

Fast Radiative-Transfer-Equation-Based Image Reconstruction Algorithms  
for Non-Contact Diffuse Optical Tomography Systems

Jingfei Jia

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy  
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2015





## ABSTRACT

It is well known that the radiative transfer equation (RTE) is the most accurate deterministic light propagation model employed in diffuse optical tomography (DOT). RTE-based algorithms provide more accurate tomographic results than codes that rely on the diffusion equation (DE), which is an approximation to the RTE, in scattering dominant media. However, RTE based DOT (RTE-DOT) has limited utility in practice due to its high computational cost and lack of support for general non-contact imaging systems. In this dissertation, I developed fast reconstruction algorithms for RTE-based DOT (RTE-DOT), which consists of three independent components: an efficient linear solver for forward problems, an improved optimization solver for inverse problem, and the first light propagation model in free space that fully considers the angular dependency, which can provide a suitable measurement operator for RTE-DOT. This algorithm is validated and evaluated with numerical experiments and clinical data. According to these studies, the novel reconstruction algorithm is up to 30 times faster than traditional reconstruction techniques, while achieving comparable reconstruction accuracy.

# Table of Contents

<b>List of Figures .....</b>	<b>v</b>
<b>List of Tables.....</b>	<b>ix</b>
<b>Chapter 1. Introduction .....</b>	<b>1</b>
<b>Chapter 2. Diffuse Optical Tomography .....</b>	<b>5</b>
<b>2.1. Light absorption and scattering in tissue .....</b>	<b>5</b>
2.1.1. Absorption.....	6
2.1.2. Scattering.....	7
<b>2.2. Models of light propagation in tissue .....</b>	<b>8</b>
2.2.1. Radiative transfer equation (RTE).....	9
2.2.2. Diffusion approximation to RTE.....	10
<b>2.3. Diffuse optical tomography as an optimization problem.....</b>	<b>11</b>
<b>2.4. Discussion.....</b>	<b>14</b>
<b>Chapter 3. Efficient Forward Solver in RTE-DOT .....</b>	<b>16</b>
<b>3.1. Introduction .....</b>	<b>16</b>
<b>3.2. Discretization and preconditioning of RTE .....</b>	<b>18</b>
3.2.1. Upwind finite volume discrete-ordinates discretization .....	18
3.2.2. Preconditioning with incomplete LU factorization on the reduced linear operator .....	22
3.2.3. Node-centered finite volume mesh generation.....	23
<b>3.3. Preconditioned block linear solver with multiple sources.....</b>	<b>29</b>
3.3.1. Krylov subspace methods and BiCGStab algorithm .....	29
3.3.2. Block Krylov subspace methods and block BiCGStab algorithm .....	30
3.3.3. Multi-threading acceleration for block BiCGStab algorithm .....	32

<b>3.4. Performance evaluation through numerical experiments .....</b>	<b>36</b>
3.4.1. Numerical phantoms .....	36
3.4.2. Performance evaluation of the novel preconditioner .....	37
3.4.3. Influence of the anisotropic factor on the performance.....	39
3.4.4. Influence of optical properties on the performance .....	40
3.4.5. Influence of spatial and angular discretization on the performance .....	45
3.4.6. Influence of the number of right-hand sides on the performance.....	47
3.4.7. Test on multi-threading block BiCGStab methods.....	48
<b>3.5. Discussion .....</b>	<b>51</b>
<b>Chapter 4. Efficient Inverse Solvers in RTE-DOT .....</b>	<b>54</b>
<b>4.1. Introduction .....</b>	<b>54</b>
<b>4.2. A line-search-free BFGS inverse solver for unconstrained optimization formulation.</b>	<b>57</b>
4.2.1. Newton's method and quasi-Newton methods.....	57
4.2.2. A line-search-free BFGS inverse solver.....	60
<b>4.3. A full-space algorithm for the constrained optimization formulation .....</b>	<b>65</b>
4.3.1. Full-space Newton's method for constrained optimization .....	65
4.3.2. Preconditioned Krylov solver for the KKT system .....	67
4.3.3. Updating with the descent direction.....	69
4.3.4. Full-space algorithm for PDE-constrained optimization in RTE-DOT .....	70
<b>4.4. Validation and performance evaluation through numerical simulations .....</b>	<b>71</b>
4.4.1. Numerical phantoms and experiment settings .....	71
4.4.2. Performance evaluation with numerical tissue models.....	74
4.4.3. Reconstruction of absorption coefficient.....	74
4.4.4. Reconstruction of reduced scattering coefficient .....	80
4.4.5. Reconstruction of both absorption and reduced scattering coefficients .....	85

4.4.6. Influence of noise .....	92
4.4.7. Influence of the initial guess.....	94
<b>4.5. Discussion.....</b>	<b>96</b>
<b>Chapter 5. Measurement Operators for general Non-contact DOT Systems.....</b>	<b>99</b>
<b>5.1. Introduction .....</b>	<b>99</b>
<b>5.2. Theory and methods .....</b>	<b>101</b>
5.2.1. Surface radiation coordinate system .....	101
5.2.2. CCD camera acceptance coordinate system .....	104
5.2.3. Light propagation in free space and coordinate transformation .....	105
5.2.4. Numerical algorithm for measurement operator .....	107
<b>5.3. Validation of the back ray-tracing model.....</b>	<b>109</b>
5.3.1. Validation through analytical solution.....	109
5.3.2. Validation through fluorescence molecular tomography result.....	112
<b>5.4. Discussion .....</b>	<b>114</b>
<b>Chapter 6. Algorithm evaluation with clinical data.....</b>	<b>117</b>
<b>6.1. Clinical data.....</b>	<b>117</b>
<b>6.2. Reconstruction, classification and performance comparison .....</b>	<b>119</b>
6.2.1. Reconstruction results .....	119
6.2.2. Classification on the reconstruction results .....	120
6.2.3. Performance comparison .....	123
<b>6.3. Conclusion .....</b>	<b>125</b>
<b>Chapter 7. Conclusion.....</b>	<b>126</b>
<b>Reference .....</b>	<b>129</b>
<b>Appendix A: Implementation details in inverse solvers.....</b>	<b>138</b>

<b>A.1. Implementation details in the line-search-free BFGS algorithm .....</b>	<b>141</b>
A.1.1. Gradient of the objective function .....	141
A.1.2. Projected descent direction on the forward variable .....	141
A.1.2. Solution for the optimization problem for step size determination.....	142
<b>A.2. Implementation details of the Newton step in the full-space algorithm.....</b>	<b>142</b>

# List of Figures

Figure 2.1: Flowchart for iterative reconstruction process in DOT.....	12
Figure 3.1: Illustration of 2D unstructured grids and node-centered FVM mesh. (a) 2D unstructured grids; (b) 2D node-centered FVM mesh (grids are surrounded with red dash boundaries).....	26
Figure 3.2: Source locations for Phantom 1 and 2; (a) 2D phantom: 20 sources are regularly distributed along the perimeter of the disk; (b) 3D phantom: 32 sources are distributed on the lateral surface of the cylinder with 4 height levels 0.375cm, 1.125cm, 1.875cm and 2.625cm. On each level, there are 8 sources distributed with even distance.....	37
Figure 3.3: Results for 2D disk phantom with homogeneous optical properties; (a) CPU time (BiCGStab); (b) CPU time (block BiCGStab) (c) Speedup factor of block BiCGStab; (d) MV number (BiCGStab); (e) MV number (block BiCGStab); (f) MV ratio (BiCGStab MV number / block BiCGStab MV number) .....	42
Figure 3.4: Results for 3D cylinder phantom with homogeneous optical properties; (a) CPU time (BiCGStab); (b) CPU time (block BiCGStab) (c) Speedup factor of block BiCGStab; (d) MV number (BiCGStab); (e) MV number (block BiCGStab); (f) MV ratio (BiCGStab MV number / block BiCGStab MV number). .....	43
Figure 3.5: Locations of two spherical objects in Phantom 2.....	44
Figure 3.6: CPU times, speedup factors and MV ratios with respect to spatial and directional discretization; (a) CPU times comparison between BiCGStab and block BiCGStab on different spatial meshes with the fixed solid angle set $S_8$ ; (b) speedup factor and MV number ratio on different spatial meshes with the fixed solid angle set $S_8$ ; (c) CPU times comparison between BiCGStab and block BiCGStab on different orders of discrete ordinates ( $S_4 \sim S_{12}$ ) with the fixed spatial mesh; (d) speedup factor and MV number ratio on different orders of discrete ordinates ( $S_4 \sim S_{12}$ ).....	46
Figure 3.7: Performance of the block BiCGStab algorithm with different number of sources: (a) Overall CPU time required in the sequential and block BiCGStab algorithms with various source numbers. (b) The speedup factor and MV ratio of the block BiCGStab algorithm compared with the traditional BiCGStab algorithm with various source numbers. ....	47
Figure 3.8: CPU time, speedup factor, MV number and MV ratio comparison respect to thread number in normal tissue case ( $\mu_a = 0.1\text{cm}^{-1}$ , $\mu'_s = 10.0\text{cm}^{-1}$ ): (a) CPU time comparison; (b) Speedup factor comparison; (c) Average MV number comparison; (d) MV ratio comparison. ..	50
Figure 3.9: CPU time, speedup factor, MV number and MV ratio comparison respect to thread number in high absorbing and low scattering medium case ( $\mu_a = 0.5\text{cm}^{-1}$ , $\mu'_s = 1.0\text{cm}^{-1}$ ): (a)	

CPU time comparison; (b) Speedup factor comparison; (c) Average MV number comparison; (d) MV ratio comparison. .... 51

Figure 4.1: Source and detector locations for Phantom 1 and 2; (a) 2D phantom: 10 sources and 40 detectors are uniformly distributed along the perimeter of the disk; (b) 3D phantom: 16 sources and 64 detectors are distributed on the lateral surface of the cylinder with 4 height levels 0.375cm, 1.125cm, 1.875cm and 2.625cm. On each level, there are 4 sources and 16 detectors distributed with equal distance..... 72

Figure 4.2: Absorption coefficient distribution in the two-dimensional phantom. (a) Exact distribution; (b) Traditional BFGS; (c) Line-search-free BFGS; (d) Full-space algorithm..... 75

Figure 4.3: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a, b) Relative function value over iteration number and CPU time; (c, d) Required iteration number and CPU time to converge the objective function value to certain levels..... 77

Figure 4.4: Absorption coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm..... 78

Figure 4.5: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels..... 79

Figure 4.6: Reduced scattering coefficient distribution in the two-dimensional phantom. (a) Exact distribution; (b) Traditional BFGS; (c) Line-search-free BFGS; (d) Full-space algorithm. .... 81

Figure 4.7: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels..... 82

Figure 4.8: Reduced scattering coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm. .... 83

Figure 4.9: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels..... 84

Figure 4.10: (Top)  $\mu_a$  and (bottom)  $\mu_s'$  distribution in the two dimensional phantom. (a1,2) Exact distribution; (b1,2) Traditional BFGS; (c1,2) Line-search-free BFGS; (d1,2) Full-space algorithm ..... 86

Figure 4.11: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels..... 88

Figure 4.12: Absorption coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm..... 89

Figure 4.13: Absorption coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm..... 90

Figure 4.14: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels..... 91

Figure 4.15: Absorption coefficient distribution in the two-dimensional phantom reconstructed with SNR level  $+\infty$  (Row 1), 20dB (Row 2), 15dB (Row 3) by different algorithms. (a1,2,3) Traditional BFGS; (b) Line-search-free BFGS; (c) Full-space algorithm..... 93

Figure 4.16: Absorption coefficient distribution in the two-dimensional phantom reconstructed with initial guess equal to (Row 1), 20% higher than (Row 2) and 20% lower than the background value by different algorithms. (a1,2,3) Traditional BFGS; (b) Line-search-free BFGS; (c) Full-space algorithm..... 95

Figure 5.1: An illustration of (a) surface discretization with triangle elements and (b) a triangle element  $\Gamma$  and a location vector  $\mathbf{r}$ ..... 103

Figure 5.2: An illustration of a simplified model of the CCD camera ..... 105

Figure 5.3: Experiment settings for the direct illumination case ..... 110

Figure 5.4: Comparison between the exact and the computed measurement on the image plane. (a) The exact measurement; (b) The measurements computed with the back ray-tracing model; (c) The relative error of the computed measurement..... 111

Figure 5.5: A picture of the double conical mirror system ..... 112

Figure 5.6: Double conical mirror scheme: the first mirror and the second mirror were designed to cover a 40mm diameter 80mm length cylinder target. .... 113



Figure 5.7: A picture of the coke can with the double conical mirror system. (a) Positioning of the coke can in the imaging unit; (b) the captured image on the camera.....	113
Figure 5.8: Reconstruction results of the tumor growth one week after subcutaneous tumor injection.....	114
Figure 6.1: Transillumination captured by a CCD camera during a scan of PIP joint II.....	118
Figure 6.2: Discretization of a sample PIP joint and locations of sources and detectors. (a) Tetrahedral mesh of a finger that contains 7424 nodes and 32207 tetrahedrons. (b) Locations of 11 sources on the dorsal surface of the joint (c) Locations of 145 detectors on the palmar surface of the joint. ....	119
Figure 6.3: Absorption (a and b) and reduced scattering (c and d) coronal cross sections of PIP joints from subjects with RA (a and c) and without RA (b and d). ....	120
Figure 6.4: Decision boundary of LDA separating affected from healthy data using $\text{std}(\mu_a)$ and $\text{max}\mu_s'$ as features. ....	123
Figure 6.5: Reconstruction results of $\mu_a$ (a and b) and $\mu_s'$ (c and d) from the traditional BFGS algorithm (a and c) and the fast reconstruction algorithm (b and d).....	124
Figure 6.6: Convergence path of relative objective value with respect to the CPU time .....	124

## List of Tables

Table 3.1: Data structures used for FVM mesh's construction. ....	24
Table 3.2: The phantom settings of the 2D disk and 3D cylinder .....	37
Table 3.3: Optical properties for Phantom 1 in performance evaluation.....	38
Table 3.4: Performance comparison between $P_0$ , $P_1$ and $P_2$ preconditioned BiCGStab algorithm.....	38
Table 3.5: Performance comparison between traditional and block BiCGStab algorithms on various combinations of $(g, \mu'_s)$ .....	40
Table 3.6: Results of tests on Phantom 2 with inhomogeneous optical properties.....	45
Table 3.7: Spatial meshes employed to study effects of spatial discretization.....	45
Table 3.8: Discrete ordinates employed to study the effects of angular discretization .....	45
Table 4.1: The phantom settings of the 2D disk and 3D cylinder .....	72
Table 4.2: Test cases for absorption coefficients reconstruction.....	73
Table 4.3: Correlation factors and deviation factors of $\mu_a$ reconstruction results from the traditional and line-search-free BFGS algorithms on Case 3. ....	75
Table 4.4: Correlation factors and deviation factors of $\mu_a$ reconstruction results from the traditional and line-search-free BFGS algorithms on Case 5. ....	77
Table 4.5: Correlation factors and deviation factors of $\mu_a$ reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 1~5. ....	80
Table 4.6: Correlation factors and deviation factors of $\mu'_s$ reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 8. ....	81
Table 4.7: Correlation factors and deviation factors of $\mu'_s$ reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 10. ....	84
Table 4.8: Correlation factors and deviation factors of $\mu'_s$ reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 6~10. ....	85
Table 4.9: Correlation factors and deviation factors of $\mu_a$ and $\mu'_s$ reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 11. ....	87

Table 4.10: Correlation factors and deviation factors of $\mu_a$ and $\mu_s'$ reconstruction results from the traditional and line-search-free BFGS algorithms on Case 12. ....	90
Table 4.11: The reconstruction quality and the CPU time with the traditional and the line-search-free BFGS algorithms for different noise levels. ....	92
Table 4.12: The reconstruction quality and the CPU time with the traditional and the line-search-free BFGS algorithms for different initial guess.....	95
Table 6.1: Diagnostic table based on clinical evaluation and radiological imaging (ultrasound and MRI). ....	118
Table 6.2: Mean values, standard deviations and p-values from the t-test for individual features .....	121
Table 6.3: Summary of classification results of feature combinations with 10 highest Youden indices .....	123

## **Acknowledgements**

First of all, I would like to thank my thesis committee members: Dr. Andrew F. Laine, Andreas H. Hielscher, X. Edward Guo, Henry Hess and Hyun K. Kim. This dissertation would not have been possible without their guidance and help.

I would like to especially express my deepest gratitude to my advisor, Dr. Andreas H. Hielscher, for his excellent guidance, caring, patience, and continuing support throughout the research.

I would like also thank my co-advisor, Dr. Hyun K. Kim, for his numerous invaluable advises and inspiring discussions in the past five years.

I would like to thank all my previous and current lab mates: Molly Flexman, Ludguier Montejo, Michael Khalil, Jonghwan Lee, Chris Fong, Jacqueline Gunther, Jennifer Hoi, Alessandro Marone, for their help and friendship throughout my study.

I would like to thank all my friends from the Industrial Engineering and Operations Research department at Columbia University. My vision had been significantly broadened from discussion with them.

Finally, I would like to thank my parents and my wife, Xinxin Cheng. Their appreciation and expectation help me realize my potential and motivate me to keep moving forward.

## **Dedication**

*To my parents and my wife  
for their endless love and support*

# Chapter 1. Introduction

The overall goal of my dissertation is to develop fast image reconstruction algorithms that can be used for radiative-transfer-equation-based diffuse optical tomography (RTE-DOT). It is widely acknowledged that the radiative transfer equations (RTEs) provide the most accurate deterministic light propagation model for biological tissues [1-6]. In the most general case RTEs simply state that as a beam of radiation travels, it loses energy to absorption, redistributes energy by scattering out of the direction of travel, and gains energy by scattering into the direction of travel or energy emission (e.g. fluorophores). RTEs have application in a wide variety of subjects including optics, astrophysics, atmospheric science, and remote sensing. Analytic solutions to the radiative transfer equation (RTE) exist for simple cases but for more realistic media, with complex multiple scattering effects, numerical methods are required.

Approximations to RTE, such as the diffusion equation (DE), are commonly employed when scattering effects dominate absorption effects. This is typically the case in many biological tissues. However, these approximations perform poorly if they are used in transport regime [1]. This regime includes cases such as small tissue volume, highly absorbing media, and tissue with void-like, fluid-filled regions. Clinically these cases occur in imaging of finger joints (small tissue volume), hematoma and hemorrhages (high absorbing media), and the brain, which is embedded in low-scattering cerebrospinal fluid. While RTE-DOT can be applied to these cases, the widespread use of the RTE in DOT is still limited because these codes are relatively inefficient and require long computation times. For example a reconstruction on a medium size finger joint can take 1~3 hours with the current traditional algorithms [7, 8], while the data acquisition only takes a few minutes. The inefficiency of these algorithms is mainly caused by the following three reasons:

- (1) The high dimensionality of the RTE, which in addition to the spatial location contains a directional variable, leads to a large-scale linear equation  $Ax = b$  in the discretization. Moreover, the integral term in the RTE increases the number of non-zero entries in  $A$  significantly and those non-zero entries are mainly distributed off-diagonally. That lowers the convergence rate of the linear equation solver and increases the complexity of the preconditioning process.
- (2) The optimization problem for the image reconstruction requires repeated evaluations of RTE and the nature of the problem is highly ill-posed, non-convex and of large-scale. This makes RTE-DOT problems even slower and more difficult to solve.
- (3) Finally, large measurement data sets are required to counterbalance the noise and the ill-posedness arising from high dimensionality of the RTE. Yet the widely employed fiber-based imaging system cannot meet this requirement. Contact free imaging systems, which, for example, employ charged-coupled-device (CCD) camera as the light detector unit, can provide a large number of measurement points. However, the corresponding transport theory based measurement operator that describes the transition of the light from the object's surface to the camera is rather complex. All existing models for this operator either assume the light propagates in a diffuse manner [9, 10] or impose strong assumption on the optical system [11]. A mathematical formulation for complex imaging system with general optical components, which takes angular-dependent photons emission into account, is still missing.

This dissertation aims to overcome the aforementioned difficulties by developing an reconstruction technique for RTE-DOT that consists of an (a) efficient line solver for the forward

problems, (b) a novel optimization algorithm for the inverse problem and (c) a light propagation model in free space that full considers the angular dependency of photons. To reduce the cost, that may potentially become a further hindrance to the broad use of RTE-DOT, all the algorithms in this dissertation are designed to run on a PC, instead of clusters. To achieve this, the following four specific aims are proposed.

- (1) **Specific Aim 1: Develop an efficient linear solver for the forward problem in RTE-DOT.** This work consists of three steps. First, an incomplete LU factorization (ILU) based preconditioner will be designed for the linear operator in the discretized RTE is preconditioned. Then a novel linear solver, called block biconjugate gradient stabilized method (block BiCGStab), will be developed to solve the forward problems in RTE-DOT in an all-at-once manner. Finally, multi-threading techniques will be implemented to further accelerate this block BiCGStab algorithm. With these steps I expect to obtain a speedup factor of at least 2, as compare to the traditional method.
- (2) **Specific Aim 2: Investigate efficient optimization solvers for inverse problem in RTE-DOT.** Two kinds of inverse solvers for RTE-DOT will be explored. First, I will improve the traditional BFGS algorithm, which is the state of the art algorithm in unconstrained optimization, by replacing the line search step with a step size estimation process. With this novel technique extra function evaluations are avoided and more precise step sizes are obtained. Second, I will implement the first full-space algorithm for RTE-DOT. This approach makes use of the full Hessian's information to achieve higher convergence rate.
- (3) **Specific Aim 3: Implement a transport-theory-based light propagation operator for free space imaging.** I will model the light propagation in free space with back ray



tracing method, which will allow inclusion of angular dependent effects. This model will be able to provide measurement operator for RTE-DOT with complex non-contact imaging system with general optical components.

- (4) **Specific Aim 4: Integrate the above three components into an image reconstruction algorithm that will be validate with existing clinical data from human finger joint imaging.**

If all the specific aims proposed in this dissertation are implemented successfully, a speedup factor over 10 as compared to the traditional reconstruction technique in RTE-DOT can be expected. Furthermore, more advanced optical designs can be employed and foreseeably make promising contribution to the RTE-DOT imaging system. All together, these improvements should lead to a greater clinical utility of RTE-DOT.

The remainder of this dissertation is organized as follow. A succinct introduction to DOT is given in Chapter 2. Efficient numerical techniques for forward problems and inverse problems related to RTE-DOT are developed in in Chapter 3 and Chapter 4, respectively. The transport-theory-based light propagation model for free space is developed in Chapter 5. The validation and performance evaluation of the overall algorithm with clinical data is presented in Chapter 6. Finally, a conclusion and discussion on future directions is provided in Chapter 7.

## Chapter 2. Diffuse Optical Tomography

Diffuse optical tomography (DOT) is a biomedical imaging modality that may be best understood by comparing it to the more familiar modality of X-ray computed tomography (CT). DOT is similar to CT in a couple of aspects. First, both imaging modalities direct the electromagnetic radiation and receive measurements at multiple locations of an imaging object. Second, they both require some numerical inverse algorithms to generate a cross sectional image. However, unlike CT, which uses high-energy radiation in the wavelength range of  $0.1\text{nm} < \lambda < 10\text{nm}$ , DOT uses low-energy near-infrared (NIR) light with wavelengths  $\lambda$  between 650nm and 900nm. Hence, patients are not exposed to the risk accompanied by high-energy ionizing radiation. In addition, DOT systems can be produced at a much lower cost compared to CT, are less complex, have a smaller footprint, and require less maintenance. Therefore, DOT has become a popular area of research over the last decade. The main areas of current applications include brain imaging [12-18], breast imaging [19-25], vascular imaging [26-31], small animal imaging [32-37] and imaging of finger joints [7, 38-41].

### 2.1. Light absorption and scattering in tissue

In DOT, near-infrared light, in the wavelength range of  $650\text{nm} < \lambda < 950\text{nm}$  is used to image tissue. This light can be absorbed and scattered when propagating inside the body. The energy being absorbed is determined by the tissue's absorption coefficient ( $\mu_a$  [ $\text{cm}^{-1}$ ]), while the scattered energy is determined by the scattering coefficient ( $\mu_s$  [ $\text{cm}^{-1}$ ]). The probability that light is scattered into any particular solid-angle is given by scattering phase function  $p(\theta, \phi)$ .

### 2.1.1. Absorption

Absorption is defined as the effect that a photon is annihilated and its energy is converted into the internal energy of molecules. Molecules that are responsible for the absorption of the visible and near-infrared light are called chromophores. In a simplified model for absorption, a chromophore has an absorption cross section with a certain radius which perfectly absorbs the beam of photons. The area of cross section of chromophores is denoted as  $a_\lambda$  [cm<sup>2</sup>], and typically, depends on the wavelength  $\lambda$  and properties of the specific chromophore. If the local volume density of chromophores is set to  $\rho_{\text{abs}}$  [cm<sup>-3</sup>], then the local absorption coefficient  $\mu_a$  for light with wavelength  $\lambda$  is given by

$$\mu_a = \rho_{\text{abs}} a_\lambda. \quad (2.1)$$

The absorption coefficient can also be expressed in another way

$$\mu_a = c \alpha_\lambda. \quad (2.2)$$

Here  $c$  represents the concentration of the absorber that is independent of the wavelength  $\lambda$  and  $\alpha_\lambda$  is the absorptivity of the absorber that depends on the  $\lambda$ .

We define  $z$  as the axis parallel to the light beam direction and  $\psi$  [W/cm<sup>2</sup>] as the light intensity of the beam. Then within an infinitesimal distance  $dz$ , the light intensity change  $d\psi$  within  $[z, z + dz]$  can be expressed as

$$d\psi(z) = -\mu_a(z)\psi(z)dz. \quad (2.3)$$

Here the absorption coefficient  $\mu_a$  and the light intensity  $\psi$  are assumed to be functions with respect to location  $z$ . This first order linear ODE (2.3) can be solved by multiplying the  $\exp(\int_0^z \mu_a(z')dz')$  as an integrating factor and then conducting an integration on its the both sides. Its solution, that is also referred to as the Beer-Lambert Law is given as below,

$$\psi(z) = \psi_0 \exp\left(-\int_0^z \mu_a(z')dz'\right). \quad (2.4)$$

In DOT,  $\mu_a$  is referred to as the overall absorption coefficient that is contributed by different species of absorbers. So equation (2.2) is generalized to read

$$\mu_{a,\lambda} = \sum_{k=1}^n c_k \alpha_{\lambda,k}. \quad (2.5)$$

Here the overall  $\mu_{a,\lambda}$  is expressed as a summation of products of concentration and the absorptivity for each individual species of absorber. Therefore, by reconstructing the overall absorption coefficient for different wavelength  $\lambda$ s in DOT, the concentrations of certain species  $\alpha_{\lambda,k}$  of absorbers can be derived by solving a linear equation. For biological tissue, the absorption coefficient usually ranges from  $0.03\text{cm}^{-1}$  to  $1.6\text{cm}^{-1}$ .

### 2.1.2. Scattering

In addition to being absorbed, light can be scattered when it is propagating in tissue. This scattering effect is mainly caused by random spatial variations in tissue density, refractive index and dielectric constant. The scattering coefficient  $\mu_s$  [ $\text{cm}^{-1}$ ] is used to describe the scattering property of a medium.

To simplify the model, we assume that light is scattered when it encounters a “scattering unit”. A scattering unit is defined as a sphere with specific radius. Its cross section area is given by  $s_\lambda$  [ $\text{cm}^2$ ] in this model. In a medium that contains many scattering units with a volume density  $\rho_{\text{scat}}$  [ $\text{cm}^{-3}$ ], the scattering coefficient  $\mu_s$  is given by

$$\mu_s = \rho_{\text{scat}} s_\lambda \quad (2.6)$$

When a scattering event occurs, light can be deflected into any directions. A phase function  $p(\theta, \phi)$  is introduced to describe the probability of a photon being deflected into a solid angle  $\{\theta (\theta \in [-\pi, \pi]), \phi (\phi \in [0, \pi])\}$  as compared to its original trajectory after being scattered. In general this phase function can be rather complex and analytical solutions of the Maxwell's equation exist for simple scatter centers only (e.g. spheres, cylinders, etc.). A simplified analytical expression for  $p$  was derived by Henyey and Greenstein [42] :

$$p(\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\theta)^{3/2}} \quad (2.7)$$

This so-called Henyey-Greenstein phase function only depends on  $\theta$  and is widely used in DOT. This equation contains the *anisotropy factor*  $g$ , which is the expectation of components on the original trajectory that is projected from the scattered direction of a photon, which is given by

$$g := \int_{-\pi}^{\pi} p(\theta) \cos \theta d\theta. \quad (2.9)$$

From this one can define the *reduced scattering coefficient*  $\mu'_s [\text{cm}^{-1}] := (1 - g)\mu_s$ . For biological tissue  $\mu'_s$  usually ranges between  $1.2\text{cm}^{-1}$  and  $40\text{cm}^{-1}$ , and the anisotropic factor  $g$  varies from 0.7 to 0.99.

## 2.2. Models of light propagation in tissue

In general, diffuse optical tomography (DOT) requires a light propagation model that describes how photons are moving inside tissue. Two popular forward models are widely used in DOT, which are the radiative transfer equation (RTE) and diffusion equation (DE).

### 2.2.1. Radiative transfer equation (RTE)

The radiative transfer equation (RTE) is widely acknowledged as the most accurate deterministic light propagation model [1-6]. The time-domain RTE (TD-RTE) models the radiance as a function of the location vector  $\mathbf{r}$  and the solid angle  $\mathbf{s}$  and the time  $t$  following the energy conservation law, where the radiance is defined as the radiant flux energy per unit solid angle, per projected area at time  $t$ . It is given by the following equation:

$$\left( \frac{1}{v} \frac{\partial}{\partial t} + \mathbf{s} \cdot \nabla + \mu_a(\mathbf{r}) + \mu_s(\mathbf{r}) \right) \psi(\mathbf{r}, \mathbf{s}', t) = \mu_s(\mathbf{r}) \int_{S^2} p(\mathbf{s} \cdot \mathbf{s}') \psi(\mathbf{r}, \mathbf{s}', t) d\mathbf{s}' + q(\mathbf{r}, \mathbf{s}, t). \quad (2.9)$$

Here  $\psi(\mathbf{r}, \mathbf{s}, t)$  is the radiance in units of  $[\text{W}/\text{cm}^2/\text{sr}]$  and  $q(\mathbf{r}, \mathbf{s}, t)$  is the source density in the unit of  $[\text{W}/\text{cm}^3/\text{sr}]$  at position  $\mathbf{r}$  at time  $t$  in direction  $\mathbf{s}$ ,  $\mu_a(\mathbf{r})$  and  $\mu_s(\mathbf{r})$  represent the absorption coefficient and scattering coefficient that are spatial functions with respect to  $\mathbf{r}$ ,  $v$  represents the light speed in the medium. The terms on the right-hand side represent the radiance gained per unit volume at position  $\mathbf{r}$  on direction  $\mathbf{s}$  at time  $t$ , including photons from the source and photon deflected into direction  $\mathbf{s}$  after scattering happens. The terms on the left hand side correspond to the radiance lost per unit volume at position  $\mathbf{r}$  on direction  $\mathbf{s}$  at time  $t$ , including the radiance change over time and photons that are absorbed by the media or scattered into other directions.

To take into account the mismatch of the refractive index at air-tissue interface, the partially reflective boundary condition is commonly imposed,

$$\psi(\mathbf{r}_b, \mathbf{s}) = q(\mathbf{r}_b, \mathbf{s}) + R(-\mathbf{s} \cdot \mathbf{n})\psi(\mathbf{r}_b, \mathbf{s}'), \quad \mathbf{s} \cdot \mathbf{n} < 0. \quad (2.10)$$

Here  $\mathbf{r}_b$  represents the position on the boundary,  $\mathbf{n}$  is the unit outgoing normal vector at  $\mathbf{r}_b$ ,  $\mathbf{s}' = \mathbf{s} - 2(\mathbf{s} \cdot \mathbf{n})\mathbf{n}$  is the specular reflection of  $\mathbf{s}$ ,  $R(-\mathbf{s} \cdot \mathbf{n})$  is the reflectivity obtained by Fresnel's equations,

$$R(\cos \theta) = \begin{cases} 1, & n \sin \theta > n' \\ \frac{1}{2} \left( \frac{n \cos \theta - n' \cos \theta'}{n \cos \theta + n' \cos \theta'} \right)^2 + \frac{1}{2} \left( \frac{n \cos \theta' - n' \cos \theta}{n \cos \theta' + n' \cos \theta} \right)^2, & n \sin \theta \leq n' \end{cases} \quad (2.11)$$

Here  $n$  and  $n'$  are the refractive index inside and outside the medium,  $\theta' := \arcsin(n \sin \theta / n')$  is determined by the Snell's law. Time  $t$  is omitted in (2.10) since it can be used as boundary condition for FD-RTE and CW-RTE that are described below.

Applying a Fourier Transform to equation (2.9), the corresponding frequency-domain RTE (FD-RTE) can be derived as [43]:

$$\left( \frac{i\omega}{v} + \mathbf{s} \cdot \nabla + \mu_a(\mathbf{r}) + \mu_s(\mathbf{r}) \right) \psi(\mathbf{r}, \mathbf{s}, \omega) = \mu_s(\mathbf{r}) \int_{S^2} p(\mathbf{s} \cdot \mathbf{s}') \psi(\mathbf{r}, \mathbf{s}', \omega) d\mathbf{s}' + q(\mathbf{r}, \mathbf{s}, \omega). \quad (2.12)$$

where  $i = \sqrt{-1}$  and  $\omega$  represents the modulation frequency, which is sometimes omitted if it is fixed.

A continuous-wave RTE (CW-RTE) is a special case of FD-RTE, which is obtained from (2.12) by setting the modulation frequency  $\omega = 0$ .

### 2.2.2. Diffusion approximation to RTE

Under this assumption of isotropic scattering (e.g. each scattering direction has the same probability), the RTE can be approximated with the diffusion equation (DE). The time-domain DE is given as [43]:

$$\left( \frac{1}{v} \frac{\partial}{\partial t} + \mu_a(\mathbf{r}) \right) \phi(\mathbf{r}, t) - \nabla \cdot D(\mathbf{r}) \nabla \phi(\mathbf{r}, t) = q(\mathbf{r}, t). \quad (2.13)$$

Here  $\phi$  [W/cm<sup>2</sup>] represents the photon density, the diffusion coefficient  $D(\mathbf{r})$  is defined as  $D(\mathbf{r}) := 1/3(\mu_a(\mathbf{r}) + \mu'_s(\mathbf{r}))$ , and  $q(\mathbf{r}, t)$  represents an isotropic source.

Similarly, the frequency-domain DE (FD-DE) can be obtained by taking Fourier Transform on (2.13):

$$\left(\frac{1}{v} \frac{\partial}{\partial t} + \mu_a(\mathbf{r})\right) \phi(\mathbf{r}, \omega) - \nabla \cdot D(\mathbf{r}) \nabla \phi(\mathbf{r}, \omega) = q(\mathbf{r}, \omega), \quad (2.14)$$

and the continuous-wave DE (CW-DE) can be given by setting the modulation frequency in (2.14).

### 2.3. Diffuse optical tomography as an optimization problem

In DOT, low-energy NIR light is used to probe biological tissue. Measurements of transmitted and reflected light intensities are used to recover a spatial distribution of various optical properties, for instance, absorption and scattering coefficients ( $\mu_a$  and  $\mu_s$ ) inside the medium under investigation. From these coefficients other important physiological information can be derived, such as the concentrations of oxy-hemoglobin ( $\text{HbO}_2$ ), deoxy-hemoglobin (Hb), total hemoglobin, and water ( $\text{H}_2\text{O}$ ). These physiological parameters and tissues optical properties vary with location inside the tissue [44].

Unlike X-ray CT, no closed-form inversion expression for the general image reconstruction problem in DOT exists. Therefore, instead of the closed-form inverse Radon transform employed in X-ray CT, DOT inverse problems are mostly formulated as an optimization problem [43, 45], which is solved through iterative algorithms. In this approach the spatial distribution of the optical properties are determined by finding the best agreement between the experimental measurements and the results of the theoretical light propagation model with a given source condition. A flowchart for the general reconstruction process in DOT is shown in Fig. 2.1.



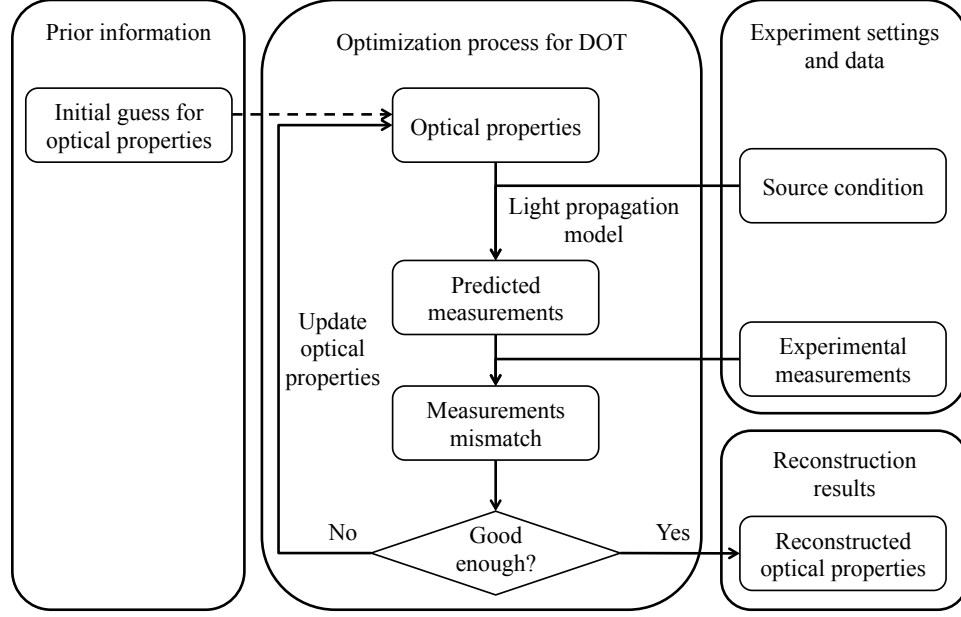


Figure 2.1: Flowchart for iterative reconstruction process in DOT.

As indicated in Fig. 2.1, the light propagation model plays a very important role in DOT. Improper light propagation models lead to incorrect reconstruction results. DOT based on the DE as the forward model is widely used because of its simplicity and low computational cost. However, DE-based results are not accurate enough when media are considered that contain areas of where  $\mu_a > \mu_s$ , void-like regions, or are optically thin (small volume). Clinically these cases occur when one images finger joints (small tissue volume), hematoma and hemorrhages in the brain (high absorbing media), or the brain surface, which is embedded in low-scattering cerebrospinal fluid. Light propagation in these types of tissues can be much more accurately modeled by RTE-based codes [1].

Compared to DE-based DOT, RTE-DOT is more challenging because of the higher ill-posedness and computational cost. This limits its widespread use in clinical applications. The difficulties mainly come from the following four aspects.

First of all, the light intensity distribution in the RTE is a function of both position and direction. Thus, unlike in the DE, the RTE requires a directional discretization in the numerical

solver in addition to a spatial discretization. This leads to a much larger scale linear system, which is much larger than the DE-based linear system that requires spatial discretization alone.

Secondly, the RTE is a partial integro-differential equation. This leads to much more non-zero off-diagonal entries in the linear system due to the integral term. In turn, this further lowers the convergence rate of any solver. Since RTE needs to be solved repeatedly during the image reconstruction process, the efficiency of the numerical solver is a critical factor that affects the overall performance of RTE-DOT.

Thirdly, the reconstruction problem in DOT is in general formulated as a large scale, ill-posed and non-convex optimization problem. The efficiency of the solver highly depends on the formulation and the optimization algorithm. Thus a proper formulation with a robust and efficient algorithm for DOT is required.

Finally, more measurements are required to alleviate the ill-posedness and improve the quality of the reconstructed images. However, the number of measurements in practice is limited by commonly used optical-fiber-based imaging systems. Non-contact imaging systems with CCD camera as the detector unit can provide more measurement points, however, in RTE-based non-contact imaging system, a method that fully models the angular dependent light propagation in free space is required.

These problems, due to the significant importance and irreplaceability of RTE-DOT in many applications, have attracted much attention in recent years [9-11, 46-52]. Yet, RTE-DOT still remains a challenging problem and there is considerable room to improve the efficiency of the solvers without sacrificing accuracy.

## 2.4. Discussion

The widespread use of the RTE-DOT is limited by its high computational cost and ill-posedness. The aim of the work in this dissertation is to construct a novel reconstruction algorithm for RTE-DOT to make it more applicable in practice. This goal is achieved through three components.

First of all, I improve the efficiency in the forward problem solver in RTE-DOT, which is an important component in the reconstruction algorithm. Specifically, the forward problems in RTE-DOT are discretized as an all-at-once linear equation with multiple right-hand sides. Instead of solving for each right-hand side separately, I introduce a novel linear solver called block biconjugate gradient stabilized method (block BiCGStab) that makes full use of the shared information between different right-hand sides to accelerate solution convergence. Additionally, two multi-threading block BiCGStab methods are proposed for additional acceleration under limited threads and shared memory situation.

Second, I investigate two novel inverse solvers for the optimization problem in RTE-DOT. First, a novel line-search-free BFGS algorithm is proposed. This algorithm improves the efficiency of the traditional BFGS algorithm, one of the state of the art algorithm in unconstrained optimization, by replacing the line search step with a more accurate and efficiency step size estimation procedure. Furthermore, I implement the first full-space algorithm for RTE-DOT, which pursue the descent direction in the full Hessian space instead of the reduced Hessian space. Therefore, it is expected to converge faster than the existing reduced-space algorithms.

Third, I proposed a back ray-tracing model to simulate the photon transport process in free-space. Unlike current available models, the new approach fully considers the angular dependency of intensity and thus can be applied in RTE-DOT. Moreover, this proposed model

could handle photon transport problems with a generalized optical system between the object and the CCD camera to collect more signals that is often limited by the size of aperture. Thus it is reasonable to expect a better performance and more reliable result in non-contact DOT with this proposed model.

Finally, a novel reconstruction algorithm for RTE-DOT is constructed by a combination of these three components. This reconstruction algorithm is then tested and validated with data from preclinical and clinical studies. It shows this newly developed algorithm enhances the applicability of RTE-DOT in clinical study by significantly reducing the reconstruction time and its capability of being applied with general non-contact image system.

## Chapter 3. Efficient Forward Solver in RTE-DOT

### 3.1. Introduction

The forward solver is a crucial component in DOT since this solver is called multiple times in most common iterative image reconstruction codes. Indeed, computing solutions of the forward problem typically makes up about 90% of the overall computational time. Hence, efficient fast forward solvers in RTE-DOT have been pursued by several groups [46, 53-55]. However, all existing algorithms for solving the RTE are designed based on solving a single right-hand side. Therefore, only one light-source is considered at any given time. Yet, in DOT one typically uses many light sources to illuminate the tissue of interest from many different positions. In this way many different light-transmission data sets (“views”) are obtained that contain different information. Used in an inverse image reconstruction algorithm this leads to more accurate images. Employing multiple light sources means the inverse problem involves multiple right-hand sides.

To increase the computational speed, some groups have recently implemented parallel computing methods, which make use of multiple processors, available on most modern computer systems [47, 56]. However, in these parallel computing methods each processor still solves a single right-hand-side problem, which required substantial computational resources that are very expensive and not always available. Therefore, I focus in this thesis on numerical methods for solving multiple right-hand sides simultaneously on a single PC with limited threads (threads number is less than or equal to the source number).

To solve linear equations with multiple right-hand sides efficiently, I make full use of the fact that the same coefficient matrix is shared among multiple right-hand sides in the linear system resulting from multiple source illumination:

$$A\Psi = B, \quad (3.1)$$

where  $\Psi$  is the matrix of solution vectors  $\psi^{(1)}, \psi^{(2)}, \dots, \psi^{(N_{\text{source}})}$  and  $B$  of right-hand sides  $b^{(1)}, b^{(2)}, \dots, b^{(N_{\text{source}})}$  pertaining to the  $i$ th light source illumination. Methods for solving such linear systems with multiple right-hand sides have been extensively studied in other areas [57-62]. The block Krylov subspace methods have been shown to be the most effective for many cases [63, 64]. In this dissertation, the Krylov subspace block BiCGStab algorithm [65] is introduced. Compared to other methods, this approach has the advantages of low memory requirement, simple structure and stable convergence. In addition, it can be readily combined with other numerical techniques such as high order differencing schemes or acceleration schemes to obtain additional speedup or increased accuracy.

Furthermore, to obtain a higher convergence rate, an efficient preconditioner is necessary for the Krylov subspace type linear solvers. Incomplete LU factorization (ILU) and its variants [66-70] are among the most popular preconditioners for large-scale sparse linear systems and have been already applied into many areas [71-73]. However, ILU's efficiency is jeopardized by the dense non-zero pattern generated by the integral term in RTE. To overcome this difficulty, a reduced linear operator is defined with much fewer non-zero entries to approximate the original linear system and the ILU factorization is conducted on this reduced linear system instead of the original one as the preconditioner. Although the convergence rate provided by the proposed preconditioner is slightly lower than the traditional ILU preconditioner, the factorization time and preconditioning time is significantly reduced, due to its much smaller non-zero entry number. Therefore, a lower computational cost for the overall solving process can be expected.

The remainder of the chapter is organized as follows. The discretization of the RTE and the novel preconditioning technique are described in Section 3.2. Then the efficient linear solvers

are introduced in Section 3.3, including the preconditioned BiCGStab, block BiCGStab and multi-threading block BiCGStab. Performance of the proposed preconditioner and block solvers is evaluated through extensive numerical experiments in Section 3.4. This chapter concludes with a discussion in Section 3.5.

## 3.2. Discretization and preconditioning of RTE

### 3.2.1. Upwind finite volume discrete-ordinates discretization

The RTE (2.12) is discretized by a finite volume scheme combined with the discrete ordinates method ( $S_n$ ) [52]. First we discretized the direction domain with discrete ordinates method, thus the integral term on the right-hand side of Eq. (2.12) can be approximated with a weighted sum of the radiance field in different directions,

$$\int_{S^2} p(\mathbf{s}_i \cdot \mathbf{s}') \psi(\mathbf{r}, \mathbf{s}') d\mathbf{s}' = \sum_{k=1}^{N_{SA}} p_{ik} w_k \psi(\mathbf{r}, \mathbf{s}_k), \quad (3.2)$$

where  $N_{SA} = n(n+2)$  is the total number of solid angles in  $S_n$ ,  $p_{ik} = p(\mathbf{s}_i \cdot \mathbf{s}_k)$ ,  $w_k$  is the spherical measure of solid angle  $\mathbf{s}_k$ .

Thus based on Eq. (3.2), Eq. (2.12) is decomposed into a set of  $N_{SA}$  coupled partial differential equations that correspond to  $N_{SA}$  discretized photon propagation directions,

$$\left( \frac{i\omega}{v} + \mathbf{s}_i \cdot \nabla + \mu_a(\mathbf{r}) + \mu_s(\mathbf{r}) \right) \psi_i(\mathbf{r}) = \mu_s(\mathbf{r}) \sum_{k=1}^{N_{SA}} p_{ik} w_k \psi_k(\mathbf{r}) + q_i(\mathbf{r}), i = 1, \dots, N_{SA}. \quad (3.3)$$

where  $\psi(\mathbf{r}, \mathbf{s}_i)$  is written in short as  $\psi_i(\mathbf{r})$ .

The spatial domain is discretized with a node-centered mesh combined with the upwind differencing scheme, which not only can handle complex geometry but also guarantee the conservation law of energy. The resulting discretized RTE is given by

$$\frac{1}{V_j} \sum_{l=1}^{N_j} A_j^l (\mathbf{n}_j^l \cdot \mathbf{s}_i) f_{i,j}^l + \mu_{a,j} \psi_{i,j} + \mu_{s,j} \left( \psi_{i,j} - \sum_{k=1}^{N_{SA}} p_{ik} w_k \psi_{k,j} \right) + \frac{i\omega}{v} \psi_{i,j} = q_{i,j}, j = 1, \dots, N_{CV} \quad (3.4)$$

where  $\mu_{a,j}$ ,  $\mu_{s,j}$ ,  $\psi_{i,j}$  and  $q_{i,j}$  represent  $\mu_a(\mathbf{r}_j)$ ,  $\mu_s(\mathbf{r}_j)$ ,  $\psi_i(\mathbf{r}_j)$  and  $q_i(\mathbf{r}_j)$  in Eq. (3.4) respectively,  $V_j$  and  $N_j$  denote the volume and the neighbor number of the control volume centered at  $\mathbf{r}_j$ ,  $A_j^l$  and  $\mathbf{n}_j^l$  are the area and outgoing unit normal vector of the  $l$ th surface of the control volume centered at  $\mathbf{r}_j$ ,  $N_{CV}$  is the total number of control volumes. With a slight abuse of notation, we will from now on use  $V_j$  and  $A_j^l$  to indicate the control volume centered at  $\mathbf{r}_j$  and its  $l$ th surface. The  $f_{i,j}^l$  in Eq. (3.4) represents the directional flux per area on  $A_j^l$ , it can be approximated by the upwind scheme as applied to unstructured meshes: if  $A_j^l$  is not on the boundary and its neighboring volume of  $V_j$  is indexed with  $j_l$ , then  $f_{i,j}^l$  is given by

$$f_{i,j}^l = \begin{cases} \psi_{i,j}, & \mathbf{n}_j^l \cdot \mathbf{s}_i > 0 \\ \psi_{i,j_l}, & \mathbf{n}_j^l \cdot \mathbf{s}_i \leq 0 \end{cases} \quad (3.5)$$

if  $A_j^l$  is on the boundary, then  $f_{i,j}^l$  can be obtained by

$$f_{i,j}^l = \begin{cases} \psi_{i,j}, & \mathbf{n}_j^l \cdot \mathbf{s}_i > 0 \\ \bar{\psi}_{i,j}, & \mathbf{n}_j^l \cdot \mathbf{s}_i \leq 0 \end{cases} \quad (3.6)$$

where  $\bar{\psi}_{i,j}$  is given by the discretization of the reflective boundary condition (2.10) as follows,

$$\bar{\psi}_{i,j} = q_{i,j} + R(-\mathbf{n}_j^l \cdot \mathbf{s}_i) \psi_{i',j} \quad (3.7)$$

where  $\psi_{i',j}$  represents  $\psi(\mathbf{r}_j, \mathbf{s}_{i'})$  and  $\mathbf{s}_{i'}$  denotes the reflective direction of  $\mathbf{s}_i$  on  $A_j^l$  that is given by  $\mathbf{s}_i - 2(\mathbf{s}_i \cdot \mathbf{n}_j^l) \mathbf{n}_j^l$ . In practice, the existence of  $\mathbf{s}_i - 2(\mathbf{s}_i \cdot \mathbf{n}_j^l) \mathbf{n}_j^l$  is not guaranteed in  $S_N$  so we choose  $\mathbf{s}_{i'}$  as the direction in  $S_N$  which is closest to  $\mathbf{s}_i - 2(\mathbf{s}_i \cdot \mathbf{n}_j^l) \mathbf{n}_j^l$ .



Finally, by combining (3.4)~(3.7), we arrive at the following system of the algebraic equations for multiple sources:

$$A\psi^{(k)} = b^{(k)}, k = 1, \dots, N_{\text{source}}. \quad (3.8)$$

Here  $b^{(k)}$  is the right-hand side generated with the  $k$ th source configuration,  $\psi^{(k)}$  is the corresponding solution which is a vector formed by  $\psi_{i,j}$  with all possible  $i$  and  $j$ .  $N_{\text{source}}$  is the source number and  $A$  is a  $N \times N$  linear operator for the discretized RTE, where  $N := N_{\text{CV}} \times N_{\text{SA}}$ . It is easy to check that the source configuration only contributes on the right-hand side term but is independent to  $A$  in Eq. (3.8). So we can write an all-at-once equation for all source configurations since they all share the same linear operator as (3.1).

In practice, the matrix  $A$  in Eq. (3.8) and (3.1) is not constructed explicitly. Instead, we write Eq. (3.4) into the following matrix expression

$$A_1\psi + A_2\psi + A_3\psi + i\frac{\omega}{v}\psi = b, \quad (3.9)$$

where  $A_1$ ,  $A_2$  and  $A_3$  are three matrices corresponding to the first three terms in Eq. (3.4). Then by definition we have

$$A = A_1 + A_2 + A_3 + i\frac{\omega}{v}I^{(N)}. \quad (3.10)$$

Here  $A_1$  is a constant matrix that only depends on the discretization,  $I^{(N)}$  is a  $N \times N$  identity matrix,  $A_2$  and  $A_3$  are two block diagonals as follows,

$$A_2 = \begin{bmatrix} \mu_{a,1}V_1I^{(N_{\text{SA}})} & & \\ & \ddots & \\ & & \mu_{a,N_{\text{CV}}}V_{N_{\text{CV}}}I^{(N_{\text{SA}})} \end{bmatrix}, \quad A_3 = \begin{bmatrix} \mu_{s,1}V_1W & & \\ & \ddots & \\ & & \mu_{s,N_{\text{CV}}}V_{N_{\text{CV}}}W \end{bmatrix}, \quad (3.11)$$

where  $I^{(N_{SA})}$  is a  $N_{SA} \times N_{SA}$  identity matrix and  $W$  is a  $N_{SA} \times N_{SA}$  matrix with  $W_{ij} = p_{ij}w_j$ . Noticed  $A_2$  and  $A_3$  are functions of  $\mu_a$  and  $\mu_s$  respectively, where  $\mu_a := (\mu_{a,1}, \dots, \mu_{a,N_{CV}})$  and  $\mu_s := (\mu_{s,1}, \dots, \mu_{s,N_{CV}})$ , (3.10) is also written as

$$A(\mu_a, \mu_s) = A_1 + A_2(\mu_a) + A_3(\mu_s) + i \frac{\omega}{v} I^{(N)}. \quad (3.12)$$

In iterative linear solvers, the explicit form of  $A$  is not necessary if the expression of the matrix-vector multiplication is provided. According to (3.10) and (3.11), the information of matrix  $A$  can be stored in a efficient way to the memory by saving three vectors  $\mu_a, \mu_s$  and  $V$ , two matrices  $A_1$  and  $W$  and a scalar  $\omega/v$ .

The scattering phase function for most biological tissues is strongly forward-peaked ( $g \geq 0.8$ ) [44]. Thus a large  $N_{SA}$  in Eq. (3.2) is required to accurately describe such highly anisotropic scattering behavior, which leads to a considerable computational burden and memory requirement. To overcome this difficulty, Hielscher and Klose introduced the Delta-Eddington method [74], which allows using only a small number of discrete ordinates in RTE to approximate the original scattering function given by Eq. (2.7). In this dissertation the zeroth-order Delta-Eddington approximation is employed since it has shown promising results as reported in literature [45, 75, 76]. The frequency-domain RTE is simplified with zeroth-order Delta-Eddington approximation as

$$\left( \frac{i\omega}{v} + \mathbf{s} \cdot \nabla + \mu_a(\mathbf{r}) + \mu'_s(\mathbf{r}) \right) \psi(\mathbf{r}, \mathbf{s}) = \frac{\mu'_s(\mathbf{r})}{4\pi} \int_{S^2} \psi(\mathbf{r}, \mathbf{s}') d\mathbf{s}' + q(\mathbf{r}, \mathbf{s}), \quad (3.13)$$

where the scattering coefficient  $\mu_s$  and the anisotropic factor  $g$  is combined into a reduced scattering coefficient  $\mu'_s := (1 - g)\mu_s$ .

### 3.2.2. Preconditioning with incomplete LU factorization on the reduced linear operator

The convergence rate of most iterative linear solvers for Eq. (3.8) or (3.1) highly depends on the matrix  $A$ 's condition number. A good preconditioner can significantly improve  $A$ 's condition number therefore reduce the required total iteration and overall computational time.

The most widely used preconditioners for solving general-sparse and large-scale linear equations are multigrid preconditioners [77-80] and incomplete-factorization-based (IFB) preconditioners [66-70, 81, 82]. Compared to multigrid preconditioners, IFB preconditioners have two main advantages. First, their implementation is often simpler. Second, they don't have the overhead required for setting up the multi-level meshes. Moreover, the factorization of the linear operator is independent of the right-hand side, so they are more suitable for solving multiple right hand sides simultaneously. Therefore, IFB preconditioners are preferred here. For RTE, the incomplete LU factorization (ILU) is used for preconditioning since it is considered as the most popular incomplete factorization based preconditioning techniques for non-Hermitian linear system.

The general idea of the ILU preconditioning technique is described as follows: First, an incomplete LU factorization is performed on  $A$  such that  $A \approx LU$ , where  $L$  and  $U$  are lower and upper triangular sparse matrices respectively. Then the preconditioner  $P$  is defined as  $P = LU$ , and instead of solving  $Ax = b$ , we solve a left preconditioned equation  $P^{-1}Ax = P^{-1}b$  (other options could be right preconditioned equation  $AP^{-1}y = b$  or two-sides preconditioned equation  $L^{-1}AU^{-1}y = b$ ). The new matrix  $P^{-1}A$  often has a much smaller condition number since  $P^{-1}A \approx I$ . On the other hand, the matrix-vector multiplication  $P^{-1}Av$  in the preconditioned linear system only introduces little extra computation as compared to  $Av$  since  $y = P^{-1}v$  can be efficiently obtained by solving  $Lw = v$  and  $Uy = w$  with back substitution algorithm.

However, there are two main difficulties with the traditional ILU preconditioner for matrix  $A$ . First, as aforementioned, the matrix  $A$ , which is required in ILU factorization, is not constructed explicitly for memory saving consideration. Moreover, the computational time related to ILU preconditioner, including the factorization time and the preconditioning time, highly depends on the sparsity of the matrix  $A$ , more non-zero entries inevitably lead to more expensive computational cost. However, the matrix  $A$  in Eq. (3.8) and (3.1) contains relative large amount of non-zero entries derived from the integral term in Eq. (2.12), that may potentially lower the preconditioner's efficiency.

To overcome these difficulties, a novel preconditioner is proposed. First, a reduced linear operator  $\bar{A}$  to approximate the original matrix is defined as

$$\bar{A} := A_1 + A_2 + \text{diag}(A_3),$$

where matrices  $A_1$ ,  $A_2$  and  $A_3$  are defined the same as shown in Eq. (3.9),  $\text{diag}(A_3)$  represents the main diagonal of  $A_3$ . Then we perform the ILU factorization on the reduced linear operator  $\bar{A}$  as  $\bar{A} \approx \bar{L}\bar{U}$ . Finally the preconditioner  $P$  for the original matrix  $A$  is given by  $P = \bar{L}\bar{U}$ . The proposed preconditioner has several advantages: First, it does not require the explicit expression of matrix  $A$  and the reduced matrix  $\bar{A}$  can be easily obtained by a slight modification on  $D$ 's diagonal. Moreover,  $\bar{A}$  has the same non-zero pattern as  $D$ , which has significantly less non-zero entries than  $A$ . That will lead to a much faster factorization and preconditioning process. Finally,  $\text{diag}(A_3)$  preserves considerable proportion information in  $A_3$  since it is a strongly diagonal dominant matrix, thus  $\bar{A}$  is a good approximation of  $A$ .

### 3.2.3. Node-centered finite volume mesh generation

To convert the discretized RTE (3.4) into a linear equation as (3.8) or (3.1), the computational domain needs to be discretized with non-overlap finite volumes and certain geometry information for every control volume is required, such as its volume, its connectivity to neighbors and so on. Such information is determined only by the discretization itself and is totally independent of the optical properties and locations of sources and detectors. Therefore, all necessary geometry information for constructing (3.8) or (3.1) are extracted during spatial discretization and store it into a data structure named with FVM mesh in this dissertation. The FVM data structure is shown in Table 3.1.

Table 3.1: Data structures used for FVM mesh's construction.

Structure name	Contents
FVM mesh	<ul style="list-style-type: none"> <li>• Vector of control volumes (control volume is defined below)</li> </ul>
Control volume	<ul style="list-style-type: none"> <li>• Volume (float number)</li> <li>• Vector of surfaces (surface is defined below)</li> </ul>
Surface	<ul style="list-style-type: none"> <li>• Area (float number)</li> <li>• Normal vector (vector of float numbers)</li> <li>• Neighbor index (integer)</li> <li>• Distance to the neighbor centroid (float number)</li> </ul>

To obtain the FVM mesh, the computational domain needs to be discretized with structured or unstructured grids with which necessary information is extracted. The fundamental difference between structured and unstructured grids is the regularity of the connectivity of grids [83]. Structure grids are identified by regular connectivity, in another word, the grids can be expressed as a 2D or 3D array in computational memory and the neighborhood relationships can be defined by their indices (examples include quadrilateral grids in 2D and hexahedra grids in 3D). Whereas unstructured grids are identified by irregular connectivity, that means the connectivity information is not contained in their indices and must be specified explicitly

(examples include triangle grids in 2D and tetrahedron grids in 3D). With structured grids, it is usually much more straightforward to extract useful information in Table 3.1 for constructing the FVM mesh. However, generating high quality structured grids for an arbitrary object with complex boundary is very challenging and time-consuming [84], which makes it impractical. On the other hand, unstructured grids are more suitable for complex geometries since those grids can fit the physical boundary more accurately.

Based on these considerations, the computational region is discretized with unstructured grids (triangle grids in 2D or tetrahedrons in 3D) in this dissertation. This process is done with commercial software named GID (Micromechatronics Inc. at <http://www.mmech.com/>). A 2D example of triangle grids generated with GID is shown in Fig. 3.1(a). In practice, the grid-node-ratio is approximately 2 for 2D triangle grids and usually 4~6 for 3D tetrahedron grids. Therefore, to reduce the number of unknowns in the linear system (3.8) and (3.1), the median dual mesh [85] of the original GID mesh is used as the FVM mesh, in which every control volume is centered at an original node and is surrounded by boundaries formed by connecting the centroids of original grids, faces and edges. Fig. 3.1(a)'s corresponding 2D FVM mesh is shown in Fig. 3.1(b), the red dash lines represent the boundaries of control volumes.

With the GID mesh, coordinates information for every node and connectivity between nodes are provided. To complete the FVM mesh structure defined in Table 3.1, Algorithm 3.1 and 3.2 have been proposed to convert a GID mesh to a FVM mesh in both 2D and 3D cases, in which  $N_{\text{node}}$  and  $N_{\text{grid}}$  represent the total number of nodes and grids in the original GID mesh,  $\mathbf{n}_k$  is used to indicate the  $k$ th node as well as its coordinates in the GID mesh, we use a set of nodes to indicate a grid, a face, or an edge. It is easy to check that Algorithm 3.1 and 3.2 have the optimal computational complexity as  $\Theta(N_{\text{elem}})$  and a very low extra memory requirement as

$O(N_{\text{elem}})$  for saving the boundary set  $S_{\text{boundary}}$ . These costs are negligible as compared to the computational cost and memory requirement in the forward iterative linear solver.

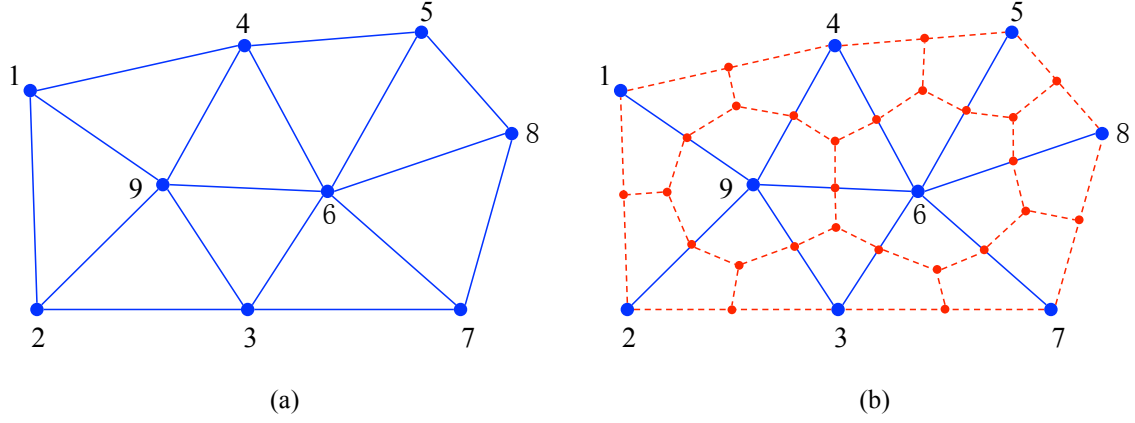


Figure 3.1: Illustration of 2D unstructured grids and node-centered FVM mesh. (a) 2D unstructured grids; (b) 2D node-centered FVM mesh (grids are surrounded with red dash boundaries)

---

**Algorithm 3.1: 2D GID mesh to FVM mesh conversion**

---

1. Allocate  $N_{\text{node}}$  spaces in the control volume vector CVs and set  
    CVs[i].volume = 0  
    CVs[i].surfaces =  $\emptyset$   
    for  $i = 1, \dots, N_{\text{node}}$ .
  2. Create an set  $S_{\text{boundary}} = \emptyset$ .
  3. Extract information for every triangle grid:  
    **for** the  $j$ th triangle grid  $T_j = \{\mathbf{n}_{k_1}, \mathbf{n}_{k_2}, \mathbf{n}_{k_3}\}, j = 1, 2, \dots, N_{\text{elem}}$ :  
        (1) Update boundary set  $S_{\text{boundary}}$ :  
            **for**  $l = 1$  to 3:  
                Construct a tuple  $t := (\{k_1, k_2, k_3\} - \{k_l\}, k_l)$   
                **if**  $t \in S_{\text{boundary}}$  **then**  $S_{\text{boundary}} = S_{\text{boundary}} - \{t\}$   
                **else**  $S_{\text{boundary}} = S_{\text{boundary}} \cup \{t\}$   
                **end if**  
            **end for**  
        (2) Update interior surfaces and volume for every control volume:  
            **for** every  $(k_1, k_2, k_3)$ 's permutation  $(k'_1, k'_2, k'_3)$ :  
                **for**  $l = 1$  to 3:  
                    Set  $\mathbf{c}_l$  as  $\{\mathbf{n}_{k'_1}, \dots, \mathbf{n}_{k'_l}\}$ 's centroid  
                **end for**  
                CVs[k'\_1].volume = CVs[k'\_1].volume + volume( $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$ )  
                Construct a surface  $s$  and set  
                     $s.\text{area} = |\mathbf{c}_2 - \mathbf{c}_3|$   
                     $s.\text{normal\_vector} = (\mathbf{c}_2 - \mathbf{c}_1) / |\mathbf{c}_2 - \mathbf{c}_1|$   
                     $s.\text{neighbor\_index} = k'_2$   
                     $s.\text{dist\_to\_neighbor} = 2|\mathbf{c}_2 - \mathbf{c}_1|$   
                    CVs[k'\_1].surfaces = CVs[k'\_1].surfaces  $\cup \{s\}$   
                **end for**  
            **end for**  
4. Update boundary surfaces for every control volume:  
    **for** every  $t = (\{k_1, k_2\}, k_3) \in S_{\text{boundary}}$ :  
        Construct a surface  $s$  and set  
             $s.\text{area} = |\mathbf{n}_{k_1} - \mathbf{n}_{k_2}|/2$   
             $s.\text{normal\_vector} = \mathbf{n}$  with  $\mathbf{n} \perp (\mathbf{n}_{k_1} - \mathbf{n}_{k_2}), |\mathbf{n}| = 1, \mathbf{n} \cdot (\mathbf{n}_{k_1} - \mathbf{n}_{k_3}) > 0$   
             $s.\text{neighbor\_index} = -1$  (boundary surface has no neighbor)  
             $s.\text{dist\_to\_neighbor} = -1$   
            CVs[k\_1].surfaces = CVs[k\_1].surfaces  $\cup \{s\}$   
            CVs[k\_2].surfaces = CVs[k\_2].surfaces  $\cup \{s\}$   
    **end for**
-



---

**Algorithm 3.2: 3D GID mesh to FVM mesh conversion**

---

1. Allocate  $N_{\text{node}}$  spaces in the control volume vector CVs and set  
CVs[i].volume = 0  
CVs[i].surfaces =  $\emptyset$   
for  $i = 1, \dots, N_{\text{node}}$ .
  2. Create an set  $S_{\text{boundary}} = \emptyset$ .
  3. Extract information for every tetrahedron grid:  
for the  $j$ th triangle grid  $T_j = \{\mathbf{n}_{k_1}, \mathbf{n}_{k_2}, \mathbf{n}_{k_3}, \mathbf{n}_{k_4}\}, j = 1, 2, \dots, N_{\text{elem}}$ :
    - (1) Update boundary set  $S_{\text{boundary}}$ :  
for  $l = 1$  to 4:  
Construct a tuple  $t := (\{k_1, k_2, k_3, k_4\} - \{k_l\}, k_l)$   
if  $t \in S_{\text{boundary}}$  then  $S_{\text{boundary}} = S_{\text{boundary}} - \{t\}$   
else  $S_{\text{boundary}} = S_{\text{boundary}} \cup \{t\}$   
end if  
end for
    - (2) Update interior surfaces and volume for every control volume:  
for every  $(k_1, k_2, k_3, k_4)$ 's permutation  $(k'_1, k'_2, k'_3, k'_4)$ :  
for  $l = 1$  to 3:  
set  $\mathbf{c}_l$  as  $\{\mathbf{n}_{k'_1}, \dots, \mathbf{n}_{k'_l}\}$ 's centroid  
end for  
CVs[k'\_1].volume = CVs[k'\_1].volume + volume( $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ )  
Construct a surface  $s$  and set  
s.area = area( $\{\mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ )  
s.normal\_vector =  $(\mathbf{c}_2 - \mathbf{c}_1) / |\mathbf{c}_2 - \mathbf{c}_1|$   
s.neighbor\_index =  $k'_2$   
s.dist\_to\_neighbor =  $2|\mathbf{c}_2 - \mathbf{c}_1|$   
CVs[k'\_1].surfaces = CVs[k'\_1].surfaces  $\cup \{s\}$   
end for  
end for
  4. Update boundary surfaces for every control volume:  
for every  $t = (\{k_1, k_2, k_3\}, k_4) \in S_{\text{boundary}}$ :  
Construct a surface  $s$  and set  
s.area = area( $\{\mathbf{n}_{k_1}, \mathbf{n}_{k_2}, \mathbf{n}_{k_3}\}$ )/3  
s.normal\_vector =  $\mathbf{n}$  with  $\mathbf{n} \perp \{\mathbf{n}_{k_1}, \mathbf{n}_{k_2}, \mathbf{n}_{k_3}\}, |\mathbf{n}| = 1, \mathbf{n} \cdot (\mathbf{n}_{k_1} - \mathbf{n}_{k_4}) > 0$   
s.neighbor\_index = -1 (boundary surface has no neighbor)  
s.dist\_to\_neighbor = -1  
CVs[k\_1].surfaces = CVs[k\_1].surfaces  $\cup \{s\}$   
CVs[k\_2].surfaces = CVs[k\_2].surfaces  $\cup \{s\}$   
CVs[k\_3].surfaces = CVs[k\_3].surfaces  $\cup \{s\}$   
end for
-

### 3.3. Preconditioned block linear solver with multiple sources

The efficient forward solvers for solving RTE are introduced in this section. We start with a brief introduction to traditional Krylov subspace methods and the biconjugate gradient stabilized method (BiCGStab). Then we move on to the block Krylov subspace methods and the block BiCGStab algorithm. Finally the further acceleration with multi-threading technique is discussed.

#### 3.3.1. Krylov subspace methods and BiCGStab algorithm

Krylov subspace methods are the most widely used iterative methods so far for large-scale sparse linear system  $Ax = b$ . In Krylov subspace methods, an affine space  $K_n(A, r_0)$  that grows with iteration number  $n$  is defined as follows:

$$K_n(A, r_0) := \text{span}(r_0, Ar_0, \dots, A^{n-1}r_0), \quad (3.14)$$

where  $r_0 := b - Ax_0$  is the residual of the linear system with the initial guess  $x_0$ , the  $n$ th iterate  $x_n$  is an approximation to the exact solution  $x_*$  that satisfies  $x_n - x_0 \in K_n(A, r_0)$ .

A large number of Krylov subspace solvers have been developed so far: the generalized minimal residual (GMRES) method [86], the biconjugate gradient (BiCG) method [87], the biconjugate gradient stabilized method (BiCGStab) method [88], the quasi-minimal residual (QMR) method [89], and the transpose-free quasi-minimal residual (TFQMR) method [90], to name a few. BiCGStab is one of the most popular solvers since it has three advantages over other types of Krylov subspace methods.

- (1) BiCGStab is a three-term recurrence method; therefore it has relatively low memory requirement.
- (2) BiCGStab does not require a transposed matrix multiplied by a vector; it only requires matrix-vector multiplication alone.

(3) BiCGStab often exhibits more stable convergence than other solvers.

Thus, BiCGStab method is widely used in solving the linear equation arising from RTE.

The details of a left preconditioned BiCGStab for the system  $Ax = b$  are described in Algorithm 3.3 [88].

---

**Algorithm 3.3: BiCGStab method with left preconditioner  $P$**

---

1. Compute residual  $r_0 = P^{-1}(b - Ax_0)$  with the initial guess  $x_0$  and the left preconditioner  $P$ .
  2. Set  $p_0 = r_0$ .
  3. Choose an arbitrary vector  $\tilde{r}$  such that  $\rho_0 = \tilde{r}^T r_0 \neq 0$
  4. Given  $\epsilon > 0$ , for  $i = 0, 1, 2 \dots$  until  $\|r_i\|/\|b\| < \epsilon$ :
    - (1)  $s_i = P^{-1}Ap_i$ ,
    - (2)  $\gamma_i = \tilde{r}^T s_i$ ,
    - (3)  $\alpha_i = \gamma_i^{-1}\rho_i$ ,
    - (4)  $x_{i+1} = x_i + \alpha_i p_i$ ,
    - (5)  $r_{i+1} = r_i - \alpha_i s_i$ ,
    - (6)  $t_i = P^{-1}Ar_{i+1}$ ,
    - (7)  $\eta_i = (t_i, r_{i+1})/(t_i, t_i)$ ,
    - (8)  $x_{i+1} = x_i + \eta_i r_{i+1}$ ,
    - (9)  $r_{i+1} = r_{i+1} - \eta_i t_i$ ,
    - (10)  $p_{i+1} = p_i - \eta_i s_i$ ,
    - (11)  $\rho_{i+1} = \tilde{r}^T r_{i+1}$ ,
    - (12)  $\beta_i = \gamma_i^{-1}\rho_{i+1}/\eta_i$ ,
    - (13)  $p_{i+1} = r_{i+1} + \beta_i p_{i+1}$ .
- 

### 3.3.2. Block Krylov subspace methods and block BiCGStab algorithm

In DOT, one has to solve the linear system with multiple right-hand sides as (3.1). Under this type of problem setting, it is more efficient to solve the linear system of  $N_{\text{source}}$  right-hand sides given by Eq. (3.1) simultaneously with block Krylov subspace methods than treat each right-hand side separately. The block Krylov subspace method extends the definition of  $K_n(A, r_0)$  in Eq. (3.14): a similar affine space  $K_n^\blacksquare(A, R_0)$  that grows with iteration number  $n$  is defined as,

$$K_n^\blacksquare(A, R_0) := \text{span}(R_0, AR_0, \dots, A^{n-1}R_0), \quad (3.15)$$

where  $R_0 := [r_0^{(1)}, r_0^{(2)}, \dots, r_0^{(N_{\text{source}})}]$ . In each iteration, the block Krylov subspace methods minimize the residual within space  $K_n^\blacksquare(A, R_0)$ . It is obvious that when  $R_0$  contains the column vector  $r_0$ , then  $K_n(A, r_0) \subset K_n^\blacksquare(A, R_0)$ , therefore faster convergence rate can be expected from block Krylov subspace methods since the residual is minimized within the larger subspace  $K_n^\blacksquare(A, R_0)$  instead of  $K_n(A, r_0)$ . It is straightforward to generalize single right-hand side BiCGStab method (Algorithm 3.3) into a block version for the linear equation with multiple right-hand sides, the left preconditioned block BiCGStab algorithm can be summarized in Algorithm 3.4 [65]:

---

**Algorithm 3.4: block BiCGStab method with left preconditioner**

---

1. Compute residual  $R_0 = P^{-1}(B - A\Psi_0)$  with the initial guess  $\Psi_0$  and the left preconditioner  $P$ .
  2. Set  $P_0 = R_0$ .
  3. Choose an arbitrary  $N_{\text{source}} \times N$  matrix  $\tilde{R}$  such that  $\rho_0 = \tilde{R}R_0 \neq 0$ .
  4. Given  $\epsilon > 0$ , for  $i = 0, 1, 2 \dots$  until  $\|R_i(:, j)\|/\|B(:, j)\| < \epsilon$  for  $j = 1, \dots, N_{\text{source}}$ :
    - (1)  $S_i = P^{-1}AP_i$
    - (2)  $\gamma_i = \tilde{R}S_i$ ,
    - (3)  $\alpha_i = \gamma_i^{-1}\rho_i$ ,
    - (4)  $\Psi_{i+1} = \Psi_i + P_i\alpha_i$ ,
    - (5)  $R_{i+1} = R_i - S_i\alpha_i$ ,
    - (6)  $T_i = P^{-1}AR_{i+1}$ ,
    - (7)  $\eta_i = \langle T_i, R_{i+1} \rangle_F / \langle T_i, T_i \rangle_F$ ,
    - (8)  $\Psi_{i+1} = \Psi_i + \eta_i R_{i+1}$ ,
    - (9)  $R_{i+1} = R_{i+1} - \eta_i T_i$ ,
    - (10)  $P_{i+1} = P_i - \eta_i S_i$ ,
    - (11)  $\rho_{i+1} = \tilde{R}R_{i+1}$
    - (12)  $\beta_i = \gamma_i^{-1}\rho_{i+1}/\eta_i$ ,
    - (13)  $P_{i+1} = R_{i+1} + P_{i+1}\beta_i$ .
- 

In the left preconditioned block BiCGStab method, only four  $N \times N_{\text{source}}$  matrices  $S_k, R_k, T_k, P_k$  and one  $N_{\text{source}} \times N$  matrix  $\tilde{R}$  (excluding  $A, \Psi, B$  and the preconditioner  $P$ ) need to be stored in memory: therefore the total number of floating numbers to save is  $(5NN_{\text{source}} + O(N_{\text{source}}^2))$ , thus making the method memory-efficient. The left preconditioned block

BiCGStab method requires the evaluation of  $2N_{\text{source}}$  matrix-vector products,  $2N_{\text{source}}$  preconditioning procedures and a total of  $6NN_{\text{source}}^2 + 4NN_{\text{source}} + O(N_{\text{source}}^3)$  multiplication operations per iteration. For a large-scale system where  $N$  is not extremely large, the matrix-vector product and preconditioning procedure dominate the overall computation time: therefore theoretically floating-point operations (FLOPS) for each single right-hand side per iteration in Algorithm 3.4 is almost the same as it in Algorithm 3.3. For any fixed relative tolerance  $\epsilon$ , it is obvious that the stopping criteria of Algorithm 3.4 is stronger than the Algorithm 3.3's, which guarantees the Algorithm 3.4's solution is at least as accurate as the Algorithm 3.3's.

### 3.3.3. Multi-threading acceleration for block BiCGStab algorithm

In DOT, multi-threading techniques are often applied in solving forward problems with multiple right-hand sides to gain extra speed. Under limited resources case (the available threads number  $N_{\text{thread}} < N_{\text{source}}$ ), the most commonly used method is as follows: (1) distribute  $N_{\text{source}}$  sources equally (or almost equally) into  $N_{\text{thread}}$  threads, (2) solve a group of forward problems within each thread. This method requires very little effort to separate the entire work into multiple tasks and there is no communication between those parallel tasks. Therefore, it has a straightforward implementation and is not affected by parallel slowdown effects, due to the overhead from communication/synchronization. However, this method still treats every single source independently, so it does not take the advantage of the shared information among different sources. Therefore, the efficiency can be improved by replacing the sequential solver with the proposed block linear solver. Here two multi-threading methods generalized from the block BiCGStab algorithm are proposed.

In the first multi-threading method (see Algorithm 3.5) Algorithm 3.4 is directly parallelized with all the sources. In this algorithm, the main procedure remains the same while

the linear operations, including matrix-vector multiplications and preconditioning steps, are divided into multiple tasks. Therefore, this algorithm will converge with the same number of iterations as in Algorithm 3.4. However, due to synchronizations and communications between threads (see Step 5, 7:(4)(10)(17)), the algorithm may cause parallel slowdown effects with the increase of thread number.

The second multi-threading approach is summarized in Algorithm 3.6. This method requires no synchronization and communication in its main computation step (Step 2); therefore there is no parallel slowdown effect due to communication or synchronization. However, since right-hand sides are split into available threads and each thread handles only part of right-hand sides, the method may require more iterations to converge especially when  $N_{\text{source}}/N_{\text{thread}}$  is close to 1.

To achieve higher efficiency in Algorithm 3.6, optimization of sources distribution across available threads is considered. According to the definition of  $K_n^\blacksquare(A, R_0)$ , strong colinearity between  $K_n(A, r_0^{(k)})$  may introduce more numerical error in projection step (Step 4:(2) in Algorithm 3.4) and therefore lead to more iterations for convergence. In practice, grouping more distant sources together in the separation step can reduce this collinearity. Therefore Step 1 in Algorithm 3.6 is implemented by solving an optimization problem, which maximizes the minimum distance of every pair of sources within the same group. This optimization problem can be formulated as follows:

$$\begin{aligned}
& \max d \\
& \text{subject to} \\
& z_i^{(k)} \in \{0,1\}, 1 \leq i \leq N_{\text{source}}, 1 \leq k \leq N_{\text{thread}}. \\
& \sum_{k=1}^{N_{\text{thread}}} z_i^{(k)} = 1, 1 \leq i \leq N_{\text{source}}. \\
& 0 \leq \sum_{i=1}^{N_{\text{source}}} z_i^{(k)} - N_{\text{source}}/N_{\text{thread}} < 1. \\
& d \leq d_{ij} z_i^{(k)} z_j^{(k)}, 1 \leq i < j \leq N_{\text{source}}, 1 \leq k \leq N_{\text{thread}}.
\end{aligned} \tag{3.16}$$

where  $z_i^{(k)}$  is a Boolean variable that indicates whether the  $i$ th source belongs to the  $k$ th thread,  $\sum_{k=1}^{N_{\text{thread}}} z_i^{(k)} = 1$  for  $1 \leq i \leq N_{\text{source}}$  means every source must be distributed to one specific thread,  $0 \leq \sum_{i=1}^{N_{\text{source}}} z_i^{(k)} - N_{\text{source}}/N_{\text{thread}} < 1$  shows the source number difference between threads must be less equal than 1,  $d \leq d_{ij} z_i^{(k)} z_j^{(k)}$  for  $1 \leq i < j \leq N_{\text{source}}, 1 \leq k \leq N_{\text{thread}}$  means only distance between sources within the same thread is considered.

---

**Algorithm 3.5: multi-threading block BiCGStab with inter-thread communication (parallel block BiCGStab-ITC)**

---

1. Divide the right-hand side  $B$  and the initial guess  $\Psi_0$  equally (or almost equally) into  $N_{TH}$  thinner matrices:  

$$B^{(k)} = [B^{(1)}, \dots, B^{(N_{\text{thread}})}], \Psi_0 = [\Psi_0^{(1)}, \dots, \Psi_0^{(N_{\text{thread}})}].$$
  2. Compute residual  $R_0^{(k)} = P^{-1}(B^{(k)} - A\Psi_0^{(k)}), k = 1, \dots, N_{\text{thread}}$ .
  3. Set  $P_0^{(k)} = R_0^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
  4. Setup a  $N_{\text{source}} \times N$  matrix  $\tilde{R} = [R_0^{(1)}, \dots, R_0^{(N_{\text{thread}})}]^T$ .
  5. Threads synchronization.
  6. Compute  $\rho_0^{(k)} = \tilde{R}R_0^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
  7. Given  $\epsilon > 0$ , for  $i = 0, 1, 2 \dots$  until  $\|R_i(:, j)\| / \|B(:, j)\| < \epsilon$  for  $j = 1, \dots, N_{\text{source}}$ :
    - (1)  $S_i^{(k)} = P^{-1}AP_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (2)  $\gamma_i^{(k)} = \tilde{R}S_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (3) Update  $P_i, S_i$  and  $\gamma_i$ :  

$$P_i = [P_i^{(1)}, \dots, P_i^{(N_{\text{thread}})}], S_i = [S_i^{(1)}, \dots, S_i^{(N_{\text{thread}})}], \gamma_i = [\gamma_i^{(1)}, \dots, \gamma_i^{(N_{\text{thread}})}].$$
    - (4) Threads synchronization.
    - (5)  $\alpha_i^{(k)} = \gamma_i^{-1}\rho_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (6)  $\Psi_{i+1}^{(k)} = \Psi_i^{(k)} + P_i^{(k)}\alpha_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (7)  $R_{i+1}^{(k)} = R_i^{(k)} - S_i^{(k)}\alpha_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (8)  $T_i^{(k)} = P^{-1}AR_{i+1}^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (9) Compute  $c_i^{(k)} = \langle T_i^{(k)}, R_{i+1}^{(k)} \rangle_F, d_i^{(k)} = \langle T_i^{(k)}, T_i^{(k)} \rangle_F$ .
    - (10) Threads synchronization and compute  $\eta_i = \sum_{k=1}^{N_{\text{thread}}} c_i^{(k)} / \sum_{k=1}^{N_{\text{thread}}} d_i^{(k)}$ .
    - (11)  $\Psi_{i+1}^{(k)} = \Psi_i^{(k)} + \eta_i R_{i+1}^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (12)  $R_{i+1}^{(k)} = R_{i+1}^{(k)} - \eta_i T_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (13)  $P_{i+1}^{(k)} = P_i^{(k)} - \eta_i S_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (14)  $\rho_{i+1}^{(k)} = \tilde{R}R_{i+1}^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (15)  $\beta_i^{(k)} = \gamma_i^{-1}\rho_{i+1}^{(k)} / \eta_i, k = 1, \dots, N_{\text{thread}}$ .
    - (16)  $P_{i+1}^{(k)} = R_{i+1}^{(k)} + P_{i+1}^{(k)}\beta_i^{(k)}, k = 1, \dots, N_{\text{thread}}$ .
    - (17) Threads synchronization.
  8. Return  $\Psi = [\Psi_i^{(1)}, \dots, \Psi_i^{(N_{\text{thread}})}]$ .
- 

**Algorithm 3.6: parallel block BiCGStab with no inter-thread communication (parallel block BiCGStab-NITC)**

---

1. Divide column vectors of  $B$  and the initial guess  $\Psi_0$  into  $N_{\text{thread}}$  groups (See (3.16))  

$$B = [B^{(1)}, \dots, B^{(N_{\text{thread}})}], \Psi_0 = [\Psi_0^{(1)}, \dots, \Psi_0^{(N_{\text{thread}})}].$$
  2. Solve  $A\Psi^{(k)} = B^{(k)}, k = 1, \dots, N_{\text{thread}}$  with Algorithm 3.6 in parallel.
  3. Return  $\Psi = [\Psi^{(1)}, \dots, \Psi^{(N_{\text{thread}})}]$ .
-



### 3.4. Performance evaluation through numerical experiments

In this section we focus on the performance of these linear solvers in comparison to the traditional sequential BiCGStab method. To be consistent with the configuration of reconstruction in practice, the Delta-Eddington approximation [91] is used as the forward model with the absorption coefficients  $\mu_a$  and the reduced scattering coefficients  $\mu'_s$  as the optical properties. Note that all the forward solvers considered here solve the same linear equation that has been obtained with the same numerical scheme as used in [52], which ensures fair comparison between different linear solvers. We set the same relative tolerance  $\epsilon = 10^{-10}$  for all the linear solvers: in fact block solvers use equal or stronger stopping criteria than the traditional BiCGStab solver (see the difference between Algorithm 3.3(4) and Algorithm 3.4(4), Algorithm 3.5(7) for comparison). An ILUT(0.01, 5) [66] preconditioner was applied here for all methods. All numerical experiments were performed with dual Intel Xeon CPU X5650 and 32GB physical memory.

The performance of preconditioned block linear solver with respect to CPU time may be most affected by the following three factors: 1) the preconditioner  $P$  that is used to approximate the matrix  $A$  in linear equation (3.8); 2) the structure of the matrix  $A$  that is determined by the anisotropic factor, optical properties, and spatial and angular discretization; 3) the total right-hand side number and the number of threads used for computation. Therefore, we investigate here how these factors influence the performance of block BiCGStab methods compared to the traditional sequential BiCGStab method through extensive numerical experiments.

#### 3.4.1. Numerical phantoms

A 2D circular phantom and a 3D cylinder phantom are considered here. For both phantoms, the refractive index is set to 1.4, which is a typical value for biological tissue, and source modulation

frequency is set to 600 MHz. The optical properties considered will be given later in the following subsections according to each specific case being examined. As shown in Fig. 3.2, the 20 light sources for the 2D phantom and 32 light sources for the 3D phantom are located on the surface. The angular domain for both phantoms is discretized with  $S_8$  and for spatial domain the 2D phantom is discretized with a finite volume mesh (FVM) with 4117 control volumes (the element size is 0.03cm) and the 3D phantom with 17158 control volumes (the element size is 0.1cm).

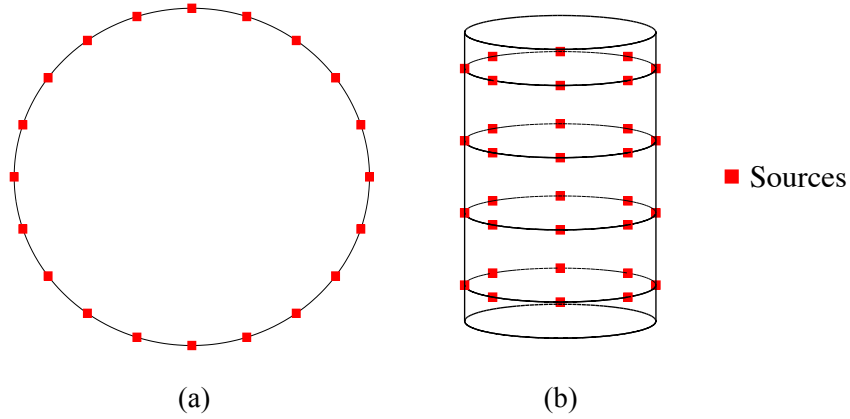


Figure 3.2: Source locations for Phantom 1 and 2; (a) 2D phantom: 20 sources are regularly distributed along the perimeter of the disk; (b) 3D phantom: 32 sources are distributed on the lateral surface of the cylinder with 4 height levels 0.375cm, 1.125cm, 1.875cm and 2.625cm. On each level, there are 8 sources distributed with even distance.

Table 3.2: The phantom settings of the 2D disk and 3D cylinder

	Phantom 1	Phantom 2
Dimension	2D	3D
Shape	Disk	Cylinder
Radius (cm)	1	1
Height (cm)	-	3

### 3.4.2. Performance evaluation of the novel preconditioner

In this subsection, we evaluate the performance of the novel preconditioner proposed in 3.2.2 on Phantom 1 with homogeneous optical properties. Multiple optical property combinations of  $\mu_a$

and  $\mu'_s$  (see Table 3.3) are tested. A 20 right-hand side linear system was solved through the traditional BiCGStab with no preconditioner ( $P_0$  in Table 3.3), traditional ILU preconditioner ( $P_1$  in Table 3.3), and the reduced linear system based ILU preconditioner ( $P_2$  in Table 3.3), respectively (In all the ILU preconditioners, the drop tolerance is set to 0.001 and the fill factor is set to 5 [66]). The total required iteration number  $n_{\text{iter}}$ , the total CPU time  $t_{\text{all}}$  and CPU time for ILU factorization  $t_f$  and preconditioning process  $t_p$  are reported in Table 3.4.

Table 3.3: Optical properties for Phantom 1 in performance evaluation

Case	Optical properties characteristics	$\mu_a[\text{cm}^{-1}]$	$\mu'_s[\text{cm}^{-1}]$
1	Normal biological tissue	0.1	10.0
2	High absorbing medium	0.5	10.0
3	Low absorbing medium	0.01	10.0
4	High scattering medium	0.1	20.0
5	Low scattering medium	0.1	1.0
6	High absorbing and low scattering medium	0.5	1.0
7	Void like medium	0.01	1.0

Table 3.4: Performance comparison between  $P_0$ ,  $P_1$  and  $P_2$  preconditioned BiCGStab algorithm

Case #	$t_f(P_0/P_1/P_2)$ [s]	$t_p(P_0/P_1/P_2)$ [s]	$t_{\text{all}}(P_0/P_1/P_2)$ [s]	$n_{\text{iter}}(P_0/P_1/P_2)$
1	- / 73 / <b>0.8</b>	- / 458 / <b>77</b>	585 / 614 / <b>177</b>	4281 / <b>510</b> / 676
2	- / 70 / <b>0.8</b>	- / 319 / <b>61</b>	404 / 450 / <b>138</b>	2957 / <b>360</b> / 506
3	- / 72 / <b>0.8</b>	- / 477 / <b>87</b>	617 / 634 / <b>196</b>	4519 / <b>545</b> / 731
4	- / 84 / <b>0.8</b>	- / 625 / <b>108</b>	579 / 806 / <b>248</b>	4245 / <b>620</b> / 951
5	- / 7 / <b>0.9</b>	- / 40 / <b>35</b>	NC / <b>75</b> / 79	NC / <b>180</b> / 278
6	- / 7 / <b>0.9</b>	- / 37 / <b>29</b>	734 / 69 / <b>66</b>	5384 / <b>159</b> / 233
7	- / 7 / <b>0.9</b>	- / 40 / <b>37</b>	NC / <b>77</b> / 82	NC / <b>180</b> / 290

NC: Not converge within 6000 iterations (300 iterations per right-hand side)

One can observe that both  $P_1$  and  $P_2$  can effectively reduce the total iteration number. However,  $P_2$  is usually more efficient in terms of CPU time. For case 1 to 4 (medium with normal or high scattering coefficients), BiCGStab preconditioned with  $P_2$  solved the forward problems about 3 times faster than the other two, while BiCGStab preconditioned with  $P_1$  is the

most time consuming solver (even slower than no preconditioning) due to its expensive factorization and preconditioning cost. For case 5 to 7 (medium with low scattering coefficients), The CPU times of BiCGStab preconditioned with  $P_1$  and  $P_2$  are very close, both are significantly less than  $P_0$ . Overall, the efficiency of the reduced linear system preconditioner  $P_2$  in all cases is optimal or very close to optimal. Therefore, it is preferred over the traditional ILU preconditioner due to its more stable performance.

### 3.4.3. Influence of the anisotropic factor on the performance

The linear operator  $A$  in the discretized RTE (3.8) is affected by the pattern  $P$  in  $A_2$  (see (3.11)), which is part of  $A$ , is generated by the integral of the Henyey-Greenstein phase function that depends on the anisotropic factor  $g$ , and therefore we investigate here the influence of the anisotropic factor  $g$  on the performance of the block BiCGStab algorithm. The tests are conducted on both 2D circular and 3D cylinder phantoms. The absorption coefficient for both phantoms is fixed to  $0.1\text{cm}^{-1}$  and multiple reduced scattering coefficients ( $\mu'_s [\text{cm}^{-1}] = 1.0, 5.0, 10.0$ ) are examined that lead to various combinations of different scattering coefficients and anisotropic factors. For each fixed  $\mu'_s$ , four different anisotropic factors were tested, including the isotropic scattering case ( $g = 0$ ) and three strong forward-peaking scattering cases ( $g = 0.8, 0.9, 0.95$ ), which corresponds to their respective scattering coefficient  $\mu'_s, 5\mu'_s, 10\mu'_s, 20\mu'_s$ . The full RTE (2.12) is used as the forward model for every combination of  $(g, \mu'_s)$  on each phantom, and the discretized linear equation with the Henyey-Greenstein phase function (2.7) is solved with both the traditional BiCGStab (Algorithm 3.3) and the block BiCGStab (Algorithm 3.4). The CPU times of both algorithms (CPU1 for BiCGStab, CPU2 for block BiCGStab) and the speedup factors (BiCGStab CPU time / block BiCGStab CPU time) are reported in Table 3.5.

Table 3.5: Performance comparison between traditional and block BiCGStab algorithms on various combinations of  $(g, \mu'_s)$

$\mu'_s[\text{cm}^{-1}]$	$g$	$\mu_s[\text{cm}^{-1}]$	2D circular phantom results (CPU1[s]/CPU2[s]/Speedup)			3D circular phantom results (CPU1[s]/CPU2[s]/Speedup)		
1.0	0	1.0	71.0 /	42.3 /	1.7	1037 /	571 /	1.8
1.0	0.8	5.0	91.9 /	49.7 /	1.8	1475 /	785 /	1.9
1.0	0.9	10.0	95.3 /	52.3 /	1.8	1553 /	877 /	1.8
1.0	0.95	20.0	87.4 /	52.1 /	1.7	1539 /	869 /	1.8
5.0	0	5.0	145.7 /	52.3 /	2.8	1410 /	575 /	2.5
5.0	0.8	25.0	239.9 /	83.2 /	2.9	2000 /	775 /	2.6
5.0	0.9	50.0	183.6 /	66.4 /	2.8	1564 /	623 /	2.5
5.0	0.95	100.0	140.4 /	51.7 /	2.7	1430 /	579 /	2.5
10.0	0	10.0	200.6 /	70.0 /	2.9	1931 /	752 /	2.6
10.0	0.8	50.0	322.6 /	108.7 /	3.0	3073 /	1163 /	2.6
10.0	0.9	100.0	267.9 /	93.3 /	2.9	2132 /	819 /	2.6
10.0	0.95	200.0	187.7 /	68.1 /	2.8	1363 /	555 /	2.5

Table 3.5 shows that with the fixed  $\mu'_s$  the speedup factor of the block BiCGStab is not sensitive to the anisotropic factor  $g$ , although different anisotropic factors lead to different CPU times for both BiCGStab and block BiCGStab. This observation indicates that the zeroth-order Delta-Eddington approximation can provide sufficiently accurate information about the performance of the block solver, giving good insight into what speedup factors can be achieved with the full RTE using the original scattering phase function. Therefore, the following sections will be focused on the speedup factors of the block BiCGStab algorithm evaluated with solving the RTE (3.13) based on zeroth-order Delta-Eddington approximation.

### 3.4.4. Influence of optical properties on the performance

#### 3.4.4.1. Tests on phantoms with homogeneous optical properties

In this section, we investigate the influence of the optical properties on the performance of the block BiCGStab algorithm with both the 2D and 3D phantoms with homogeneous optical properties. A total of 100 combinations are generated of absorption and reduced scattering coefficients by uniformly sampling 10 each from their respective range  $[0.01, 1][\text{cm}^{-1}]$  and

$[0.1, 10][\text{cm}^{-1}]$ , which is sufficient enough to characterize the performance of the two linear solvers being compared. These absorption and reduced scattering coefficient pairs are tested with the 2D and 3D phantoms. The overall CPU time and matrix-vector multiplication (MV) number per single right-hand side are reported for both block and traditional BiCGStab algorithms with respect to each pair of  $(\mu_a, \mu'_s)$ . Speedup factors based on CPU times and MV ratios (BiCGStab MV number / block BiCGStab MV number) with respect to  $(\mu_a, \mu'_s)$  are also reported.

In the 2D numerical experiments, the conventional (or standard) BiCGStab algorithm requires 16 to 74.7 MV operations per right-hand side and 50 to 228 seconds in total to solve the entire linear system with 20 sources, while block BiCGStab only needs 14 to 34 MV operations per right-hand side and 32 to 74 seconds in total to solve the same system with 20 sources. CPU times and MV operation numbers of the methods depend on optical properties as shown in Fig. 3.3(a)(b)(d)(e): it is observed that both algorithms take more MV operations and thus more computational time to converge when scattering coefficient increases and absorption coefficient decreases, and vice versa. However, the speedup factors of  $1.3 \sim 3.1$  and the ratios of MV number of  $1.0 \sim 2.2$  (see Fig. 3.3(c)(f)) show that block BiCGStab is less dependent on the optical properties: the speedup factor of 2.5 and the MV ratio of 1.8 are achieved with absorption coefficients ( $< 0.3\text{cm}^{-1}$ ) and scattering coefficients ( $> 3\text{cm}^{-1}$ ) and the speedup factor of 1.8 and the MV ratio of 1.3 are always guaranteed when scattering coefficient is greater than  $2.0\text{cm}^{-1}$ . Similar results are observed for the 3D numerical experiments (see Fig. 3.4). The BiCGStab algorithm requires 22 to 68.75 MV operations per right-hand side and a total of 713 to 2205 seconds of CPU time to solve the linear system with 32 right-hand sides, while block BiCGStab algorithm only takes 18 to 38 MV operations per right-hand side and a total of 404 to 918 seconds, which leads to the speedup factors of  $1.3 \sim 2.7$  and the MV ratios of  $1.0 \sim 2.0$ . With

scattering coefficients  $\geq 2.0\text{cm}^{-1}$ , a speedup factor of at least 1.8 and a MV ratio of at least 1.3 can be achieved.

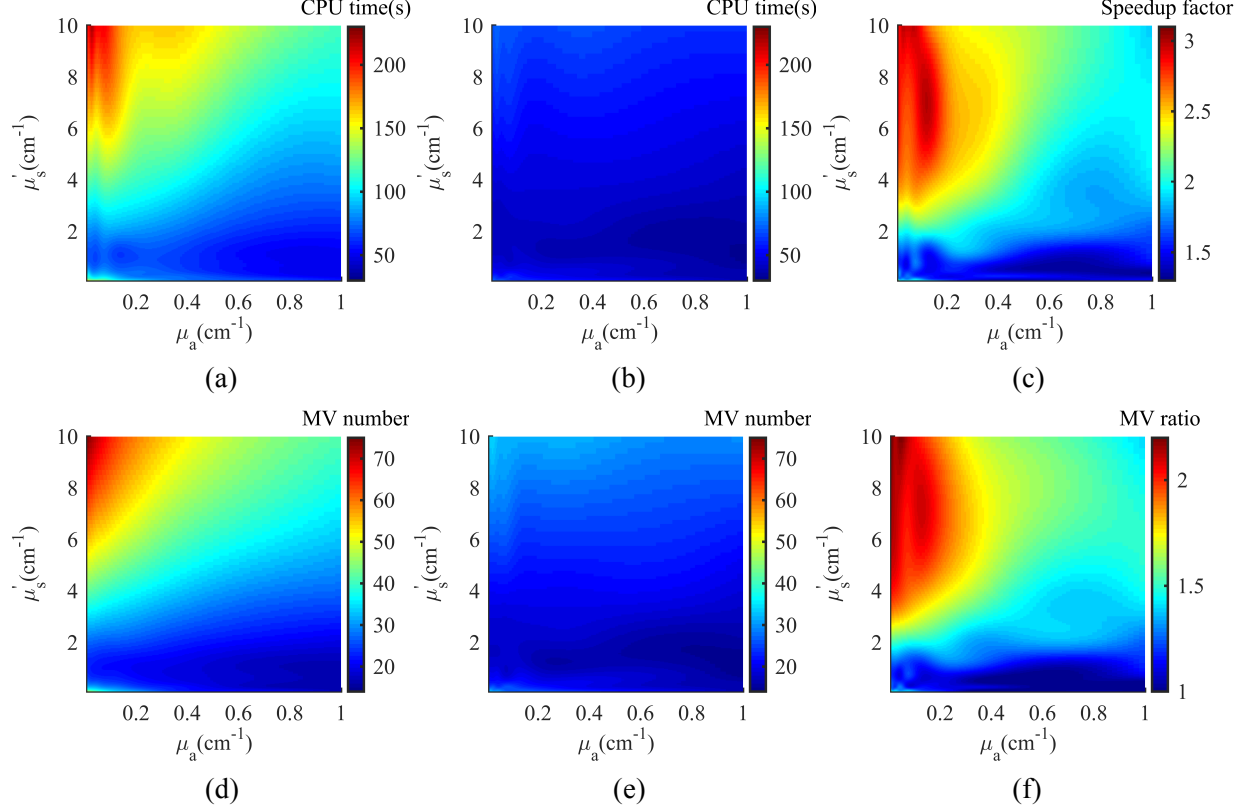


Figure 3.3: Results for 2D disk phantom with homogeneous optical properties; (a) CPU time (BiCGStab); (b) CPU time (block BiCGStab) (c) Speedup factor of block BiCGStab; (d) MV number (BiCGStab); (e) MV number (block BiCGStab); (f) MV ratio (BiCGStab MV number / block BiCGStab MV number)

We observe that the profile of speedup factor and the MV ratio are very similar in both 2D and 3D cases (See Fig. 3.3(c)(f) and Fig. 3.4(c)(f)). This strong correlation shows the higher efficiency of block BiCGStab, as compared to single right-hand side BiCGStab. In the block BiCGStab approach fewer MV operations are required since it searches for solutions in a larger subspace in every iteration. Furthermore, in this experiment, the speedup factor is usually 30% higher than the MV ratio, which is robust to the optical property. This extra efficiency can be explained by the better memory cache usage of matrix-matrix multiplication in block BiCGStab as compared to multiple matrix-vector multiplications for multiple sources in sequential

BiCGStab [92]. We can observe that the CPU time of the block BiCGStab is less dependent on optical property because it treats multiple right hand sides simultaneously and therefore the solution search through extended Krylov subspace rather than the matrix nature by optical properties is a dominant factor affecting the overall CPU time, whereas the sequential solver deals with multiple right-hand sides individually and thus the individual CPU time that highly depends on the optical properties determines the total CPU time.

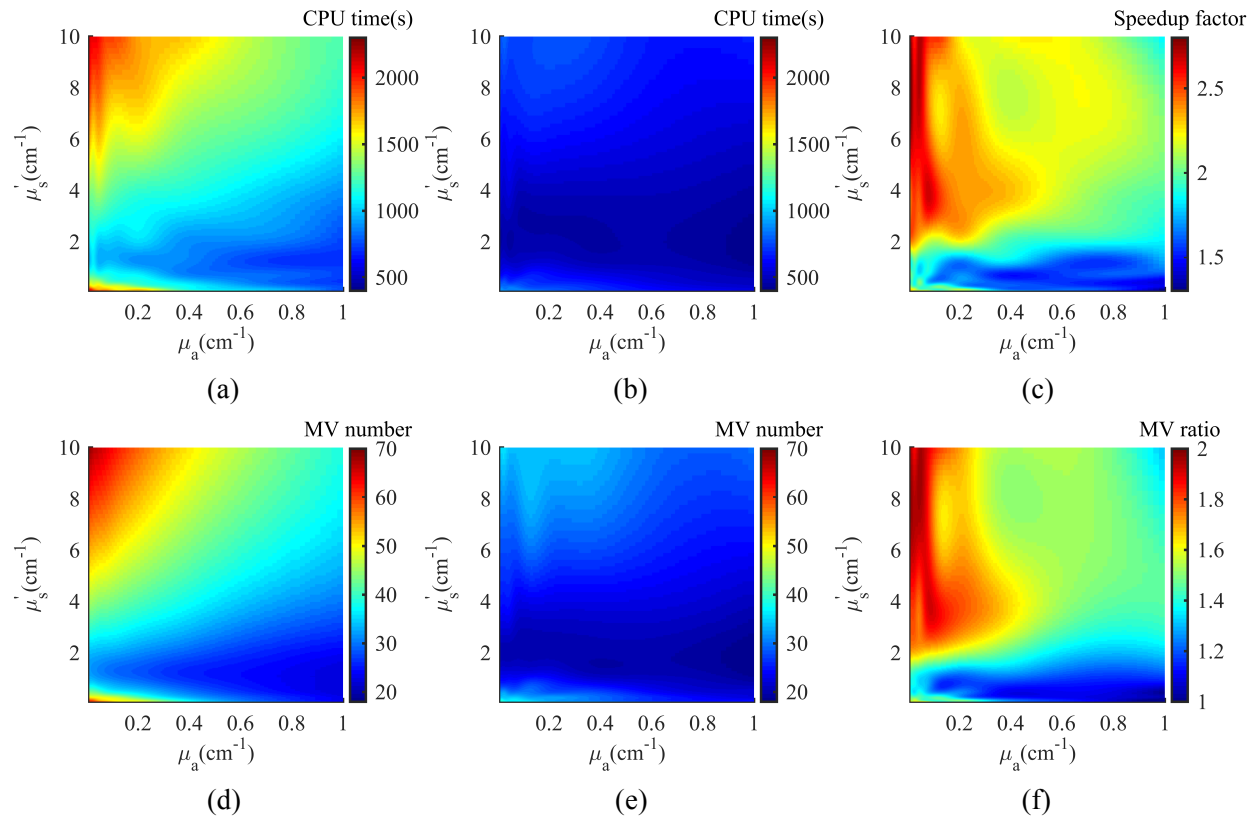


Figure 3.4: Results for 3D cylinder phantom with homogeneous optical properties; (a) CPU time (BiCGStab); (b) CPU time (block BiCGStab) (c) Speedup factor of block BiCGStab; (d) MV number (BiCGStab); (e) MV number (block BiCGStab); (f) MV ratio (BiCGStab MV number / block BiCGStab MV number).

### 3.4.4.2. Tests on phantoms with inhomogeneous optical properties

In this subsection we conduct numerical experiments on Phantom 2 with inhomogeneous optical properties. Two spherical objects are located with heights 1cm and 2cm, respectively, and their



radius is 0.3cm for both (see Fig. 3.5). The background optical properties are set to  $\mu_a = 0.1\text{cm}^{-1}$ ,  $\mu'_s = 10.0\text{cm}^{-1}$ .

With this phantom two cases were tested. In the first case the two objects are high absorbing and low scattering ( $\mu_a = 1.0\text{cm}^{-1}$ ,  $\mu'_s = 10.0\text{cm}^{-1}$ ); in the second case the phantom contains low-scattering and low-absorbing void like regions ( $\mu_a = 0.01\text{cm}^{-1}$ ,  $\mu'_s = 1.0\text{cm}^{-1}$ ). Thus in both cases the two objects are in the transport regime. The results are shown in Table 3.6, where CPU1 and CPU2 represent the computational time of the sequential and block BiCGStab algorithms respectively, MV1 and MV2 denote the average MV number per right hand side of the sequential and block BiCGStab algorithms respectively.

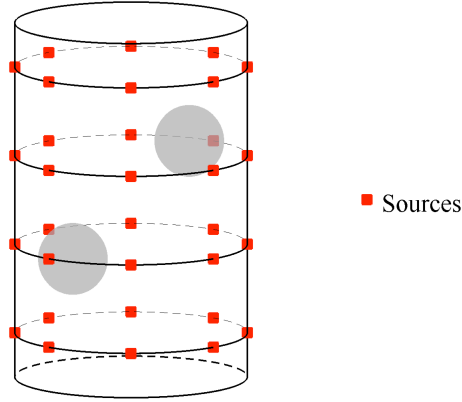


Figure 3.5: Locations of two spherical objects in Phantom 2

In the first case (high-absorbing and low-scattering objects), the MV ratio of 1.80 and speedup factor of 2.40 is achieved. In the second case (void-like inclusions), the MV ratio is 1.89 and the speedup factor is 2.53. In general, the block BiCGStab provides very similar MV ratios and speedup factors. These results are very similar to the ones from homogeneous optical property case as discussed earlier with  $\mu_a = 0.1\text{cm}^{-1}$ ,  $\mu'_s = 10.0\text{cm}^{-1}$  (See Row 3 in Table 3.6). This indicates that the MV ratio and the speedup factor depend mostly on the background optical properties, but little on the object optical properties.

Table 3.6: Results of tests on Phantom 2 with inhomogeneous optical properties

Object type	CPU1 (s)	CPU2 (s)	Speedup	MV1	MV2	MV ratio
High $\mu_a$ , low $\mu'_s$ region	1780	744	2.40	70.1	39	1.80
Void like region	1774	702	2.53	69.9	37	1.89
No objects	1820	716	2.54	70.6	37	1.91

### 3.4.5. Influence of spatial and angular discretization on the performance

When optical properties are fixed in (3.13), the structure of the matrix  $A$  in the linear equation is determined by both the spatial and angular discretization but in different ways. For example, for matrix  $A_3$  in (3.11) finer spatial discretization leads to more diagonal blocks as well as smaller coefficients for pattern  $W$  in every block but  $W$  itself is kept the same while different angular discretization leads to different pattern  $W$  but with total diagonal block number and  $W$ 's coefficients unchanged. So the spatial and angular discretization may potentially have different impact on the performance of linear solvers. Therefore in this subsection, the influence of spatial and angular discretization on the performance of block BiCGStab is examined separately. The 3D phantom with homogeneous optical properties is re-considered in the experiments: the absorption and reduced scattering coefficients are set to  $0.1\text{cm}^{-1}$  and  $10.0\text{cm}^{-1}$ . To test how discretization affects the speedup factor, a number of spatial meshes varying element sizes in the range  $0.08 \sim 0.13\text{cm}$  in the spatial domain together with different orders of discrete ordinates ( $S_4 \sim S_{12}$ ) in the angular domain have been created. Detailed information can be found in Table 3.7 and 3.8.

Table 3.7: Spatial meshes employed to study effects of spatial discretization

Element size [cm]	0.08	0.09	0.10	0.11	0.12	0.13
Node number	32940	23412	17158	12974	10021	7929

Table 3.8: Discrete ordinates employed to study the effects of angular discretization

Discrete Ordinates order	$S_4$	$S_6$	$S_8$	$S_{10}$	$S_{12}$
Solid angle number	24	48	80	120	168

We examine the effects of spatial and angular discretization separately: First we investigate spatial discretization effect using variable spatial meshes with a fixed angle set of  $S_8$ , and then we fix the spatial mesh at element size of 0.13cm and examine the influence of angular discretization. The CPU times, speedup factors and MV ratios are reported in Fig. 3.6.

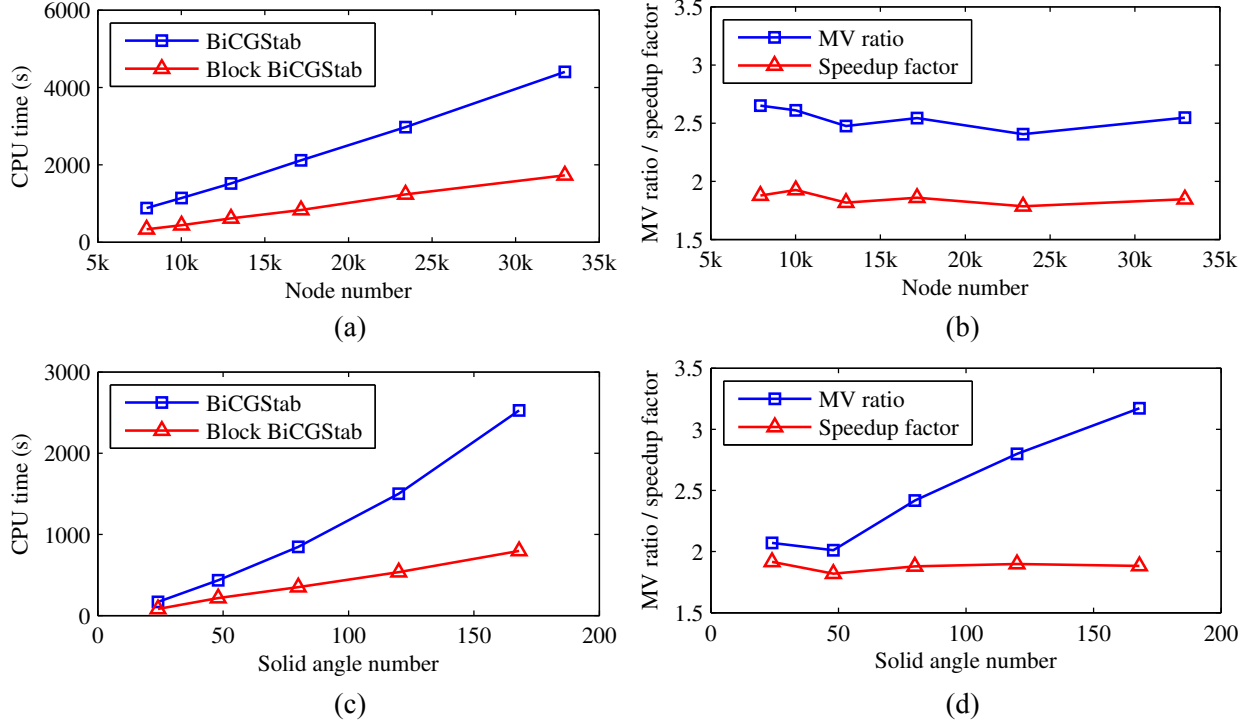


Figure 3.6: CPU times, speedup factors and MV ratios with respect to spatial and directional discretization; (a) CPU times comparison between BiCGStab and block BiCGStab on different spatial meshes with the fixed solid angle set  $S_8$ ; (b) speedup factor and MV number ratio on different spatial meshes with the fixed solid angle set  $S_8$ ; (c) CPU times comparison between BiCGStab and block BiCGStab on different orders of discrete ordinates ( $S_4 \sim S_{12}$ ) with the fixed spatial mesh; (d) speedup factor and MV number ratio on different orders of discrete ordinates ( $S_4 \sim S_{12}$ ).

As shown in Fig. 3.6(a), the CPU times of both algorithms increase linearly with the node number with the fixed solid angle set, which leads to almost constant speedup factors over the node numbers tested (Fig. 3.6(b)). However, the methods exhibit a different behavior in the CPU times when the solid angle set is altered with the node number fixed. Fig. 3.6(c)(d) show that with increasing solid angle numbers block BiCGStab has linearly increasing CPU times whereas for the traditional BiCGStab algorithm the CPU time increases superlinearly, which leads to a

linear increase in the speedup. We can also observe in Fig. 3.6(b)(d) that the MV ratio is not sensitive to the spatial discretization and the solid angle set, thus the result indicates that the block BiCGStab algorithm can benefit most the RTE solution that deals with a large number of solid angles since the extra efficiency of the better cache usage can provide will increase with large solid angle set.

### 3.4.6. Influence of the number of right-hand sides on the performance

In this subsection, we explore the influence of the number of sources on the performance of the block BiCGStab algorithm. To this end, the 2D phantom is considered here. The optical property is set as  $\mu_a = 0.1\text{cm}^{-1}$  and  $\mu'_s = 10.0\text{cm}^{-1}$ . The performance of the block BiCGStab algorithm was tested with various numbers of sources ( $N_s = 1 \sim 25$ ). The sources are uniformly distributed on the medium's surface. The average CPU time, speedup factor and MV ratio are recorded and reported in Fig. 3.7.

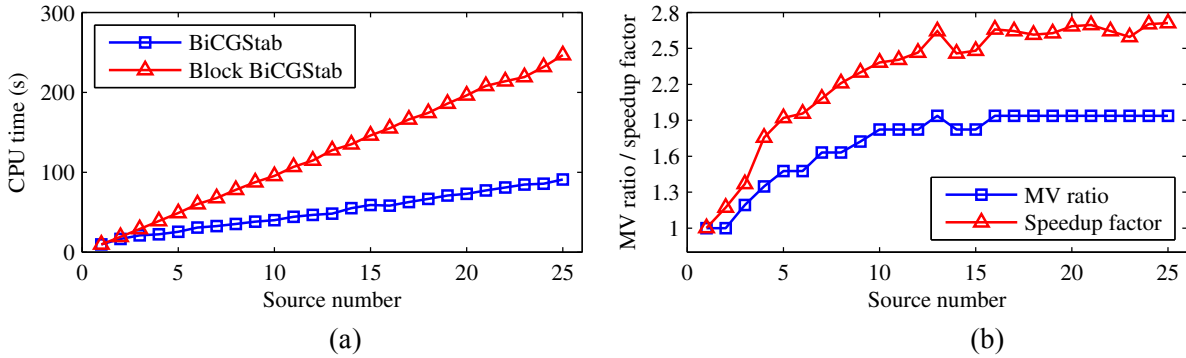


Figure 3.7: Performance of the block BiCGStab algorithm with different number of sources: (a) Overall CPU time required in the sequential and block BiCGStab algorithms with various source numbers. (b) The speedup factor and MV ratio of the block BiCGStab algorithm compared with the traditional BiCGStab algorithm with various source numbers.

As shown in Fig. 3.7(a), the CPU time grows approximately linearly for both solvers with respect to the source number when  $N_{\text{source}} \geq 10$  but with different rates. The slope of the block BiCGStab CPU time curve is significantly smaller, which means that the block BiCGStab treats

the increased right-hand side due to the increased number of sources computationally much more efficiently than the BiCGStab solver. The speedup factor of the block BiCGStab and the MV ratio between two solvers are shown in Fig. 3.7(b). We can find when  $N_{\text{source}} \leq 15$ , the MV ratio and speedup factor increase sublinearly with respect to source number. When  $N_{\text{source}} > 15$ , the MV ratio has stopped increasing and the speedup factor remains approximately constant, thus there being no additional speedup that can be obtained by increasing the source number. On the other hand, Fig. 3.7(b) shows that the speedup factor is always higher than the MV ratio one, which clearly indicates that the block solver obtains additional speedup from more efficient memory usage. This also explains why a considerable speedup factor (1.5 or higher) can also be achieved even with small source number ( $\sim 5$ ), which makes the block BiCGStab algorithm more attractive in practical applications.

### 3.4.7. Test on multi-threading block BiCGStab methods

To evaluate the performance under the multi-threading computation environment, two proposed parallel block BiCGStab methods (Algorithm 3.5 and 3.6) were tested with a 3D homogeneous cylinder phantom. We consider two cases with the optical properties: (1)  $\mu_a = 0.1\text{cm}^{-1}$ ,  $\mu'_s = 10.0\text{cm}^{-1}$  for normal tissue; (2)  $\mu_a = 0.5\text{cm}^{-1}$ ,  $\mu'_s = 1.0\text{cm}^{-1}$  for high absorbing and low scattering case. In both cases, the forward problems were solved with the single threading and multi-threading block solver with 1, 2, 4, 8, 16 threads, the performance from both algorithms is reported and compared with the traditional sequential solver.

Theoretically, with the increase of the thread number, the speedup factor of both Algorithm 3.5 and 3.6 as compared to the traditional sequential BiCGStab solver will decrease for different reasons. For Algorithm 3.5, this is mainly because more threads lead to more synchronization and communication cost, while for Algorithm 3.6 more threads will result in

fewer right-hand sides on each thread which in turn limits the search space and therefore the block solver will take MV operations to convergence. The preference of these two algorithms will also depend on the optical properties of the problem being solved.

For the normal tissue case, the performance of two parallel block solvers as compared to the parallelized traditional sequential solver and is shown in Fig. 3.8. The CPU time in seconds and the average MV number are shown in Fig. 3.8(a) and (c), and the speedup factor and MV ratio in Fig. 3.8(b) and (d). As expected, Fig. 3.8(c) shows that the average MV number remains constant with various thread numbers both for parallel sequential method and Algorithm 3.5 but it increases for Algorithm 3.6. Fig. 3.8(a)(b) shows the effect of the number of threads on the speedup factor of two parallel codes. As mentioned in the preceding paragraph, while the speedup factors of both parallel algorithms generally decrease with increasing threads, Algorithm 3.5 exhibits less deterioration in performance with increasing threads (especially with 2, 4, and 8 threads). That is because as the number of threads increases additional computational efforts due to increased MVs in Algorithm 3.6 dominates the parallel slowdown effect due to communication in Algorithm 3.5. With 16 threads, very little improvement is obtained for both of the two parallel solvers since the synchronization and communication cost in Algorithm 3.5 become as much dominant as extra MVs in Algorithm 3.6.

For high-absorbing and low-scattering media, the CPU time, speedup factor, average MV number and MV ratio with respect to thread number are shown in Fig. 3.9 for parallel sequential method and two parallel block BiCGStab algorithms. In this case, there is an insignificant difference in required MV number between traditional sequential solver and Algorithm 3.5 (see Fig. 3.9(c)). Therefore for Algorithm 3.5, the parallel slowdown by the synchronization and communication cost, become the most important factor. On the other hand, Algorithm 3.6 suffers

mush less from this effect; therefore, a considerable speedup factor can still be achieved by more efficient memory usage in the block solver. So in this case, Algorithm 3.6 gives better performance than Algorithm 3.5, with thread number less than or equal to 8, a speedup factor of  $\sim 1.5$  can be achieved.

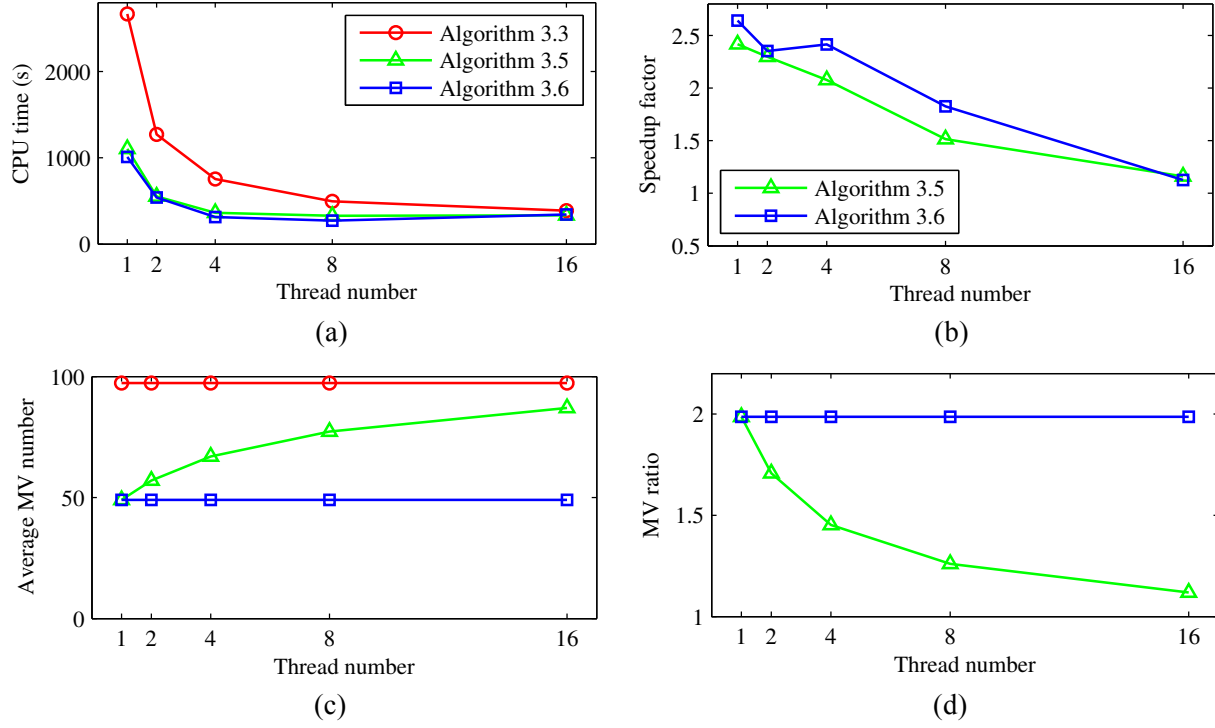


Figure 3.8: CPU time, speedup factor, MV number and MV ratio comparison respect to thread number in normal tissue case ( $\mu_a = 0.1\text{cm}^{-1}$ ,  $\mu'_s = 10.0\text{cm}^{-1}$ ): (a) CPU time comparison; (b) Speedup factor comparison; (c) Average MV number comparison; (d) MV ratio comparison.

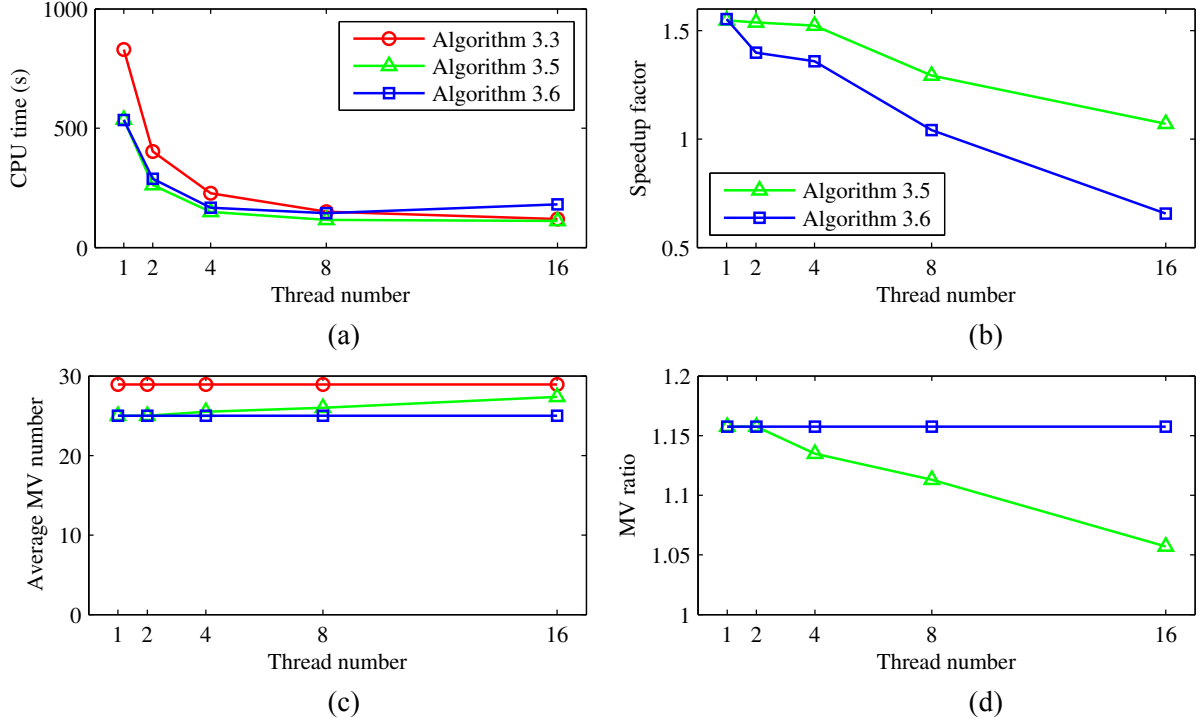


Figure 3.9: CPU time, speedup factor, MV number and MV ratio comparison respect to thread number in high absorbing and low scattering medium case ( $\mu_a = 0.5\text{cm}^{-1}$ ,  $\mu'_s = 1.0\text{cm}^{-1}$ ): (a) CPU time comparison; (b) Speedup factor comparison; (c) Average MV number comparison; (d) MV ratio comparison.

### 3.5. Discussion

In this chapter, a novel method has been presented for solving the forward problem in RTE-DOT. By exploiting the common optical properties shared among multiple right-hand sides, the new method solves the forward problems with multiple right-hand sides simultaneously within the framework of block Krylov subspace. In particular, I have focused on the block BiCGStab algorithm that searches an approximate solution in each iteration within the span of Krylov subspace generated with all right hand sides and therefore requires much fewer iterations than the traditional sequential algorithm. In addition, the results showed that more efficient cache memory usage in the block BiCGStab algorithm could lead to additional savings in the total computational time. Moreover, two multi-threading block BiCGStab algorithms are also proposed here for limited computation resource cases (thread number  $\leq$  sources number). The



performance of the proposed block solver has been evaluated in terms of MVs numbers and CPU times with comparison to the traditional sequential solver.

The numerical simulation shows that the block BiCGStab algorithm provides results 1.3 to 3.0 times faster than traditional one-hand side approaches depending on the number of right-hand sides, the number of threads, and on the optical properties. For normal-tissue optical properties (e.g.  $\mu_a = 0.1cm^{-1}$ ,  $\mu'_s = 10.0cm^{-1}$ ), the, serial block BiCGStab gave a speedup factor of 2.5 or higher and the parallel block BiCGStab solvers showed a speedup factor of up to 2.4 for over 20 sources. In this range of optical properties, the efficiency in the parallel block BiCGStab-ITC was less deteriorated with increasing threads than it in the parallel block BiCGStab-NITC. On the other hand, with high absorbing and low scattering medium (e.g.  $\mu_a = 0.5cm^{-1}$ ,  $\mu'_s = 1.0cm^{-1}$ ), a speedup factor of 1.5 or higher is obtained. In this case, however, the speedup mainly comes from the more efficient memory usage rather than the reduction on the required MV operation. Therefore the parallel block BiCGStab-NITC, which has no synchronization and communication cost, leads to a higher speedup factor ( $\sim 1.5$ ) than the parallel block BiCGStab-ITC. Additional tests show that the performance of the block BiCGStab solver is not strongly tied to a specific minimum number of sources to obtain a significant speedup factor, which means a considerable speedup (1.5 or higher) can still be obtained even for the limited source number ( $\sim 5$ ),

Overall, the forward problems with multiple sources in RTE-DOT can be solved much more efficiently without losing any accuracy with the proposed block BiCGStab solver. Moreover, the efficiency of the block solver can be further improved when using multiple threads (thread number  $<$  source number) by the proposed algorithms, depending on the optical properties. Since the forward solver is usually considered as the most time-consuming

component in the inverse solver, one can expect a significant speed on image reconstruction by applying the block BiCGStab as the forward solver, which makes RTE-DOT more practical in clinical applications.

## Chapter 4. Efficient Inverse Solvers in RTE-DOT

### 4.1. Introduction

RTE-DOT is an imaging modality that recovers the spatial distributions of optical properties and important physiological parameters inside the target medium. It does so by using light transmission and/or reflection measurements obtained from the surface and information of the source locations and boundary conditions. In this inverse problem the RTE is employed as a forward model for light propagation [93]. There is no closed-form inversion for general cases, and hence, the resulting inverse problem is typically formulated as the following optimization problem,

$$\begin{aligned} \min_{\mu, \Psi} \quad & \frac{1}{2} \|Q\Psi - M\|_F^2 + R(\mu) \\ \text{s. t.} \quad & A(\mu)\Psi = B \end{aligned} \tag{4.1}$$

where  $\mu$  represents the optical properties that are to be reconstructed;  $A(\mu)\Psi = B$  denote the discretized forward model for all sources and  $A(\mu)$  denotes the linear operator that depends on  $\mu$ ;  $\Psi$  represents the predicted light intensity distribution that leads to experimental measurements  $M$  for all sources;  $Q$  is the measurement operator that transforms the light intensity distribution  $\Psi$  into the detector readings;  $\|\cdot\|_F$  represents the Frobenius norm [94] and  $R(\mu)$  is the regularization term on  $\mu$ .

The optimization problem (4.1) is very challenging and the efficiency of the RTE-DOT highly depends on its numerical solver. Therefore it has attracted lots of researchers' attention in the past decade and many numerical methods have already been developed. These algorithms can be mainly classified into two categories: algorithms based on unconstrained optimization and algorithms based on constrained optimization. Algorithms of unconstrained optimization treat  $\Psi$

as a dependent variable of an independent variable  $\mu$  and eliminate the constraints  $A(\mu)\Psi = B$  in Eq. (4.1), which allows one to obtain the following unconstrained optimization problem,

$$\min_{\mu} \frac{1}{2} \|QA(\mu)^{-1}B - M\|_F^2 + R(\mu). \quad (4.2)$$

Then some classical unconstrained optimization techniques are employed for solving problem (4.2): the non-linear conjugate gradient method [95-99], the Gauss-Newton method [49, 100-102], the BFGS method [103-106] and the Levenberg-Marquardt method [50, 107, 108] to name a few. The main disadvantage of unconstrained algorithms is their computational inefficiency mainly caused by the requirement that the accurate solution to the forward problem  $\Psi = A(\mu)^{-1}B$  should be evaluated repeatedly, especially within a line search step. In RTE-DOT, with a given updated direction, a step size that satisfies certain criteria needs to be found and this process often requires multiple trials on various step sizes. However, for every potential step size, the objective function evaluation requires solving forward problems  $A(\mu)\Psi = B$  accurately, which is the most time-consuming part of an unconstrained optimization solver. To overcome this difficulty, the other type of algorithms, which is often referred to as PDE-constrained optimization and solves (4.1) directly with more sophisticated optimization techniques, has been extensively explored. These algorithms treat both  $\mu$  and  $\Psi$  as an independent variable, so the objective function evaluation in (4.1) becomes trivial with descent directions on both variables provided, thus not having to solve the forward problem repeatedly. Existing research in this type can be found in [48, 52, 56, 109, 110]. These constrained optimization algorithms are all based on the reduced-space methods (RSMs) [111]. RSMs replace the full Hessian system with the reduced Hessian system that is generated with Quasi-Newton methods and the linearized constraint equation. The reduced Hessian is then used to find the descent direction for both forward and inverse variables. However, compared to the full-space methods (FSMs), which

make use of the full Hessian information to update all the variables, RSMs often have a lower convergence rate and take more iterations to converge [111]. FSM-based optimization algorithms are widely used in many other areas [112-114], but have not been applied in the field of DOT so far.

In this dissertation, I explore the possibility to overcome the aforementioned limitations for existing constrained and unconstrained algorithms in two directions. For the unconstrained optimization formulation (4.2), a novel method is developed to determine the step size without multiple objective function evaluations. This method makes use of partial information of the second order derivative of the objective function and generates accurate step size that minimizes the objective function along the descent direction with low computational cost. Then a line-search-free BFGS inverse solver is constructed by combining this line-search-free step size scheme with the traditional BFGS method. The proposed algorithm requires no computational effort on line search process in each inverse iteration as well as maintains its simple structure as an unconstrained optimization solver. In practice, substantial speedup can be expected from the proposed line-search-free BFGS unconstrained method as compared to the traditional line search based BFGS unconstrained methods.

For the original constrained optimization formulation (4.1), the full-space algorithm for RTE-DOT has been developed for the first time. This algorithm utilizes the information of full Hessian for finding descent direction; therefore, it is able to provide higher quality descent direction as compared to the reduced-space algorithms. As in line-search-free BFGS unconstrained algorithm, I propose here a new efficient method to determine the step size for the full-space method. Due to higher quality descent direction by the proposed full-space method, one can expect the method to have a higher convergence rate with significantly fewer iterations.

The remainder of this chapter is organized as follows. The new step size estimation scheme and the line-search-free BFGS algorithm for unconstrained optimization formulation are described in Section 4.2. The full-space inverse solver for the constrained optimization formulation is then proposed in Section 4.3. Validation and performance evaluation are conducted in Section 4.4. This chapter concludes with a discussion in Section 4.5.

## 4.2. A line-search-free BFGS inverse solver for unconstrained optimization formulation

### 4.2.1. Newton's method and quasi-Newton methods

In this section I briefly introduce Newton's method and quasi-Newton methods. Consider a general unconstrained optimization problem

$$\min_{\mu} f(\mu), \quad (4.3)$$

where  $f(\mu)$  is a twice continuously differentiable objective function.

In Newton's method,  $f(\mu)$  is approximated with the second order Taylor expansion around the  $k$ th iterate,

$$f(\mu) \approx f(\mu_k) + g_k^T(\mu - \mu_k) + \frac{1}{2}(\mu - \mu_k)^T G_k(\mu - \mu_k), \quad (4.4)$$

where  $g_k$  and  $G_k$  are the gradient and the Hessian matrix of  $f$  at  $\mu_k$  respectively. Under the assumption that  $G_k$  is positive definite, the right-hand side of (4.4) is minimized as below to get the updating direction  $\Delta\mu_k := \mu - \mu_k$ ,

$$\min_{\Delta\mu_k} f(\mu_k) + g_k^T \Delta\mu_k + \frac{1}{2} \Delta\mu_k^T G_k \Delta\mu_k. \quad (4.5)$$

Thus the quadratic approximation (4.5) to the original objective function (4.4) is minimized to give an updating direction as below,

$$\Delta\mu_k = -G_k^{-1}g_k. \quad (4.6)$$

Then  $\mu_k$  is updated to  $\mu_{k+1}$  by  $\mu_{k+1} = \mu_k + \Delta\mu_k$ .

It has been proved that Newton's method has a locally quadratic convergence rate [111].

And it can be modified to achieve global convergence by performing a line search step and some regularization on the Hessian matrix when it is near singular [111].

However, Newton's method requires in each iteration an explicit Hessian matrix to achieve quadratic convergence. In many applications, including the unconstrained optimization formulation (4.2), the exact Hessian matrix is computationally very expensive to obtain. Moreover, the exact Hessian matrix sometimes is dense and not separable, which increases numerical inefficiency in solving (4.6) when the problem scale is large. Therefore, alternatives that only require the information of the objective function and its first order derivative are often considered in practice.

Quasi-Newton methods are believed as the most effective first order optimization technique for unconstrained optimization when they are applicable. Instead of computing the exact Hessian matrix in (4.6), they construct an approximate inverse Hessian (or Hessian) using current and previous objective function values and gradient information to reduce the

computational burden. On the other hand, the approximate inverse Hessian (or Hessian) can guarantee a super-linear convergence rate under certain conditions [111].

To understand the conditions that the approximate inverse Hessian (or Hessian) must satisfy, we take another derivative on the Taylor expansion (4.4) in the  $(k + 1)$ th iteration,

$$g(\mu) \approx g_{k+1} + G_{k+1}(\mu - \mu_{k+1}).$$

We let  $\mu = \mu_k$ ,  $s_k = \mu_{k+1} - \mu_k$  and  $y_k = g_{k+1} - g_k$ , then we have

$$G_{k+1}^{-1}y_k \approx s_k.$$

Based on this relation, we require the approximate inverse Hessian matrix  $H_{k+1}$  (or the approximate Hessian matrix  $B_{k+1}$ ) in quasi-Newton methods to satisfy the following condition,

$$H_{k+1}y_k = s_k \quad (B_{k+1}s_k = y_k). \quad (4.7)$$

Equation (4.7) is often referred to as the quasi-Newton condition. In every iteration in quasi-Newton methods, the existing approximate inverse Hessian (or approximate Hessian) is updated with low cost to satisfy this condition. A general quasi-Newton method is described as Algorithm 4.1.

In real world applications, Quasi-Newton methods are often preferred over Newton's method for solving unconstrained optimization problems due to the following advantages:

- (1) Quasi-Newton methods only require the first order derivatives, while Newton's method requires both the first order and second order derivatives.



- (2) The approximate inverse Hessian  $H_k$  (the approximate Hessian  $B_k$ ) in quasi-Newton methods is always positive definite, therefore the direction found in Step 3:(1) of Algorithm 4.1 is guaranteed to be a descent direction, while the exact Hessian matrix  $G_k$  can be indefinite for non-convex objective functions.
- (3) In quasi-Newton methods the descent direction can be efficiently computed, while in Newton's method, a linear equation is required to obtain the descent direction.

---

**Algorithm 4.1: General quasi-Newton methods**

---

1. Given the starting point  $\mu_0$ , relative convergence tolerance  $1 > \epsilon > 0$ , and initial inverse Hessian approximation  $H_0 \in R^{n \times n}$  (or  $B_0 \in R^{n \times n}$ ).
  2. Let  $k = 0$ .
  3. While  $\|g_k\| > \epsilon\|g_0\|$ :
    - (1) Compute the descent direction by  $\Delta\mu_k = -H_k g_k$  (or  $\Delta\mu_k = -B_k^{-1} g_k$ ),
    - (2) Find a proper step size  $\alpha_k$  along the  $\Delta\mu_k$  direction, then let  $\mu_{k+1} = \mu_k + \alpha\Delta\mu_k$ .
    - (3) Modify  $H_k$  (or  $B_k$ ) to get a  $H_{k+1}$  (or  $B_{k+1}$ ), which is positive definite and satisfies the quasi-Newton condition (4.7).
    - (4)  $k = k + 1$ .
- 

#### 4.2.2. A line-search-free BFGS inverse solver

The key component in quasi-Newton methods is the updating scheme (Step 3:(3) in Algorithm 4.1) for  $H_k$  (or  $B_k$ ). Numerous updating schemes have been proposed so far, including the DFP method [115], the BFGS method [116], the SR1 method [117], the Broyden's method [118], etc. The BFGS method, which is one of the most popular quasi-Newton methods, is used for the unconstrained optimization problem. The updating formula for  $H_k$  is given as below:

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (4.8)$$

To reduce the computational cost and memory requirement, the approximate inverse Hessian  $H_k$  in the BFGS algorithm does not have to be explicitly constructed, instead, two sequences of  $\{s_k\}$  and  $\{y_k\}$  with the initial guess  $H_0$  are memorized for the descent direction's computation (Step 2:(1) in Algorithm 4.2). A general description of the BFGS algorithm can be found in Algorithm 4.2 (outer loop) and its descent direction calculation is described in Algorithm 4.3 (inner loop).

---

**Algorithm 4.2: The BFGS algorithm for unconstrained optimization problem (4.3).  
(outer loop)**

---

1. Given the starting point  $\mu_0$ , relative convergence tolerance  $1 > \epsilon > 0$ , and initial inverse Hessian approximation  $H_0 \in R^{n \times n}$  (or  $B_0 \in R^{n \times n}$ ).
  2. While  $\|g_k\| > \epsilon \|g_0\|$ :
    - (1) Compute the descent direction by  $\Delta\mu_k$  according to Algorithm 4.3.
    - (2) Find a proper step size  $\alpha$  along the  $\Delta\mu_k$  direction, then let  $\mu_{k+1} = \mu_k + \alpha\Delta\mu_k$ .
    - (3) Compute and save  $s_k = \alpha\Delta\mu_k$  and  $y_k = g_{k+1} - g_k$ .
- 

---

**Algorithm 4.3: The BFGS algorithm for unconstrained optimization problem (4.3).  
(inner loop)**

---

Target: Compute  $\Delta\mu_k$  with  $g_k$  and  $\{s_i\}_{i=0}^{k-1}, \{y_i\}_{i=0}^{k-1}$ .

1. Let  $q = g_k$ .
  2. For  $i = (k-1): -1: 0$ 
    - (1)  $\alpha_i = (s_i^T y_i)^{-1} s_i^T q$ .
    - (2)  $q = q - \alpha_i y_i$ .
  3. Let  $\Delta\mu_k = H_0 q$ .
  4. For  $i = 0: (k-1)$ 
    - (1)  $\beta = (s_i^T y_i)^{-1} y_i^T \Delta\mu_k$ .
    - (2)  $\Delta\mu_k = \Delta\mu_k + (\alpha_i - \beta) s_i$ .
  5. Return  $\Delta\mu_k$ .
- 

To apply Algorithm 4.2 on (4.2), we need to derive the formulation of the gradient  $g$  with the objective function  $f(\mu)$  defined as  $\|Q\Psi - M\|_F^2/2 + R(\mu)$ , where  $\Psi$  is an implicit function of  $\mu$  given by

$$C(\mu, \Psi) = A(\mu)\Psi - B = 0. \quad (4.9)$$

According to the chain rule, we have

$$g = \nabla f = \nabla_\mu f + \nabla_\Psi f \nabla_\mu \Psi = \nabla_\mu R + Q^T(Q\Psi - M)\nabla_\mu \Psi, \quad (4.10)$$

where  $\nabla_\mu \Psi$  can be given by the implicit function algorithm,

$$\nabla_\mu \Psi = -(\nabla_\Psi C)^{-1} \nabla_\mu C = -A(\mu)^{-1} \left( \nabla_\mu A(\mu) \right). \quad (4.11)$$

By plugging (4.11) back into (4.10), we have the final expression of  $g$ , which is applied in Algorithm 4.2.

$$g = \nabla_\mu R - \left( A(\mu)^{-T} Q^T (Q\Psi - M) \right)^T \left( \nabla_\mu A(\mu) \right) \Psi \quad (4.12)$$

Another important part required in Algorithm 4.2 is the step size  $\alpha$  in Step 2:(2). Traditionally  $\alpha$  is usually determined with a backtracking line search method [119], which starts with a relative large estimate of  $\alpha$  and then shrinks it iteratively until certain stopping criteria are satisfied. Such stopping criteria include Armijo rule [120], Wolfe conditions [121, 122], Goldstein condition [123], etc. However, the backtracking line search method requires multiple function evaluations and/or gradient computations, which are computationally very expensive for the unconstrained optimization formulation in DOT. To overcome this difficulty, a step size estimation algorithm that is specifically designed for Eq. (4.2) is proposed. It can provide quite a good estimation of  $\alpha$  with lower computational cost as compared to the traditional line search-based step size searching.

In the  $k$ th iteration, we can derive the  $\Psi$ 's directional derivative along  $\Delta\mu_k$  with the implicit function theorem (4.11) as

$$\Delta\Psi_k := \nabla_{\Delta\mu_k}\Psi_k = \nabla_{\mu}\Psi_k \cdot \Delta\mu_k = -A(\mu_k)^{-1} \left( (\nabla_{\mu}A(\mu_k) \cdot \Delta\mu_k) \Psi_k \right). \quad (4.13)$$

For sufficient small  $\alpha$ , we have the following Taylor expansion to the first order,

$$\Psi(\mu_k + \alpha\Delta\mu_k) = \Psi_k + \alpha\Delta\Psi_k + o(\alpha\|\Delta\Psi_k\|) \approx \Psi_k + \alpha\Delta\Psi_k. \quad (4.14)$$

Therefore, the objective function  $f(\mu_k + \alpha\Delta\mu_k)$  can be approximated with

$$f(\mu_k + \alpha\Delta\mu_k) \approx h(\alpha) := \frac{1}{2} \|Q(\Psi_k + \alpha\Delta\Psi_k) - M\|_F^2 + R(\mu_k + \alpha\Delta\mu_k) \quad (4.15)$$

Then the step size  $\alpha$  can be determined by solving the following optimization problem,

$$\min_{\alpha} h(\alpha). \quad (4.16)$$

It should be noted that  $h(\alpha)$  is a function of a scalar variable  $\alpha$  alone and therefore a closed-form solution that minimizes (4.16) can be found easily, which reduces computation time significantly since no line search is required.

The line-search-free BFGS inverse solver for unconstrained optimization problem (4.2) is summarized in Algorithm 4.4.

---

**Algorithm 4.4: A line-search-free BFGS algorithm for unconstrained optimization problem (4.2)**

---

1. Given the starting point  $\mu_0$ , relative convergence tolerance  $1 > \epsilon > 0$ , maximum iteration number  $n_{iter}$  and initial inverse Hessian approximation  $H_0 \in R^{n \times n}$  (or  $B_0 \in R^{n \times n}$ ).
  2. Set  $k = 0$ .
  3. While  $k < n_{iter}$ :
    - (1) Compute  $\Psi_k = A(\mu_k)^{-1}B$ .
    - (2) Compute  $g_k$  according to (4.12).
    - (3) If  $\|g_k\| < \epsilon\|g_0\|$ , then break, else continue.
    - (4) Compute  $\Delta\mu_k$  according to Algorithm 4.3.
    - (5) Compute  $\Delta\Psi_k$  according to (4.13).
    - (6) Solve (4.16) to get the step size  $\alpha$ , then let  $\mu_{k+1} = \mu_k + \alpha\Delta\mu_k$ .
    - (7)  $k = k + 1$ .
  4. Output  $\mu_k$ .
-

In Algorithm 4.4, three forward solvers are called (Step 3:(1)(2)(5)) in each inverse iteration. They are the most time consuming steps and their efficiency can be significantly improved by the following three ways:

- (1) Using the (multi-thread) block BiCGStab solver proposed in Chapter 3 instead of traditional sequential solvers.
- (2) Setting a proper relative tolerance  $\epsilon_F$  for the forward solvers. Lower relative tolerance converges fast. On the other hand, the accuracy of the forward solver also depends on  $g_k$ ; in other words, the solution provided by the forward solver should be relatively more accurate than  $\|g_k\|$  such that an efficient decrease is allowed. Therefore, an adaptive forward relative tolerance  $\epsilon_{F,k}$  in the  $k$ th iteration defined as  $10^{-3} \min(1.0, \|g_k\|)$  is applied in this work, which turned out to be much more efficient than a uniform  $\epsilon_F$ .
- (3) Providing proper initial guess for some of the forward solvers. For Step 3:(1) in Algorithm 4.4, one can use  $\Psi_{k-1}$  (i.e., the previous iterate) as an initial guess to solve for the next iterate  $\Psi_k$ , which reduces the forward iterations in Step 3:(1).

The implementation details for computing the gradient  $g$  in (4.12) and solving (4.16) can be found in Appendix A.1.

### 4.3. A full-space algorithm for the constrained optimization formulation

#### 4.3.1. Full-space Newton's method for constrained optimization

The following more general constrained optimization problem is considered in this subsection,

$$\begin{aligned} \min f(\Psi, \mu) \\ \text{s. t. } C(\Psi, \mu) = 0 \end{aligned} \quad (4.17)$$

where  $f(\Psi, \mu) := \frac{1}{2} \|Q\Psi - M\|_F^2 + R(\mu)$  and  $C(\mu, \Psi) := A(\mu)\Psi - B$ ,  $\mu$  and  $\Psi$  are both variables to be optimized in (4.17). Since  $\Psi$  is an implicit function with respect to  $\mu$  given by  $C(\Psi, \mu) = 0$ , both  $\mu$  and  $\Psi$  are considered as the independent variable.

To derive the optimality conditions for (4.17), the Lagrangian function  $L$  is introduced as

$$L(\Psi, \mu; \lambda) := f(\Psi, \mu) + \lambda^T C(\Psi, \mu), \quad (4.18)$$

where  $\lambda$  is the Lagrangian multiplier (or the adjoint variable). Then the first order optimality conditions, also referred to as the Karush-Kuhn-Tucker (KKT) conditions [124], state that the gradient of  $L$  at a local minimum must vanish:

$$\partial L = \begin{Bmatrix} \partial_\Psi L \\ \partial_\mu L \\ \partial_\lambda L \end{Bmatrix} = \begin{Bmatrix} \partial_\Psi f + (\partial_\Psi C)^T \lambda \\ \partial_\mu f + (\partial_\mu C)^T \lambda \\ C \end{Bmatrix} = 0 \quad (4.19)$$

To simplify the notations, we use  $F_x$  and  $F_{xy}$  to denote  $\partial_x F$  and  $\partial_y(\partial_x F)$  hereafter. So the KKT conditions (4.19) is simplified as

$$\partial L = \begin{Bmatrix} L_\Psi \\ L_\mu \\ L_\lambda \end{Bmatrix} = \begin{Bmatrix} f_\Psi + C_\Psi^T \lambda \\ f_\mu + C_\mu^T \lambda \\ C \end{Bmatrix} = 0 \quad (4.20)$$

In the full-space algorithm, the KKT conditions (4.20) are treated as a system of nonlinear equations, which can be solved with Newton's method. A single Newton step is given by

$$\begin{bmatrix} L_{\Psi\Psi} & L_{\Psi\mu} & C_{\Psi}^T \\ L_{\mu\Psi} & L_{\mu\mu} & C_{\mu}^T \\ C_{\Psi} & C_{\mu} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\Psi \\ \Delta\mu \\ \Delta\lambda \end{Bmatrix} = - \begin{Bmatrix} f_{\Psi} + C_{\Psi}^T \lambda \\ f_{\mu} + C_{\mu}^T \lambda \\ c \end{Bmatrix}, \quad (4.21)$$

where the concatenated vector  $(\Delta\Psi, \Delta\mu, \Delta\lambda)$  is the descent direction for the concatenated variable  $(\Psi, \mu, \lambda)$  in the current iteration. To simplify the notation, a short form  $Kd = g$  is used to refer to (4.21), where  $d := (\Delta\Psi, \Delta\mu, \Delta\lambda)$  and  $g$  represents the right-hand side. Solving (4.21) with iterative linear solver in practice requires derivation of the right-hand side, the specific definition of the inner product of two vectors, and detailed expression for matrix-vector multiplication on the left hand side. These implementation details are provided in Appendix A.2.

The wide use of traditional Newton's method is often limited due to the numerical difficulties brought by the exact Hessian matrix. In another word, the exact Hessian matrix of the Lagrangian  $L$  may be very difficult to derive and also dense in many cases, which makes a linear solvers computationally very expensive. However, these limitations do not exist in the constrained optimization formulation of DOT. Firstly, the Hessian matrix  $K$  in (4.21) is a  $3 \times 3$  block matrix and each block can be simply derived. Moreover, the sparsity of the Hessian matrix  $K$  is guaranteed due to the special structure of the objective function and constraints in (4.9). Compared to the reduced Hessian methods, the full Hessian information is used in Newton's method. Therefore, a quadratic convergence rate instead of superlinear convergence rate can be expected. The Lagrange-Newton method for (4.9) is given in Algorithm 4.5.

---

**Algorithm 4.5: Lagrange-Newton method for (4.9)**

---

1. Choose initial guess  $\mu$ ,  $\Psi$  and  $\lambda$ .
  2. While NOT converge
    - (1) Construct the Lagrange function and its first order KKT conditions.
    - (2) Construct linear equation  $Kd = g$  (Eq. (4.21)).
    - (3) Solve  $Kd = g$  (Eq. (4.21)) to get  $d = (\Delta\Psi, \Delta\mu, \Delta\lambda)$ .
    - (4) Find an appropriate step size  $\alpha$  on the descent direction  $d$ .
    - (5) Update  $\mu$  by  $\mu = \mu + \alpha\Delta\mu$ .
    - (6) Update  $\Psi$  by  $\Psi = \Psi + \alpha\Delta\Psi$ .
    - (7) Update  $\lambda$  by  $\lambda = \lambda + \alpha\Delta\lambda$ .
- 

### 4.3.2. Preconditioned Krylov solver for the KKT system

The key step in Algorithm 4.5 is Step 2:(3), which solves the linear equation (4.21). Compared to linear problems in RSM-based methods, this one is much more challenging for several reasons. First, the linear system (4.21) is more than  $2N_{\text{source}}$  times larger than linear systems in the RSM-based methods. Furthermore, it is usually ill-posed and may not be positive definite on the constraint null space. Therefore, an appropriate preconditioner becomes mandatory.

Several preconditioning variants based on block decomposition for matrix  $K$  have already been proposed [112-114, 125]. In this dissertation, the preconditioner  $P$  for Eq. (4.21) is based on the most popular and robust one that was proposed in [125] with a slight modification,

$$P = \begin{bmatrix} L_{\Psi\Psi}C_{\Psi}^{-1} & 0 & I \\ L_{\mu\Psi}C_{\Psi}^{-1} & I & C_{\mu}^T C_{\Psi}^{-T} \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} C_{\Psi} & C_{\mu} & 0 \\ 0 & I & 0 \\ 0 & L_{\Psi\mu} - L_{\Psi\Psi}C_{\Psi}^{-1}C_{\mu} & C_{\Psi}^T \end{bmatrix} \quad (4.22)$$

According to (4.9),  $C_{\Psi} = A(\mu)$  and therefore  $C_{\Psi}^{-1}$  and  $C_{\Psi}^{-T}$  in (4.22) are well defined. With this preconditioner, the resulting preconditioned KKT matrix is given by

$$P^{-1}K = \begin{bmatrix} I & 0 & 0 \\ 0 & W & 0 \\ 0 & 0 & I \end{bmatrix}, \quad (4.23)$$

where  $W = C_{\mu}^T C_{\Psi}^{-T} L_{\Psi\Psi} C_{\Psi}^{-1} C_{\mu} - C_{\mu}^T C_{\Psi}^{-T} L_{\Psi\mu} - L_{\mu\Psi} C_{\Psi}^{-1} C_{\mu} + L_{\mu\mu}$ . We can see that  $P^{-1}K$  is an identity matrix except for the central block  $W$ , which has a relatively small size as compared



other blocks in RTE-DOT. Therefore a faster convergence rate can be expected in solving the following left preconditioned KKT system,

$$P^{-1}Kd = P^{-1}g \quad (4.24)$$

Krylov type linear solvers [112, 114] are often suggested to solve the linear equation (4.24). In this dissertation, the traditional BiCGStab algorithm (Algorithm 3.3), which has been introduced in Chapter 3, is reused here. So the preconditioning process within the matrix-vector multiplication needs to be provided, which is described in Algorithm 4.6.

---

**Algorithm 4.6: Preconditioning process with the preconditioner in (4.22)**

---

Target: With a right-hand side  $b = (b_\Psi, b_\mu, b_\lambda)^T$ , compute  $x = P^{-1}b$

1.  $t_1 = C_\Psi^{-1}b_\lambda$ .
  2.  $y_\lambda = b_\Psi - L_{\Psi\Psi}t_1$ .
  3.  $x_\mu = b_\mu - L_{\mu\Psi}t_1 - C_\mu^T C_\Psi^{-T}y_\lambda$ .
  4.  $t_2 = C_\Psi^{-1}C_\mu x_\mu$ .
  5.  $x_\Psi = t_1 - t_2$ .
  6.  $x_\lambda = C_\Psi^{-T}(L_{\Psi\mu}x_\mu - L_{\Psi\Psi}t_2)$ .
  7. Return  $x = (x_\Psi, x_\mu, x_\lambda)^T$ .
- 

As can be seen in Algorithm 4.6 (Step 1, 3, 4, 6), four forward linear systems have to be solved in the preconditioning process, which is computationally inefficient. So an approximated preconditioner  $\bar{P}$  is constructed here by

$$\bar{P} = \begin{bmatrix} L_{\Psi\Psi}\bar{C}_\Psi^{-1} & 0 & I \\ L_{\mu\Psi}\bar{C}_\Psi^{-1} & I & C_\mu^T\bar{C}_\Psi^{-T} \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{C}_\Psi & C_\mu & 0 \\ 0 & I & 0 \\ 0 & L_{\Psi\mu} - L_{\Psi\Psi}\bar{C}_\Psi^{-1}C_\mu & \bar{C}_\Psi^T \end{bmatrix}, \quad (4.25)$$

where  $\bar{C}_\Psi$  and  $\bar{C}_\Psi^T$  are the proposed ILU-based preconditioners in Chapter 3 of  $A$  and  $A^T$ , respectively. With this modification, computational efficiency is significantly boosted because no linear solver is required for the preconditioning process. On the other hand, the preconditioned linear system (4.24) is still very ill-posed, so the norm of the residual decreases

very slow or even gets stuck at certain stage of iterative process. Therefore a maximum iteration number is used here as the stopping criterion in order to avoid the case that the relative tolerance is never achieved.

### 4.3.3. Updating with the descent direction

After the descent direction  $d = (\Delta\Psi, \Delta\mu, \Delta\lambda)$  in (4.21) is obtained, traditional full-space methods often employ a line search procedure, with  $\|L(\Psi, \mu; \lambda)\|^2$  as the merit function, to find a proper step size  $\alpha$  (Algorithm 4.5, Step 2:(4)) to update the independent variables  $\Psi$  and  $\mu$ , and the adjoint variable  $\lambda$  all together (Algorithm 4.5, Step 2:(5)(6)(7)). However, in RTE-DOT, the traditional linear search and updating scheme may lead the full-space method to fail for several reasons. First, the accuracy of the solution to (4.24) is not guaranteed with the maximum iteration as the stopping criteria, and then with inaccurate updating direction,  $\|L(\Psi, \mu; \lambda)\|^2$  will converge very slowly due to its very large dimensionality. Furthermore, the inaccuracy in the previous inverse iteration will have negative impacts on all the following iterations. Moreover, because of the inconsistency between the merit function and the objective function,  $f(\Psi, \mu)$  is not guaranteed to decrease for every iteration, which leads to instability problems. These problems are also mentioned in some existing research [113].

Therefore, another novel updating scheme is proposed here for the RTE-DOT, which borrows ideas from both the line-search-free BFGS algorithm and RSM-based methods [52]. In this updating scheme,  $\Delta\mu$  is retained after  $d$  in (4.21) is solved. Then the same procedure as in the line-search-free BFGS algorithm in Section 4.2 is used to get  $\Delta\Psi$  according to (4.13) and a proper step size  $\alpha$  according to (4.16). Then the inverse variable  $\mu$  is updated according to Step 2:(5) in Algorithm 4.5. Finally, the forward variable  $\Psi$  and adjoint variable  $\lambda$  are updated by

vanishing the first and third KKT equations in (4.20), which is equivalent to solving the forward and adjoint equations in the RSM-based algorithms.

Two main advantages can be obtained by this updating scheme:

- (1) There is no stability issue. This new updating scheme inherits its stability property from the line-search-free BFGS method, and therefore the objective function is guaranteed to decrease in each inverse iteration.
- (2) Better initial guess for  $(\Psi, \mu, \lambda)$  for equation (4.20) is regenerated for each inverse iteration. After  $\mu$  is updated, the new  $\Psi$  and  $\lambda$  are obtained by solving the forward and adjoint equations in RSM-based method, and therefore in equation (4.20) the first and third equations are very close to zero, which tends to make the overall residual of equation (4.20) smaller as compared to updating  $\Psi$  and  $\lambda$ , simply with Step 2:(6)(7) in Algorithm 4.5.

On the other hand, the fast convergence property of the full-space algorithm is still retained since the updating direction  $\Delta\mu$  is given by solving Newton's linear system (4.21), which makes use of the exact Hessian information.

#### **4.3.4. Full-space algorithm for PDE-constrained optimization in RTE-DOT**

The full-space algorithm for PDE-constrained optimization in RTE-DOT is summarized in Algorithm 4.7. As in the line-search-free BFGS algorithm, the relative tolerance for forward solvers in the  $k$ th iteration in Algorithm 4.7 can be adaptively set to  $\epsilon_{F,k} = \epsilon_F \|g_k\|$  to gain extra efficiency in CPU time.

---

**Algorithm 4.7: Full-space based PDE-constrained optimization algorithm for (4.16)**

---

1. Given an initial guess  $\mu_0$ , the inverse and forward relative tolerance  $1 > \epsilon_I > 0$ ,  $1 > \epsilon_F > 0$  and maximum iteration numbers  $n_I$  and  $n_F$  for inverse and forward solvers.
  2.  $k = 0$ .
  3. Compute  $\Psi_k = C_{\Psi,k}^{-1}B$  by the block BiCGStab linear solver with  $\epsilon_F$  as the relative tolerance, also record the residual  $r_\Psi := C_{\Psi,k}\Psi_k - B$ .
  4. for  $k = 1:n_I$ 
    - (1) Compute  $\lambda_k = -C_{\Psi,k}^{-T}f_{\Psi,k}$  by the block BiCGStab linear solver with  $\epsilon_F$  as the relative tolerance, also record the residual  $r_\lambda := C_{\Psi,k}^T\lambda_k + f_{\Psi,k}$
    - (2) Compute all the block matrices for construction of  $K$ .
    - (3) Compute  $g_\mu = f_\mu + C_{\mu,k}^T\lambda_k$  and set  $g_k = (r_\lambda, g_\mu, r_\Psi)^T$ .
    - (4) If  $k = 0$ , set  $\epsilon = \epsilon_I\|g_k\|$ .
    - (5) If  $\|g_k\| < \epsilon$  then break, else continue.
    - (6) Solve  $\bar{P}^{-1}Kx = \bar{P}^{-1}g_k$  with the sequential BiCGStab linear solver with  $n_F$  as total iteration number to get  $x = (\sim, \Delta\mu, \sim)$ , where  $\bar{P}$  is given by (4.25)
    - (7) Compute  $\Delta\Psi$  according to (4.13).
    - (8) Compute a proper step size  $\alpha$  by solving the scalar optimization problem (4.16).
    - (9) Update  $\mu_{k+1} = \mu_k + \alpha\Delta\mu$  and compute  $\bar{\Psi}_{k+1} = \Psi_k + \alpha\Delta\Psi$
    - (10) Compute  $\Psi_{k+1} = C_{\Psi,k+1}^{-1}B$  by the block BiCGStab linear solver with  $\epsilon_F$  as the relative tolerance and  $\bar{\Psi}_{k+1}$  as an initial guess, also record the residual  $r_\Psi := C_{\Psi,k+1}\Psi_{k+1} - B$ .
  5. Return  $\mu_k$ .
- 

## 4.4. Validation and performance evaluation through numerical simulations

### 4.4.1. Numerical phantoms and experiment settings

For numerical simulations a 2D circular phantom and a 3D cylinder phantom are considered here. For the 2D phantom, 10 sources and 40 detectors are distributed with equal distance on its perimeter (see Fig. 4.1(a)). For the 3D phantom, 16 sources and 64 detectors are distributed into four rings corresponding to four different height levels (0.375cm, 1.125cm, 1.875cm, 2.625cm) and all the sources and detectors within the same ring are located with equal distance on a circle on its height level (see Fig. 4.1(b)). The refractive index of both phantoms is set to 1.4, which is typical for biological tissue, and the modulation frequency of light source is set to 600MHz. The background absorption coefficient and reduced scattering coefficient of both phantoms are set to  $0.1\text{cm}^{-1}$  and  $10.0\text{cm}^{-1}$  respectively. The 2D phantom has 3 circular objects inside it and the 3D

phantom has 2 ball shape objects inside it. The geometric details on both phantoms can be found in Table 4.1.

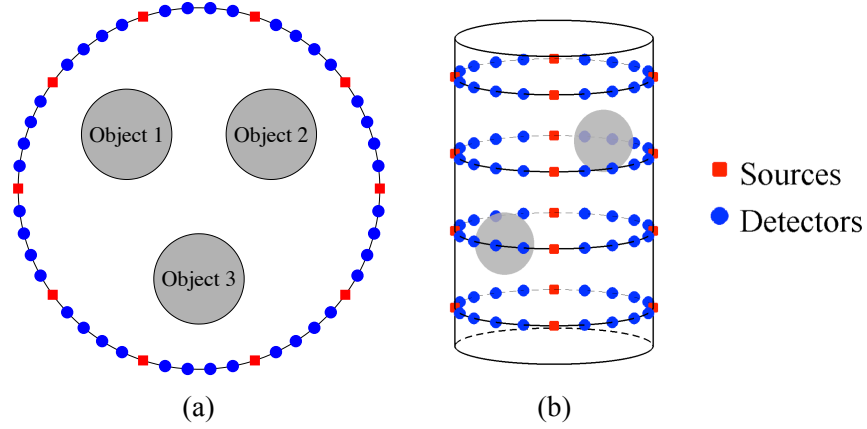


Figure 4.1: Source and detector locations for Phantom 1 and 2; (a) 2D phantom: 10 sources and 40 detectors are uniformly distributed along the perimeter of the disk; (b) 3D phantom: 16 sources and 64 detectors are distributed on the lateral surface of the cylinder with 4 height levels 0.375cm, 1.125cm, 1.875cm and 2.625cm. On each level, there are 4 sources and 16 detectors distributed with equal distance.

Table 4.1: The phantom settings of the 2D disk and 3D cylinder

	Phantom 1	Phantom 2
Dimension	2D	3D
Shape	Disk	Cylinder
Radius (cm)	1	1
Height (cm)	-	3
Source number	10	16
Detector number	40	48
Object 1 center	(-0.4, 0.3)	(-0.5, 0, 1)
Object 1 radius	0.25	0.3
Object 2 center	(0.4, 0.3)	(0.5, 0, 2)
Object 2 radius	0.25	0.3
Object 3 center	(0, -0.5)	-
Object 3 radius	0.25	-

To test the proposed inverse solvers, extensive numerical experiments are conducted with the 2D and 3D phantoms. In these experiments, 12 cases of different optical properties within the objects are considered: the phantoms in Case 1~5 has homogeneous reduced scattering coefficient but objects with different absorption coefficient from the background value; the

phantoms in Case 6~10 has homogeneous absorption coefficient but objects with different reduced scattering coefficient from the background value; in Case 11 and Case 12 the phantoms has inhomogeneity in both absorption and reduced scattering coefficients. The optical properties for all cases are shown in Table 4.2.

Table 4.2: Test cases for absorption coefficients reconstruction

Case	Dimension	# of objects	$\mu_{a,1}/\mu_{a,2}/\mu_{a,3}$ (cm <sup>-1</sup> )	$\mu'_{s,1}/\mu'_{s,2}/\mu'_{s,3}$ (cm <sup>-1</sup> )
1	2D	1	0.20 / ---- / ----	10.0 / ---- / ----
2	2D	2	0.20 / ---- / 0.05	10.0 / ---- / 10.0
3	2D	3	0.20 / 0.15 / 0.05	10.0 / 10.0 / 10.0
4	3D	1	0.20 / ---- / ----	10.0 / ---- / ----
5	3D	2	0.20 / 0.05 / ----	10.0 / 10.0 / ----
6	2D	1	0.10 / ---- / ----	20.0 / ---- / ----
7	2D	2	0.10 / ---- / 0.10	20.0 / ---- / 5.0
8	2D	3	0.10 / 0.10 / 0.10	20.0 / 15.0 / 5.0
9	3D	1	0.10 / ---- / ----	20.0 / ---- / ----
10	3D	2	0.10 / 0.10 / ----	20.0 / 5.0 / ----
11	2D	3	0.30 / 0.30 / 0.10	10.0 / 10.0 / 15.0
12	3D	2	0.30 / 0.10 / ----	10.0 / 5.0 / ----

In all the numerical experiments, a forward problem is solved with very accurate stopping criterion  $\epsilon = 10^{-12}$  to generate pseudo measurements. Then reconstructions are run with the pseudo measurements through the line-search-free BFGS algorithm or the full-space algorithm. Finally the results from the proposed inverse solver are compared with the results from the traditional BFGS algorithm. The RTE is discretized with the upwind finite volume discrete ordinates discretization as introduced in Section 3.2.1. The angular domain of both phantoms is discretized with  $S_8$  and for spatial domain the 2D phantom is discretized with a FVM mesh with 4117 control volumes (the element size is 0.03cm) and the 3D phantom is discretized with a FVM mesh with 10021 control volumes (the element size is 0.12cm). All numerical experiments were conducted on a PC with dual Intel Xeon CPU X5650 and 32GB physical memory.

#### 4.4.2. Performance evaluation with numerical tissue models

The correlation factor  $c(\mu, \mu^*)$  and the deviation factor  $d(\mu, \mu^*)$  are used to quantify the quality of reconstruction, where  $\mu$  and  $\mu^*$  are the reconstructed optical property and the ground truth optical property respectively. The definition of  $c(\mu, \mu^*)$  and  $d(\mu, \mu^*)$  is given as

$$\begin{cases} c(\mu, \mu^*) := \frac{\sum_{i=1}^{N_{cv}} V_i (\mu_i - \bar{\mu})(\mu_i^* - \bar{\mu}^*)}{(\sum_{i=1}^{N_{cv}} V_i) \sigma(\mu) \sigma(\mu^*)} \\ d(\mu, \mu^*) := \frac{\sqrt{(\sum_{i=1}^{N_{cv}} V_i (\mu_i - \bar{\mu}^*)^2) / (\sum_{i=1}^{N_{cv}} V_i)}}{\sigma(\mu^*)} \end{cases} \quad (4.26)$$

where  $V_i$  is the volume of the  $i$ th control volume,  $\bar{\mu}$  and  $\sigma(\mu)$  are the weighted average and standard deviation, which are defined as

$$\bar{\mu} := \left( \sum_{i=1}^{N_{cv}} V_i \mu_i \right) / \left( \sum_{i=1}^{N_{cv}} V_i \right), \quad \sigma(\mu) := \sqrt{\left( \sum_{i=1}^{N_{cv}} V_i (\mu_i - \bar{\mu})^2 \right) / \left( \sum_{i=1}^{N_{cv}} V_i \right)}$$

The correlation factor  $c(\mu, \mu^*)$  characterizes the correlation between the reconstructed and the benchmark optical property, while the deviation factor  $d(\mu, \mu^*)$  describe the absolute discrepancy between them. The ranges of  $c(\mu, \mu^*)$  and  $d(\mu, \mu^*)$  are  $[-1, 1]$  and  $[0, +\infty)$ , larger  $c(\mu, \mu^*)$  and smaller  $d(\mu, \mu^*)$  indicate higher quality of the reconstruction result.

#### 4.4.3. Reconstruction of absorption coefficient

In this subsection, the reconstruction is conducted only on  $\mu_a$  using phantoms (Case 1~5) with homogeneous reduced scattering coefficients.

First, the 2D phantom (Case 3) is considered with one low absorbing object and two high absorbing objects. The absorption coefficients for these three objects are  $0.2\text{cm}^{-1}$ ,  $0.15\text{cm}^{-1}$  and  $0.05\text{cm}^{-1}$  respectively. The absorption coefficient distribution is reconstructed with noise

free measurements with the traditional BFGS, the line-search-free BFGS, and the full-space algorithm. To make the overall CPU time comparable, I run the line-search-free BFGS and the traditional BFGS algorithms for 30 iterations and the full-space algorithm for 15 iterations. The reconstructed distributions of absorption coefficients are shown in Fig. 4.2 for each case. The results show that the line-search-free BFGS gives better quality than the other two, which yield similar results.

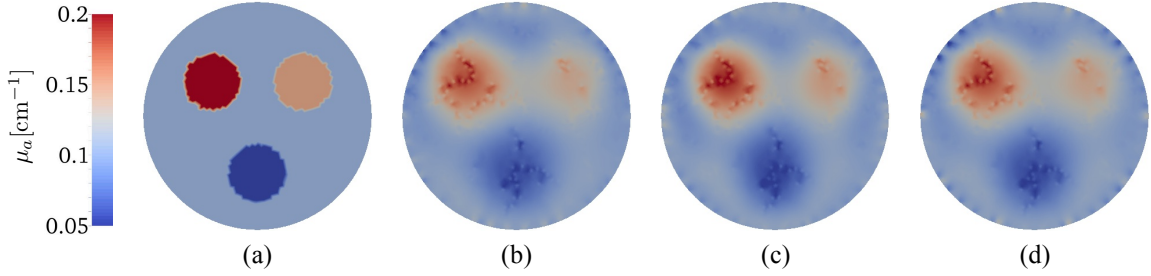


Figure 4.2: Absorption coefficient distribution in the two-dimensional phantom. (a) Exact distribution; (b) Traditional BFGS; (c) Line-search-free BFGS; (d) Full-space algorithm

To quantify the quality of the reconstructions, the correlation factor  $c(\mu_a, \mu_a^*)$  and deviation factor  $d(\mu_a, \mu_a^*)$  are reported in Table 4.3 for the three algorithms, where  $\mu_a^*$  represents the exact absorption distribution. As shown in Table 4.3, the line-search-free BFGS algorithm has highest  $c(\mu_a, \mu_a^*)$  and lowest  $d(\mu_a, \mu_a^*)$ , which indicates the highest reconstruction accuracy. The accuracy of the results from the traditional BFGS algorithm and the full-space algorithm is about the same.

Table 4.3: Correlation factors and deviation factors of  $\mu_a$  reconstruction results from the traditional and line-search-free BFGS algorithms on Case 3.

Algorithm	$c(\mu_a, \mu_a^*)$	$d(\mu_a, \mu_a^*)$
Traditional BFGS	0.80	0.60
Line-search-free BFGS	<b>0.85</b>	<b>0.53</b>
Full-space algorithm	0.80	0.60

The convergence behavior of the traditional BFGS, the line-search-free BFGS and the full-space algorithm for this case are shown in Fig. 4.3, where the relative change in the



objective function value is plotted over iterations and CPU time, as well as the required iteration number and CPU time taken to decrease the relative objective function to certain tolerance levels. BFGS1 and BFGS2 are used to represent the traditional BFGS and the line-search-free BFGS respectively. It can be observed that the line-search-free BFGS algorithm is more efficient than the traditional BFGS algorithm both in terms of iteration number and CPU time. The full-space algorithm shows fastest convergence with respect to the iteration number and its objective value drops significantly faster than the other two algorithms at the early stage of iterations. However, in terms of CPU time, the full-space method converges about at the same rate as the traditional BFGS algorithm since each inverse iteration in the full-space algorithm is computationally more expensive than in the BFGS type algorithms. In other words, to reduce the relative objective function value to  $10^{-3}$ , the full-space requires about 65% fewer iterations than the traditional BFGS algorithm and 45% fewer iterations than the line-search-free BFGS algorithm according to Fig. 4.3(c), while the line-search-free BFGS algorithm requires about 40% less CPU time than the other two algorithms according to Fig. 4.3(d).

The next example for absorption coefficient reconstruction is Case 5, in which  $\mu_a$  is reconstructed for a 3D cylinder phantom with high absorbing object and low absorbing object. The absorption coefficients for the two ball-shape objects are  $0.20\text{cm}^{-1}$  and  $0.05\text{cm}^{-1}$  respectively. As before, the reconstruction is conducted with noise free measurement. To make the overall CPU time comparable, the line-search-free BFGS and the traditional BFGS algorithms are run for 50 iterations and the full-space algorithm for 20 iterations. The cross sections of the reconstructed absorption distribution are shown in Fig. 4.4. It can be seen from the results that the line-search-free BFGS algorithm produces the highest resolution around the objects as well as the best contrast between the objects and the background. The traditional

BFGS algorithm generates the second best reconstruction result. The correlation and deviation factors are reported in Table 4.4 for the reconstruction results from all three algorithms.

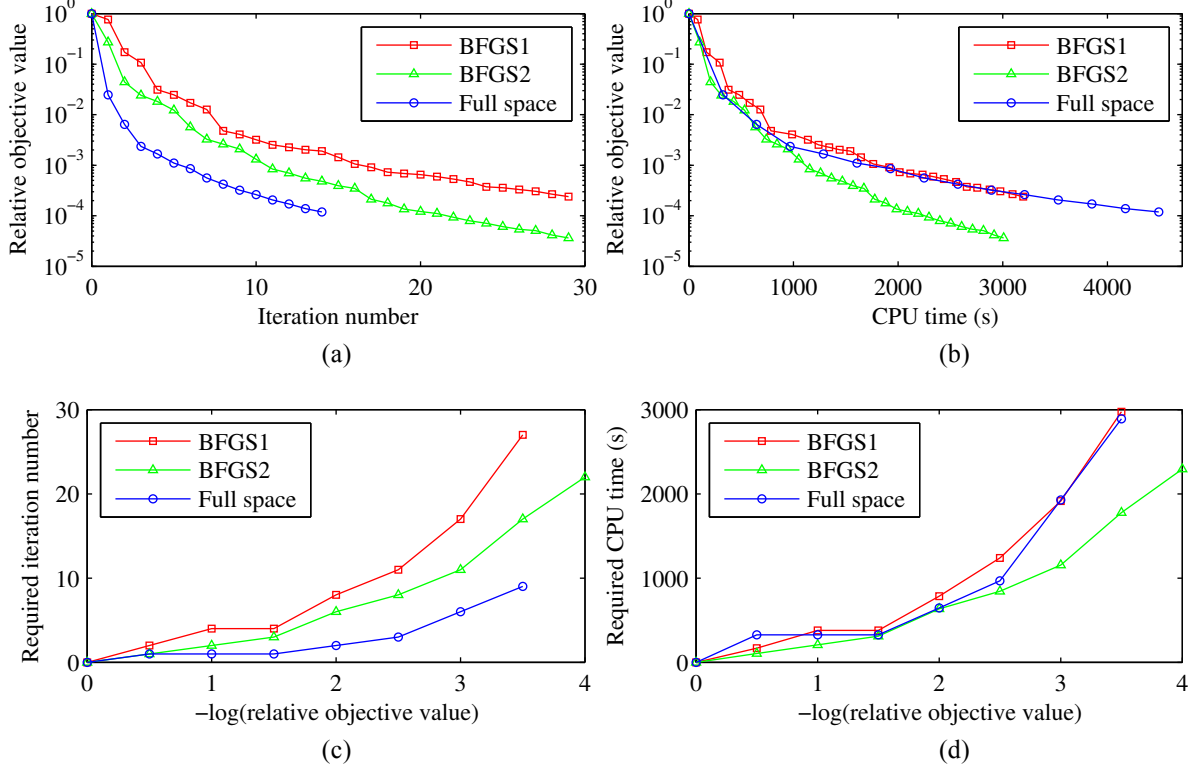


Figure 4.3: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a, b) Relative function value over iteration number and CPU time; (c, d) Required iteration number and CPU time to converge the objective function value to certain levels.

Table 4.4: Correlation factors and deviation factors of  $\mu_a$  reconstruction results from the traditional and line-search-free BFGS algorithms on Case 5.

Algorithm	$c(\mu_a, \mu_a^*)$	$d(\mu_a, \mu_a^*)$
Traditional BFGS	0.74	0.67
Line-search-free BFGS	<b>0.79</b>	<b>0.62</b>
Full-space algorithm	0.69	0.73

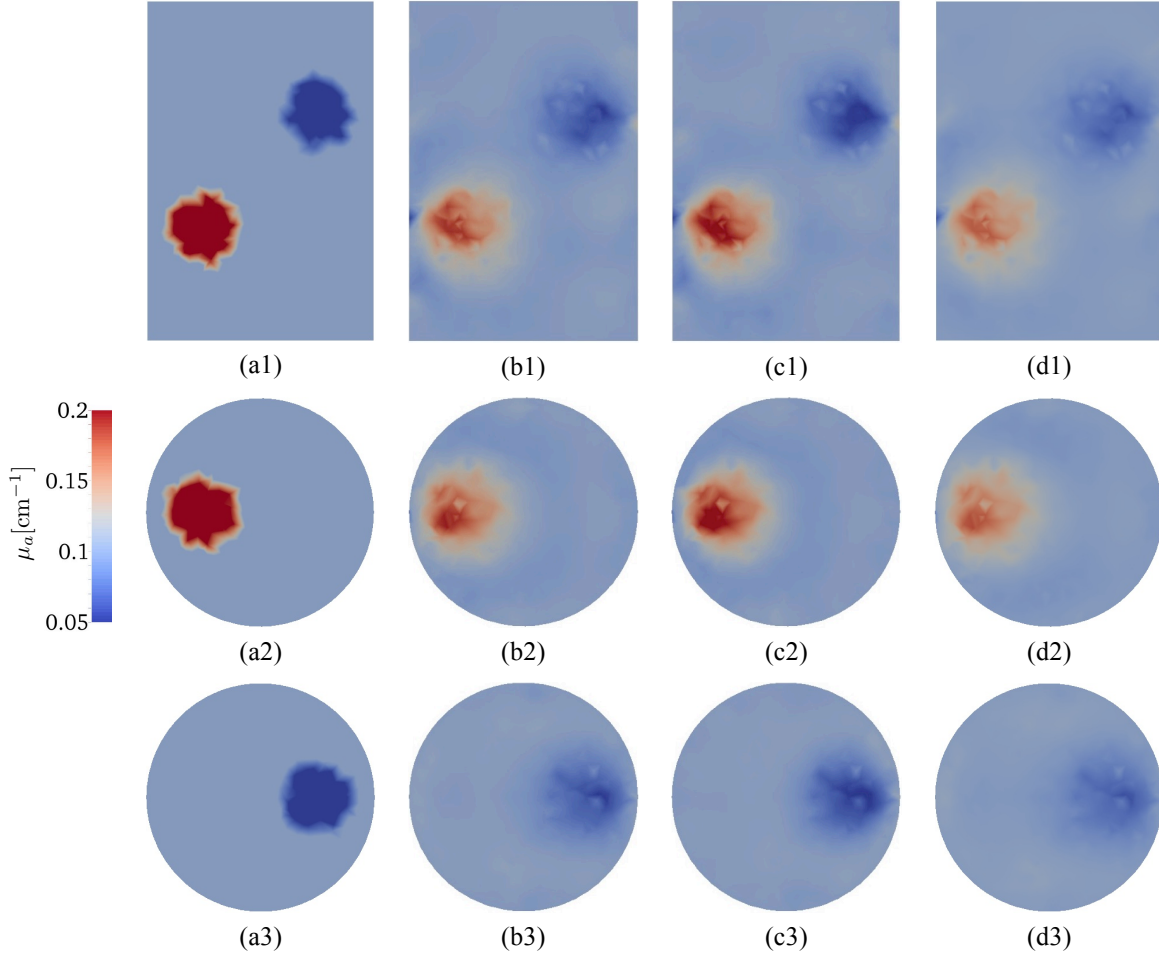


Figure 4.4: Absorption coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm.

The convergence history of the traditional BFGS, the line-search-free BFGS, and the full-space algorithm in this case are shown in Fig. 4.5. The relative change of the objective function value is plotted over iterations and CPU time, as well as the required iteration number and CPU time taken to decrease the relative objective function to certain tolerance levels in Fig. 4.5 (a)(b)(c)(d), respectively. As shown in Fig. 4.5, the line-search-free BFGS algorithm is more efficient than the traditional BFGS algorithm both with respect to the iteration number and CPU time. According to Fig. 4.5(a), the objective value reduction in the full-space algorithm is significantly larger in the first iteration than the other two methods, but the efficiency of the full-

space method significantly deteriorates with increasing the iteration number. Fig. 4.5(b) shows the full-space method in this case is the least efficient algorithm out of the three in terms of CPU time, while the line-search-free BFGS is the most efficient one. Fig. 4.5(c)(d) shows that the full-space algorithm and the line-search-free BFGS algorithm require about 30%~40% fewer iterations than the traditional BFGS algorithm to reach the relative objective function value of  $10^{-2.5}$ , while the line-search-free BFGS algorithm takes about 25% less CPU time than the traditional BFGS algorithm and over 50% less CPU time than the full-space algorithm.

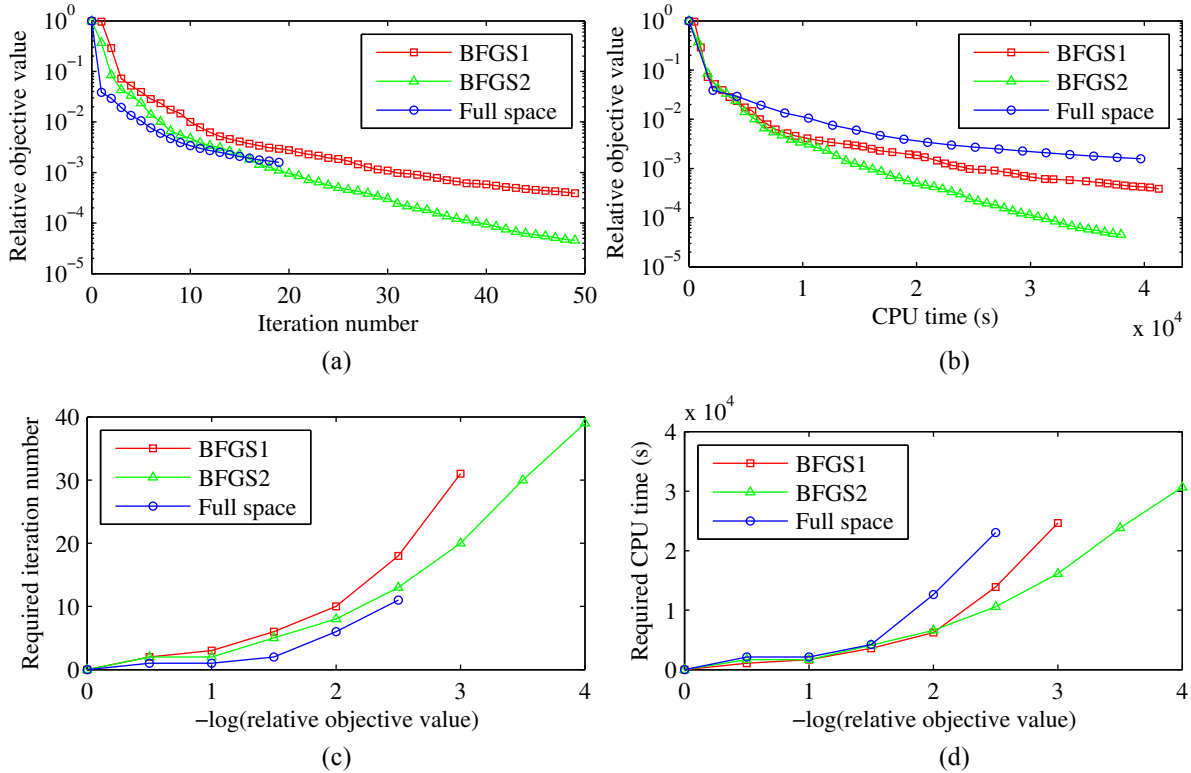


Figure 4.5: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels.

The correlation and deviation factors on the reconstructed absorption coefficient are given in Table 4.5 for all methods tested. With the BFGS type methods, 30 iterations are run for the 2D phantom and 50 iterations for the 3D phantom, while with the full-space algorithm, 15 iterations are taken for the 2D phantom and 20 iterations for the 3D phantom. Thus as shown in

Fig. 4.3 and 4.5, the overall CPU time for the full-space algorithm is longer than or comparable to the CPU time for the other two algorithms. The correlation factors are denoted by  $c_1, c_2, c_3$  and the deviation factors by  $d_1, d_2, d_3$  to compare the accuracy between the traditional BFGS, the line-search-free BFGS, and the full-space algorithm. As shown in Table 4.5, the line-search-free BFGS always gives the best results to the  $\mu_a$  reconstruction with highest correlation factor and lowest deviation factor. The comparison between the traditional BFGS and the full-space algorithm depends on the dimensionality of the phantom. The correlation and deviation factors for the 2D reconstructions are very close for both algorithms, while for the 3D phantoms the traditional BFGS algorithm outperforms the full-space algorithm in both accuracy metrics.

Table 4.5: Correlation factors and deviation factors of  $\mu_a$  reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 1~5.

Case	$c_1/c_2/c_3$	$d_1/d_2/d_3$
1	0.80 / <b>0.85</b> / 0.80	0.60 / <b>0.53</b> / 0.60
2	0.81 / <b>0.86</b> / 0.82	0.58 / <b>0.51</b> / 0.57
3	0.80 / <b>0.85</b> / 0.80	0.60 / <b>0.53</b> / 0.60
4	0.74 / <b>0.78</b> / 0.69	0.67 / <b>0.62</b> / 0.73
5	0.74 / <b>0.79</b> / 0.69	0.67 / <b>0.62</b> / 0.73

#### 4.4.4. Reconstruction of reduced scattering coefficient

In this subsection, the numerical simulations are conducted on the reconstruction of the reduced scattering coefficient using phantoms (Case 6~10) with the homogeneous absorption coefficient.

A 2D phantom is reconsidered here with two low scattering objects and one high scattering object. The reduced scattering coefficients of these three objects are  $1\text{cm}^{-1}$ ,  $5\text{cm}^{-1}$  and  $20\text{cm}^{-1}$  respectively. The distribution of  $\mu'_s$  is reconstructed with noise free measurements. For comparison, both the line-search-free BFGS and the traditional BFGS algorithms are run for 30 iterations, and the full-space algorithm for 15 iterations. The reconstructed  $\mu'_s$  distribution is shown in Fig. 4.6. The reconstructions of the two low scattering objects as shown in figures are similar for all the methods, while the line-search-free BFGS algorithm reconstructs the high

scattering object more accurately. The correlation factor and deviation factor are reported in Table 4.6 for all the reconstruction results for this case. As expected, the line-search-free BFGS algorithm has the highest  $c(\mu'_s, \mu'^*_s)$  and lowest  $d(\mu'_s, \mu'^*_s)$  out of the three, and the traditional BFGS method is slightly more accurate than the full-space algorithm.

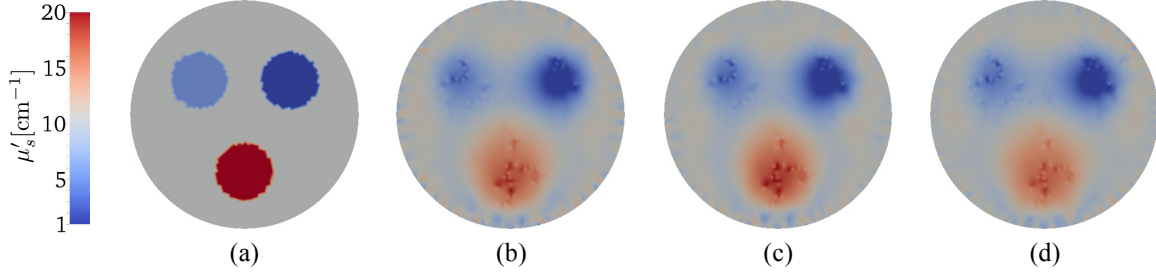


Figure 4.6: Reduced scattering coefficient distribution in the two-dimensional phantom. (a) Exact distribution; (b) Traditional BFGS; (c) Line-search-free BFGS; (d) Full-space algorithm.

Table 4.6: Correlation factors and deviation factors of  $\mu'_s$  reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 8.

Algorithm	$c(\mu'_s, \mu'^*_s)$	$d(\mu'_s, \mu'^*_s)$
Traditional BFGS	0.85	0.52
Line-search-free BFGS	<b>0.87</b>	<b>0.50</b>
Full-space algorithm	0.84	0.55

The comparison in the convergence rate between all the three algorithms is shown in Fig. 4.7, in which the relative objective function values are plotted over iterations and CPU time, the required iteration number and CPU time. Fig. 4.7 shows that the line-search-free BFGS is only slightly more efficient than the traditional BFGS algorithm both in terms of the iteration number and CPU time. The full-space algorithm has much slower convergence with respect to CPU time than the other two. For instance, to reduce the objective function value to  $10^{-2.5}$  and  $10^{-3}$ , the full-space algorithm requires about 100% and 200% CPU time than the other two algorithms. On the other hand, the full-space algorithm shows a dramatic decrease in the objective function value for the first 5 iterations but significantly slows down after the first 5 iterations. This can be observed in Fig. 4.7(c) as well: compared to the line-search-free BFGS algorithm, the full-space

algorithm requires about 45% fewer iterations to reduce the objective function to  $10^{-2}$ , while the algorithm exhibits slow convergence to  $10^{-3}$ .

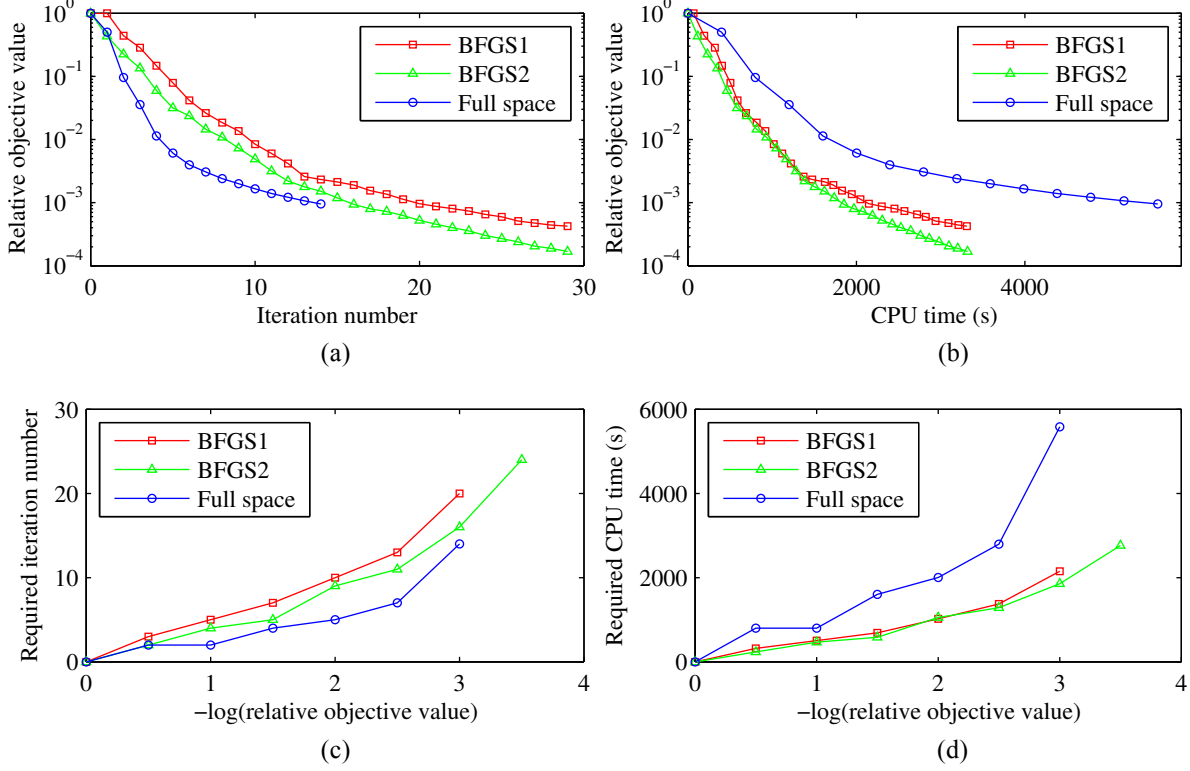


Figure 4.7: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels.

Next the numerical experiments are performed on the 3D phantom with homogeneous absorption coefficient (Case 10). A high scattering object and a low scattering object are located in the phantom, with reduced scattering coefficient as  $20\text{cm}^{-1}$  and  $5\text{cm}^{-1}$ . The reconstruction is conducted with noise free measurements. For comparison, the line-search-free BFGS and the traditional BFGS algorithms are run for 50 iterations and the full-space algorithm for 20 iterations. Three cross sections of the reconstructed  $\mu'_s$  distribution are shown in Fig. 4.8. In this case, the reconstruction quality of the low scattering object is very similar for all the three algorithms. However, the high scattering object is reconstructed less accurately by the full-space algorithm than the other two. No significant difference is observed in Fig. 4.8 between the

reconstruction results from the traditional BFGS and the line-search-free BFGS algorithm. The correlation and deviation factors for  $\mu'_s$  reconstruction are given in Table 4.7 for all the three methods. The results show that the line-search-free BFGS algorithm is of highest accuracy, and the traditional BFGS algorithm is almost as accurate as the line-search-free BFGS algorithm. The reconstruction accuracy of the full-space algorithm is significantly lower than the other two.

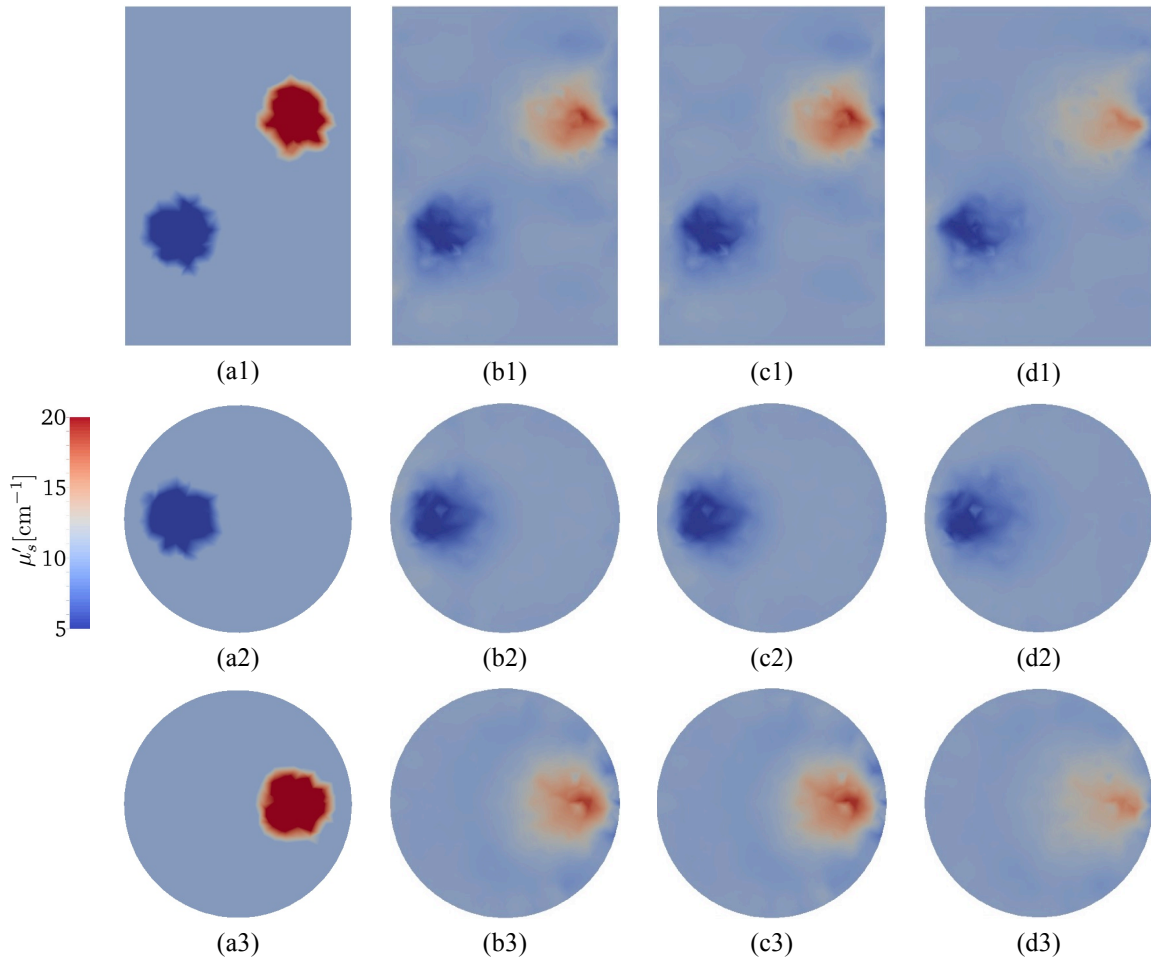


Figure 4.8: Reduced scattering coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm.

The convergence behavior is shown in Fig. 4.9 for the traditional BFGS, the line-search-free BFGS and the full-space algorithm. The traditional BFGS and the line-search-free BFGS in this case have very similar convergence properties as shown in both Fig. 4.9(a) and Fig. 4.9(b),



where the line-search-free BFGS is slightly more efficient than the others in terms of both iteration number and CPU time. The full-space algorithm does not show any advantage over the BFGS type algorithms for this case.

Table 4.7: Correlation factors and deviation factors of  $\mu'_s$  reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 10.

Algorithm	$c(\mu'_s, \mu'^*_s)$	$d(\mu'_s, \mu'^*_s)$
Traditional BFGS	0.75	0.67
Line-search-free BFGS	<b>0.76</b>	<b>0.66</b>
Full-space algorithm	0.67	0.76

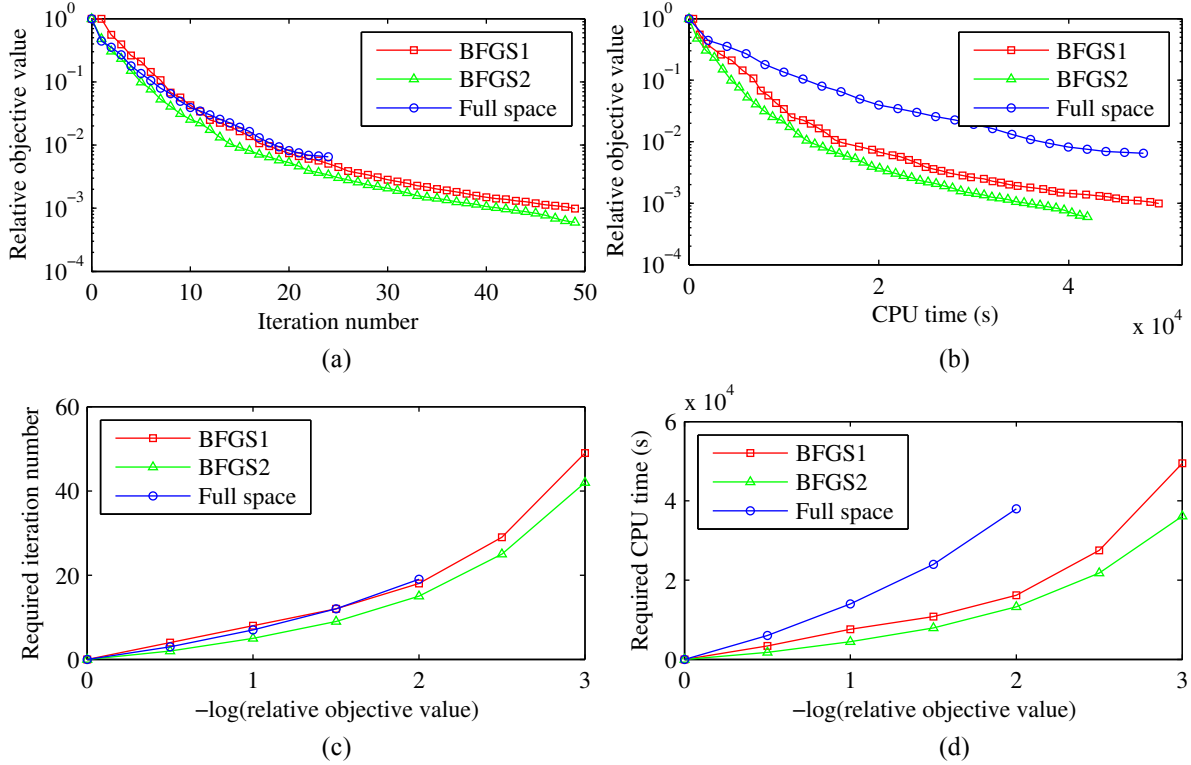


Figure 4.9: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels.

The correlation factors and deviation factors on the scattering coefficient reconstructions are reported in Table 4.8 for both methods on Case 6~10. With the BFGS type methods, 30 iterations are taken for the 2D phantom and 50 iterations for the 3D phantom. With the full-space algorithm, 15 iterations are run for the 2D phantom and 20 iterations for the 3D phantom. As

shown in Fig. 4.7 and 4.9, the overall CPU time for the BFGS type methods is less than or comparable to it in the full-space algorithm. In Table 4.8,  $c_1, c_2, c_3$  are used to denote the correlation factors and  $d_1, d_2, d_3$  are used to denote the deviation factors for the results from the traditional BFGS, the line-search-free BFGS, and the full-space algorithm, for the comparison purpose. Table 4.8 shows the most accurate reconstruction results of  $\mu'_s$  are from the line-search-free BFGS, which have the highest correlation factors and the lowest deviation factors. The  $\mu'_s$  reconstruction results from the traditional BFGS are slightly less accurate than them from the line-search-free BFGS algorithm but the difference is insignificant. The comparison between BFGS type algorithms and the full-space algorithm's performance highly depends on the dimensionality of the experimental phantom: for 2D phantoms, their accuracies are very close, while with the 3D phantoms, the accuracy of the full-space algorithm is significantly lower than the BFGS type algorithms.

Table 4.8: Correlation factors and deviation factors of  $\mu'_s$  reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 6~10.

Case	$c_1/c_2/c_3$	$d_1/d_2/d_3$
6	0.87 / <b>0.88</b> / 0.87	0.48 / <b>0.47</b> / 0.50
7	0.85 / <b>0.87</b> / 0.83	0.53 / <b>0.50</b> / 0.57
8	0.85 / <b>0.87</b> / 0.84	0.53 / <b>0.50</b> / 0.55
9	0.80 / <b>0.81</b> / 0.74	0.60 / <b>0.58</b> / 0.68
10	0.75 / <b>0.76</b> / 0.67	0.67 / <b>0.66</b> / 0.76

#### 4.4.5. Reconstruction of both absorption and reduced scattering coefficients

In this subsection, Case 11 and Case 12 are considered with the reconstruction of both the absorption coefficient and the reduced scattering coefficient.

The 2D phantom (Case 11) with three objects is considered first. The absorption coefficients for these objects are  $0.3\text{cm}^{-1}$ ,  $0.3\text{cm}^{-1}$  and  $0.1\text{cm}^{-1}$ , while the scattering coefficients are  $10\text{cm}^{-1}$ ,  $10\text{cm}^{-1}$  and  $15\text{cm}^{-1}$  respectively. The absorption coefficient  $\mu_a$  and

the reduced scattering coefficient  $\mu'_s$  are reconstructed with noise free measurements. For comparison, the line-search-free BFGS and the traditional BFGS algorithms are taken for 50 iterations and the full-space algorithm for 20 iterations, the reconstructed  $\mu_a$  and  $\mu'_s$  are shown in Fig. 4.10. For reconstructions of  $\mu_a$ , three high absorbing regions can be seen in the results from all the three algorithms, with two corresponding to the high absorbing objects and one corresponding to the cross-talk from the high scattering object. With the line-search-free BFGS algorithm,  $\mu_a$  of two high absorbing objects are reconstructed most accurately compared to the benchmark and the least cross-talk out of the three. The traditional BFGS algorithm has the second best result on the  $\mu_a$  reconstruction. For the full-space method, the two high absorbing objects are not clearly separated and their absorption coefficients are highly undervalued. For the reconstruction of  $\mu'_s$ , the line-search-free BFGS algorithm has the best contrast to the background value and the traditional BFGS algorithm has the second best reconstruction result, while the full-space algorithm again highly underestimates the  $\mu'_s$  in the high scattering object.

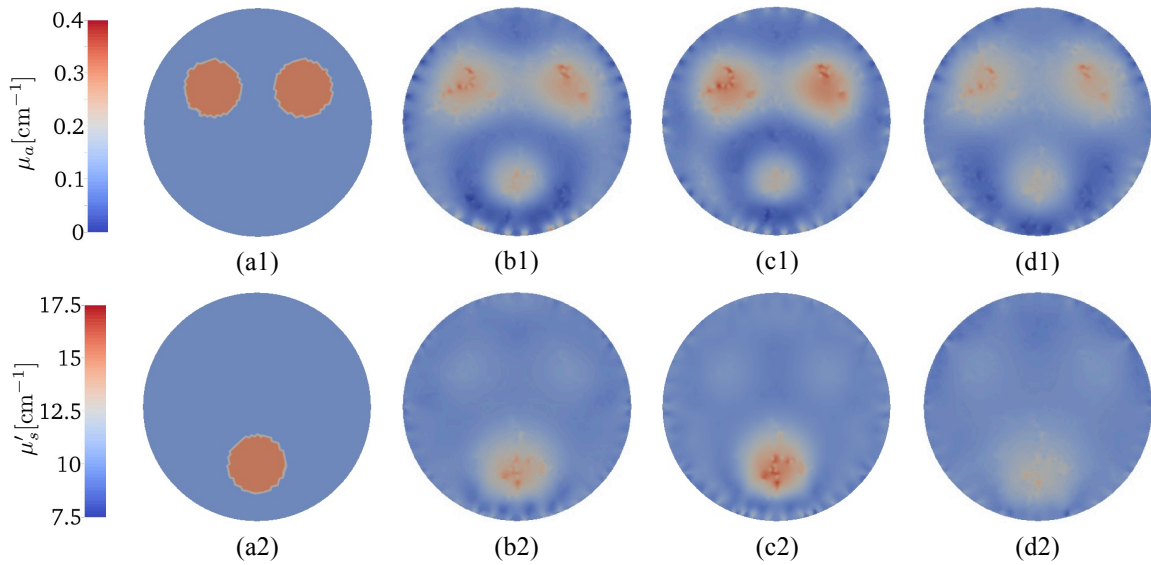


Figure 4.10: (Top)  $\mu_a$  and (bottom)  $\mu'_s$  distribution in the two dimensional phantom. (a1,2) Exact distribution; (b1,2) Traditional BFGS; (c1,2) Line-search-free BFGS; (d1,2) Full-space algorithm

To conduct quality comparison between three algorithms, their correlation factors and deviation factors on  $\mu_a$  and  $\mu'_s$  reconstruction are given in Table 4.9. The line-search-free BFGS algorithm is shown to have highest correlation and lowest deviation factors of the three. The correlation factor and deviation factor from the traditional BFGS algorithm, which is the second most accurate algorithm in this case, are significantly higher than those from the full-space algorithm.

Table 4.9: Correlation factors and deviation factors of  $\mu_a$  and  $\mu'_s$  reconstruction results from the traditional BFGS, the line-search-free BFGS and the full-space algorithm on Case 11.

Algorithm	$c(\mu_a, \mu_a^*)$	$d(\mu_a, \mu_a^*)$	$c(\mu'_s, \mu'^*_s)$	$d(\mu'_s, \mu'^*_s)$
Traditional BFGS	0.67	0.74	0.81	0.60
Line-search-free BFGS	<b>0.77</b>	<b>0.65</b>	<b>0.87</b>	<b>0.51</b>
Full-space algorithm	0.63	0.78	0.71	0.74

The convergence characteristics on Case 11 is examined and reported in Fig. 4.11 for the traditional BFGS, the line-search-free BFGS and the full-space algorithm. It is observed that the full-space algorithm has the lowest objective function value for the first 5 iterations, and the line-search-free BFGS method takes the lowest thereafter. Therefore, the line-search-free BFGS and the full-space algorithm require the same number of iterations, which is about 30% less than the traditional BFGS algorithm, to reduce the relative objective function to  $10^{-2}$ . In terms of CPU time, the line-search-free BFGS algorithm is the most efficient out of the three, while the full-space algorithm is the most time consuming one. To reduce the objective function value to  $10^{-2}$  and  $10^{-3}$ , the line-search-free BFGS requires about 30% and 40% less CPU time than the traditional BFGS algorithm respectively, and both BFGS algorithms are several time faster than the full-space algorithm.

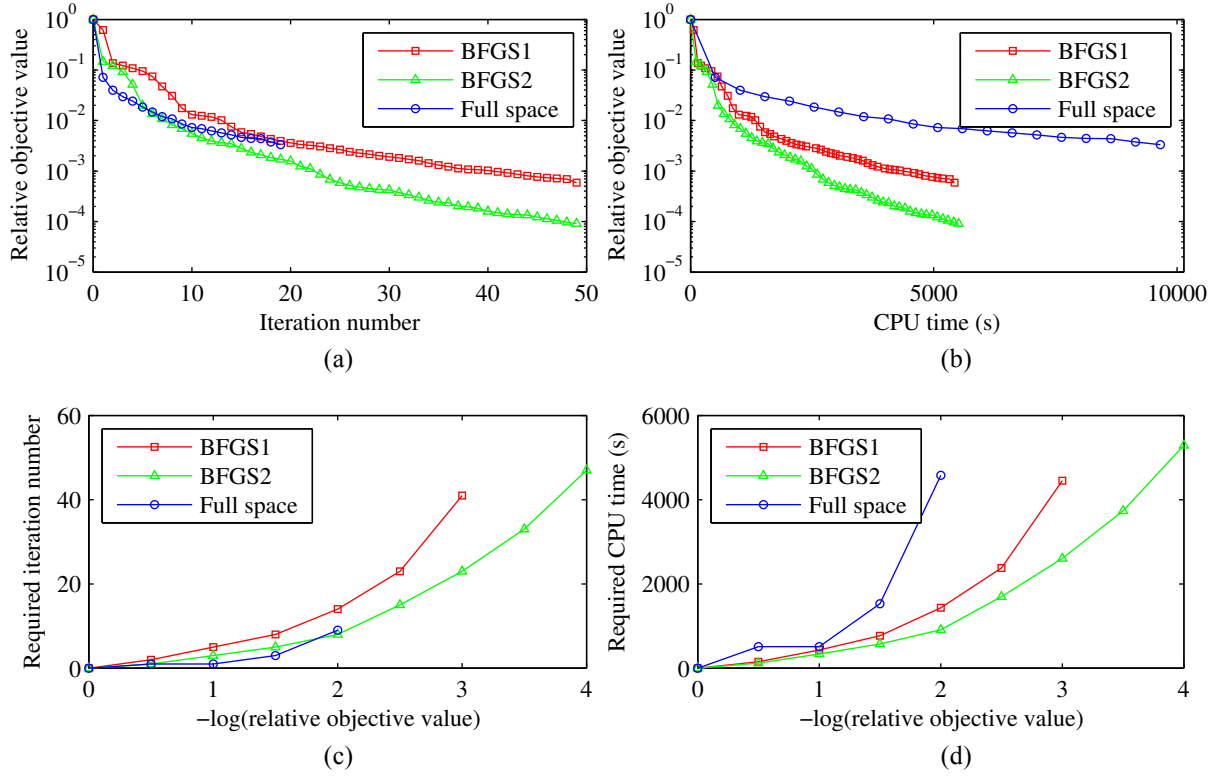


Figure 4.11: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels.

Next the 3D phantom (Case 12) is considered that contains a ball shape high absorbing object with  $\mu_a = 0.3\text{cm}^{-1}$  and a ball shape low scattering object with  $\mu'_s = 5\text{cm}^{-1}$ . Therefore, both objects are in the transport regime. The absorption coefficient  $\mu_a$  and the reduced scattering coefficient  $\mu'_s$  are reconstructed with the traditional BFGS, the line-search-free and the full-space algorithm. BFGS type algorithms are run for 70 iterations and the full-space method for 25 iterations in order to guarantee comparable CPU time. Three cross sections of  $\mu_a$  and  $\mu'_s$  reconstruction results are shown in Fig. 4.12 and Fig. 4.13 respectively. From the first two rows in Fig. 4.12, one can observe that  $\mu_a$  of the object is more accurately reconstructed with the line-search-free BFGS algorithm. Moreover, artifacts due to crosstalk are least in the line-search-free BFGS algorithm according to Row 1 and 3 in Fig. 4.12. For the  $\mu'_s$  reconstruction, the  $\mu'_s$  reconstruction by the line-search-free BFGS algorithm has the most accurate size and scattering

coefficient for the low scattering object as shown in the first and the third row of Fig. 4.13. The traditional BFGS is the second most accurate one in both  $\mu_a$  and  $\mu'_s$  reconstructions.

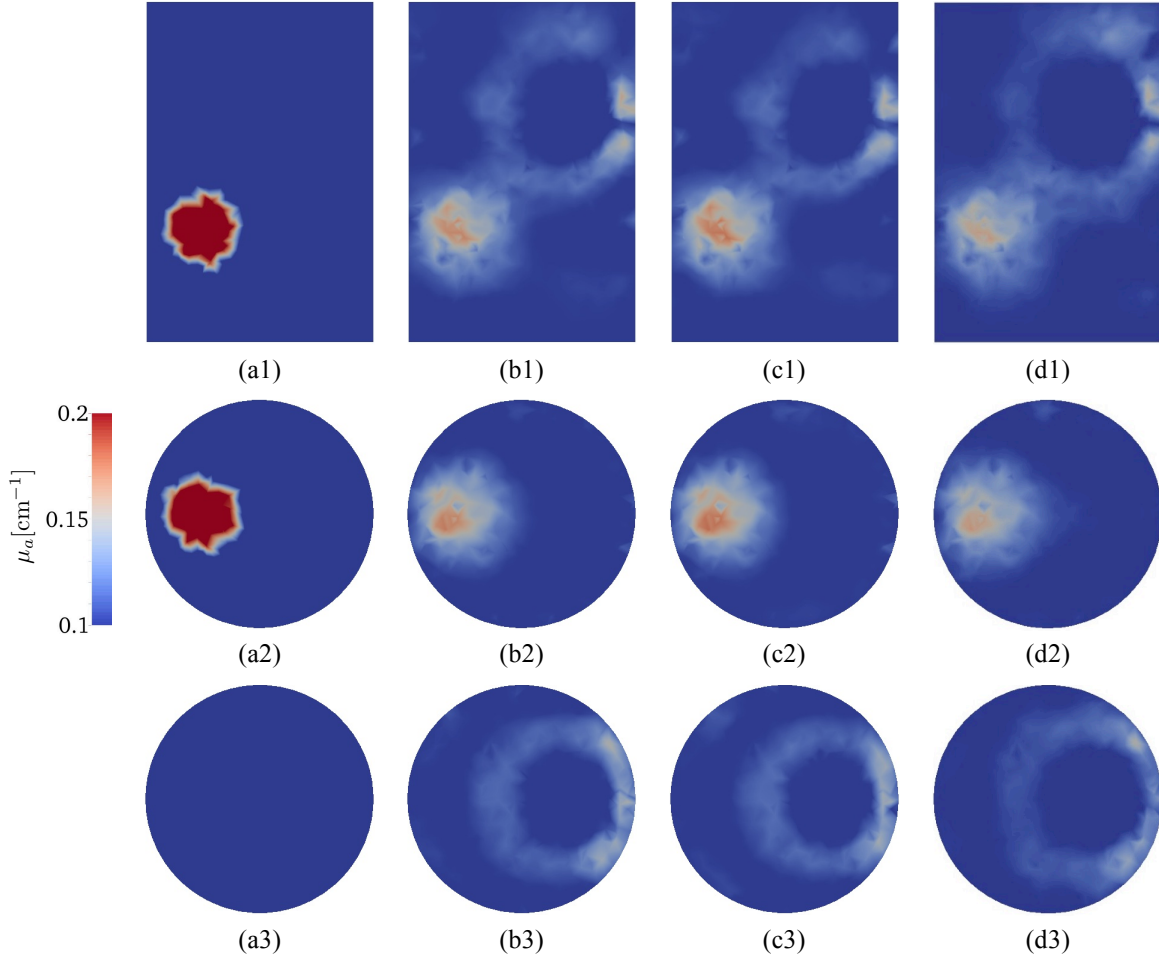


Figure 4.12: Absorption coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm.

The correlation and deviation factors for both  $\mu_a$  and  $\mu'_s$  reconstruction results are computed and reported in Table 4.10 for all algorithms. As expected, the line-search-free BFGS algorithm outperforms the other two algorithms in all the accuracy metrics. The traditional BFGS algorithm ranks the second in accuracy and the full-space algorithm ranks the last.

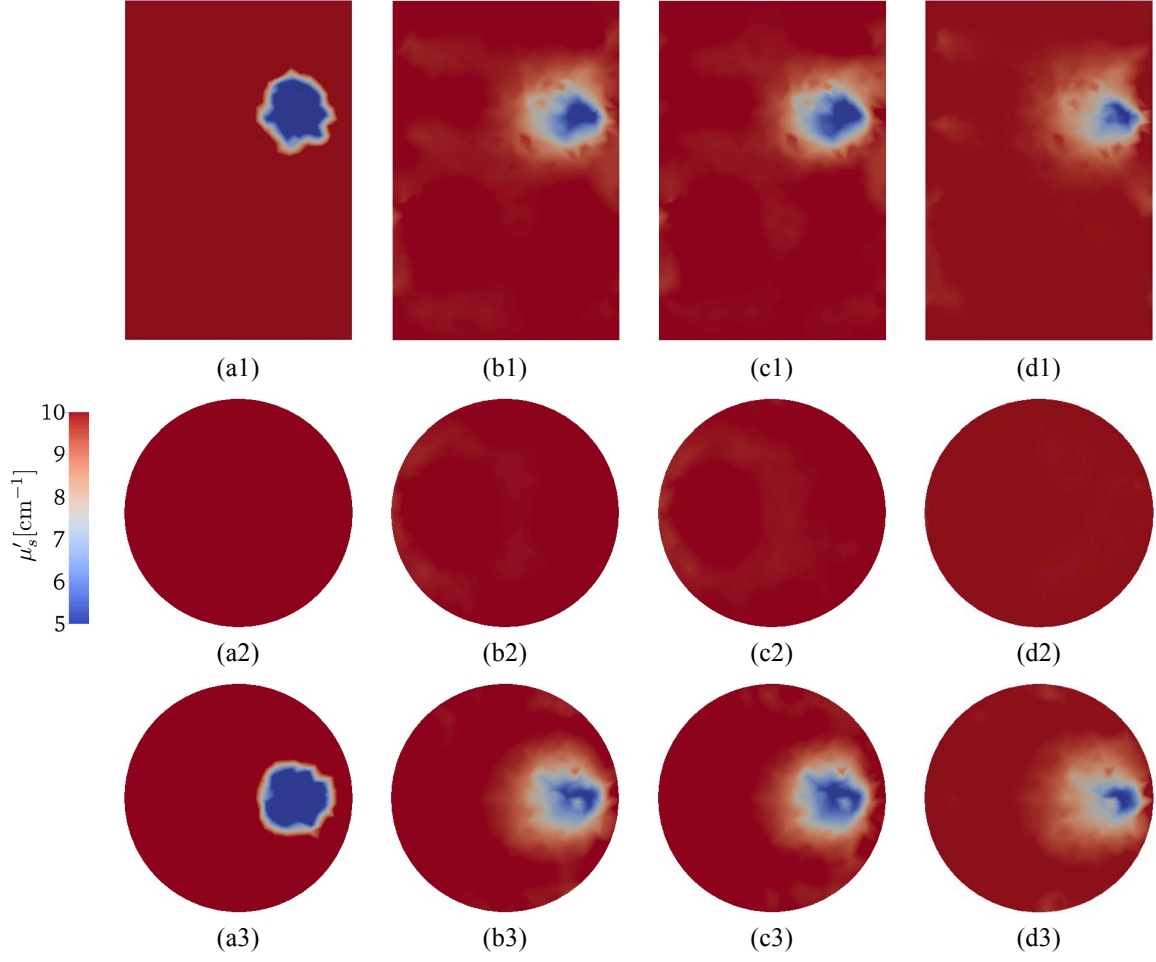


Figure 4.13: Absorption coefficient distribution in cross section  $x = 0$  (Row 1),  $z = 1$  (Row 2) and  $z = 2$  (Row 3) of the three-dimensional phantom: (a1,2,3) Exact distribution; (b1,2,3) Traditional BFGS; (c1,2,3) Line-search-free BFGS; (d1,2,3) Full-space algorithm.

Table 4.10: Correlation factors and deviation factors of  $\mu_a$  and  $\mu'_s$  reconstruction results from the traditional and line-search-free BFGS algorithms on Case 12.

Algorithm	$c(\mu_a, \mu_a^*)$	$d(\mu_a, \mu_a^*)$	$c(\mu'_s, \mu'_s^*)$	$d(\mu'_s, \mu'_s^*)$
Traditional BFGS	0.43	1.05	0.78	0.63
Line-search-free BFGS	<b>0.48</b>	<b>0.99</b>	<b>0.80</b>	<b>0.59</b>
Full-space algorithm	0.38	1.09	0.68	0.73

The convergence property of the traditional BFGS, the line-search-free BFGS and the full-space algorithm are examined and reported in Fig. 4.14, which shows the relative objective function values with respect to iterations and CPU time, and required iteration number and CPU time to decrease the objective function to some certain levels. Fig. 4.14(b) shows that in terms of

CPU time the line-search-free BFGS reduces the objective function with the fastest speed, while the full-space algorithm is the slowest. According to Fig. 4.14(d), the line-search-free BFGS algorithm requires about 30% and 65% less CPU time than the traditional BFGS and the full-space algorithm to reduce the objective function value to  $10^{-2}$ . In terms of iteration number, the objective value in the full-space algorithm decreases faster than the other two algorithms at the very early stage. However, its convergence speed drops down quickly with growing iteration number. Therefore the objective function value in the full-space algorithm becomes larger than the line-search-free BFGS and the traditional BFGS algorithm after 10 and 15 iterations respectively.

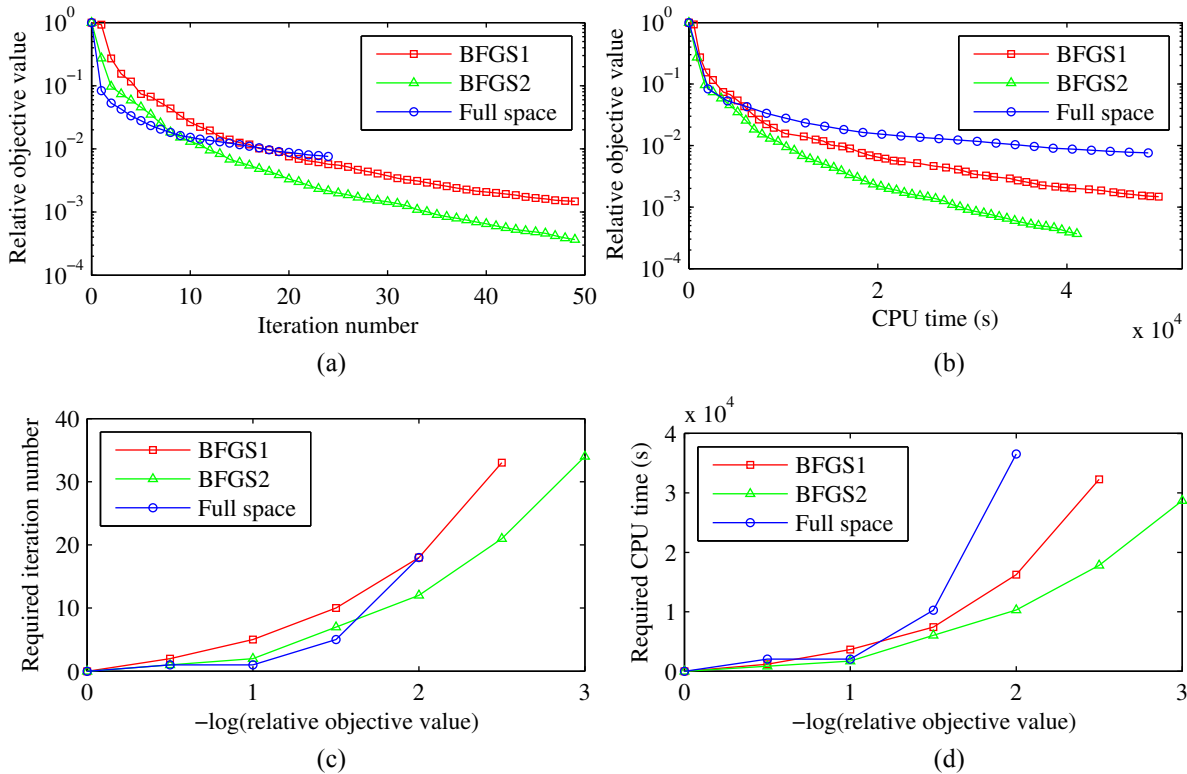


Figure 4.14: Performance profile of the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2) and the full-space algorithm: (a,b) Relative function value over iteration number and CPU time; (c,d) Required iteration number and CPU time to converge the objective function value to certain levels.



#### 4.4.6. Influence of noise

In the previous numerical experiments, it is assumed the measurements are noise free and no regularization is conducted on the optimization problem. However, actual measurements are contaminated with noise, which will affect the accuracy of the reconstruction result. Therefore, in this section the effects of noise on the traditional BFGS, the line-search-free BFGS, and the full-space algorithm is examined by varying the SNR from the infinity (noise free) to 20dB and 15dB, with the later two representing the typical noise levels in DOT [126]. The regularization function  $R(\mu)$  is employed in this case, which is given by

$$R(\mu) = \|\mu_a - 0.1\|^2/10. \quad (4.27)$$

The relative tolerance  $\epsilon$  is set to  $10^{-3}$  as the stopping criteria for the traditional BFGS (BFGS1), the line-search-free BFGS (BFGS2), and the full-space algorithm. The correlation factor, deviation factor, iteration number and CPU time for different SNR level are evaluated and reported in Table 4.11 for the three algorithms tested.

Table 4.11: The reconstruction quality and the CPU time with the traditional and the line-search-free BFGS algorithms for different noise levels.

	SNR	BFGS1	BFGS2	Full-space
$c(\mu_a, \mu_a^*)$	Noise free	0.66	0.66	0.66
	20dB	0.57	0.57	0.57
	15dB	0.44	0.44	0.44
$d(\mu_a, \mu_a^*)$	Noise free	0.78	0.78	0.78
	20dB	0.87	0.87	0.86
	15dB	1.28	1.28	1.28
CPU time (s)	Noise free	1343	<b>799</b>	1596
	20dB	1243	<b>937</b>	1943
	15dB	1220	<b>791</b>	1920
Iteration number	Noise free	12	9	<b>5</b>
	20dB	13	10	<b>6</b>
	15dB	13	9	<b>6</b>

As expected, higher noise level leads to lower correlation factor and higher deviation factor, which indicates the lower reconstruction quality. With the same optimization formulation and stopping criteria,  $c(\mu_a, \mu_a^*)$ s and  $d(\mu_a, \mu_a^*)$ s are approximately the same for all algorithms. The full-space algorithm requires the fewest iterations but the most CPU time to converge while the line-search-free BFGS algorithm needs the second fewest iterations and the least CPU time for convergence. With the traditional BFGS algorithm as a benchmark, the line-search-free BFGS requires about 25% fewer iterations and 25% ~ 40% less CPU time, while the full-space algorithm requires about 55% fewer iterations but 20% ~ 55% more CPU time to meet the same stopping criteria.

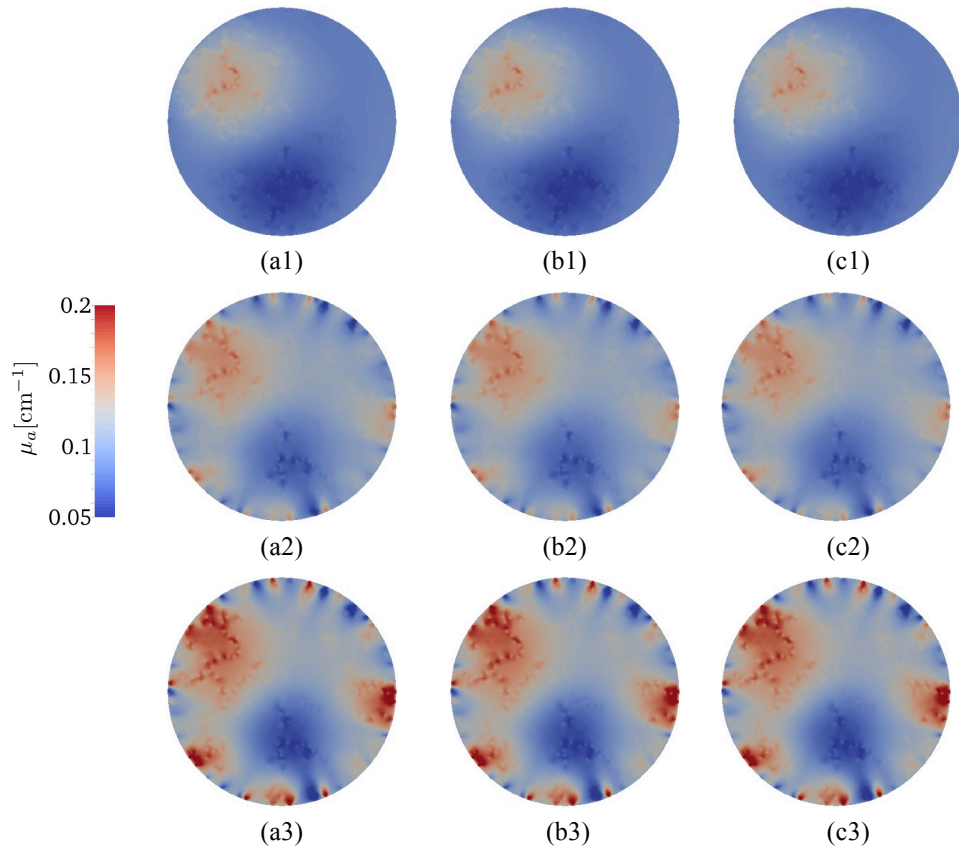


Figure 4.15: Absorption coefficient distribution in the two-dimensional phantom reconstructed with SNR level  $+\infty$  (Row 1), 20dB (Row 2), 15dB (Row 3) by different algorithms. (a1,2,3) Traditional BFGS; (b) Line-search-free BFGS; (c) Full-space algorithm.

The reconstruction results of the traditional BFGS, the line-search-free BFGS, and the full-space algorithm with different SNR level are shown in Fig. 4.15. With the same noise level, no visible difference is found between reconstruction results from different methods, which indicate that these three algorithm converges to the same point with the same optimization formulation, measurements and initial guess used as inputs.

#### 4.4.7. Influence of the initial guess

In all the previous experiments, the background optical properties are used as an initial guess to the reconstructions. However, this background optical property may not be known a priori in practice. Thus there can be some mismatch between the true background optical property and the initial guess value used, which has significant impacts on the reconstruction quality. Therefore, the effect of the biased initial guess on the reconstruction results is investigated in this section.

For this study the Case 2 is revisited again. The initial guess  $\mu_a$  is set to the exact background value  $0.1\text{cm}^{-1}$  and  $0.08\text{cm}^{-1}$ ,  $0.12\text{cm}^{-1}$  respectively in three trials, with the later two 20% lower and higher than the exact  $\mu_a$ . The regularization term is given as (4.27), and the stopping criteria is set to  $\epsilon = 10^{-4}$  for both traditional BFGS and line-search-free BFGS algorithms. The correlation factor, deviation factor, iteration number and CPU time for different initial guess  $\mu_a$  are reported in Table 4.12 for two algorithms.

In Table 4.12 we can observe that with the exact initial guess, the deviation factor of the reconstruction result is significantly lower but the correlation factor is just slightly lower than the other two. That means the optical property may be undervalued or overvalued but its general trend can be reconstructed in the reconstruction results. The line-search-free BFGS algorithm converges the fastest with respect the CPU time, while the full-space algorithm requires the fewest iteration number to reach the stopping criteria.

Table 4.12: The reconstruction quality and the CPU time with the traditional and the line-search-free BFGS algorithms for different initial guess.

	Initial guess	BFGS1	BFGS2	Full-space
$c(\mu_a, \mu_a^*)$	Exact	0.66	0.66	0.66
	20% higher	0.64	0.64	0.64
	20% lower	0.60	0.60	0.60
$d(\mu_a, \mu_a^*)$	Exact	0.78	0.78	0.78
	20% higher	1.00	1.00	1.00
	20% lower	0.98	0.98	0.98
CPU time (s)	Exact	1343	<b>799</b>	1596
	20% higher	1218	<b>768</b>	982
	20% lower	974	<b>565</b>	1307
Iteration number	Exact	12	9	<b>5</b>
	20% higher	12	8	<b>3</b>
	20% lower	10	7	<b>4</b>

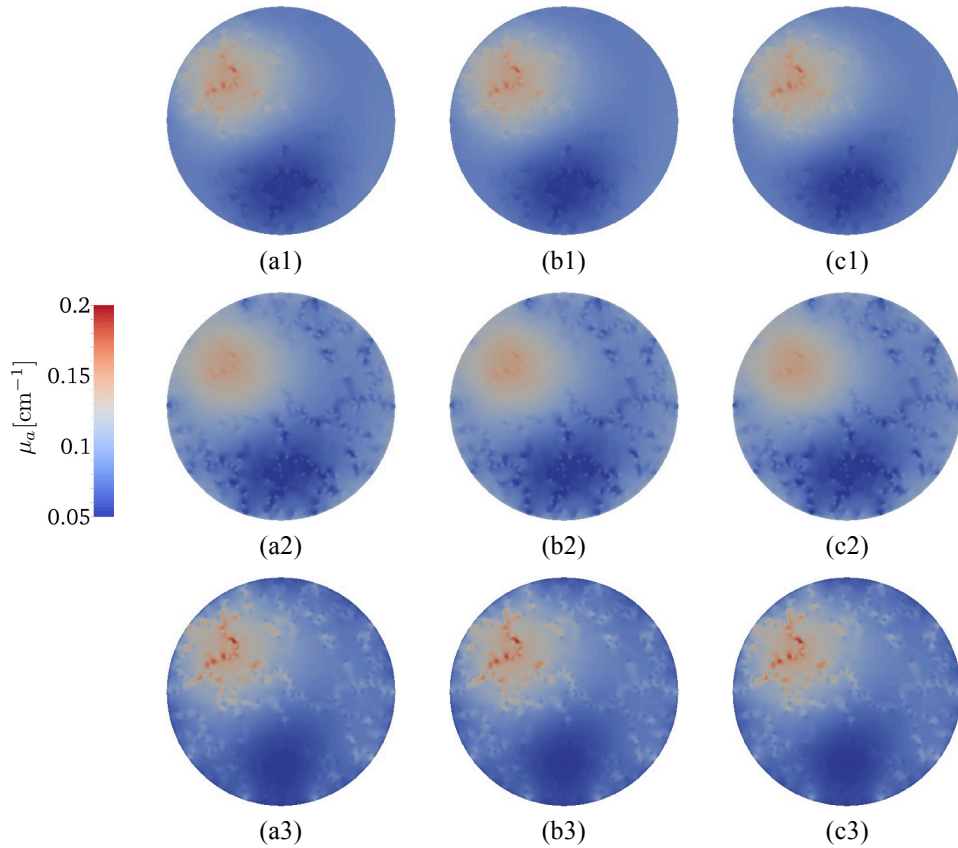


Figure 4.16: Absorption coefficient distribution in the two-dimensional phantom reconstructed with initial guess equal to (Row 1), 20% higher than (Row 2) and 20% lower than the background value by different algorithms. (a1,2,3) Traditional BFGS; (b) Line-search-free BFGS; (c) Full-space algorithm.

The reconstruction results of the traditional BFGS, the line-search-free BFGS and the full-space algorithm with different initial guess are shown in Fig. 4.15. In all the three trials, the high absorbing and low absorbing areas are successfully reconstructed, which shows the robustness of the proposed algorithms. However, with higher initial guess  $\mu_a$ , some sparse low absorption noise can be found in the reconstruction results and vice versa. This can be explained by the effect of the regularization term. For instance, if one regularizes the objective function with the higher initial guess, then the reconstruction will tend to generate locally low absorption noise with lower  $R(\mu)$  in the background to correct for the mismatch between the initial guess and the actual background optical property. There is no noticeable difference in the reconstructions from different algorithms in this case, which again shows that with the same optimization formulation, initial guess and stopping criteria, all three algorithms gives the same results.

## 4.5. Discussion

In his chapter, I presented the line-search-free BFGS algorithm and the first full-space algorithm for RTE-base DOT in this chapter. In the line-search-free BFGS algorithm, the image reconstruction in DOT is formulated as an unconstrained optimization problem. A BFGS algorithm is then used to find the descent direction  $\Delta\mu$  for the inverse variables  $\mu$ . Instead of conducting a line search process to find a proper step size  $\alpha$ , I use the first order Taylor approximation to the forward RTE to derive the change  $\Delta\Psi$  of the forward variables  $\Psi$  corresponding to  $\Delta\mu$ . Finally an optimization problem is solved to obtain  $\alpha$  with the original objective function as the merit function. This optimization has only one scalar variable  $\alpha$  and the function it minimizes often has a very straightforward form. Therefore, one can expect very low computational costs. The line-search-free BFGS algorithm has two main advantages: Firstly, one

can obtain the step size by solving an optimization problem. The determined step size  $\alpha$  is approximately the optimal step size, with which the objective function decreases the most. Secondly, the proposed line-search-free BFGS algorithm does not need extra objective function evaluations from the line search process, which requires solving forward problems in the traditional BFGS algorithm. Therefore, one can expect higher efficiency of the line-search-free BFGS algorithm in term of both iteration number and CPU time as compared to the traditional BFGS algorithm.

I also developed the first full-space algorithm for RTE-DOT. In this algorithm, the RTE-DOT is formulated as a constrained optimization problem. The Lagrangian multiplier method is applied in order to get the first order KKT condition, which is later solved with Newton's method with a left preconditioned BiCGStab linear solver. Due to the high ill-posedness in the DOT problem, only the component corresponding to the inverse variables  $\mu$  is retained as a descent direction after the KKT condition is solved to avoid the instability issue. This is different from all existing full-space algorithms, which use all the components to update the forward, inverse, and adjoint variables. Since the full Hessian is used to update the descent direction, higher convergence rate can be expected, which lead to fewer iterations in the image reconstruction process.

Both algorithms are validated and their performances are evaluated with traditional BFGS, the state of art algorithm in unconstrained optimization, as a benchmark. The numerical experiments show the line-search-free BFGS is more efficient than the traditional BFGS algorithm in terms of both iteration number and CPU time. For instance, to reduce the relative objective function value to  $10^{-2}$  with the noise free measurements, the line-search-free BFGS algorithm requires 10% ~ 40% fewer iterations and 10% ~ 40% less computational time than the

traditional BFGS algorithm, depending on the problem. For the full-space algorithm, its objective value with respect to iteration number usually decreases much faster than the traditional BFGS algorithm at the early stage; however on the other hand, each inverse iteration is much more time consuming than in the traditional BFGS algorithm. Therefore, its CPU time is usually longer than the traditional BFGS to meet a certain criterion, though the required iteration number in most of the time is less. It is also observed that the efficiency of the full-space algorithm drops quickly with growing iteration number. This is because any preconditioning was not conducted on the Hessian matrix  $W$  corresponding to the inverse variable  $\mu$ . Therefore the improvement of the condition number in the KKT system is limited, which makes it difficult to solve, especially when  $\mu$  is close to the optimal solution. It still remains as a challenging topic and future work on how to precondition the Hessian matrix  $W$  in RTE-DOT. One can expect significant efficiency boost of the full-space algorithm in RTE-DOT if one can find a proper preconditioner for  $W$ .

The robustness of the proposed line-search-free BFGS algorithm and the full-space algorithm were tested by running reconstructions with noise-contaminated measurements and different initial guess. A very strong agreement in the results was observed between the traditional BFGS algorithm, the line-search-free BFGS algorithm and the full-space algorithm, which shows that the second two algorithms are reliable and robust to noise and biased initial guess.

Due to the high efficiency in CPU time, the line-search-free BFGS algorithm is recommended for the RTE-DOT.

## **Chapter 5. Measurement Operators for general Non-contact DOT Systems**

### **5.1. Introduction**

Imaging systems for RTE-DOT have long relied on bringing and collecting light to the tissue of interest through optical fibers. These fibers are typically in contact with the tissue [127-129]. This has often limited the number of detectors, which highly restricts the resolution of the reconstructed image in RTE-DOT. Furthermore, the fibers need to be rearranged according to the geometry of the object for every different experiment, which brings lots of inconvenience and may introduce very strong noise if any fiber does not well contact the tissue surface.

To overcome these disadvantages, non-contact measurements that involve so-called charged-coupled-device (CCD) or complementary metal–oxide–semiconductor (CMOS) cameras have been explored in recent years. These approaches have shown significant advantages in detection sensitivity, image quality and system simplicity [130-132]. In non-contact imaging a way to model photon transport in free space is required, in addition to a model for light propagation inside tissue. All proposed numerical algorithms [133, 134] to simulate the light propagation in tissue are very computationally expensive for photon-transport process in free space. Monte-Carlo simulation [135-137], as a universal method, can handle this problem. However to obtain a reasonable result, a large number of photons is needed for the simulations, which leads to a relatively low efficiency. To overcome this problem, Ripoll et al proposed an efficient free-space light transport model [11] that does not employ statistical MC methods. Instead they compute an integral of light intensity for every point on the focal plane over the directions that can lead the photon to the aperture through that point. Based on this model, Schulz et al. proposed a simplified model [130] using perspective projection method by replacing



the camera lens with a virtual pinhole. In 2009 and 2010, X. Chen et al. published two improved models [138, 139] that are based on the hybrid radiosity-radiance theorem. The influence of the camera lens was analyzed with the thin lens assumption in both models. In addition, in one of the models, the influence of the camera lens diaphragm was taken into account.

Yet, all existing models so far still have two major limitations. First, they do not fully consider the angular dependency of the light intensity in the model so they only work in the limit of the diffusion equation (DE). Second, they do not take additional optical elements that are typically used into account. To gather more information of the surface light intensity distribution, an optical system, such as mirrors system or lens group, is often placed between the object and the CCD camera. Lack of consideration of such cases also limits the applications of the aforementioned researches in practice.

To overcome these problems, a back ray-tracing model is proposed to simulate the photon transport process in free-space. In this model, the pseudo photons are shot from the CCD chip and transported back to the object's surface. In this way a mapping is established between the angular dependent photons on the surface of the object and those on the detector of CCD camera. Then a coordinate transformation is applied to convert the integral over the solid angle on the object surface to an integral over solid angle on the CCD chip. The determinant of the transformation Jacobian matrix is estimated with the perturbation method. With this model, the contribution of photons from different surface locations and directions to signals received on the CCD camera can be expressed as an linear operator, which is required for formulating the optimization problem in DOT. The proposed model fully considers the angular dependency of intensity, and thus can be applied in RTE-DOT, which provides higher accuracy in many cases. Moreover, this proposed model could handle photon transport problems with a generalized

optical system between the object and the CCD camera to collect more signals, which is often limited by the size of aperture. That can highly improve the performance in receiving photons from more perspectives. Thus it is reasonable to expect a better performance and more reliable result in non-contact DOT with this proposed model.

The remainder of the chapter is organized as follows. The detailed derivation of the novel backward ray-tracing algorithm is given in Section 5.2. Then this model is validated with numerical experiments and reconstruction results in Section 5.3. This chapter finally concludes with a discussion in Section 5.4.

## 5.2. Theory and methods

### 5.2.1. Surface radiation coordinate system

According to the knowledge of surface radiation theory [140], once the radiative light intensity distribution  $\psi(\mathbf{r}, \mathbf{s})$  [W/mm<sup>2</sup>/sr] in an object  $O$  has been already given, the total emission power of  $P_{\text{surf}}$  [W] on a small area  $\Gamma$  on its surface  $\partial O$  can be given by

$$P_{\text{surf}}(\Gamma) = \int_{\Gamma} \int_{2\pi} \psi(\mathbf{r}, \mathbf{s}) \left(1 - R(\mathbf{s} \cdot \mathbf{n}(\mathbf{r}))\right) \mathbf{s} \cdot \mathbf{n}(\mathbf{r}) d\mathbf{s} d\mathbf{r}, \quad (5.1)$$

where  $\mathbf{r}$  is a position vector that indicate the location on the surface of the object;  $\mathbf{s}$  and  $\mathbf{n}(\mathbf{r})$  are two unit vectors that represent the photon's propagation direction and the outgoing normal vector on the surface respectively, where the normal vector is a function of the location;  $R(\cdot)$  represents the reflectivity on the tissue-air boundary, which is defined in the same way as it in (2.10). Therefore, the infinitesimal power  $dP(\mathbf{r}, \mathbf{s})$  that emits from the infinitesimal solid angle  $\mathbf{s}$  centered at  $\mathbf{r}$  is given by

$$dP_{\text{surf}}(\mathbf{r}, \mathbf{s}) = \psi(\mathbf{r}, \mathbf{s}) \left(1 - R(\mathbf{s} \cdot \mathbf{n}(\mathbf{r}))\right) \mathbf{s} \cdot \mathbf{n}(\mathbf{r}) d\mathbf{s} d\mathbf{r}, \quad \mathbf{r} \in \Gamma. \quad (5.2)$$

To use the (5.2) numerically, we express the vector  $\mathbf{r}$  and  $\mathbf{s}$  with parametric coordinate system. The solid angle vector  $\mathbf{s}$  is commonly represented with the spherical coordinate system

$$\mathbf{s} = \mathbf{s}(\theta, \varphi) := (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta). \quad (5.3)$$

Since the location vector  $\mathbf{r}$  in (5.2) is on a local piece  $\Gamma$  of a 2D manifold  $\partial O$ , it is convenient to parameterize it by a locally differentiable parametric equation with two free parameters  $(\lambda_1, \lambda_2)$ ,

$$\mathbf{r} = \mathbf{r}(\lambda_1, \lambda_2). \quad (5.4)$$

Options for  $\lambda_1$  and  $\lambda_2$  are not unique. However, with the tetrahedron mesh discretization, the object surface is discretized with triangle elements correspondingly (see a cylinder example in Fig. 5.1(a)). It is natural to assume  $\Gamma$  is a triangle element on the surface (See Fig. 5.1(b)) and to consider the parametric coordinate system for  $\mathbf{r} \in \Gamma$ .

Here the 2D barycentric coordinate system [141] (also referred to as the areal coordinates system) is briefly introduced. Assume the location vectors of the three vertices of the host triangle element are  $\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C$ , we define three vectors  $\mathbf{v}_{AB} := \mathbf{r}_B - \mathbf{r}_A$ ,  $\mathbf{v}_{AC} := \mathbf{r}_C - \mathbf{r}_A$  and  $\mathbf{v} := \mathbf{r} - \mathbf{r}_A$  (See Fig. 5.1(b)). Therefore, equation  $\mathbf{v} = \lambda_1 \mathbf{v}_{AB} + \lambda_2 \mathbf{v}_{AC}$  has a unique pair of solution  $(\lambda_1, \lambda_2)$  and it is used as  $\mathbf{r}$ 's local parametric coordinates. This coordinate system leads to a straightforward expression of  $\mathbf{r}(\lambda_1, \lambda_2)$  (in Fig. 5.1(b)), which is given by

$$\mathbf{r}(\lambda_1, \lambda_2) := \lambda_1 \mathbf{v}_{AB} + \lambda_2 \mathbf{v}_{AC} + \mathbf{r}_A \quad (5.5)$$

Another benefit from this coordinate system is the normal vector  $\mathbf{n}$  is a constant and does not depend on the location  $\mathbf{r}$  within  $\Gamma$ . Therefore, according to (5.3) and (5.5), we have

$$\begin{cases} \mathbf{n} d\mathbf{r} = (\partial \mathbf{r} / \partial \lambda_1) \times (\partial \mathbf{r} / \partial \lambda_2) d\lambda_1 d\lambda_2 = 2|\Gamma| d\lambda_1 d\lambda_2 \\ s ds = \sin \theta s d\varphi d\theta \end{cases} \quad (5.6)$$

where  $|\Gamma|$  is the area of triangle element  $\Gamma$  and can be pre-calculated after the mesh generation.

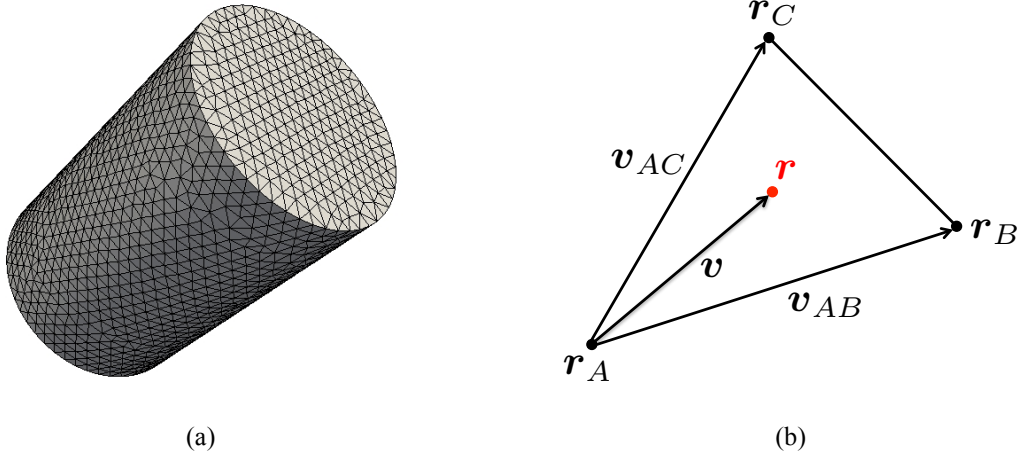


Figure 5.1: An illustration of (a) surface discretization with triangle elements and (b) a triangle element  $\Gamma$  and a location vector  $\mathbf{r}$ .

With (5.6), we can construct a four dimensional coordinate system to express the angular dependent the light intensity on  $A$  and the infinitesimal power  $dP(\Gamma)$  in (5.2) can be given by

$$P_{\text{surf}}(\Gamma) = \iint_{\Omega_{\mathbf{r}}(\Gamma) \times \Omega_{\mathbf{s}}(\Gamma)} q_{\text{surf}}(\lambda_1, \lambda_2, \varphi, \theta) d\varphi d\theta d\lambda_1 d\lambda_2, \quad (5.7)$$

where  $\Omega_{\mathbf{r}}(\Gamma)$  and  $\Omega_{\mathbf{s}}(\Gamma)$  represents the feasible set for location vector  $\mathbf{r}$  and direction vector  $\mathbf{s}$  under this coordinate system,

$$\begin{aligned} \Omega_{\mathbf{r}}(\Gamma) &:= \{(\lambda_1, \lambda_2) : \mathbf{r}(\lambda_1, \lambda_2) \in \Gamma\}, \\ \Omega_{\mathbf{s}}(\Gamma) &:= \{(\varphi, \theta) : \mathbf{s}(\varphi, \theta) \cdot \mathbf{n}(\Gamma) > 0\}. \end{aligned}$$

$q_{\text{surf}}(\lambda_1, \lambda_2, \varphi, \theta)$  is the power density function on the surface under this coordinate system, which is given by

$$q_{\text{surf}}(\lambda_1, \lambda_2, \varphi, \theta) = 2 \sin \theta |\Gamma| (1 - R(\mathbf{s} \cdot \mathbf{n})) \psi(\mathbf{r}, \mathbf{s}) \quad (5.8)$$

In practice,  $\psi(\mathbf{r}, \mathbf{s})$  is solved with discretized forward solvers thus it is only available for the triangle vertices  $\{\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C\}$  and for certain solid angles  $\{s_i\}_{i=1}^{N_{\text{SA}}}$ , so we can approximate

$\psi(\mathbf{r}, \mathbf{s})$  for any  $\mathbf{s}$  and  $\mathbf{r} \in \Gamma$  with the linear interpolation in the spatial domain and the nearest neighbor method in the solid angle domain.

$$\begin{aligned}\psi(\mathbf{r}(\lambda_1, \lambda_2), \mathbf{s}) &\approx (1 - \lambda_1 - \lambda_2)\psi(\mathbf{r}_A, \mathbf{s}) + \lambda_1\psi(\mathbf{r}_B, \mathbf{s}) + \lambda_2\psi(\mathbf{r}_C, \mathbf{s}) \\ &\approx (1 - \lambda_1 - \lambda_2)\psi(\mathbf{r}_A, \mathbf{s}_k) + \lambda_1\psi(\mathbf{r}_B, \mathbf{s}_k) + \lambda_2\psi(\mathbf{r}_C, \mathbf{s}_k),\end{aligned}\quad (5.9)$$

where  $k = \operatorname{argmin}_k \|\mathbf{s} - \mathbf{s}_k\|$ .

### 5.2.2. CCD camera acceptance coordinate system

To reduce the complexity of the model, the CCD camera is assumed to only consist of a single thin lens  $\Omega_{\text{lens}}$  (the aperture) with a radius as  $R_{\text{lens}}[\text{mm}]$  and a CCD chip  $\Omega_{\text{CCD}}$  with size  $l_{\text{CCD},1}[\text{mm}] \times l_{\text{CCD},2}[\text{mm}]$  (See Fig. 5.2). Under this assumption, the readings on the CCD chip are contributed by effective photons those pass through the aperture and finally reach the CCD chip after the refraction (the red solid arrow line in Fig. 5.2). Any effective photon can be uniquely identified with two location vectors: (1)  $\mathbf{r}_{\text{lens}}$  defined as the intersection point between its optical path and  $\Omega_{\text{lens}}$  and (2)  $\mathbf{r}_{\text{CCD}}$  defined as its final position on  $\Omega_{\text{CCD}}$  (see Fig. 5.2). Therefore the total energy that received by the CCD chip  $P_{\text{CCD}} [\text{W}]$  can be calculated with an integral over the light intensity  $J(\mathbf{r}_{\text{lens}}, \mathbf{r}_{\text{CCD}})$  on  $\Omega_{\text{lens}} \times \Omega_{\text{CCD}}$ ,

$$P_{\text{CCD}} = \int_{\Omega_{\text{lens}} \times \Omega_{\text{CCD}}} J(\mathbf{r}_{\text{lens}}, \mathbf{r}_{\text{CCD}}) (\mathbf{r}_{\text{CCD}} - \mathbf{r}_{\text{lens}}) \cdot \mathbf{n}_{\text{CCD}} d\mathbf{r}_{\text{lens}} d\mathbf{r}_{\text{CCD}}, \quad (5.10)$$

where  $\mathbf{n}_{\text{CCD}}$  is the normal vector of the CCD chip that points to the other side of  $\mathbf{r}_{\text{lens}}$ .

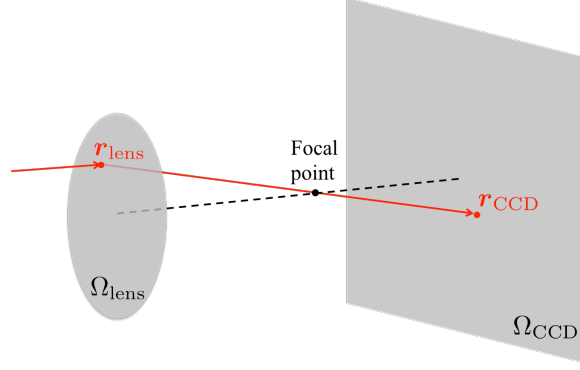


Figure 5.2: An illustration of a simplified model of the CCD camera

Based on the circular shape of the aperture and the rectangular shape of the CCD chip, we represent  $\mathbf{r}_{\text{lens}}$  and  $\mathbf{r}_{\text{CCD}}$  with the polar coordinate system

$$\begin{cases} \mathbf{r}_{\text{lens}} = \mathbf{r}_{\text{lens}}(\rho, \omega), & 0 \leq \rho \leq R_{\text{lens}}, 0 \leq \omega < 2\pi \\ \mathbf{r}_{\text{CCD}} = \mathbf{r}_{\text{CCD}}(x, y), & 0 \leq x \leq l_{\text{CCD},1}, 0 \leq y \leq l_{\text{CCD},2} \end{cases} \quad (5.11)$$

With this parametric expression (5.10) can be written as

$$P_{\text{CCD}} = \int_0^{l_{\text{CCD},2}} \int_0^{l_{\text{CCD},1}} \int_0^{2\pi} \int_0^{R_{\text{lens}}} q_{\text{CCD}}(\rho, \omega, x, y) d\rho d\omega dx dy, \quad (5.12)$$

where the energy density  $q_{\text{CCD}}(\rho, \omega, x, y)$  [W/mm<sup>3</sup>/sr] under this coordinate system is to be calculated in the next section.

### 5.2.3. Light propagation in free space and coordinate transformation

The status of any photon that emits from the object surface and is finally received by the CCD chip (also referred as the effective photons) can be described with the coordinates  $(\rho, \omega, x, y)$  or with an triangle element  $\Gamma$  and its corresponding ordinates  $(\lambda_1, \lambda_2, \varphi, \theta)$ . Therefore, we can define two set based on these two coordinate systems to describe the status of effective photons.

$$\begin{aligned} S_{\text{surf}} &:= \{(\Gamma, \lambda_1, \lambda_2, \varphi, \theta): \text{photon with status } (\Gamma, \lambda_1, \lambda_2, \varphi, \theta) \text{ is effective} \} \\ S_{\text{CCD}} &:= \{(\rho, \omega, x, y): \text{photon with status } (\rho, \omega, x, y) \text{ is effective} \} \end{aligned}$$

Therefore for any  $(\Gamma, \lambda_1, \lambda_2, \varphi, \theta) \in S_{\text{surf}}$ , we have  $(\lambda_1, \lambda_2) \in \Omega_r(\Gamma)$  and  $(\varphi, \theta) \in \Omega_s(\Gamma)$ ; for any  $(\rho, \omega, x, y) \in S_{\text{CCD}}$ ,  $0 \leq \rho \leq R_{\text{lens}}$ ,  $0 \leq \omega < 2\pi$ ,  $0 \leq x \leq l_{\text{CCD},1}$  and  $0 \leq y \leq l_{\text{CCD},2}$  are satisfied. We can also consider  $S_{\text{surf}}$  as the initial status set of effective photons since it characterizes the starting positions and directions, on the other hand,  $S_{\text{CCD}}$  can be considered as the final status set of effective photons since it contains information on the CCD camera side.

In non-contact imaging, some optical systems, such as conical mirror, are often employed to magnify the signal received by the CCD camera. To track the contribution of photons in complex optical system, we represent the light propagation in free space with an operator  $F$ , which maps from the initial status set  $S_{\text{surf}}$  to its final status set  $S_{\text{CCD}}$ . In this dissertation, the following assumptions are imposed on the operator  $F$ :

- (1) The operator  $F$  is a one to one and onto deterministic function from  $S_{\text{surf}}$  to  $S_{\text{CCD}}$ .

Therefore the light propagation operator  $G := F^{-1}$  is well defined.

- (2) The operator  $F$  is locally differentiable. In another word,  $F_\Gamma: (\lambda_1, \lambda_2, \varphi, \theta) \rightarrow (\rho, \omega, x, y)$ , which is defined as  $F$  constrained on the surface triangle element  $\Gamma$ , and its inverse  $G_\Gamma$  are differentiable.
- (3) There is no loss of energy during the travel of the light from the surface to the CCD chip.

These assumptions are not strong and satisfied by most of the general optical systems. Under these assumptions, we consider the total energy received from an infinitesimal unit volume  $d\rho d\omega dx dy$  around  $(\rho, \omega, x, y)$  by the CCD chip, which is given by

$$dP_{\text{CCD}}(\rho, \omega, x, y) = q_{\text{CCD}}(\rho, \omega, x, y) d\rho d\omega dx dy. \quad (5.13)$$

On the other hand, we can assume  $dP_{\text{CCD}}(\rho, \omega, x, y)$  is all contributed from photons emitted from one triangle element  $\Gamma$ . Therefore a coordinate transformation can be conducted on (5.13) to express  $dP_{\text{CCD}}(\rho, \omega, x, y)$  with  $(\lambda_1, \lambda_2, \varphi, \theta)$  in  $\Gamma$ ,

$$dP_{\text{CCD}}(\rho, \omega, x, y) = q_{\text{CCD}}(F_{\Gamma}(\lambda_1, \lambda_2, \varphi, \theta)) \left| \frac{\partial F_{\Gamma}(\lambda_1, \lambda_2, \varphi, \theta)}{\partial(\lambda_1, \lambda_2, \varphi, \theta)} \right| d\lambda_1 d\lambda_2 d\varphi d\theta, \quad (5.14)$$

where  $\partial F_{\Gamma}(\lambda_1, \lambda_2, \varphi, \theta)/\partial(\lambda_1, \lambda_2, \varphi, \theta)$  is the Jacobian matrix of the coordinate transformation,  $|\cdot|$  represents the absolute value of the determinate of a matrix. In short, (5.14) is also written as

$$dP_{\text{CCD}}(\rho, \omega, x, y) = q_{\text{CCD}}(F_{\Gamma}(\lambda_1, \lambda_2, \varphi, \theta)) |\nabla F_{\Gamma}| d\lambda_1 d\lambda_2 d\varphi d\theta \quad (5.15)$$

In (5.15),  $q_{\text{CCD}}(F_{\Gamma}(\lambda_1, \lambda_2, \varphi, \theta)) |\nabla F_{\Gamma}|$  on the right-hand can also interpreted as the power density under coordinates  $(\lambda_1, \lambda_2, \varphi, \theta)$ , which should be identical to  $q_{\text{surf}}(\lambda_1, \lambda_2, \varphi, \theta)$  in (5.7). Thus, we have

$$\begin{cases} q_{\text{surf}}(\lambda_1, \lambda_2, \varphi, \theta) = q_{\text{CCD}}(F_{\Gamma}(\lambda_1, \lambda_2, \varphi, \theta)) |\nabla F_{\Gamma}|, & \text{in } \Gamma \\ q_{\text{CCD}}(\rho, \omega, x, y) = q_{\text{surf}}(G_{\Gamma}(\rho, \omega, x, y)) |\nabla G_{\Gamma}| \end{cases} \quad (5.16)$$

Since  $q_{\text{surf}}$ 's values are known according to (5.8) if the intensity distribution  $\psi(\mathbf{r}, \mathbf{s})$  is derived, (5.16) provides an analytical solution for  $q_{\text{CCD}}(\rho, \omega, x, y)$  in (5.12).

#### 5.2.4. Numerical algorithm for measurement operator

In non-contact RTE-DOT, the detector reading of the  $i$ th pixel centered at  $(x_i, y_i)$  on the CCD chip is given by

$$M_i \approx A_{\text{pixel}} \int_0^{2\pi} \int_0^{R_{\text{lens}}} \bar{q}_{\text{CCD}}(\rho, \theta, x_i, y_i) d\rho d\omega \approx A_{\text{pixel}} \Delta\rho \Delta\omega \sum_{j=1}^{N_{\rho}} \sum_{k=1}^{N_{\omega}} \bar{q}_{\text{CCD}}(\rho_j, \omega_k, x_i, y_i), \quad (5.17)$$



where  $A_{\text{pixel}}$  represents the area of a pixel on the CCD chip,  $\{\rho_j\}_{j=1}^{N_\rho}$  and  $\{\omega_k\}_{k=1}^{N_\omega}$  are uniform discretization point sets for  $[0, R_{\text{CCD}}]$  and  $[0, 2\pi)$ , with  $\Delta\rho$  and  $\Delta\omega$  as the step size respectively,  $\bar{q}_{\text{CCD}}(\rho, \theta, x, y)$  is the extended energy density function which is given by

$$\bar{q}_{\text{CCD}}(\rho, \theta, x, y) := q_{\text{CCD}}(\rho, \theta, x, y)1_{S_{\text{CCD}}}(\rho, \theta, x, y) = q_{\text{surf}}(G_\Gamma(\rho, \omega, x, y))|\nabla G_\Gamma|1_{S_{\text{CCD}}}(\rho, \theta, x, y) \quad (5.18)$$

where  $1_{S_{\text{CCD}}}(\rho, \theta, x, y)$  is the indicator function that returns 1 if  $(\rho, \theta, x, y) \in S_{\text{CCD}}$  and returns 0 otherwise. In (5.18),  $\Gamma$  and  $G_\Gamma(\rho, \omega, x, y)$  can be obtained by the backward ray-tracing technique that tracks the photon reversely from the CCD chip to the object surface, then  $|\nabla G_\Gamma|$  can be estimated with the perturbation method.

In this model, one can notice that the detector readings in the non-contact RTE-base DOT system is a linear function with respect to the light intensity distribution in the object. In the discretized model, the light intensity is only given on the  $j$ th solid angle in the  $i$ th control volume, which is represented with  $\psi_{i,j}$  ( $1 \leq i \leq N_{\text{CV}}, 1 \leq j \leq N_{\text{SA}}$ ). In practice, one usually use a vector  $\psi := (\psi_{1,1}, \psi_{1,2}, \dots, \psi_{1,N_{\text{SA}}}, \dots, \psi_{N_{\text{CV}},1}, \dots, \psi_{N_{\text{CV}},N_{\text{SA}}})^T$  to save all the  $\psi_{i,j}$ s and a vector  $M := (M_1, M_2, \dots, M_{N_{\text{pixel}}})^T$  to represent readings from the pixels on the CCD chip, where  $N_{\text{pixel}}$  is the total pixel number used for measurement collection. Then the model for light propagation in free space can be represented with

$$M = Q\psi, \quad (5.19)$$

where  $Q$  is a sparse matrix with its row indices corresponding the pixels on the CCD chip and its column indices corresponding the control volumes and solid angles. Matrix  $Q$  is also referred to as the measurement operator, which is a mandatory component in optimization problem (4.1).

This model provides a way to derive  $Q$  with (5.17), (5.18), (5.8) and (5.9), the algorithm is summarized as Algorithm 5.1.

---

**Algorithm 5.1: Back ray-tracing algorithm for  $Q$ 's construction**

---

1. Discretize  $[0, R_{\text{CCD}}]$  and  $[0, 2\pi)$  uniformly with  $\{\rho_j\}_{j=1}^{N_\rho}$  and  $\{\omega_k\}_{k=1}^{N_\omega}$  with step size equal to  $\Delta\rho$  and  $\Delta\omega$ .
  2. Set  $Q$  as a  $N_{\text{pixel}} \times (N_{\text{CV}} N_{\text{SA}})$  matrix with all entries equal to 0.
  3. **for**  $i = 1: N_{\text{pixel}}$   
 Get the coordinates  $(x_i, y_i)$  and the pixel size  $A_i$ .  
**for**  $j = 1: N_\rho$   
**for**  $k = 1: N_\omega$   
 Backtrack the photon with  $(\rho_j, \omega_k, x_i, y_i)$  as its final status.  
**if** this photon hits a triangle element  $\Gamma$  on the object surface
    - Get the three node indices  $i_1, i_2$  and  $i_3$  of the vertices of  $\Gamma$  (the control volume indices in FVM mesh)
    - Get the normal vector  $\mathbf{n}_\Gamma$  and the area  $|\Gamma|$ .
    - Compute  $(\lambda_1, \lambda_2, \varphi, \theta) = G_\Gamma(\rho_j, \omega_k, x_i, y_i)$ , define  $\lambda_3 := 1 - \lambda_1 - \lambda_2$ .
    - Estimate  $|\nabla G_\Gamma|$  with perturbation method.
    - Find the index  $t$  of the solid angle closest to the direction  $\mathbf{s}(\varphi, \theta)$   
 $t = \text{argmin}_m \|\mathbf{s}(\varphi, \theta) - \mathbf{s}_m\|$
    - Compute  $c = 2A_i \Delta\rho \Delta\omega |\nabla G_\Gamma| \sin \theta |\Gamma| (1 - R(\mathbf{s}(\varphi, \theta) \cdot \mathbf{n}_\Gamma))$ .
    - **for**  $l = 1: 3$   
 $Q(i, (i_l - 1)N_{\text{SA}} + t) = Q(i, (i_l - 1)N_{\text{SA}} + t) + \lambda_l c$   
**end for****end if**  
**end for**  
**end for**  
**end for**
  4. Return  $Q$ .
- 

## 5.3. Validation of the back ray-tracing model

### 5.3.1. Validation through analytical solution

In this section, I designed and performed a phantom experiment to test the validity and accuracy of the proposed algorithm. In the experiment, the light propagation in the direct illumination case (no intermediary optical components), which has an analytical solution, is examined. The experiment is set as shown in Fig. 5.3.

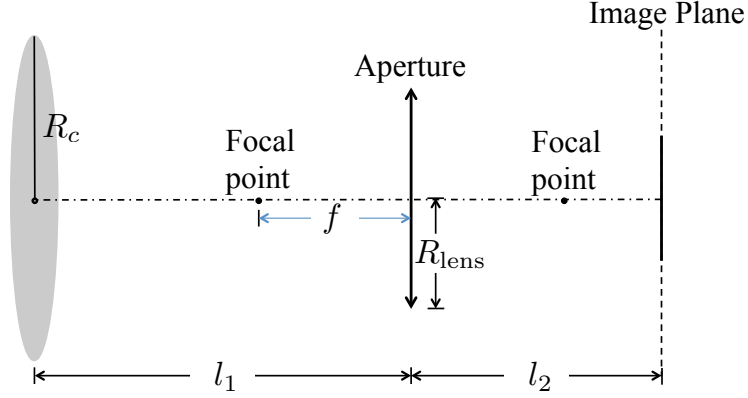


Figure 5.3: Experiment settings for the direct illumination case

A thin round plate with radius  $R_c$  was set perpendicular to the optical axis with the center aligned on the optical axis, beyond the lens with radius  $R_{\text{lens}}$  and focal length  $f$ . The light intensity was set uniformly on the plate as  $\psi$ . The distance from the aperture to the plate and the image plane were denoted by  $l_1$  and  $l_2$  respectively. Furthermore, the focal length  $f$  and two distances  $l_1$  and  $l_2$  are set to follow the equation  $1/f = 1/l_1 + 1/l_2$  in the testing system. Then according to the Cosine fourth Power Law [142], the analytical solution  $M^e$  for the received power per area on the image plane under these settings can be written as:

$$M^e(x, y) = \begin{cases} \frac{\pi\psi R_{\text{lens}}^2 l_1^2 l_2^2}{(l_1^2 + R_{\text{lens}}^2)(l_2^2 + x^2 + y^2)^2}, & x^2 + y^2 \leq \frac{R_c^2 l_2^2}{l_1^2} \\ 0, & x^2 + y^2 > \frac{R_c^2 l_2^2}{l_1^2} \end{cases} \quad (5.20)$$

In the numerical validation,  $\psi$  was set as  $1\text{W/mm}^2/\text{sr}$ ,  $l_1, l_2$  and  $f$  were set as 1050 mm, 52.5mm and 50mm respectively,  $R_{\text{lens}}$  and  $R_c$  were set as 6.25mm and 1050 mm respectively.

For validation purpose, the size of the numerical CCD chip is set to  $105\text{mm} \times 105\text{mm}$ . The CCD chip is discretized by  $501 \times 501$  pixels. The predicted measurement  $M^c$  is computed on every pixel with the light propagation model. The aperture is discretized with  $N_p = 10$  and  $N_\omega = 30$ . The comparison between  $M^e$  and  $M^c$  is shown in Fig. 5.4. No noticeable difference

between the analytical measurement  $M^e$  (Fig. 5.4(a)) and the predicted measurement  $M^c$  (Fig. 5.4(b)) is observed. Fig. 5.4(c) shows the relative error of  $M^c$  is always in range  $[0, 5 \times 10^{-5}]$ . To further quantify the performance of the back ray-tracing algorithm, the correlation factor  $c(M^e, M^c)$  and deviation factor  $d(M^e, M^c)$  of the computed measurement are reported, with the definitions of  $c(M^e, M^c)$  and  $d(M^e, M^c)$  given in (4.26).

$$\begin{cases} c(M^e, M^c) = 1 - 1.4564 \times 10^{-10} \\ d(M^e, M^c) = 3.5114 \times 10^{-5} \end{cases} \quad (5.21)$$

Therefore, the high correlation factor and low deviation factor and relative error indicate that the free space light propagation model has very high accuracy for this case.

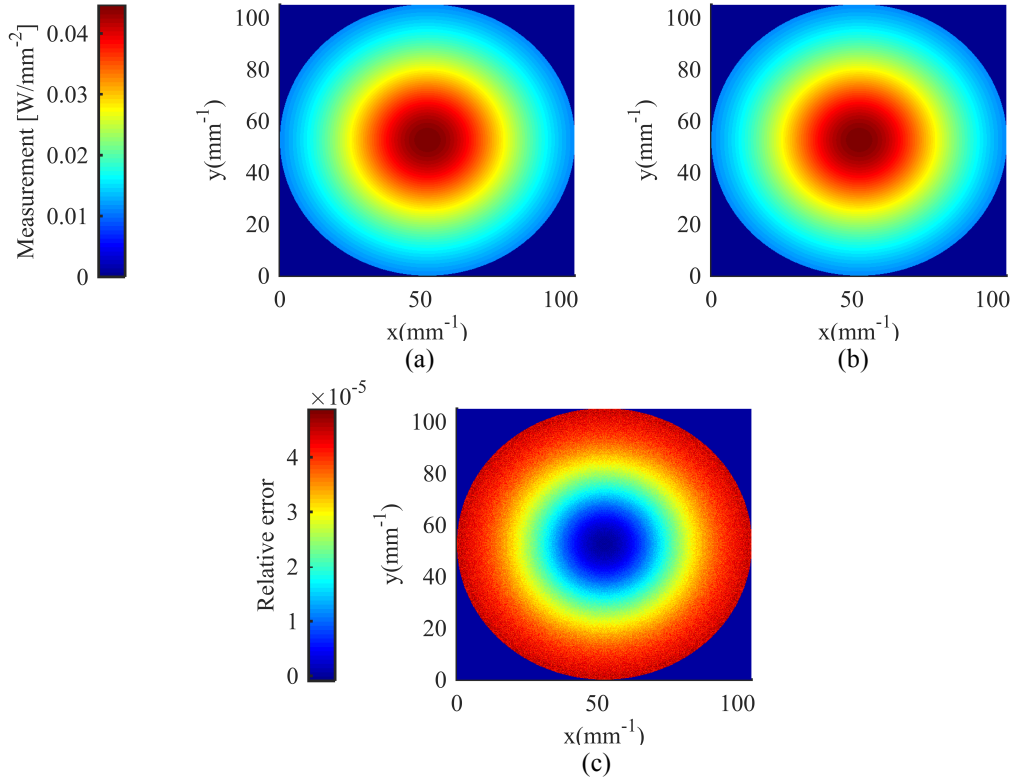


Figure 5.4: Comparison between the exact and the computed measurement on the image plane. (a) The exact measurement; (b) The measurements computed with the back ray-tracing model; (c) The relative error of the computed measurement.

### 5.3.2. Validation through fluorescence molecular tomography result

A double conical mirror system has been designed to capture multi-directional views simultaneously in small animal imaging [143] (See Fig. 5.5). The schematic plot of the system is shown in Fig. 5.6, where the field of view (dot line) by the camera B is equivalent to the field of view by two mirrors and the camera A. The first mirror facing a target animal captures a surface of a target and the second mirror facing a detection camera reflects and projects the captured images by the first mirror onto the detection camera. Depending on the shape of a target, the shape of first and second mirrors can vary like flat or conical or oval or two combined shape. For this system, since the target is a small animal like mice, a conical shape is chose to capture whole body surface of a small animal. The conical mirror size was designed to cover a 40 mm diameter, 80 mm length cylinder, the size of which is enough to cover a small animal.

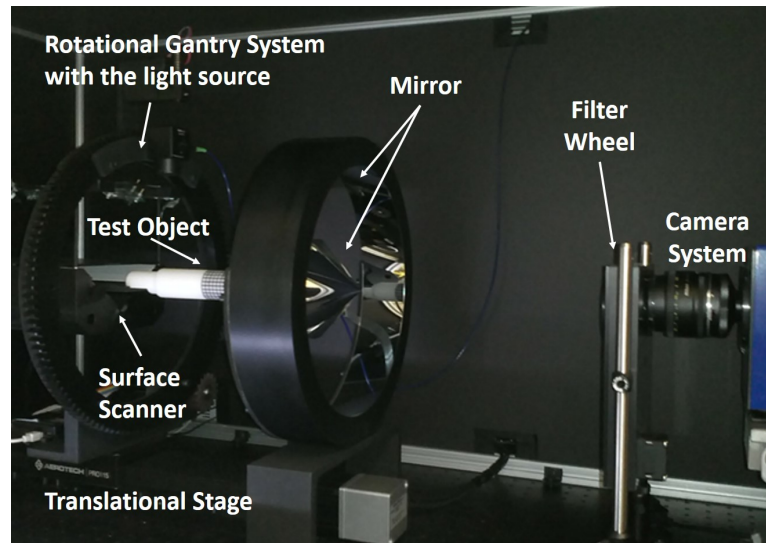


Figure 5.5: A picture of the double conical mirror system

Fig. 5.7 shows the positioning of a coke can as a target in the imaging unit and the taken image by a camera. As shown in the captured image, because of the conical shape of mirrors, there is an image distortion and to do tomographic reconstruction, we need to know the location

on the target and the corresponding direction of the photons in the captured image came from. Therefore with any fixed geometry of the imaging object, the measurement operator that measures the contribution of photons from various surface locations and direction to the image is constructed with the back ray-tracing model that is proposed in the previous section.

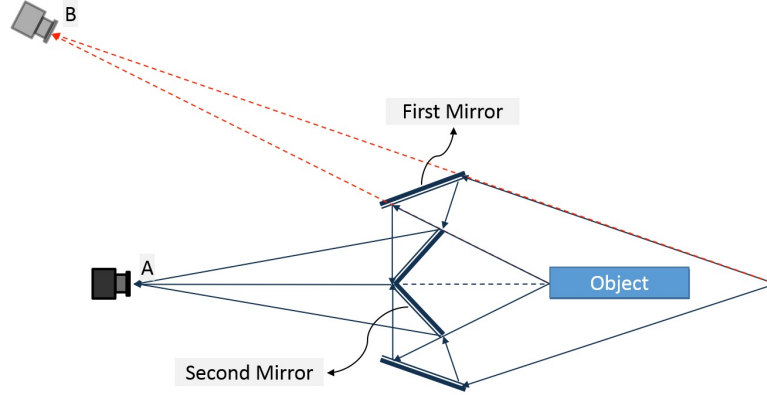


Figure 5.6: Double conical mirror scheme: the first mirror and the second mirror were designed to cover a 40mm diameter 80mm length cylinder target.

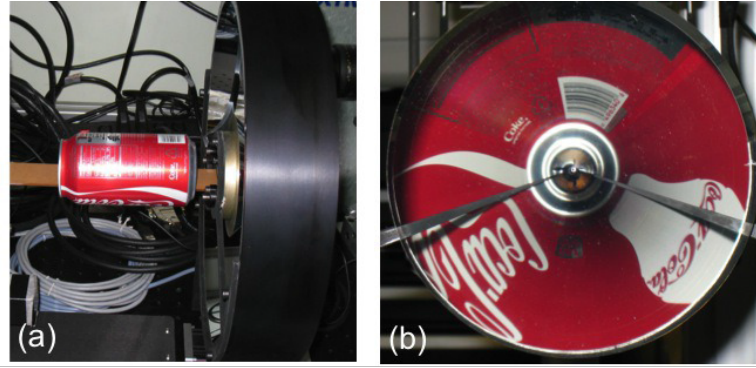


Figure 5.7: A picture of the coke can with the double conical mirror system. (a) Positioning of the coke can in the imaging unit; (b) the captured image on the camera.

A small animal fluorescence imaging experiment was conducted with a tumor-bearing mouse with the double conical mirror imaging system. The tumor cells ( $1 \times 10^6$  cells/ml in 100 $\mu$ L PBS) were injected subcutaneously near the left kidney of a mouse. Then the mouse was imaged one week after to measure the tumor growth with fluorescence molecular tomography (FMT). Details can be found in [144].

With one single illumination on the tumor area, 37558 data points were obtained on the CCD image as the measurements for reconstruction. The reconstructed tumor location is shown in Fig. 5.7, which was confirmed by the results from planar imaging using the Kodak In-vivo Multispectral imaging System FX (Carestream Health, Inc.). Moreover, the proposed model has been already applied for measurement operator derivation in several published work [143-145] and more promising results were achieved. These results evidently support the validity of the back ray-tracing model for light propagation in free space.

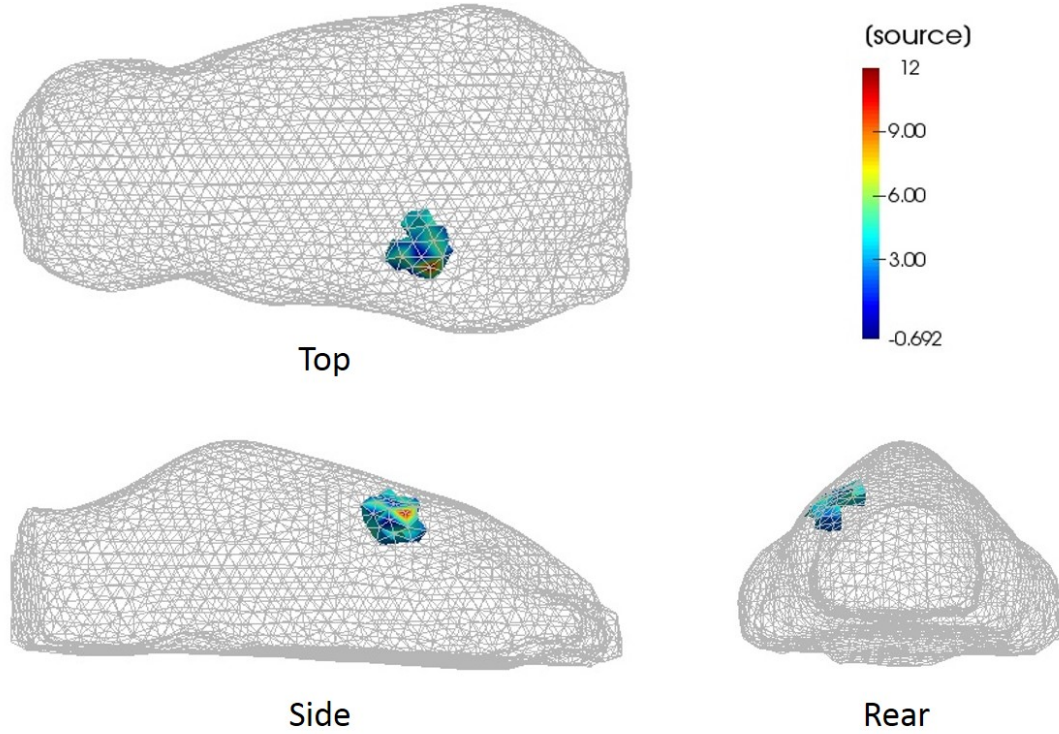


Figure 5.8: Reconstruction results of the tumor growth one week after subcutaneous tumor injection.

## 5.4. Discussion

In this work, a novel free-space, angular-dependent photon transport model was developed for deriving the measurement operator in non-contact DOT imaging system. In this model, the backward ray-tracing method was employed to calculate the contribution of the surface radiation

to each pixel on the CCD chip in an efficient way. For every pixel on the CCD chip, multiple possible final statuses for photons are generated according to the discretization of the aperture and the location of that pixel. Then for each final status, a corresponding pseudo photon is shot reversely into the system and the back ray-tracing technique is applied to locate its initial direction  $\mathbf{s}$  and position  $\mathbf{r}$  on the object surface. Finally the weight of the contribution to this pixel from the local light intensity  $\psi(\mathbf{r}, \mathbf{s})$  is determined with a coordinate transformation and perturbation method. Thus, this model provides a numerical method to calculate the measurement operator that maps linearly from the light intensity distribution of the imaging object to the signals received by the CCD camera, which is a fundamental component in optimization problem setup DOT. Additionally, since the angular dependency of the light intensity is fully considered and only several weak assumptions are imposed on the imaging system, this model can be used in RTE-DOT imaging system with general optical components.

To test this algorithm, firstly a numerical experiment with an analytical solution was considered. The performance of the proposed algorithm was evaluated with correlation factor, deviation factor and relative error. From the results, we can observe that the correlation factor was very close to 1 and the deviation factor was very close to 0, which indicates an extremely strong agreement between the output of the proposed model and the analytical solution, yet indicated the validity and accuracy of the back ray-tracing model. Secondly, the proposed algorithm has been applied to provide the measurement operator for a double conical mirror imaging system. Promising results including source reconstruction of the phantom study and small animal fluorescence imaging have provided strong evidence for its validity [143, 145].



To sum up, this model will aid image reconstruction by providing angular dependent measurements without expensive computation and further makes RTE-based non-contact DOT with complex optical components realistic in practice.

## Chapter 6. Algorithm evaluation with clinical data

In this chapter I discuss a performance of the proposed algorithm as applied to clinical data obtained for finger joints. Optical properties ( $\mu_a$  and  $\mu'_s$ ) of human finger joints are reconstructed with the reconstruction algorithm as described in Chapter 4. Then feature extraction and classification are conducted on the reconstructions results. The performance of the proposed algorithm is evaluated in terms of diagnostic accuracy (i.e., sensitivity and specificity) as compared to the traditional reconstruction algorithm.

### 6.1. Clinical data

In this study, the optical properties of finger proximal interphalangeal (PIP) joints are reconstructed. The clinical data set includes 33 unhealthy subjects with various stage of rheumatoid arthritis (RA) and 20 healthy control subjects [7]. Each subject was evaluated by a rheumatologist and diagnosed for RA according to guidelines set by the ACR [146]. The clinically dominant hand of each subject was imaged with ultrasound and low-field MRI.

The ultrasound and MRI images were evaluated by a radiologist and a rheumatologist in a double blind review. The images were evaluated for the presence of effusion, synovitis, and erosion in PIP joints II to IV. Each reviewer classified each subject into one of five sub-groups on the basis of these findings (Table 1). A third reviewer served as a tiebreaker in cases where the initial reviewers had differing opinions (none in this study). Subjects without signs of joint effusion, synovitis, and erosion were divided into two subgroups: (1) subjects with RA and (2) subjects without RA.

Table 6.1: Diagnostic table based on clinical evaluation and radiological imaging (ultrasound and MRI).

Group	Effusion	Erosion	Synovitis	RA
A	No	No	No	Yes
B	Yes	No	No	Yes
C	No	Yes	No	Yes
D	No	No	Yes	Yes
E	Yes	Yes	Yes	Yes
H	No	No	No	No

Imaging with a frequency-domain DOT sagittal laser scanner of PIP joints II to IV was performed on the clinically dominant hand of subjects with RA and on both hands of the control group. A frequency-modulated laser beam (670 nm, 8 mW, 1.0 mm diameter) scanned the dorsal side of the finger from proximal to distal end, stopping at 11 discrete locations to allow for data acquisition. Transillumination was recorded from each source position on the ventral side of the finger with an intensified CCD camera (See an example in Fig. 6.1). The 3-D geometry of the scanned finger was obtained with a separate laser-scanning unit (650 nm, 5 mW, 0.2 mm line width). Imaging was performed at 0, 300, and 600 MHz. In total, 219 fingers were imaged. The 600 MHz transillumination measurements were used to reconstruct tissue  $\mu_a$  and  $\mu'_s$  coefficients with the new RTE-based algorithm as described in Chapter 4.

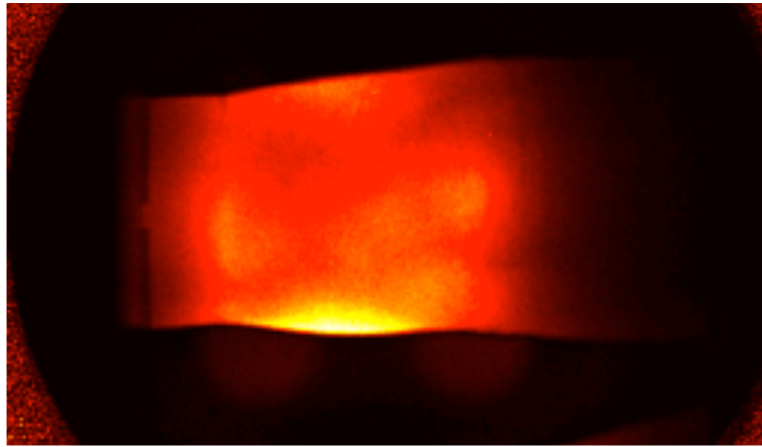


Figure 6.1: Transillumination captured by a CCD camera during a scan of PIP joint II.

To perform image reconstruction, the geometry of each finger is discretized with about 7500 nodes and over 30000 tetrahedral grids (See 4.18(a)). The locations of 11 sources and about

150 selected detectors are mapped into the surface nodes on the discretized mesh (See 6.2(b)(c)). The angular domain is discretized with  $S_4$ , which contains 24 solid angles. The zeroth-order Delta-Eddington approximation (3.13) is used in the forward problem for reconstruction. The reconstructions are run on an Intel i7-4960HQ processor with 16GB DDR3 memory.

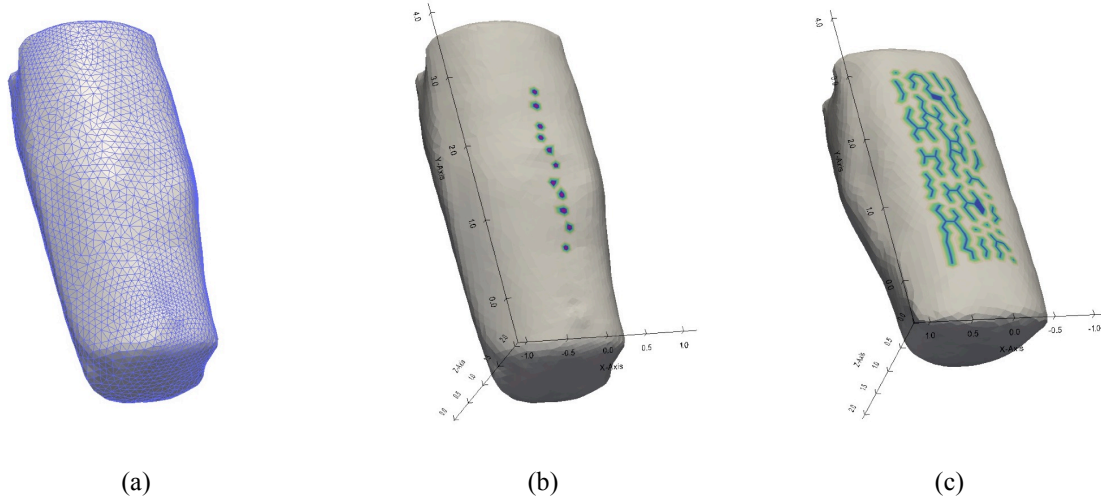


Figure 6.2: Discretization of a sample PIP joint and locations of sources and detectors. (a) Tetrahedral mesh of a finger that contains 7424 nodes and 32207 tetrahedrons. (b) Locations of 11 sources on the dorsal surface of the joint (c) Locations of 145 detectors on the palmar surface of the joint.

## 6.2. Reconstruction, classification and performance comparison

### 6.2.1. Reconstruction results

A volumetric distribution of  $\mu_a$  and  $\mu'_s$  within a given finger is obtained for each subject. Examples of cross sections of  $\mu_a$  and  $\mu'_s$  are shown in Fig. 6.3. The most pronounced differences between joints of subjects affected by RA and of subjects not affected by RA occur at the center of the images, where the joint cavity is located. As expected, in healthy joints, both  $\mu_a$  and  $\mu'_s$  are significantly lower in this region than in the surrounding tissues, due to the lower  $\mu_a$  and  $\mu'_s$  of the synovial fluid that fills the joint cavity. Joints affected by RA typically do not show a strong drop in optical properties in these regions.

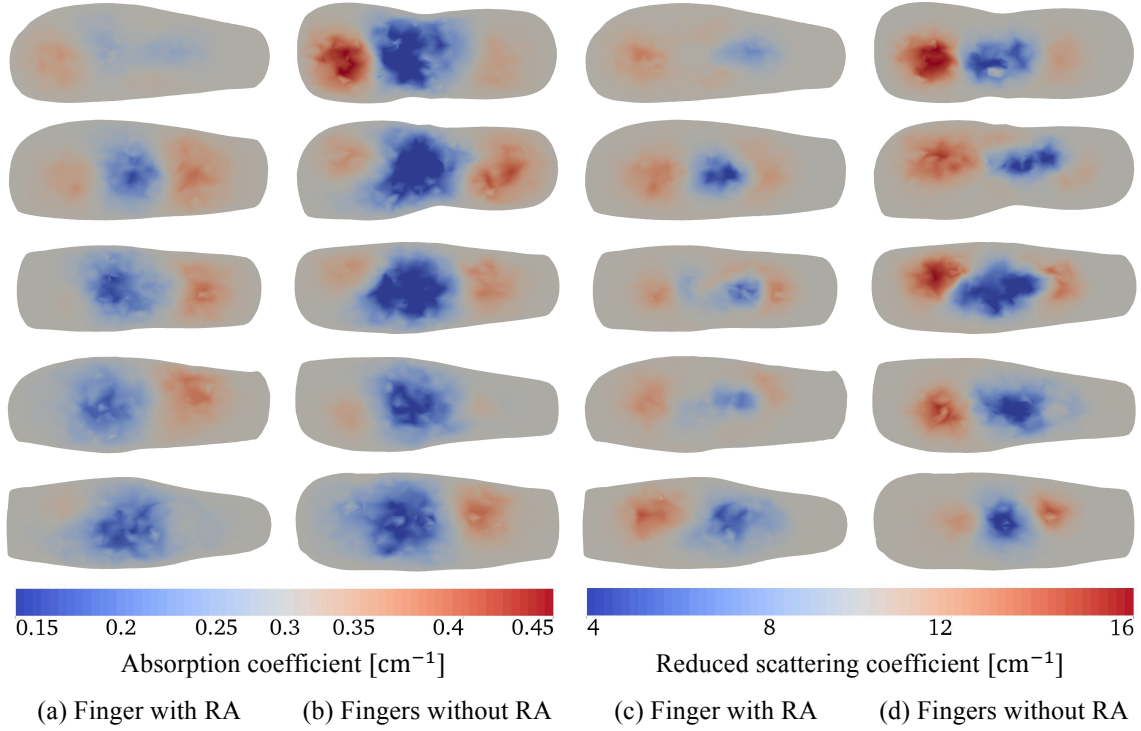


Figure 6.3: Absorption (a and b) and reduced scattering (c and d) coronal cross sections of PIP joints from subjects with RA (a and c) and without RA (b and d).

However, diagnosis on the basis of visual inspection alone may mislead false positive or missed diagnose. Therefore, I explore feature extraction and classification to enhance diagnostic accuracy.

## 6.2.2. Classification on the reconstruction results

### 6.2.2.1. Feature extraction

Features are extracted separately from the 3D  $\mu_a$  and  $\mu'_s$  reconstruction images. To reduce negative impacts by artifacts present in the 3D reconstructed image, the highest 5% and lowest 5% of the reconstructed values are removed from the data set. Then the maximum, minimum, mean, variance, and the ratio of the minimum to maximum of each data set are calculated as features. Therefore 10 features in total are obtained for each finger. The mean, standard deviation (std) and the p-value from the student t-test were calculated for all these features for all the fingers

with and without RA. The summary is presented in Table 6.2. All the features show significant difference ( $p < 0.05$ ) between the affected group and the healthy control group except for  $\min(\mu'_s)$ , with  $p = 0.064 > 0.05$ . This is consistent with what has been reported in [7].

Table 6.2: Mean values, standard deviations and p-values from the t-test for individual features

Features	Affected Group (mean $\pm$ std)	Healthy Group (mean $\pm$ std)	p-value
$\max(\mu_a)$	$0.3212 \pm 0.0125$	$0.3259 \pm 0.0158$	0.018
$\min(\mu_a)$	$0.2141 \pm 0.0319$	$0.1928 \pm 0.0353$	$6.34 \times 10^{-8}$
$\text{mean}(\mu_a)$	$0.2929 \pm 0.0045$	$0.2916 \pm 0.0048$	0.041
$\text{std}(\mu_a)$	$0.0171 \pm 0.0060$	$0.0210 \pm 0.0070$	$1.53 \times 10^{-5}$
$\text{ratio}(\mu_a)$	$0.6661 \pm 0.0938$	$0.5932 \pm 0.1135$	$6.89 \times 10^{-7}$
$\max(\mu'_s)$	$11.220 \pm 0.4691$	$11.692 \pm 0.7187$	$5.44 \times 10^{-8}$
$\min(\mu'_s)$	$8.5238 \pm 0.5932$	$8.3793 \pm 0.5473$	0.064
$\text{mean}(\mu'_s)$	$10.001 \pm 0.0641$	$10.031 \pm 0.0802$	$3.17 \times 10^{-3}$
$\text{std}(\mu'_s)$	$0.3232 \pm 0.1055$	$0.3966 \pm 0.1232$	$5.22 \times 10^{-6}$
$\text{ratio}(\mu'_s)$	$0.7610 \pm 0.0620$	$0.7193 \pm 0.0620$	$2.22 \times 10^{-6}$

#### 6.2.2.2. Classification results

Previous studies show that multiple features can be combined together to achieve higher performance than single feature classification in this problem [7, 147-149]. However, on the other hand, less feature number is always preferred since the predictive power reduces as the dimensionality increases with a fixed number of training samples, which is also referred as to the Hughes phenomenon [150, 151]. To provide guidance on the number of features that should be used in the classification, the principle component analysis (PCA) [152] was conducted on the normalized feature set. The result shows over 99.5% of the variance can be explained with the first two principle components. In another word, for any feature combination, there must exist at most two features that all other features can be approximately expressed as their linear combination. This suggests to use only one or two features in the classification. Therefore, in this thesis, the performances of all the individual features and two-feature combinations were tested with the linear discriminant analysis (LDA) [153] as the classifier.

To remove the bias that might be introduced from treating each finger as an independent sample, a modified version of the leave-one-out cross-validation (LOOCV) is employed here to train the test. In contrast to the standard LOOCV procedure, where one sample (finger) is used for testing while the remaining samples are used for training, all samples (fingers) belonging to one single subject (three fingers for subjects with RA six fingers for subjects without RA) are left out. The remaining samples are used for training the algorithm. In the testing phase, each of the testing samples is classified as true positive (TP), true negative (TN), false positive (FP), or false negative (FN). This process is repeated for each of the 53 distinct subjects (each repetition is called an iteration). The overall performance of an algorithm is computed by summing the TP, TN, FP, and FN values computed from each of 53 LOOCV iterations. From these results, the sensitivity ( $Se := TP/(TP + FN)$ ), the specificity ( $Sp := TN/(TN + FP)$ ), and the Youden index ( $Y := Se + Sp - 1$ ) are computed for each feature combination. A summary of feature combination with 10 highest Youden indices is presented in Table 6.3. It can be seen that the highest Youden index achieved is 0.66 with  $std(\mu_a)$  and  $max(\mu'_s)$  as the features. Fig. 6.4 show the decision boundary with the features  $std(\mu_a)$  and  $max(\mu'_s)$  as applied to the entire dataset. The feature combination of  $mean(\mu_a)$  and  $max(\mu'_s)$  gives the highest sensitivity 0.93, while  $ratio(\mu_a)$  combined with  $ratio(\mu'_s)$  generates the highest specificity 0.76. The individual feature that gives the best performance on Youden index is  $ratio(\mu_a)$ , with  $Se = 0.86$ ,  $Sp = 0.74$ , and  $Y = 0.60$ . These promising results strongly support that the proposed fast reconstruction algorithm can be used with reliable accuracy for clinical diagnosis.

Table 6.3: Summary of classification results of feature combinations with 10 highest Youden indices

Rank	Feature combination	Sensitivity	Specificity	Youden index
1	$\text{std}(\mu_a), \max(\mu'_s)$	0.92	0.74	<b>0.66</b>
2	$\min(\mu_a), \max(\mu'_s)$	0.91	0.73	0.64
3	$\text{ratio}(\mu_a), \max(\mu'_s)$	0.90	0.74	0.64
4	$\text{mean}(\mu_a), \max(\mu'_s)$	<b>0.93</b>	0.71	0.64
5	$\min(\mu_a), \text{mean}(\mu'_s)$	0.92	0.72	0.64
6	$\text{mean}(\mu_a), \text{ratio}(\mu_a)$	0.88	0.74	0.62
7	$\text{std}(\mu_a), \text{mean}(\mu'_s)$	0.91	0.71	0.62
8	$\text{ratio}(\mu_a), \text{mean}(\mu'_s)$	0.90	0.72	0.62
9	$\text{ratio}(\mu_a), \text{ratio}(\mu'_s)$	0.85	<b>0.76</b>	0.61
10	$\min(\mu_a), \min(\mu'_s)$	0.88	0.72	0.61

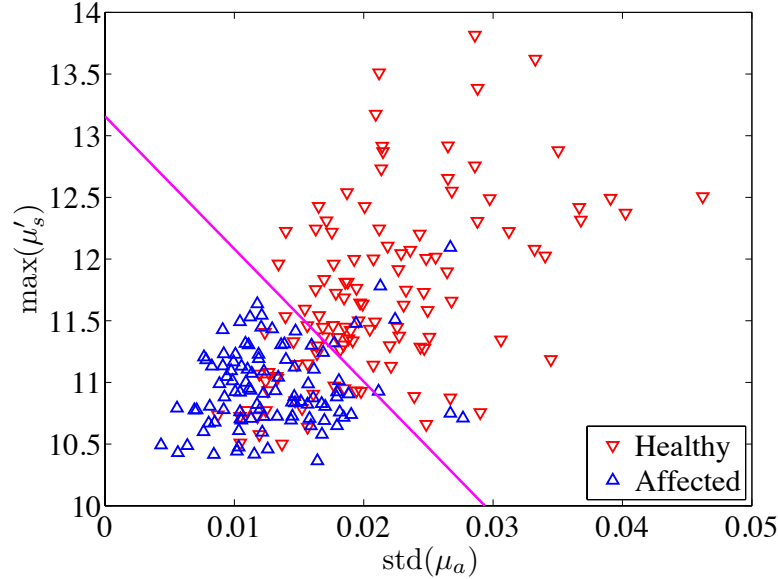


Figure 6.4: Decision boundary of LDA separating affected from healthy data using  $\text{std}(\mu_a)$  and  $\max(\mu'_s)$  as features.

### 6.2.3. Performance comparison

The CPU times between the new reconstruction algorithm and the traditional reconstruction algorithm are reported in this section. To this end, the numerical experiment is conducted on the 3D mesh as obtained for the real finger with 11 sources and 155 detectors. In the new reconstruction algorithm, the line-search-free BFGS is used as the inverse solver and the forward problems in RTE-DOT are solved with the new efficient solver described in Chapter 3. In the



traditional reconstruction algorithm, the single thread sequential BiCGStab algorithm is used for solving the forward problems in RTE-DOT, which are preconditioned with the traditional ILU preconditioner. In both algorithms, the reconstruction was terminated when the relative objective function value becomes below 0.2. The reconstructions of  $\mu_a$  and  $\mu'_s$  from the two algorithms are shown in Fig. 6.5, in which no significant discrepancy is observed in results from different algorithms. Fig. 6.6 shows the convergence history for both algorithms with respect to CPU time. To reduce the relative objective value to 0.2, the traditional reconstruction algorithm takes 3812 seconds, while the new reconstruction algorithm presented here only takes 127 seconds, which is 30 times faster.

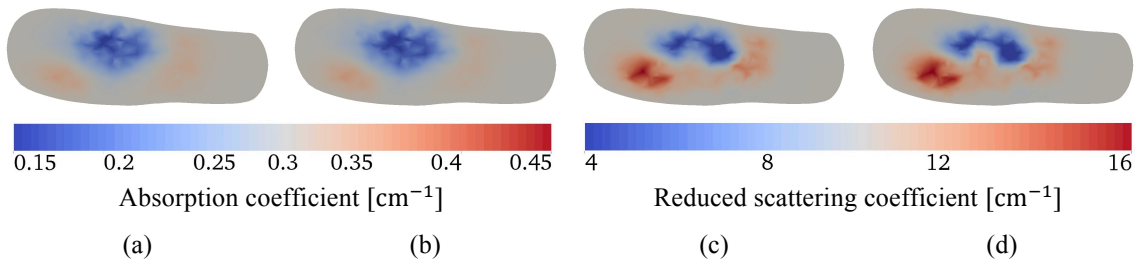


Figure 6.5: Reconstruction results of  $\mu_a$  (a and b) and  $\mu'_s$  (c and d) from the traditional BFGS algorithm (a and c) and the fast reconstruction algorithm (b and d).

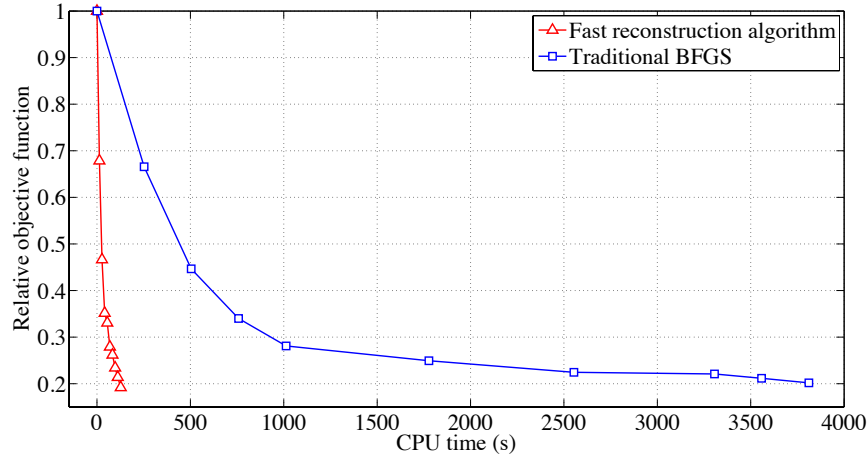


Figure 6.6: Convergence path of relative objective value with respect to the CPU time

### 6.3. Conclusion

In this Chapter, the new reconstruction algorithm has been applied to reconstructions of clinical data obtained for human finger joints. The algorithm is validated and evaluated both with respect to diagnostic performance and CPU times with clinical data, which contains 99 affected joints and 120 healthy ones.

First, clinical performance of the algorithm is evaluated through classification. To this end, all the fingers are reconstructed with the new reconstruction algorithm. Then 10 statistical features (5 for  $\mu_a$  and 5 for  $\mu'_s$ ) are extracted from the reconstruction result of each individual finger. All the individual features and two-feature combinations are evaluated in leave-one-subject-out cross validation with LDA as the classifier. The highest 10 Youden indices achieved are all above 0.6, with the highest one as 0.66. These promising results strongly support that the new reconstruction algorithm provides reliable DOT images for clinical diagnosis with RA.

A CPU time is compared between the new reconstruction algorithm and the traditional reconstruction algorithm as applied to the 3D finger case. Both algorithms are of similar accuracy, but in terms of CPU times the new reconstruction algorithm is faster by a factor 30 than the traditional reconstruction algorithm. In other words, only 2 ~ 3 minutes are taken for one finger reconstruction by the new reconstruction algorithm. It is expected that this significant improvement in CPU time will broadly expand clinical applicability of RTE-DOT.

## Chapter 7. Conclusion

The focus of this dissertation was to develop an efficient reconstruction algorithm for RTE-DOT that can be widely used in practice. To achieve the goal, the developments and the implementations of numerical algorithms are presented that enhance the computational efficiency of the image reconstruction as well as expand the applicability of the RTE-DOT to the more complex imaging system with general optical components.

This dissertation firstly presented a fast forward solver that calculates the light propagation in biological tissue. This solver consists of a preconditioner, a block-linear solver for linear equations with multiple right-hand sides, and two multi-threading acceleration technique. I showed that this code can solve the forward problems in the RTE-DOT up to 25 times faster than traditional single-thread sequential solvers. The code acceleration mainly depends on the optical properties, the number of sources, and the thread number. This acceleration is of great significance, as the efficiency bottleneck comes from the forward solver in many RTE-DOT reconstruction algorithms. Moreover, this forward solver was designed to run on a PC, instead of clusters. So it does not require extensive resource as other parallel computing algorithms do. Finally, except for the preconditioner, the other two techniques in this forward solver are independent of the numerical discretization scheme or even the forward model. Thus this forward solver is not only limited to RTE-DOT but can be adapted to DE-based DOT problems as well.

Secondly, two optimization schemes, named as the line-search-free BFGS algorithm and the full-space algorithm, have been explored. I used the widely employed BFGS algorithm as benchmark for the validation and performance evaluation of the two algorithms. In numerical studies, I found that the line-search-free BFGS compares favorably with the traditional BFGS

and the full-space algorithm in terms of computational efficiency. It showed about 10% ~ 40% faster speed than the traditional BFGS algorithm due to the improvement of the line search process. The full-space algorithm usually converges very fast in the early stag of the iterative process. However, the convergence rate drops down quickly due to the insufficient preconditioning of the Hessian matrix. Another drawback of this algorithm is that each inverse iteration is much more time consuming. Thus the overall CPU time is much longer when compared to algorithms based on reduced Hessian matrix. Therefore, this algorithm is considered impractical for clinical applications.

Third, I proposed the first light propagation model in free space that fully considers the angular dependency., This imposes only very weak assumptions on the optical system and thus can be used in most image system with general optical components. This model was validated with numerical simulations as well as with a pre-clinical experiment involving a fluorescent mouse. This model is a great asset to the RTE-DOT since it provides researchers with bread flexibility in system design as well as possibilities for potential cost reduction in RTE-based imaging system.

The overall reconstruction algorithm was applied to data obtained in a previous clinical study concerning rheumatoid arthritis in human finger joints. The data set contains measurement data from 99 fingers with RA and 120 healthy fingers joints. Overall 10 features were extracted from the  $\mu_a$  and  $\mu'_s$  reconstruction results. According to the analysis, significant statistical difference between fingers with RA and healthy fingers can be found in 9 out of these 10 features. All the single feature and two-feature combinations were tested in classification with LDA as the classifier. The best Youden index achieved is 0.66 and all the highest 10 Youden indices are above 0.60. For the computational performance aspect, the comparison between the

reconstruction results and CPU times from the fast reconstruction algorithm and the traditional reconstruction technique was conducted. It showed a reconstruction for an individual finger only cost 120~180 seconds on a PC with the reconstruction algorithm, while similar results were generated with the traditional reconstruction algorithm in about one hour.

To sum up, a reconstruction algorithm in RTE-DOT has been develop in this dissertation. A speedup factor about 30 as compared to the traditional reconstruction technique in RTE-DOT was achieved. Furthermore, this algorithm can provide enough support for more advanced designs and foreseeably make promising contribution to the RTE-DOT imaging system. Therefore, one can expect the work in this dissertation to be widely applied in RTE-DOT and yet broadly expand its applicability DOT in clinical use.

## Reference

1. A. H. Hielscher, R. E. Alcouffe, and R. L. Barbour, "Comparison of finite-difference transport and diffusion calculations for photon migration in homogeneous and heterogeneous tissues," *Phys Med Biol* **43**, 1285-1302 (1998).
2. O. Dorn, "A transport-backtransport method for optical tomography," *Inverse Probl* **14**, 1107-1130 (1998).
3. A. D. Klose and A. H. Hielscher, "Iterative reconstruction scheme for optical tomography based on the equation of radiative transfer," *Med Phys* **26**, 1698-1707 (1999).
4. K. Ren, G. S. Abdoulaev, G. Bal, and A. H. Hielscher, "Algorithm for solving the equation of radiative transfer in the frequency domain," *Opt Lett* **29**, 578-580 (2004).
5. H. H. Andreas, E. A. Raymond, and L. B. Randall, "Comparison of finite-difference transport and diffusion calculations for photon migration in homogeneous and heterogeneous tissues," *Phys Med Biol* **43**, 1285 (1998).
6. A. D. Klose, U. Netz, J. Beuthan, and A. H. Hielscher, "Optical tomography using the time-independent equation of radiative transfer — Part 1: forward model," *Journal of Quantitative Spectroscopy and Radiative Transfer* **72**, 691-713 (2002).
7. A. H. Hielscher, H. K. Kim, L. D. Montejo, S. Blaschke, U. J. Netz, P. A. Zwaka, G. Illing, G. A. Muller, and J. Beuthan, "Frequency-Domain Optical Tomographic Imaging of Arthritic Finger Joints," *Ieee T Med Imaging* **30**, 1725-1736 (2011).
8. L. D. Montejo, "Computational Methods For The Diagnosis of Rheumatoid Arthritis With Diffuse Optical Tomography," ProQuest Dissertations and Theses, (2014).
9. X. Chen, X. Gao, X. Qu, D. Chen, X. Ma, J. Liang, and J. Tian, "Generalized free-space diffuse photon transport model based on the influence analysis of a camera lens diaphragm," *Appl Optics* **49**, 5654-5664 (2010).
10. X. Chen, X. Gao, X. Qu, J. Liang, L. Wang, D. a. Yang, A. Garofalakis, J. Ripoll, and J. Tian, "A study of photon propagation in free-space based on hybrid radiosity-radiance theorem," *Opt Express* **17**, 16266-16280 (2009).
11. J. Ripoll, R. B. Schulz, and V. Ntziachristos, "Free-space propagation of diffuse light: Theory and experiments," *Phys Rev Lett* **91**(2003).
12. C. Habermehl, C. Schmitz, and J. Steinbrink, "Contrast-enhanced diffuse optical tomography of brain perfusion in humans using ICG," *Photonic Therapeutics and Diagnostics VIII, Pts 1 and 2* **8207**(2012).
13. E. E. Vidal-Rosas, S. A. Billings, Y. Zheng, J. E. Mayhew, D. Johnston, A. J. Kennerley, and D. Coca, "Reduced-order modeling of light transport in tissue for real-time monitoring of brain hemodynamics using diffuse optical tomography," *J Biomed Opt* **19**(2014).
14. A. T. Eggebrecht, S. L. Ferradal, A. Robichaux-Viehoever, M. S. Hassanpour, H. Dehghani, A. Z. Snyder, T. Hershey, and J. P. Culver, "Mapping distributed brain function and networks with diffuse optical tomography," *Nat Photonics* **8**, 448-454 (2014).

15. Z. J. Lin, L. Li, M. Cazzell, and H. L. Liu, "Monitoring of human brain functions in risk decision-making task by diffuse optical tomography using voxel-wise general linear model," *Photonic Therapeutics and Diagnostics Ix* **8565**(2013).
16. F. H. Tian, H. J. Niu, B. Khan, G. Alexandrakis, K. Behbehani, and H. L. Liu, "Enhanced Functional Brain Imaging by Using Adaptive Filtering and a Depth Compensation Algorithm in Diffuse Optical Tomography," *Ieee T Med Imaging* **30**, 1239-1251 (2011).
17. A. Custo and Boas, "Anatomical atlas-guided diffuse optical tomography of brain activation," *NeuroImage (Orlando, Fla.)* **49**, 561-567 (2010).
18. A. T. Eggebrecht, A. T. Eggebrecht, S. L. Ferradal, A. Robichaux-Viehoever, and M. S. Hassanpour, "Mapping distributed brain function and networks with diffuse optical tomography," *Nat Photonics* **8**, 448-454 (2014).
19. M. L. Flexman, H. K. Kim, J. E. Gunther, E. A. Lim, M. C. Alvarez, E. Desperito, K. Kalinsky, D. L. Hershman, and A. H. Hielscher, "Optical biomarkers for breast cancer derived from dynamic diffuse optical tomography," *J Biomed Opt* **18**(2013).
20. M. L. Flexman, M. A. Khalil, R. Al Abdi, H. K. Kim, C. J. Fong, E. Desperito, D. L. Hershman, R. L. Barbour, and A. H. Hielscher, "Digital optical tomography system for dynamic breast imaging," *J Biomed Opt* **16**(2011).
21. S. M. W. Y. van de Ven, S. G. Elias, A. J. Wiethoff, M. van der Voort, T. Nielsen, B. Brendel, C. Bontus, F. Uhlemann, R. Nachabe, R. Harbers, M. van Beek, L. Bakker, M. B. van der Mark, P. Luijten, and W. P. T. M. Mali, "Diffuse optical tomography of the breast: preliminary findings of a new prototype and comparison with magnetic resonance imaging," *Eur Radiol* **19**, 1108-1113 (2009).
22. J. Wang, S. D. Jiang, Z. Z. Li, R. M. diFlorio-Alexander, R. J. Barth, P. A. Kaufman, B. W. Pogue, and K. D. Paulsen, "In vivo quantitative imaging of normal and cancerous breast tissue using broadband diffuse optical tomography," *Med Phys* **37**, 3715-3724 (2010).
23. M. S. Xiao, Y. X. Jiang, Q. L. Zhu, S. S. You, J. C. Li, H. Y. Wang, X. J. Lai, J. Zhang, H. Liu, and J. Zhang, "Diffuse Optical Tomography of Breast Carcinoma: Can Tumor Total Hemoglobin Concentration be Considered as a New Promising Prognostic Parameter of Breast Carcinoma?," *Acad Radiol* **22**, 439-446 (2015).
24. Q. L. Zhu, M. S. Xiao, S. S. You, J. Zhang, Y. X. Jiang, X. J. Lai, and Q. Dai, "Ultrasound-guided diffuse optical tomography (DOT) of invasive breast carcinoma: Does tumour total haemoglobin concentration contribute to the prediction of axillary lymph node status?," *Eur J Radiol* **81**, 3185-3189 (2012).
25. V. Ntziachristos, A. G. Yodh, M. Schnall, and B. Chance, "Concurrent MRI and diffuse optical tomography of breast after indocyanine green enhancement," *P Natl Acad Sci USA* **97**, 2767-2772 (2000).
26. M. A. Khalil, H.-K. K. Kim, I.-K. Kim, R. Dayal, and A. H. Hielscher, "Detecting lower extremity vascular dynamics in patients with peripheral artery disease using diffuse optical tomography," in 2011), 78962Z-78962Z-78966.
27. M. A. Khalil, H. K. Kim, I. K. Kim, M. Flexman, R. Dayal, G. Shrikhande, and A. H. Hielscher, "Dynamic diffuse optical tomography imaging of peripheral arterial disease," *Biomed Opt Express* **3**, 2288-2298 (2012).

28. M. A. Khalil, H. K. Kim, M. Flexman, and I.-K. Kim, "Measurement of vascular response within the foot using dynamic diffuse optical tomography," (2012), p. BTu2A.4.
29. J. W. Hoi, M. A. Khalil, H. K. Kim, and A. Hielscher, "Contact-free diffuse optical tomography system for dynamic imaging of peripheral vasculature," in *Biomedical Optics*, (Optical Society of America, 2014), BM3A. 72.
30. M. A. Khalil, "Development of a Vascular Optical Tomographic Imaging System for the Diagnosis and Monitoring of Peripheral Arterial Disease," Ph.D. (Columbia University, Ann Arbor, 2014).
31. J. W. Hoi, H. K. Kim, M. A. Khalil, C. J. Fong, A. Marone, G. Shrikhande, and A. H. Hielscher, "Non-contact continuous-wave diffuse optical tomographic system to capture vascular dynamics in the foot," in 2015), 931312-931312-931315.
32. A. H. Hielscher, "Optical tomographic imaging of small animals," *Curr Opin Biotech* **16**, 79-88 (2005).
33. G. Gulsen, O. Birgul, M. B. Unlu, R. Shafiiha, and O. Nalcioglu, "Combined diffuse optical tomography (DOT) and MRI system for cancer imaging in small animals," *Technol Cancer Res T* **5**, 351-363 (2006).
34. Y. T. Lin, D. Thayer, O. Nalcioglu, and G. Gulsen, "Tumor characterization in small animals using magnetic resonance-guided dynamic contrast enhanced diffuse optical tomography," *J Biomed Opt* **16**(2011).
35. J. Chamorro-Servent, J. Aguirre, J. Ripoll, J. J. Vaquero, and M. Desco, "Feasibility of U-curve method to select the regularization parameter for fluorescence diffuse optical tomography in phantom and small animal studies," *Opt Express* **19**, 11490-11506 (2011).
36. E. Lapointe, J. Pichette, and Y. Bérubé-Lauzière, "A multi-view time-domain non-contact diffuse optical tomography scanner with dual wavelength detection for intrinsic and fluorescence small animal imaging," *Rev Sci Instrum* **83**, 063703 (2012).
37. A. G. James, R. A. B. Hector, F. Jon, B. S. Iain, and D. Hamid, "Multi-modal molecular diffuse optical tomography system for small animal imaging," *Measurement Science and Technology* **24**, 105405 (2013).
38. U. J. Netz, J. Beuthan, and A. H. Hielscher, "Multipixel system for gigahertz frequency-domain optical imaging of finger joints," *Rev Sci Instrum* **79**(2008).
39. Z. Yuan, Q. Zhang, E. S. Sobel, and H. Jiang, "Tomographic x-ray-guided three-dimensional diffuse optical tomography of osteoarthritis in the finger joints," *J Biomed Opt* **13**, 044006-044006-044010 (2008).
40. Z. Yuan, Q. Zhang, E. Sobel, and H. Jiang, "Three-dimensional diffuse optical tomography of osteoarthritis: initial results in the finger joints," *J Biomed Opt* **12**, 034001-034001-034011 (2007).
41. A. Scheel, M. Backhaus, A. Klose, B. Moa-Anderson, U. Netz, K. Hermann, J. Beuthan, G. Muller, G. Burmester, and A. Hielscher, "First clinical evaluation of sagittal laser optical tomography for detection of synovitis in arthritic finger joints," *Annals of the Rheumatic Diseases* **64**, 239-245 (2005).
42. L. G. Henyey and J. L. Greenstein, "Diffuse radiation in the galaxy," *Astrophys J* **93**, 70-83 (1941).



43. S. R. Arridge, "Optical tomography in medical imaging," *Inverse Probl* **15**, R41-R93 (1999).
44. D. A. Boas, C. Pitris, and N. Ramanujam, "Handbook of biomedical optics," (CRC Press,, Boca Raton, 2011).
45. A. H. Hielscher, A. Y. Bluestone, G. S. Abdoulaev, A. D. Klose, J. Lasker, M. Stewart, U. Netz, and J. Beuthan, "Near-infrared diffuse optical tomography," *Dis Markers* **18**, 313-337 (2002).
46. H. Gao and H. K. Zhao, "A Fast-Forward Solver of Radiative Transfer Equation," *Transport Theor Stat* **38**, 149-192 (2009).
47. H. Gao, L. Phan, and Y. T. Lin, "Parallel multigrid solver of radiative transfer equation for photon transport via graphics processing unit," *J Biomed Opt* **17**(2012).
48. G. S. Abdoulaev, K. Ren, and A. H. Hielseher, "Optical tomography as a PDE-constrained optimization problem," *Inverse Probl* **21**, 1507-1530 (2005).
49. M. Schweiger, S. R. Arridge, and I. Nissila, "Gauss-Newton method for image reconstruction in diffuse optical tomography," *Phys Med Biol* **50**, 2365-2386 (2005).
50. T. Feng, P. Edstrom, and M. Gulliksson, "Levenberg-Marquardt methods for parameter estimation problems in the radiative transfer equation," *Inverse Probl* **23**, 879-891 (2007).
51. T. Tarvainen, M. Vauhkonen, and S. R. Arridge, "Gauss-Newton reconstruction method for optical tomography using the finite element solution of the radiative transfer equation," *J Quant Spectrosc Ra* **109**, 2767-2778 (2008).
52. H. K. Kim and A. H. Hielscher, "A PDE-constrained SQP algorithm for optical tomography based on the frequency-domain equation of radiative transfer," *Inverse Probl* **25**(2009).
53. W. M. Han, J. G. Huang, and J. A. Eichholz, "Discrete-Ordinate Discontinuous Galerkin Methods for Solving the Radiative Transfer Equation," *Siam J Sci Comput* **32**, 477-497 (2010).
54. M. Kindelan, F. Bernal, P. González-Rodríguez, and M. Moscoso, "Application of the RBF meshless method to the solution of the radiative transport equation," *J Comput Phys* **229**, 1897-1908 (2010).
55. H. Gao and H. Zhao, "ANALYSIS OF A NUMERICAL SOLVER FOR RADIATIVE TRANSPORT EQUATION," *Math Comput* **82**, 153-172 (2013).
56. X. J. Gu, H. K. Kim, J. Masciotti, and A. H. Hielscher, "A Parallel Reduced-Space Sequential-Quadratic Programming Algorithm for Frequency-domain Small Animal Optical Tomography," *Proc Spie* **7174**(2009).
57. V. Simoncini and E. Gallopoulos, "An Iterative Method for Nonsymmetric Systems with Multiple Right-Hand Sides," *Siam J Sci Comput* **16**, 917-933 (1995).
58. R. W. Freund and M. Malhotra, "A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides," *Linear Algebra Appl* **254**, 119-157 (1997).
59. H. Calandra, S. Gratton, R. Lago, X. Vasseur, and L. M. Carvalho, "A Modified Block Flexible Gmres Method with Deflation at Each Iteration for the Solution of Non-Hermitian Linear Systems with Multiple Right-Hand Sides," *Siam J Sci Comput* **35**, S345-S367 (2013).
60. T. Sakurai, H. Tadano, and Y. Kuramashi, "Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD," *Comput Phys Commun* **181**, 113-117 (2010).

61. J. V. Lambers, "Application of Block Krylov Subspace Spectral Methods to Maxwell's Equations," *Aip Conf Proc* **1174**, 1-15 (2009).
62. A. Stathopoulos and K. Orginos, "Computing and Deflating Eigenvalues While Solving Multiple Right-Hand Side Linear Systems with an Application to Quantum Chromodynamics," *Siam J Sci Comput* **32**, 439-462 (2010).
63. M. H. Gutknecht, "Block Krylov space methods for linear systems with multiple right-hand sides: an introduction," (2006).
64. A. El Guennouni, K. Jbilou, and H. Sadok, "The block Lanczos method for linear systems with multiple right-hand sides," *Appl Numer Math* **51**, 243-256 (2004).
65. A. El Guennouni, K. Jbilou, and H. Sadok, "A block version of BiCGSTAB for linear systems with multiple right-hand sides," *Electron T Numer Ana* **16**, 129-142 (2003).
66. Y. Saad, "ILUT: A dual threshold incomplete LU factorization," *Numerical Linear Algebra with Applications* **1**, 387-402 (1994).
67. Y. Saad, "ILUM: A multi-elimination ILU preconditioner for general sparse matrices," *Siam J Sci Comput* **17**, 830-847 (1996).
68. M. Pakzad, J. L. Lloyd, and C. Phillips, "Independent columns: A new parallel ILU preconditioner for the PCG method," *Parallel Comput* **23**, 637-647 (1997).
69. E. H. Ayachour, "Expanded systems and the ILU preconditioner for solving non-Hermitian linear systems," *Linear Algebra Appl* **293**, 243-256 (1999).
70. E. F. F. Botta and F. W. Wubs, "Matrix renumbering ILU: An effective algebraic multilevel ILU preconditioner for sparse matrices," *Siam J Matrix Anal A* **20**, 1007-1026 (1999).
71. M. J. Gander and F. Nataf, "An incomplete LU preconditioner for problems in acoustics," *J Comput Acoust* **13**, 455-476 (2005).
72. K. M. Eickhoff and W. L. Engl, "Levelized Incomplete Lu Factorization and Its Application to Large-Scale Circuit Simulation," *Ieee T Comput Aid D* **14**, 720-727 (1995).
73. Y. T. Tsai, J. F. Dai, and M. K. Tsai, "An improved levelized incomplete LU method and its application to 2D semiconductor device simulation," *J Chin Inst Eng* **24**, 389-396 (2001).
74. A. D. Klose and A. H. Hielseher, "Modeling photon propagation in anisotropically scattering media with the equation of radiative transfer," *P Soc Photo-Opt Ins* **4955**, 624-633 (2003).
75. L. D. Montejo, J. F. Jia, H. K. Kim, U. J. Netz, S. Blaschke, G. A. Muller, and A. H. Hielscher, "Computer-aided diagnosis of rheumatoid arthritis with optical tomography, Part 1: feature extraction," *J Biomed Opt* **18**(2013).
76. A. H. Hielscher, A. D. Klose, A. K. Scheel, B. Moa-Anderson, M. Backhaus, U. Netz, and J. Beuthan, "Sagittal laser optical tomography for imaging of rheumatoid finger joints," *Phys Med Biol* **49**, 1147-1163 (2004).
77. B. Turcksin, J. C. Ragusa, and J. E. Morel, "Angular Multigrid Preconditioner for Krylov-Based Solution Techniques Applied to the S-n Equations with Highly Forward-Peaked Scattering," *Transport Theor Stat* **41**, 1-22 (2012).
78. V. E. Henson and U. M. Yang, "BoomerAMG: A parallel algebraic multigrid solver and preconditioner," *Appl Numer Math* **41**, 155-177 (2002).
79. J. C. Meza and R. S. Tuminaro, "A multigrid preconditioner for the semiconductor equations," *Siam J Sci Comput* **17**, 118-132 (1996).

80. J. C. Xu and J. S. Qin, "Some Remarks on a Multigrid Preconditioner," *Siam J Sci Comput* **15**, 172-184 (1994).
81. D. S. Kershaw, "Incomplete Cholesky-Conjugate Gradient Method for Iterative Solution of Systems of Linear Equations," *J Comput Phys* **26**, 43-65 (1978).
82. I. E. Kaporin, "Using the modified 2nd order incomplete Cholesky decomposition as the conjugate gradient preconditioning," *Numerical Linear Algebra with Applications* **9**, 401-408 (2002).
83. J. E. Castillo, S. Steinberg, and P. J. Roache, "Mathematical Aspects of Variational Grid Generation .2.," *J Comput Appl Math* **20**, 127-135 (1987).
84. C. B. Allen, "Towards automatic structured multiblock mesh generation using improved transfinite interpolation," *Int J Numer Meth Eng* **74**, 697-733 (2008).
85. T. J. Barth, *Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes* (American Institute of Aeronautics and Astronautics, 1991).
86. Y. Saad and M. H. Schultz, "Gmres - a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear-Systems," *Siam J Sci Stat Comp* **7**, 856-869 (1986).
87. I. E. Rana and N. G. Alexopoulos, "Current Distribution and Input Impedance of Printed Dipoles," *Ieee T Antenn Propag* **29**, 99-105 (1981).
88. H. A. Vandervorst, "Bi-Cgstab - a Fast and Smoothly Converging Variant of Bi-Cg for the Solution of Nonsymmetric Linear-Systems," *Siam J Sci Stat Comp* **13**, 631-644 (1992).
89. R. W. Freund and N. M. Nachtigal, "Qmr - a Quasi-Minimal Residual Method for Non-Hermitian Linear-Systems," *Numer Math* **60**, 315-339 (1991).
90. R. W. Freund, "A Transpose-Free Quasi-Minimal Residual Algorithm for Non-Hermitian Linear-Systems," *Siam J Sci Comput* **14**, 470-482 (1993).
91. A. D. Klose and A. H. Hielscher, "Modeling photon propagation in anisotropically scattering media with the equation of radiative transfer," in 2003), 624-633.
92. Y. Nakamura, K. I. Ishikawa, Y. Kuramashi, T. Sakurai, and H. Tadano, "Modified block BiCGSTAB for lattice QCD," *Comput Phys Commun* **183**, 34-37 (2012).
93. S. R. Arridge and J. C. Schotland, "Optical tomography: forward and inverse problems," *Inverse Probl* **25**(2009).
94. G. A. F. Seber, *A Matrix Handbook for Statisticians* (Wiley-Interscience, 2007), p. 559.
95. H. K. Kim and A. Charette, "Frequency domain optical tomography using a conjugate gradient method without line search," *J Quant Spectrosc Ra* **104**, 248-256 (2007).
96. B. Tavakoli and Q. Zhu, "Two-step reconstruction method using global optimization and conjugate gradient for ultrasound-guided diffuse optical tomography," *J Biomed Opt* **18**, 016006-016006 (2013).
97. M. Schweiger and S. Arridge, "The Toast++ software suite for forward and inverse modeling in optical tomography," *J Biomed Opt* **19**, 040801-040801 (2014).
98. S. Okawa, Y. Hoshi, and Y. Yamada, "Improvement of image quality of time-domain diffuse optical tomography with lp sparsity regularization," *Biomed Opt Express* **2**, 3334-3348 (2011).
99. C. Xu, H. Vavadi, A. Merkulov, H. Li, M. Erfanzadeh, A. Mostafa, Y. Gong, H. Salehi, S. Tannenbaum, and Q. Zhu, "Ultrasound-Guided Diffuse Optical Tomography for Predicting and Monitoring Neoadjuvant Chemotherapy of Breast Cancers Recent Progress," *Ultrasonic imaging*, 0161734615580280 (2015).

100. E. A. Ustinov, "Adjoint sensitivity analysis of radiative transfer equation: temperature and gas mixing ratio weighting functions for remote sensing of scattering atmospheres in thermal IR," *J Quant Spectrosc Ra* **68**, 195-211 (2001).
101. B. Gregory, L. M. Eric, L. Ang, Z. Quan, C. Tina, H. B. Dana, and A. B. David, "Quantitative spectroscopic diffuse optical tomography of the breast guided by imperfect a priori structural information," *Phys Med Biol* **50**, 3941 (2005).
102. T. Zhang, J. Zhou, P. R. Carney, and H. Jiang, "Towards real-time detection of seizures in awake rats with GPU-accelerated diffuse optical tomography," *Journal of Neuroscience Methods* **240**, 28-36 (2015).
103. M. A. O'Leary, D. A. Boas, B. Chance, and A. G. Yodh, "Experimental Images of Heterogeneous Turbid Media by Frequency-Domain Diffusing-Photon Tomography," *Opt Lett* **20**, 426-428 (1995).
104. G. S. Abdoulaev and A. H. Hielscher, "Three-dimensional optical tomography with the equation of radiative transfer," *J Electron Imaging* **12**, 594-601 (2003).
105. L. Spinelli, A. Torricelli, A. Pifferi, P. Taroni, and R. Cubeddu, "Experimental test of a perturbation model for time-resolved imaging in diffusive media," *Appl Optics* **42**, 3145-3153 (2003).
106. K. Ren, G. Bal, and A. H. Hielscher, "Frequency domain optical tomography based on the equation of radiative transfer," *Siam J Sci Comput* **28**, 1463-1489 (2006).
107. J. He, R. Weersink, I. Veilleux, K. Mayo, A. Zhang, D. Piao, A. Alam, J. Trachtenberg, and B. C. Wilson, "Development of transrectal diffuse optical tomography combined with 3D-transrectal ultrasound imaging to monitor the photocoagulation front during interstitial photothermal therapy of primary focal prostate cancer," in 2013), 85781J-85781J-85788.
108. B. Bi, B. Han, W. Han, J. Tang, and L. Li, "Image Reconstruction for Diffuse Optical Tomography Based on Radiative Transfer Equation," *Computational and Mathematical Methods in Medicine* **2015**(2015).
109. H. K. Kim, M. Flexman, D. J. Yamashiro, J. J. Kandel, and A. H. Hielscher, "PDE-constrained multispectral imaging of tissue chromophores with the equation of radiative transfer," *Biomed Opt Express* **1**, 812-824 (2010).
110. H. K. Kim, X. J. Gu, and A. H. Hielscher, "An all-at-once reduced Hessian SQP algorithm for frequency domain optical tomography," *Proc Spie* **7174**(2009).
111. J. Nocedal and S. J. Wright, "Numerical Optimization, Second Edition," Springer Ser Oper Re, 1-664 (2006).
112. G. Biros and O. Ghattas, "Parallel Lagrange--Newton--Krylov--Schur Methods for PDE-Constrained Optimization. Part I: The Krylov--Schur Solver," *Siam J Sci Comput* **27**, 687-713 (2005).
113. G. Biros and O. Ghattas, "Parallel Lagrange--Newton--Krylov--Schur Methods for PDE-Constrained Optimization. Part II: The Lagrange--Newton Solver and Its Application to Optimal Control of Steady Viscous Flows," *Siam J Sci Comput* **27**, 714-739 (2005).
114. E. E. Prudencio, R. Byrd, and X. C. Cai, "Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems," *Siam J Sci Comput* **27**, 1305-1328 (2006).
115. W. C. Davidon, "Variable Metric Method for Minimization," *Siam J Optimiz* **1**, 1-17 (1991).

116. D. Goldfarb, "A Family of Variable-Metric Methods Derived by Variational Means," *Math Comput* **24**, 23-& (1970).
117. H. F. Khalfan, R. H. Byrd, and R. B. Schnabel, "A Theoretical and Experimental Study of the Symmetric Rank-One Update," *Siam J Optimiz* **3**, 1-24 (1993).
118. C. G. Broyden, "A Class of Methods for Solving Nonlinear Simultaneous Equations," *Math Comput* **19**, 557-& (1965).
119. J. E. Dennis and R. B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equations," in *Classics in applied mathematics 16*, (Society for Industrial and Applied Mathematics,, Philadelphia, 1996).
120. L. Armijo, "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives," *Pac J Math* **16**, 1-& (1966).
121. P. Wolfe, "Convergence Conditions for Ascent Methods," *Siam Rev* **11**, 226-& (1969).
122. P. Wolfe, "Convergence Conditions for Ascent Methods .2. Some Corrections," *Siam Rev* **13**, 185-& (1971).
123. A. A. Goldstein, "On Steepest Descent," *Journal of the Society for Industrial and Applied Mathematics. Series A, On control* **3**, 147-151 (1965).
124. H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Second Berkeley Symposium on Mathematical Statistics and Probability (University of California Press, 1951), 481-492.
125. G. Biros and O. Ghattas, "Parallel Newton-Krylov methods for PDE-constrained optimization," in *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, (ACM, Portland, Oregon, USA, 1999), p. 28.
126. A. D. Klose and A. H. Hielscher, "Quasi-Newton methods in optical tomographic image reconstruction," *Inverse Probl* **19**, 387-409 (2003).
127. H. Eda, I. Oda, Y. Ito, Y. Wada, Y. Oikawa, Y. Tsunazawa, M. Takada, Y. Tsuchiya, Y. Yamashita, M. Oda, A. Sassaroli, Y. Yamada, and M. Tamura, "Multichannel time-resolved optical tomographic imaging system," *Rev Sci Instrum* **70**, 3595-3602 (1999).
128. M. Solomon, B. R. White, R. E. Nothdruff, W. Akers, G. Sudlow, A. T. Eggebrecht, S. Achilefu, and J. P. Culver, "Video-rate fluorescence diffuse optical tomography for in vivo sentinel lymph node imaging," *Biomed Opt Express* **2**, 3267-3277 (2011).
129. A. D. Mehta, J. C. Jung, B. A. Flusberg, and M. J. Schnitzer, "Fiber optic in vivo imaging in the mammalian nervous system," *Current Opinion in Neurobiology* **14**, 617-628 (2004).
130. R. B. Schulz, J. Ripoll, and V. Ntziachristos, "Noncontact optical tomography of turbid media," *Opt Lett* **28**, 1701-1703 (2003).
131. H. Meyer, A. Garofaiakis, G. Zacharakis, S. Psycharakis, C. Mamalaki, D. Kioussis, E. N. Economou, V. Ntziachristos, and J. Ripoll, "Noncontact optical imaging in mice with full angular coverage and automatic surface extraction," *Appl Optics* **46**, 3617-3627 (2007).
132. R. B. Schulz, J. Peter, W. Semmler, C. D'Andrea, G. Valentini, and R. Cubeddu, "Comparison of noncontact and fiber-based fluorescence-mediated tomography," *Opt Lett* **31**, 769-771 (2006).

133. R. Elaloufi, S. Arridge, R. Pierrat, and R. Carminati, "Light propagation in multilayered scattering media beyond the diffusive regime," *Appl Optics* **46**, 2528-2539 (2007).
134. H. Dehghani, B. Brooksby, K. Vishwanath, B. W. Pogue, and K. D. Paulsen, "The effects of internal refractive index variation in near-infrared optical tomography: a finite element modelling approach," *Phys Med Biol* **48**, 2713-2727 (2003).
135. L. H. Wang, S. L. Jacques, and L. Q. Zheng, "Mcml - Monte-Carlo Modeling of Light Transport in Multilayered Tissues," *Comput Meth Prog Bio* **47**, 131-146 (1995).
136. S. Kumari and A. K. Nirala, "Study of light propagation in human and animal tissues by Monte Carlo simulation," *Indian J Phys* **86**, 97-100 (2012).
137. M. Atif, A. Khan, and M. Ikram, "Modeling of Light Propagation in Turbid Medium Using Monte Carlo Simulation Technique," *Opt Spectrosc+* **111**, 107-112 (2011).
138. X. L. Chen, X. B. Gao, X. C. Qu, D. F. Chen, X. P. Ma, J. M. Liang, and J. Tian, "Generalized free-space diffuse photon transport model based on the influence analysis of a camera lens diaphragm," *Appl Optics* **49**, 5654-5664 (2010).
139. X. L. Chen, X. B. Gao, X. C. Qu, J. M. Liang, L. Wang, D. A. Yang, A. Garofalakis, J. Ripoll, and J. Tian, "A study of photon propagation in free-space based on hybrid radiosity-radiance theorem," *Opt Express* **17**, 16266-16280 (2009).
140. J. Meseguer, I. Perez-Grande, and A. Sanz-Andres, "Thermal radiation heat transfer," *Woodhead Publ Mech E*, 73-86 (2012).
141. E. W. Weisstein, "Barycentric coordinates," (2003).
142. I. C. Gardner, "Validity of the Cosine 4th Power Law of Illumination," *J Res Nat Bur Stand* **39**, 213-219 (1947).
143. J. H. Lee, H. K. Kim, C. Chandhanayingyong, F. Y. I. Lee, and A. H. Hielscher, "Non-contact small animal fluorescence imaging system for simultaneous multi-directional angular-dependent data acquisition," *Biomed Opt Express* **5**, 2301-2316 (2014).
144. J. H. Lee, "A small animal optical tomographic imaging system with omni-directional, non-contact, angular-resolved fluorescence measurement capabilities," (Columbia University, 2014).
145. J. H. Lee, H. K. Kim, J. F. Jia, C. Fong, and A. H. Hielscher, "A Fast full-body fluorescence/bioluminescence imaging system for small animals," *Optical Tomography and Spectroscopy of Tissue X* **8578**(2013).
146. D. Aletaha, T. Neogi, A. J. Silman, J. Funovits, D. T. Felson, C. O. Bingham, N. S. Birnbaum, G. R. Burmester, V. P. Bykerk, M. D. Cohen, B. Combe, K. H. Costenbader, M. Dougados, P. Emery, G. Ferraccioli, J. M. W. Hazes, K. Hobbs, T. W. J. Huizinga, A. Kavanaugh, J. Kay, T. K. Kvien, T. Laing, P. Mease, H. A. Menard, L. W. Moreland, R. L. Naden, T. Pincus, J. S. Smolen, E. Stanislawska-Biernat, D. Symmons, P. P. Tak, K. S. Upchurch, J. Vencovsky, F. Wolfe, and G. Hawker, "2010 Rheumatoid Arthritis Classification Criteria An American College of Rheumatology/European League Against Rheumatism Collaborative Initiative," *Arthritis Rheum-Us* **62**, 2569-2581 (2010).
147. L. D. Montejo, H. K. Kim, Y. Hame, J. F. Jia, J. D. Montejo, U. J. Netz, S. Blaschke, P. Zwaka, G. A. Mueller, J. Beuthan, and A. H. Hielscher, "Analysis and Classification of Optical Tomographic Images of Rheumatoid Fingers with ANOVA and Discriminate Analysis," *Advanced Biomedical and Clinical Diagnostic Systems Ix* **7890**(2011).

148. L. D. Montejo, J. F. Jia, H. K. Kim, U. J. Netz, S. Blaschke, G. A. Muller, and A. H. Hielscher, "Computer-aided diagnosis of rheumatoid arthritis with optical tomography, Part 2: image classification," *J Biomed Opt* **18**(2013).
149. L. D. Montejo, H. K. Kim, J. Jia, and A. H. Hielscher, "Classification of OT Images of Arthritic Joints Using Spatial-Fourier Frequency Coefficients," in *Biomedical Optics and 3-D Imaging*, OSA Technical Digest (Optical Society of America, 2012), BTu3A.38.
150. T. Oommen, D. Misra, N. K. C. Twarakavi, A. Prakash, B. Sahoo, and S. Bandopadhyay, "An objective analysis of Support Vector Machine based classification for remote sensing," *Math Geosci* **40**, 409-424 (2008).
151. G. Hughes, "On the mean accuracy of statistical pattern recognizers," *Information Theory, IEEE Transactions on* **14**, 55-63 (1968).
152. H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 433-459 (2010).
153. C. M. Bishop, *Pattern recognition and machine learning*, Information science and statistics (Springer, New York, 2006), pp. xx, 738 p.

## List or relevant publications

### Journal

**J. Jia**, H. K. Kim, A. H. Hielcher, "Fast linear solver for radiative transport equation with multiple right hand sides in diffuse optical tomography," *J Quant Spectrosc Ra* (submitted)

L. D. Montejo, **J. Jia**, H. K. Kim, U. J. Netz, S. Blaschke, G. A. Muller, and A. H. Hielscher, "Computer-aided diagnosis of rheumatoid arthritis with optical tomography, Part 1: feature extraction," *J Biomed Opt* **18**(2013).

L. D. Montejo, **J. Jia**, H. K. Kim, U. J. Netz, S. Blaschke, G. A. Muller, and A. H. Hielscher, "Computer-aided diagnosis of rheumatoid arthritis with optical tomography, Part 2: image classification," *J Biomed Opt* **18**(2013).

### Abstracts and Proceedings

**J. Jia**, L. Montejo, H. K. Kim, and A. Hielscher, "Optical tomography with the third-order frequency-domain Simplified Spherical approximation (FD-SP3)," in *Biomedical Optics 2014*, OSA Technical Digest (online) (Optical Society of America, 2014), BM3A.71.

**J. Jia**, J. Lee, L. D. Montejo, H. K. Kim, and A. H. Hielscher, "Measurement Operator for Angular Dependent Photon Propagation in Contact-Free Optical Tomography," *Optical Tomography and Spectroscopy of Tissue X* 8578(2013).

L. D. Montejo, **J. Jia**, H. K. Kim, and A. H. Hielscher, "Evaluation of Fourier Transform Coefficients for The Diagnosis of Rheumatoid Arthritis From Diffuse Optical Tomography Images," *Optical Tomography and Spectroscopy of Tissue X* 8578(2013).

J. H. Lee, H. K. Kim, **J. Jia**, C. Fong, and A. H. Hielscher, "A Fast full-body fluorescence / bioluminescence imaging system for small animals," Optical Tomography and Spectroscopy of Tissue X 8578(2013).

L. D. Montejo, H. K. Kim, **J. Jia**, and A. H. Hielscher, "Classification of OT Images of Arthritic Joints Using Spatial-Fourier Frequency Coefficients," in Biomedical Optics and 3-D Imaging, OSA Technical Digest (Optical Society of America, 2012), BTu3A.38.

L. D. Montejo, H. K. Kim, Y. Hame, **J. Jia**, J. D. Montejo, U. J. Netz, S. Blaschke, P. Zwaka, G. A. Mueller, J. Beuthan, and A. H. Hielscher, "Analysis and Classification of Optical Tomographic Images of Rheumatoid Fingers with ANOVA and Discriminate Analysis," Advanced Biomedical and Clinical Diagnostic Systems Ix 7890(2011).



## Appendix A: Implementation details in inverse solvers

We provide the implementation details for the line-search-free BFGS algorithm and the full-space algorithm in appendix A. The original constrained optimization problem we consider is (4.1), where  $\mu = (\mu_a, \mu'_s)$  and  $A(\mu) = A_1 + A_2(\mu_a) + A_3(\mu_s) + i\omega/v I^{(N)}$  with detail definition in (3.12). The regularization term  $R(\mu)$  is given as

$$R(\mu) = \frac{1}{2} t_{\mu_a} \|\mu_a - \mu_a^*\|^2 + \frac{1}{2} t_{\mu'_s} \|\mu'_s - \mu'^{*}_s\|^2, \quad (\text{A.1})$$

where  $t_{\mu_a}$  and  $t_{\mu'_s}$  are regularization parameters,  $\mu_a^*$  and  $\mu'^{*}_s$  are the prior belief on the background  $\mu_a$  and  $\mu'_s$  in the imaging object.

In the optimization problem, we often need to consider vectors with length  $N = N_{\text{SA}} N_{\text{CV}}$ , with the its  $k$ th  $N_{\text{SA}}$  consecutive entries representing values corresponding to the  $N_{\text{SA}}$  solid angles in the  $k$ th control volume. Therefore we define the following symbol for a vector  $x$  with length  $N$ ,

$$x^{(k)} = (x_{(k-1)N_{\text{SA}}+1}, x_{(k-1)N_{\text{SA}}+2}, \dots, x_{kN_{\text{SA}}})^T. \quad (\text{A.1})$$

With the above definition, we define an operator “ $*$ ” that projects two complex vectors  $x$  and  $y$  with length  $N$  to a real vector with length  $N_{\text{CV}}$ ,

$$x * y = \left( \text{Re}(\bar{x}^{(1)} \cdot y^{(1)}), \text{Re}(\bar{x}^{(2)} \cdot y^{(2)}), \dots, \text{Re}(\bar{x}^{(N_{\text{CV}})} \cdot y^{(N_{\text{CV}})}) \right), \quad (\text{A.2})$$

where “ $\cdot$ ” represents the inner product of two vectors,  $\bar{v}$  represents the conjugate of a complex vector,  $\text{Re}(\cdot)$  is a function on a complex value that returns the real part of it.

## A.1. Implementation details in the line-search-free BFGS algorithm

### A.1.1. Gradient of the objective function

The gradient  $g$  of the objective function in the unconstrained optimization is given by

$$g = \nabla_{\mu} R - (A(\mu)^{-T} Q^T (Q\Psi - M))^T (\nabla_{\mu} A(\mu)) \Psi \quad (\text{A.3})$$

It can be obtained by the following steps:

1. Compute  $b = Q^T (Q\Psi - M)$ .
2. Solve the linear equation  $A(\mu)^T \lambda = b$ .
3. Set  $v$  as a length  $N_{CV}$  vector with all entries equal to 1.
4. Compute  $g_{\mu_a} = t_{\mu_a}(\mu_a - \mu_a^*) - \lambda * (A_2(v)\Psi)$ .
5. Compute  $g_{\mu'_s} = t_{\mu'_s}(\mu'_s - \mu'^*_s) - \lambda * (A_3(v)\Psi)$ .
6.  $g = -\left(g_{\mu_a}^T, g_{\mu'_s}^T\right)^T$

For  $\mu_a$  or  $\mu'_s$  reconstruction only, the gradient corresponding to the other optical property is set to 0 after all the steps above.

### A.1.2. Projected descent direction on the forward variable

Once the descent direction  $\Delta\mu$  is found, the projected descent direction  $\Delta\Psi$  on the forward variable is given by

$$\Delta\Psi = -A(\mu)^{-1} \left( (\nabla_{\mu} A(\mu) \cdot \Delta\mu) \Psi \right) \quad (\text{A.5})$$

It can be obtained by the following steps:

1. Decompose  $\Delta\mu = (\Delta\mu_a^T, \Delta\mu'_s{}^T)^T$ .
2. Compute  $b = (A_2(\Delta\mu_a) + A_3(\Delta\mu'_s))$ .
3. Solve linear function  $A(\mu)\Delta\Psi = -b$  to obtain  $\Delta\Psi$ .

### A.1.2. Solution for the optimization problem for step size determination

The following optimization problem needs to be solved in order to determine the step size after  $\Delta\mu$  and  $\Delta\Psi$  are obtained,

$$\min_{\alpha} \frac{1}{2} \|Q(\Psi + \alpha\Delta\Psi) - M\|_F^2 + R(\mu + \alpha\Delta\mu) \quad (\text{A.6})$$

This optimization problem can be solved with the following steps:

1. Decompose  $\Delta\mu = (\Delta\mu_a^T, \Delta\mu_s'^T)^T$ ,  $\mu = (\mu_a^T, \mu_s'^T)^T$ .
2. Compute  $a = \|Q\Delta\Psi\|_F^2 + t_{\mu_a}\|\Delta\mu_a\|^2 + t_{\mu_s'}\|\Delta\mu_s'\|^2$ .
3. Compute  $b = \text{Re}((Q\Delta\Psi) \cdot (Q\Psi - M)) + t_{\mu_a}\Delta\mu_a \cdot (\mu_a - \mu_a^*) + t_{\mu_s'}\Delta\mu_s' \cdot (\mu_s' - \mu_s'^*)$ .
4. Set  $\alpha = -b/2a$ .

## A.2. Implementation details of the Newton step in the full-space algorithm

The Newton step in the full-space algorithm that is described in (4.21) and revisit below:

$$\begin{bmatrix} L_{\Psi\Psi} & L_{\Psi\mu} & C_{\Psi}^T \\ L_{\mu\Psi} & L_{\mu\mu} & C_{\mu}^T \\ C_{\Psi} & C_{\mu} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\Psi \\ \Delta\mu \\ \Delta\lambda \end{Bmatrix} = - \begin{Bmatrix} f_{\Psi} + C_{\Psi}^T\lambda \\ f_{\mu} + C_{\mu}^T\lambda \\ C \end{Bmatrix}. \quad (\text{A.7})$$

With a decomposition of  $\mu$ , the above equation is converted to

$$\begin{bmatrix} L_{\Psi\Psi} & L_{\Psi\mu_a} & L_{\Psi\mu_s'} & C_{\Psi}^T \\ L_{\mu_a\Psi} & L_{\mu_a\mu_a} & L_{\mu_a\mu_s'} & C_{\mu_a}^T \\ L_{\mu_s'\Psi} & L_{\mu_s'\mu_a} & L_{\mu_s'\mu_s'} & C_{\mu_s'}^T \\ C_{\Psi} & C_{\mu_a} & C_{\mu_s'} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\Psi \\ \Delta\mu_a \\ \Delta\mu_s' \\ \Delta\lambda \end{Bmatrix} = - \begin{Bmatrix} f_{\Psi} + C_{\Psi}^T\lambda \\ f_{\mu_a} + C_{\mu_a}^T\lambda \\ f_{\mu_s'} + C_{\mu_s'}^T\lambda \\ C \end{Bmatrix}. \quad (\text{A.8})$$

We define  $v$  as a length  $N_{CV}$  vector with all entries equal to 1. The first three terms on right-hand side of (A.8) is calculated as below (The fourth entry is already given in (4.9)):

$$\begin{cases} f_\Psi + C_\Psi^T \lambda = Q^T(Q\Psi - M) \\ f_{\mu_a} + C_{\mu_a}^T \lambda = t_{\mu_a}(\mu_a - \mu_a^*) + \lambda * (A_2(v)\Psi) \\ f_{\mu'_s} + C_{\mu'_s}^T \lambda = t_{\mu'_s}(\mu'_s - \mu'^*_s) + \lambda * (A_3(v)\Psi) \end{cases} \quad (\text{A.9})$$

To solve with Krylov subspace algorithm, we need to define the inner product for vectors and matrix-vector multiplication in (A.8). The inner product of two vectors  $(\Psi_1, \mu_{a,1}, \mu'_{s,1}, \lambda_1)$  and  $(\Psi_2, \mu_{a,2}, \mu'_{s,2}, \lambda_2)$  is defined as

$$(\Psi_1, \mu_{a,1}, \mu'_{s,1}, \lambda_1) \cdot (\Psi_2, \mu_{a,2}, \mu'_{s,2}, \lambda_2) := \text{Re}(\bar{\Psi}_1 \cdot \Psi_2 + \bar{\lambda}_1 \cdot \lambda_2) + \mu_{a,1} \cdot \mu_{a,2} + \mu'_{s,1} \cdot \mu'_{s,2} \quad (\text{A.10})$$

We give the expression of matrix-vector multiplication for every non-zero block in the matrix on the left hand side of (A.8) as below,

$$\begin{cases} L_{\Psi\Psi}x &= Q^T Qx \\ L_{\Psi\mu_a}y &= A_2(y)^T \lambda \\ L_{\Psi\mu'_s}y &= A_3(y)^T \lambda \\ C_\Psi^T x &= A(\mu)x \\ L_{\mu_a\Psi}x &= \lambda * (A_2(v)x) \\ L_{\mu_a\mu_a}y &= t_{\mu_a} I^{(N_{CV})} \\ L_{\mu_a\mu'_s}y &= 0 \\ C_{\mu_a}^T x &= \lambda * (A_2(v)x) \\ L_{\mu'_s\Psi}x &= \lambda * (A_3(v)x) \\ L_{\mu'_s\mu_a}y &= 0 \\ L_{\mu'_s\mu'_s}y &= t_{\mu'_s} I^{(N_{CV})} \\ C_{\mu'_s}^T x &= \lambda * (A_3(v)x) \\ C_\Psi x &= A(\mu)x \\ C_{\mu_a} y &= A_2(y)\Psi \\ C_{\mu'_s} y &= A_3(y)\Psi \end{cases} \quad (\text{A.11})$$

where  $x$  and  $y$  are vectors with length  $N$  and  $N_{CV}$  respectively,  $I^{(N_{CV})}$  is a  $N_{CV} \times N_{CV}$  identity matrix.