6 times the number of nodes (10710) to 6 times the number of regions (18). The same findings (not shown) were found for SP_1 and SP_3 cases.

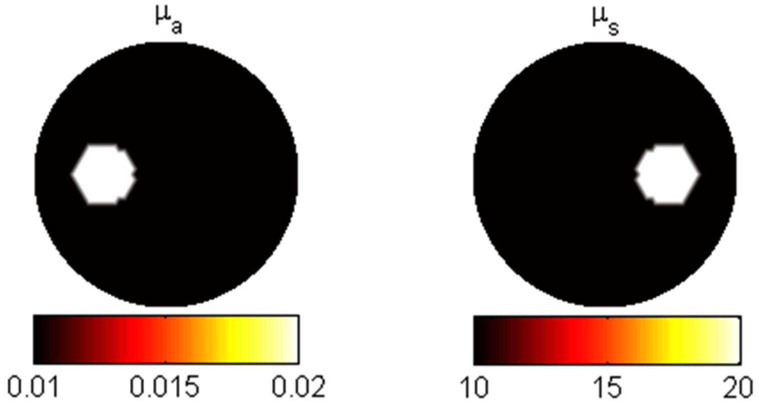


Figure 8.14 Recovered optical map generated using SP5 reconstruction with prior information.

8.7 Diffusion based image reconstruction using the SP_N approximation

As discussed earlier, the SP_N approximations, equations 7.12(a)-(d), are based on composite moments of fluence with the total flux calculated using equation 7.14. The SP_N reconstruction algorithms introduced so far were therefore based on various moments of the phase and amplitude boundary data which attempted to recover all moments of absorption and scattering coefficients. In practice, however, there are no experimental methods to easily measure the angular components of amplitude and phase and so image reconstructions must be based on total values of fluence.

A new reconstruction algorithm was developed that used only the absolute fluence calculated using the SP$_5$ approximation. The complex moments were first calculated using the SP$_5$ algorithm and then summed according to equation 7.14 to find the total fluence. The boundary data therefore consisted of a single set of amplitude and phase measurements for each source-detector pair. In eliminating the composite moments of fluence, however, it is only possible to reconstruct for μ_a. and μ_s (assuming g = 0.9). The inverse problem then took the form

$$\begin{bmatrix} J_{\mu_s} & J_{\mu_a} \end{bmatrix} \begin{bmatrix} \mu_s \\ \mu_a \end{bmatrix} = \phi \tag{8.5}$$

where the individual kernels of the Jacobian are calculated using the reciprocal method where

$$J_{\mu_a}(i,j) = \Phi_i \times \Phi_j^{Adj} \tag{8.6}$$

$$J_{\mu_s}(i,j) = \nabla\Phi_i \times \nabla\Phi_j^{Adj} \tag{8.7}$$

and have the form

$$J_{\mu_s} = \left[\frac{\delta \log\phi}{\delta\mu_s} \quad ; \quad \frac{\delta\theta}{\delta\mu_s} \right] \tag{8.8}$$

This new reconstruction algorithm is tested using SP$_5$ forward data and the resulting optical map is shown in Figure 8.15a. The reconstruction has performed well recovering the optical parameters within 10% of the target values. For comparison, the same forward data is also used to reconstruct optical parameters using the DA based algorithm, Figure 8.15b. In this case, the target values have been over-estimated and boundary artifact has been introduced, indicating the maximum errors seen from high order model mismatch, is as expected near the source / detector positions, which lead to image inaccuracy and artefacts.

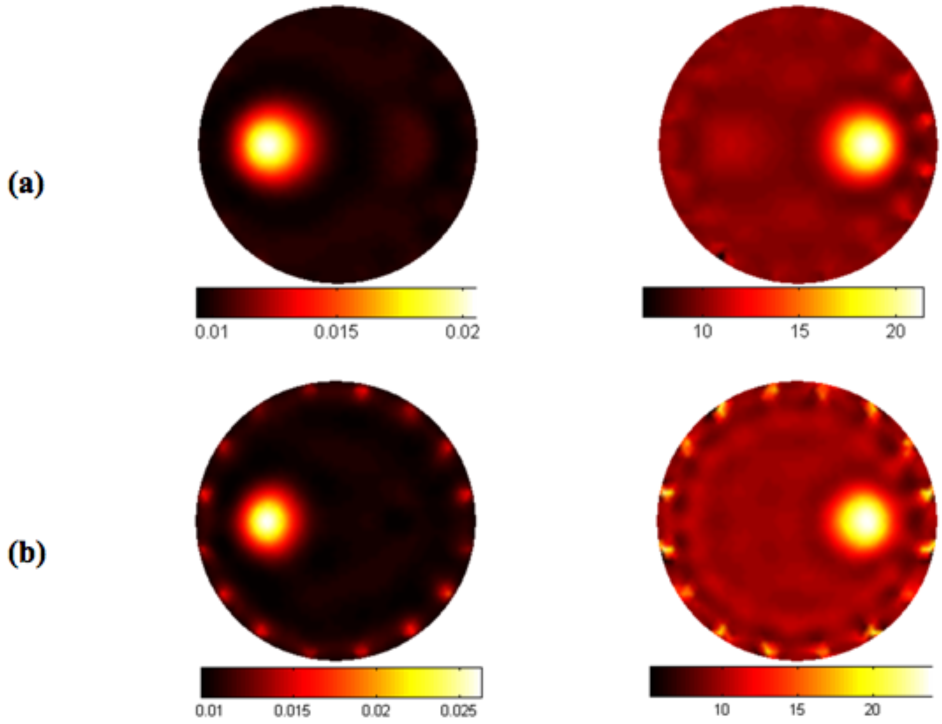


Figure 8.15 Diffusion based images from SP5 data using (a) SP5 based Jacobian and (b) SP1 based Jacobian.

8.8 Discussions and Conclusions

Forward models for image reconstruction based on the simplified spherical harmonic approximation have been presented. Sensitivity maps for amplitude and phase data are calculated, Figure 8.2 and Figure 8.3, using SP_7 and it is shown that the sensitivity of measured boundary data for changes in the anisotropy factor are many orders of magnitude lower than those of absorption and scattering coefficients. It is also shown that the sensitivity to both scattering coefficient anisotropy factor is highly localized to the regions adjacent to the source and detector (Figure 8.2 and Figure 8.3). This suggests that beyond a few scattering lengths, the light distribution is fully diffuse and as such, distant changes in scattering and anisotropy factors have little effect on measured data. Although it is possible to normalize the sensitivity maps prior to use in image reconstruction (Eames and Dehghani 2008), it is also evident that an increase in scattering parameter will have an effect on boundary data which is opposite of that due to anisotropy factor. This in turn indicates that an increase in one parameter can be compensated by a decrease on the other and vice-versa. This effect is unlike that seen for absorption and scattering, whereby the two parameters can be separated using log amplitude and phase data, but it is unclear whether all 3 parameters can be resolved using these 2 data-types.

Through the use of error maps, whereby the difference in some reference data with respect to data in the presence of an anomaly is calculated, it is shown that using frequency domain data, it is possible to distinguish, with some accuracy, the absorption and scattering properties of the medium, Figure 8.5, which has also been presented previously (Arridge 1999). The same principles have been applied to higher order models and additionally the effect of the anisotropy factor has been studied as well as the optical absorption and scatter. It is demonstrated, Figure 8.6 and Figure 8.7, that using frequency domain data, a large and non-unique range of all three optical properties can lead to the same measured data, indicating that it is not possible to reconstruct all three parameters using only 2 data-types. This finding is in line with earlier results, Figure 8.3, and demonstrate that it is difficult to separate for both scattering and anisotropic factors simultaneously using frequency domain data. Further work is needed to investigate whether other data-types, for example using time resolved data, and / or spectral data, would allow separation of these variables.

Residual maps for different orders of absorption coefficient (μ_{an}) as available using SP_N models have been calculated to investigate the possibility of resolving scattering and anisotropy factors, if μ_{an} coefficients can be measured and calculated. It is shown, Figure 8.8, that even if all higher moments of the absorption coefficients can be calculated, there still exists a large range of scattering and anisotropy factors that would lead to the same μ_{an} values, thus indicating that these cannot be easily resolved.

Reconstruction algorithms based on the SP_N approximation have been developed and tested. For image reconstruction, the same FEM model as for the forward data was used and no noise was added. The inverse crime has been strictly committed, since we are concerned with the accuracy of each model in determining the unknown parameters associated with each model. Additionally, noise free data have only used to only highlight the error seen due to model mismatch, rather than the effects of noise within the data, which can show similar trends. It is important to note that the asymptotic expansion of the transport equation used to verify the SP_N equations (Larsen 1993) considers an optically thick system in which scatter dominates. As such, the resulting SP_N equations are still of a diffuse nature (Josef 1996), although the higher orders provide corrections to the DA in partially non-diffuse problems. As such, the SPn approximation has not been tested for fully non-diffuse problems.

It was shown, Figure 8.10a-c, that for N=1, 3 and 5 the reconstructions performed well, with the reconstructed values being within 24% of expected values with the worse results being obtained from SP_1 and absorption coefficient. This accuracy could be further improved by the optimization of the regularization parameter and stopping criteria. The N=7 model, however, failed to accurately recover the target values, Figure 8.10d. The absorption coefficient, although located, is extremely over estimated, with some cross-talk from the scattering value. The reconstructed scatter image contains high valued artifacts and can be considered inaccurate. By limiting the presented results to these larger domain, it is also worth noting that the majority of image artifacts are seen at the boundary, near the source locations, where SP_1 solution is known the be less accurate. It is therefore expected that the errors seen due to the lack of higher order approximations to be substantially more significant in small geometry imaging experiments. The poor performance of the SP_7 model was most likely due to the under-determined nature of the problem. As stated earlier, the SP_7 model contains 8 unique unknown variables which need to be calculated at each spatial location (FEM node), i.e. 14280 unknowns, based on just 1920 boundary measurements (240 log amplitude and 240 Phase, based on 16 co-located source and detectors and 4 composite moments of fluence).

The reconstructions described above were also repeated using forward data obtained using a FEM mesh with a 20% higher resolution, Figure 8.11.(a)-(d). It was seen that the optical properties of the highly absorbing and scattering inserts being recovered to within 25% of the expected values. The background regions of these reconstructions, however, showed a greater level of noise, particularly near the boundaries. It was found that the SP_5 approximation provided the best combination of accurate recovery of target values while surpressing background noise.

In order to eliminate the large degree of freedoms, the accuracy of the reconstruction algorithms was further improved by creating a better determined problem. This was achieved for the SP_1, SP_3 and SP_5

approximations with the use of prior information. The prior information was used to identify a number of homogeneous regions, reducing the number of properties to be recovered. This simplification of the problem led to improvements in all three of the reconstructions (for N = 1, 3 and 5). Both the absorbing and scattering targets were accurately recovered by all models to within 2%, Figure 8.14.

The SP_N equations are based on composite moments of fluence, equation 7.11(a)-(d). In reality, however, experimental systems can only measure the total fluence at the boundary of the domain and so the full SP_N reconstruction algorithms are of limited use in image reconstruction where all composite moments are needed. An alternative method has been proposed in which the forward models are based on absolute fluence but the Jacobian matrix was calculated using the SP_5 model to allow the calculation of absorption (μ_a) and scattering (μ_s) coefficient only. The diffusion parameter based images reconstructed from simulated SP_5 data whereby the Jacobian is based on either SP_1 or SP_5, Figure 8.15, indicate that although both models can be used, the higher order model is more accurate both in terms of quantitative and qualitative analysis. The target values of the test problem calculated using SP_5 are recovered with more accuracy as compared to the SP_1 based model with much less artifact. Additionally, eliminating the use of complex moments also results in decreased computation time and memory requirements.

The results presented in this work indicate that for image reconstruction whereby the DA is less valid, in for example, small animal imaging and / or where the absorption coefficient is more dominant, the higher order models based on simplified spherical harmonics can be used to generate the sensitivity matrix for diffusion based image reconstruction, without the additional computation complexity in terms of the number of unknown parameters. The incorporation of these more accurate models can however allow for a better accuracy in terms of light propagation models.

Chapter 9 - Development of a SP5 – Diffusion Hybrid Model

9.1 Introduction

In Chapter 7, the SP_N approximation was shown to provide an accurate model of light propagation in tissue. The higher ordered models were shown to have improved accuracy over the DA, particularly in regions in close proximity to the source. Although the SP_N approximation leads to a lower computational load than either the S_N or P_N approximations, the number of unknowns to be solved is still several times larger than for the DA. As the light distribution within a tissue becomes effectively diffuse within a few scattering distances, it is possible to minimise the computational load by using a higher-ordered approximation to model light propagation in the regions surrounding the source and the diffusion approximation elsewhere.

In this chapter, some of the existing hybrid models will be discussed and the development of a novel SP_5-SP_1 hybrid model will be introduced.

9.1.1 Existing Hybrid Models

Tarvainen et al, developed a hybrid model that solved the RTE in the regions surrounding the source and the DA elsewhere (Tarvainen et al. 2005). Their approach was to divide a domain Ω into two sub-domains Ω_{rte}, in which the RTE is used, and Ω_{da} in which the DA is used, Figure 9.1. The RTE is then solved for the domain Ω_{be} that is larger than Ω_{rte} and is marked by a dashed line in Figure 9.1. The radiance at the interface between Ω_{rte} and Ω_{da} as calculated by the RTE is then used as a Dirichlet boundary condition, effectively a distributed source, for the DA forward model in Ω_{da}.

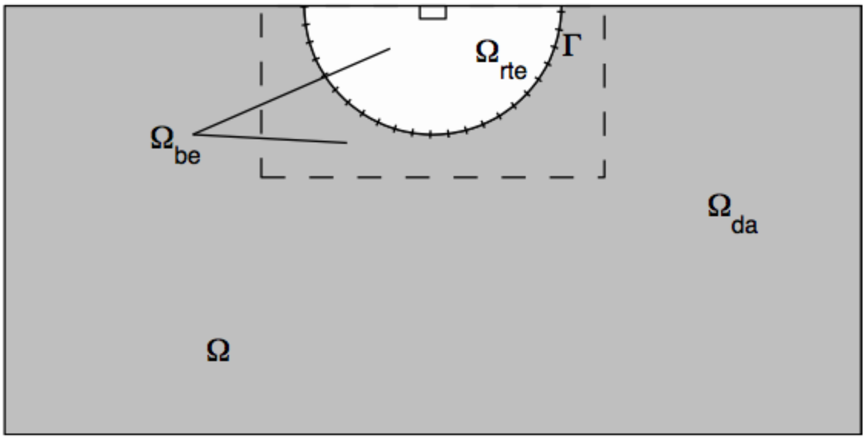


Figure 9.1 Geometry used for the hybrid model by Tarveinen et al (2005). The domain is divided into a number of sub-domains, one in which the RTE is used, and one in which the DA is used.

Other hybrid models have used Monte Carlo methods to model the regions near the source (Wang and Jacques 1993; Alexandrakis et al. 2000). These approaches generally involve determining a critical depth *z*, below which, the DA is assumed to be accurate. At depths above *z*, a Monte Carlo model is used to record the track of sufficient numbers of photons to provide good statistics. Any photons that cross the critical depth are treated as isotropic sources, which are then applied to the DA calculations. These methods have been shown to give good agreement with full MC models but with solution times which are several hundred times faster.

Hybrid models are also of interest for applications involving imaging of the brain. The accuracy of the diffusion approximation is known to be limited in regions of low scattering. Whilst the brain tissue itself may meet the requirements for the DA, it is surrounded by a layer of clear cerebrospinal fluid (CSF) in which the DA is not accurate. In these non-scattering regions, light transport can be modelled using radiosity theory (Cohen and Wallace

1993). The radiosity theory calculates the irradiance on a surface due to a source point at a given angle and distance. A radiosity-diffusion hybrid model has been developed by Arridge et al, in which the DA is used to model light transport in all scattering regions, with radiosity theory used in non-scattering regions which has been shown to give good agreement with both Monte Carlo and experimental results (Firbank et al. 1996; Arridge et al. 2000b). The radiosity-diffusion hybrid has also been applied to image reconstruction problems and has been shown to provide accurate optical maps for non-scattering layers of ~1mm or less (Dehghani et al. 2000).

The radiosity approach requires the CSF to be modelled as a non-scattering region. In reality, however, the CSF layer contains fine arachnoid trabeculae which is likely to cause scattering (Okada and Delpy 2003b; Okada and Delpy 2003a). By modelling the fine anatomy of the brain using a MC model, Okada and Delpy (Okada and Delpy 2003b) found that the reduced scattering coefficient of the CSF layers actually lies between 0.16-0.32 mm^{-1}. This result suggests that neither the radiosity model or the diffusion approximation are valid in the CSF region. Instead, Hayashi et al (Hayashi et al. 2003) have developed a Monte Carlo – diffusion hybrid for non-scattering regions. The use of the MC method allows the CSF layer to be modelled as a region with low scattering, rather than the assumption of zero scattering. As the MC method is only used for the thin CSF layer, it is several times faster than a full MC model whilst maintaining much of the accuracy.

The SP$_N$ based forward models presented in chapter 3 are capable of overcoming the limitations of the DA. These models use the higher ordered approximation throughout the entire domain, leading to an increase in the number of variables at each node of the FEM mesh. The corresponding increase in computational load can be prohibitively large, especially at higher mesh resolutions. An alternative approach is to use a higher ordered model in the regions surrounding the sources and detectors and the diffusion approximation elsewhere. This allows the light distribution to be modelled more accurately whilst minimising the number of computations required.

9.2 Hybrid SP_5 - SP_1 model

The existing hybrid models, discussed above, require two independent steps. The higher accuracy model is first solved in the regions surrounding the source. The results of this initial model are then used as source terms or boundary conditions in the solution of the DA in the remaining regions.

The SP_N models introduced in chapter 7 are solved using the finite element method. If the SP_5 approximation is used in the regions close to the source with the rest of the domain being modelled using the SP_1 approximation, the two regions will have a corresponding and related mass matrix. By coupling these two mass matrices, it is possible to create a hybrid model that can be solved in a single step. Such a method has been developed for problems involving discontinuities in refractive index (Dehghani et al. 2003a).

9.2.1 Meshing

The SP_5-SP_1 hybrid model requires the test geometry to be split into two separate meshes, one in which the SP_5 model will be used and another in which the SP_1 model will be used. The hybrid model was initially tested on a 20×10mm slab geometry with a single source and detector, Figure 9.2(a) which was divided into two separate FEM meshes. The two meshes met at a boundary $\partial\Omega_{SP5}/\Omega_{SP_1}$. The first half of the geometry was solved using the SP_5

approximation whilst the remainder was solved using the SP$_1$ approximation. The nodes lying on $\partial\Omega_{SP5}/\Omega_{SP_1}$ were duplicated as shown in Figure 9.2(b). This was necessary in order to couple the SP$_5$ and SP$_1$ fields.

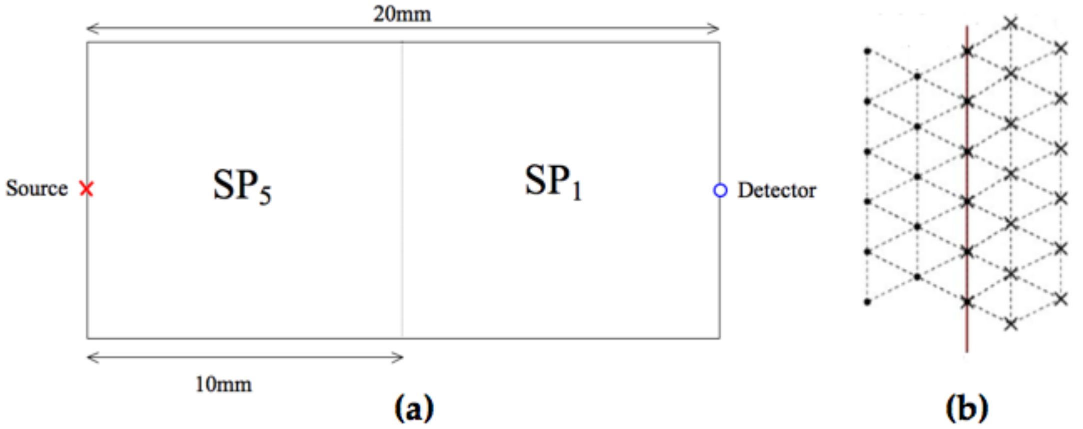


Figure 9.2 (a) 10×20mm slab geometry with a single source and detector used to test the hybrid model with a dashed line marking the boundary between the SP$_5$ and SP$_1$ regions (b) a close up of the nodes lying on the boundary between the SP5 and SP1 regions were duplicated

In this simple slab geometry, the area surrounding the source was solved using the SP5 model whilst the region near the detector was solved using the SP1 model.

9.2.2 The Mass Matrix

The mass matrix of the SP$_1$ model has dimensions of NN×NN whereas the mass matrix of the SP$_5$ model is (3NN)×(3NN). In order to allow the solution of the hybrid model in a single step, the two mass matrices relating to each region were combined into one matrix as shown in Figure 9.3.

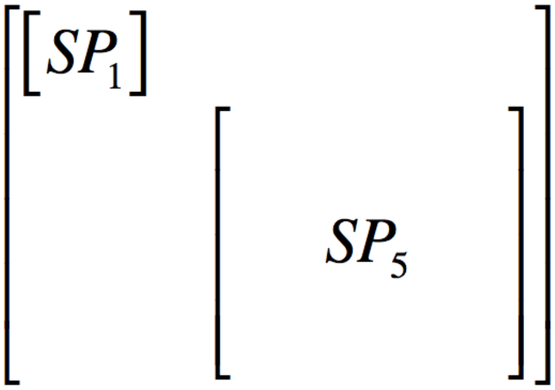


Figure 9.3 The layout of the SP5-SP1 hybrid mass matrix

The fluence across the boundary between the SP$_5$ and SP$_1$ regions should be continuous. In other words

$$\phi_{SP_1} = \phi_{SP_5} \tag{9.1}$$

This condition was enforced by manipulating the hybrid mass matrix. The nodes lying on $\partial\Omega_{SP5}\Omega_{SP_1}$ contained duplicate entries within the hybrid mass matrix, one row within the SP$_1$ segment and 3 rows within the SP$_5$ segment (referring to the compound moments described in equations 7.11(a)-(c)). In order to establish a relationship between these nodes, the contents of the SP$_5$ entries were moved to the SP$_1$ row and the SP$_5$ entries were then set to zero in order to avoid duplicating the contribution of the boundary nodes.

In setting the entire row to zero, however, the relationship of the co-located node with itself was also removed. In order to reinstate this relationship, the element located on the diagonal was set to 1 which, for node *i,* defined $\varphi_1^i = \varphi_1^i$, $\varphi_2^i = \varphi_2^i$ and $\varphi_3^i = \varphi_3^i$. For each co-located node, there was also a relationship between the various other orders. These relationships were determined by re-arranging equation 7.14 to find the relationship between each moment of fluence with the remaining moments. These relationships were applied in the form of a scaling variable to the values of the mass matrix elements that linked the orders of fluence for node *i.* The correct values were found when the fluence, determined by running the forward model, at the co-located nodes was found to be identical. By re-arranging equation 7.14, the relationships are found to be

$$\varphi_1 = -\phi + \frac{2}{3}\varphi_2 - \frac{8}{15}\varphi_3 \tag{7.2}$$

$$\varphi_2 = \frac{3}{2}\phi - \frac{3}{2}\varphi_1 - \frac{4}{5}\varphi_3 \tag{7.3}$$

$$\varphi_3 = -\frac{15}{8}\phi + \frac{15}{8}\varphi_1 + \frac{5}{4}\varphi_2 \tag{7.4}$$

Using the basic mesh in Figure 9.4, for example, the nodes 4,5, and 6 are co-located with nodes 7, 8 and 9. The resulting hybrid mass matrix has the form shown in Figure 9.5, which labels each row with its corresponding node number. The entries for node 3 and node 5, for example, refer to the same point in space and so the fields should be coupled. The matrix entries referring to node 5 are moved to row 3 of the mass matrix and then each of the rows referring to node 5 are set to zero. The diagonal elements referring to node 5 are then set to 1 to re-establish its relationship with itself. The relationship with the other orders of fluence were then established. To enforce equation 7.2 for node 5, for example, the element linking φ_1 and ϕ in row 5 was set to -1, the element linking φ_1 and φ_2 was set to $\frac{2}{3}$ and the element linking φ_1 and φ_3 was set to $-\frac{8}{15}$.

In order to run a hybrid problem, the two individual meshes, along with a list of the duplicated nodes is input into the hybrid model. The model then identifies each of the rows relating to a duplicated node and performs the coupling process described above. The two individual meshes are combined into one single slab mesh. The fluence at all points is then calculated and mapped onto this single mesh.

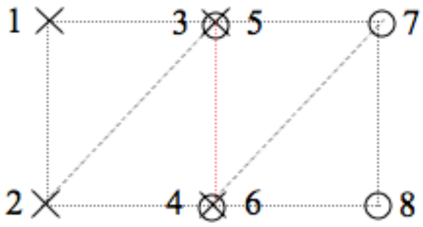


Figure 9.4 A highly simplified example of a hybrid mesh containing 2 pairs of co-located nodes.

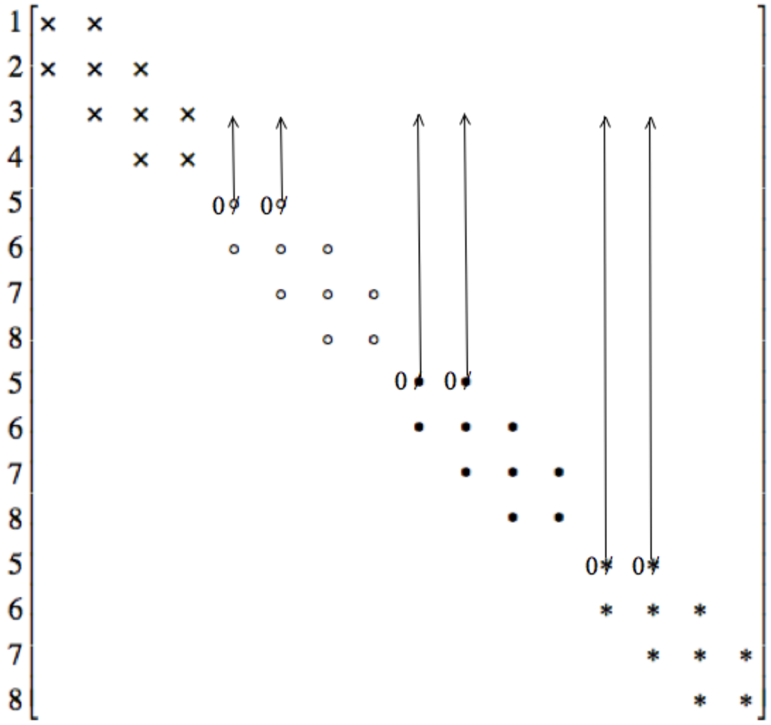


Figure 9.5 Hybrid mass matrix. Numbers on left identify the node from figure 9.3 relating to that row. SP5 and SP1 fields are coupled by moving SP5 entries to SP1 row and setting the original rows to zero. The ∘, • and ∗ entries relate to the φ_1, φ_2 and $\varphi_3$moments of fluence as defined by equations 7.11(a)-(d)

9.3 Results

The SP$_5$ hybrid model was used on the mesh used in figure 9.1(a) for a range of optical property combinations. An equivalent single mesh was constructed and used to obtain full SP$_5$ and SP$_1$ solutions. A cross-section of the field was taken along the line linking the source and detector.

Figure 9.6 shows the cross sectional fluence obtained for optical properties of $\mu_a = 0.01mm^{-1}$, $\mu_s = 10mm^{-1}$, $g = 0.9$, and $n = 1.33$. It can be seen that the hybrid data is almost identical to the full SP$_5$, diverging only at the last few millimetres. The SP$_1$ data, however, is consistently lower than the SP$_5$ data. The percentage error in the hybrid and SP$_1$ results, in reference to the SP$_5$ data is shown in figure 1b. It can be seen that the SP$_1$ data has an error of over 70% near the source and approximately 15% near the detector. The hybrid data, however, has almost 0% error through the majority of the domain, although it increases to around 10% within 1mm of the detector.

The same geometry was then modelled with increased scattering coefficients of $\mu_s = 15$, Figure 9.7, and $\mu_s = 20$, Figure 9.8. The results for the hybrid model were similar to those seen for the initial case. The error in the SP$_1$ data, however, fell more rapidly than in the previous case.

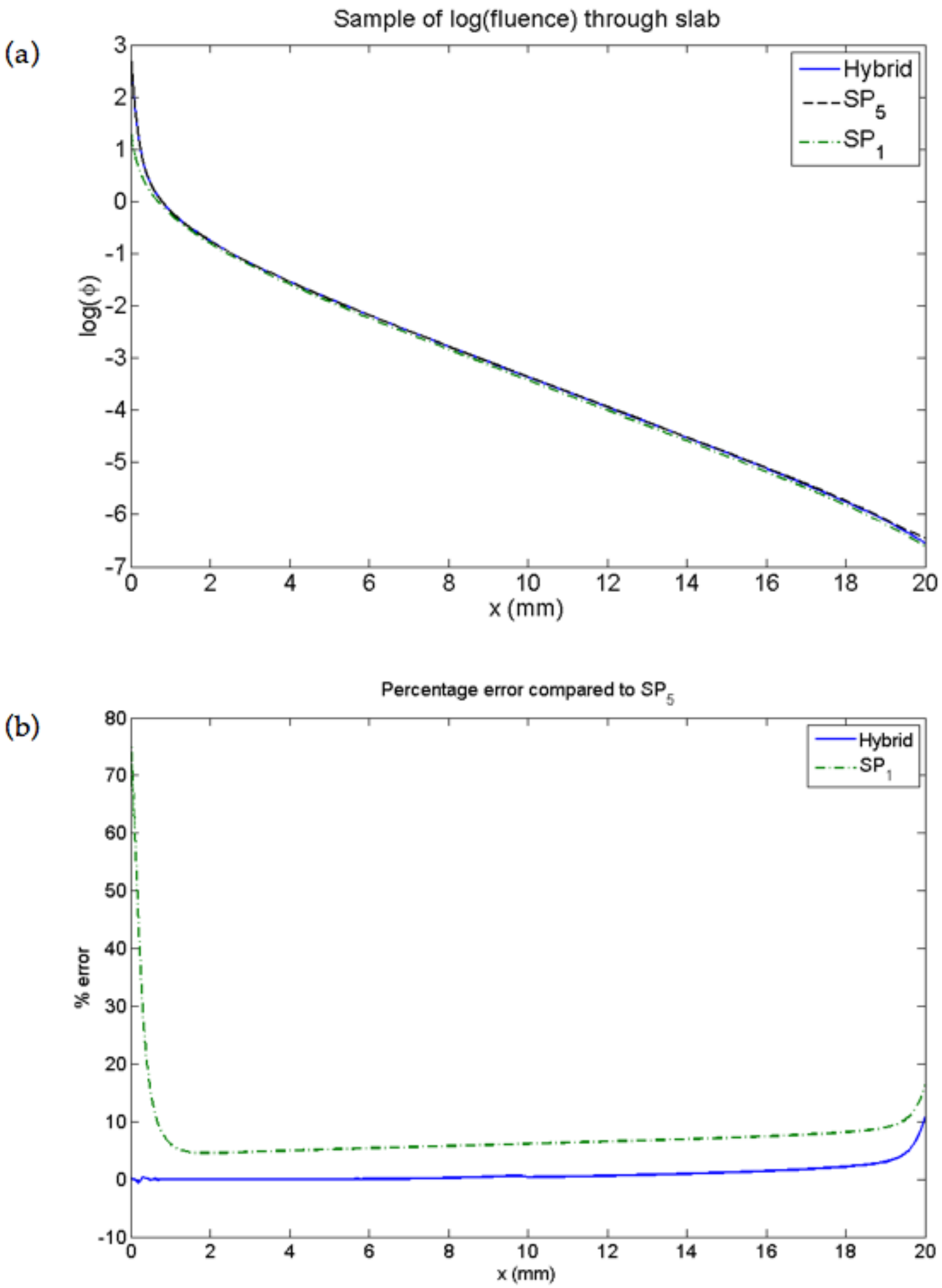


Figure 9.6 (a) Cross section of fluence through a slab geometry with optical properties of $\mu_a = 0.01mm^{-1}$, $\mu_s = 10mm^{-1}$, $g = 0.9$, and $n = 1.33$ (b) the percentage error of hybrid and SP$_1$ data compared to SP$_5$ along the same cross section.

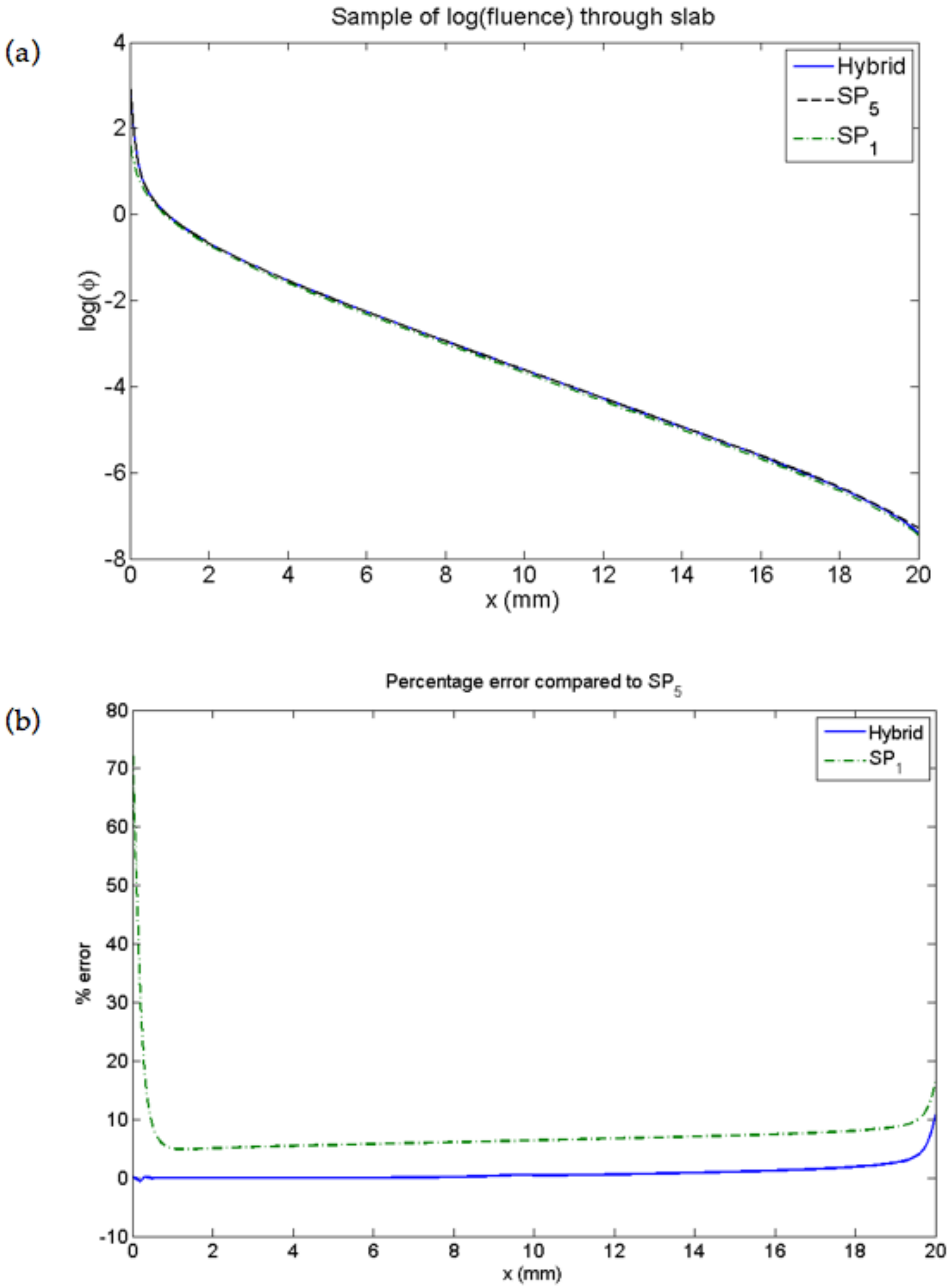


Figure 9.7 As in figure 9.6, but with increased scattering coefficient of $\mu_s = 15mm^{-1}$

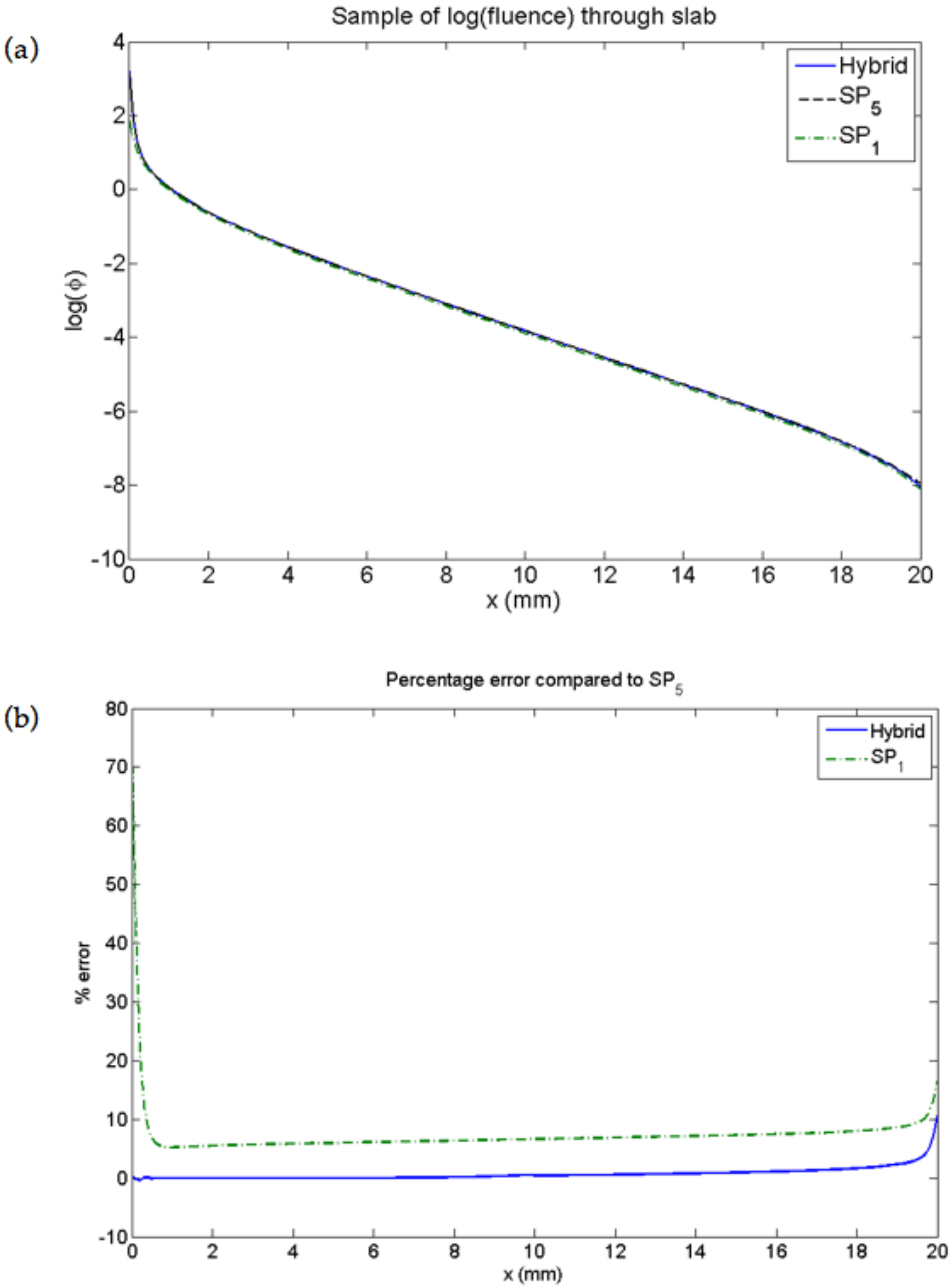


Figure 9.8 As in figure 9.6, but with increased scattering coefficient of $\mu_s = 20mm^{-1}$

In order to observe the effect of the hybrid model in less diffuse conditions, the absorption coefficient was increased to $\mu_a = 0.05mm^{-1}$. The percentage errors for scattering coefficients of $\mu_s = 10$, 15 and $20mm^{-1}$ are displayed in

Figure 9.9. Figure 9.9(a), which refers to optical properties of $\mu_a = 0.05mm^{-1}$ and $\mu_s = 10mm^{-1}$ again shows that the hybrid model has only a very small error near the source whilst the error near the detector rises to approximately 10%. A small increase in error is visible around 10mm into the slab, which relates to the boundary between the SP$_5$ and SP$_1$ regions. As the scattering coefficient increases, Figure 9.9(b)-(c), this error at the boundary decreases.

As the absorption coefficient is further increased to $\mu_a = 0.1mm^{-1}$, the error at the SP$_5$-SP$_1$ boundary becomes more pronounced and is present even at higher scattering coefficients, Figure 9.10. The hybrid data results in an error of approximately 15% near to the boundary, compared to the 20% error in SP$_1$ data at the same point.

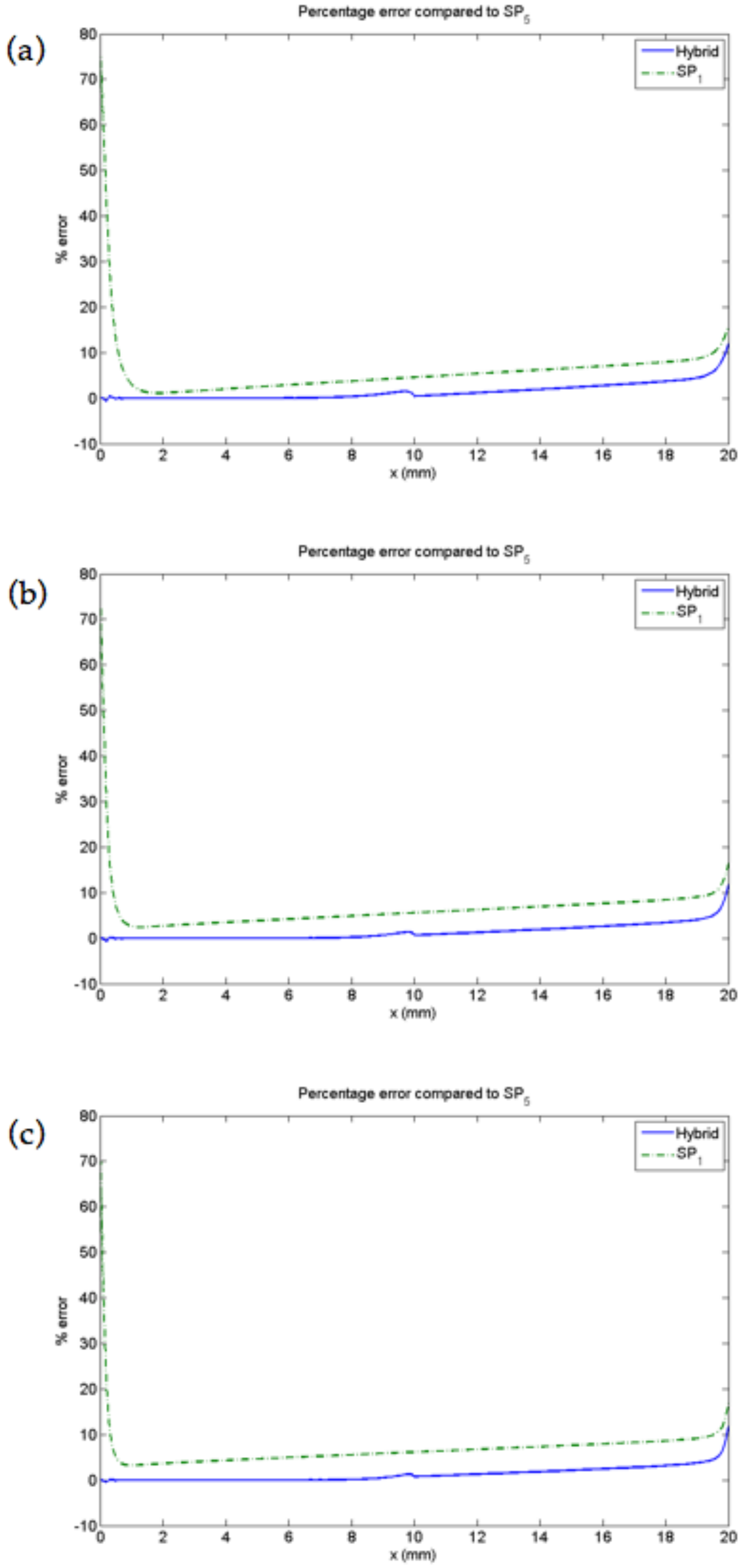


Figure 9.9 Percentage error of hybrid and SP$_1$ data compared to SP$_5$ for optical properties of $\mu_a = 0.05mm^{-1}$ and (a) $\mu_s = 10mm^{-1}$ (b) $\mu_s = 15mm^{-1}$ and (c) $\mu_s = 20mm^{-1}$

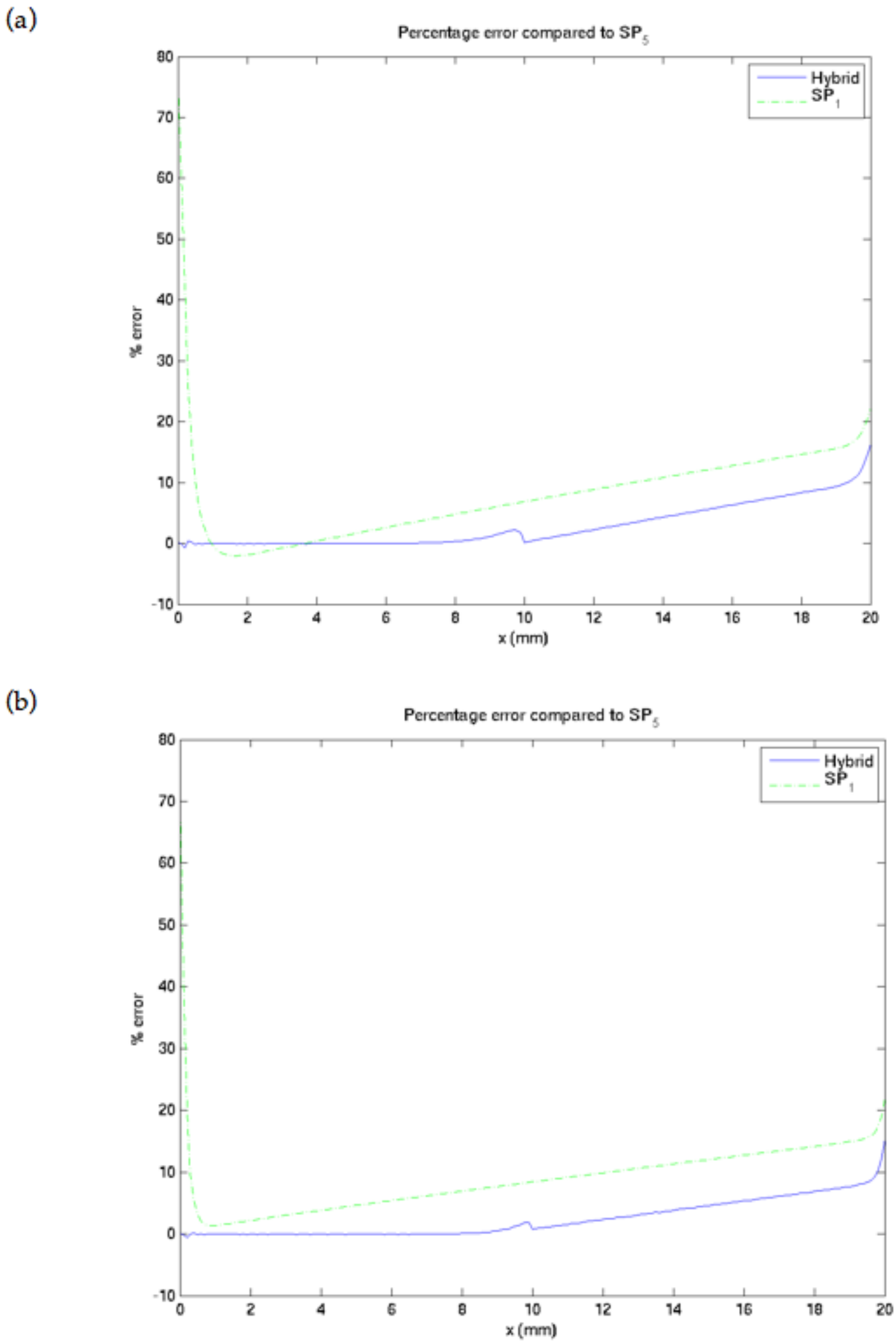


Figure 9.10 Percentage error of hybrid and SP$_1$ data compared to SP$_5$ for optical properties of $\mu_a = 0.1mm^{-1}$ and (a) $\mu_s = 10mm^{-1}$ (b) $\mu_s = 20mm^{-1}$

At all optical property combinations, it was noted that the error in the hybrid model started to increase near the detector. This suggested that the SP_5 model was required in these regions also. To test this hypothesis, a circular mesh with radius of 25mm was constructed. The outer layer, up to a depth of 10mm, is solved using the SP_5 approximation, and the SP_1 approximation was used in the inner layer, Figure 9.11.

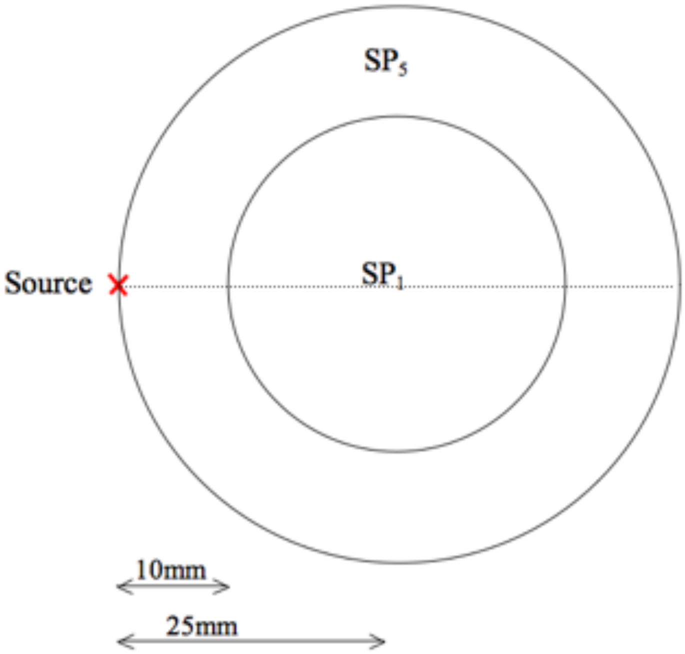


Figure 9.11 Circular hybrid mesh with a total radius of 25mm. The SP_5 approximation was used in the first 10mm, whilst the remainder was solved using the SP_1 approximation.

This geometry was then modelled using the same optical property combinations as used in the slab geometry above and a cross section of data was taken through the centre of the domain. In the first case, with $\mu_a = 0.01mm^{-1}$, $\mu_s = 10mm^{-1}$, $g = 0.9$, and $n = 1.33$, Figure 9.12, the error in the hybrid model is almost zero near to the source. The error gradually increases throughout the centre portion of the circle in which the SP_1 approximation is used to a maximum of approximately 4%. In the final 10mm, which is again modelled using the SP_5 approximation, the error remains constant. By comparison, the SP_1 data sees an error of over 50% at the source. The error then falls to around 6% before increasing to approximately 14% near the detector.

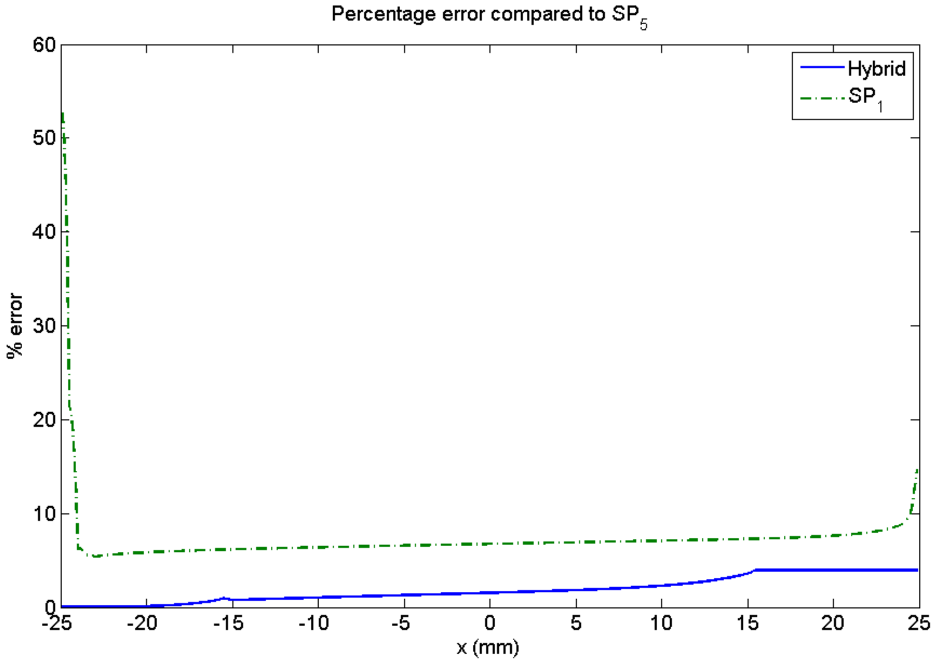


Figure 9.12 Percentage error of hybrid and SP1 data compared to SP5 for a cross section through the circular geometry shown in figure 9.10 with optical properties of $\mu_a = 0.01mm^{-1}$, $\mu_s = 10mm^{-1}$, $g = 0.9$ and $n = 1.33$

In the case with strong absorption, e.g. $\mu_a = 0.1mm^{-1}$ and $\mu_s = 20mm^{-1}$, Figure 9.13, the error at the end of the SP_1 region reaches a peak of approximately 30%. In the second SP_5 region, the error begins to fall to around 25% at the boundary.

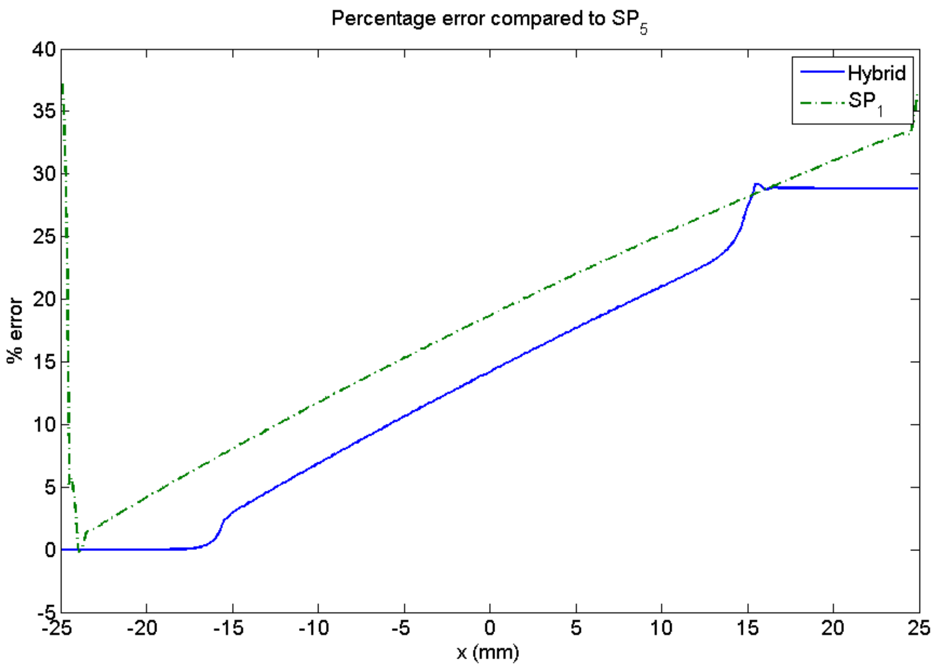


Figure 9.13 Percentage error of hybrid and SP1 data compared to SP5 for a cross section through the circular geometry shown in Figure 6 with optical properties of $\mu_a = 0.1mm^{-1}$, $\mu_s = 20mm^{-1}$, $g = 0.9$ and $n = 1.33$

As in the slab geometry, there was a notable artefact at the SP_5 SP_1 boundary when the absorption coefficient was increased as the domain becomes less diffuse. To overcome this artefact, the boundary between the two regions could be determined automatically for each individual mesh. Ideally, only one mesh would be required and the SP_5 region would be determined automatically.

9.3.1 Efficiency

The hybrid mass matrix has dimensions of $NN_{SP_1} + 3NN_{SP_5}$ where NN_{SP_1} is the number of nodes in the SP_1 part of the hybrid mesh and NN_{SP_1} is the number of nodes in the SP_5 part. As in this case, the SP_1 and SP_5 meshes are of equal size, this is equivalent to $4n_{SP_5}$. If the whole domain were to be modelled using the SP_5 approximation, the mass matrix would have dimensions of $8n_{SP_5}$

The reduction in the size of the mass matrix results in savings in the time it takes to invert the mass matrix. Table 9.1 shows the time taken to invert the mass matrix for the SP_1, SP_5, and hybrid models. It was seen that the hybrid model was almost 50% faster than the full SP_5 model whilst delivering almost identical results.

Table 9.1 Mass matrix inversion times for SP1, SP5 and hybrid models

	Mass Matrix Inversion Time (s)
SP_1	0.74
SP_5	6.4
Hybrid	3.5

9.4 Discussion & Conclusion

A novel SP_5/SP_1 hybrid model has been presented and has been tested using both a layered slab and circular geometries, both involving a single source and detector. The SP_5 region is used only in regions where the DA is likely to be inaccurate. In the slab geometry, the SP_5 approximation is used in the first layer, closest to the source, whilst the remainder is solved using the DA. The circular geometry is also divided into two layers. The outer region is solved using the SP_5 approximation and the inner region is solved with the DA. Unlike existing hybrid models which require two steps to obtain a set of calculated boundary data (Wang and Jacques 1993; Alexandrakis et al. 2000; Okada and Delpy 2003b; Tarvainen et al. 2005), the presented model requires just a single step.

After running the hybrid model on the slab and circular meshes for a range of optical properties, a cross section through the domain was taken and compared with the SP_5 and DA models and, using the SP_5 data as a gold standard, a percentage error was calculated.

In cases involving the slab geometry with low absorption coefficients, the hybrid and SP_5 data was almost identical. The percentage error was negligible throughout the majority of the domain, although the error increased to approximately 10% at the detector. The SP_1 data, by comparison showed large errors were seen near the source, and these errors appeared to propagate through the domain. As the scattering coefficient increased, however, the SP_1 data improved, although large errors were still present near the source and detector, while the hybrid data was unaffected. The error in the hybrid data near the detector, suggested these regions would also benefit form the higher ordered model.

As the absorption coefficient was increased, the hybrid data showed an increase in error at the SP_5/SP_1 boundary, which was more pronounced at lower scattering coefficients. This error was to be expected as greater depths are required for a sufficient number of scattering events to occur to create a diffuse light distribution. In these cases, the SP_5 region would need to be larger in order to accurately model the problem.

The circular mesh used the SP_5 approximation in the regions surround both the source and detector. In this case, the SP_1 data showed errors of up to 60% at the source although this did fall to approximately 5% before rising to around 15% at the detector. The hybrid data, however showed almost 0% error near the source. The error began to increase in the SP_1 region, before flattening off near the detector. Similar results were seen when a much larger absorption coefficient was used, although they were much more pronounced.

The results do, however, confirm that the higher order model is necessary in the regions surrounding both the source and detector.

The hybrid model was shown to provide the same degree of accuracy as the SP_5 model but with a significant speed increase of approximately 50%. These time savings, however, are a conservative estimate. The mesh used in this problem divided the slab geometry into two meshes of equal size. In reality, the region in which the SP_5 model would be necessary would be considerably smaller than the whole geometry. This would mean that the mass matrix would be considerably smaller and computation times would be more in line with the SP_1 model.

In the preliminary development of the hybrid model, two meshes must be created separately. Although this only needs to be performed once for a given geometry, it does not offer much flexibility for use with new geometries and varying critical depths. In future, it would be beneficial for a single mesh to be used with a routine capable of identifying nodes within a pre-determined distance of the source and duplicating the nodes on the boundary automatically.

At present, the process of manipulating the mass matrix in order to couple the SP_5 and SP_1 fields is performed by an external routine written as a MATLAB m-file. This is currently an inefficient process as it loops over large number of options. This could be drastically improved by simply re-writing the routine in a c based code.

Chapter 10 - Summary

Diffuse Optical Tomography (DOT) is an emerging imaging modality that can be used to non-invasively observe physiological function by probing tissue with near-infrared light. DOT promises to be a useful tool in a wide range of applications, including breast imaging, functional brain imaging and, with the use of fluorescent or bioluminescent contrast agents, molecular imaging in small animals. There are a range of alternative functional imaging modalities in existence that could have similar applications, such as fMRI or PET, but these have drawbacks such as high costs, or the use of ionising radiation. DOT has the ability to provide absolute functional information with relatively low costs using non-harmful near-infrared light.

The reconstruction of an image from experimental boundary data is an ill-posed and under-determined problem. The process generally involves a forward problem, in which the propagation of light through tissue is modelled, and an inverse problem in which the forward problem is repeatedly solved whilst the optical property map is updated iteratively. The accuracy of the final image is therefore dependent on an accurate forward model.

At present, the majority of forward models employ the diffusion approximation to the more accurate radiative transfer equation. This approximation has been shown to be acceptable in problems where the light distribution is fully diffuse. Such conditions could be encountered in large domains where photons undergo a large number of scattering events take place before reaching the detectors, or in domains involving scattering

coefficients that are much larger than the absorption coefficient. There are, however, many situations in which this approximation is no longer valid.
One of the rapidly growing applications of DOT is in the bioluminescent imaging of small animals. Commonly used bioluminescent agents emit light at a wavelength of approximately 560nm. This corresponds to strong absorption due to haemoglobin, which makes the DA invalid. The small geometries encountered in small animal imaging also limit the accuracy of the DA.

In this thesis, the development of a forward model based on the Simplified Spherical Harmonics approximation has been introduced. Before this work, just one other SP_N based model existed, which was solved using the finite difference method for continuous wave systems and was applied only to 2D geometries (Klose and Larsen 2006). The FDM, however, is limited to solving only simple geometries as a uniform grid of nodes is required. The model presented in this work, however, is solved for frequency domain systems using the Finite Element Method, which uses triangular or tetrahedral elements and can be used for complex 2D or 3D geometries. The model has been validated using a Monte Carlo model and was shown to give good agreement for a range of optical properties. It has also been shown that, for all orders N, the SP_N approximation provides an increase in accuracy over the commonly used DA while incurring a lower computational load than existing transport based models, such as the spherical harmonics or discrete ordinates method.

One of the features of the DA is that it combines the scattering coefficient and anisotropy factor into a single variable, the reduced scattering coefficient. As such, it is impossible to reconstruct the two properties individually. The

SP_N approximation, however, does not use the reduced scattering coefficient suggesting that both the scattering coefficient and anisotropy factor may be recoverable. Sensitivity mapping, however, showed that the sensitivity to changes in anisotropy factor is orders of magnitude lower than for absorption or scattering. By plotting a range of error maps, it was also shown that it was not possible to uniquely recover both scattering coefficient and anisotropy, even with higher SP_N orders.

The SP_7 based multi-parameter image reconstruction algorithm was found to be somewhat unstable due to the presence of strong artefacts and cross talk due to the under-determined nature of the problem. In order to reduce the number of unknowns, a SP_N region-based image reconstruction algorithm was implemented, using prior information which led to great improvements for orders N=1, 3 and 5.

The multi-parameter image reconstruction algorithms are based on the assumption that various orders of fluence are measurable at the boundary. In reality, this is not the case, and so a diffusion-based reconstruction algorithm was developed. This approach aimed to recover just two parameters per node, the absorption and scattering coefficients and is based on measurements of absolute fluence which were calculated using the SP_5 approximation. This approach was shown to recover optical property distributions with greater accuracy and less artefact than an equivalent model based on the DA.

In chapter 9, a novel SP_N- DA hybrid model was presented. Although a number of hybrid models have been developed to overcome the deficiencies of the DA, most of these required two independent steps in order to obtain a solution whereas the hybrid model presented here obtains results in a single step. The SP_N model is used in the regions close to the sources and detectors where the DA is known to be less accurate and the DA is solved in the remaining regions. It was shown, for a range of optical properties, that the hybrid model could produce data of similar accuracy to the full SP_5 model whilst providing an increase in efficiency. Due to the reduction in the size of the mass matrix, the inversion was completed up to 50% faster using the hybrid model. At present, however, the process of linking the two regions is performed using a MATLAB loop. As this is an inefficient process, the overall computation time for the hybrid is currently longer than for the full SP_5 model. Re-writing the region linking algorithm in C, however, will drastically reduce the computation time and the true efficiency savings of the hybrid model will be seen.

The newly developed SP_N models have been validated using Monte Carlo methods and have been compared to a previously validated DA model. The image reconstruction algorithms were tested using noise-free data in order to observe the effects of the various forward models alone. The sensitivity of the SP_N approximation to noisy data has not yet been studied.

The SP_N approximation would be particularly applicable to the field of bioluminescent imaging in small animals. Whereas the presented models have only been tested using sources located on the boundary, the aim of a bioluminescent imaging is to recover the distribution of internal light sources. It would be an interesting development to study the effects of the SP_N based models on this problem.

The SP_N toolbox developed in this work has now been included in the latest release of the NIRFAST package which is freely downloadable (www.nirfast.org). As NIRFAST currently has a wide user base, it is hoped that the SP_N based models will be applied to a wide range of experimental studies.

Appendix A – Boundary Coefficients

Boundary Coefficients where R_n values are calculated by numerically integrating equation 7.18

$$A_1 = -R_1,$$

$$B_1 = 3R_2,$$

$$C_1 = -\frac{3}{2}R_1 + \frac{5}{2}R_3,$$

$$D_1 = \frac{3}{2}R_2 - \frac{5}{2}R_4,$$

$$E_1 = \frac{15}{8}R_1 - \frac{35}{4}R_3 + \frac{63}{8}R_5,$$

$$F_1 = -\frac{15}{8}R_2 + \frac{35}{4}R_4 - \frac{63}{8}R_6,$$

$$G_1 = -\frac{35}{16}R_1 + \frac{315}{16}R_3 - \frac{693}{16}R_5 + \frac{429}{16}R_7,$$

$$H_1 = \frac{35}{16}R_2 - \frac{315}{16}R_4 + \frac{693}{16}R_6 - \frac{429}{16}R_8,$$

$$A_2 = -\frac{9}{4}R_1 + \frac{15}{2}R_3 - \frac{25}{4}R_5,$$

$$B_2 = \frac{63}{4}R_2 - \frac{105}{2}R_4 + \frac{175}{4}R_6,$$

$$C_2 = -\frac{3}{2}R_1 + \frac{5}{2}R_3,$$

$$D_2 = \frac{3}{2}R_2 - \frac{5}{2}R_4,$$

$$E_2 = -\frac{45}{16}R_1 + \frac{285}{16}R_3 - \frac{539}{16}R_5 + \frac{315}{16}R_7,$$

$$F_2 = \frac{45}{16}R_2 - \frac{285}{16}R_4 + \frac{539}{16}R_6 - \frac{315}{16}R_8,$$

$$G_2 = \frac{105}{32}R_1 - 35R_3 + \frac{1827}{16}R_5 - \frac{297}{2}R_7 + \frac{2145}{32}R_9,$$

$$H_2 = -\frac{105}{32}R_2 + 35R_4 - \frac{1827}{16}R_6 + \frac{297}{2}R_8 - \frac{2145}{32}R_{10},$$

$$A_3 = -\frac{225}{64}R_1 + \frac{525}{16}R_3 - \frac{3395}{32}R_5 + \frac{2205}{16}R_7 - \frac{3969}{64}R_9,$$

$$B_3 = \frac{2475}{64}R_2 - \frac{5775}{16}R_4 + \frac{37345}{32}R_6 - \frac{24255}{16}R_8 + \frac{43659}{64}R_{10},$$

$$C_3 = \frac{15}{8}R_1 - \frac{35}{4}R_3 + \frac{63}{8}R_5,$$

$$D_3 = -\frac{15}{8}R_2 + \frac{35}{4}R_4 - \frac{63}{8}R_6,$$

$$E_3 = -\frac{45}{16}R_1 + \frac{285}{16}R_3 - \frac{539}{16}R_5 + \frac{315}{16}R_7,$$

$$F_3 = \frac{45}{16}R_2 - \frac{285}{16}R_4 + \frac{539}{16}R_6 - \frac{315}{16}R_8,$$

$$G_3 = -\frac{525}{128}R_1 + \frac{7175}{128}R_3 - \frac{17325}{64}R_5 + \frac{37395}{64}R_7 - \frac{73689}{128}R_9 + \frac{27027}{128}R_{11},$$

$$H_3 = \frac{525}{128}R_2 - \frac{7175}{128}R_4 + \frac{17325}{64}R_6 - \frac{37395}{64}R_8 + \frac{73689}{128}R_{10} - \frac{27027}{128}R_{12},$$

$$A_4 = -\frac{1225}{256}R_1 + \frac{11025}{128}R_3 - \frac{147735}{256}R_5 + \frac{116655}{64}R_7 - \frac{750519}{256}R_9 + \frac{297297}{128}R_{11} - \frac{184041}{256}R_{13}$$

,

$$B_4 = 15\left(\frac{1225}{256}R_2 - \frac{11025}{128}R_4 + \frac{147735}{256}R_6 - \frac{116655}{64}R_8 + \frac{750519}{64}R_{10} - \frac{297297}{128}R_{12} + \frac{184041}{256}R_{14}\right)$$

,

$$C_4 = -\frac{35}{16}R_1 + \frac{315}{16}R_3 - \frac{693}{16}R_5 + \frac{429}{16}R_7,$$

$$D_4 = \frac{35}{16}R_2 - \frac{315}{16}R_4 + \frac{693}{16}R_6 - \frac{429}{16}R_8,$$

$$E_4 = \frac{105}{32}R_1 - 35R_3 + \frac{1827}{16}R_5 - \frac{297}{2}R_7 + \frac{2145}{32}R_9,$$

$$F_4 = -\frac{105}{32}R_2 + 35R_4 - \frac{1827}{16}R_6 + \frac{297}{2}R_8 - \frac{2145}{32}R_{10},$$

$$G_4 = -\frac{525}{128}R_1 + \frac{7175}{128}R_3 - \frac{17325}{64}R_5 + \frac{37395}{64}R_7 - \frac{73689}{128}R_9 + \frac{27027}{128}R_{11},$$

$$H_4 = \frac{525}{128}R_2 - \frac{7175}{128}R_4 + \frac{17325}{64}R_6 - \frac{37395}{64}R_8 + \frac{73689}{128}R_{10} - \frac{27027}{128}R_{12},$$

Appendix B – SP$_N$ Code

The code developed for the SP$_7$ model is included. The full NIRFAST toolbox, including the SP$_N$ based models, can be found at www.nirfast.org

```
function [mesh] = load_mesh(fn,plotflag)

% load_mesh NIRFAST mesh loading function
%   mesh = load_mesh(fn)
%   Loads meshes from specified root name fn
%   Must contain *.nodes, *. elem, *.param
%   Optional *.source, *.meas, *.link, *.region, *.excoeff
%   Mesh can be either:
%           standard 'stnd'
%           Fluorescence 'fluor',
%          spn 'stnd_spn',
%           Spectral 'spec',
%           Fluorescence Spectral 'spec_fluor'
%
%   mesh = load_mesh(fn,1)
%   where the last argument is optional
%   Plots boundary nodes and the location of sources and detectors


% Set mesh name
mesh.name = fn;

%********************************************
%Read mesh nodes
if exist([fn '.node']) == 0
  disp([fn '.node file is not present']);
  mesh = [];
  return
elseif exist([fn '.node']) == 2
  [mesh.nodes] = myload([fn '.node']);
  mesh.bndvtx = mesh.nodes(:,1); %sets 1 if boundary node, 0 if internal
  mesh.nodes = mesh.nodes(:,2:end);
end

%********************************************
%Read mesh element
if exist([fn '.elem']) == 0
  disp([fn '.elem file is not present']);
  mesh = [];
  return
elseif exist([fn '.elem']) == 2
  mesh.elements = myload([fn '.elem']);
  [junk,dim]=size(mesh.elements);
  mesh.dimension = dim-1;
  if mesh.dimension == 2
    mesh.nodes(:,3) = 0;
  end
end

%********************************************
```

```
%Read appriopriate mesh parameters
if exist([fn '.param']) == 0
  disp([fn '.param file is not present']);
  mesh = [];
  return

  % Loading up mesh parameters from fn.param
elseif exist([fn '.param']) == 2

  param = importdata([fn '.param']);
  % Convert from Version 1
  if isfield(param,'textdata') == 0
    [junk,pa] = size(param);
    % Convert old standard
    if pa == 3
      p.data = param;
      p.textdata = cellstr('stnd');
      % Convert old Fluorescence
    elseif pa == 8
      p.data = param;
      p.textdata = cellstr('fluor');
    else
      disp('Error in *.param file');
      mesh = [];
      return;
    end
    clear param; param = p;
    clear pa junk p
  end

  if isfield(param,'textdata') == 1
    % Load standard Nirfast Mesh
    if strcmp(param.textdata(1,1),'stnd') == 1
      mesh.type = 'stnd';
      param = param.data;
      mesh.mua = param(:,1);
      mesh.kappa = param(:,2);
      mesh.ri = param(:,3);
      mesh.mus = ((1./mesh.kappa)./3)-mesh.mua;

    % Load standard Fluorfast Mesh
    elseif strcmp(param.textdata(1,1),'fluor') == 1
      mesh.type = 'fluor';
      param = param.data;
      mesh.muax = param(:,1);
      mesh.kappax = param(:,2);
      mesh.musx = ((1./mesh.kappax)./3)-mesh.muax;
      mesh.ri = param(:,3);
      mesh.muam = param(:,4);
      mesh.kappam = param(:,5);
      mesh.musm = ((1./mesh.kappam)./3)-mesh.muam;
      mesh.muaf =  param(:,6);
      mesh.eta =  param(:,7);
      mesh.tau =  param(:,8);
      clear param

    % Load SPN Mesh
    elseif strcmp(param.textdata(1,1),'stnd_spn') == 1
      mesh.type = 'stnd_spn';
      param = param.data;
      mesh.mua = param(:,1);
      mesh.mus = param(:,2);
      mesh.g=param(:,3);
      mesh.ri = param(:,4);

    % Load spectral Mesh
    elseif strcmp(param.textdata(1,1),'spec') == 1
      mesh.type = 'spec';
      mesh.chromscattlist = param.textdata(2:end,1);
      % Get Scatter Amplitude
      ind = find(strcmpi(mesh.chromscattlist,'S-Amplitude'));
      if isempty(ind)
        disp('Scatter Amplitude not present in *.param file')
        mesh = [];
        return
      else
        mesh.sa = param.data(:,ind);
      end
      % Get Scatter Power
      ind = find(strcmpi(mesh.chromscattlist,'S-Power'));
      if isempty(ind)
        disp('Scatter Power not present in *.param file')
        mesh = [];
        return
      else
        mesh.sp = param.data(:,ind);
      end
      % Get Chromophore concentrations
      mesh.conc = [];
      k = 1;
      for i = 1 : length(mesh.chromscattlist)
        if strcmpi(mesh.chromscattlist(i),'S-Amplitude') == 0 & ...
           strcmpi(mesh.chromscattlist(i),'S-Power') == 0
          mesh.conc(:,k) = param.data(:,i);
          k = k + 1;
        end
      end
      clear k

      % Get extintion coefficient values
      if exist([fn '.excoef']) ~= 0
        excoef = importdata([fn '.excoef']);
      else
        excoef = importdata('excoef.txt');
      end
      mesh.wv = excoef.data(:,1);
      k = 1;
      for i = 1 : length(mesh.chromscattlist)
        if strcmpi(mesh.chromscattlist(i),'S-Amplitude') == 0 & ...
           strcmpi(mesh.chromscattlist(i),'S-Power') == 0
          ind = find(strcmpi(excoef.textdata,mesh.chromscattlist(i,1)));
          if isempty(ind)
            disp(['The Chromophore ' char(mesh.chromscattlist(i,1)) ...
           ' is not defined in extinction coefficient file']);
            mesh = [];
            return;
          else
            mesh.excoef(:,k) = excoef.data(:,ind+1);
            k = k + 1;
          end
        end
      end
    end
  end
end
```

```
%********************************************
if exist([fn '.region']) ~= 0
 region = myload([fn '.region']);
 mesh.region = region;
elseif exist([fn '.region']) ~= 2
 mesh.region = zeros(length(mesh.nodes),1);
end

%********************************************
% Load source locations
if exist([fn '.source']) == 0
 disp([fn '.source file is not present']);
elseif exist([fn '.source']) == 2
 source = importdata([fn '.source']);
 if isfield(source,'textdata') == 0
  mesh.source.fixed = 0;
  [ns,nc]=size(source);
  if nc == mesh.dimension
    mesh.source.fwhm = ones(ns,1).*3;
    mesh.source.coord = source;
  elseif nc == mesh.dimension+1
    mesh.source.fwhm = source(:,mesh.dimension+1);
    mesh.source.coord = source(:,1:mesh.dimension);
  end
  if strcmp(mesh.type,'stnd') == 1
   mus_eff = mesh.mus;
  elseif strcmp(mesh.type,'fluor') == 1
   mus_eff = mesh.musx;
  elseif strcmp(mesh.type,'spec') == 1
   [mua,mus] = calc_mua_mus(mesh,mesh.wv(1));
   mus_eff = mus;
   clear mua mus
  end
  disp('Moving Sources');
  [mesh]=move_source(mesh,mus_eff,3);
 elseif isfield(source,'textdata') == 1
  mesh.source.fixed = 1;
  [ns,nc]=size(source.data);
  if nc == mesh.dimension
    mesh.source.coord = source.data;
    mesh.source.fwhm = ones(ns,1).*0;
  elseif nc == mesh.dimension+1
    mesh.source.fwhm = source.data(:,mesh.dimension+1);
    mesh.source.coord = source.data(:,1:mesh.dimension);
  end
  disp('Fixed Sources');
  if mesh.dimension == 2
   [ind,int_func] = tsearchn(mesh.nodes(:,1:2),...
     mesh.elements,...
     mesh.source.coord(:,1:2));
  elseif mesh.dimension == 3
   [ind,int_func] = tsearchn(mesh.nodes,...
     mesh.elements,...
     mesh.source.coord);
  end
  if any(isnan(ind)) == 1
   disp('Sources outside mesh');
  end
 end
 clear source mus_eff
end
```

```
%*********************************************
% Load detector locations
if exist([fn '.meas']) == 0
  disp([fn '.meas file is not present']);
elseif exist([fn '.meas']) == 2
  meas = importdata([fn '.meas']);
  if isfield(meas,'textdata') == 0
    mesh.meas.fixed = 0;
    mesh.meas.coord = meas;
    disp('Moving Detectors');
    [mesh]=move_detector(mesh);
  elseif isfield(meas,'textdata') == 1
    mesh.meas.fixed = 1;
    mesh.meas.coord = meas.data;
    disp('Fixed Detectors');
    if mesh.dimension == 2
      [ind,int_func] = tsearchn(mesh.nodes(:,1:2),...
          mesh.elements,...
          mesh.meas.coord(:,1:2));
    elseif mesh.dimension == 3
      [ind,int_func] = tsearchn(mesh.nodes,...
          mesh.elements,...
          mesh.meas.coord);
    end
    if any(isnan(ind)) == 1
      disp('Detectors outside mesh');
    else
      mesh.meas.int_func = [ind int_func];
    end
  end
  clear meas
end

%*********************************************
% Load link list for source and detector
if exist([fn '.link']) == 0
  disp([fn '.link file is not present']);
elseif exist([fn '.link']) == 2
  mesh.link = myload([fn '.link']);
end

%*********************************************
% Load identidity list if exists for the internal RI boundary nodes
if exist([fn '.ident']) == 2
  mesh.ident = myload([fn '.ident']);
end

%*********************************************
% speed of light in medium
% If a spectral mesh, assume Refractive index = 1.33
if strcmp(mesh.type,'spec') == 1
  mesh.ri = ones(length(mesh.nodes),1).*1.33;
end
mesh.c=(3e11./mesh.ri);

% Set boundary coefficient using definition of A using the Fresenel's law:
if strcmp(mesh.type,'stnd_spn') ~= 1
  f=0.9;
  Ro=((mesh.ri-1).^2)./((mesh.ri+1).^2);
  thetac=asin(1./mesh.ri);
  cos_theta_c=abs(cos(asin(1./mesh.ri)));
  A=((2./(1-Ro)) - 1 + cos_theta_c.^3) ./ (1 - cos_theta_c.^2);
  mesh.ksi=1./(2*A);
end

%*********************************************
% area of each element
if mesh.dimension == 2
  mesh.element_area = ele_area_c(mesh.nodes(:,1:2),...
        mesh.elements);
  mesh.support = mesh_support(mesh.nodes(:,1:2),...
            mesh.elements,...
            mesh.element_area);
elseif  mesh.dimension == 3
  mesh.element_area = ele_area_c(mesh.nodes,mesh.elements);
  mesh.support = mesh_support(mesh.nodes,...
            mesh.elements,...
            mesh.element_area);
end

%*********************************************
if nargin == 1
  plotflag = 0;
end
if plotflag == 1 % will plot source/detector positions
  figure;
  ind = find(mesh.bndvtx==1);
  if mesh.dimension == 2
    plot(mesh.source.coord(:,1),...
        mesh.source.coord(:,2),'ro','LineWidth',2,'MarkerSize',8);
    hold on;
    plot(mesh.meas.coord(:,1),...
        mesh.meas.coord(:,2),'bx','LineWidth',2,'MarkerSize',8);
    plot(mesh.nodes(ind,1),mesh.nodes(ind,2),'c.');
    axis equal;
    legend('Source','Detector');
  elseif mesh.dimension == 3
    plot3(mesh.source.coord(:,1),...
        mesh.source.coord(:,2),...
        mesh.source.coord(:,3),'ro',...
        'LineWidth',2,'MarkerSize',8);
    hold on;
    plot3(mesh.meas.coord(:,1),...
        mesh.meas.coord(:,2),...
        mesh.meas.coord(:,3),'bx',...
        'LineWidth',2,'MarkerSize',8);
    plot3(mesh.nodes(ind,1),...
        mesh.nodes(ind,2),...
        mesh.nodes(ind,3),'c.');
    axis equal;
    legend('Source','Detector');
  end
end
```

```
function [data,mesh]=femdata_sp7(mesh,frequency)

% [data,mesh]=femdata_sp7(mesh,frequency)
% Calculates boundary data (phase and amplitude) for
% each complex moment of fluence (eq. 7.11) and total fluence
% for a given spn mesh at a given frequency (MHz).
% outputs phase and amplitude in structure data
% and mesh information in mesh
% If not a workspace variable, load mesh
if ischar(mesh)== 1
  mesh = load_mesh(mesh);
end

% modulation frequency
omega = 2*pi*frequency*1e6;
nvtx=length(mesh.nodes);

% Calculate boundary coefficients
[f1,f2,f3,f4,g1,g2,g3,g4,h1,h2,h3,h4,i1,i2,i3,i4] = ksi_calc_sp7(mesh);

% Create FEM matrices
if mesh.dimension==2

  [i,j,k1,k3,k5,k7,c,c23,c49,c64,c8,c16,c32,c128,c256,c2_59,c2_16,c2_49,...
    c2_8,c2_32,c2_64,c4_9,c4_54,c4_324,c6_13,B,F1,F2,F3,F4,G1,G2,G3,G4,...
    H1,H2,H3,H4,I1,I2,I3,I4,ib,jb]=gen_matrices_2d_sp7(mesh.nodes(:,1:2),...
    sort(mesh.elements')',...
    mesh.bndvtx,...
    mesh.mua,...
    mesh.mus,...
    mesh.g,...
    f1,...
    f2,...
    f3,...
    f4,...
    g1,...
    g2,...
    g3,...
    g4,...
    h1,...
    h2,...
    h3,...
    h4,...
    i1,...
    i2,...
    i3,...
    i4,...
    mesh.c,...
    omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
K7= sparse(i(1:nz_i),j(1:nz_i),k7(1:nz_i));
C = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23 = sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49 = sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64 = sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16 = sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C32 = sparse(i(1:nz_i),j(1:nz_i),c32(1:nz_i));
C128 = sparse(i(1:nz_i),j(1:nz_i),c128(1:nz_i));
C256 = sparse(i(1:nz_i),j(1:nz_i),c256(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C2_8 = sparse(i(1:nz_i),j(1:nz_i),c2_8(1:nz_i));
C2_32 = sparse(i(1:nz_i),j(1:nz_i),c2_32(1:nz_i));
C2_64 = sparse(i(1:nz_i),j(1:nz_i),c2_64(1:nz_i));
```

```
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
C4_54 = sparse(i(1:nz_i),j(1:nz_i),c4_54(1:nz_i));
C4_324 = sparse(i(1:nz_i),j(1:nz_i),c4_324(1:nz_i));
C6_13 = sparse(i(1:nz_i),j(1:nz_i),c6_13(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));

[i,j]=size(C);
nz_i = length(nonzeros(ib));
F1 = sparse(ib(1:nz_i),jb(1:nz_i),F1(1:nz_i));
F2 = sparse(ib(1:nz_i),jb(1:nz_i),F2(1:nz_i));
F3 = sparse(ib(1:nz_i),jb(1:nz_i),F3(1:nz_i));
F4 = sparse(ib(1:nz_i),jb(1:nz_i),F4(1:nz_i));
G1 = sparse(ib(1:nz_i),jb(1:nz_i),G1(1:nz_i));
G2 = sparse(ib(1:nz_i),jb(1:nz_i),G2(1:nz_i));
G3 = sparse(ib(1:nz_i),jb(1:nz_i),G3(1:nz_i));
G4 = sparse(ib(1:nz_i),jb(1:nz_i),G4(1:nz_i));
H1 = sparse(ib(1:nz_i),jb(1:nz_i),H1(1:nz_i));
H2 = sparse(ib(1:nz_i),jb(1:nz_i),H2(1:nz_i));
H3 = sparse(ib(1:nz_i),jb(1:nz_i),H3(1:nz_i));
H4 = sparse(ib(1:nz_i),jb(1:nz_i),H4(1:nz_i));
I1 = sparse(ib(1:nz_i),jb(1:nz_i),I1(1:nz_i));
I2 = sparse(ib(1:nz_i),jb(1:nz_i),I2(1:nz_i));
I3 = sparse(ib(1:nz_i),jb(1:nz_i),I3(1:nz_i));
I4 = sparse(ib(1:nz_i),jb(1:nz_i),I4(1:nz_i));

elseif mesh.dimension==3
 [i,j,k1,k3,k5,k7,c,c23,c49,c64,c8,c16,c32,c128,c256,c2_59,c2_16,c2_49,...
   c2_8,c2_32,c2_64,c4_9,c4_54,c4_324,c6_13,B,F1,F2,F3,F4,G1,G2,G3,G4,...
   H1,H2,H3,H4,I1,I2,I3,I4,ib,jb]=gen_matrices_3d_sp7(mesh.nodes,...
   sort(mesh.elements')',...
   mesh.bndvtx,...
   mesh.mua,...
   mesh.mus,...
   mesh.g,...
   f1,...
   f2,...
   f3,...
   f4,...
   g1,...
   g2,...
   g3,...
   g4,...
   h1,...
   h2,...
   h3,...
   h4,...
   i1,...
   i2,...
   i3,...
   i4,...
   mesh.c,...
  omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
K7= sparse(i(1:nz_i),j(1:nz_i),k7(1:nz_i));
C = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23 = sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49 = sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64 = sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
```

```
C16 = sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C32 = sparse(i(1:nz_i),j(1:nz_i),c32(1:nz_i));
C128 = sparse(i(1:nz_i),j(1:nz_i),c128(1:nz_i));
C256 = sparse(i(1:nz_i),j(1:nz_i),c256(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C2_8 = sparse(i(1:nz_i),j(1:nz_i),c2_8(1:nz_i));
C2_32 = sparse(i(1:nz_i),j(1:nz_i),c2_32(1:nz_i));
C2_64 = sparse(i(1:nz_i),j(1:nz_i),c2_64(1:nz_i));
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
C4_54 = sparse(i(1:nz_i),j(1:nz_i),c4_54(1:nz_i));
C4_324 = sparse(i(1:nz_i),j(1:nz_i),c4_324(1:nz_i));
C6_13 = sparse(i(1:nz_i),j(1:nz_i),c6_13(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));
F1 = sparse(i(1:nz_i),j(1:nz_i),F1(1:nz_i));
F2 = sparse(i(1:nz_i),j(1:nz_i),F2(1:nz_i));
F3 = sparse(i(1:nz_i),j(1:nz_i),F3(1:nz_i));
F4 = sparse(i(1:nz_i),j(1:nz_i),F4(1:nz_i));
G1 = sparse(i(1:nz_i),j(1:nz_i),G1(1:nz_i));
G2 = sparse(i(1:nz_i),j(1:nz_i),G2(1:nz_i));
G3 = sparse(i(1:nz_i),j(1:nz_i),G3(1:nz_i));
G4 = sparse(i(1:nz_i),j(1:nz_i),G4(1:nz_i));
H1 = sparse(i(1:nz_i),j(1:nz_i),H1(1:nz_i));
H2 = sparse(i(1:nz_i),j(1:nz_i),H2(1:nz_i));
H3 = sparse(i(1:nz_i),j(1:nz_i),H3(1:nz_i));
H4 = sparse(i(1:nz_i),j(1:nz_i),H4(1:nz_i));
I1 = sparse(i(1:nz_i),j(1:nz_i),I1(1:nz_i));
I2 = sparse(i(1:nz_i),j(1:nz_i),I2(1:nz_i));
I3 = sparse(i(1:nz_i),j(1:nz_i),I3(1:nz_i));
I4 = sparse(i(1:nz_i),j(1:nz_i),I4(1:nz_i));

end

clear i* j* k* c* omega nz_i

% Add complex component to absorption moments due to
% frequency dependence
C=C+B;
C23=C23+B;
C49=C49+B;
C64=C64+B;
C8=C8+B;
C16=C16+B;
C32=C32 +B;
C128 = C128+B;
C256=C256+B;
C2_59=C2_59+B;
C2_16=C2_16+B;
C2_49=C2_49+B;
C2_8=C2_8+B;
C2_32=C2_32+B;
C2_64=C2_64+B;
C4_9=C4_9+B;
C4_54 = C4_54+B;
C4_324 = C4_324+B;
C6_13 = C6_13+B;

M1=K1+C+F1;
M2=K3+C49+C2_59+F2;
M3=K5+C64+C2_16+C4_9+F3;
M4=K7+C256+C2_64+C4_324+C6_13+F4;
```

```
%=============================================

if isfield(mesh,'ident') == 1
  disp('Modifying for refractive index')
  M = bound_int(MASS,mesh);
  MASS = M;
  clear M
end

% Now calculate source vector
% NOTE last term in mex file 'qvec' is the source FWHM
[nnodes,junk]=size(mesh.nodes);
[nsource,junk]=size(mesh.source.coord);
qvec = spalloc(nnodes,nsource,nsource*100);
if mesh.dimension == 2
  for i = 1 : nsource
    if mesh.source.fwhm(i) == 0
        qvec(:,i) = gen_source_point(mesh,mesh.source.coord(i,1:2));
    else
      qvec(:,i) = gen_source(mesh.nodes(:,1:2),...
           sort(mesh.elements')',...
           mesh.dimension,...
           mesh.source.coord(i,1:2),...
           mesh.source.fwhm(i));
    end
  end
elseif mesh.dimension == 3
  for i = 1 : nsource
    if mesh.source.fwhm(i) == 0
        qvec(:,i) = gen_source_point(mesh,mesh.source.coord(i,1:3));
    else
    qvec(:,i) = gen_source(mesh.nodes,...
           sort(mesh.elements')',...
           mesh.dimension,...
           mesh.source.coord(i,:),...
           mesh.source.fwhm(i));
    end
  end
end

% Build Source term
qvec=[qvec;(-2/3)*qvec;(8/15)*qvec;(-16/35)*qvec];

% Build system matrix eq. 7.23
MASS=[M1 -(C23-G1) +(C8-H1) -(C16-I1);...
    -(C23-G2) M2 -(C16-C2_49-H2) (C32+C2_8-I2);...
    (C8-G3) -(C16+C2_49-H3) M3 -(C128+C2_32+C4_54-I3);...
    (-C16-G4) (C32+C2_8-H4) -(C128+C2_32+C4_54-I4) M4];

% ======================================
% Optimise MASS matrix
[MASS_opt,Q_opt,invsort]=optimise(MASS,qvec);

%======================================

% Calculate fluence
phi_all=get_field(MASS_opt,mesh,Q_opt);

% Re-order elements
phi_all=phi_all(invsort,:);

% Extract complex moments of fluence
```

```
data.phi1=phi_all(1:nvtx,:);
data.phi2=phi_all((nvtx+1):(2*nvtx),:);
data.phi3=phi_all(((2*nvtx)+1):3*nvtx,:);
data.phi4=phi_all(((3*nvtx)+1):4*nvtx,:);

% Calculate total fluence
data.phi = data.phi1 - (2/3).*data.phi2+...
      (8/15).*data.phi3-(16/35).*data.phi4;
clear phi*

% Calculate boundary data
[data.complex1]=get_boundary_data(mesh,data.phi1);
[data.complex2]=get_boundary_data(mesh,data.phi2);
[data.complex3]=get_boundary_data(mesh,data.phi3);
[data.complex4]=get_boundary_data(mesh,data.phi4);
[data.complex]=get_boundary_data(mesh,data.phi);

% Map complex data to amplitude and phase

% Phi 1
data.amplitude1 = abs(data.complex1);
data.phase1 = atan2(imag(data.complex1),...
          real(data.complex1));
data.phase1(find(data.phase1<0)) = data.phase1(find(data.phase1<0)) + (2*pi);
data.phase1 = data.phase1*180/pi;
data.paa1 = [data.amplitude1 data.phase1];

% Phi 2
data.amplitude2 = abs(data.complex2);
data.phase2 = atan2(imag(data.complex2),...
          real(data.complex2));
data.phase2(find(data.phase2<0)) = data.phase2(find(data.phase2<0)) + (2*pi);
data.phase2 = data.phase2*180/pi;
data.paa2 = [data.amplitude2 data.phase2];

% Phi 3
data.amplitude3 = abs(data.complex3);
data.phase3 = atan2(imag(data.complex3),...
          real(data.complex3));
data.phase3(find(data.phase3<0)) = data.phase3(find(data.phase3<0)) + (2*pi);
data.phase3 = data.phase3*180/pi;
data.paa3 = [data.amplitude3 data.phase3];

% Phi 4
data.amplitude4 = abs(data.complex4);
data.phase4 = atan2(imag(data.complex4),...
          real(data.complex4));
data.phase4(find(data.phase4<0)) = data.phase4(find(data.phase4<0)) + (2*pi);
data.phase4 = data.phase4*180/pi;
data.paa4 = [data.amplitude4 data.phase4];

% Total
data.amplitude = abs(data.complex);
data.phase = atan2(imag(data.complex),...
          real(data.complex));
data.phase(find(data.phase<0)) = data.phase(find(data.phase<0)) + (2*pi);
data.phase = data.phase*180/pi;
data.paa = [data.amplitude data.phase];
```

gen_matrices.c

calculates FEM matrices. Implemented in c for efficiency and compiled within MATLAB.

```
#include <stdio.h>
#include <math.h>
#include <mex.h>
double gradphi(double g[3][2], double kappa[3], double val[3][3]);
double phidotphi(double g[3][2], double mua[3], double val[3][3]);
double boundary_int(double g[3][2], double bnd_index[3],
          double ksir[3], double valr[2][2]);
double bound2(double g[2][2], int il, int im, double ksir[2], double *val);

/* -------- Heart of the mex file----------- */
/* Define variables to be used:  f – i relate to boundary conditions, k_n are diffusion terms, */
/* C_n are various orders and fractions of absorption coefficient */

void mainloop(double *nodes,
           double *elements, double *bndvtx,
           double *mua,double *mus,
           double *g,double *f1,
           double *f2,double *f3,
double *f4,
double *g1,
double *g2,
double *g3,
double *g4,
double *h1,
double *h2,
double *h3,
double *h4,
double *i1,
double *i2,
double *i3,
double *i4,
double *c,double *omega,
int nodem,
int noden,
int elemm,
int elemn,
double *I,
double *J,
double *K1,
double *K3,
double *K5,
double *K7,
double *C,
double *C23,
double *C49,double *C64,
double *C8,
double *C16,
double *C32,
double *C128,
double *C256,
double *C2_59,
double *C2_16,
double *C2_49,
double *C2_8,
double *C2_32,
double *C2_64,
double *C4_9,
double *C4_54,
double *C4_324,
double *C6_13,
double *B,
double *F1,
double *F2,
double *F3,
double *F4,
double *G1,
double *G2,
double *G3,
double *G4,
double *H1,
double *H2,
double *H3,
double *H4,
double *I1,
double *I2,
double *I3,
double *I4,
double *X,
double *Y
)

{
   int ele, i, j, index[3], k,b;
   double indextmp, tri_vtx[3][2];
   double bnd_index[3], kappa1_index[3], kappa3_index[3],kappa5_index[3],
kappa7_index[3];
   double mus_index[3], g_index[3];
   double c_index[3], mua_index[3],mua2_59_index[3], mua23_index[3],mua49_index[3];
   double mua64_index[3],
mua8_index[3],mua16_index[3],mua32_index[3],mua128_index[3],mua256_index[3];
   double
mua2_16_index[3],mua2_49_index[3],mua2_8_index[3],mua2_32_index[3],mua2_64_index[3];
   double mua4_9_index[3],mua4_54_index[3],mua4_324_index[3],mua6_13_index[3];
   double
f1_index[3],f2_index[3],f3_index[3],f4_index[3],g1_index[3],g2_index[3],g3_index[3],g4_index
[3];
   double
h1_index[3],h2_index[3],h3_index[3],h4_index[3],i1_index[3],i2_index[3],i3_index[3],i4_index
[3];

   double k1_val[3][3],k3_val[3][3], k5_val[3][3],k7_val[3][3],b_val[3][3];
   double c_val[3][3],c23_val[3][3],c49_val[3][3],c64_val[3][3],c8_val[3][3],c16_val[3][3];
   double c32_val[3][3],c128_val[3][3],c256_val[3][3];
   double c2_59_val[3][3],
c2_49_val[3][3],c2_8_val[3][3],c2_16_val[3][3],c2_32_val[3][3],c2_64_val[3][3];
   double c4_9_val[3][3],c4_54_val[3][3],c4_324_val[3][3],c6_13_val[3][3];
   double f1_valr[2][2], f2_valr[2][2],f3_valr[2][2],f4_valr[2][2];
   double g1_valr[2][2], g2_valr[2][2],g3_valr[2][2],g4_valr[2][2];
   double h1_valr[2][2], h2_valr[2][2],h3_valr[2][2],h4_valr[2][2];
   double i1_valr[2][2], i2_valr[2][2],i3_valr[2][2],i4_valr[2][2];

/* For each element, calculate optical properties at nodes */

   k = 0;
   for (ele=0; ele<elemm; ++ele){
      for (i=0; i<elemn; ++i){
         indextmp = *(elements+(ele+(i*elemm)));
         index[i] = indextmp;
      }

      for (i=0; i<elemn; ++i){
         mua_index[i] = *(mua+(index[i]-1));
         mus_index[i] = *(mus+(index[i]-1));
         g_index[i] = *(g+(index[i]-1));
         mua23_index[i] = (2./3.)*mua_index[i];
         mua49_index[i] = (4./9.)*mua_index[i];
         mua64_index[i] = (64./225.)*mua_index[i];
         mua8_index[i] = (8./15.)*mua_index[i];
         mua16_index[i] = (16./45.)*mua_index[i];
         mua32_index[i] = (32./105.)*mua_index[i];
         mua128_index[i]= (128./525.)*mua_index[i];
         mua256_index[i]= (256./1225.)*mua_index[i];
         mua2_59_index[i]          =          (5./9.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i])));
         mua2_16_index[i]          =          (16./45.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i])));
         mua2_49_index[i]          =          (4./9.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i])));
         mua2_8_index[i]=(8./21.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i])));
         mua2_32_index[i]=(32./105.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i])));
         mua2_64_index[i]=(64./245.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i])));
         mua4_9_index[i]          =          (9./25.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i]*g_index[i]*g_index[i])));
         mua4_54_index[i]=(54./175.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i]*g_index[i]*g_index[i])));
         mua4_324_index[i]=(324./1225.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i]*g_index[i]*g_index[i])));
         mua6_13_index[i]          =          (13./49.)*((mua_index[i]+mus_index[i])-
(mus_index[i]*(g_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i])));
         kappa1_index[i] = (1./(3.*((mua_index[i]+mus_index[i])-(mus_index[i]*g_index[i]))));
         kappa3_index[i]          =          (1./(7.*((mua_index[i]+mus_index[i])-
(mus_index[i]*g_index[i]*g_index[i]*g_index[i]))));
         kappa5_index[i]          =          (1./(11.*((mua_index[i]+mus_index[i])-
(mus_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i]))));
         kappa7_index[i]          =          (1./(15.*((mua_index[i]+mus_index[i])-
(mus_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i]*g_index[i]))));

c_index[i] = *omega / *(c+(index[i]-1)) * -1;

         for (j=0; j<noden; ++j){
            tri_vtx[i][j] = *(nodes+(index[i]-1+(j*nodem)));
         }

      }

       /* Apply shape functions to each variable*/
      gradphi(tri_vtx,kappa1_index,k1_val);
      gradphi(tri_vtx,kappa3_index,k3_val);
      gradphi(tri_vtx,kappa5_index,k5_val);
         gradphi(tri_vtx,kappa7_index,k7_val);
      phidotphi(tri_vtx,mua_index,c_val);
      phidotphi(tri_vtx,mua23_index,c23_val);
      phidotphi(tri_vtx,mua49_index,c49_val);
      phidotphi(tri_vtx,mua64_index,c64_val);
      phidotphi(tri_vtx,mua8_index,c8_val);
      phidotphi(tri_vtx,mua16_index,c16_val);
```

```
    phidotphi(tri_vtx,mua32_index,c32_val);
    phidotphi(tri_vtx,mua128_index,c128_val);
    phidotphi(tri_vtx,mua256_index,c256_val);
    phidotphi(tri_vtx,mua2_59_index,c2_59_val);
    phidotphi(tri_vtx,mua2_16_index,c2_16_val);
    phidotphi(tri_vtx,mua2_49_index,c2_49_val);
    phidotphi(tri_vtx,mua2_8_index,c2_8_val);
    phidotphi(tri_vtx,mua2_32_index,c2_32_val);
    phidotphi(tri_vtx,mua2_64_index,c2_64_val);
    phidotphi(tri_vtx,mua4_9_index,c4_9_val);
    phidotphi(tri_vtx,mua4_54_index,c4_54_val);
    phidotphi(tri_vtx,mua4_324_index,c4_324_val);
    phidotphi(tri_vtx,mua6_13_index,c6_13_val);
    phidotphi(tri_vtx,c_index,b_val);

/* Add element contribution to Mass matrix */

    for (i=0; i<3; ++i){
      for (j=0; j<3; ++j){
        I[k] = index[i];
        J[k] = index[j];
        K1[k] = k1_val[j][i];
        K3[k] = k3_val[j][i];
        K5[k] = k5_val[j][i];
        K7[k] = k7_val[j][i];
        C[k] = c_val[j][i];
        C23[k] = c23_val[j][i];
        C49[k] = c49_val[j][i];
        C64[k] = c64_val[j][i];
        C8[k] = c8_val[j][i];
        C16[k] = c16_val[j][i];
        C32[k] = c32_val[j][i];
        C128[k] = c128_val[j][i];
        C256[k] = c256_val[j][i];
        C2_59[k] = c2_59_val[j][i];
        C2_16[k] = c2_16_val[j][i];
        C2_49[k] = c2_49_val[j][i];
        C2_8[k] = c2_8_val[j][i];
        C2_32[k] = c2_32_val[j][i];
        C2_64[k] = c2_64_val[j][i];
        C4_9[k] = c4_9_val[j][i];
        C4_54[k] = c4_54_val[j][i];
        C4_324[k] = c4_324_val[j][i];
        C6_13[k] = c6_13_val[j][i];
        B[k]= b_val[j][i];
        ++k;
      }
    }
  }

/* Calculate contribution to boundary conditions for each element */
  k = 0;
  for (ele=0; ele<elemm; ++ele){
    for (i=0; i<elemn; ++i){
      indextmp = *(elements+(ele+(i*elemm)));
      index[i] = indextmp;
    }
    for (i=0; i<elemn; ++i){
      bnd_index[i] = *(bndvtx+(index[i]-1));
      f1_index[i] = *(f1+(index[i]-1));
      f2_index[i] = *(f2+(index[i]-1));
      f3_index[i] = *(f3+(index[i]-1));
      f4_index[i] = *(f4+(index[i]-1));
```

```
            g1_index[i] = *(g1+(index[i]-1));
            g2_index[i] = *(g2+(index[i]-1));
            g3_index[i] = *(g3+(index[i]-1));
            g4_index[i] = *(g4+(index[i]-1));
            h1_index[i] = *(h1+(index[i]-1));
            h2_index[i] = *(h2+(index[i]-1));
            h3_index[i] = *(h3+(index[i]-1));
            h4_index[i] = *(h4+(index[i]-1));
            i1_index[i] = *(i1+(index[i]-1));
            i2_index[i] = *(i2+(index[i]-1));
            i3_index[i] = *(i3+(index[i]-1));
            i4_index[i] = *(i4+(index[i]-1));

            for (j=0; j<noden; ++j){
                tri_vtx[i][j] = *(nodes+(index[i]-1+(j*nodem)));
            }
        }
        if (bnd_index[0]+bnd_index[1]+bnd_index[2] != 0){

            boundary_int(tri_vtx, bnd_index, f1_index, f1_valr);
            boundary_int(tri_vtx, bnd_index, f2_index, f2_valr);
            boundary_int(tri_vtx, bnd_index, f3_index, f3_valr);
            boundary_int(tri_vtx, bnd_index, f4_index, f4_valr);
            boundary_int(tri_vtx, bnd_index, g1_index, g1_valr);
            boundary_int(tri_vtx, bnd_index, g2_index, g2_valr);
            boundary_int(tri_vtx, bnd_index, g3_index, g3_valr);
            boundary_int(tri_vtx, bnd_index, g4_index, g4_valr);
            boundary_int(tri_vtx, bnd_index, h1_index, h1_valr);
            boundary_int(tri_vtx, bnd_index, h2_index, h2_valr);
            boundary_int(tri_vtx, bnd_index, h3_index, h3_valr);
            boundary_int(tri_vtx, bnd_index, h4_index, h4_valr);
            boundary_int(tri_vtx, bnd_index, i1_index, i1_valr);
            boundary_int(tri_vtx, bnd_index, i2_index, i2_valr);
            boundary_int(tri_vtx, bnd_index, i3_index, i3_valr);
            boundary_int(tri_vtx, bnd_index, i4_index, i4_valr);

/* Add contribution to system Boundary Condition matrix depending on which nodes are on
boundary*/

            if (bnd_index[0]==1 && bnd_index[2]==1){
                int index1[2];
                index1[0] = index[0];
                index1[1] = index[2];
                for (i=0; i<2; ++i){
                    for (j=0; j<=i; ++j){
                        X[k] = index1[i];
                        Y[k] = index1[j];
                        F1[k] = f1_valr[i][j];
                        F2[k] = f2_valr[i][j];
                        F3[k] = f3_valr[i][j];
                        F4[k] = f4_valr[i][j];
                        G1[k] = g1_valr[i][j];
                        G2[k] = g2_valr[i][j];
                        G3[k] = g3_valr[i][j];
                        G4[k] = g4_valr[i][j];
                        H1[k] = h1_valr[i][j];
                        H2[k] = h2_valr[i][j];
                        H3[k] = h3_valr[i][j];
                        H4[k] = h4_valr[i][j];
                        I1[k] = i1_valr[i][j];
                        I2[k] = i2_valr[i][j];
                        I3[k] = i3_valr[i][j];
                        I4[k] = i4_valr[i][j];
                    ++k;
                }
            }
        }
        else if (bnd_index[0]==1 && bnd_index[1]==1){
            int index1[2];
            index1[0] = index[0];
            index1[1] = index[1];
            for (i=0; i<2; ++i){
                for (j=0; j<=i; ++j){
                    X[k] = index1[i];
                    Y[k] = index1[j];
                    F1[k] = f1_valr[i][j];
                    F2[k] = f2_valr[i][j];
                    F3[k] = f3_valr[i][j];
                    F4[k] = f4_valr[i][j];
                    G1[k] = g1_valr[i][j];
                    G2[k] = g2_valr[i][j];
                    G3[k] = g3_valr[i][j];
                    G4[k] = g4_valr[i][j];
                    H1[k] = h1_valr[i][j];
                    H2[k] = h2_valr[i][j];
                    H3[k] = h3_valr[i][j];
                    H4[k] = h4_valr[i][j];
                    I1[k] = i1_valr[i][j];
                    I2[k] = i2_valr[i][j];
                    I3[k] = i3_valr[i][j];
                    I4[k] = i4_valr[i][j];
                    ++k;
                }
            }
        }
        else if (bnd_index[1]==1 && bnd_index[2]==1){
            int index1[2];
            index1[0] = index[1];
            index1[1] = index[2];
            for (i=0; i<2; ++i){
                for (j=0; j<=i; ++j){
                    X[k] = index1[i];
                    Y[k] = index1[j];
                    F1[k] = f1_valr[i][j];
                    F2[k] = f2_valr[i][j];
                    F3[k] = f3_valr[i][j];
                    F4[k] = f4_valr[i][j];
                    G1[k] = g1_valr[i][j];
                    G2[k] = g2_valr[i][j];
                    G3[k] = g3_valr[i][j];
                    G4[k] = g4_valr[i][j];
                    H1[k] = h1_valr[i][j];
                    H2[k] = h2_valr[i][j];
                    H3[k] = h3_valr[i][j];
                    H4[k] = h4_valr[i][j];
                    I1[k] = i1_valr[i][j];
                    I2[k] = i2_valr[i][j];
                    I3[k] = i3_valr[i][j];
                    I4[k] = i4_valr[i][j];
                    ++k;
                }
            }
        }
    }
}
```

```
return;
}

/* Shape function for diffusion terms */
double gradphi(double g[3][2], double kappa[3], double val[3][3])
{
   int ii, jj;
   static int L[2][3] = {{-1.0,1.0,0.0}, \
{-1.0,0.0,1.0}};
static double w[3] = {1.0/6.0,1.0/6.0,1.0/6.0};
static double ip[3][2] = {{1.0/2.0,0.0},{1.0/2.0,1.0/2.0},{0.0,1.0/2.0}};
double Jt[2][2], dJt, iJt[2][2], S[3], G[2][3], GG[3][3], tmp;
double x0 = g[0][0]; double x1 = g[1][0];
double x2 = g[2][0];
double y0 = g[0][1]; double y1 = g[1][1];
double y2 = g[2][1];
double k0 = kappa[0]; double k1 = kappa[1];
double k2 = kappa[2];

Jt[0][0] = L[0][0]*x0 + L[0][1]*x1 + L[0][2]*x2;
Jt[0][1] = L[0][0]*y0 + L[0][1]*y1 + L[0][2]*y2;
 Jt[1][0] = L[1][0]*x0 + L[1][1]*x1 + L[1][2]*x2;
Jt[1][1] = L[1][0]*y0 + L[1][1]*y1 + L[1][2]*y2;

dJt = (Jt[0][0]*Jt[1][1]) - (Jt[0][1]*Jt[1][0]);

iJt[0][0] = +(Jt[1][1])/dJt;
iJt[0][1] = -(Jt[0][1])/dJt;
iJt[1][0] = -(Jt[1][0])/dJt;
iJt[1][1] = +(Jt[0][0])/dJt;

dJt = sqrt(dJt*dJt);

G[0][0] = iJt[0][0]*L[0][0] + iJt[0][1]*L[1][0];
G[0][1] = iJt[0][0]*L[0][1] + iJt[0][1]*L[1][1];
G[0][2] = iJt[0][0]*L[0][2] + iJt[0][1]*L[1][2];
G[1][0] = iJt[1][0]*L[0][0] + iJt[1][1]*L[1][0];
G[1][1] = iJt[1][0]*L[0][1] + iJt[1][1]*L[1][1];
G[1][2] = iJt[1][0]*L[0][2] + iJt[1][1]*L[1][2];

GG[0][0] = G[0][0]*G[0][0] + G[1][0]*G[1][0];
GG[0][1] = G[0][0]*G[0][1] + G[1][0]*G[1][1];
GG[0][2] = G[0][0]*G[0][2] + G[1][0]*G[1][2];
GG[1][0] = G[0][1]*G[0][0] + G[1][1]*G[1][0];
GG[1][1] = G[0][1]*G[0][1] + G[1][1]*G[1][1];
GG[1][2] = G[0][1]*G[0][2] + G[1][1]*G[1][2];
GG[2][0] = G[0][2]*G[0][0] + G[1][2]*G[1][0];
GG[2][1] = G[0][2]*G[0][1] + G[1][2]*G[1][1];
GG[2][2] = G[0][2]*G[0][2] + G[1][2]*G[1][2];

for (ii=0; ii<3; ++ii){
   for (jj=0; jj<3; ++jj){
      val[ii][jj]=0;
   }
}

for (ii=0; ii<3; ++ii){
   S[0] = 1-ip[ii][0]-ip[ii][1];
   S[1] = ip[ii][0];
   S[2] = ip[ii][1];
   tmp = k0*S[0] + k1*S[1] + k2*S[2];
```

```
    tmp = tmp*w[ii];
    val[0][0] += GG[0][0]*tmp;
    val[0][1] += GG[1][0]*tmp;
    val[0][2] += GG[2][0]*tmp;
    val[1][0] += GG[0][1]*tmp;
    val[1][1] += GG[1][1]*tmp;
    val[1][2] += GG[2][1]*tmp;
    val[2][0] += GG[0][2]*tmp;
    val[2][1] += GG[1][2]*tmp;
    val[2][2] += GG[2][2]*tmp;
}
for (ii=0; ii<3; ++ii){
    for (jj=0; jj<3; ++jj){
        val[ii][jj] = val[ii][jj]*dJt;
    }
}

return;
}

/* Shape function for absorption terms */
double phidotphi(double g[3][2], double mua[3], double val[3][3])
{
    int ii, jj;
    static int L[2][3] = {{-1.0,1.0,0.0}, \
{-1.0,0.0,1.0}};
static double w[3] = {1.0/6.0,1.0/6.0,1.0/6.0};
static double ip[3][2] = {{1.0/2.0,0.0},{1.0/2.0,1.0/2.0},{0.0,1.0/2.0}};
double Jt[2][2], dJt, S[3], tmp;
double x0 = g[0][0]; double x1 = g[1][0];
double x2 = g[2][0];
double y0 = g[0][1]; double y1 = g[1][1];
double y2 = g[2][1];
double k0 = mua[0]; double k1 = mua[1];
double k2 = mua[2];

Jt[0][0] = L[0][0]*x0 + L[0][1]*x1 + L[0][2]*x2;
Jt[0][1] = L[0][0]*y0 + L[0][1]*y1 + L[0][2]*y2;
Jt[1][0] = L[1][0]*x0 + L[1][1]*x1 + L[1][2]*x2;
Jt[1][1] = L[1][0]*y0 + L[1][1]*y1 + L[1][2]*y2;

dJt = (Jt[0][0]*Jt[1][1]) - (Jt[0][1]*Jt[1][0]);

dJt = sqrt(dJt*dJt);

for (ii=0; ii<3; ++ii){
    for (jj=0; jj<3; ++jj){
        val[ii][jj]=0;
    }
}

for (ii=0; ii<3; ++ii){
    S[0] = 1-ip[ii][0]-ip[ii][1];
    S[1] = ip[ii][0];
    S[2] = ip[ii][1];
    tmp = k0*S[0] + k1*S[1] + k2*S[2];
    tmp = tmp*w[ii];
    val[0][0] += S[0]*S[0]*tmp;
    val[0][1] += S[1]*S[0]*tmp;
    val[0][2] += S[2]*S[0]*tmp;
    val[1][0] += S[0]*S[1]*tmp;
    val[1][1] += S[1]*S[1]*tmp;
    val[1][2] += S[2]*S[1]*tmp;
    val[2][0] += S[0]*S[2]*tmp;
    val[2][1] += S[1]*S[2]*tmp;
    val[2][2] += S[2]*S[2]*tmp;
}
for (ii=0; ii<3; ++ii){
    for (jj=0; jj<3; ++jj){
        val[ii][jj] = val[ii][jj]*dJt;
    }
}
return;
}


double boundary_int(double gg[3][2], double bnd_index[3], double ksir[3], \
double valr[2][2])
{
    double ksirtmp[2];
    int i, j;
    double g[2][2], val;

    for (i=0; i<2; ++i){
        for (j=0; j<2; ++j){
            valr[i][j]=0;
        }
    }

    if (bnd_index[0]==1 && bnd_index[2]==1){
        ksirtmp[0] = ksir[0];
        ksirtmp[1] = ksir[2];
        g[0][0] = gg[0][0];
        g[0][1] = gg[0][1];
        g[1][0] = gg[2][0];
        g[1][1] = gg[2][1];
        for (i=0; i<2; ++i){
            for (j=0; j<=i; ++j){
                bound2(g, i, j, ksirtmp, &val);
                valr[i][j] = val;
            }
        }
    }
    else if (bnd_index[0]==1 && bnd_index[1]==1){
        ksirtmp[0] = ksir[0];
        ksirtmp[1] = ksir[1];
        g[0][0] = gg[0][0];
        g[0][1] = gg[0][1];
        g[1][0] = gg[1][0];
        g[1][1] = gg[1][1];

        for (i=0; i<2; ++i){
            for (j=0; j<=i; ++j){
                bound2(g, i, j, ksirtmp, &val);
                valr[i][j] = val;
            }
        }
    }
    else if (bnd_index[1]==1 && bnd_index[2]==1){
        ksirtmp[0] = ksir[1];
        ksirtmp[1] = ksir[2];
        g[0][0] = gg[1][0];
        g[0][1] = gg[1][1];
        g[1][0] = gg[2][0];
        g[1][1] = gg[2][1];
```

```
            for (i=0; i<2; ++i){
                for (j=0; j<=i; ++j){
                    bound2(g, i, j, ksirtmp, &val);
                    valr[i][j] = val;
                }
            }
        }
    return;
}
double bound2(double g[2][2], int il, int im, double ksir[2], \
double *val)
{
    static double w[2] = {1.0/2.0,1.0/2.0};
    static double ip[2] = {0.21132486540519,0.78867513459481};
    double dJt, S[2], tmp, tmpval;
    int ii;
    double x0 = g[0][0]; double x1 = g[1][0];
    double y0 = g[0][1]; double y1 = g[1][1];
    double k0 = ksir[0]; double k1 = ksir[1];

    dJt = sqrt(((g[1][0]-g[0][0])*(g[1][0]-g[0][0])) +                                      \
    ((g[1][1]-g[0][1])*(g[1][1]-g[0][1])));


    tmpval = 0.0;
    for (ii=0; ii<2; ++ii){
        S[0] = 1-ip[ii];
        S[1] = ip[ii];
        tmp = k0*S[0] + k1*S[1];
        tmp = tmp*w[ii];
        tmpval += S[il]*S[im]*tmp;
    }
    *val = tmpval*dJt;
}
/* -------- Gate-way to matlab  ------------ */

void mexFunction(int nlhs,
mxArray *plhs[],
int nrhs,
const mxArray *prhs[])

{
    double *nodes,*elements,*bndvtx,*mua, *mus, *g;
    double *f1,*f2,*f3,*f4,*g1,*g2,*g3,*g4,*h1,*h2,*h3,*h4,*i1,*i2,*i3,*i4;
    double *c,*omega;
    int nodem,noden,elemm,elemn,nzmax;
    double *I,*J,*K1,*K3,*K5,*K7, *B, *X, *Y;
    double *C,*C23, *C49,*C64,*C8,*C32,*C128,*C256;
    double *C2_16, *C2_49, *C16,*C2_59,*C2_8,*C2_32,*C2_64;
    double *C4_9,*C4_54, *C4_324,*C6_13;
    double *F1,*F2,*F3,*F4,*G1,*G2,*G3,*G4,*H1,*H2,*H3,*H4,*I1,*I2,*I3,*I4;

  /* Error checking  */

    if (nrhs < 24 )
        mexErrMsgTxt(" There are not enough input arguments");

    if(nlhs!=44)
        mexErrMsgTxt("This routine requires 44 ouput arguments");

/* define input parameters */
    nodes=mxGetPr(prhs[0]);         /*  nodes of mesh */
    elements=mxGetPr(prhs[1]);      /*  elements of mesh */
    bndvtx=mxGetPr(prhs[2]);        /*  boundary nodes */
    mua=mxGetPr(prhs[3]);           /*  absorption, nodal */
    mus=mxGetPr(prhs[4]);         /*  scattering*/
    g=mxGetPr(prhs[5]);          /*  anisotropy */
    f1=mxGetPr(prhs[6]);          /*  reflection parameters, nodal  */
    f2=mxGetPr(prhs[7]);
    f3=mxGetPr(prhs[8]);
    f4=mxGetPr(prhs[9]);
    g1=mxGetPr(prhs[10]);
    g2=mxGetPr(prhs[11]);
    g3=mxGetPr(prhs[12]);
    g4=mxGetPr(prhs[13]);
    h1=mxGetPr(prhs[14]);
    h2=mxGetPr(prhs[15]);
    h3=mxGetPr(prhs[16]);
    h4=mxGetPr(prhs[17]);
    i1=mxGetPr(prhs[18]);
    i2=mxGetPr(prhs[19]);
    i3=mxGetPr(prhs[20]);
    i4=mxGetPr(prhs[21]);
    c=mxGetPr(prhs[22]);            /*  speed of light in tissue, nodal */
    omega=mxGetPr(prhs[23]);         /*  modulation frequency */


    nodem=mxGetM(prhs[0]);          /*  Number of of nodes */
    noden=mxGetN(prhs[0]);         /*  Number of node freedom */
    elemm=mxGetM(prhs[1]);          /*  Number of elements */
    elemn=mxGetN(prhs[1]);         /*  Number of nodes per elements */


    nzmax = elemm*(elemn*(elemn+1));

/*define output parameters */
    plhs[0]=mxCreateDoubleMatrix(nzmax,1,mxREAL);    /* i */
    plhs[1]=mxCreateDoubleMatrix(nzmax,1,mxREAL);    /* j */
    plhs[2]=mxCreateDoubleMatrix(nzmax,1,mxREAL);    /* Diffusion terms */
    plhs[3]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[4]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[5]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[6]=mxCreateDoubleMatrix(nzmax,1,mxREAL);    /* Absorption terms */
    plhs[7]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[8]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[9]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[10]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[11]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[12]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[13]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[14]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[15]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[16]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[17]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[18]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[19]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[20]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[21]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[22]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[23]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[24]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[25]=mxCreateDoubleMatrix(nzmax,1,mxCOMPLEX);
    plhs[26]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[27]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[28]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[29]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
```

```
    plhs[30]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[31]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[32]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[33]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[34]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[35]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[36]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[37]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[38]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[39]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[40]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[41]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[42]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
    plhs[43]=mxCreateDoubleMatrix(nzmax,1,mxREAL);


    I=mxGetPr(plhs[0]);
    J=mxGetPr(plhs[1]);
    K1=mxGetPr(plhs[2]);
    K3=mxGetPr(plhs[3]);
    K5=mxGetPr(plhs[4]);
    K7=mxGetPr(plhs[5]);
    C=mxGetPr(plhs[6]);
    C23=mxGetPr(plhs[7]);
    C49=mxGetPr(plhs[8]);
    C64=mxGetPr(plhs[9]);
    C8=mxGetPr(plhs[10]);
    C16=mxGetPr(plhs[11]);
    C32=mxGetPr(plhs[12]);
    C128=mxGetPr(plhs[13]);
    C256=mxGetPr(plhs[14]);
    C2_59= mxGetPr(plhs[15]);
    C2_16=mxGetPr(plhs[16]);
    C2_49=mxGetPr(plhs[17]);
    C2_8=mxGetPr(plhs[18]);
    C2_32=mxGetPr(plhs[19]);
    C2_64=mxGetPr(plhs[20]);
    C4_9=mxGetPr(plhs[21]);
    C4_54=mxGetPr(plhs[22]);
    C4_324=mxGetPr(plhs[23]);
    C6_13=mxGetPr(plhs[24]);
    B=mxGetPi(plhs[25]);
    F1=mxGetPr(plhs[26]);
    F2=mxGetPr(plhs[27]);
    F3=mxGetPr(plhs[28]);
    F4=mxGetPr(plhs[29]);
    G1=mxGetPr(plhs[30]);
    G2=mxGetPr(plhs[31]);
    G3=mxGetPr(plhs[32]);
    G4=mxGetPr(plhs[33]);
    H1=mxGetPr(plhs[34]);
    H2=mxGetPr(plhs[35]);
    H3=mxGetPr(plhs[36]);
    H4=mxGetPr(plhs[37]);
    I1=mxGetPr(plhs[38]);
    I2=mxGetPr(plhs[39]);
    I3=mxGetPr(plhs[40]);
    I4=mxGetPr(plhs[41]);
    X=mxGetPr(plhs[42]);
    Y=mxGetPr(plhs[43]);

mainloop(nodes,elements,bndvtx,mua,mus,g,f1,f2,f3,f4,g1,g2,g3,g4,h1,h2,h3,h4,i1,i2,i3,i4,c,om
ega,nodem, \

noden,elemm,elemn,I,J,K1,K3,K5,K7,C,C23,C49,C64,C8,C16,C32,C128,C256,C2_59,C2_16,C2_
49,C2_8,C2_32,C2_64,C4_9, \
   C4_54,C4_324,C6_13,B,F1,F2,F3,F4,G1,G2,G3,G4,H1,H2,H3,H4,I1,I2,I3,I4,X,Y);

   return;
}
```

---

num_int.c

calculates orders of reflection by numerically integrating equation 7.16. Implemented in c for efficiency and compiled within MATLAB

```
#include <stdio.h>
#include <math.h>
#include <mex.h>


/* -------- Heart of the mex file----------- */
void mainloop(int nnodes,
double *nm,
double *n0,
double *Ca,
double *Rn,
double *I,
double *R,
double *bndvtx)
{
   int N, NN, j, w, k,l,b;
   double m, Ca_tmp;

   /*      initialise k*/
   k = 0;

   for (N=0; N<15; ++N){
      /* initialise I*/
      for (NN=0; NN<nnodes; ++NN){
         *(I+NN) = 0;
      }


      for (j=0; j<nnodes; ++j){
         b= *(bndvtx+j);
          if (b==1){
            for(w=0; w<1999; ++w){
               m= (w+1)/2000.0;

              /* If angle is less than critical angle, total internal reflectance */
               Ca_tmp = *(Ca+j);
               if (m < Ca_tmp){
                  R[j]=1.0;
               }
               else {

                 /* Otherwise calculate reflectance  by integrating equation 7.18*/

                  R[j]=0.5*pow((((nm[j]*sqrt(1-pow(nm[j],2)*  \
                  (1-pow(m,2))))-(n0[j]*m))/((nm[j]*sqrt(1-pow(nm[j],2)*\
                  (1-pow(m,2))))+(n0[j]*m))),2)+ 0.5*pow((((nm[j]*m)-  \
                  (n0[j]*sqrt(1-pow(nm[j],2)*(1-pow(m,2)))))/((nm[j]*m)+ \
                  (n0[j]*sqrt(1-pow(nm[j],2)*(1-pow(m,2)))))),2);
               }

               I[j]=I[j]+R[j]*pow(m,N)*0.0005;
            }

         }
         else{
            I[j]=0;
         }
      }

      for (l=0; l<nnodes; ++l){

         Rn[k]=I[l];
         ++k;
      }
   }
}



/* -------- Gate-way to matlab  ------------ */

void mexFunction(int nlhs,
mxArray *plhs[],
int nrhs,
const mxArray *prhs[])

{
   double *Rn, *nm, *n0,*Ca, *I, *R, *bndvtx;
   int nnodes, nzmax;


  /* Error checking  */

   if (nrhs < 4 )
      mexErrMsgTxt(" There are not enough input arguments");

   if(nlhs!=1)
      mexErrMsgTxt("This routine requires one ouput argument");

nm=mxGetPr(prhs[0]);
   n0=mxGetPr(prhs[1]);
   Ca=mxGetPr(prhs[2]);
   bndvtx=mxGetPr(prhs[3]);

   nnodes=mxGetM(prhs[0]);          /*  Number of of nodes */

   nzmax=nnodes*15;

   plhs[0]=mxCreateDoubleMatrix(nzmax,1,mxREAL);
   plhs[1]=mxCreateDoubleMatrix(nnodes,1,mxREAL);
   plhs[2]=mxCreateDoubleMatrix(nnodes,1,mxREAL);
   Rn=mxGetPr(plhs[0]);
   I=mxGetPr(plhs[1]);
   R=mxGetPr(plhs[2]);


   mainloop(nnodes,nm,n0,Ca,Rn,I,R, bndvtx);

   return;
}
```

```
function [ksi]=bound_coeffs(mesh,sp)

%Calculates boundary condition coefficients based on reflectivity.

nm=mesh.ri;
n0=ones(length(nm),1);
Ca=zeros(length(nm),1);
Ca(:)=cos(asin(n0./nm));     %w relating to Critical Angle


%only calculates integral once for each unique RI.
ri=unique(mesh.ri);

% Numerically integrate eq. 7.16
Rn1=num_int(ri,n0,Ca,ones(size(ri)));
Rn1=reshape(Rn1,length(ri),15);

Rn=zeros(length(nm),15);
for i=1:length(ri)
   ind=find(nm==ri(i));
   for j=1:15
       Rn(ind,j)=Rn1(i,j);
   end
end

Rn=reshape(Rn,length(nm),15);



% Calculate boundary coefficients as defined in Appendix A

if sp==1

   BC.A=-Rn(:,2);
   BC.B=3*Rn(:,3);

   ksi=[BC.A BC.B];

   clear BC

elseif sp==3

   % A1,A2
   BC.A(:,1)=-Rn(:,2);
   BC.A(:,2)=(-9/4)*Rn(:,2)+(15/2)*Rn(:,4)-(25/4)*Rn(:,6);

   %B1,B2
   BC.B(:,1)=3*Rn(:,3);
   BC.B(:,2)=(63/4)*Rn(:,3)-(105/2)*Rn(:,5)+(175/4)*Rn(:,7);

   %C1,C2
   BC.C(:,1)=(-3/2)*Rn(:,2)+(5/2)*Rn(:,4);
   BC.C(:,2)=(-3/2)*Rn(:,2)+(5/2)*Rn(:,4);

   %D1,D2
   BC.D(:,1)=(3/2)*Rn(:,3)-(5/2)*Rn(:,5);
   BC.D(:,2)=(3/2)*Rn(:,3)-(5/2)*Rn(:,5);

   ksi = [BC.A BC.B BC.C BC.D];

   clear BC

elseif sp==5
```

```
   %A1, A2, A3
   BC.A(:,1)=-Rn(:,2);
   BC.A(:,2)=(-9/4)*Rn(:,2)+(15/2)*Rn(:,4)-(25/4)*Rn(:,6);
   BC.A(:,3)=(-225/64)*Rn(:,2)+(525/16)*Rn(:,4)-(3395/32)*Rn(:,6)+...
      (2205/16)*Rn(:,8)-(3969/64)*Rn(:,10);

   %B1, B2, B3
   BC.B(:,1)=3*Rn(:,3);
   BC.B(:,2)=(63/4)*Rn(:,3)-(105/2)*Rn(:,5)+(175/4)*Rn(:,7);
   BC.B(:,3)=(2475/64)*Rn(:,3)-(5775/16)*Rn(:,5)+(37345/32)*Rn(:,7)-...
      (24255/16)*Rn(:,9)+(43659/64)*Rn(:,11);

   %C1, C2,C3
   BC.C(:,1)=(-3/2)*Rn(:,2)+(5/2)*Rn(:,4);
   BC.C(:,2)=(-3/2)*Rn(:,2)+(5/2)*Rn(:,4);
   BC.C(:,3)=(15/8)*Rn(:,2)-(35/4)*Rn(:,4)+(63/8)*Rn(:,6);

   %D1,D2,D3
   BC.D(:,1)=(3/2)*Rn(:,3)-(5/2)*Rn(:,5);
   BC.D(:,2)=(3/2)*Rn(:,3)-(5/2)*Rn(:,5);
   BC.D(:,3)=(-15/8)*Rn(:,3)+(35/4)*Rn(:,5)-(63/8)*Rn(:,7);

   %E1,E2,E3
   BC.E(:,1)=(15/8)*Rn(:,2)-(35/4)*Rn(:,4)+(63/8)*Rn(:,6);
   BC.E(:,2)=(-45/16)*Rn(:,2)+(285/16)*Rn(:,4)-(539/16)*Rn(:,6)+(315/16)*Rn(:,8);
   BC.E(:,3)=(-45/16)*Rn(:,2)+(285/16)*Rn(:,4)-(539/16)*Rn(:,6)+(315/16)*Rn(:,8);

   %F1,F2,F3
   BC.F(:,1)=(-15/8)*Rn(:,3)+(35/4)*Rn(:,5)-(63/8)*Rn(:,7);
   BC.F(:,2)=(45/16)*Rn(:,3)-(285/16)*Rn(:,5)+(539/16)*Rn(:,7)-(315/16)*Rn(:,9);
   BC.F(:,3)=(45/16)*Rn(:,3)-(285/16)*Rn(:,5)+(539/16)*Rn(:,7)-(315/16)*Rn(:,9);

   ksi=[BC.A BC.B BC.C BC.D BC.E BC.F];

   clear BC

elseif sp==7
   %A1,A2,A3,A4
   BC.A(:,1)=-Rn(:,2);
   BC.A(:,2)=(-9/4)*Rn(:,2)+(15/2)*Rn(:,4)-(25/4)*Rn(:,6);
   BC.A(:,3)=(-225/64)*Rn(:,2)+(525/16)*Rn(:,4)-(3395/32)*Rn(:,6)+...
      (2205/16)*Rn(:,8)-(3969/64)*Rn(:,10);
   BC.A(:,4)=(-1225/256)*Rn(:,2)+(11025/128)*Rn(:,4)-(147735/256)*Rn(:,6)+...
    (116655/64)*Rn(:,8)-(750519/256)*Rn(:,10)+(297297/128)*Rn(:,12)-...
    (184041/256)*Rn(:,14);

   %B1, B2, B3,B4
   BC.B(:,1)=3*Rn(:,3);
   BC.B(:,2)=(63/4)*Rn(:,3)-(105/2)*Rn(:,5)+(175/4)*Rn(:,7);
   BC.B(:,3)=(2475/64)*Rn(:,3)-(5775/16)*Rn(:,5)+(37345/32)*Rn(:,7)-...
      (24255/16)*Rn(:,9)+(43659/64)*Rn(:,11);
   BC.B(:,4)=15*((1225/256)*Rn(:,3)-(11025/128)*Rn(:,5)+(147735/256)*Rn(:,7)-...
    (116655/64)*Rn(:,9)+(750519/256)*Rn(:,11)-(297297/128)*Rn(:,13)+(184041/256)*Rn(:,15));

   %C1, C2,C3,C4
   BC.C(:,1)=(-3/2)*Rn(:,2)+(5/2)*Rn(:,4);
   BC.C(:,2)=(-3/2)*Rn(:,2)+(5/2)*Rn(:,4);
   BC.C(:,3)=(15/8)*Rn(:,2)-(35/4)*Rn(:,4)+(63/8)*Rn(:,6);
   BC.C(:,4)=(-35/16)*Rn(:,2)+(315/16)*Rn(:,4)-(693/16)*Rn(:,6)+(429/16)*Rn(:,8);

   %D1,D2,D3,D4
   BC.D(:,1)=(3/2)*Rn(:,3)-(5/2)*Rn(:,5);
   BC.D(:,2)=(3/2)*Rn(:,3)-(5/2)*Rn(:,5);
   BC.D(:,3)=(-15/8)*Rn(:,3)+(35/4)*Rn(:,5)-(63/8)*Rn(:,7);
   BC.D(:,4)=(35/16)*Rn(:,3)-(315/16)*Rn(:,5)+(693/16)*Rn(:,7)-(429/16)*Rn(:,9);

   %E1,E2,E3,E4
   BC.E(:,1)=(15/8)*Rn(:,2)-(35/4)*Rn(:,4)+(63/8)*Rn(:,6);
   BC.E(:,2)=(-45/16)*Rn(:,2)+(285/16)*Rn(:,4)-(539/16)*Rn(:,6)+(315/16)*Rn(:,8);
   BC.E(:,3)=(-45/16)*Rn(:,2)+(285/16)*Rn(:,4)-(539/16)*Rn(:,6)+(315/16)*Rn(:,8);
   BC.E(:,4)=(105/32)*Rn(:,2)-35*Rn(:,4)+(1827/16)*Rn(:,6)-(297/2)*Rn(:,8)+(2145/32)*Rn(:,10);

   %F1,F2,F3,F4
   BC.F(:,1)=(-15/8)*Rn(:,3)+(35/4)*Rn(:,5)-(63/8)*Rn(:,7);
   BC.F(:,2)=(45/16)*Rn(:,3)-(285/16)*Rn(:,5)+(539/16)*Rn(:,7)-(315/16)*Rn(:,9);
   BC.F(:,3)=(45/16)*Rn(:,3)-(285/16)*Rn(:,5)+(539/16)*Rn(:,7)-(315/16)*Rn(:,9);
   BC.F(:,4)=(-105/32)*Rn(:,3)+35*Rn(:,5)-(1827/16)*Rn(:,7)+(297/2)*Rn(:,9)-(2145/32)*Rn(:,11);

   %G1,G2,G3,G4
   BC.G(:,1)=(-35/16)*Rn(:,2)+(315/16)*Rn(:,4)-(693/16)*Rn(:,6)+(429/16)*Rn(:,8);
   BC.G(:,2)=(105/32)*Rn(:,2)-35*Rn(:,4)+(1827/16)*Rn(:,6)-(297/2)*Rn(:,8)+(2145/32)*Rn(:,10);
   BC.G(:,3)=(-525/128)*Rn(:,2)+(7175/128)*Rn(:,4)-(17325/64)*Rn(:,6)+(37395/64)*Rn(:,8)-...
   (73689/128)*Rn(:,10)+(27027/128)*Rn(:,12);
   BC.G(:,4)=(-525/128)*Rn(:,2)+(7175/128)*Rn(:,4)-(17325/64)*Rn(:,6)+(37395/64)*Rn(:,8)-...
   (73689/128)*Rn(:,10)+(27027/128)*Rn(:,12);

   %H1,H2,H3,H4
   BC.H(:,1)=(35/16)*Rn(:,3)-(315/16)*Rn(:,5)+(693/16)*Rn(:,7)-(429/16)*Rn(:,9);
   BC.H(:,2)=(-105/32)*Rn(:,3)+35*Rn(:,5)-(1827/16)*Rn(:,7)+(297/2)*Rn(:,9)-
(2145/32)*Rn(:,11);
   BC.H(:,3)=(525/128)*Rn(:,3)-(7175/128)*Rn(:,5)+(17325/64)*Rn(:,7)-(37395/64)*Rn(:,9)+...
      (73689/128)*Rn(:,11)-(27027/128)*Rn(:,13);
   BC.H(:,4)=(525/128)*Rn(:,3)-(7175/128)*Rn(:,5)+(17325/64)*Rn(:,7)-(37395/64)*Rn(:,9)+...
      (73689/128)*Rn(:,11)-(27027/128)*Rn(:,13);

   ksi=[BC.A BC.B BC.C BC.D BC.E BC.F BC.G BC.H];

   clear BC
else
   ksi=[];
end
```

```
function [f1,f2,f3,f4,g1,g2,g3,g4,h1,h2,h3,h4,i1,i2,i3,i4] = ksi_calc_sp7(mesh);

%Calculates the boundary condition values for SP7 based on equations 28a-d
%of Klose 2006 paper.

% Calculate boundary coefficients
BC=bound_coeffs(mesh,7);

A=BC(:,1:4);
B=BC(:,5:8);
C=BC(:,9:12);
D=BC(:,13:16);
E=BC(:,17:20);
F=BC(:,21:24);
G=BC(:,25:28);
H=BC(:,29:32);

% Variables M to T are temporary variables to simplify input of excessive algebra
M1=(7/24+A(:,2))-3.*D(:,2).*((1+B(:,1)).\((1/8)+C(:,1)));
M2=(1+B(:,2))-21*D(:,2).*((1+B(:,1)).\D(:,1));
```

```
M3=((1/8)+C(:,2))-3*D(:,2).*((1+B(:,1)).\((1/2)+A(:,1)));
M4= ((41/384)+E(:,2))+3*D(:,2).*((1+B(:,1)).\((-1/16)+E(:,1)));
M5=11*F(:,2)+33*D(:,2).*((1+B(:,1)).\F(:,1));
M6=((-1/16)+G(:,2))+3*D(:,2).*((1+B(:,1)).\((5/128)+G(:,1)));
M7=15*H(:,2)+45*D(:,2).*((1+B(:,1)).\H(:,1));

N1=((407/1920)+A(:,3))-3*D(:,3).*((1+B(:,1)).\((-1/16)+E(:,1)));
N2=(1+B(:,3))-33*D(:,3).*((1+B(:,1)).\F(:,1));
N3= ((-1/16)+C(:,3))-3*D(:,3).*((1+B(:,1)).\((1/2)+A(:,1)));
N4= ((41/384)+E(:,3))+3*D(:,3).*((1+B(:,1)).\((1/8)+C(:,1)));
N5=7*F(:,3)+21*D(:,3).*((1+B(:,1)).\D(:,1));
N6=((233/2560)+G(:,3))+3*D(:,3).*((1+B(:,1)).\((5/128)+G(:,1)));
N7=15*H(:,3)+45*D(:,3).*((1+B(:,1)).\H(:,1));

P1=((3023/17920)+A(:,4))-3*D(:,4).*((1+B(:,1)).\((5/128)+G(:,1)));
P2=(1+B(:,4))-45*D(:,4).*((1+B(:,1)).\H(:,1));
P3=((5/128)+C(:,4))-3*D(:,4).*((1+B(:,1)).\((1/2)+A(:,1)));
P4=((-1/16)+E(:,4))+3*D(:,4).*((1+B(:,1)).\((1/8)+C(:,1)));
P5=7*F(:,4)+21*D(:,4).*((1+B(:,1)).\D(:,1));
P6=((233/2560)+G(:,4))+3*D(:,4).*((1+B(:,1)).\((-1/16)+E(:,1)));
P7=11*H(:,4)+33*D(:,4).*((1+B(:,1)).\F(:,1));

R1=M1-M5.*((N2-N5.*(M2.\M5)).\(N4-N5.*(M2.\M1)));
R2=M3+M5.*((N2-N5.*(M2.\M5)).\(N3+N5.*(M2.\M3)));
R3=M4-M5.*((N2-N5.*(M2.\M5)).\(N1-N5.*(M2.\M4)));
R4=M6+M5.*((N2-N5.*(M2.\M5)).\(N6+N5.*(M2.\M6)));
R5=M7+M5.*((N2-N5.*(M2.\M5)).\(N7+N5.*(M2.\M7)));

S1=((1/2)+A(:,1))-7*D(:,1).*(M2.\M3)-(11*F(:,1)+7.*D(:,1).*(M2.\M5)).*((N2-
N5.*(M2.\M5)).\(N3+N5.*(M2.\M3)));
S2=((1/8)+C(:,1))+7*D(:,1).*(M2.\M1)+(11*F(:,1)+7*D(:,1).*(M2.\M5)).*((N2-
N5.*(M2.\M5)).\(N4-N5.*(M2.\M1)));
S3=((-1/16)+E(:,1))+7*D(:,1).*(M2.\M4)-(11*F(:,1)+7*D(:,1).*(M2.\M5)).*((N2-
N5.*(M2.\M5)).\(N1-N5.*(M2.\M4)));
S4=((5/128)+G(:,1))+7*D(:,1).*(M2.\M6)+(11*F(:,1)+7*D(:,1).*(M2.\M5)).*((N2-
N5.*(M2.\M5)).\(N6+N5.*(M2.\M6)));
S5=15*H(:,1)+7*D(:,1).*(M2.\M7)+(11*F(:,1)+7*D(:,1).*(M2.\M5)).*((N2-
N5.*(M2.\M5)).\(N7+N5.*(M2.\M7)));

T1=P1-P5.*(M2.\M6)-(P7+P5.*(M2.\M5)).*((N2-N5.*(M2.\M5)).\(N6+N5.*(M2.\M6)));
T2=P2-P5.*(M2.\M7)-(P7+P5.*(M2.\M5)).*((N2-N5.*(M2.\M5)).\(N7+N5.*(M2.\M7)));
T3=P3+P5.*(M2.\M3)+(P7+P5.*(M2.\M5)).*((N2-N5.*(M2.\M5)).\(N3+N5.*(M2.\M3)));
T4=P4-P5.*(M2.\M1)+(P7+P5.*(M2.\M5)).*((N2-N5.*(M2.\M5)).\(N4-N5.*(M2.\M1)));
T5=P6+P5.*(M2.\M4)-(P7+P5.*(M2.\M5)).*((N2-N5.*(M2.\M5)).\(N1-N5.*(M2.\M4)));

f1= (1+B(:,1)).\(S1-S5.*(T2.\T3));
g1= -(1+B(:,1)).\(S2+S5.*(T2.\T4));
h1= -(1+B(:,1)).\(S3+S5.*(T2.\T5));
i1=-(1+B(:,1)).\(S4-S5.*(T2.\T1));

f2= M2.\(R1-R5.*(T2.\T4));
g2=M2.\(R2+R5.*(T2.\T3));
h2= M2.\(R3+R5.*(T2.\T5));
i2=M2.\(R4-R5.*(T2.\T1));

f3= (N2-N5.*(M2.\M5)).\(N1-N5.*(M2.\M4)-(N7+N5.*(M2.\M7)).*(T2.\T5));
g3= (N2-N5.*(M2.\M5)).\(N3+N5.*(M2.\M3)+(N7+N5.*(M2.\M7)).*(T2.\T3));
h3= (N2-N5.*(M2.\M5)).\(N4-N5.*(M2.\M1)+(N7+N5.*(M2.\M7)).*(T2.\T4));
i3= (N2-N5.*(M2.\M5)).\(N6+N5.*(M2.\M6)-(N7+N5.*(M2.\M7)).*(T2.\T1));

f4= T2.\T1;
g4=T2.\T3;
h4=T2.\T4;
```

```
i4=T2.\T5;

clear M* N* P* R* S* T*
```

```
function [J1,J2,J3,J4,data,mesh]=jacobian_sp7(mesh,frequency)

% [J1,J2,J3,J4,data,mesh]=jacobian_spn(fn,frequency)
% Calculates the Jacobian for each moment of fluence
% (both complex version and separate parts
% in terms of mus and mua for log amplitude and phase
% (radians). See any of Dartmouth Publications regarding the
% structure. Also calculates data (phase and amplitude) for a given
% problem (fn) at a given frequency (MHz).
% outputs phase and amplitude in structure data
% and mesh information in mesh

% If not a workspace variable, load mesh
if ischar(mesh)== 1
  mesh = load_mesh(mesh);
end

% modulation frequency
omega = 2*pi*frequency*1e6;
nvtx=length(mesh.nodes);

% Create FEM matrices
[f1,f2,f3,f4,g1,g2,g3,g4,h1,h2,h3,h4,i1,i2,i3,i4] = ksi_calc_sp7(mesh);

if mesh.dimension==2

 [i,j,k1,k3,k5,k7,c,c23,c49,c64,c8,c16,c32,c128,c256,c2_59,c2_16,c2_49,...
   c2_8,c2_32,c2_64,c4_9,c4_54,c4_324,c6_13,B,F1,F2,F3,F4,G1,G2,G3,G4,...
   H1,H2,H3,H4,I1,I2,I3,I4,ib,jb]=gen_matrices_2d_sp7(mesh.nodes(:,1:2),...
   sort(mesh.elements')',...
   mesh.bndvtx,...
   mesh.mua,...
   mesh.mus,...
   mesh.g,...
   f1,...
   f2,...
   f3,...
   f4,...
   g1,...
   g2,...
   g3,...
   g4,...
   h1,...
   h2,...
   h3,...
   h4,...
   i1,...
   i2,...
   i3,...
   i4,...
   mesh.c,...
   omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
```

```
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
K7= sparse(i(1:nz_i),j(1:nz_i),k7(1:nz_i));
C = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23 = sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49 = sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64 = sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16 = sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C32 = sparse(i(1:nz_i),j(1:nz_i),c32(1:nz_i));
C128 = sparse(i(1:nz_i),j(1:nz_i),c128(1:nz_i));
C256 = sparse(i(1:nz_i),j(1:nz_i),c256(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C2_8 = sparse(i(1:nz_i),j(1:nz_i),c2_8(1:nz_i));
C2_32 = sparse(i(1:nz_i),j(1:nz_i),c2_32(1:nz_i));
C2_64 = sparse(i(1:nz_i),j(1:nz_i),c2_64(1:nz_i));
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
C4_54 = sparse(i(1:nz_i),j(1:nz_i),c4_54(1:nz_i));
C4_324 = sparse(i(1:nz_i),j(1:nz_i),c4_324(1:nz_i));
C6_13 = sparse(i(1:nz_i),j(1:nz_i),c6_13(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));

[i,j]=size(C);
nz_i = length(nonzeros(ib));
F1 = sparse(ib(1:nz_i),jb(1:nz_i),F1(1:nz_i));
F2 = sparse(ib(1:nz_i),jb(1:nz_i),F2(1:nz_i));
F3 = sparse(ib(1:nz_i),jb(1:nz_i),F3(1:nz_i));
F4 = sparse(ib(1:nz_i),jb(1:nz_i),F4(1:nz_i));
G1 = sparse(ib(1:nz_i),jb(1:nz_i),G1(1:nz_i));
G2 = sparse(ib(1:nz_i),jb(1:nz_i),G2(1:nz_i));
G3 = sparse(ib(1:nz_i),jb(1:nz_i),G3(1:nz_i));
G4 = sparse(ib(1:nz_i),jb(1:nz_i),G4(1:nz_i));
H1 = sparse(ib(1:nz_i),jb(1:nz_i),H1(1:nz_i));
H2 = sparse(ib(1:nz_i),jb(1:nz_i),H2(1:nz_i));
H3 = sparse(ib(1:nz_i),jb(1:nz_i),H3(1:nz_i));
H4 = sparse(ib(1:nz_i),jb(1:nz_i),H4(1:nz_i));
I1 = sparse(ib(1:nz_i),jb(1:nz_i),I1(1:nz_i));
I2 = sparse(ib(1:nz_i),jb(1:nz_i),I2(1:nz_i));
I3 = sparse(ib(1:nz_i),jb(1:nz_i),I3(1:nz_i));
I4 = sparse(ib(1:nz_i),jb(1:nz_i),I4(1:nz_i));

elseif mesh.dimension==3
  [i,j,k1,k3,k5,k7,c,c23,c49,c64,c8,c16,c32,c128,c256,c2_59,c2_16,c2_49,...
    c2_8,c2_32,c2_64,c4_9,c4_54,c4_324,c6_13,B,F1,F2,F3,F4,G1,G2,G3,G4,...
    H1,H2,H3,H4,I1,I2,I3,I4,ib,jb]=gen_matrices_3d_sp7(mesh.nodes,...
    sort(mesh.elements')',...
    mesh.bndvtx,...
    mesh.mua,...
    mesh.mus,...
    mesh.g,...
    f1,...
    f2,...
    f3,...
    f4,...
    g1,...
    g2,...
    g3,...
    g4,...
    h1,...
    h2,...
    h3,...
    h4,...
  i1,...
  i2,...
  i3,...
  i4,...
  mesh.c,...
  omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
K7= sparse(i(1:nz_i),j(1:nz_i),k7(1:nz_i));
C = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23 = sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49 = sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64 = sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16 = sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C32 = sparse(i(1:nz_i),j(1:nz_i),c32(1:nz_i));
C128 = sparse(i(1:nz_i),j(1:nz_i),c128(1:nz_i));
C256 = sparse(i(1:nz_i),j(1:nz_i),c256(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C2_8 = sparse(i(1:nz_i),j(1:nz_i),c2_8(1:nz_i));
C2_32 = sparse(i(1:nz_i),j(1:nz_i),c2_32(1:nz_i));
C2_64 = sparse(i(1:nz_i),j(1:nz_i),c2_64(1:nz_i));
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
C4_54 = sparse(i(1:nz_i),j(1:nz_i),c4_54(1:nz_i));
C4_324 = sparse(i(1:nz_i),j(1:nz_i),c4_324(1:nz_i));
C6_13 = sparse(i(1:nz_i),j(1:nz_i),c6_13(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));
F1 = sparse(i(1:nz_i),j(1:nz_i),F1(1:nz_i));
F2 = sparse(i(1:nz_i),j(1:nz_i),F2(1:nz_i));
F3 = sparse(i(1:nz_i),j(1:nz_i),F3(1:nz_i));
F4 = sparse(i(1:nz_i),j(1:nz_i),F4(1:nz_i));
G1 = sparse(i(1:nz_i),j(1:nz_i),G1(1:nz_i));
G2 = sparse(i(1:nz_i),j(1:nz_i),G2(1:nz_i));
G3 = sparse(i(1:nz_i),j(1:nz_i),G3(1:nz_i));
G4 = sparse(i(1:nz_i),j(1:nz_i),G4(1:nz_i));
H1 = sparse(i(1:nz_i),j(1:nz_i),H1(1:nz_i));
H2 = sparse(i(1:nz_i),j(1:nz_i),H2(1:nz_i));
H3 = sparse(i(1:nz_i),j(1:nz_i),H3(1:nz_i));
H4 = sparse(i(1:nz_i),j(1:nz_i),H4(1:nz_i));
I1 = sparse(i(1:nz_i),j(1:nz_i),I1(1:nz_i));
I2 = sparse(i(1:nz_i),j(1:nz_i),I2(1:nz_i));
I3 = sparse(i(1:nz_i),j(1:nz_i),I3(1:nz_i));
I4 = sparse(i(1:nz_i),j(1:nz_i),I4(1:nz_i));

end

% Add complex contribution  to absorption terms to account for frequency dependance
C=C+B;
C23=C23+B;
C49=C49+B;
C64=C64+B;
C8=C8+B;
C16=C16+B;
C32=C32 +B;
C128 = C128+B;
C256=C256+B;
C2_59=C2_59+B;
C2_16=C2_16+B;
```

```
C2_49=C2_49+B;
C2_8=C2_8+B;
C2_32=C2_32+B;
C2_64=C2_64+B;
C4_9=C4_9+B;
C4_54 = C4_54+B;
C4_324 = C4_324+B;
C6_13 = C6_13+B;


clear i j k1 c f1 g1 ib jb nz_i omega

M1=K1+C+F1;
M2=K3+C49+C2_59+F2;
M3=K5+C64+C2_16+C4_9+F3;
M4=K7+C256+C2_64+C4_324+C6_13+F4;

% Calculate the RHS (the source vectors. For simplicity, we are
% just going to use a Gaussian Source, The width of the Gaussian is
% changeable (last argument). The source is assumed to have a
% complex amplitude of complex(cos(0.15),sin(0.15));

% Now calculate source vector
% NOTE last term in mex file 'qvec' is the source FWHM
%
[nnodes,junk]=size(mesh.nodes);
[nsource,junk]=size(mesh.source.coord);
qvec = spalloc(nnodes,nsource,nsource*100);
if mesh.dimension == 2
  for i = 1 : nsource
    if mesh.source.fwhm(i) == 0
        qvec(:,i) = gen_source_point(mesh,mesh.source.coord(i,1:2));
    else
      qvec(:,i) = gen_source(mesh.nodes(:,1:2),...
          sort(mesh.elements')',...
          mesh.dimension,...
          mesh.source.coord(i,1:2),...
          mesh.source.fwhm(i));
    end
  end
elseif mesh.dimension == 3
  for i = 1 : nsource
    if mesh.source.fwhm(i) == 0
        qvec(:,i) = gen_source_point(mesh,mesh.source.coord(i,1:3));
    else
    qvec(:,i) = gen_source(mesh.nodes,...
          sort(mesh.elements')',...
          mesh.dimension,...
          mesh.source.coord(i,:),...
          mesh.source.fwhm(i));
    end
  end
end

i=nnodes;

% Build source term
qvec=[qvec;(-2/3)*qvec;(8/15)*qvec;(-16/35)*qvec];

% Build mass matrix
MASS=[M1 -(C23-G1) +(C8-H1) -(C16-I1);...
    -(C23-G2) M2 -(C16-C2_49-H2) (C32+C2_8-I2);...
    (C8-G3) -(C16+C2_49-H3) M3 -(C128+C2_32+C4_54-I3);...
```

```
        (-C16-G4) (C32+C2_8-H4) -(C128+C2_32+C4_54-I4) M4];

% %=======================================
% % Optimise MASS matrix

[MASS_opt,Q_opt,invsort]=optimise(MASS,qvec);

% %========================================

[phi_all,R]=get_field(MASS_opt,mesh,Q_opt);

% Re-order elements
phi_all=phi_all(invsort,:);
data.phi1=phi_all(1:nvtx,:);
data.phi2=phi_all((nvtx+1):(2*nvtx),:);
data.phi3=phi_all(((2*nvtx)+1):3*nvtx,:);
data.phi4=phi_all(((3*nvtx)+1):4*nvtx,:);

% Build source term
data.phi = data.phi1 - (2/3).*data.phi2+...
      (8/15).*data.phi3-(16/35).*data.phi4;
clear phi*;

[qvec] = gen_source_adjoint(mesh);

% Adjoint source
qvec = [qvec;(-2/3)*qvec;(8/15)*qvec;(-16/35)*qvec];

[MASS_opt,qvec_opt,invsort]=optimise(MASS,qvec);

[phi]=get_field(conj(MASS_opt),mesh,conj(qvec_opt),R);
phi=phi(invsort,:);
data.aphi1=phi(1:nvtx,:);
data.aphi2=phi((nvtx+1):(2*nvtx),:);
data.aphi3=phi(((2*nvtx)+1):3*nvtx,:);
data.aphi4=phi(((3*nvtx)+1):4*nvtx,:);

% Calculate scalar flux from contributions of Phi1 Phi2 Phi3 Phi 4

data.aphi = data.aphi1 - (2/3).*data.aphi2+...
      (8/15).*data.aphi3-(16/35).*data.aphi4;
clear phi* qvec* R* MASS*;
% =======================================
%

% Calculate boundary data
[data.complex]=get_boundary_data(mesh,data.phi);
[data.complex1]=get_boundary_data(mesh,data.phi1);
[data.complex2]=get_boundary_data(mesh,data.phi2);
[data.complex3]=get_boundary_data(mesh,data.phi3);
[data.complex4]=get_boundary_data(mesh,data.phi4);

% Map complex data to amplitude and phase
% Phi 1
data.amplitude1 = abs(data.complex1);

data.phase1 = atan2(imag(data.complex1),...
          real(data.complex1));
data.phase1(find(data.phase1<0)) = data.phase1(find(data.phase1<0)) + (2*pi);
data.phase1 = data.phase1*180/pi;

data.paa1 = [data.amplitude1 data.phase1];
```

```
% Phi 2
data.amplitude2 = abs(data.complex2);

data.phase2 = atan2(imag(data.complex2),...
          real(data.complex2));
data.phase2(find(data.phase2<0)) = data.phase2(find(data.phase2<0)) + (2*pi);
data.phase2 = data.phase2*180/pi;

data.paa2 = [data.amplitude2 data.phase2];

% Phi 3
data.amplitude3 = abs(data.complex3);

data.phase3 = atan2(imag(data.complex3),...
          real(data.complex3));
data.phase3(find(data.phase3<0)) = data.phase3(find(data.phase3<0)) + (2*pi);
data.phase3 = data.phase3*180/pi;

data.paa3 = [data.amplitude3 data.phase3];

% Phi 4
data.amplitude4 = abs(data.complex4);

data.phase4 = atan2(imag(data.complex4),...
          real(data.complex4));
data.phase4(find(data.phase4<0)) = data.phase4(find(data.phase4<0)) + (2*pi);
data.phase4 = data.phase4*180/pi;

data.paa4 = [data.amplitude4 data.phase4];

% Total
data.amplitude = abs(data.complex);

data.phase = atan2(imag(data.complex),...
          real(data.complex));
data.phase(find(data.phase<0)) = data.phase(find(data.phase<0)) + (2*pi);
data.phase = data.phase*180/pi;

data.paa = [data.amplitude data.phase];

% Calculate Jacobian for each order of fluence and total fluence
% Catch zero frequency (CW) here
if frequency == 0
  [J] = build_jacobian_cw(mesh,data);
else
  data1.phi = data.phi1;
  data1.aphi = data.aphi1;
  data1.complex = data.complex1;
  [J1] = build_jacobian(mesh,data1);
  data1.phi = data.phi2;
  data1.aphi = data.aphi2;
  data1.complex = data.complex2;
  [J2] = build_jacobian(mesh,data1);
  data1.phi = data.phi3;
  data1.aphi = data.aphi3;
  data1.complex = data.complex3;
  [J3] = build_jacobian(mesh,data1);
 data1.phi = data.phi4;
  data1.aphi = data.aphi4;
  data1.complex = data.complex4;
  [J4] = build_jacobian(mesh,data1);
  clear data1
end
```

```
function [J,data,mesh]=jacobian_spn(mesh,frequency)

% [J,data,mesh]=jacobian_stnd(fn,frequency,param)
% Calculates a single Jacobian (both complex version and separate parts
% in terms of mus and mua for log amplitude and phase
% (radians). See any of Dartmouth Publications regarding the
% structure. Also calculates data (phase and amplitude) for a given
% problem (fn) at a given frequency (MHz).
% outputs phase and amplitude in structure data
% and mesh information in mesh
%
% As used in Section 8.7 - Diffusion based image recon
% using the SPN approximation

% If not a workspace variable, load mesh
if ischar(mesh)== 1
   mesh = load_mesh(mesh);
end

% modulation frequency
omega = 2*pi*frequency*1e6;
nvtx=length(mesh.nodes);

% Create FEM matricex
[f1,f2,f3,g1,g2,g3,h1,h2,h3] = ksi_calc_sp5(mesh);

if mesh.dimension==2
[i,j,k1,k3,k5,c,c23,c49,c64,c8,c16,c2_59,c2_16,c2_49,c4_9,B,F1,F2,F3,G1,G2,G3,H1,H2,H3,ib,jb]=
gen_matrices_2d_sp5(mesh.nodes(:,1:2),...
   sort(mesh.elements')',...
   mesh.bndvtx,...
   mesh.mua,...
   mesh.mus,...
   mesh.g,...
   f1,...
   f2,...
   f3,...
   g1,...
   g2,...
   g3,...
   h1,...
   h2,...
   h3,...
   mesh.c,...
   omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
C  = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23= sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49= sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64= sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16= sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
```

```
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));
nz_i = length(nonzeros(ib));
F1 = sparse(ib(1:nz_i),jb(1:nz_i),F1(1:nz_i));
F2 = sparse(ib(1:nz_i),jb(1:nz_i),F2(1:nz_i));
F3 = sparse(ib(1:nz_i),jb(1:nz_i),F3(1:nz_i));
G1 = sparse(ib(1:nz_i),jb(1:nz_i),G1(1:nz_i));
G2 = sparse(ib(1:nz_i),jb(1:nz_i),G2(1:nz_i));
G3 = sparse(ib(1:nz_i),jb(1:nz_i),G3(1:nz_i));
H1 = sparse(ib(1:nz_i),jb(1:nz_i),H1(1:nz_i));
H2 = sparse(ib(1:nz_i),jb(1:nz_i),H2(1:nz_i));
H3 = sparse(ib(1:nz_i),jb(1:nz_i),H3(1:nz_i));

elseif mesh.dimension==3
[i,j,k1,k3,k5,c,c23,c49,c64,c8,c16,c2_59,c2_16,c2_49,c4_9,B,F1,F2,F3,G1,G2,G3,H1,H2,H3,ib,jb]=
gen_matrices_3d_sp5(mesh.nodes,...
   sort(mesh.elements')',...
   mesh.bndvtx,...
   mesh.mua,...
   mesh.mus,...
   mesh.g,...
   f1,...
   f2,...
   f3,...
   g1,...
   g2,...
   g3,...
   h1,...
   h2,...
   h3,...
   mesh.c,...
   omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
C  = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23= sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49= sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64= sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16= sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));
F1 = sparse(i(1:nz_i),j(1:nz_i),F1(1:nz_i));
F2 = sparse(i(1:nz_i),j(1:nz_i),F2(1:nz_i));
F3 = sparse(i(1:nz_i),j(1:nz_i),F3(1:nz_i));
G1 = sparse(i(1:nz_i),j(1:nz_i),G1(1:nz_i));
G2 = sparse(i(1:nz_i),j(1:nz_i),G2(1:nz_i));
G3 = sparse(i(1:nz_i),j(1:nz_i),G3(1:nz_i));
H1 = sparse(i(1:nz_i),j(1:nz_i),H1(1:nz_i));
H2 = sparse(i(1:nz_i),j(1:nz_i),H2(1:nz_i));
H3 = sparse(i(1:nz_i),j(1:nz_i),H3(1:nz_i));

end

clear i j k1 c f1 g1 ib jb nz_i omega

% Add complex component to account for frequency dependence
```

```
C=C+B;
C23=C23+B;
C49=C49+B;
C64=C64+B;
C8=C8+B;
C16=C16+B;
C2_59=C2_59+B;
C2_16=C2_16+B;
C2_49=C2_49+B;
C4_9=C4_9+B;
Cs=C16+C2_49;


M1=K1+C+F1;
M2=K3+C49+C2_59+F2;
M3=K5+C64+C2_16+C4_9+F3;


% Calculate the RHS (the source vectors. For simplicity, we are
% just going to use a Gaussian Source, The width of the Gaussian is
% changeable (last argument). The source is assumed to have a
% complex amplitude of complex(cos(0.15),sin(0.15));

% Now calculate source vector
% NOTE last term in mex file 'qvec' is the source FWHM
%
[nnodes,junk]=size(mesh.nodes);
[nsource,junk]=size(mesh.source.coord);
qvec = spalloc(nnodes,nsource,nsource*100);
if mesh.dimension == 2
  for i = 1 : nsource
    if mesh.source.fwhm(i) == 0
      qvec(:,i) = gen_source_point(mesh,mesh.source.coord(i,1:2));
    else
      qvec(:,i) = gen_source(mesh.nodes(:,1:2),...
        sort(mesh.elements')',...
        mesh.dimension,...
        mesh.source.coord(i,1:2),...
        mesh.source.fwhm(i));
    end
  end
elseif mesh.dimension == 3
  for i = 1 : nsource
    if mesh.source.fwhm(i) == 0
      qvec(:,i) = gen_source_point(mesh,mesh.source.coord(i,1:3));
    else
      qvec(:,i) = gen_source(mesh.nodes,...
        sort(mesh.elements')',...
        mesh.dimension,...
        mesh.source.coord(i,:),...
        mesh.source.fwhm(i));
    end
  end
end

i=nnodes;

% Build source term
qvec=[qvec;(-2/3)*qvec;(8/15)*qvec];

% Build mass matrix
MASS=[M1 -(C23-G1) (C8-H1);-(C23-G2) M2 -(C16+C2_49-H2);
   (C8-G3) -(C16+C2_49-H3) M3];
```

```
% %======================================
% % Optimise MASS matrix

[MASS_opt,Q_opt,invsort]=optimise(MASS,qvec);

% %=======================================

[phi_all,R]=get_field(MASS_opt,mesh,Q_opt);

% Re-order elements
phi_all=phi_all(invsort,:);
data.phi1=phi_all(1:nvtx,:);
data.phi2=phi_all((nvtx+1):(2*nvtx),:);
data.phi3=phi_all(((2*nvtx)+1):3*nvtx,:);

% Calculate total fluence
data.phi = data.phi1 - (2/3).*data.phi2+...
   (8/15).*data.phi3;
clear phi*;

[qvec] = gen_source_adjoint(mesh);
qvec = [qvec;(-2/3)*qvec;(8/15)*qvec;(-16/35)*qvec];

[MASS_opt,qvec_opt,invsort]=optimise(MASS,qvec);
[aphi]=get_field(conj(MASS_opt),mesh,conj(qvec_opt),R);
aphi=aphi(invsort,:);
data.aphi1=aphi(1:nvtx,:);
data.aphi2=aphi((nvtx+1):(2*nvtx),:);
data.aphi3=aphi(((2*nvtx)+1):3*nvtx,:);

% Adjoint source term
data.aphi = data.aphi1 - (2/3).*data.aphi2+...
   (8/15).*data.aphi3;

clear phi* qvec* R* MASS*;
% ======================================
%

% Calculate boundary data
[data.complex]=get_boundary_data(mesh,data.phi);

% Map complex data to amplitude and phase

data.amplitude = abs(data.complex);

data.phase = atan2(imag(data.complex),...
   real(data.complex));
data.phase(find(data.phase<0)) = data.phase(find(data.phase<0)) + (2*pi);
data.phase = data.phase*180/pi;

data.paa = [data.amplitude data.phase];

% Calculate Jacobian for total fluence calculated from combination of moments.
% Catch zero frequency (CW) here
if frequency == 0
   [J] = build_jacobian_cw(mesh,data);
else
   [J] = build_jacobian(mesh,data);
end
```

```
function [fwd_mesh,pj_error1,pj_error2,pj_error3,pj_error4] = reconstruct_sp7(fwd_mesh,...
        recon_basis,   frequency, data_fn, iteration, lambda, output_fn, filter_n)

% Multi parameter image reconstruction program for spn meshes
% See chapters 6 & 8 for more details


%***************************************
% If not a workspace variable, load mesh
if ischar(fwd_mesh)== 1
  fwd_mesh = load_mesh(fwd_mesh);
end

if ischar(recon_basis)
  recon_mesh = load_mesh(recon_basis);
  [fwd_mesh.fine2coarse,...
   recon_mesh.coarse2fine] = second_mesh_basis(fwd_mesh,recon_mesh);
else
  [fwd_mesh.fine2coarse,recon_mesh] = pixel_basis(recon_basis,fwd_mesh);
end

% read data
% Need to load 4 data files, phi1, 2, 3 and 4
% Phi 1 data
anom = load_data(data_fn);
anom1(:,1) = log(anom(:,1));
anom1(:,2) = anom(:,5)/180.0*pi;
anom1(find(anom1(:,2)<0),2) = anom1(find(anom1(:,2)<0),2) + (2*pi);
anom1(find(anom1(:,2)>(2*pi)),2) = anom1(find(anom1(:,2)>(2*pi)),2) - (2*pi);
anom1 = reshape(anom1',length(anom1)*2,1);

% Phi 2 data
anom2(:,1) = log(anom(:,2));
anom2(:,2) = anom(:,6)/180.0*pi;
anom2(find(anom2(:,2)<0),2) = anom2(find(anom2(:,2)<0),2) + (2*pi);
anom2(find(anom2(:,2)>(2*pi)),2) = anom2(find(anom2(:,2)>(2*pi)),2) - (2*pi);
anom2 = reshape(anom2',length(anom2)*2,1);

% Phi 3 data
anom3(:,1) = log(anom(:,3));
anom3(:,2) = anom(:,7)/180.0*pi;
anom3(find(anom3(:,2)<0),2) = anom3(find(anom3(:,2)<0),2) + (2*pi);
anom3(find(anom3(:,2)>(2*pi)),2) = anom3(find(anom3(:,2)>(2*pi)),2) - (2*pi);
anom3 = reshape(anom3',length(anom3)*2,1);

% Phi 4 data
anom4(:,1) = log(anom(:,4));
anom4(:,2) = anom(:,8)/180.0*pi;
anom4(find(anom4(:,2)<0),2) = anom4(find(anom4(:,2)<0),2) + (2*pi);
anom4(find(anom4(:,2)>(2*pi)),2) = anom4(find(anom4(:,2)>(2*pi)),2) - (2*pi);
anom4 = reshape(anom4',length(anom4)*2,1);


% Initiate projection error for each moment of fluence
pj_error1 = [];
pj_error2 = [];
pj_error3 = [];
pj_error4 = [];

% Initiate log file
fid_log = fopen([output_fn '.log'],'w');
```

```
fprintf(fid_log,'Forward Mesh   = %s\n',fwd_mesh.name);
if ischar(recon_basis)
  fprintf(fid_log,'Basis          = %s\n',recon_basis);
else
  fprintf(fid_log,'Basis          = %s\n',num2str(recon_basis));
end
fprintf(fid_log,'Frequency      = %f MHz\n',frequency);
fprintf(fid_log,'Data File      = %s\n',data_fn);
fprintf(fid_log,'Initial Reg    = %d\n',lambda);
fprintf(fid_log,'Filter         = %d\n',filter_n);
fprintf(fid_log,'Output Files   = %s_mua.sol\n',output_fn);
fprintf(fid_log,'               = %s_mus.sol\n',output_fn);


for it = 1 : iteration

  % Calculate separate Jacobian for each moment of fluence
  [J1,J2,J3,J4,data]=jacobian_sp7(fwd_mesh,frequency);

   % Read reference data
   % Phi 1 reference data
    clear ref*
    ref1(:,1) = log(abs(data.amplitude1));
    ref1(:,2) = data.phase1;
    ref1(:,2) = ref1(:,2)/180.0*pi;
    ref1(find(ref1(:,2)<0),2) = ref1(find(ref1(:,2)<0),2) + (2*pi);
    ref1(find(ref1(:,2)>(2*pi)),2) = ref1(find(ref1(:,2)>(2*pi)),2) - (2*pi);
    ref1 = reshape(ref1',length(ref1)*2,1);

   % Phi 2 reference data
    ref2(:,1) = log(data.amplitude2);
    ref2(:,2) = data.phase2;
    ref2(:,2) = ref2(:,2)/180.0*pi;
    ref2(find(ref2(:,2)<0),2) = ref2(find(ref2(:,2)<0),2) + (2*pi);
    ref2(find(ref2(:,2)>(2*pi)),2) = ref2(find(ref2(:,2)>(2*pi)),2) - (2*pi);
    ref2 = reshape(ref2',length(ref2)*2,1);

    % Phi 3 reference data
    ref3(:,1) = log(data.amplitude3);
    ref3(:,2) = data.phase3;
    ref3(:,2) = ref3(:,2)/180.0*pi;
    ref3(find(ref3(:,2)<0),2) = ref3(find(ref3(:,2)<0),2) + (2*pi);
    ref3(find(ref3(:,2)>(2*pi)),2) = ref3(find(ref3(:,2)>(2*pi)),2) - (2*pi);
    ref3 = reshape(ref3',length(ref3)*2,1);

    % Phi 4 reference data
    ref4(:,1) = log(data.amplitude4);
    ref4(:,2) = data.phase4;
    ref4(:,2) = ref4(:,2)/180.0*pi;
    ref4(find(ref4(:,2)<0),2) = ref4(find(ref4(:,2)<0),2) + (2*pi);
    ref4(find(ref4(:,2)>(2*pi)),2) = ref4(find(ref4(:,2)>(2*pi)),2) - (2*pi);
    ref4 = reshape(ref4',length(ref4)*2,1);

    data_diff1 = abs(anom1-ref1);
    data_diff2 = abs(anom2-ref2);
    data_diff3 = abs(anom3-ref3);
    data_diff4 = abs(anom4-ref4);
    data_diff = [data_diff1; data_diff2; data_diff3; data_diff4];

% Calculate projection error
    pj_error1 = [pj_error1 sum(data_diff1.^2)];
    pj_error2 = [pj_error2 sum(data_diff2.^2)];
    pj_error3 = [pj_error3 sum(data_diff3.^2)];
    pj_error4 = [pj_error4 sum(data_diff4.^2)];

    disp('---------------------------------');
    disp(['Iteration Number         = ' num2str(it)]);
    disp(['Projection error 1       = ' num2str(pj_error1(end))]);
    disp(['Projection error 2       = ' num2str(pj_error2(end))]);
    disp(['Projection error 3       = ' num2str(pj_error3(end))]);
    disp(['Projection error 4       = ' num2str(pj_error4(end))]);
    fprintf(fid_log,'---------------------------------\n');
    fprintf(fid_log,'Iteration Number          = %d\n',it);
    fprintf(fid_log,'Projection error - phi 1       = %f\n',pj_error1(end));
    fprintf(fid_log,'Projection error - phi 2       = %f\n',pj_error2(end));
    fprintf(fid_log,'Projection error - phi 3       = %f\n',pj_error3(end));
    fprintf(fid_log,'Projection error - phi 4       = %f\n',pj_error4(end));

    if it ~= 1
      p1 = (pj_error1(end-1)-pj_error1(end))*100/pj_error1(end-1);
      p2 = (pj_error2(end-1)-pj_error2(end))*100/pj_error2(end-1);
      p3 = (pj_error3(end-1)-pj_error3(end))*100/pj_error3(end-1);
      p4 = (pj_error4(end-1)-pj_error4(end))*100/pj_error4(end-1);
      disp(['Projection error change for phi1   = ' num2str(p1) '%']);
      disp(['Projection error change for phi2   = ' num2str(p2) '%']);
      disp(['Projection error change for phi3   = ' num2str(p3) '%']);
      disp(['Projection error change for phi4   = ' num2str(p4) '%']);
      fprintf(fid_log,'Projection error change for phi1   = %f %%\n',p1);
      fprintf(fid_log,'Projection error change for phi2   = %f %%\n',p2);
      fprintf(fid_log,'Projection error change for phi3   = %f %%\n',p3);
      fprintf(fid_log,'Projection error change for phi4   = %f %%\n',p4);

% Test for stopping criteria
    if (p1) <= 0.1 & (p2) <= 0.1 & (p3)<= 0.1 & (p4)<= 0.1 % set stopping criteria < 2%
        disp('---------------------------------');
        disp('STOPPING CRITERIA REACHED');
        fprintf(fid_log,'---------------------------------\n');
        fprintf(fid_log,'STOPPING CRITERIA REACHED\n');
        break
      end
    end

     % Interpolate onto recon mesh
    [J1,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J1.complete);
    [J2,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J2.complete);
    [J3,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J3.complete);
    [J4,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J4.complete);

    % Calculate higher ordered optical property terms
    mua = recon_mesh.mua;
    mua1 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g));
    mua2 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^2));
    mua3 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^3));
    mua4 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^4));
    mua5 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^5));
    mua6 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^6));
    mua7 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^7));
    muai = ((4/9).*mua)+((5/9).*mua2);
    muaii= ((16/45).*mua)+((4/9).*mua2);
    muaiii=((32/105).*mua)+((8/21).*mua2);
    muaiv= ((64/225).*mua)+((16/45).*mua2)+((9/25).*mua4);
    muav = ((128/525).*mua)+((32/105).*mua2)+((54/175).*mua4);
    muavi= ((256/1225).*mua)+((64/245).*mua2)+((324/1225).*mua4)+ ...
                  ((13/49).*mua6);
    kappa1 = 1./(3.*mua1);
    kappa3 = 1./(7.*mua3);
    kappa5 = 1./(11.*mua5);
    kappa7 = 1./(15.*mua7);

    %% Find part of Jacobian that relates to other moments of fluence
    foo1 = J1(:,end/2+1:end);
    foo2 = J2(:,end/2+1:end);
    foo3 = J3(:,end/2+1:end);
    foo4 = J4(:,end/2+1:end);
    [nrow,ncol]=size(J1);

    % Construct complete jacobian and normalise at same time
    dmua = diag(mua);
    J1 = [J1*diag([kappa1;mua]) zeros(nrow,3.*ncol) -foo2*dmua foo3*dmua -foo4*dmua
zeros(nrow,(3.*ncol)./2)];
    J2 = [zeros(nrow,ncol) J2*diag([kappa3;muai]) zeros(nrow,2.*ncol) -foo1*dmua
zeros(nrow,ncol) ...
        -foo3*diag(muaii) foo4*diag(muaiii) zeros(nrow,ncol./2)];
    J3 = [zeros(nrow,2.*ncol) J3*diag([kappa5; muaiv]) zeros(nrow,(3.*ncol)./2) foo1*dmua ...
        zeros(nrow,ncol./2) -foo2*diag(muaii) zeros(nrow,ncol./2) -foo4*diag(muav)];
    J4 = [zeros(nrow,3.*ncol) J4*diag([kappa7; muavi]) zeros(nrow,ncol) -foo1*dmua
foo2*diag(muaii) ...
        zeros(nrow,ncol./2) -foo3*diag(muav)];

    clear foo* dmua

    J = [J1; J2; J3; J4];



    % build hessian
    [nrow,ncol]=size(J);
    Hess = zeros(nrow);
    Hess = (J*J');

    % Add regularization
    if it ~= 1
      lambda = lambda./10^0.125;
    end

    % Hess
    reg_amp = lambda*max(diag(Hess(1:2:end,1:2:end)));
    reg_phs = lambda*max(diag(Hess(2:2:end,2:2:end)));
    reg = ones(nrow,1);
    reg(1:2:end) = reg_amp;
    reg(2:2:end) = reg_phs;
    reg = diag(reg);
    reg = lambda.*eye(size(Hess));

    disp(['Amplitude Regularization      = ' num2str(reg_amp)]);
    disp(['Phase Regularization          = ' num2str(reg_phs)]);
    fprintf(fid_log,'Amplitude Regularization      = %f\n',reg_amp);
    fprintf(fid_log,'Phase Regularization          = %f\n',reg_phs);

    Hess = Hess+reg;

    % Calculate update
    foo = J'*(Hess\data_diff);
    foo = foo.*[kappa1;mua; kappa3; muai; kappa5; muaiv; kappa7; muavi; ...
        mua; mua; mua; muaii; muaiii; muav];

    % Update values
    recon_mesh.kappa1 = kappa1 + foo(1:end/14);
    recon_mesh.mua = mua + foo(end/14+1:2*end/14);
    recon_mesh.kappa3 = kappa3 + foo(2*end/14+1:3*end/14);
    recon_mesh.muai = muai + foo(3*end/14+1:4*end/14);
    recon_mesh.kappa5 = kappa5 + foo(4*end/14+1:5*end/14);
    recon_mesh.muaiv = muaiv + foo(5*end/14+1:6*end/14);
    recon_mesh.kappa7 = kappa7 +foo(6*end/14+1:7*end/14);
    recon_mesh.muavi = muavi + foo(7*end/14+1:8*end/14);
    recon_mesh.mua23 = (2/3)*mua +foo(8*end/14+1:9*end/14);
    recon_mesh.mua815 = (8/15)*mua + foo(9*end/14+1:10*end/14);
    recon_mesh.mua1635 = (16/35)*mua + foo(10*end/14+1:11*end/14);
    recon_mesh.muaii = muaii + foo(11*end/14+1:12*end/14);
    recon_mesh.muaiii = muaiii + foo(12*end/14+1:13*end/14);
    recon_mesh.muav = muav + foo(13*end/14+1:end);

    recon_mesh.mua = (recon_mesh.mua + ((3/2).*recon_mesh.mua23)+ ...
            ((15/8).*recon_mesh.mua815)+ ...
        ((35/16).*recon_mesh.mua1635))./4;
    recon_mesh.mua1 = 1./(3.*recon_mesh.kappa1);
    recon_mesh.mua3 = 1./(7.*recon_mesh.kappa3);
    recon_mesh.mua5 = 1./(11.*recon_mesh.kappa5);
    recon_mesh.mua7 = 1./(15.*recon_mesh.kappa7);

    mua2_1=(9/5).*(recon_mesh.muai-((4/9).*recon_mesh.mua));
    mua2_2 = (9/4).*(recon_mesh.muaii-(16/45).*recon_mesh.mua);
    mua2_3 = (21/8).*(recon_mesh.muaiii-(32/105).*recon_mesh.mua);
    recon_mesh.mua2 = (mua2_1+mua2_2+mua2_3)./3;

    mua4_1 = (25/9).*(muaiv - ((64/225).*recon_mesh.mua)- ((16/45).*recon_mesh.mua2));
    mua4_2 = (175/54).*(muav - ((128/525).*recon_mesh.mua)- ((32/105).*recon_mesh.mua2));
    recon_mesh.mua4 = (mua4_1+mua4_2)./2;

    recon_mesh.mua6 = (49/13).*(recon_mesh.muavi-((256/1225).*recon_mesh.mua)-
        ((64/245).*recon_mesh.mua2)- ((324/1225).*recon_mesh.mua4));

    mus1 = (recon_mesh.mua1-recon_mesh.mua)./(1-recon_mesh.g.^1);
    mus2 = (recon_mesh.mua2-recon_mesh.mua)./(1-recon_mesh.g.^2);
    mus3 = (recon_mesh.mua3-recon_mesh.mua)./(1-recon_mesh.g.^3);
    mus4 = (recon_mesh.mua4-recon_mesh.mua)./(1-recon_mesh.g.^4);
    mus5 = (recon_mesh.mua5-recon_mesh.mua)./(1-recon_mesh.g.^5);
    mus6 = (recon_mesh.mua6-recon_mesh.mua)./(1-recon_mesh.g.^6);
    mus7 = (recon_mesh.mua7-recon_mesh.mua)./(1-recon_mesh.g.^7);

    recon_mesh.mus = (mus1+mus2+mus3+mus4+mus5+mus6+mus7)./7;

    clear foo* Hess* Hess_norm tmp data_diff* G mus*

    % Interpolate optical properties to fine mesh
    [fwd_mesh,recon_mesh] = interpolatep2f(fwd_mesh,recon_mesh);

    % We dont like -ve mua or mus! so if this happens, terminate
    if (any(fwd_mesh.mua<0) | any(fwd_mesh.mus<0))
      disp('---------------------------------');
      disp('-ve mua or mus calculated...not saving solution');
      fprintf(fid_log,'---------------------------------\n');
      fprintf(fid_log,'STOPPING CRITERIA REACHED\n');
      break
    end

    % Filtering if needed!
    if filter_n > 1
      fwd_mesh = mean_filter(fwd_mesh,abs(filter_n));
    elseif filter_n < 0
      fwd_mesh = median_filter(fwd_mesh,abs(filter_n));
    end
```

```
    if it == 1
        fid = fopen([output_fn '_mua.sol'],'w');
    else
        fid = fopen([output_fn '_mua.sol'],'a');
    end
    fprintf(fid,'solution %g ',it);
    fprintf(fid,'-size=%g ',length(fwd_mesh.nodes));
    fprintf(fid,'-components=1 ');
    fprintf(fid,'-type=nodal\n');
    fprintf(fid,'%f ',fwd_mesh.mua);
    fprintf(fid,'\n');
    fclose(fid);

  if it == 1
        fid = fopen([output_fn '_mus.sol'],'w');
    else
        fid = fopen([output_fn '_mus.sol'],'a');
    end
    fprintf(fid,'solution %g ',it);
    fprintf(fid,'-size=%g ',length(fwd_mesh.nodes));
    fprintf(fid,'-components=1 ');
    fprintf(fid,'-type=nodal\n');
    fprintf(fid,'%f ',fwd_mesh.mus);
    fprintf(fid,'\n');
    fclose(fid);
end

% close log file!
time = toc;
fprintf(fid_log,'Computation TimeRegularization = %f\n',time);
fclose(fid_log);
```

```
function                    [fwd_mesh,pj_error1,pj_error2,pj_error3,pj_error4]                    =
reconstruct_sp7_region(fwd_mesh, recon_basis, frequency, data_fn, iteration, lambda, output_fn,...
        filter_n, region)

% multiparameter reconstruction program for spn meshes
% using hard priors
% See chapters 6 & 8 for details


tic;

%****************************************
% If not a workspace variable, load mesh
if ischar(fwd_mesh)== 1
  fwd_mesh = load_mesh(fwd_mesh);
end

if ischar(recon_basis)
  recon_mesh = load_mesh(recon_basis);
  [fwd_mesh.fine2coarse,...
   recon_mesh.coarse2fine] = second_mesh_basis(fwd_mesh,recon_mesh);
else
  [fwd_mesh.fine2coarse,recon_mesh] = pixel_basis(recon_basis,fwd_mesh);
end

% read data
% Need to load 2 data files, phi1 and phi2
% Phi 1 data
anom = load_data(data_fn);
anom1(:,1) = log(anom(:,1));
anom1(:,2) = anom(:,5)/180.0*pi;
anom1(find(anom1(:,2)<0),2) = anom1(find(anom1(:,2)<0),2) + (2*pi);
anom1(find(anom1(:,2)>(2*pi)),2) = anom1(find(anom1(:,2)>(2*pi)),2) - (2*pi);
anom1 = reshape(anom1',length(anom1)*2,1);

% Phi 2 data
anom2(:,1) = log(anom(:,2));
anom2(:,2) = anom(:,6)/180.0*pi;
anom2(find(anom2(:,2)<0),2) = anom2(find(anom2(:,2)<0),2) + (2*pi);
anom2(find(anom2(:,2)>(2*pi)),2) = anom2(find(anom2(:,2)>(2*pi)),2) - (2*pi);
anom2 = reshape(anom2',length(anom2)*2,1);

% Phi 3 data
anom3(:,1) = log(anom(:,3));
anom3(:,2) = anom(:,7)/180.0*pi;
anom3(find(anom3(:,2)<0),2) = anom3(find(anom3(:,2)<0),2) + (2*pi);
anom3(find(anom3(:,2)>(2*pi)),2) = anom3(find(anom3(:,2)>(2*pi)),2) - (2*pi);
anom3 = reshape(anom3',length(anom3)*2,1);

% Phi 4 data
anom4(:,1) = log(anom(:,4));
anom4(:,2) = anom(:,8)/180.0*pi;
anom4(find(anom4(:,2)<0),2) = anom4(find(anom4(:,2)<0),2) + (2*pi);
anom4(find(anom4(:,2)>(2*pi)),2) = anom4(find(anom4(:,2)>(2*pi)),2) - (2*pi);
anom4 = reshape(anom4',length(anom4)*2,1);


% Initiate projection error
pj_error1 = [];
pj_error2 = [];
pj_error3 = [];
pj_error4 = [];

% Initiate log file
fid_log = fopen([output_fn '.log'],'w');
fprintf(fid_log,'Forward Mesh   = %s\n',fwd_mesh.name);
if ischar(recon_basis)
  fprintf(fid_log,'Basis          = %s\n',recon_basis);
else
  fprintf(fid_log,'Basis          = %s\n',num2str(recon_basis));
end
fprintf(fid_log,'Frequency      = %f MHz\n',frequency);
fprintf(fid_log,'Data File      = %s\n',data_fn);
fprintf(fid_log,'Initial Reg    = %d\n',lambda);
fprintf(fid_log,'Filter         = %d\n',filter_n);
fprintf(fid_log,'Output Files   = %s_mua.sol\n',output_fn);
fprintf(fid_log,'               = %s_mus.sol\n',output_fn);

disp('calculating regions');
if region ~= -1
   K = region_mapper(recon_mesh,region);
else
   K = 1;
end


for it = 1 : iteration

  % Calculate jacobian
  [J1,J2,J3,J4,data]=jacobian_sp7(fwd_mesh,frequency);
```

```
% Read reference data
% Phi 1 reference data
 clear ref*
 ref1(:,1) = log(abs(data.amplitude1));
 ref1(:,2) = data.phase1;
 ref1(:,2) = ref1(:,2)/180.0*pi;
 ref1(find(ref1(:,2)<0),2) = ref1(find(ref1(:,2)<0),2) + (2*pi);
 ref1(find(ref1(:,2)>(2*pi)),2) = ref1(find(ref1(:,2)>(2*pi)),2) - (2*pi);
 ref1 = reshape(ref1',length(ref1)*2,1);

% Phi 2 reference data
 ref2(:,1) = log(data.amplitude2);
 ref2(:,2) = data.phase2;
 ref2(:,2) = ref2(:,2)/180.0*pi;
 ref2(find(ref2(:,2)<0),2) = ref2(find(ref2(:,2)<0),2) + (2*pi);
 ref2(find(ref2(:,2)>(2*pi)),2) = ref2(find(ref2(:,2)>(2*pi)),2) - (2*pi);
 ref2 = reshape(ref2',length(ref2)*2,1);

 % Phi 3 reference data
 ref3(:,1) = log(data.amplitude3);
 ref3(:,2) = data.phase3;
 ref3(:,2) = ref3(:,2)/180.0*pi;
 ref3(find(ref3(:,2)<0),2) = ref3(find(ref3(:,2)<0),2) + (2*pi);
 ref3(find(ref3(:,2)>(2*pi)),2) = ref3(find(ref3(:,2)>(2*pi)),2) - (2*pi);
 ref3 = reshape(ref3',length(ref3)*2,1);

 % Phi 4 reference data
 ref4(:,1) = log(data.amplitude4);
 ref4(:,2) = data.phase4;
 ref4(:,2) = ref4(:,2)/180.0*pi;
 ref4(find(ref4(:,2)<0),2) = ref4(find(ref4(:,2)<0),2) + (2*pi);
 ref4(find(ref4(:,2)>(2*pi)),2) = ref4(find(ref4(:,2)>(2*pi)),2) - (2*pi);
 ref4 = reshape(ref4',length(ref4)*2,1);

 data_diff1 = abs(anom1-ref1);
 data_diff2 = abs(anom2-ref2);
 data_diff3 = abs(anom3-ref3);
 data_diff4 = abs(anom4-ref4);
 data_diff = [data_diff1; data_diff2; data_diff3; data_diff4];

 pj_error1 = [pj_error1 sum(data_diff1.^2)];
 pj_error2 = [pj_error2 sum(data_diff2.^2)];
 pj_error3 = [pj_error3 sum(data_diff3.^2)];
 pj_error4 = [pj_error4 sum(data_diff4.^2)];

 disp('---------------------------------');
 disp(['Iteration Number          = ' num2str(it)]);
 disp(['Projection error 1        = ' num2str(pj_error1(end))]);
 disp(['Projection error 2        = ' num2str(pj_error2(end))]);
 disp(['Projection error 3        = ' num2str(pj_error3(end))]);
 disp(['Projection error 4        = ' num2str(pj_error4(end))]);
 fprintf(fid_log,'---------------------------------\n');
 fprintf(fid_log,'Iteration Number         = %d\n',it);
 fprintf(fid_log,'Projection error - phi 1       = %f\n',pj_error1(end));
 fprintf(fid_log,'Projection error - phi 2       = %f\n',pj_error2(end));
 fprintf(fid_log,'Projection error - phi 3       = %f\n',pj_error3(end));
 fprintf(fid_log,'Projection error - phi 4       = %f\n',pj_error4(end));

 if it ~= 1
    p1 = (pj_error1(end-1)-pj_error1(end))*100/pj_error1(end-1);
    p2 = (pj_error2(end-1)-pj_error2(end))*100/pj_error2(end-1);
    p3 = (pj_error3(end-1)-pj_error3(end))*100/pj_error3(end-1);
    p4 = (pj_error4(end-1)-pj_error4(end))*100/pj_error4(end-1);
    disp(['Projection error change for phi1   = ' num2str(p1) '%']);
    disp(['Projection error change for phi2   = ' num2str(p2) '%']);
    disp(['Projection error change for phi3   = ' num2str(p3) '%']);
    disp(['Projection error change for phi4   = ' num2str(p4) '%']);
    fprintf(fid_log,'Projection error change for phi1   = %f %%\n',p1);
    fprintf(fid_log,'Projection error change for phi2   = %f %%\n',p2);
    fprintf(fid_log,'Projection error change for phi3   = %f %%\n',p3);
    fprintf(fid_log,'Projection error change for phi4   = %f %%\n',p4);

if (p1) <= 0.1 & (p2) <= 0.1 & (p3)<= 0.1 & (p4)<= 0.1 % set stopping criteria < 2%
      disp('---------------------------------');
      disp('STOPPING CRITERIA REACHED');
      fprintf(fid_log,'---------------------------------\n');
      fprintf(fid_log,'STOPPING CRITERIA REACHED\n');
      break
    end
end

  % Interpolate onto recon mesh
[J1,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J1.complete);
[J2,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J2.complete);
[J3,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J3.complete);
[J4,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J4.complete);

% Normalize Jacobian wrt optical values
mua = recon_mesh.mua;
mua1 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g));
mua2 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^2));
mua3 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^3));
mua4 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^4));
mua5 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^5));
mua6 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^6));
mua7 = (mua+recon_mesh.mus)-(recon_mesh.mus.*(recon_mesh.g.^7));
muai = ((4/9).*mua)+((5/9).*mua2);
muaii= ((16/45).*mua)+((4/9).*mua2);
muaiii=((32/105).*mua)+((8/21).*mua2);
muaiv= ((64/225).*mua)+((16/45).*mua2)+((9/25).*mua4);
muav = ((128/525).*mua)+((32/105).*mua2)+((54/175).*mua4);
muavi= ((256/1225).*mua)+((64/245).*mua2)+((324/1225).*mua4)+ ...
           ((13/49).*mua6);
kappa1 = 1./(3.*mua1);
kappa3 = 1./(7.*mua3);
kappa5 = 1./(11.*mua5);
kappa7 = 1./(15.*mua7);

%% Adding dependance of phi1 and phi2
KK = [K zeros(size(K)); zeros(size(K)) K];
foo1 = J1(:,end/2+1:end);
foo2 = J2(:,end/2+1:end);
foo3 = J3(:,end/2+1:end);
foo4 = J4(:,end/2+1:end);
[nrow,ncol]=size(J1);
K1 = zeros(nrow,ncol./2)*K;
KK2 = zeros(nrow,ncol)*KK;
J1 = [J1*KK KK2 KK2 KK2 -foo2*K foo3*K -foo4*K K1 KK2];
J2 = [KK2 J2*KK KK2 KK2 -foo1*K KK2 -foo3*K foo4*K K1];
J3 = [KK2 KK2 J3*KK K1 KK2 foo1*K K1 -foo2*K K1 -foo4*K];
J4 = [KK2 KK2 KK2 J4*KK KK2 -foo1*K foo2*K K1 -foo3*K];

clear foo* KK K1 KK2

% Normalize Jacobian wrt optical values
```

```
  N = [kappa1'*K./sum(K) mua'*K./sum(K) kappa3'*K./sum(K) ...
    muai'*K./sum(K) kappa5'*K./sum(K) muaiv'*K./sum(K) ...
    kappa7'*K./sum(K) muavi'*K./sum(K) mua'*K./sum(K) ...
    mua'*K./sum(K) mua'*K./sum(K) muaii'*K./sum(K) ...
    muaiii'*K./sum(K) muav'*K./sum(K)];

  J = [J1; J2; J3; J4]*diag(N);

  % build hessian
  [nrow,ncol]=size(J);
  Hess = zeros(ncol);
  Hess = (J'*J);

reg = ones(ncol,1).*max(diag(Hess)).*lambda;
  reg = diag(reg);

  disp(['Regularization        = ' num2str(reg(1,1))]);
  fprintf(fid_log,'Regularization        = %f\n',reg(1,1));

  Hess = Hess+reg;

  % Calculate update
  foo = Hess\(J'*data_diff);
  foo = foo.*N';
  foo1 = [];
  for i = 1 : 14
    foo1 = [foo1; K*foo((i-1)*end/14+1:i*end/14)];
  end
  foo = foo1;

  % Update values
  recon_mesh.kappa1 = kappa1 + foo(1:end/14);
  recon_mesh.mua = mua + foo(end/14+1:2*end/14);
  recon_mesh.kappa3 = kappa3 + foo(2*end/14+1:3*end/14);
  recon_mesh.muai = muai + foo(3*end/14+1:4*end/14);
  recon_mesh.kappa5 = kappa5 + foo(4*end/14+1:5*end/14);
  recon_mesh.muaiv = muaiv + foo(5*end/14+1:6*end/14);
  recon_mesh.kappa7 = kappa7 +foo(6*end/14+1:7*end/14);
  recon_mesh.muavi = muavi + foo(7*end/14+1:8*end/14);
  recon_mesh.mua23 = (2/3)*mua +foo(8*end/14+1:9*end/14);
  recon_mesh.mua815 = (8/15)*mua + foo(9*end/14+1:10*end/14);
  recon_mesh.mua1635 = (16/35)*mua + foo(10*end/14+1:11*end/14);
  recon_mesh.muaii = muaii + foo(11*end/14+1:12*end/14);
  recon_mesh.muaiii = muaiii + foo(12*end/14+1:13*end/14);
  recon_mesh.muav = muav + foo(13*end/14+1:end);

  recon_mesh.mua = (recon_mesh.mua + ((3/2).*recon_mesh.mua23)+ ...
        ((15/8).*recon_mesh.mua815)+ ...
         ((35/16).*recon_mesh.mua1635))./4;
  recon_mesh.mua1 = 1./(3.*recon_mesh.kappa1);
  recon_mesh.mua3 = 1./(7.*recon_mesh.kappa3);
  recon_mesh.mua5 = 1./(11.*recon_mesh.kappa5);
  recon_mesh.mua7 = 1./(15.*recon_mesh.kappa7);

  mua2_1=(9/5).*(recon_mesh.muai-((4/9).*recon_mesh.mua));
  mua2_2 = (9/4).*(recon_mesh.muaii-(16/45).*recon_mesh.mua);
  mua2_3 = (21/8).*(recon_mesh.muaiii-(32/105).*recon_mesh.mua);
  recon_mesh.mua2 = (mua2_1+mua2_2+mua2_3)./3;

  mua4_1 = (25/9).*(muaiv - ((64/225).*recon_mesh.mua)- ...
         ((16/45).*recon_mesh.mua2));
  mua4_2 = (175/54).*(muav - ((128/525).*recon_mesh.mua)- ...
     ((32/105).*recon_mesh.mua2));
  recon_mesh.mua4 = (mua4_1+mua4_2)./2;

  recon_mesh.mua6 = (49/13).*(recon_mesh.muavi- ((256/1225).*recon_mesh.mua)-…
      ((64/245).*recon_mesh.mua2)- ((324/1225).*recon_mesh.mua4));

  mus1 = (recon_mesh.mua1-recon_mesh.mua)./(1-recon_mesh.g.^1);
  mus2 = (recon_mesh.mua2-recon_mesh.mua)./(1-recon_mesh.g.^2);
  mus3 = (recon_mesh.mua3-recon_mesh.mua)./(1-recon_mesh.g.^3);
  mus4 = (recon_mesh.mua4-recon_mesh.mua)./(1-recon_mesh.g.^4);
  mus5 = (recon_mesh.mua5-recon_mesh.mua)./(1-recon_mesh.g.^5);
  mus6 = (recon_mesh.mua6-recon_mesh.mua)./(1-recon_mesh.g.^6);
  mus7 = (recon_mesh.mua7-recon_mesh.mua)./(1-recon_mesh.g.^7);

  recon_mesh.mus = (mus1+mus2+mus3+mus4+mus5+mus6+mus7)./7;

  clear foo* Hess* Hess_norm tmp data_diff* G mus*

  % Interpolate optical properties to fine mesh
  [fwd_mesh,recon_mesh] = interpolatep2f(fwd_mesh,recon_mesh);

  % We dont like -ve mua or mus! so if this happens, terminate
  if (any(fwd_mesh.mua<0) | any(fwd_mesh.mus<0))
    disp('---------------------------------');
    disp('-ve mua or mus calculated...not saving solution');
    fprintf(fid_log,'---------------------------------\n');
    fprintf(fid_log,'STOPPING CRITERIA REACHED\n');
    break
  end

  % Filtering if needed!
  if filter_n > 1
    fwd_mesh = mean_filter(fwd_mesh,abs(filter_n));
  elseif filter_n < 0
    fwd_mesh = median_filter(fwd_mesh,abs(filter_n));
  end

  if it == 1
    fid = fopen([output_fn '_mua.sol'],'w');
  else
    fid = fopen([output_fn '_mua.sol'],'a');
  end
  fprintf(fid,'solution %g ',it);
  fprintf(fid,'-size=%g ',length(fwd_mesh.nodes));
  fprintf(fid,'-components=1 ');
  fprintf(fid,'-type=nodal\n');
  fprintf(fid,'%f ',fwd_mesh.mua);
  fprintf(fid,'\n');
  fclose(fid);

 if it == 1
    fid = fopen([output_fn '_mus.sol'],'w');
  else
    fid = fopen([output_fn '_mus.sol'],'a');
  end
  fprintf(fid,'solution %g ',it);
  fprintf(fid,'-size=%g ',length(fwd_mesh.nodes));
  fprintf(fid,'-components=1 ');
  fprintf(fid,'-type=nodal\n');
  fprintf(fid,'%f ',fwd_mesh.mus);
  fprintf(fid,'\n');
  fclose(fid);
end
```

```
% close log file!
time = toc;
fprintf(fid_log,'Computation TimeRegularization = %f\n',time);
fclose(fid_log);




function [val_int,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,val)

% This function interpolates fwd_mesh into recon_mesh
% For the Jacobian it is an integration!
NNC = size(recon_mesh.nodes,1);
NNF = size(fwd_mesh.nodes,1);
NROW = size(val,1);
val_int = zeros(NROW,NNC*2);

for i = 1 : NNF
  if recon_mesh.coarse2fine(i,1) ~= 0
    val_int(:,recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)) = ...
    val_int(:,recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)) + ...
        val(:,i)*recon_mesh.coarse2fine(i,2:end);
    val_int(:,recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)+NNC) = ...
    val_int(:,recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)+NNC) + ...
        val(:,i+NNF)*recon_mesh.coarse2fine(i,2:end);
  elseif recon_mesh.coarse2fine(i,1) == 0
    dist = distance(fwd_mesh.nodes,fwd_mesh.bndvtx,recon_mesh.nodes(i,:));
    mindist = find(dist==min(dist));
    mindist = mindist(1);
    val_int(:,i) = val(:,mindist);
    val_int(:,i+NNC) = val(:,mindist+NNF);
  end
end

for i = 1 : NNC
  if fwd_mesh.fine2coarse(i,1) ~= 0
    recon_mesh.mua(i,1) = (fwd_mesh.fine2coarse(i,2:end) * ...
    fwd_mesh.mua(fwd_mesh.elements(fwd_mesh.fine2coarse(i,1),:)));
    recon_mesh.mus(i,1) = (fwd_mesh.fine2coarse(i,2:end) * ...
    fwd_mesh.mus(fwd_mesh.elements(fwd_mesh.fine2coarse(i,1),:)));
    recon_mesh.c(i,1) = (fwd_mesh.fine2coarse(i,2:end) * ...
    fwd_mesh.c(fwd_mesh.elements(fwd_mesh.fine2coarse(i,1),:)));
    recon_mesh.g(i,1) = (fwd_mesh.fine2coarse(i,2:end) * ...
    fwd_mesh.g(fwd_mesh.elements(fwd_mesh.fine2coarse(i,1),:)));
    recon_mesh.region(i,1) = ...
    median(fwd_mesh.region(fwd_mesh.elements(fwd_mesh.fine2coarse(i,1),:)));
  elseif fwd_mesh.fine2coarse(i,1) == 0
    dist = distance(fwd_mesh.nodes,...
            fwd_mesh.bndvtx,...
            [recon_mesh.nodes(i,1:2) 0]);
    mindist = find(dist==min(dist));
    mindist = mindist(1);
    recon_mesh.mua(i,1) = fwd_mesh.mua(mindist);
    recon_mesh.mus(i,1) = fwd_mesh.mus(mindist);
    recon_mesh.c(i,1) = fwd_mesh.c(mindist);
    recon_mesh.g(i,1) = fwd_mesh.g(mindist);
    recon_mesh.region(i,1) = fwd_mesh.region(mindist);
  end
end

function [fwd_mesh,recon_mesh] = interpolatep2f(fwd_mesh,recon_mesh)
```

```
for i = 1 : length(fwd_mesh.nodes)
 fwd_mesh.mua(i,1) = ...
    (recon_mesh.coarse2fine(i,2:end) * ...
     recon_mesh.mua(recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)));
 fwd_mesh.c(i,1) = ...
    (recon_mesh.coarse2fine(i,2:end) * ...
     recon_mesh.c(recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)));
 fwd_mesh.mus(i,1) = ...
    (recon_mesh.coarse2fine(i,2:end) * ...
     recon_mesh.mus(recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)));
  fwd_mesh.g(i,1) = ...
    (recon_mesh.coarse2fine(i,2:end) * ...
     recon_mesh.g(recon_mesh.elements(recon_mesh.coarse2fine(i,1),:)));
end


function KKK=region_mapper(mesh,region)
% Region mapper for region-based reconstruction

nregion = length(region)
nnodes = length(mesh.nodes);

% create empty mapping matrix
K = sparse(nnodes,nregion);

% Assign mapping functions, for each node belonging to each region
for j = 1 : nregion
   K(find(mesh.region==region(j)),j) = 1;
end

% find the total number of assigned nodes
N = full(sum(sum(K)));

% Here if some node is not in region, must account for it
if N ~= length(mesh.nodes)
   KK = sparse(nnodes,nnodes-N);
   for k = 1 : length(region)
      if k == 1
         a = find(mesh.region~=region(k));
      else
         a = intersect(find(mesh.region~=region(k)),a);
      end
   end
   for i = 1 : length(a)
      KK(a(i),i) = 1;
   end
   KKK = [K KK];
else
   KKK = K;
end
```

```
function [fwd_mesh,pj_error] = reconstruct_spn(fwd_mesh, recon_basis, frequency, data_fn,
      iteration, lambda, output_fn, filter_n)

% Diffusion based reconstruction program for spn meshes
% See chapters 6 & 8 for details
%

tic;
```

```
%**************************************
% If not a workspace variable, load mesh
if ischar(fwd_mesh)== 1
  fwd_mesh = load_mesh(fwd_mesh);
end

if ischar(recon_basis)
  recon_mesh = load_mesh(recon_basis);
  [fwd_mesh.fine2coarse,...
   recon_mesh.coarse2fine] = second_mesh_basis(fwd_mesh,recon_mesh);
else
  [fwd_mesh.fine2coarse,recon_mesh] = pixel_basis(recon_basis,fwd_mesh);
end

% read data
tmp = load_data(data_fn);
[i j]=size(tmp);
if j ==2
  anom(:,1) = log(tmp(:,1));
  anom(:,2) = tmp(:,2)/180.0*pi;
  anom(find(anom(:,2)<0),2) = anom(find(anom(:,2)<0),2) + (2*pi);
  anom(find(anom(:,2)>(2*pi)),2) = anom(find(anom(:,2)>(2*pi)),2) - (2*pi);
  anom = reshape(anom',length(anom)*2,1);
else
disp('Forward data is in wrong format. Use save_data to save phase/amp data');
end

% Initiate projection error
pj_error = [];

% Initiate log file
fid_log = fopen([output_fn '.log'],'w');
fprintf(fid_log,'Forward Mesh   = %s\n',fwd_mesh.name);
if ischar(recon_basis)
  fprintf(fid_log,'Basis          = %s\n',recon_basis);
else
  fprintf(fid_log,'Basis          = %s\n',num2str(recon_basis));
end
fprintf(fid_log,'Frequency      = %f MHz\n',frequency);
fprintf(fid_log,'Data File      = %s\n',data_fn);
fprintf(fid_log,'Initial Reg    = %d\n',lambda);
fprintf(fid_log,'Filter         = %d\n',filter_n);
fprintf(fid_log,'Output Files   = %s_mua.sol\n',output_fn);
fprintf(fid_log,'               = %s_mus.sol\n',output_fn);


for it = 1 : iteration

  % Calculate jacobian
  [J,data]=jacobian_spn(fwd_mesh,frequency);

  % Read reference data
  clear ref;
  ref(:,1) = log(data.amplitude);
  ref(:,2) = data.phase;
  ref(:,2) = ref(:,2)/180.0*pi;
  ref(find(ref(:,2)<0),2) = ref(find(ref(:,2)<0),2) + (2*pi);
  ref(find(ref(:,2)>(2*pi)),2) = ref(find(ref(:,2)>(2*pi)),2) - (2*pi);
  ref = reshape(ref',length(ref)*2,1);
 data_diff = (anom-ref);

  pj_error = [pj_error sum((anom-ref).^2)];
```

```
disp('---------------------------------');
disp(['Iteration Number          = ' num2str(it)]);
disp(['Projection error          = ' num2str(pj_error(end))]);

fprintf(fid_log,'---------------------------------\n');
fprintf(fid_log,'Iteration Number          = %d\n',it);
fprintf(fid_log,'Projection error          = %f\n',pj_error(end));

if it ~= 1
  p = (pj_error(end-1)-pj_error(end))*100/pj_error(end-1);
  disp(['Projection error change   = ' num2str(p) '%']);
  fprintf(fid_log,'Projection error change   = %f %%\n',p);
  if (p) <= 0.1
    disp('---------------------------------');
    disp('STOPPING CRITERIA REACHED');
    fprintf(fid_log,'---------------------------------\n');
    fprintf(fid_log,'STOPPING CRITERIA REACHED\n');
   break
  end
end

% Interpolate onto recon mesh
[J,recon_mesh] = interpolatef2r(fwd_mesh,recon_mesh,J.complete);

recon_mesh.kappa=1./(3.*((recon_mesh.mua+((1-recon_mesh.g).*recon_mesh.mus))));
% Normalize Jacobian wrt optical values
J = J*diag([recon_mesh.kappa;recon_mesh.mua]);

% build hessian
[nrow,ncol]=size(J);
Hess = zeros(nrow);
Hess = (J*J');

% initailize temp Hess, data and mesh, incase PJ increases.
Hess_tmp = Hess;
mesh_tmp = recon_mesh;
data_tmp = data_diff;

% Add regularization
if it ~= 1
  lambda = lambda./10^0.125;
end

% Seems that scatter part is always more noisey. So we will make
% sure that the regularization for phase is some factor (ratio
% of amplitude vs phase diagonals of the Hessian) higher.

reg_amp = lambda*max(diag(Hess(1:2:end,1:2:end)));
reg_phs = lambda*max(diag(Hess(2:2:end,2:2:end)));
reg = ones(nrow,1);
reg(1:2:end) = reg(1:2:end).*reg_amp;
reg(2:2:end) = reg(2:2:end).*reg_phs;
reg = diag(reg);

disp(['Amp Regularization        = ' num2str(reg(1,1))]);
disp(['Phs Regularization        = ' num2str(reg(2,2))]);
fprintf(fid_log,'Amp Regularization        = %f\n',reg(1,1));
fprintf(fid_log,'Phs Regularization        = %f\n',reg(2,2));
Hess = Hess+reg;

% Calculate update
foo = J'*(Hess\data_diff);
```

```
  foo = foo.*[recon_mesh.kappa;recon_mesh.mua];

  % Update values
  recon_mesh.kappa = recon_mesh.kappa + (foo(1:end/2));
  recon_mesh.mua = recon_mesh.mua + (foo(end/2+1:end));
  recon_mesh.mus = ((1./(3.*recon_mesh.kappa))-recon_mesh.mua)./(1-recon_mesh.g);

  clear foo Hess Hess_norm tmp data_diff G


  % Interpolate optical properties to fine mesh
  [fwd_mesh,recon_mesh] = interpolatep2f(fwd_mesh,recon_mesh);

  % We dont like -ve mua or mus! so if this happens, terminate
  if (any(fwd_mesh.mua<0) | any(fwd_mesh.mus<0))
    disp('---------------------------------');
    disp('-ve mua or mus calculated...not saving solution');
    fprintf(fid_log,'---------------------------------\n');
    fprintf(fid_log,'STOPPING CRITERIA REACHED\n');
    break
  end

  % Filtering if needed!
  if filter_n > 1
    fwd_mesh = mean_filter(fwd_mesh,abs(filter_n));
  elseif filter_n < 0
    fwd_mesh = median_filter(fwd_mesh,abs(filter_n));
  end

  if it == 1
    fid = fopen([output_fn '_mua.sol'],'w');
  else
    fid = fopen([output_fn '_mua.sol'],'a');
  end
  fprintf(fid,'solution %g ',it);
  fprintf(fid,'-size=%g ',length(fwd_mesh.nodes));
  fprintf(fid,'-components=1 ');
  fprintf(fid,'-type=nodal\n');
  fprintf(fid,'%f ',fwd_mesh.mua);
  fprintf(fid,'\n');
  fclose(fid);

  if it == 1
    fid = fopen([output_fn '_mus.sol'],'w');
  else
    fid = fopen([output_fn '_mus.sol'],'a');
  end
  fprintf(fid,'solution %g ',it);
  fprintf(fid,'-size=%g ',length(fwd_mesh.nodes));
  fprintf(fid,'-components=1 ');
  fprintf(fid,'-type=nodal\n');
  fprintf(fid,'%f ',fwd_mesh.mus);
  fprintf(fid,'\n');
  fclose(fid);
end

% close log file!
time = toc;
fprintf(fid_log,'Computation TimeRegularization = %f\n',time);
fclose(fid_log);
```

```
function [data,mesh]=femdata_spnhybrid(mesh,frequency,sp5lim)

% Hybrid SP5-SP1/Diffusion model. sp5lim is size of square region around source to be
solved using sp5
% See Chapter 9 for details

tic

if ischar(mesh)== 1
  mesh = load_mesh(mesh);
end

% modulation frequency
omega = 2*pi*frequency*1e6;


%==================================
% Define regions to be solved by SP5 & SP1

ind=find(abs(mesh.nodes(:,1))<=sp5lim & abs(mesh.nodes(:,2))<=sp5lim);
mesh1.nodes=mesh.nodes(ind,:);
mesh1.elements=delaunay(mesh1.nodes(:,1),mesh1.nodes(:,2));
mesh1.mua=mesh.mua(ind);
mesh1.mus=mesh.mus(ind);
mesh1.g=mesh.g(ind);
mesh1.ri=mesh.ri(ind);
mesh1.c=mesh.c(ind);
mesh1.bndvtx=mesh.bndvtx(ind);
mesh1.dimension=mesh.dimension;
mesh1.region=mesh.region(ind);
mesh1.source=mesh.source;

ind=unique([find(abs(mesh.nodes(:,1))>=sp5lim);find(abs(mesh.nodes(:,2))>=sp5lim)]);
mesh2.nodes=mesh.nodes(ind,:);
mesh2.elements=delaunay(mesh2.nodes(:,1),mesh2.nodes(:,2));
mesh2.mua=mesh.mua(ind);
mesh2.mus=mesh.mus(ind);
mesh2.g=mesh.g(ind);
mesh2.ri=mesh.ri(ind);
mesh2.c=mesh.c(ind);
mesh2.bndvtx=mesh.bndvtx(ind);
mesh2.dimension=mesh.dimension;
mesh2.region=mesh.region(ind);

clear ind

nvtx=length(mesh1.nodes)+length(mesh2.nodes);

r1=length(mesh1.nodes);
r2=length(mesh2.nodes);

ident=[];
ident(:,1)=unique([find(abs(mesh2.nodes(:,1))==sp5lim &
abs(mesh2.nodes(:,2))<=sp5lim);find(abs(mesh2.nodes(:,1))<=sp5lim &
abs(mesh2.nodes(:,2))==sp5lim)]);

ident(:,2)=unique([find(abs(mesh1.nodes(:,1))==sp5lim);find(abs(mesh1.nodes(:,2))==sp5lim)]
);
ident(:,2)=ident(:,2)+r2;
```

```
% ========================================
% Calculate SP5 Mass matrix

  [f1,f2,f3,g1,g2,g3,h1,h2,h3] = ksi_calc_sp5(mesh1);

% Create FEM matrices
if mesh1.dimension==2
[i,j,k1,k3,k5,c,c23,c49,c64,c8,c16,c2_59,c2_16,c2_49,c4_9,B,F1,F2,F3,G1,G2,G3,H1,H2,H3,ib,jb]=
gen_matrices_2d_sp5(mesh1.nodes(:,1:2),...
   sort(mesh1.elements')',...
   mesh1.bndvtx,...
   mesh1.mua,...
   mesh1.mus,...
   mesh1.g,...
   f1,...
   f2,...
   f3,...
   g1,...
   g2,...
   g3,...
   h1,...
   h2,...
   h3,...
   mesh1.c,...
   omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
C  = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23= sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49= sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64= sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16= sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));
nz_i = length(nonzeros(ib));
F1 = sparse(ib(1:nz_i),jb(1:nz_i),F1(1:nz_i));
F2 = sparse(ib(1:nz_i),jb(1:nz_i),F2(1:nz_i));
F3 = sparse(ib(1:nz_i),jb(1:nz_i),F3(1:nz_i));
G1 = sparse(ib(1:nz_i),jb(1:nz_i),G1(1:nz_i));
G2 = sparse(ib(1:nz_i),jb(1:nz_i),G2(1:nz_i));
G3 = sparse(ib(1:nz_i),jb(1:nz_i),G3(1:nz_i));
H1 = sparse(ib(1:nz_i),jb(1:nz_i),H1(1:nz_i));
H2 = sparse(ib(1:nz_i),jb(1:nz_i),H2(1:nz_i));
H3 = sparse(ib(1:nz_i),jb(1:nz_i),H3(1:nz_i));

elseif mesh1.dimension==3
[i,j,k1,k3,k5,c,c23,c49,c64,c8,c16,c2_59,c2_16,c2_49,c4_9,B,F1,F2,F3,G1,G2,G3,H1,H2,H3,ib,jb]=
gen_matrices_3d_sp5(mesh1.nodes,...
   sort(mesh1.elements')',...
   mesh1.bndvtx,...
   mesh1.mua,...
   mesh1.mus,...
   mesh1.g,...
   f1,...
   f2,...
   f3,...
  g1,...
  g2,...
  g3,...
  h1,...
  h2,...
  h3,...
  mesh1.c,...
  omega);

nz_i=length(nonzeros(i));
K1 = sparse(i(1:nz_i),j(1:nz_i),k1(1:nz_i));
K3 = sparse(i(1:nz_i),j(1:nz_i),k3(1:nz_i));
K5 = sparse(i(1:nz_i),j(1:nz_i),k5(1:nz_i));
C  = sparse(i(1:nz_i),j(1:nz_i),c(1:nz_i));
C23= sparse(i(1:nz_i),j(1:nz_i),c23(1:nz_i));
C49= sparse(i(1:nz_i),j(1:nz_i),c49(1:nz_i));
C64= sparse(i(1:nz_i),j(1:nz_i),c64(1:nz_i));
C8 = sparse(i(1:nz_i),j(1:nz_i),c8(1:nz_i));
C16= sparse(i(1:nz_i),j(1:nz_i),c16(1:nz_i));
C2_59 = sparse(i(1:nz_i),j(1:nz_i),c2_59(1:nz_i));
C2_16 = sparse(i(1:nz_i),j(1:nz_i),c2_16(1:nz_i));
C2_49 = sparse(i(1:nz_i),j(1:nz_i),c2_49(1:nz_i));
C4_9 = sparse(i(1:nz_i),j(1:nz_i),c4_9(1:nz_i));
B = sparse(i(1:nz_i),j(1:nz_i),B(1:nz_i));
F1 = sparse(i(1:nz_i),j(1:nz_i),F1(1:nz_i));
F2 = sparse(i(1:nz_i),j(1:nz_i),F2(1:nz_i));
F3 = sparse(i(1:nz_i),j(1:nz_i),F3(1:nz_i));
G1 = sparse(i(1:nz_i),j(1:nz_i),G1(1:nz_i));
G2 = sparse(i(1:nz_i),j(1:nz_i),G2(1:nz_i));
G3 = sparse(i(1:nz_i),j(1:nz_i),G3(1:nz_i));
H1 = sparse(i(1:nz_i),j(1:nz_i),H1(1:nz_i));
H2 = sparse(i(1:nz_i),j(1:nz_i),H2(1:nz_i));
H3 = sparse(i(1:nz_i),j(1:nz_i),H3(1:nz_i));

end



clear j* k* c* nz*

% Add complex component to absorption moments due to
% frequency dependence
C=C+B;
C23=C23+B;
C49=C49+B;
C64=C64+B;
C8=C8+B;
C16=C16+B;
C2_59=C2_59+B;
C2_16=C2_16+B;
C2_49=C2_49+B;
C4_9=C4_9+B;
Cs=C16+C2_49;


M1=K1+C+F1;
M2=K3+C49+C2_59+F2;
M3=K5+C64+C2_16+C4_9+F3;

MASS_sp5=[M1 -(C23-G1) (C8-H1);-(C23-G2) M2 -(C16+C2_49-H2);
   (C8-G3) -(C16+C2_49-H3) M3];
```

```
%===================================
%Calculate diffusion Mass matrix
clear K1 B F1 C
% Calculate boundary condition coefficients
[f1] = ksi_calc_sp1(mesh2);

% Create FEM matrices
if mesh2.dimension == 2
  [i,j,k1,c,B,F1,ib,jb]=gen_matrices_2d_sp1(mesh2.nodes(:,1:2),...
    sort(mesh2.elements')',...
    mesh2.bndvtx,...
    mesh2.mua,...
    mesh2.mus,...
    mesh2.g,...
    f1,...
    mesh2.c,...
    omega);

nz_i = nonzeros(i);
K1 = sparse(i(1:length(nz_i)),j(1:length(nz_i)),k1(1:length(nz_i)));
C = sparse(i(1:length(nz_i)),j(1:length(nz_i)),c(1:length(nz_i)));
B = sparse(i(1:length(nz_i)),j(1:length(nz_i)),B(1:length(nz_i)));
nz_i = nonzeros(ib);
F1 = sparse(ib(1:length(nz_i)),jb(1:length(nz_i)),F1(1:length(nz_i)));

elseif mesh2.dimension == 3
     [i,j,k1,c,B,F1,ib,jb]=gen_matrices_3d_sp1(mesh2.nodes,...
    sort(mesh2.elements')',...
   mesh2.bndvtx,...
    mesh2.mua,...
    mesh2.mus,...
    mesh2.g,...
    f1,...
    mesh2.c,...
    omega);
nz_i = nonzeros(i);
K1 = sparse(i(1:length(nz_i)),j(1:length(nz_i)),k1(1:length(nz_i)));
C = sparse(i(1:length(nz_i)),j(1:length(nz_i)),c(1:length(nz_i)));
B = sparse(i(1:length(nz_i)),j(1:length(nz_i)),B(1:length(nz_i)));
F1 = sparse(i(1:length(nz_i)),j(1:length(nz_i)),F1(1:length(nz_i)));

end

clear i j k1 c f1 g1 ib jb nz_i

MASS_sp1 = K1 + C + B +F1;
%====================================================
% Combine SP1 and SP3 Mass matricies
[i,j]=size(MASS_sp1);
[ii,jj]=size(MASS_sp5);
MASS=[MASS_sp1 sparse(i,jj);sparse(ii,j) MASS_sp5];
%=======================================================

% Calculate the RHS (the source vectors. For simplicity, we are
% just going to use a Gaussian Source, The width of the Gaussian is
% changeable (last argument). The source is assumed to have a
% complex amplitude of complex(cos(0.15),sin(0.15));

[nnodes,junk]=size(mesh1.nodes);
[nsource,junk]=size(mesh.source.coord);
qvec = spalloc(nnodes,nsource,nsource*100);
if mesh1.dimension == 2
  for i = 1 : nsource
```

```
    if mesh1.source.fwhm(i) == 0
      qvec(:,i) = gen_source_point(mesh1,mesh1.source.coord(i,1:2));
    else
     qvec(:,i) = gen_source(mesh1.nodes(:,1:2),...
          sort(mesh1.elements')',...
          mesh1.dimension,...
          mesh1.source.coord(i,1:2),...
          mesh1.source.fwhm(i));
    end
  end
elseif mesh1.dimension == 3
  for i = 1 : nsource
    if mesh1.source.fwhm(i) == 0
      qvec(:,i) = gen_source_point(mesh1,mesh1.source.coord(i,1:3));
    else
    qvec(:,i) = gen_source(mesh1.nodes,...
          sort(mesh1.elements')',...
          mesh1.dimension,...
          mesh1.source.coord(i,:),...
          mesh1.source.fwhm(i));
    end
  end
end

clear junk i nnodes nsource;

qvec=sparse([zeros(length(mesh2.nodes),1);qvec;-(2/3)*qvec;(8/15)*qvec]);

%============================================
% Adjust Mass matrix to account for shared nodes between 2 mesh regions.
 [MASS]=link_nodes_sp5(MASS,ident,r1);

% =====================================
% Optimise MASS matrix

 [MASS_opt,Q_opt,invsort]=optimise(MASS,qvec);


% ======================================
%
 phi_all=get_field(MASS_opt,mesh,Q_opt);

% Re-order elements
 phi_all=phi_all(invsort,:);

% Extract composite moments of phi
phi_sp1=phi_all(1:r2,:);
phi1=phi_all(r2+1:(r2+r1),:);
phi2=phi_all((r2+r1)+1:(r2+2*r1),:);
phi3=phi_all((r2+2*r1)+1:end,:);

phi_sp5=phi1-(2/3).*phi2+(8/15).*phi3;

data.phi=[phi_sp5;phi_sp1];
data.phi1=phi_sp1;
data.phi5=phi_sp5;
clear qvec* M* C;

% Calculate boundary data

% For 2d
mesh_temp.nodes=[mesh1.nodes;mesh2.nodes];
mesh_temp.elements=delaunay(mesh_temp.nodes(:,1),mesh_temp.nodes(:,2));
```

```
[data.complex]=get_boundary_data(mesh,data.phi);

% Map complex data to amplitude and phase
data.amplitude = abs(data.complex);

data.phase = atan2(imag(data.complex),...
        real(data.complex));
data.phase(find(data.phase<0)) = data.phase(find(data.phase<0)) + (2*pi);
data.phase = data.phase*180/pi;

data.paa = [data.amplitude data.phase];
```

```
function [MASS_hybrid]=link_nodes_sp5(MASS,ident,r1,r2)

% Links Mass matrix elements to account for nodes shared between 2 regions

  MASS_hybrid=MASS;
ind=[ident(:,2)-r1 ident(:,1)+r2 ident(:,1)+r1+r2 ident(:,1)+(2*r1)+r2];

for i=1:length(ident)
  MASS_hybrid(ind(i,1),:)=MASS(ind(i,1),:)+MASS(ind(i,2),:)-(2/3).*MASS(ind(i,3),:)+...
                                   (8/15).*MASS(ind(i,4),:);

  MASS_hybrid(ind(i,2),:)=0;
  MASS_hybrid(ind(i,3),:)=0;
  MASS_hybrid(ind(i,4),:)=0;

  MASS_hybrid(ind(i,2),ind(i,2))=1;
  MASS_hybrid(ind(i,3),ind(i,3))=1;
  MASS_hybrid(ind(i,4),ind(i,4))=1;

  MASS_hybrid(ind(i,2),ind(i,1))=-1;
  MASS_hybrid(ind(i,2),ind(i,3))=2/3;
  MASS_hybrid(ind(i,2),ind(i,4))=-8/15;

  MASS_hybrid(ind(i,3),ind(i,1))=3/2;
  MASS_hybrid(ind(i,3),ind(i,2))=-3/2;
  MASS_hybrid(ind(i,3),ind(i,4))=-4/5;

  MASS_hybrid(ind(i,4),ind(i,1))=-15/8;
  MASS_hybrid(ind(i,4),ind(i,2))=15/8;
  MASS_hybrid(ind(i,4),ind(i,3))=5/4;

end
```

References Bibliography

Abookasis, D., C. C. Lay, M. S. Mathews, M. E. Linskey, R. D. Frostig and B. Tromberg (2009). "Imaging cortical absorption, scattering and hemodynamic response during ischemic stroke using spatially modulated near-infrared illumination." Journal of Biomedical Optics **14**(2).

Alexandrakis, G., T. J. Farrel and M. S. Patterson (2000). "Monte Carlo diffusion hybrid model for photon migration in a two-layer turbid medium in the frequency domain." Appl. Opt. **39**(13).

Alexandrakis, G., F. Rannou and A. F. Chatziioannou (2005). "Tomographic bioluminescence imaging by use of a combined optical-PET (OPET) system: a computer simulation feasibility study." Phys. Med. Biol **50**: 4225:4241.

Amestoy, P. R., Enseeiht-Irit, T. A. Davis and I. S. Duff (2004). "Algorithm 837: AMD, an approximate minimum degree ordering algorithm." ACM Trans. Math. Softw. **30**(3): 381-388.

Arridge, S. (1995a). "Photon-measurement density functions. Part 1: Analytical Forms." Applied Optics **34**(31).

Arridge, S., J. C. Hebden, M. Schweiger, F. E. W. Schmidt, M. E. Fry, E. M. C. Hillman, H. Dehghani and D. T. Delpy (2000a). "A Method for Three-Dimensional Time-Resolved Optical Tomography." International Journal of Imaging Systems and Technology **11**(1): 2-11.

Arridge, S. R. (1995b). "Photon-measurement density functions. Part I: Analytical forms." Appl. Opt. **34**(31): 7395-7409.

Arridge, S. R. (1999). "Optical tomography in medical imaging." Inverse Problems **15**(2): R41-R93.

Arridge, S. R., H. Dehghani, M. Schweiger and E. Okada (2000b). "The finite element model for the propagation of light in scattering media: A direct method for domains with nonscattering regions." Medical Physics **27**(1): 252-264.

Arridge, S. R. and J. C. Hebden (1997). "Optical imaging in medicine 2: Modelling and reconstruction." Physics in Medicine and Biology **42**(5): 841-853.

Arridge, S. R. and W. R. B. Lionheart (1998). "Nonuniqueness in diffusion-based optical tomography." Optics Letters **23**(11): 882-884.

Arridge, S. R. and M. Schweiger (1995). "Photon-Measurement Density-Functions .2. Finite-Element-Method Calculations." Applied Optics **34**(34): 8026-8037.

Arridge, S. R., M. Schweiger, M. Hiraoka and D. T. Delpy (1993). "A Finite-Element Approach for Modeling Photon Transport in Tissue." Medical Physics **20**(2): 299-309.

Aydin, E. D., C. R. E. de Oliveira and A. J. H. Goddard (2002). "A comparison between transport and diffusion calculations using a finite element-spherical harmonics radiation transport method." Medical Physics **29**(9): 2013-2023.

Aydin, E. D., C. R. E. de Oliveira and A. J. H. Goddard (2004). "A finite element-spherical harmonics radiation transport model for photon migration in turbid media." Journal of Quantitative Spectroscopy and Radiative Transfer **84**(3): 247-260.

Becker, W. (2005). Advanced Time-Correlated Single Photon Counting Techniuqes. Berlin, Springer.

Benaron, D. A., S. R. Hintz, A. Villringer, D. Boas, A. Kleinschmidt, J. Frahm, C. Hirth, H. Obrig, J. C. van Houten, E. L. Kermit, W.-F. Cheong and D. K. Stevenson (2000). "Noninvasive Functional Imaging of Human Brain Using Light." J Cereb Blood Flow Metab **20**(3): 469-477.

Bluestone, A., G. Abdoulaev, C. Schmitz, R. Barbour and A. Hielscher (2001). "Three-dimensional optical tomography of hemodynamics in the human head." Opt. Express **9**(6): 272-286.

Bluestone, A. Y., M. Stewart, J. Lasker, G. S. Abdoulaev and A. H. Hielscher (2004a). "Three-dimensional optical tomographic brain imaging in small animals, part 1: hypercapnia." Journal of Biomedical Optics **9**(5): 1046-1062.

Bluestone, A. Y., M. Stewart, B. Lei, I. S. Kass, J. Lasker, G. S. Abdoulaev and A. H. Hielscher (2004b). "Three-dimensional optical tomographic brain imaging in small animals, part 2: Unilateral carotid occlusion." Journal of Biomedical Optics **9**(5): 1063-1073.

Boas, D., J. Culver, J. Stott and A. Dunn (2002). "Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head." Opt. Express **10**(3): 159-170.

Boas, D. A., T. Gaudette and S. R. Arridge (2001). "Simultaneous imaging and optode calibration with diffuse optical tomography." Optics Express **8**(5): 263-270.

Boas, D. A., M. A. O'Leary, B. Chance and A. G. Yodh (1994). "Scattering of Diffuse Photon Density Waves by Spherical Inhomogeneities Within Turbid Media: Analytic Solution and Applications." Proceedings of the National Academy of Sciences of the United States of America **91**(11): 4887-4891.

Bolin, F. P., L. E. Preuss, R. C. Taylor and R. J. Ference (1989). "Refractive index of some mammalian tissues using a fiber optic cladding method." Appl. Opt. **28**(12): 2297-2303.

Brantley, P. S. and E. W. Larsen (2000). "The simplified P_3 approximation." Nuclear Science and Engineering **134**(1).

Brebbia, C. A. and J. Dominguez (1992). Boundary Elements: An Introductory Course. Southampton, UK, Computational Mechanics Publications.

Bright, R. (1831). Diseases of the Brain and Nervous System. London, Longman.

Brooksby, B., H. Dehghani, K. Vishwanath, B. W. Pogue and K. D. Paulsen (2003). Internal refractive index changes affect light transport in tissue. Optical Tomography and Spectroscopy of Tissue V. B. Chance, R. R. Alfano, B. J. Tromberg, M. Tamura and E. M. SevickMuraca. Bellingham, Spie-Int Soc Optical Engineering. **4955:** 296-304.

Brooksby, B., S. Jiang, H. Dehghani, B. W. Pogue, K. Paulsen, C. Kogel, M. Doyley, J. B. Weaver and S. P. Poplack (2004). "Magnetic resonance-guided near-infrared tomography of the breast." Review of Scientific Instruments **75**(12).

Brooksby, B., S. Jiang, H. Dehghani, B. W. Pogue, K. Paulsen, J. Weaver, C. Kogel and S. P. Poplack (2005). "Combining near-infrared tomography and magnetic resonance imaging to study in vivo breast tissue: implementation of a Laplacian-type regularization to incorporate magnetic resonance structure." Journal of Biomedical Optics **10**(5).

Buzug, T. M. (2008). Computed Tomography: From Photon Statistics to Modern Cone-Beam CT. Berlin, Springer-Verlag.

Calvetti, D., S. Morigi, L. Reichel and F. Sgallari (2000). "Tikhonov regularization and the L-curve for large discrete ill-posed problems." Journal of Computational and Applied Mathematics **123**(1-2).

Case, K. M. and P. F. Zweifel (1967). Linear Transport Theory. Reading, MA, Addison-Wesley.

Celentano, L., P. Laccetti, R. Liuzzi, G. Mettivier, M. C. Montesi, M. Autiero, P. Riccio, G. Roberti, P. Russo and M. Salvatore (2003). "Preliminary tests of a prototype system for optical and radionuclide imaging in small animals." IEEE Trans Nucl Sci. **50**(5).

Chance, B., M. Cope, E. Gratton, N. Ramanujam and B. Tromberg (1998). "Phase measurement of light absorption and scatter in human tissue." Review of Scientific Instruments **69**.

Chang, J., H. L. Graber and R. L. Barbour (1997). "Imaging of fluorescence in highly scattering media." IEEE Trans Biomed Eng **44**.

Chaudhari, A. J., F. Darvas, J. R. Bading, R. A. Moats, P. S. Conti, D. J. Smith, S. R. Cherry and R. M. Leahy (2005). "Hyperspectral and multispectral bioluminescence optical tomography for small animal imaging." Physics in Medicine & Biology **50**.

Chen, Y., G. Zheng, Z. H. Zhang, D. Blessington, M. Zhang, H. Li, Q. Liu, L. Zhou, X. Intes, S. Achilefu and B. Chance (2003). "Metabolism-enhanced tumor localization by fluorescence imaging: in vivo animal studies." Optics Letters **28**(21).

Cheong, W. F., S. A. Prahl and A. J. Welch (1990). "A Review of the Optical Properties of Biological Tissues." IEEE Journal of Quantum Electronics **26**(12).

Chernomordik, V., D. W. Hattery, D. Grosenick, H. Wabnitz, H. Rinneberg, K. T. Moesta, P. M. Schlag and A. Gandjbakhche (2001). "Quantification of optical properties of a breast tumor using random walk theory." Journal of Biomedical Optics **7**(1).

Choe, R., A. Corlu, K. Lee, T. Durduran, S. D. Konecky, M. Grosicka-Koptyra, S. R. Arridge, B. J. Czerniecki, D. L. Fraker, A. DeMichele, B. Chance, M. A. Rosen and A. G. Yodh (2005). "Diffuse optical tomography of breast cancer during neoadjuvant chemotherapy: A case study with comparison to MRI." Medical Physics **32**: 1128-1139.

Chu, M. K., A. D. Klose and H. Dehghani (2008). Light Transport in Soft Tissue Based on Simplified Spherical Harmonics Approximation to Radiative Transport Equation. Biomedical Optics, Optical Society of America.

Ciolini, R., Coppa, G. G. M., Montagnini, B., Ravetto, P. (2002). "Simplified PN and AN methods in neutron transport." Prog. Nucl. Energy **40**(2).

Cohen, M. F. and J. R. Wallace (1993). Radiosity and Realistic Image Synthesis. London, Academic.

Cong, A. X. and G. Wang (2005). "Multispectral Bioluminescence Tomography: Method and Simulation." Int. J. Biomed. Imaging **2006**.

Corlu, A., T. Durduran, R. Choe, M. Schweiger, E. M. C. Hillman, S. R. Arridge and A. G. Yodh (2003). "Uniqueness and wavelength optimization in continuous-wave multispectral diffuse optical tomography." Opt. Lett. **28**(23): 2339-2341.

Culver, J. P., R. Choe, M. J. Holboke, L. Zubkov, T. Durduran, A. Slemp, V. Ntziachristos, B. Chance and A. G. Yodh (2003a). "Three-dimensional diffuse optical tomography in the parallel plane transmission geometry: Evaluation of a hybrid frequency domain/continuous wave clinical system for breast imaging." Medical Physics **30**(2): 235-247.

Culver, J. P., T. Durduran, D. Furuya, C. Cheung, J. H. Greenberg and A. G. Yodh (2003b). "Diffuse Optical Tomography of Cerebral Blood Flow, Oxygenation, and Metabolism in Rat During Focal Ischemia." J Cereb Blood Flow Metab **23**(8): 911-924.

Culver, J. P., A. M. Siegel, J. J. Stott and D. A. Boas (2003c). "Volumetric diffuse optical tomography of brain activity." Opt. Lett. **28**(21): 2061-2063.

Curling, T. B. (1843). A Practical Treatise on the Diseases of the Testis and of the Spermatic Cord and Scrotum. London, Samuel Highley.

Curry, T. S., J. E. Dowdey and R. C. Murry (1990). Christensen's Physics of Diagnostic Radiology. Philadelphia, Lippincott Williams & Wilkins.

Cutler, M. (1929). "Transillumination as an aid in the diagnosis of breast lesions." Surg. Gynecol. Obstet **48**: 721-728.

Davis, S. C., B. W. Pogue, R. Springett, C. Leussier, P. Mazurkewitz, S. B. Tuttle, S. L. Gibbs-Strauss, S. Jiang, H. Dehghani and K. Paulsen (2008). "Magnetic resonance-coupled fluorescence tomography scanner for molecular imaging of tissue." Review of Scientific Instruments **79**.

Dehghani, H., S. R. Arridge, M. Schweiger and D. T. Delpy (2000). "Optical tomography in the presence of void regions." Journal of the Optical Society of America a-Optics Image Science and Vision **17**(9): 1659-1670.

Dehghani, H., B. Brooksby, K. Vishwanath, B. W. Pogue and K. D. Paulsen (2003a). "The effects of internal refractive index variation in near-infrared optical tomography: a finite element modelling approach." Physics in Medicine and Biology **48**(16): 2713-2727.

Dehghani, H., M. E. Eames, P. K. Yalavarthy, S. C. Davis, S. Srinivasan, C. M. Carpenter, B. W. Pogue and K. D. Paulsen (2008). "Near infrared optical tomography using NIRFAST: Algorithm for numerical model and image reconstruction." Communications in Numerical Methods in Engineering.

Dehghani, H., B. W. Pogue, S. Jiang, B. Brooksby and K. Paulsen (2003b). "Three-dimensional optical tomography: resolution in small-object imaging." Appl. Opt. **42**(16).

Dehghani, H., B. W. Pogue, S. P. Poplack and K. D. Paulsen (2003c). "Multiwavelength three-dimensional near-infrared tomography of the breast: initial simulation, phantom, and clinical results." Applied Optics **42**(1): 135-145.

Dogdas, B., D. Stout, A. Chatziioannou and R. M. Leahy (2007). "Digimouse: a 3D whole body mouse atlas from CT and cryosection data." Physics in Medicine & Biology **52**.

DomÌnguez, J. B. and Y. BÈrubÈ-LauziËre (2010). "Diffuse light propagation in biological media by a time-domain parabolic simplified spherical harmonics approximation with ray-divergence effects." Appl. Opt. **49**(8): 1414-1429.

Dorn, O. (1998). "A transport-backtransport method for optical tomography." Inverse Problems **14**(5): 1107.

Durduran, T. and et al. (2002). "Bulk optical properties of healthy female breast tissue." Physics in Medicine and Biology **47**(16): 2847.

Eames, M. E. and H. Dehghani (2008). "Wavelength dependence of sensitivity in spectral diffuse optical imaging: effect of normalization on image reconstruction." Optics Express **16**(22): 17780-17791.

Eames, M. E., B. W. Pogue, P. K. Yalavarthy and H. Dehghani (2007). "An efficient Jacobian reduction method for diffuse optical image reconstruction." Opt. Express **15**(24): 15908-15919.

Eda, H., I. Oda, Y. Ito, Y. Wada, Y. Okawa, Y. Tsunazawa, M. Takada, Y. Tsuchiya, Y. Yamashita, M. Oda, A. Sassaroli, Y. Yamada and M. Tamura (1999).

"Multichannel time-resolved optical tomographic imaging system." Review of Scientific Instruments **70**(9).

Edinger, M., T. J. Sweeney, A. A. Tucker, A. B. Olomu, R. S. Negrin and C. H. Contag (1999). "Noninvasive Assessment of Tumor Cell Proliferation in Animal Models." Neoplasia **1**(4).

Edwards, D. K., J. T. Gier, K. E. Nelson and R. D. Roddick (1961). "Integrating Sphere for Imperfectly Diffuse Samples." J. Opt. Soc. Am. **51**(11): 1279-1288.

Elisee, J., A. Gibson and S. Arridge (2010). "Combination of Boundary Element Method and Finite Element Method in Diffuse Optical Tomography." IEEE Trans Biomed Eng(in press).

Enfield, L. C., A. P. Gibson, N. L. Everell, D. T. Delpy, M. Schweiger, S. R. Arridge, C. Richardson, M. Keshtagar, M. Douek and J. C. Hebden (2007). "Three-dimensional time-resolved optical mammography of the uncompressed breast." Applied Optics **46**: 3628-3638.

Fang, Q. and D. A. Boas (2009). "Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units." Opt. Express **17**(22): 20178-20190.

Firbank, M., S. R. Arridge, M. Schweiger and D. T. Delpy (1996). "An investigation of light transport through scattering bodies with non-scattering regions." Physics in Medicine and Biology **41**(4): 767-783.

Gao, F., H. Niu, H. Zhao and H. Zhang (1998). "The forward and inverse models in time-resolved optical tomography imaging and their finite-element method solutions." Image and Vision Computing **16**(9-10): 703-712.

Gelbard, E. M. (1968). Spherical harmonic methods. Computing methods in reactor physics. New York, Gordon and Breach.

Ghosh, N., S. K. Mohanty, S. K. Majumder and P. K. Gupta (2001). "Measurement of Optical Transport Properties of Normal and Malignant Human Breast Tissue." Appl. Opt. **40**(1): 176-184.

Gibson, A. P., J. C. Hebden and S. R. Arridge (2005). "Recent advances in diffuse optical imaging." Physics in Medicine and Biology **50**(4): R1-R43.

Glasser, O. (1993). Wilhelm Conrad Roentgen And the Early History of the Roentgen Rays. San Francisco, CA, Norman Publishing.

Grubb, R. L., M. E. Raichle, J. O. Eichling and M. H. Gado (1977). "Effects of subarachnoid hemorrhage on cerebral blood volume, blood flow and oxygen utilization in humans." Journal of Neurosurgery **46**(4).

Hayashi, T., Y. Kashio and E. Okada (2003). "Hybrid Monte Carlo-diffusion method for light propagation in tissue with a low-scattering region." Appl. Opt. **42**(16).

Hebden, J. C. (2003). "Advances in optical imaging of the newborn infant brain." Psychophysiology **40**(4).

Hebden, J. C., S. R. Arridge and D. T. Delpy (1997). "Optical imaging in medicine 1: Experimental techniques." Physics in Medicine and Biology **42**(5): 825-840.

Hebden, J. C., A. Gibson, T. Austin, R. M. Yusof, N. Everdell, D. T. Delpy, S. R. Arridge, J. H. Meek and J. S. Wyatt (2004). "Imaging changes in blood volume and oxygenation in the newborn infant brain using three-dimensional optical tomography." Physics in Medicine and Biology **49**(7): 1117-1130.

Hebden, J. C., A. Gibson, R. M. Yusof, N. Everdell, E. M. C. Hillman, D. T. Delpy, S. R. Arridge, T. Austin, J. H. Meek and J. S. Wyatt (2002). "Three-dimensional optical tomography of the premature infant brain." Physics in Medicine and Biology **47**(23): 4155-4166.

Hielscher, A. (1997). Model-based iterative image reconstruction for photon migration tomography. Computational, Experimental and Numerical Methods for Solving Ill-posed Inverse Imaging Problems: Medical and Nonmedical Applications, SPIE-The International Society of Optical Engineering.

Hielscher, A. (2005). "Optical tomographic imaging of small animals." Medical Physics **32**(6): 2096-2096.

Hielscher, A. H., R. E. Alcouffe and R. L. Barbour (1998). "Comparison of finite-difference transport and diffusion calculations for photon migration in homogeneous and heterogeneous tissues." Physics in Medicine and Biology **43**(5): 1285-1302.

Hielscher, A. H., A. D. Klose and K. M. Hanson (1999). "Gradient-based iterative image reconstruction scheme for time- resolved optical tomography." IEEE Transactions On Medical Imaging **18**(3): 262-271.

Hillman, E. M. C. (2002). Experimental and theoretical investigation of near infrared tomographic imaging methods and clinical applications. Department of Medical Physics and Bioengineering. London, University College London. **Doctor of Philosophy**.

Hintz, S. R., D. A. Benaron, J. P. van Housten, J. L. Duckworth, F. W. Liu, S. D. Spilman, D. K. Stevenson and C. W. F. (1998). "Stationary headband for clinical time-of flight optical imaging at the bedside." Photochem Photobiol **68**(3).

Hintz, S. R., C. W. F., J. P. van Housten, D. K. Stevenson and D. A. Benaron (1999). "Bedside imaging of intracranial hemorrhage in the neonate using light: comparison with ultrasound, computed tomography and magnetic resonance imaging." Pediatr Res **45**(1).

Hiraoka, M., M. Firbank, M. Essenpreis, M. Cope, S. R. Arridge, P. Vanderzee and D. T. Delpy (1993). "A Monte-Carlo Investigation of Optical Pathlength in Inhomogeneous Tissue and Its Application to near-Infrared Spectroscopy." Physics in Medicine and Biology **38**(12): 1859-1876.

Holdsworth, D. W. and M. M. Thornton (2002). "Micro-CT in small animal and specimen imaging." Trends in Biotechnology **20**(8): S34-S39.

Huber, J. S., D. Sudar and W. W. Moses (2001). Conceptual design of a dual modality optical and radionuclide imaging camera. High Resolution Imaging in Small Animals. Rockville, MD.

Iyer, M., M. Berenji, N. S. Templeton and S. S. Gambhir (2002). "Noninvasive Imaging of Cationic Lipid-Mediated Delivery of Optical and PET Reporter Genes in Living Mice." Molecular Therapy **6**.

Jacques, S. L., A. Alter and S. A. Prahl (1987). "Angular Dependenceof HeNe Laser Light Scattering by Human Dermis." Lasers Life Sci. **1**(4): 309-334.

Jacquez, J. A. and H. F. Kuppenheim (1955). "Theory of the Integrating Sphere." J. Opt. Soc. Am. **45**(6): 460-466.

Jiang, H. (1998). "Frequency-domain fluorescent diffusion tomography: a finite-element-based algorithm and simulations." Applied Optics **37**(22): 5337-5343.

Jiang, H. and K. D. Paulsen (1995). A finite element based higher-order diffusion approximation of light propagation in tissues. Proc SPIE: Optical Tomography, Photon migration and spectroscopy of tissue and model media, San Jose, CA.

Jobsis, F. F. (1977). "Noninvasive, infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters." Science **198**(4323).

Josef, J. A. (1996). A Simplified Spherical Harmonic Method for Coupled Electron-Photon Transport Calculations. The Graduate School, Pensylvania State University. **Doctor of Philosophy**.

Joshi, A., W. Bangerth and E. M. Sevick-Muraca (2006). "Non-contact fluorescence optical tomography with scanning patterned illumination." Opt. Express **14**(14): 6516-6534.

Kepshire, D., S. C. Davis, H. Dehghani, K. Paulsen and B. W. Pogue (2008). "Fluorescence tomography characterization for sub-surface imaging with protoporphyrin IX." Opt. Express **16**(12).

Kienle, A., F. K. Forster and R. Hibst (2001). "Influence of the phase function on determination of the optical properties of biological tissue by spatially resolved reflectance." Opt. Lett. **26**(20): 1571-1573.

Kim, K. H., U. J. Netz, J. Beuthan and A. Hielscher (2008). "Optimal source-modulation frequencies for transport-theory-based optical tomography of small-tissue volumes." Optics Express **16**(22).

Klose, A. D. and B. J. Beattie (2008). Bioluminescence tomography with SP3 equations. Biomedical Optics Topical Meeting, Florida, The Optical Society of America.

Klose, A. D., B. J. Beattie, H. Dehghani, L. Vider, C. Le, V. Ponomarev and R. Blasberg (2010). "*In Vivo* bioluminescence tomogtaphy with a blocking-off finite-difference SP3 method and MRI/CT coregistration." Medical Physics **37**(1).

Klose, A. D. and E. W. Larsen (2003). "Fluorescence tomography with simulated data based on the equation of radiative transfer." Optics Letters **28**(12): 1019-1021.

Klose, A. D. and E. W. Larsen (2006). "Light transport in biological tissue based on the simplified spherical harmonics equations." Journal of Computational Physics **220**(1): 441-470.

Klose, A. D., U. Netz, J. Beuthan and A. H. Heilscher (2002). "Optical tomography using the time-independent equation of radiative transfer -- Part 1: forward model." Journal of Quantitative Spectroscopy and Radiative Transfer **72**(5): 691-713.

Klose, A. D., V. Ntziachristos and A. H. Hielscher (2005). "The inverse source problem based on the radiative transfer equation in optical molecular imaging." Journal of Computational Physics **202**(1): 323-345.

Konecky, S. D., R. Choe, A. Corlu, K. Lee, R. Wiener, S. M. Srinivas, J. R. Saffer, R. Freifelder, J. S. Karp, N. Hajjioui, F. Azar and A. Yodh (2008). "Comparison of diffuse optical tomography of human breast with whole-body and breast only positron emission tomography." Medical Physics **35**(2).

Kotiluoto, P. H., P. (2001). "Application of the new MultiTrans SP3 radiation transport code in BNCT dose planning." Medical Physics **28**(9).

Kumar, A. T. N., S. B. Raymond, A. Dunn, B. J. Bacskai and D. Boas (2008). "A Time Domain Fluorescence Tomography System for Small Animal Imaging." IEEE Transactions On Medical Imaging **27**(8).

Kuo, C., O. Coquoz, T. L. Troy, H. Xu and B. W. Rice (2007). "Three-dimensional reconstruction of in vivo bioluminescent sources based on multispectral imaging." Journal of Biomedical Optics **12**(2): 024007-024012.

Larsen, E. W. (1993). Asymptotic derivation of the multigroup P[sub 1] and SP[sub N] equations, United States.

Larsen, E. W., J. E. Morel and J. M. McGhee (1996). "Asymptotic derivation of the multigroup P-1 and simplified P-N equations with anisotropic scattering." Nuclear Science and Engineering **123**(3): 328-342.

Larsen, E. W., Thoemmes, G., Klar, A., Deaid, M., Goetz, T. (2002). "Simplified PN approximations to the equations of radiative heat transfer and applications." J. Comput. Phys **183**.

Leblond, F., H. Dehghani, D. Kepshire and B. W. Pogue (2009). "Early-photon fluorescence tomography: spatial resolution improvements and noise stability considerations." J. Opt. Soc. Am. **26**(6).

Lee, J. and E. M. Sevick-Muraca (2002). "Three-dimensional fluorescence enhanced optical tomography using referenced frequency-domain migration measurements at emission and excitation wavelengths." J. Opt. Soc. Am. **19**.

Li, A., Q. Zhang, J. P. Culver, E. L. Miller and D. A. Boas (2004). "Reconstructing chromosphere concentration images directly by continuous-wave diffuse optical tomography." Opt. Lett. **29**(3): 256-258.

Liao, S. M., N. M. Gregg, B. R. White, B. W. Zeff, K. A. Bjerkaas, T. E. Inder and J. P. Culver "Neonatal hemodynamic response to visual cortex activity: high-density near-infrared spectroscopy study." Journal of Biomedical Optics **15**(2): 026010-026019.

Liemert, A., Kienle, A. (2010). "Analytical solutions of the simplified spherical harmonics equations." Optics Letters **35**(20).

Luker, G. D., V. Sharma, C. M. Pica, J. L. Dahlheimer, W. Li, J. Ochesky, C. E. Ryan, H. Piwinca-Worms and D. Piwinca-Worms (2002). "Noninvasive imaging of protein-protein interactions in living animals." Proceedings of the National Academy of Sciences of the United States of America **99**(10).

Martelli, F., A. Sassaroli, Y. Yamada and G. Zaccanti (2002). "Analytical approximate solutions of the time-domain diffusion equation in layered slabs." J. Opt. Soc. Am. A **19**(1): 71-80.

McBride, T. O., B. W. Pogue, S. Jiang, U. L. Osterberg and K. D. Paulsen (2001). "A parallel-detection frequency-domain near-infrared tomography system for hemoglobin imaging of the breast in vivo." Review of Scientific Instruments **72**(3): 1817-1824.

Mourant, J. R., T. Fuselier, J. Boyer, T. M. Johnson and I. J. Bigio (1997). "Predictions and measurements of scattering and absorption over broad wavelength ranges in tissue phantoms." Applied Optics **36**(4).

Niedre, M. J., R. H. de Kleine, E. Aikawa, D. G. Kirsch, R. Weissleder and V. Ntziachristos (2008). "Early photon tomography allows fluorescence detection of lung carcinomas and disease progression in mice in vivo." Proceedings of the National Academy of Sciences of the United States of America **105**(49): 19126-19131.

Ntziachristos, V., E. A. Schellenberger, J. Ripoll, D. Yessayan, E. Graves, A. Bogdanov, L. Josephson and R. Weissleder (2004). "Visualization of antitumor treatment by means of fluorescence molecular tomography with an annexin V-Cy5.5 conjugate." Proceedings of the National Academy of Sciences of the United States of America **101**(33): 12294-12299.

Ntziachristos, V., A. Soubret, G. Turner and Ieee (2005). Early photon optical tomography. 2005 Ieee International Conference on Acoustics, Speech, and Signal Processing, Vols 1-5 - Speech Processing. New York, Ieee**:** 837-840.

Ntziachristos, V., C. H. Tung, C. Bremer and R. Weissleder (2002a). "Fluoresence molecular tomography resolves protease activity in vivo." Nature Medicine **8**.

Ntziachristos, V., A. Yodh, M. Schnall and B. Chance (2002b). "MRI-Guided Diffuse Optical Spectroscopy of Malignant and Benign Breast Lesions." Neoplasia **4**(4): 347-354.

Ntziachristos, V., A. G. Yodh, M. Schnall and B. Chance (2000). "Concurrent MRI and diffuse optical tomography of breast after indocyanine green enhancement." Proceedings of the National Academy of Sciences in the United States of America **97**(6): 2767-2772.

Nziachristos, V. (2006). "Fluorescence Molecular Imaging." Annual Review of Biomedical Engineering **8**: 1-33.

Okada, E. and D. T. Delpy (2003a). Effect of discrete scatterers in CSF layer on optical path length in the brain. Photon Migration, Diffuse Spectroscopy and Optical Coherence Tomography: Imaging and Functional Assesment, Amsterdam, Netherlands.

Okada, E. and D. T. Delpy (2003b). "Near-infrared light propagation in an adult head model. I. Modeling of low-level scattering in the cerebrospinal fluid layer." Appl. Opt. **42**(16).

Okawa, S. and Y. Yamada (2010). "Reconstruction of fluorescence/bioluminescence sources in biological medium with spatial filter." Opt. Express **18**(12).

Orlova, A., I. Turchin, V. Plehanov, N. Shakhova, I. Fiks, M. Kleshnin, N. Konuchenko and V. Kamensky (2008). "Frequency-domain diffuse optical tomography with single source-detector pair for breast cancer detection." Laser Physics Letters **5**(4): 321-327.

Perkins, A. C. and G. Vivian (2009). "Molybdenum supplies and nuclear medicine services." Nucl Med Commun **30**(9).

Peters, V. G. and et al. (1990). "Optical properties of normal and diseased human breast tissues in the visible and near infrared." Physics in Medicine and Biology **35**(9): 1317.

Pickering, J. W., S. A. Prahl, N. van Wieringen, J. F. Beek, H. J. C. M. Sterenborg and M. J. C. van Gemert (1993). "Double-integrating-sphere system for measuring the optical properties of tissue." Appl. Opt. **32**(4): 399-410.

Pogue, B. W., S. C. Davis, X. M. Song, B. Brooksby, H. Dehghani and K. Paulsen (2006). "Image analysis methods for diffuse optical tomography." Journal of Biomedical Optics **11**(3).

Pogue, B. W. and et al. (1995). "Initial assessment of a simple system for frequency domain diffuse optical tomography." Physics in Medicine and Biology **40**(10): 1709.

Pogue, B. W. and M. S. Patterson (1994). "Frequency-domain optical absorption spectroscopy of finite tissue volumes using diffusion theory." Phys. Med. Biol **39**.

Pogue, B. W., M. Testorf, T. McBride, U. Osterberg and K. Paulsen (1997). "Instrumentation and design of a frequency-domaindiffuse optical tomography imager for breast cancer detection." Opt. Express **1**(13): 391-403.

Prahl, S. A. (2001). "http://omlc.ogi.edu/spectra/." 2010.

Prahl, S. A. (2007). "http://omlc.ogi.edu/software/mc/."

Prahl, S. A., M. J. C. van Gemert and A. J. Welch (1993). "Determining the optical properties of turbid mediaby using the adding?doubling method." Appl. Opt. **32**(4): 559-568.

Premaratne, M., E. Premaratne and A. Lowery (2005). "The photon transport equation for turbid biological media with spatially varying isotropic refractive index." Opt. Express **13**(2): 389-399.

Proud, D. L., R. W. Silverman and A. Chatziioannou (2005). "Readout of the Optical PET (OPET) detector." IEEE Trans Nucl Sci. **52**(1).

Radhakrishnan, H., A. Senapati, D. Kashyap, Y. B. Peng and H. Liu (2005). "Light scattering from rat nervous system measured intraoperatively by near-infrared reflectance spectroscopy." Journal of Biomedical Optics **10**(5): 051405-051408.

Ray, P., A. M. Wu and S. S. Gambhir (2003). "Optical bioluminescence and positron emission tomography imaging of a novel fusion reporter gene in tumor xenografts of living mice." Cancer Research **63**(6).

Ren, N., J. Liang, X. Qu, J. Li, B. Lu and J. Tian (2010). "GPU-based Monte Carlo simulation for light propagation in complex heterogeneous tissues." Opt. Express **18**(7): 6811-6823.

Rice, A. and C. M. Quinn (2002). "Angiogenesis, thrombosis and ductal carcinoma in situ of the breast." Journal of Clinical Pathology **55**: 569-574.

Riley, J., H. Dehghani, M. Schweiger, S. R. Arridge, J. Ripoll and M. Nieto-Vesperinas (2000). "3D optical tomography in the presence of void regions." Opt. Express **7**(13): 462-467.

Ritman, E. L. (2002). "Molecular imaging in small animals†-†roles for micro-CT." Journal of Cellular Biochemistry **87**(S39): 116-124.

Sasov, A. (2002). Desktop x-ray micro-CT instruments. Developments in X-Ray Tomography III, San Diego, CA, USA, SPIE.

Schulz, R. B., J. Ripoll and V. Ntziachristos (2004). "Experimental fluorescence tomography of tissues with noncontact measurements." IEEE Transactions on Medical Imaging **23**(4).

Schulz, R. B., J. Ripoll and V. Nziachristos (2003). "Noncontact optical tomography of turbid media." Optics Letters **28**(18).

Schweiger, M. (2008). "http://web4.cs.ucl.ac.uk/research/vis/toast/index.html."

Schweiger, M. and S. Arridge (1999). "Optical tomographic reconstruction in a complex head model using a priori region boundary information." Physics in Medicine & Biology **44**(11): 2703-2721.

Schweiger, M. and S. R. Arridge (2003). "Image reconstruction in optical tomography using local basis functions." Journal of Electronic Imaging **12**(4): 583-593.

Schweiger, M., S. R. Arridge, M. Hiraoka and D. T. Delpy (1995). "The Finite-Element Method for the Propagation of Light in Scattering Media - Boundary and Source Conditions." Medical Physics **22**(11): 1779-1792.

Senapati, A. K., H. Radhakrishnan, H. Liu and Y. B. Pen (2005). "Detection of degeneration in rat sciatic nerbe by in vivo near infrared spectroscopy." Brain Research Protocols **14**(2).

Sharma, S. K. and S. Banerjee (2003). "Role of approximate phase functions in Monte Carlo simulation of light propagation in tissues." Journal of Optics A: Pure and Applied Optics **5**(3): 294-302.

Sharp, P. F., H. G. Gemmell and A. D. Murray (2005). Practical Nuclear Medicine. London, Springer Verlag.

Siegel, A. M., J. P. Culver, J. B. Mandeville and D. Boas (2003). "Temporal comparison of functional brain imaging with diffuse optical tomography and fMRI during rat forepaw stimulation." Phys. Med. Biol **48**: 1391-1403.

Simpson, C. R. and et al. (1998). "Near-infrared optical properties of ex vivo human skin and subcutaneous tissues measured using the Monte Carlo inversion technique." Physics in Medicine and Biology **43**(9): 2465.

Srinivasan, S., B. W. Pogue, C. Carpenter, P. K. Yalavarthy and K. Paulsen (2007). "A boundary element approach for image-guided near-infrared absorption and scatter estimation." Medical Physics **34**(11): 4545-4557.

Srinivasan, S., B. W. Pogue, H. Jiang, H. Dehghani and K. Paulsen (2003). Validation of hemoglobin and water molar absorption spectra in near-infrared diffuse optical tomography. Optical Tomography and Spectroscopy of Tissue V. B. Chance, R. R. Alfano, B. Tromberg, M. Tamura and E. M. Sevick-Muraca. Bellingham, WA, SPIE-INT Soc Optical Engineering.

Srinivasan, S., B. W. Pogue, S. Jiang, H. Dehghani, C. Kogel, S. Soho, J. J. Gison, T. D. Tosteson, S. P. Poplack and K. Paulsen (2006). "In Vivo Hemoglobin and Water Concentrations, Oxygen Saturation, and Scattering Estimates From Near-Infrared Breast Tomography Using Spectral Reconstruction." Acad Radiol **13**.

Srinivasan, S., B. W. Pogue, S. Jiang, H. Dehghani and K. Paulsen (2005). "Spectrally constrained chromophore and scattering near-infrared tomography provides quantitative and robust reconstruction." Appl. Opt. **44**(10).

Tarvainen, T., M. Vauhkonen, V. Kolehmainen and J. P. Kaipio (2005). "Hybrid radiative-transfer-diffusion model for optical tomography." Applied Optics **44**(6): 876-886.

Thayer, D., M. B. Unlu, Y. Lin, K. Yan, O. Nalcioglu and G. Gulsen (2010). "Dual-contrast dynamic MRI-DOT for small animal imaging." Technol Cancer Res Treat **9**(1).

Tomasevic, D. I. and E. W. Larsen (1996). "The simplified P_2 approximation." Nuclear Science and Engineering **122**(3).

Tromberg, B., O. Coquoz, J. Fishkin, T. Pham, E. R. Anderson, J. Butler, M. Cahn, J. D. Gross, V. Venugoplan and D. Pham (1997). "Non-invasive measurements of breast tissue optical properties using frequency-domain photon migration." Philosophical Transactions of the Royal Society **352**.

Tsukamoto, E. and S. Ochi (2006). "PET/CT today: System and its impact on cancer diagnosis." Annals of Nuclear Medicine **20**(4): 255-267.

van Staveren, H. J., C. J. M. Moes, J. van Marle, S. A. Prahl and M. J. C. van Gemert (1991). "Light Scattering in Intralipid-10% in the wavelength range of 400-1100 nm." Applied Optics **30**(21).

Vishwanath, K., B. W. Pogue and M. Myeck (2002). "Quantitative fluorescence lifetime spectroscopy in turbid media: comparison of theoretical, experimental and computational methods." Physics in Medicine and Biology **47**(18): 3387.

Wang, G., W. Cong, K. Durairaj, X. Qian, H. Shen, P. Sinn, E. Hoffman, G. McLennan and M. Henry (2006a). "*In vivo* mouse studies with bioluminescence tomography." Opt. Express **14**(17).

Wang, L. and S. L. Jacques (1993). "Hybrid model of Monte Carlo simulation and diffusion theory for light reflectance by turbid media." J. Opt. Soc. Am. **10**(8).

Wang, L., S. L. Jacques and L. Zheng (1995). "MCML - Monte Carlo modeling of photon transport in multi-layered tissues." Computer Methods and Programs in Biomedicine **47**: 131-146.

Wang, X., B. W. Pogue, S. D. Jiang, H. Dehghani, X. M. Song, S. Srinivasan, B. Brooksby, K. Paulsen, C. Kogel, S. P. Poplack and W. A. Wells (2006b). "Image Reconstruction of effective Mie scattering parameters of breast tissue in vivo with near-infrared tomography." Journal of Biomedical Optics **11**(4).

Warren, R. (2001). "Screening women at high risk of breast cancer." European Journal of Radiology **39**(1): 50-59.

Webb, P. M., M. C. Cummings, C. J. Bain and C. M. Furnival (2004). "Changes in survival after breast cancer: improvements in diagnosis or treatment?" The Breast **13**: 7-14.

Westbrook, C. and C. Roth (2005). MRI in Practice. Oxford, Blackwell Publishing Ltd.

Wolbarst, A. B. (1999). Looking Within: How X-ray, CT, MRI, Ultrasound and Other Medical Images are Created and How they Help Physicians Save Lives. Berkeley, CA, University of California Press.

Wolf, R., U. Lindauer, U. Reuter, T. Back, A. Villringer, K. Einhaupl and U. Dirnagl (1997). "Noninvasive Near Infrared Spectroscopy Monitoring of Regional Cerebral Blood Oxygenation Changes During Peri-Infarct Depolarizations in Focal Cerebral Ischemia in the Rat." Journal of Cerebral Blood Flow & Metabolism **17**: 950-954.

Wright, S., M. Schweiger and S. R. Arridge (2006). "Reconstruction in optical tomography using the P_N approximations." Measurement Science and Technology **18(**2007): 79-86.

Yalavarthy, P. K., H. Dehghani, B. W. Pogue and K. Paulsen (2006). "Critical computational aspects of near infrared circular tomographic imaging: Analysis of measurement number, mesh resolution and reconstruction basis." Opt. Express **14**: 6113-6127.

Yalavarthy, P. K., B. W. Pogue, H. Dehghani and K. Paulsen (2007). "Weight-matrix structured regularization provides optimal generalized least-squares estimate in diffuse optical tomography." Medical Physics **34**(6).

Yoon, G., A. J. Welch, M. Motamedi and M. C. J. Van Gemert (1987). "Development and application of three-dimensional light distribution model for laser irradiated tissue." Journal Name: IEEE J. Quant. Electron.; (United States); Journal Volume: QE-23:10: Medium: X; Size: Pages: 1721-1733.

Yu, G., T. Durduran, T. M. Busch, H. W. Wang, C. Zhou, H. M. Saunders, C. M. Sehgal and A. G. Yodh (2003). Non-invasive monitoring hemodynamic responses in RIF tumors during and after PDT. SPIE Photonics West, San Jose, Ca.

Zacharopoulos, A. D., S. R. Arridge, O. Dorn, V. Kolehmainen and J. Sikora (2006). "Three-dimensional reconstruction of shape and piecewise constant region values for optical tomography using spherical harmonic parametrization and a boundary element method." Inverse Problems **22**(5): 1509-1532.

Zeff, B. W., B. R. White, H. Dehghani, B. L. Schlaggar and J. Culver (2007a). "Retinotopic mapping of adult human visual cortex with high-density diffuse optical tomography." Proceedings of the National Academy of Sciences in the United States of America **104**(29): 12169-12174.

Zeff, B. W., B. R. White, H. Dehghani, B. L. Schlaggar and J. P. Culver (2007b). "Retinotopic mapping of adult human visual cortex with high-density diffuse optical tomography." PNAS **104**: 12169-12174.

Zinn, K. R., T. R. Chaudhuri, A. A. Szafran, D. O'Quinn, C. Weaver, K. Dugger, D. Lamar, R. A. Kesterson, X. Wang and S. Frank (2008). "Noninvasive Bioluminescence Imaging in Small Animals." ILAR Journal **49**(1).