Redbird – a comprehensive model-based image reconstruction platform for diffuse optical tomography and microwave tomography

Qianqian Fang

Version: 0.5

Chapter 1

History

Redbird is an open-source MATLAB/GNU Octave toolbox designed for solving the forward and inverse problems for diffuse optical tomography (DOT), with the additional support of several other non-linear model-based imaging modalities. Redbird was ported from Redbird90 [1] – a FORTRAN90 based DOT image reconstruction software written by the same author, Dr. Qianqian Fang, between 2005 and 2008. The FORTRAN90-based Redbird90 was derived from an earlier implementation, code-named "WhiteDragon" [2, 3], also developed by Dr. Fang, between 2001 and 2005 as part of his PhD dissertation [4], designed to perform iterative 3-D microwave tomography (MWT) image reconstructions [5]. The majority of the functionalities in WhiteDragon, including the 3-D finite element method (FEM) for forward modeling [2], Gauss-Newton based nonlinear inverse solver [3], a multi-right-hand-side quasi-minimal residual (BL-QMR) [6] sparse linear solver, the adjoint method [2] for building Jacobians, and a dual-mesh reconstruction approach [7], had been inherited to Redbird90, and subsequently ported to Redbird. To extend the modality from MWT to DOT, Redbird further introduced multi-spectral DOT reconstruction, derived from the multi-spectral MWT reconstruction algorithm developed in Dr. Fang's thesis [5], a compositional-prior-guided DOT reconstruction [8, 9], as well as a simultaneous source-detector (SD) coupling coefficient estimation [10, 11].

Both names "Redbird" and "WhiteDragon" were rooted in ancient Chinese astrology. In ancient China, astrologists partitioned the sky into 4 quadrants – each named after an animal deity that the constellations resemble: East – pale/white dragon, West – white tiger, North – black turtle, and South – red bird. The name "Redbird" was selected because DOT uses red and near-infrared light for imaging.

This document summarizes the mathematical treatment for the forward and

inverse problems used in this algorithm.

Chapter 2

Mathematical models

2.1 DOT forward model

The diffusion equation in the time-domain can be expressed as [12]

$$-\nabla \cdot D(r)\nabla\Phi(r,t) + \mu_a(r)\Phi(r,t) + \frac{1}{c}\frac{\partial\Phi(r,t)}{\partial t} = S_0(r,t) \tag{2.1}$$

where $D(r) = \frac{1}{3(\mu_a+\mu_s')}$ is the diffusion coefficient (unit cm); $c = \frac{c_0}{n}$ is the speed of light in the medium (unit cm/s); $S_0(r,t)$ is the source. With assumed time dependence $\exp(j\omega t)$, the frequency-domain (FD) diffusion equation can be written as

$$-\nabla \cdot D(r)\nabla\Phi(r) + \left(\mu_a(r) + \frac{j\omega}{c}\right)\Phi(r) = S_0(r) \tag{2.2}$$

where ω is the angular frequency and $\Phi(r)$ is the phasor of $\Phi(r,t)$.

Integrating both sides of (2.2) with a set of weight functions $\varphi_j(r)(j = 1, ..., M)$ where M is the total number of basis functions, over the forward space Ω, subsequently, applying the following vector identity

$$\nabla \cdot f(r)\vec{g}(r) = f(r)\nabla \cdot \vec{g}(r) + \vec{g}(r) \cdot \nabla f(r), \tag{2.3}$$

we get

$$\begin{aligned} -\int_\Omega (D(r)\nabla^2\Phi(r))\varphi_j(r)dr \quad &- \quad \int_\Omega \nabla D(r) \cdot \nabla\Phi(r)\varphi_j(r)dr \\ &+ \quad \int_\Omega \left(\mu_a(r) + \frac{j\omega}{c}\right)\varphi_j(r)\Phi(r)dr = \int_\Omega S_0(r)\varphi_j(r)dr. \end{aligned} \tag{2.4}$$

Assume $D(r)$ is constant for each forward element, i.e. element-based properties, we have $\nabla D(r) = 0$ and the second term in (2.4) becomes zero. However, if we define the optical properties on the nodes using a reconstruction mesh, the second term will be non-zero. In this case, we expand $D(r)$ and $\mu_a(r)$ by piecewise linear basis functions

$$D(r) = \sum_i D_i \phi_i(r) \mu_a(r) = \sum_i \mu_{a_i} \phi_i(r) \tag{2.5}$$

as well as the Green's first identity

$$\int_\Omega u(r) \nabla^2 v(r) dr = -\int_\Omega \nabla u(r) \cdot \nabla v(r) dr + \int_{\partial\Omega} u(r) \nabla v(r) dr \tag{2.6}$$

we can rewrite (2.2) as

$$\begin{aligned} \int_\Omega D(r) \nabla \varphi_j(r) \cdot \nabla \Phi(r) dr \quad &- \quad \int_{\partial\Omega} D(r) \varphi_j(r) \nabla \Phi(r) \cdot d\hat{r} \\ &+ \quad \int_\Omega \left(\mu_a(r) + \frac{j\omega}{c} \right) \varphi_j(r) \Phi(r) dr = \int_\Omega S_0(r) \varphi_j(r) dr \end{aligned} \tag{2.7}$$

With Galerkin's method, i.e. the basis function is identical as the weight function, $\Phi(r)$ is expanded as $\Phi(r) = \sum_{i=1}^4 \Phi_i \varphi_i(r)$ over each linear Largarange forward element, and Equ. (2.7) becomes

$$\begin{aligned} &\sum_i \Phi_i \left(\left\langle D(r) \nabla \varphi_i(r) \cdot \nabla \varphi_j(r) \right\rangle - \left\langle D(r) \nabla \varphi_i(r) \varphi_j(r) \right\rangle_{\partial\Omega} \right. \\ &\left. + \left\langle \left(\mu_a(r) + \frac{j\omega}{c} \right) \varphi_i(r) \varphi_j(r) \right\rangle \right) = \left\langle S_0(r) \varphi_j(r) \right\rangle \end{aligned} \tag{2.8}$$

where $\langle u(r) \rangle$ denotes $\int_\Omega u(r) dr$ and $\langle u(r) \rangle_{\partial\Omega}$ denotes $\int_{\partial\Omega} u(r) dr$.

Expanding $\mu_a(r)$ and $D(r)$ on the parameter mesh basis as $\mu_a(r) = \sum_{k=1}^4 \mu_a^k \phi_k(r)$ and $D(r) = \sum_{k=1}^4 D_k \phi_k(r)$, we get the Galerkin weak form equation as

$$\begin{aligned} &\sum_i \Phi_i \left(\sum_k D_k \left\langle \phi_k(r) \nabla \varphi_i(r) \cdot \nabla \varphi_j(r) \right\rangle - \sum_k D_k \left\langle \phi_k(r) \nabla \varphi_i(r) \varphi_j(r) \right\rangle_{\partial\Omega} \right. \\ &\left. + \sum_k \left(\mu_a^k + \frac{j\omega}{c} \right) \left\langle \phi_k(r) \varphi_i(r) \varphi_j(r) \right\rangle \right) = \left\langle S_0(r) \varphi_j(r) \right\rangle \end{aligned} \tag{2.9}$$

where ϕ_k is the basis function of the parameter mesh.

2.2 Boundary condition

If the extrapolation boundary condition is used, the boundary integration term in (2.9) is then dropped out, leaving

$$\sum_i \Phi_i \left(\sum_k D_k \left\langle \phi_k(r) \nabla \varphi_i(r) \cdot \nabla \varphi_j(r) \right\rangle \right. \tag{2.10}$$
$$\left. + \sum_k \left(\mu_a^k + \frac{j\omega}{c} \right) \left\langle \phi_k(r) \varphi_i(r) \varphi_j(r) \right\rangle \right) = \left\langle S_0(r) \varphi_j(r) \right\rangle$$

However, it has been shown in Haskell *et al.* [13], the partial current boundary condition is more accurate. In this boundary condition, the fluence satisfies the following condition on the boundary $\partial\Omega$

$$(\mathbf{\Phi} = l_s \nabla \mathbf{\Phi}) \,|_{\partial\Omega} \tag{2.11}$$

where l_s is defined in Eq. 2.4.1 in Haskell, as

$$l_s = \frac{1 + R_{eff}}{1 - R_{eff}} 2D \tag{2.12}$$

and $R_{eff} = \frac{R_\phi + R_j}{2 - R_\phi + R_j}$ is the effective reflection coefficient, and

$$R_\phi = \int_0^{\pi/2} 2 \sin\theta \cos\theta R_{Fresnel(\theta)} d\theta \tag{2.13}$$

$$R_j = \int_0^{\pi/2} 3 \sin\theta \cos^2\theta R_{Fresnel(\theta)} d\theta \tag{2.14}$$

$$R_{Fresnel}(\theta) = \frac{1}{2} \left(\frac{n \cos\theta' - n_{out} \cos\theta}{n \cos\theta' + n_{out} \cos\theta} \right)^2 \tag{2.15}$$

$$+ \frac{1}{2} \left(\frac{n \cos\theta - n_{out} \cos\theta'}{n \cos\theta + n_{out} \cos\theta'} \right)^2 \tag{2.16}$$

Replacing $\nabla\mathbf{\Phi} = \mathbf{\Phi}/l_s$ to the boundary term in (2.9), we have

$$\sum_i \Phi_i \left(\sum_k D_k \left\langle \phi_k(r) \nabla \varphi_i(r) \cdot \nabla \varphi_j(r) \right\rangle - \sum_k \frac{1 - R_{eff}}{2(1 + R_{eff})} \left\langle \phi_k(r) \varphi_j(r) \right\rangle_{\partial\Omega} \right.$$
$$\left. + \sum_k \left(\mu_a^k + \frac{j\omega}{c} \right) \left\langle \phi_k(r) \varphi_i(r) \varphi_j(r) \right\rangle \right) = \left\langle S_0(r) \varphi_j(r) \right\rangle \tag{2.17}$$

and the boundary integration $\langle \phi_k(r)\varphi_j(r)\rangle_{\partial\Omega} = \frac{A}{6}$ if $k = j$ or $\frac{A}{12}$ if $k \neq j$. The general expression for the surface element integration is

$$\int_{\Omega_e} \varphi_1^l \varphi_2^m \varphi_3^n dr = \frac{l!m!n!}{(l+m+n+2)!} 2A_e \tag{2.18}$$

Equation (2.17) is the final formula used in the reconstruction code.

2.3 Widefield illumination

When a light source is located outside of the domain, i.e. a wide-field source, at the boundary, the in-bound flux, J^- is no longer zero, thus, the boundary condition should be modified as

$$(\mathbf{\Phi} - l_s \nabla \mathbf{\Phi})\,|_{\partial\Omega} = J^- \tag{2.19}$$

Thus, we should replace $\nabla\mathbf{\Phi}$ by $(\mathbf{\Phi} - J^-)/l_s$ in (2.9), and we get

$$\begin{aligned} &\sum_i \Phi_i \left(\sum_k D_k \left\langle \phi_k(r) \nabla\varphi_i(r) \cdot \nabla\varphi_j(r) \right\rangle - \sum_k \frac{1 - R_{eff}}{2(1 + R_{eff})} \left\langle \phi_k(r)\varphi_j(r) \right\rangle_{\partial\Omega} \right. \\ &\left. + \sum_k \left(\mu_a^k + \frac{j\omega}{c} \right) \left\langle \phi_k(r)\varphi_i(r)\varphi_j(r) \right\rangle \right) = \\ &\left\langle S_0(r)\varphi_j(r) \right\rangle + \sum_k \frac{J_k^- D_k}{l_s} \langle \phi_k(r)\varphi_j(r)\rangle|_{\partial\Omega} \end{aligned} \tag{2.20}$$

2.4 Nodal and element-based adjoint method for computing the Jacobian matrix

Equation (2.17) for each element written in the matrix form is

$$\mathbf{A\Phi} = \mathbf{b} \tag{2.21}$$

where the element of $\mathbf{A}$ can be written as

$$a_{i,j} = \sum_{i,j\in e} \left(\sum_{k=1}^{4} D_k \left\langle \phi_k(r) \nabla\varphi_i(r) \cdot \nabla\varphi_j(r) \right\rangle + \sum_{k=1}^{4} \left(\mu_a^k + \frac{j\omega}{c} \right) \left\langle \phi_k(r)\varphi_i(r)\varphi_j(r) \right\rangle \right) \tag{2.22}$$

for node-based FEM matrix, and

$$a_{i,j} = \sum_{i,j\in e}\left(D_e\left\langle\nabla\varphi_i(r)\cdot\nabla\varphi_j(r)\right\rangle + \left(\mu_a^e + \frac{j\omega}{c}\right)\left\langle\varphi_i(r)\varphi_j(r)\right\rangle\right) \tag{2.23}$$

for element-based FEM matrix, where D_e and μ_a^e are the diffusion and absorption coefficients of the e-th element.

Taking derivative of μ_a and D on both sides of (2.21), we get

$$\frac{\partial\mathbf{A}}{\partial\mu_a^k}\mathbf{\Phi} = -\mathbf{A}\frac{\partial\mathbf{\Phi}}{\partial\mu_a^k} \tag{2.24}$$

$$\frac{\partial\mathbf{A}}{\partial D_k}\mathbf{\Phi} = -\mathbf{A}\frac{\mathbf{\Phi}}{\partial D^k} \tag{2.25}$$

where the elements of matrix $\frac{\partial\mathbf{A}}{\partial\mu_a^k}$ and $\frac{\partial\mathbf{A}}{\partial D_k}$ are written as

$$k_{i,j} = \frac{\partial a_{i,j}}{\mu_a^k} = \left\langle\phi_k(r)\varphi_i(r)\varphi_j(r)\right\rangle \tag{2.26}$$

$$h_{i,j} = \frac{\partial a_{i,j}}{D_k} = \left\langle\phi_k(r)\nabla\varphi_i(r)\cdot\nabla\varphi_j(r)\right\rangle \tag{2.27}$$

respectively.

Similarly, taking derivatives with respect to D_e and μ_a^e yields

$$k_{i,j} = \frac{\partial a_{i,j}}{\mu_a^e} = \left\langle\varphi_i(r)\varphi_j(r)\right\rangle \tag{2.28}$$

$$h_{i,j} = \frac{\partial a_{i,j}}{D_e} = \left\langle\nabla\varphi_i(r)\cdot\nabla\varphi_j(r)\right\rangle \tag{2.29}$$

respectively. Again, the angular bracket $\langle\cdot\rangle$ represents volume integration inside an element, i.e.

$$\langle f(r)\rangle = \int_{\Omega_e} f(r)dr \tag{2.30}$$

Therefore, the deriatives to μ_a^e can be calculated using this relationship:

$$\int_{\Omega_e}\varphi_1^l\varphi_2^m\varphi_3^n\varphi_4^s dr = \frac{l!m!n!s!}{(l+m+n+s+3)!}6V_e \tag{2.31}$$

As a result, for element-based μ_a Jacobian, we have

$$\mathbf{K}_e = \left[\frac{\partial a_{i,j}}{\mu_a^e}\right] = \sum_e\frac{V_e}{20}\begin{pmatrix}2&1&1&1\\1&2&1&1\\1&1&2&1\\1&1&1&2\end{pmatrix} \tag{2.32}$$

For element-based D Jacobian, we have

$$\begin{aligned}\mathbf{H}_e &= \left[\frac{\partial a_{i,j}}{D^e}\right] \\ &= \sum_e \frac{1}{(6V_e)^2}\begin{pmatrix} \mathbf{C}_1\cdot\mathbf{C}_1 & \mathbf{C}_1\cdot\mathbf{C}_2 & \mathbf{C}_1\cdot\mathbf{C}_2 & \mathbf{C}_1\cdot\mathbf{C}_4 \\ \mathbf{C}_2\cdot\mathbf{C}_1 & \mathbf{C}_2\cdot\mathbf{C}_2 & \mathbf{C}_2\cdot\mathbf{C}_2 & \mathbf{C}_2\cdot\mathbf{C}_4 \\ \mathbf{C}_3\cdot\mathbf{C}_1 & \mathbf{C}_3\cdot\mathbf{C}_2 & \mathbf{C}_3\cdot\mathbf{C}_2 & \mathbf{C}_3\cdot\mathbf{C}_4 \\ \mathbf{C}_4\cdot\mathbf{C}_1 & \mathbf{C}_4\cdot\mathbf{C}_2 & \mathbf{C}_4\cdot\mathbf{C}_2 & \mathbf{C}_5\cdot\mathbf{C}_4 \end{pmatrix}\end{aligned} \tag{2.33}$$

where $\mathbf{C}_i = [a_i, b_i, c_i](i = 1, 4)$ are the linear coefficients in the expressions of the basis functions φ_i as

$$\begin{pmatrix}\varphi_1\\ \varphi_2\\ \varphi_3\\ \varphi_4\end{pmatrix} = \frac{1}{6V_e}\begin{pmatrix} 6V_{01} & a_1 & b_1 & c_1 \\ 6V_{02} & a_2 & b_2 & c_2 \\ 6V_{03} & a_3 & b_3 & c_3 \\ 6V_{04} & a_4 & b_4 & c_4 \end{pmatrix}\begin{pmatrix}1\\ x\\ y\\ z\end{pmatrix} \tag{2.34}$$

Here, the matrix is the inversion of the Jacobian ($\mathbf{J}_e$) of the tetrahedron, i.e.

$$\frac{1}{6V_e}\begin{pmatrix} 6V_{01} & a_1 & b_1 & c_1 \\ 6V_{02} & a_2 & b_2 & c_2 \\ 6V_{03} & a_3 & b_3 & c_3 \\ 6V_{04} & a_4 & b_4 & c_4 \end{pmatrix} = \mathrm{inv}\begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{pmatrix} = \mathrm{inv}(\mathbf{J}_e) \tag{2.35}$$

and the element volume $V_e = \det(\mathbf{J}_e)/6$

For the Jacobian corresponding to the μ_a^e in the e-th element, the element-based Jacobian can be expressed as

$$\begin{aligned} J_{\mu_a}((s,r),e) &= \frac{\partial \Phi_{s,r}}{d\mu_a^e} \\ &= \left(\mathbf{K}_e\mathbf{\Phi}_s^e\right)^T\mathbf{\Phi}_r^e \end{aligned} \tag{2.36}$$

where vectors $\mathbf{\Phi}_s^e$ and $\mathbf{\Phi}_r^e$ are defined by $\mathbf{\Phi}_s^e = \{\Phi_s(\vec{p}_k)\}$, $\mathbf{\Phi}_r^e = \{\Phi_r(\vec{p}_k)\}, (k = 1, 2, 3, 4)$, respectively.

For the Jacobian corresponding to the τ-th μ_a, the nodal adjoint method was applied and the Jacobian for the measurement at the r-th detector with illumination of the s-th source is expressed as

$$\begin{aligned} J_{\mu_a}((s,r),\tau) &= \frac{\partial \Phi_{s,r}}{d\mu_a^\tau} \\ &= \sum_{n\in\Omega_\tau}\left(\frac{\sum_{e\in\Omega_n} V_e}{4}\right)\phi(\vec{p}_n)\Phi_s(\vec{p}_n)\Phi_r(\vec{p}_n) \end{aligned} \tag{2.37}$$

where $\sum_{n\in\Omega_\tau}$ refers to the summation over the forward nodes which fall inside Ω_τ and $\sum_{e\in\Omega_n}$ refers to the summation over the forward elements that share the n-th forward node; V_e is the volume of the element, $\Phi_s(\vec{p}_n)$ and $\Phi_r(\vec{p}_n)$ are the forward field and the adjoint field at the n-th forward node, respectively.

The Jacobian with respect to D is computed by the traditional element-based adjoint method, which is written as

$$\begin{aligned} J_D((s,r),\tau) &= \frac{\partial \Phi_{s,r}}{dD_\tau} \\ &= \sum_{e\in\Omega_\tau} \left(\mathbf{H}_\tau^e \mathbf{\Phi}_s^e\right)^T \mathbf{\Phi}_r^e \end{aligned} \tag{2.38}$$

where $\sum_{e\in\Omega_\tau}$ denotes the summation over all elements that in the immediate vicinity of the parameter node τ; matrix $\mathbf{H}_\tau^e$ has form of

$$h_{i,j}^\tau = \left\langle \phi_\tau(r)\nabla\varphi_i(r)\cdot\nabla\varphi_j(r)\right\rangle \tag{2.39}$$

2.5 Gauss-Newton iterative image reconstruction

As we derived above, the Jacobian matrix is simply the first order Fréchet derivative of a multi-input (parameters) and multi-output (measurements) function. The 1st order derivative also gives the gradient direction, using which, one can solve for the inverse problem more efficiently.

Overall, the goal of the image reconstruction problem is to solve the below optimization problem

$$\arg\min_\mu \|\mathbf{y} - \mathbf{\Phi}(\mu)\|_2^2 \tag{2.40}$$

where $\mathbf{y}$ is the measurements collected at all source-detector pairs, $\mathbf{\Phi}$ is the forward model that maps the parameters μ to the forward solution. In nearly all cases, the measurement $\mathbf{y}$ is not a complete sample of the forward solution, but rather a sparse subset acquired at discrete locations. To match the two terms, the second term in the above expression is often represented by $\mathbf{\Lambda\Phi}(\mu)$ where $\mathbf{\Lambda}$ is a sampling matrix. For simplicity, here we assume $\mathbf{\Lambda}$ is absorbed into the forward operator $\mathbf{\Phi}$. We want to note here that the model output $\mathbf{\Phi}(\mu)$ is a nonlinear function of the parameter μ. Thus, in order to solve for μ that minimizes (2.40), iterative estimation methods based on numerical gradient are often used.

The Gauss-Newton method is one of the widely applied algorithm to solve for the above optimization problem. It is a well-balanced algorithm with a convergence speed between the linear convergence algorithm, such as conjugate-gradient method, and the 2nd-order convergence algorithm, such as the Raphson-Newton algorithm. Compare to the Raphson-Newton algorithm, where the 2nd-order derivative, i.e. the Hessian matrix ($H = \frac{\partial^2 \mathbf{\Phi}}{\partial \mu^2}$), has to be computed, the Gauss-Newton method uses an approximated Hessian matrix, i.e. the Gauss-Newton Hessian matrix, defined by $G = J^T J$ where J is the Jacobian, i.e. the 1st order derivative.

2.6 Data calibration and simultaneous source-detector coupling coefficient estimation

The source-detector coupling coefficients (SD) are multiplicative factors associated with each source and detector, representing the optode-dependent variations, including varied source power, detector sensitivities, fiber coupling efficiency, optical fiber light leakage, bending and fiber-tissue coupling (such as hair, pressure differences, etc.). In practices, such coupling coefficients are typically considered to be removed after applying the below calibration formula:

$$\mathbf{\Phi}^t_{calib} = \frac{\hat{\mathbf{\Phi}}^t_{meas}}{\hat{\mathbf{\Phi}}^p_{meas}} \mathbf{\Phi}^p_{calc} \tag{2.41}$$

If we assume both the target measurement $\hat{\mathbf{\Phi}}^t$ and the phantom measurement $\hat{\mathbf{\Phi}}^p$ both contain the multiplicative SD coefficients, $s_i d_j$, for the (i, j)-th measurement pair, i.e. $\hat{\mathbf{\Phi}}^t = SD \times \mathbf{\Phi}^t$ and $\hat{\mathbf{\Phi}}^p = SD' \times \mathbf{\Phi}^p$. If we make the assumption that $SD = SD'$, applying Eq. 2.41, sd will be removed from the measurement data.

However, in realistic measurements, SD' may not be the same as SD, as a result, the calibrated data $\mathbf{\Phi}^t_{calib}$ may still present uncalibrated coupling differences. The uncalibrated SD coefficients are unknowns. It is possible to simultaneously fit for the optical properties using the measurement data as well as to fit for these unknown SD coefficients.

If we denote $\mathbf{\Phi}' = sd \times \mathbf{\Phi}$, where $sd = s \otimes d$ is the Kronecker product of the per-source-channel coupling coefficients s_i and the per-detector-channel coupling coefficients d_j. The simultaneous sd estimation aims to solve the following

equation in least-square estimation

$$\left(\mathbf{J}_{\mu_a}, \mathbf{J}_D, \mathbf{J}_{sd}\right)\begin{pmatrix} \mathbf{\Delta}\mu_a \\ \mathbf{\Delta}D \\ s \\ d \end{pmatrix} = \mathbf{\Phi}^{meas} - \mathbf{\Phi}^{calc} \tag{2.42}$$

where the *sd* Jacobian is defined as

$$\mathbf{J}_{sd} = [\mathbf{J}_s, \mathbf{J}_d] \tag{2.43}$$

and

$$\mathbf{J}_s = \frac{\partial(\{s_i d_j \times \Phi(i,j)\})}{\{s_i\}} = \{d_j \Phi(i,j)\} \tag{2.44}$$

$$\mathbf{J}_d = \frac{\partial(\{s_i d_j \times \Phi(i,j)\})}{\{d_j\}} = \{s_i \Phi(i,j)\} \tag{2.45}$$

$$\tag{2.46}$$

In frequency-domain (FD) systems, the measurement $\mathbf{\Phi}$ are complex numbers, therefore, $\mathbf{J}_{sd}$ is also complex. In this case, the s and d coupling coefficients can also be complex numbers. This makes them different from $\Delta\mu_a$ and Δ_D which can only take real-values.

2.7 Log-magnitude/unwrapped phase form of Jacobian

The complex form update equation is

$$\left(\mathbf{J}_{\mu_a}, \mathbf{J}_D\right)\begin{pmatrix} \mathbf{\Delta}\mu_a \\ \mathbf{\Delta}D \end{pmatrix} = \mathbf{\Phi}^{meas} - \mathbf{\Phi}^{calc} \tag{2.47}$$

or

$$\mathbf{J}\begin{pmatrix} \mathbf{\Delta}\mu_a \\ \mathbf{\Delta}D \end{pmatrix} = \mathbf{\Delta\Phi} \tag{2.48}$$

For FD-systems, the misfit vector $\mathbf{\Delta\Phi}$ is a complex number, so are the Jacobian matrices. In these cases the above equation can be solved in the real-form of (2.48) is given by

$$\begin{pmatrix} \Re e\mathbf{J}_{\mu_a} & \Re e\mathbf{J}_D \\ \Im m\mathbf{J}_{\mu_a} & \Im m\mathbf{J}_D \end{pmatrix} \begin{pmatrix} \Delta\mu_a \\ \Delta D \end{pmatrix} = \begin{pmatrix} \Re e\mathbf{\Delta\Phi} \\ \Im m\mathbf{\Delta\Phi} \end{pmatrix} \tag{2.49}$$

When *sd* is used, the above equation can be further expanded as

$$\begin{pmatrix} \Re e\mathbf{J}_{\mu_a} & \Re e\mathbf{J}_D & \Re e\mathbf{J}_{sd} & -\Im m\mathbf{J}_{sd} \\ \Im m\mathbf{J}_{\mu_a} & \Im m\mathbf{J}_D & \Im m\mathbf{J}_{sd} & \Re e\mathbf{J}_{sd} \end{pmatrix} \begin{pmatrix} \Delta\mu_a \\ \Delta D \\ \Re e(sd) \\ \Im m(sd) \end{pmatrix} = \begin{pmatrix} \Re e\mathbf{\Delta\Phi} \\ \Im m\mathbf{\Delta\Phi} \end{pmatrix} \tag{2.50}$$

When the log-amplitude/unwrapped phase reconstruction is enabled, instead of fitting for the complex-valued misfit vector $\mathbf{\Delta\Phi}$, one fits the concatenated log-amplitude ($\mathbf{\Delta\Gamma(\Phi)} = \log(|\mathbf{\Phi}^{meas}|) - \log(|\mathbf{\Phi}^{calc}|)$) and the phase ($\mathbf{\Delta\angle(\Phi)}$) vectors of the misfit. The above equation can be rewrite to

$$\begin{pmatrix} \mathbf{J}_1 & \mathbf{J}_3 & \mathbf{J}_5 \\ \mathbf{J}_2 & \mathbf{J}_4 & \mathbf{J}_6 \end{pmatrix} \begin{pmatrix} \Delta\mu_a \\ \Delta D \\ \Re e(sd) \\ \Im m(sd) \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta\Gamma(\Phi)} \\ \mathbf{\Delta\angle(\Phi)} \end{pmatrix} \tag{2.51}$$

where

$$\begin{aligned} \mathbf{J}_1 &= \frac{\Re e\mathbf{\Delta\Phi}\Re e\mathbf{J}_{\mu_a} + \Im m\mathbf{\Delta\Phi}\Im m\mathbf{J}_{\mu_a}}{|\mathbf{\Delta\Phi}|^2} \\ \mathbf{J}_2 &= \frac{\Re e\mathbf{\Delta\Phi}\Im m\mathbf{J}_{\mu_a} - \Im m\mathbf{\Delta\Phi}\Re e\mathbf{J}_{\mu_a}}{|\mathbf{\Delta\Phi}|^2} \\ \mathbf{J}_3 &= \frac{\Re e\mathbf{\Delta\Phi}\Re e\mathbf{J}_{\boldsymbol{D}} + \Im m\mathbf{\Delta\Phi}\Im m\mathbf{J}_{\boldsymbol{D}}}{|\mathbf{\Delta\Phi}|^2} \\ \mathbf{J}_4 &= \frac{\Re e\mathbf{\Delta\Phi}\Im m\mathbf{J}_{\boldsymbol{D}} - \Im m\mathbf{\Delta\Phi}\Re e\mathbf{J}_{\boldsymbol{D}}}{|\mathbf{\Delta\Phi}|^2} \\ \mathbf{J}_5 &= \frac{1}{|sd|} \\ \mathbf{J}_6 &= I \end{aligned} \tag{2.52}$$

where J_5 and J_6 are approximated versions of the SD Jacobian in this case.

2.8 Multi-spectral image reconstruction

When data from multiple wavelengths are simultaneously used, one can reconstruct wavelength-independent chromorphore molar-concentrations c_i or volume

fractions f_i instead of μ_a and D. The relationship between μ_a and c is expressed as

$$\mu_a(\lambda) = \epsilon_{HbO}(\lambda)c_{HbO} + \epsilon_{HbR}(\lambda)c_{HbR} + \epsilon_{water}(\lambda)f_{water} + \epsilon_{lipids}(\lambda)f_{lipids} + \cdots \quad (2.53)$$

where λ is the wavelength, $\epsilon_i(\lambda)$ are the extinction-coefficients of each chromorphore, and f_{water} and f_{lipids} are the volume fractions of water and lipids in the tissue. The extinction coefficients over wavelength can be found in the literature.

The inverse-power law is used to express the reduced scattering coefficient μ'_s in terms of wavelengths, i.e.

$$\mu'_s(\lambda) = a \times \lambda^{-b} \quad (2.54)$$

where a is the scattering amplitude, and b is the scattering power. In some literature, a normalization is performed, making it easier for unit conversions

$$\mu'_s(\lambda) = a \times \left(\frac{\lambda}{500\text{nm}}\right)^{-b} \quad (2.55)$$

in this case, a has the unit of μ'_s (1/mm) and b becomes unit-less.

In the multi-spectral reconstructions, the following equation is solved

$$\begin{pmatrix} \mathbf{J}_c(\lambda_1) & \mathbf{J}_a(\lambda_1) & \mathbf{J}_b(\lambda_1) & \mathbf{J}_{sd}(\lambda_1) \\ \mathbf{J}_c(\lambda_2) & \mathbf{J}_a(\lambda_2) & \mathbf{J}_b(\lambda_2) & \mathbf{J}_{sd}(\lambda_2) \\ \mathbf{J}_c(\lambda_3) & \mathbf{J}_a(\lambda_3) & \mathbf{J}_b(\lambda_3) & \mathbf{J}_{sd}(\lambda_3) \\ & \cdots & \cdots & \end{pmatrix} \begin{pmatrix} \mathbf{\Delta} c_{HbO} \\ \mathbf{\Delta} c_{HbR} \\ \mathbf{\Delta} f_{water} \\ \mathbf{\Delta} f_{lipids} \\ a \\ b \\ \Re e(sd) \\ \Im m(sd) \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta \Phi(\lambda_1)} \\ \mathbf{\Delta \Phi(\lambda_2)} \\ \mathbf{\Delta \Phi(\lambda_3)} \\ \cdots \end{pmatrix} \quad (2.56)$$

and $\mathbf{J}_c = \left[\mathbf{J}_{HbO}, \mathbf{J}_{HbR}, \mathbf{J}_{water}, \mathbf{J}_{lipids}, \cdots\right]$, with

$$\mathbf{J}_c^i(\lambda_j) = \epsilon_i(\lambda_j)\mathbf{J}_{\mu_a}(\lambda_j) \quad (2.57)$$

i ={HbO, HbR, water, lipids, ...}; also, we have

$$\mathbf{J}_a(\lambda) = -\frac{\lambda^b}{3a^2} = -\left[3D^2(\lambda)\lambda^{-b}\right]\mathbf{J}_D(\lambda) \quad (2.58)$$

$$\mathbf{J}_b(\lambda) = \frac{3a\lambda^{-b}}{\log(\lambda)} = \left[D(\lambda)\log(\lambda)\right]\mathbf{J_D}(\lambda) \quad (2.59)$$

Here we assume the *sd* coefficients are wavelength independent.

2.9 Solving the inverse problem

The perturbation equation in the 1st order, i.e. Eqs. 2.56, 2.49 and 2.51 can be solved using ordinary least-square (OLS), weighted least-square (WLS) or generalized least-square (GLS) methods.

The OLS solution solves the below parameter estimation problem

$$\arg\min_{\mu} \|\mathbf{y} - \mathbf{\Phi}(\mu)\|_2^2 \tag{2.60}$$

Assuming iid Gaussian noise in the measurement $\mathbf{y}$, the optimal solution is given by

$$\Delta\mu = (\mathbf{J^T J} + \lambda I)^{-1}\mathbf{J}^T\mathbf{\Delta\Phi}, \text{ or} \tag{2.61}$$

$$\Delta\mu = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda I)^{-1}\mathbf{\Delta\Phi} \tag{2.62}$$

where λ denotes the Tikhonov regularization parameter. Both the above forms are equivalent, according to the *Sherm-Morrison-Woodbury identity*. However, due to the matrix sizes, it is generally preferred to solve (2.61) when $\mathbf{J}$ has more rows than columns (i.e. overdetermined form: number of measurements is more than that of unknowns), and solve (2.62) when $\mathbf{J}$ has more columns than rows (i.e. underdetermined form: number of unknowns is more than that of the measurements).

If WLS is used, the measurement data are considered random variables that may have independent variances. The residual is weighted according to the inverse square-root of the covariance matrix ($\mathbf{C_y}$) of the data - so that the data presenting the highest variance will be trusted the least, and those presenting the lowest variance will contribute the most in the residual calculations. This is equivalent to solving the below optimization problem

$$\arg\min_{\mu} \|\mathbf{C_y}^{-1/2}\left(\boldsymbol{y} - \mathbf{\Phi}(\boldsymbol{\mu})\right)\|_2^2 \tag{2.63}$$

and this gives us the solution as

$$\Delta\mu = (\mathbf{J^T C_y^{-1} J} + \lambda I)^{-1}\mathbf{J^T C_y}^{-1}\mathbf{\Delta\Phi}, \text{ or} \tag{2.64}$$

$$\Delta\mu = \mathbf{J^T}(\mathbf{J}\mathbf{J^T} + \lambda\mathbf{C_y})^{-1}\mathbf{\Delta\Phi} \tag{2.65}$$

In the case of GLS, both the measurements and the unknowns are considered random variables, and their "assumed" covariance matrices, i.e. $\mathbf{C_y}$ and $\mathbf{C}_\mu$ respectively, as *a priori* information, should be used. This is equivalent to the below optimization problem

$$\arg\min_{\mu} \|\mathbf{C_y}^{-1/2}\left(\boldsymbol{y} - \mathbf{\Phi}(\boldsymbol{\mu})\right)\|_2^2 + \lambda\|\mathbf{C}_\mu{}^{-1/2}\boldsymbol{\mu}\|_2^2 \tag{2.66}$$

and the corresponding update equation can be seen as

$$\Delta\mu = (\mathbf{J}^{\mathbf{T}}\mathbf{C}_{\mathbf{y}}^{-1}\mathbf{J} + \lambda\mathbf{C}_{\mu}^{-1})^{-1}\mathbf{J}^{\mathbf{T}}{\mathbf{C}_{\mathbf{y}}}^{-1}\Delta\mathbf{\Phi}, \text{ or} \tag{2.67}$$

$$\Delta\mu = \mathbf{C}_{\mu}\mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{C}_{\mu}\mathbf{J}^{\mathbf{T}} + \lambda\mathbf{C}_{\mathbf{y}})^{-1}\Delta\mathbf{\Phi} \tag{2.68}$$

In reality, the covariance matrix of the measurements $\mathbf{C}_{\mathbf{y}}$ can be estimated using repeated measurements, whereas the covariance matrix of the unknowns $\mathbf{C}_{\mu}$ need to be provided as the prior information.

When the prior values (or the expectations) of the unknowns are assumed as $\boldsymbol{\mu}_0$, one can solve the below extended optimization problem as

$$\arg\min_{\mu} \|{\mathbf{C}_{\mathbf{y}}}^{-1/2}(\boldsymbol{y} - \boldsymbol{\Phi}(\boldsymbol{\mu}))\|_2^2 + \lambda\|{\mathbf{C}_{\mu}}^{-1/2}(\boldsymbol{\mu} - \boldsymbol{\mu_0})\|_2^2 \tag{2.69}$$

and the corresponding solutions are

$$\Delta\mu = (\mathbf{J}^{\mathbf{T}}\mathbf{C}_{\mathbf{y}}^{-1}\mathbf{J} + \lambda\mathbf{C}_{\mu}^{-1})^{-1}\left[\mathbf{J}^{\mathbf{T}}{\mathbf{C}_{\mathbf{y}}}^{-1}\Delta\mathbf{\Phi} - \boldsymbol{C}_{\mu}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu_0})\right], \text{ or} \tag{2.70}$$

$$\Delta\mu = \left[I - \mathbf{C}_{\mu}\mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{C}_{\mu}\mathbf{J}^{\mathbf{T}} + \lambda\mathbf{C}_{\mathbf{y}})^{-1}\mathbf{J}\right] \times\left[\mathbf{C}_{\mu}\mathbf{J}^{\mathbf{T}}\mathbf{C}_{\mathbf{y}}^{-1}\Delta\mathbf{\Phi} - (\boldsymbol{\mu} - \boldsymbol{\mu_0})\right] \tag{2.71}$$

2.10 Tissue bulk optical property fitting

The tissue bulk optical property refers to the set of averaged optical property across a heterogeneous domain. The estimation of the tissue bulk properties is achieved via solving the same optimization problem as shown above, except that the unknown space is compressed dramatically and contain only a single set of parameters, one parameter per unknown specie.

Using the general reconstruction equation Eq. 2.56, the bulk optical properties can be estimated by solving the below equation

$$\begin{pmatrix} \sum_i J_c(\lambda_1) & \sum_i J_a(\lambda_1) & \sum_i J_b(\lambda_1) & \sum_i J_{sd}(\lambda_1) \\ \sum_i J_c(\lambda_2) & \sum_i J_a(\lambda_2) & \sum_i J_b(\lambda_2) & \sum_i J_{sd}(\lambda_2) \\ \sum_i J_c(\lambda_3) & \sum_i J_a(\lambda_3) & \sum_i J_b(\lambda_3) & \sum_i J_{sd}(\lambda_3) \\ & \cdots & \cdots & \end{pmatrix} \begin{pmatrix} \overline{\Delta c_{HbO}} \\ \overline{\Delta c_{HbR}} \\ \overline{\Delta f_{water}} \\ \overline{\Delta f_{lipids}} \\ \bar{a} \\ \bar{b} \\ \overline{\Re e(sd)} \\ \overline{\Im m(sd)} \end{pmatrix} = \begin{pmatrix} \Delta\mathbf{\Phi}(\boldsymbol{\lambda_1}) \\ \Delta\mathbf{\Phi}(\boldsymbol{\lambda_2}) \\ \Delta\mathbf{\Phi}(\boldsymbol{\lambda_3}) \\ \cdots \end{pmatrix} \tag{2.72}$$

where $\sum_i$ denotes the summation of all the columns (the dimension corresponds to the unknowns) of the Jacobian matrices and $\overline{\mathbf{\Delta} c_i}$ is the update to correct the i-th bulk optical property from the initial background value.

The above optical property fitting algorithm can be generalized in the below two scenarios. In the first case, where a piecewise constant segmentation is provided (as priors), the above equation can be modified to fit multiple sets of optical property, one for each segmentation

$$\begin{pmatrix} \sum_{i\in\Omega_1} J(\lambda_1) & \sum_{i\in\Omega_2} J(\lambda_1) & \cdots & \sum_{i\in\Omega_{N_s}} J(\lambda_1) \\ \sum_{i\in\Omega_1} J(\lambda_2) & \sum_{i\in\Omega_2} J(\lambda_2) & \cdots & \sum_{i\in\Omega_{N_s}} J(\lambda_2) \\ \sum_{i\in\Omega_1} J(\lambda_3) & \sum_{i\in\Omega_2} J(\lambda_3) & \cdots & \sum_{i\in\Omega_{N_s}} J(\lambda_3) \\ & \cdots & \cdots & \end{pmatrix} \begin{pmatrix} \overline{\mathbf{\Delta}\mu_{\Omega_1}} \\ \overline{\mathbf{\Delta}\mu_{\Omega_2}} \\ \overline{\mathbf{\Delta}\mu_{\Omega_3}} \\ \cdots \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta\Phi(\lambda_1)} \\ \mathbf{\Delta\Phi(\lambda_2)} \\ \mathbf{\Delta\Phi(\lambda_3)} \\ \cdots \end{pmatrix} \tag{2.73}$$

where $\Delta\mu_{\Omega_i} = [\overline{c_{HbO,\Omega_i}}, \overline{c_{HbR,\Omega_i}}, \cdots]^T$ is the average optical properties for the i-th segment. The above segmentation-based bulk-property fitting method is sometimes referred to as the "hard-priors".

If a probabilistic segmentation is provided to replace the binary segmentations, where every node is represented as a set of probabilities (or fraction of volumes) of being different tissue types, so that the probility map is defined as a matrix

$$P = \left\{p_i^k\right\}_{i=1,\cdots,N;k=1,\cdots,N_s} \tag{2.74}$$

where $0 \le p_i^k \le 1$ is the probability for the i-th node/voxel to be tissue type k, and $\sum_k p_i^k = 1$ holds for any i. The above segmentation-based bulk-property fitting equation can be generalized to

$$\begin{pmatrix} \mathbf{J}(\lambda_1)P \\ \mathbf{J}(\lambda_2)P \\ \mathbf{J}(\lambda_3)P \\ \cdots \end{pmatrix} \begin{pmatrix} \overline{\mathbf{\Delta}\mu_{\Omega_1}} \\ \overline{\mathbf{\Delta}\mu_{\Omega_2}} \\ \overline{\mathbf{\Delta}\mu_{\Omega_3}} \\ \cdots \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta\Phi(\lambda_1)} \\ \mathbf{\Delta\Phi(\lambda_2)} \\ \mathbf{\Delta\Phi(\lambda_3)} \\ \cdots \end{pmatrix} \tag{2.75}$$

Once the tissue-type averaged optical property update is calculated using the above equation, the node values are subsequently updated by $\{\Delta\mu_i\}_i = P \times \{\overline{\Delta\mu_{\Omega_k}}\}_{k=1,\cdots,N_s}$.

2.11 Defining priors using the L-matrix

In many cases, Eq. 2.69 is solved in the following form in the context of Tikhonov regularization

$$\arg\min_{\mu} \|\boldsymbol{y} - \mathbf{\Phi}(\boldsymbol{\mu})\|_2^2 + \lambda \|L(\boldsymbol{\mu} - \boldsymbol{\mu_0})\|_2^2 \tag{2.76}$$

Compared with Eq. 2.69, it is clear that the regularization matrix (L) is the assumed inverse square-root of the covariance matrix of unknowns ($\mathbf{C}_\mu{}^{-1/2}$), determined by the user as the *a prior* information.

Various non-identity L-matrices have been proposed. For example, it can be defined as a simple Laplacian to penalize rapid changes between neighboring nodes or voxels. For an FEM-mesh based solution, L can be defined as

$$l_{i,j} = \begin{cases} -1 & \text{if } j \text{ and } j \text{ are connected and } i \neq j \\ 0 & \text{if } i \text{ and } j \text{ are not connected and } i \neq j \\ N_i & \text{if } i = j \end{cases} \tag{2.77}$$

where N_i is the degree of the node, denoting the total neighbor count of the node i.

The above Laplacian operator can be also applied to segmented domains derived from prior information (such as a co-registered structural-scan of the same volume), where the nodes/elements within the same tissue regions are labeled identically, we have

$$l_{i,j} = \begin{cases} -\frac{1}{N_k} & \text{if } i \text{ and } j \text{ belong to the same region } \Omega_k \\ 0 & \text{if } i \text{ and } j \text{ belong to different regions} \\ 1 & \text{if } i = j \end{cases} \tag{2.78}$$

where N_k is the total node/voxels within the k-th region Ω_k. The above method is sometimes referred to as the "soft-priors".

Another reported L-matrix example is the so-called "Helmholtz operator".

$$l_{i,j} = \begin{cases} -\frac{1}{N_k+(h/h_f)^2} & \text{if } i \text{ and } j \text{ belong to the same region } \Omega_k \\ 0 & \text{if } i \text{ and } j \text{ belong to different regions} \\ 1 & \text{if } i = j \end{cases} \tag{2.79}$$

where h is the average distance between nodes, and h_f is the desired average image feature size.

2.12 Compositional-prior guided reconstructions

The "compositional prior" guided reconstruction algorithm was proposed in 2010 [8]. In this algorithm, the L matrix is also derived from a co-registered structure-image but with the added ability to fully utilize the detailed gray-scale image features [9]. Different from all above mentioned prior-guided reconstruction algorithms, where

spatial nodes are initially segmented as piecewise constant regions, the compositional prior utilizes probabilistic segmentation of the tissue, as described in Section 2.10.

In short, the compositional prior approach penalizes reconstruct recovered property differences in the compositional space. It computes the similarities in tissue compositions between spatial locations, derived from the intensity maps from *a priori* structural images, and build a weighted graph connecting all nodes in the compositional space (N_s dimensional space with p^k as coordinates), then create the L-matrix as the Laplacian of the weighted graph in the compositional space.

Specifically, the L-matrix encodes the compositional prior is constructed as

$$l_{i,j} = \begin{cases} -\frac{u_{i,j}}{\beta\sqrt{d_i d_j}} & \text{if } \|C_i - C_j\|_2 < \alpha N_s, \text{ i.e. connected} \\ 0 & \text{if } \|C_i - C_j\|_2 \geq \alpha N_s, \text{ i.e. not connected} \\ 1 & \text{if } i = j \end{cases} \tag{2.80}$$

where $\|\cdot\|_2$ is the L_2-norm, N_s is the total number of tissue types, $C_i = \{p_i^1, p_i^2, \cdots, p_i^{N_s}\}$ is the compositional vector at i-th node, and $\alpha \in [0, 1]$ is a user-defined parameter to control the sparsity of the L-matrix - the lower the α value, the sparser the L-matrix; $\beta > 1$ is another user-specified parameter to control the diagonal dominance of the L-matrix - the larger the β, the less penalization between nodes. The typical values for α is between 0.1 and 0.2, and that for β is 1.2.

Parameter $u_{i,j}$ is the weight between nodes i and j, defined as the average per-component Euclidean distance in the compositional space between the two nodes, as $u_{i,j} = \alpha - \|C_i - C_j\|_2/N_s \geq 0$, and d_i and d_j are the "degrees" of the two nodes in a weighted graph, defined as $d_i = \sum_{k,\|C_i - C_k\|_2 < \alpha N_s} u_{i,k}$.

As one can see, the above definition of L-matrix does not require piecewise constant segmentations, and avoids the loses of the gray-scale features from the structural images. The probabilistic segmentation is also better suited for medical imaging because a small region in the biological tissue typically contains a mixture of various types of tissues. The tissue probability can be treated as the volume fractions of different tissue types inside the smallest spatial discretization unit, such as a voxel or a tetrahedron.

It is apparent that there are many ways to segment the *a priori* structural image intensity maps to the probability maps in the compositional space. We call this step as the "fuzzy segmentation". The above proposed algorithm is quite general and can accommodate many fuzzy segmentation methods. A simple linear mapping was proposed in the original paper, and various alternative mapping methods,

including those based on the Gaussian-mixture models and threshold-based algorithms were studied in Deng 2015. From the latter paper, we concluded that the the reconstructed images are not sensitive to the fuzzy segmentation algorithm; however, all of these methods showed significantly around 2× errors in the reconstructed images.

As we show in Section 2.11, the L-matrix is equivalent to the $\mathbf{C}_\mu{}^{-1/2}$ term in GLS, and $\mathbf{C}_\mu = (L^T L)^{-1}$. Replacing $\mathbf{C}_\mu$ to (2.68), we have

$$\mathbf{\Delta\mu} = (\mathbf{J^T J} + \lambda \mathbf{L^T L})^{-1} J^T \mathbf{\Delta\Phi}, \text{ or} \tag{2.81}$$

$$\mathbf{\Delta\mu} = (\mathbf{L^T L})^{-1}\mathbf{J^T}\left[\mathbf{J}(\mathbf{L^T L})^{-1}\mathbf{J^T} + \lambda I\right]^{-1}\mathbf{\Delta\Phi} \tag{2.82}$$

When multiple properties are defined on every node, the compositional-prior derived L-matrix is shared between multiple blocks of the parameters. Given the block structure as shown in (2.56), we have the following equation for the over-determined form

$$\mathbf{\Delta\mu} = \left[\begin{pmatrix}\mathbf{J}_1\\ \mathbf{J}_2\\ \mathbf{J}_3\\ \cdots\end{pmatrix}[\mathbf{J_1}, \mathbf{J_2}, \cdots] + \lambda\begin{pmatrix}\mathbf{C}_\mu\mathbf{J}_1\\ \mathbf{C}_\mu\mathbf{J}_2\\ \mathbf{C}_\mu\mathbf{J}_3\\ \cdots\end{pmatrix}\right]^{-1}\mathbf{\Delta\Phi} \tag{2.83}$$

Similarly, for the under-determined form, we have

$$\begin{aligned}\mathbf{\Delta\mu} &= \begin{pmatrix}\mathbf{C}_\mu\mathbf{J}_1\\ \mathbf{C}_\mu\mathbf{J}_2\\ \mathbf{C}_\mu\mathbf{J}_3\\ \cdots\end{pmatrix} \\ &\times \left[\begin{pmatrix}\mathbf{J}_1\mathbf{C}_\mu\mathbf{J}_1^T & 0 & 0 & \cdots\\ 0 & \mathbf{J}_2\mathbf{C}_\mu\mathbf{J}_2^T & 0 & \cdots\\ 0 & 0 & \mathbf{J}_3\mathbf{C}_\mu\mathbf{J}_3^T & \cdots\\ & \cdots & & \end{pmatrix} + \lambda I\right]^{-1}\mathbf{\Delta\Phi}\end{aligned} \tag{2.84}$$

where the Jacobian blocks $\mathbf{J}_k$ are the Jacobians corresponding to each optical property (chromorphore concentrations, absorption, scattering etc) defined on the nodes.

In Deng 2015, we propose a fast algorithm to compute the above under-determined form by first computing the QR decomposition of L as $L = QR$, and then compute R^{-1} and construct an intermediate matrix $Z_i = \mathbf{J}_i R^{-1}$. Therefore, we have $\mathbf{C}_\mu\mathbf{J}_i = R^{-1}Z^T$ and $\mathbf{J}_i\mathbf{C}_\mu\mathbf{J}_i^T = Z_i Z^T$. Replacing these two terms into (2.84), we can solve for the under-determined solution more efficiently.

2.13 Additional techniques

2.13.1 Dual-mesh method for solving decoupled forward and inverse problems

2.13.2 Multiple right-hand-side linear solvers for accelerating matrix solutions

2.13.3 Joint use of continuous-wave and frequency-domain data

2.13.4 Matrix scaling and variance equalization

2.13.5 Emperical Tikhonov regularization method

2.13.6 Two-dimensional prior guided 3D reconstructions

2.13.7 Other supported linear and least-square solvers

Chapter 3

Software implementations

Chapter 4

Example implementations

Bibliography

[1] Q. Fang, S. A. Carp, J. Selb, R. Moore, D. B. Kopans, E. L. Miller, D. H. Brooks, and D. A. Boas. A multi-modality image reconstruction platform for diffuse optical tomography. In *Biomedical Optics*, page BMD24, Washington, D.C., 2008. OSA.

[2] Q. Fang, P. M. Meaney, S. D. Geimer, and and K. D. Paulsen A. V. Streltsov. Microwave image reconstruction from 3D fields coupled to 2D parameter estimation. *IEEE Trans. Med. Imag.*, 23:475–484, Apr 2004.

[3] Qianqian Fang, Paul M. Meaney, and Keith D. Paulsen. Viable three-dimensional medical microwave tomography: Theory and numerical experiments. *IEEE Transactions on Antennas and Propagation*, 58(2):449–458, 2010.

[4] Qianqian Fang. *Computational Methods for Microwave Medical Imaging*. Ph.d. dissertation, Dartmouth College, Hanover, NH, 2004.

[5] Q. Fang. *Computational methods for microwave medical imaging*. Ph.D., Dartmouth College, Hanover, NH, U.S.A., Dec 2004.

[6] W. E. Boyse and A. A. Seidl. A block QMRmethod for computing multiple simultaneous solutions to complex symmetric systems. *SIAM J. Sci. Comput.*, 17:263–274, 1996.

[7] K. D. Paulsen, P. M. Meaney, M. J. Moskowitz, and J. M. Sullivan Jr. A dual mesh scheme for finite element based reconstruction algorithms. *IEEE Trans. Med. Imag.*, 14:504–514, 1995.

[8] Qianqian Fang, Richard H. Moore, Daniel B. Kopans, and David A. Boas. Compositional-prior-guided image reconstruction algorithm for multi-modality imaging. *Biomedical Optics Express*, 1:223, 2010.

[9] Bin Deng, Dana H. Brooks, David A. Boas, Mats Lundqvist, and Qianqian Fang. Characterization of structural-prior guided optical tomography using realistic breast models derived from dual-energy x-ray mammography. *Biomedical Optics Express*, 6:2366, 2015.

[10] David Boas, Thomas Gaudette, and Simon Arridge. Simultaneous imaging and optode calibration with diffuse optical tomography. *Optics Express*, 8:263, 2 2001.

[11] Martin Schweiger, Ilkka Nissilä, David A. Boas, and Simon R. Arridge. Image reconstruction in optical tomography in the presence of coupling errors. *Applied Optics*, 46(14):2743, may 2007.

[12] S. R. Arridge. Optical tomography in medical imaging. *Inv. Problems*, 5:R41–R93, 1999.

[13] R. C. Haskell, L. O. Svaasand, T. Tsay, T. Feng, M. S. McAdams, and B. J. Tromberg. Boundary conditions for the diffusion equation in radiative transfer. *J. Opt. Soc. Am. A*, 11:2727–2741, 1994.