Published in final edited form as:

J Quant Spectrosc Radiat Transf. 2015 December 1; 167: 10–22. doi:10.1016/j.jqsrt.2015.07.015.

Fast linear solver for radiative transport equation with multiple right hand sides in diffuse optical tomography

Jingfei Jia[a], **Hyun K. Kim**[b], and **Andreas H. Hielscher**[a,b,c]

Jingfei Jia: jj2523@columbia.edu; Andreas H. Hielscher: ahh2004@columbia.edu

[a]Columbia University, Department of Biomedical Engineering, 351 Engineering Terrace, 1210 Amsterdam Avenue, New York, New York 10027

[b]Columbia University Medical Center, Department of Radiology, 630 West 168th Street, New York, New York 10032

[c]Columbia University, Department of Electrical Engineering, 1300 S.W. Mudd, 500 West 120th Street, New York, New York 10027

Abstract

It is well known that radiative transfer equation (RTE) provides more accurate tomographic results than its diffusion approximation (DA). However, RTE-based tomographic reconstruction codes have limited applicability in practice due to their high computational cost. In this article, we propose a new efficient method for solving the RTE forward problem with multiple light sources in an *all-at-once* manner instead of solving it for each source separately. To this end, we introduce here 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. Two parallelized block BiCGStab methods are proposed for additional acceleration under limited threads situation. We evaluate the performance of this algorithm with numerical simulation studies involving the Delta-Eddington approximation to the scattering phase function. The results show that the single threading block RTE solver proposed here reduces computation time by a factor of 1.5~3 as compared to the traditional sequential solution method and the parallel block solver by a factor of 1.5 as compared to the traditional parallel sequential method. This block linear solver is, moreover, independent of discretization schemes and preconditioners used; thus further acceleration and higher accuracy can be expected when combined with other existing discretization schemes or preconditioners.

Keywords

Diffuse optical tomography; Radiative transfer equation; Block BiCGStab algorithm; Multiple right hand sides; Multi-threading acceleration

Correspondence to: Hyun K. Kim; Andreas H. Hielscher, ahh2004@columbia.edu.

1. Introduction

Diffuse optical tomography (DOT) has become a popular area of research that attracts significant and increasing attentions [1, 2]. In DOT, low-energy near-infrared (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 inside the medium under investigation. Tissues optical properties vary depending on the type and location of tissue [3]; thus reconstructed optical properties can provide physiologically important information such as oxy-hemoglobin (HbO_2) and deoxy-hemoglobin (Hb) in tissue. DOT has applied mainly to brain imaging [4, 5], breast imaging [6–9], vascular imaging [10], small animal imaging [11, 12] and imaging of finger joints [13, 14].

The DOT problem can be described in general terms as an inverse problem that is defined to find an optimal set of optical properties that minimizes a mismatch between predictions and measurements of light intensities. Predictions and measurements are made on the tissue surface with a known distribution of light sources. Multiple forward problems need to be solved in each inverse iteration to generate an updating direction for the target variables from the current estimate of optical properties in tissue. Traditional unconstrained approach formulates the DOT problem as [15]:

$$\min_{\mu} f(\mu) = \frac{1}{2}\sum_{k=1}^{N_S} \|QF_{\mu}^{-1}\left(b^{(k)}\right) - M^{(k)}\|^2 + R(\mu) \quad (1)$$

where μ is the optical property of the imaging object and F_μ is the forward model of light intensity distribution defined as a function of μ, N_S is the number of sources used, $M^{(k)}$ is the measurement and $b^{(k)}$ represents the right hand side of the discretized forward model given by $A\psi^{(k)} = b^{(k)}$ with the kth light source, where A is the linear operator generated by the discretization of the forward light propagation model and $\psi^{(k)}$ is the vector that contains light intensities of all discretized directions and locations in the medium for the kth light source, Q denotes the measurement operator that models the light propagation from the object surface to the detectors and $R(\mu)$ is the regularization term on the optical property. The PDE-constrained approach that doesn't require an explicit solution of the forward model of light intensities can be formulated as follows [16, 17]:

$$\min f\left(\mu, \psi^{(1)}, \ldots, \psi^{(N_S)}\right) = \frac{1}{2}\sum_{k=1}^{N_S} \|Q\psi^{(k)} - M^{(k)}\|^2 + R(\mu) \quad (2)$$

$$\text{s.t.} F_{\mu}\left(\psi^{(k)}\right) = b^{(k)} \text{for} k = 1, \ldots, N_S$$

where $\psi^{(k)}$ denotes the light intensity distribution of the forward model, $R(\mu)$ is the regularization term.

The forward light propagation model plays a very important role in DOT since an improper light propagation model will lead to inaccurate reconstruction results. One frequent used approach to model photon propagation is the Monte Carlo (MC) method [18]. In this approach photons are considered as individual particles. Launching millions of them into a medium and tracking each one individually, one obtains a statistical approximation of the real distribution of photons in the medium. One can show that for an infinite number of photons, the so-calculated distribution of photons converges to the correct results. However, because of the substantial computational cost this approach is not very practical when used in combination with large-scale inverse solvers. Less computationally demanding are deterministic light propagation models that are based on the radiative transfer equation (RTE) and its diffusion approximation (DA). Of these two the DA model is most commonly used in tissue optics, because it is easy to implement and provides solutions very fast. However, DA-based results are often not accurate enough when considering small-tissue geometries, high-absorbing and low-scattering tissues, and void-like region. In these cases the diffusion approximation is not valid and RTE-based codes need to be employed [19].

The RTE is a partial differential-integro equation in which a dependent variable (i.e., radiance in units of W/cm^2/sr) is defined as a function of two independent variables (i.e., spatial position and angular direction). Due to strong coupling in directions, analytic solutions of RTE are not available for most cases and numerical solvers need to be implemented. Recently, several efficient RTE solvers have been developed (see [20–23] and their references). However, to our best knowledge, all existing algorithms are designed to solve a single right hand side. Therefore, only one source is considered. On the other hand, RTE-based DOT codes are based on multiple right hand sides, which correspond to multiple light sources illumination. Traditional methods to solve multiple right hand sides are to solve each right hand side separately or solve multiple right hand sides simultaneously in parallel [24, 25]. However, the extensive computational power those parallel solvers required are not always available; hence we focus here on the numerical method for solving multiple right hand sides simultaneously on a single thread or limited threads (thread number is less than source number).

In order to solve multiple right hand sides efficiently, we 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 sources illumination:

$$F_\mu(\Psi)=B \quad (3)$$

where Ψ is the matrix of solution vectors $\psi^{(1)}$, $\psi^{(2)},\dots,\psi^{(NS)}$ and B of right hand sides $b^{(1)}$, $b^{(2)},\dots,$ $b^{(NS)}$ pertaining to the ith light source illumination. Methods for solving such linear systems with multiple right hand sides have been extensively studied in other areas [26–31]. The block Krylov subspace methods have been shown to be effective compared to other solvers designed for multiple right hand sides [32, 33]. In our work, we introduce the Krylov subspace block BiCGStab algorithm [34]. Compared to other methods, it has the advantages of low memory requirement, simple structure and stable convergence. Besides these, 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 with little effort.

The remainder of the paper is organized as follows. We first review Krylov subspace and block Krylov subspace methods and introduce the block BiCGStab algorithm in section 2. A second order finite volume scheme combined with discrete ordinates for discretization of RTE in frequency-domain is introduced in section 3. Then two parallelization methods of the block BiCGStab are proposed in section 4. Numerical results are presented in section 5 that address the performance evaluation of the block BiCGStab algorithm. Finally our conclusions are summarized in Section 6.

2. Preconditioned block linear solver for RTE with multiple sources

2.1. Krylov subspace method and preconditioned BiCGStab algorithm

To understand the block Krylov subspace algorithm, we begin with a brief introduction of traditional Krylov subspace algorithms. Krylov subspace methods are the most widely used iterative methods so far for large-scale sparse linear systems $Ax = b$. In Krylov subspace methods, we define an affine space $K_n(A, r_0)$ that grows with n as follows:

$$K_n(A, r_0) := \mathrm{span}(r_0, Ar_0, \ldots, A^{n-1}r_0) \quad (4)$$

where $r_0 \coloneqq b - Ax_0$ is the initial residual with an initial guess x_0, the nth 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 [35], the biconjugate gradient (BiCG) method [36], the biconjugate gradient stabilized method (BiCGStab) method [37], the quasi-minimal residual (QMR) method [38], and the transpose-free quasi-minimal residual (TFQMR) method [39], to name a few. BiCGStab is one of the most popular solvers since it has three advantages over other type of Krylov subspace methods.

1. BiCGStab is a three-term recurrence method; therefore it has relatively low memory requirement.
2. BiCGStab doesn't 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 1 [37].

Algorithm 1

BiCGStab method with left preconditioner

```
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 $r$ ˜such that $\rho_0 = r^{T}\tilde{r}_0 \neq 0$.
4.  Given $\varepsilon > 0$, for $i = 0, 1, 2 \ldots$ until $\|r_i\|/\|b\| < \varepsilon$:
    (1)    $S_i = P^{-1}Ap_i$.
    (2)    $\gamma_i = r^{T}\tilde{S}_i$.
    (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)   $\rho_{i+1} = r^{T}\tilde{r}_{i+1}$.
    (12)   $\beta_i = \gamma_i^{-1}\rho_{i+1}/\eta_i$.
```

2.2. Krylov subspace method and preconditioned BiCGStab algorithm

In DOT, one has to solve the linear system with multiple right hand sides given as:

$$A\Psi = B \quad (5)$$

where A is an $N \times N$ sparse non-Hermitian matrix obtained from the discretization of the light propagation model, Ψ and B are $N \times N_S$ rectangular complex matrices whose column vectors are $\psi^{(1)}, \psi^{(2)}, \ldots, \psi^{(N_S)}$ and $b^{(1)}, b^{(2)}, \ldots, b^{(N_S)}$ respectively. In the case of RTE-based DOT, $N = N_{CV} \times N_{SA}$, where N_{CV} and N_{SA} represent the number of control volumes and solid angles, N_S denotes the source number used in the reconstruction.

Under this type of problem setting, it is more efficient to solve the linear system of N_S right hand sides given by Eq. (5) 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. (4): a similar affine space $K_n^{\blacksquare}(A, R_0)$ that grows with n is defined as,

$$K_n^{\blacksquare}(A, R_0) := \mathrm{span}(R_0, AR_0, \ldots, A^{n-1}R_0) \quad (6)$$

where $R_0 := \left[r_0^{(1)}, r_0^{(2)}, \ldots, r_0^{(N_S)}\right]$. 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 1) into a block version for the linear equation with multiple right hand sides, the left preconditioned block BiCGStab algorithm can be summarized in Algorithm 2 [34]:

Algorithm 2

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 $N_S \times N$ matrix $\tilde{R}$ such that $\rho_0 = R\tilde{R}_0 \neq 0$.
4. Given $\varepsilon > 0$, for $i = 0, 1, 2 \ldots$ until $\|R_i(:, j)\|/\|B(:, j)\| < \varepsilon$ for $j = 1,\ldots, N_S$:
   (1)  $S_i = P^{-1} AP_i$.
   (2)  $\gamma_i = R\tilde{S}_i$.
   (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) $\rho_{i+1} = R\tilde{R}_{i+1}$.
   (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 $N \times N_S$ matrices S_k, R_k, T_k, P_k and one $N_S \times N$ matrix $\tilde{R}$ (excluding A, Ψ, B and the preconditioner P) need to be stored in memory: therefore the total number of floating numbers to save is ($5NN_S + O(N_S^2)$), thus making the method memory-efficient. The left preconditioned block BiCGStab method requires the evaluation of $2N_S$ matrix vector products, $2N_S$ preconditioning procedures and a total of $6NN_S^2 + 4NN_S + O(N_S^3)$ 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 2 is almost the same as it in Algorithm 1.

3. Finite-volume discrete-ordinates discretization with RTE

The frequency-domain RTE can be written as

$$\left(\frac{i\omega}{\nu} + \boldsymbol{\Omega} \cdot \nabla + \mu_a(\boldsymbol{r}) + \mu_s(\boldsymbol{r})\right) \psi(\boldsymbol{r}, \boldsymbol{\Omega}) = \mu_s(\boldsymbol{r}) \int_{S^{d-1}} p(\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}')\psi(\boldsymbol{r}, \boldsymbol{\Omega}')d\boldsymbol{\Omega}' + q(\boldsymbol{r}, \boldsymbol{\Omega}) \quad (7)$$

where $i = \sqrt{-1}$, ω is the modulation frequency and ν is the light speed in the medium, $\boldsymbol{r}$ is the spatial position vector, d is the dimensionality constant of the medium, $\Omega \in S^{d-1}$ is the vector that denotes the photon propagation direction and its spherical measure is represented by $d\Omega$, μ_a ($\boldsymbol{r}$) and μ_s ($\boldsymbol{r}$) represent the absorption coefficient and scattering coefficient which are spatial functions with respect to $\boldsymbol{r}$, $\psi(\boldsymbol{r}, \Omega)$ is the radiance in units of [W/ cm^2/ sr] (d = 3) or [W/ cm/ rad] (d = 2), q ($\boldsymbol{r}$, Ω) represents the source configuration, $p(\Omega{\cdot}\Omega')$ is the phase function that represents the proportion of energy that is scattered from direction Ω' into direction Ω. The Henyey-Greenstein phase function [40] as commonly used in tissue optics is given as:

$$p(\Omega\cdot\Omega\text{'}){=}f(x){=}\begin{cases}\frac{1-g^2}{2\pi(1+g^2-2g(\Omega\cdot\Omega\text{'}))}, & d{=}2\\ \frac{1-g^2}{4\pi(1+g^2-2g(\Omega\cdot\Omega\text{'}))^{3/2}}, & d{=}3\end{cases} \quad (8)$$

where g is the anisotropic factor.

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

$$\psi(\boldsymbol{r}_b,\Omega){=}q(\boldsymbol{r}_b,\Omega){+}R(-\Omega\cdot\boldsymbol{n})\psi(\boldsymbol{r}_b,\Omega\text{'}),\Omega\cdot\mathbf{n}{<}0 \quad (9)$$

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

The frequency-domain RTE is discretized by a node-centered finite volume scheme combined with discrete ordinates method [17]. First we discretized the solid angle domain with discrete-ordinates method (S_n), thus the integral on the right hand side of Eq. (7) can be approximated with a weighted sum of the radiance field in different directions,

$$\int_{S^2} p(\Omega_i\cdot\Omega\text{'})\psi(\boldsymbol{r},\Omega\text{'})d\Omega\text{'}{=}\sum_{k=1}^{N_{SA}} p_{ik}w_k\psi(\boldsymbol{r},\Omega_k) \quad (10)$$

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

hus based on Eq. (10), Eq. (7) is decomposed into a set of N_{SA} coupled partial differential equations which correspond to N_{sa} discretized photon propagation directions,

$$\left(\frac{i\omega}{\nu}{+}\Omega_i\cdot\nabla{+}\mu_a(\boldsymbol{r}){+}\mu_s(\boldsymbol{r})\right)\psi_i(\boldsymbol{r}){=}\mu_s(\boldsymbol{r})\sum_{k=1}^{N_{SA}} p_{ik}w_k\psi_k(\boldsymbol{r}){+}q_i(\boldsymbol{r}),i{=}1,\ldots,N_{SA} \quad (11)$$

where $\psi(\boldsymbol{r}, \Omega_i)$ is written in short as ψ_i ($\boldsymbol{r}$).

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 frequency-domain RTE is given by

$$\left(\frac{i\omega}{\nu}{+}\mu_{a,j}{+}\mu_{s,j}\right)\psi_{i,j}{+}\frac{1}{V_j}\sum_{l=1}^{N_j} A_j^l(\boldsymbol{n}_j^l{\cdot}\Omega_i)f_{i,j}^l{=}\mu_{s,j}\sum_{k=1}^{N_{SA}} p_{ik}w_k\psi_{k,j}{+}q_{i,j},j{=}1,\ldots,N_{CV} \quad (12)$$

where $\mu_{a,j}$, $\mu_{s,j}$, $\psi_{i,j}$ and $q_{i,j}$ represent μ_a ($\boldsymbol{r}_j$), μ_s ($\boldsymbol{r}_j$), ψ_i($\boldsymbol{r}_j$) and q_i ($\boldsymbol{r}_j$) in Eq. (11) respectively, V_j and N_j denote the volume and the neighbor number of the control volume centered at $\boldsymbol{r}_j$, A_j^l and $\boldsymbol{n}_j^l$ are the area and outgoing unit normal vector of the l th surface of the control volume centered at $\boldsymbol{r}_j$. 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 $\boldsymbol{r}_j$ and its l th surface. The, $f_{i,j}^l$ in Eq. (12) 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_j, then, $f_{i,j}^l$ is given by

$$f_{i,j}^l = \begin{cases} \psi_{i,j}, \boldsymbol{n}_j^l \cdot \Omega_i > 0 \\ \psi_{i,j_l}, \boldsymbol{n}_j^l \cdot \Omega_i \leq 0 \end{cases} \quad (13)$$

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

$$f_{i,j}^l = \begin{cases} \psi_{i,j}, \boldsymbol{n}_j^l \cdot \Omega_i > 0 \\ \psi_{i,j}, \boldsymbol{n}_j^l \cdot \Omega_i \leq 0 \end{cases} \quad (14)$$

where $\psi_{i,j}$ is given by the discretization of the reflective boundary condition (9) as follows,

$$\psi_{i,j} = q_{i,j} + R\left(-\boldsymbol{n}_j^l \cdot \Omega_i\right)\psi_{i,j} \quad (15)$$

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

Finally, a fully discretized linear equation can be obtained by plugging (13)~(15) into (12). The linear equation has the following form:

$$D\psi = S\psi + b \quad (16)$$

where the vector ψ ($\psi_{1,1}$, $\psi_{1,2}$ …, $\psi_{1,N_{SA}}$, $\psi_{2,1}$, $\psi_{2,2}$ …,$\psi_{2,N_{SA}}$,…,$\psi_{N_{CV},1}$,$\psi_{N_{CV},2}$ …,$\psi_{N_{CV},N_{SA}}$)^T. D and S are two matrices corresponding to the linear terms on the left and right hand sides respectively in Eq. (12). In practice, since S is block diagonal matrix having special pattern as follows:

$$S = \begin{bmatrix} V_1\mu_{s,1}P & & & & \\ & V_2\mu_{s,2}P & & & \\ & & \ddots & & \\ & & & V_{N_{cv}-1}\mu_{s,N_{cv}-1}P & \\ & & & & V_{N_{cv}}\mu_{s,N_{cv}}P \end{bmatrix}$$

where P is a $N_{SA} \times N_{SA}$ matrix with $P_{ij} = p_{ij}w_j$, it can be very efficiently saved into memory by saving P and the vector μ_s $(\mu_{s,1},\ldots,\mu_{s,N_{CV}})$.

Thus in DOT, forward problem with multiple sources can be written as

$$A\psi^{(k)}=b^{(k)} \quad (17)$$

where A $D - S$ is the discretized linear operator for frequency-domain RTE and k is the source index. It is easy to check that the source configuration only contributes on the right hand side $b^{(k)}$ but not the matrix A. So we can write an all-at-once equation as given in Eq. (5) for all source configurations since they all share the same linear operator.

In practice biological tissue are strongly forward-peaked scattering ($g \geq 0.8$) [3]. Thus a large N_{SA} in Eq. (11) is required to accurately describe such highly anisotropic scattering behavior, which leads to a considerable computational burden and memory requirement. In order to overcome this difficulty, Hielscher and Klose introduced the Delta-Eddington method [41], which allows using only a small number of discrete ordinates in RTE to approximate the original scattering function given by Eq. (8). In our work we employ the zeroth-order Delta-Eddington approximation since it has shown promising results as reported in literature [1, 14, 42]. The frequency domain RTE is simplified with zeroth-order Delta-Eddington approximation as

$$\left(\frac{i\omega}{\nu}+\Omega\cdot\nabla+\mu_a(\boldsymbol{r})+\mu_s^{'}(\boldsymbol{r})\right)\psi(\boldsymbol{r},\Omega)=\mu_s^{'}(\boldsymbol{r})\int_{S^{d-1}}\psi(\boldsymbol{r},\Omega')d\Omega'/\left(\int_{S^{d-1}}d\Omega'\right)+q(\boldsymbol{r},\Omega) \quad (18)$$

where the scattering coefficient μ_s and the anisotropic factor g is combined into a reduced scattering coefficient $\mu_s^{'}(1-g)\mu_s$. In this paper, we focus on solving Eq. (18) along with a problem setup of multiple right hand sides.

4. Multi-threading acceleration for block BiCGStab algorithm

In DOT, multi-threading technique is often applied in solving forward problems with multiple right hand sides in order to gain extra speedup. Under limited resources case (the available threads number $N_{TH} < N_S$), the most commonly used method is as follows: (1) distribute N_S sources equally (or almost equally) into N_{TH} threads, (2) solve a group of forward problems with 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, its implementation is straightforward and it is not affected by parallel slowdown effects, due to communication/synchronization overhead. However, this method still treats every single source independently, so it doesn’t 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 we propose two parallel methods generalized from the block BiCGStab algorithm.

In our first parallel method (see Algorithm 3) we parallelize Algorithm 2 directly 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 2. However, due to synchronizations and communications between threads (see Step 5, 7(4), 7(10), 7(17)), the algorithm may cause parallel slowdown effects with the increase of thread number.

Algorithm 3

parallel block BiCGStab with inter-thread communication (parallel block BiCGStab-ITC)

Divide the right hand side B and the initial guess Ψ_0 equally (or almost equally) into N_{TH} thinner matrices:

```
$B=\left[B^{(1)},\dots,B^{(N_{TH})}\right],\Psi_0=\left[\Psi_0^{(1)},\dots,\Psi_0^{(N_{TH})}\right].$

Compute residual $R_0^{(k)}=P^{-1}\left(B^{(k)}-A\Psi_0^{(k)}\right)$, $K$ = 1,…,$N_{TH}$.

Set $P_0^{(k)}=R_0^{(k)}$, $k$ = 1,…,$N_{TH}$.

Setup a $N_S \times N$ matrix $\tilde{R}=\left[R_0^{(1)},\dots,R_0^{(N_{TH})}\right]^T$.

1.                Threads synchronization.

Compute $\rho_0^{(k)}=\tilde{R}R_0^{(k)}$, $k$ = 1,…,$N_{TH}$.
Given $\varepsilon > 0$, for $i$ = 0,1, 2… until $R_i(:, j) / B(:, j) < \varepsilon$ for $j$ =1,…,$N_S$:

$S_i^{(k)}=P^{-1}AP_i^{(k)}$, $k$ = 1,…,$N_{TH}$.

$\gamma_i^{(k)}=\tilde{R}S_i^{(k)}$, $k$ = 1,…,$N_{TH}$.

Update $P_i$, $S_i$ and
$\gamma_i$:$P_i=\left[P_i^{(1)},\dots,P_i^{(N_{TH})}\right],S_i=\left[S_i^{(1)},\dots,S_i^{(N_{TH})}\right],\gamma_i=\left[\gamma_i^{(1)},\dots,\gamma_i^{(N_{TH})}\right].$

                  (1) Threads synchronization.

$\alpha_i^{(k)}=\gamma_i^{-1}\rho_i^{(k)}$, $k$ = 1,…,$N_{TH}$.

$\Psi_{i+1}^{(k)}=\Psi_i^{(k)}+P_i^{(k)}\alpha_i^{(k)}$, $k$ = 1,…,$N_{TH}$.

$R_{i+1}^{(k)}=R_i^{(k)}-S_i^{(k)}\alpha_i^{(k)}$, $k$ = 1,…,$N_{TH}$.

$T_i^{(k)}=P^{-1}AR_{i+1}^{(k)}$, $k$ = 1,…,$N_{TH}$.

Compute $c_i^{(k)}=T_i^{(k)},R_{i+1_F}^{(k)},d_i^{(k)}=T_i^{(k)},T_i^{(k)}{}_F$.
```

```
Threads synchronization and compute $\eta_i = \sum_{k=1}^{N_{TH}} c_i^{(k)} / \sum_{k=1}^{N_{TH}} d_i^{(k)}$.
$\Psi_{i+1}^{(k)} = \Psi_i^{(k)} + \eta_i R_{i+1}^{(k)}$, $k = 1,\ldots,N_{TH}$.
$R_{i+1}^{(k)} = R_{i+1}^{(k)} - \eta_i T_i^{(k)}$, $k = 1,\ldots,N_{TH}$.
$P_{i+1}^{(k)} = P_i^{(k)} - \eta_i S_i^{(k)}$, $k = 1,\ldots,N_{TH}$.
$\rho_{i+1}^{(k)} = \tilde{R} R_{i+1}^{(k)}$, $k = 1,\ldots,N_{TH}$.
$\beta_i^{(k)} = \gamma_i^{-1} \rho_{i+1}^{(k)} / \eta_i$, $k = 1,\ldots,N_{TH}$.
$P_{i+1}^{(k)} = R_{i+1}^{(k)} + P_{i+1}^{(k)} \beta_i^{(k)}$, $k = 1,\ldots,N_{TH}$.
        (2) Threads synchronization.
Return $\Psi = \left[\Psi_i^{(1)}, \ldots, \Psi_i^{(N_{TH})}\right]$.
```

The second parallel approach is summarized in Algorithm 4. 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 requires more iterations to converge especially when N_S / N_{TH} is close to 1.

Algorithm 4

parallel block BiCGStab with no inter-thread communication (parallel block BiCGStab-NITC)

Divide column vectors of B and the initial guess Ψ_0 into N_{TH} subgroups (See (19)):

```
$B = \left[B^{(1)}, \ldots, B^{(N_{TH})}\right], \Psi_0 = \left[\Psi_0^{(1)}, \ldots, \Psi_0^{(N_{TH})}\right]$.
Solve $A\Psi^{(k)} = B^{(k)}$, $k = 1,\ldots,N_{TH}$ with Algorithm 2 in parallel.
Return $\Psi = [\Psi^{(1)},\ldots,\Psi^{(N_{TH})}]$.
```

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

$$\max d$$

$$\text{subject to}$$

$$z_i^{(k)} \in \{0,1\}, 1 \le i \le N_S, 1 \le k \le N_{TH} \quad (z_i^{(k)}{=}1 \text{if source} i \text{belongs to group} k)$$

$$\sum_{k=1}^{N_{TH}} z_i^{(k)}{=}1, 1 \le i \le N_S \quad (\text{One source belongs to exactly one group}) \qquad (19)$$

$$0 \le \sum_{i=1}^{N_S} z_i^{(k)} - N_S/N_{TH}{<}1 \quad (\text{Equally separation})$$

$$d \le d_{ij} z_i^{(k)} z_j^{(k)}, 1 \le i{<}j \le N_S, 1 \le k \le N_{TH} \quad (d \le \text{the minimum in}{-}\text{group distance})$$

This problem can be solved quickly when N_{TH} and N_S are not very large with the mixed integer linear programming solver in Matlab.

5. Numerical results

The performance of block linear solver with respect to CPU time may be most affected by the following two factors: 1) the structure of the matrix A in linear equation (5) that is determined by the anisotropic factor, optical properties, and spatial and angular discretization; 2) 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. 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. 1, 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). 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 the literature [17], which ensures fair comparison between different linear solvers. We set the same relative tolerance $\varepsilon = 10^{-10}$ for all the aforementioned linear solvers: in fact block solvers use equal or stronger stopping criteria than the traditional BiCGStab solver (see the difference between Algorithm 1(4) and Algorithm 2(4), Algorithm 3(7) for comparison). An ILUT(0.01, 5) preconditioner [43] was applied here for all methods. All numerical experiments were performed with dual Intel Xeon CPU X5650 and 32GB physical memory.

5.1. Influence of the anisotropic factor on the performance

The linear operator *A* in the discretized RTE (17) is affected by the pattern *P* in *S* (see (16)), 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 0.1cm^{-1} and multiple reduced scattering coefficients ($\mu_s^{'}[\mathrm{cm}^{-1}]=1.0, 5.0, 10.0$) are examined that lead to various combinations of different scattering coefficients and anisotroic factors. For each fixed $\mu_s^{'}$, we test four different anisotropic factors, 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 FD-RTE (7) is used as the forward model for every combination of (g, μ_s) on each phantom, and the discretized linear equation with the Henyey-Greenstein phase function (8) is solved with both the traditional BiCGStab (Algorithm 1) and the block BiCGStab (Algorithm 2). 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 2.

Table 2 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 (18) based on zeroth-order Delta-Eddington approximation.

5.2. Influence of optical properties on the performance

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] (cm^{-1}) and [0.1,10](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. We reported the overall CPU time and matrix-vector multiplication (MV) number per single right hand side for both block and traditional BiCGStab algorithms with respect to each pair of (μ_a, $\mu_s^{'}$). Speedup factors based on CPU times and MV ratios (BiCGStab MV number / block BiCGStab MV number) with respect to (μ_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. 2(a)(b) and (d)(e): it is observed that both algorithms take more MV operations and thus more computational time to converge when reduced scattering coefficient increases and absorption coefficient decreases, and vice versa. However, the speedup factors of 1.3 ~ 3.1 and the ratios of MV number of 1.0 ~ 2.2 (see Fig. 2(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 (< 0.3cm^{-1}) and reduced scattering coefficients (> 3cm^{-1}) and the speedup factor of 1.8 and the MV ratio of 1.3 are always guaranteed when reduced scattering coefficient is greater than 2.0cm^{-1}. Similar results are observed for the 3D numerical experiments (see Fig. 3). 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 ~ 2.7 and the MV ratios of 1.0 ~ 2.0. With reduced scattering coefficients ≥ 2.0cm^{-1}, a speedup factor of at least 1.8 and a MV ratio of at least 1.3 can be achieved.

We observe that the profile of speedup factor and the MV ratio are very similar in both 2D and 3D cases (See Fig. 2(c)(f) and Fig. 3(c)(f)). This strong correlation shows the higher efficiency of block BiCGStab, as compared to single right hand side BiCGStab, mainly comes from the less required MV operations since it searches solution in larger subspace in every iteration. Besides that, 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 [44]. 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.

5.3. Influence of spatial and angular discretization on the performance

When optical properties are fixed in (18), 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 S in (16) finer spatial discretization leads to more diagonal blocks as well as smaller coefficients for pattern P in every block but P itself is kept the same while different angular discretization leads to different pattern P but with total diagonal block number and P '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: the absorption and reduced scattering coefficients are set to 0.1cm^{-1} and 10.0cm^{-1}. In order to test how discretization affects the speedup factor, we have generated a number of spatial meshes varying element sizes in the range 0.08 ~ 0.13cm in the spatial domain together with different orders of discrete ordinates (S_4 ~ S_{12}) in the angular domain. Detailed information can be found in Table 3 and 4. 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. 4.

As shown in Fig. 4(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 4(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. 4(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. 4(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.

5.4. 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 μ_a = 0.1cm^{-1} and $\mu_s^{'}=10.0\text{cm}^{-1}$. We test the performance of the block BiCGStab algorithm with various numbers of sources (N_S = 1 ~ 25), which are uniformly located on the medium surface. The average CPU time, speedup factor and MV ratio are recorded and reported in Fig. 5.

As shown in Fig. 5(a), the CPU time grows approximately linearly for both solvers with respect to the source number when $N_S \geq 10$ 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. 5(b). We can find when $N_S \leq 15$, the MV ratio and speedup factor increase sublinearly with respect to source number. When $N_S > 15$, 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. 5(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 (~5), which makes the block BiCGStab algorithm more attractive in practical applications.

5.5. Test on multi-threading block BiCGStab methods

In order to evaluate the performance under the multi-threading computation environment, we test the two proposed parallel block BiCGStab methods (Algorithm 3 and 4) with the 3D homogeneous cylinder phantom. We consider two cases with the optical properties: (1) μ_a = 0.1cm^{-1}, $\mu_s^{'}=10.0\mathrm{cm}^{-1}$ for normal tissue; (2) μ_a = 0.5cm^{-1}, $\mu_s^{'}=1.0\mathrm{cm}^{-1}$ for high absorbing and low scattering case. In both cases, we solved the forward problems with the single threading or 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 and 4 as compared to the traditional sequential BiCGStab solver will decrease for different reasons. For Algorithm 3, this is mainly because more threads lead to more synchronization and communication cost, while for Algorithm 4 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. 6. The CPU time in seconds and the average MV number are shown in Fig. 6(a) and (c), and the speedup factor and MV ratio in Fig. 6(b) and (d). As expected, Fig. 6 (c) shows that the average MV number remains constant with various thread numbers both for parallel sequential method and Algorithm 3 but it increases for Algorithm 4. Fig. 6(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 treads, Algorithm 3 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 4 dominates the parallel slowdown effect due to communication in Algorithm 3. With 16 threads, very little improvement is obtained for both of the two parallel solvers since the synchronization and communication cost in Algorithm 3 become as much dominant as extra MVs in Algorithm 4.

For high absorbing and low scattering medium case, the CPU time, speedup factor, average MV number and MV ratio with respect to thread number are shown in Fig. 7 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 (see Fig. 7(c)). Therefore for Algorithm 3, the parallel slowdown by the synchronization and communication cost, become the most important factor. On the other hand, Algorithm 4 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 4 gives better performance than Algorithm 3, with thread number less than or equal to 8, a speedup factor of ~1.5 can be achieved.

6. Conclusion

In this work, we present a novel method for solving the forward problem in RTE-based optical tomography with multiple right hand sides simultaneously. 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 a framework of block Krylov subspace. In particular, we 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, our 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 ≤ 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.

We found 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. $\mu_a = 0.1\text{cm}^{-1}$, $\mu_s^{'}=10.0\text{cm}^{-1}$), 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. $\mu_a = 0.5\text{cm}^{-1}$, $\mu_s^{'}=1.0\text{cm}^{-1}$), we obtained a speedup factor of 1.5 or higher. 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 (~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 (~5),

Overall, the forward problems with multiple sources in RTE-based DOT can be solved much more efficiently without losing any accuracy with our proposed block BiCGStab solver. Moreover, the efficiency of the block solver can be further improved when using multiple threads (thread number < source number) by our proposed algorithms, depending on the optical properties. Since the forward solver is usually considered as the most time-consuming component in the inverse solver, we can expect a significant speed on image reconstruction by applying the block BiCGStab as the forward solver, which makes RTE-based DOT more practical in clinical applications.

Acknowledgments

This work has been supported in part by grants from the National Institutes of Health (NHLBI 1R01HL115336-01) and NIAMS 5R01AR050026-10), the Wallace H. Coulter Foundation (WHCF CU11-2443) and the New York City Partnership Fund - Bioaccelerate Program.

Reference

1. Hielscher AH, Bluestone AY, Abdoulaev GS, Klose AD, Lasker J, Stewart M, Netz U, Beuthan J. Near-infrared diffuse optical tomography. Dis Markers. 2002; 18:313–337. [PubMed: 14646043]
2. Gibson AP, Hebden JC, Arridge SR. Recent advances in diffuse optical imaging. Phys Med Biol. 2005; 50:R1–R43. [PubMed: 15773619]
3. Boas, DA.; Pitris, C.; Ramanujam, N. Handbook of biomedical optics. Boca Raton: CRC Press; 2011.
4. Hebden JC, Gibson A, Austin T, Yusof RM, Everdell N, Delpy DT, Arridge SR, Meek JH, Wyatt JS. Imaging changes in blood volume and oxygenation in the newborn infant brain using three-dimensional optical tomography. Phys Med Biol. 2004; 49:1117–1130. [PubMed: 15128193]
5. Culver JP, Siegel AM, Stott JJ, Boas DA. Volumetric diffuse optical tomography of brain activity. Opt Lett. 2003; 28:2061–2063. [PubMed: 14587815]
6. Ntziachristos V, Yodh AG, Schnall M, Chance B. Concurrent MRI and diffuse optical tomography of breast after indocyanine green enhancement. P Natl Acad Sci USA. 2000; 97:2767–2772.
7. Corlu A, Choe R, Durduran T, Rosen MA, Schweiger M, Arridge SR, Schnall MD, Yodh AG. Three-dimensional in vivo fluorescence diffuse optical tomography of breast cancer in humans. Opt Express. 2007; 15:6696–6716. [PubMed: 19546980]
8. Flexman ML, Khalil MA, Al Abdi R, Kim HK, Fong CJ, Desperito E, Hershman DL, Barbour RL, Hielscher AH. Digital optical tomography system for dynamic breast imaging. J Biomed Opt. 2011; 16
9. Flexman ML, Kim HK, Gunther JE, Lim EA, Alvarez MC, Desperito E, Kalinsky K, Hershman DL, Hielscher AH. Optical biomarkers for breast cancer derived from dynamic diffuse optical tomography. J Biomed Opt. 2013; 18
10. Khalil MA, Kim HK, Kim IK, Flexman M, Dayal R, Shrikhande G, Hielscher AH. Dynamic diffuse optical tomography imaging of peripheral arterial disease. Biomed Opt Express. 2012; 3:2288–2298. [PubMed: 23024920]
11. Hielscher AH. Optical tomographic imaging of small animals. Curr Opin Biotech. 2005; 16:79–88. [PubMed: 15722019]
12. Gulsen G, Birgul O, Unlu MB, Shafiiha R, Nalcioglu O. Combined diffuse optical tomography (DOT) and MRI system for cancer imaging in small animals. Technol Cancer Res T. 2006; 5:351–363.
13. Klose AD, Hielscher AH, Hanson KM, Beuthan J. Two- and three-dimensional optical tomography of finger joints for diagnostics of rheumatoid arthritis. Photon Propagation in Tissues Iv, Proceedings. 1998; 3566:151–160.
14. Hielscher AH, Klose AD, Scheel AK, Moa-Anderson B, Backhaus M, Netz U, Beuthan J. Sagittal laser optical tomography for imaging of rheumatoid finger joints. Phys Med Biol. 2004; 49:1147–1163. [PubMed: 15128195]
15. Ren K, Bal G, Hielscher AH. Frequency domain optical tomography based on the equation of radiative transfer. Siam J Sci Comput. 2006; 28:1463–1489.
16. Abdoulaev GS, Ren K, Hielseher AH. Optical tomography as a PDE-constrained optimization problem. Inverse Probl. 2005; 21:1507–1530.
17. Kim HK, Hielscher AH. A PDE-constrained SQP algorithm for optical tomography based on the frequency-domain equation of radiative transfer. Inverse Probl. 2009; 25
18. Zhu CG, Liu Q. Review of Monte Carlo modeling of light transport in tissues. J Biomed Opt. 2013; 18

19. Hielscher AH, Alcouffe RE, Barbour RL. Comparison of finite-difference transport and diffusion calculations for photon migration in homogeneous and heterogeneous tissues. Phys Med Biol. 1998; 43:1285–1302. [PubMed: 9623656]

20. Gao H, Zhao HK. A Fast-Forward Solver of Radiative Transfer Equation. Transport Theor Stat. 2009; 38:149–192.

21. Han WM, Huang JG, Eichholz JA. Discrete-Ordinate Discontinuous Galerkin Methods for Solving the Radiative Transfer Equation. Siam J Sci Comput. 2010; 32:477–497.

22. Mayer B, Kylling A. Technical note: The libRadtran software package for radiative transfer calculations - description and examples of use. Atmos. Chem. Phys. 2005; 5:1855–1877.

23. Klose AD, Netz U, Beuthan J, Hielscher AH. Optical tomography using the time-independent equation of radiative transfer — Part 1: forward model. Journal of Quantitative Spectroscopy and Radiative Transfer. 2002; 72:691–713.

24. Gu XJ, Kim HK, Masciotti J, Hielscher AH. A Parallel Reduced-Space Sequential-Quadratic Programming Algorithm for Frequency-domain Small Animal Optical Tomography. Proc Spie. 2009; 7174

25. Gao H, Phan L, Lin YT. Parallel multigrid solver of radiative transfer equation for photon transport via graphics processing unit. J Biomed Opt. 2012; 17

26. Simoncini V, Gallopoulos E. An Iterative Method for Nonsymmetric Systems with Multiple Right-Hand Sides. Siam J Sci Comput. 1995; 16:917–933.

27. Freund RW, Malhotra M. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides. Linear Algebra Appl. 1997; 254:119–157.

28. Calandra H, Gratton S, Lago R, Vasseur X, Carvalho LM. 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. 2013; 35:S345–S367.

29. Sakurai T, Tadano H, Kuramashi Y. Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD. Comput Phys Commun. 2010; 181:113–117.

30. Lambers JV. Application of Block Krylov Subspace Spectral Methods to Maxwell's Equations. Aip Conf Proc. 2009; 1174:1–15.

31. Stathopoulos A, Orginos K. Computing and Deflating Eigenvalues While Solving Multiple Right-Hand Side Linear Systems with an Application to Quantum Chromodynamics. Siam J Sci Comput. 2010; 32:439–462.

32. Gutknecht MH. Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. 2006

33. El Guennouni A, Jbilou K, Sadok H. The block Lanczos method for linear systems with multiple right-hand sides. Appl Numer Math. 2004; 51:243–256.

34. El Guennouni A, Jbilou K, Sadok H. A block version of BiCGSTAB for linear systems with multiple right-hand sides. Electron T Numer Ana. 2003; 16:129–142.

35. Saad Y, Schultz MH. Gmres - a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear-Systems. Siam J Sci Stat Comp. 1986; 7:856–869.

36. Rana IE, Alexopoulos NG. Current Distribution and Input Impedance of Printed Dipoles. Ieee T Antenn Propag. 1981; 29:99–105.

37. Vandervorst HA. Bi-Cgstab - a Fast and Smoothly Converging Variant of Bi-Cg for the Solution of Nonsymmetric Linear-Systems. Siam J Sci Stat Comp. 1992; 13:631–644.

38. Freund RW, Nachtigal NM. Qmr - a Quasi-Minimal Residual Method for Non-Hermitian Linear-Systems. Numer Math. 1991; 60:315–339.

39. Freund RW. A Transpose-Free Quasi-Minimal Residual Algorithm for Non-Hermitian Linear-Systems. Siam J Sci Comput. 1993; 14:470–482.

40. Henyey LG, Greenstein JL. Diffuse radiation in the galaxy. Astrophys J. 1941; 93:70–83.

41. Klose AD, Hielseher AH. Modeling photon propagation in anisotropically scattering media with the equation of radiative transfer. P Soc Photo-Opt Ins. 2003; 4955:624–633.

42. Montejo LD, Jia JF, Kim HK, Netz UJ, Blaschke S, Muller GA, Hielscher AH. Computer-aided diagnosis of rheumatoid arthritis with optical tomography, Part 1: feature extraction. J Biomed Opt. 2013; 18

43. Saad Y. ILUT: A dual threshold incomplete LU factorization. Numerical Linear Algebra with Applications. 1994; 1:387–402.

44. Nakamura Y, Ishikawa KI, Kuramashi Y, Sakurai T, Tadano H. Modified block BiCGSTAB for lattice QCD. Comput Phys Commun. 2012; 183:34–37.

- We solve the multiple-right-hand-side problem in DOT with a block BiCGStab method.
- We examine the CPU times of the block solver and the traditional sequential solver.
- The block solver is faster than the sequential solver by a factor of 1.5~3.0.
- Multi-threading block solvers give additional speedup under limited threads situation.

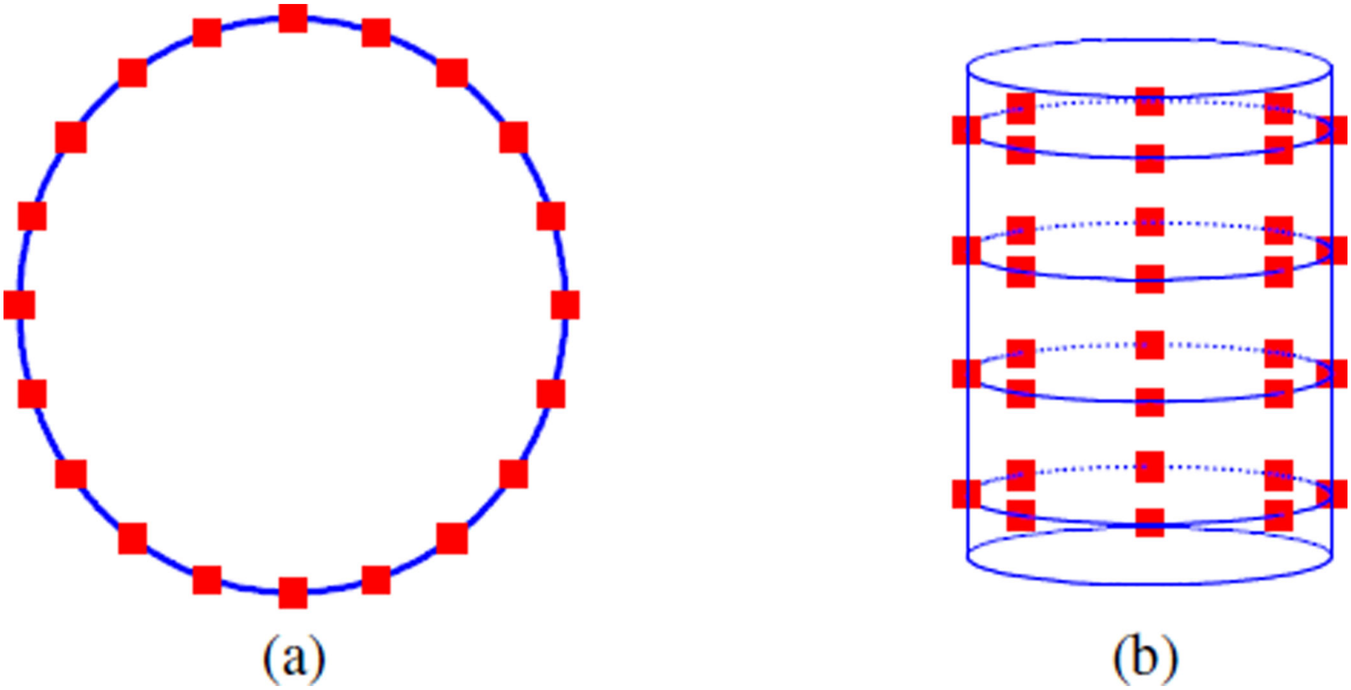


Fig 1.

Source distribution 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.

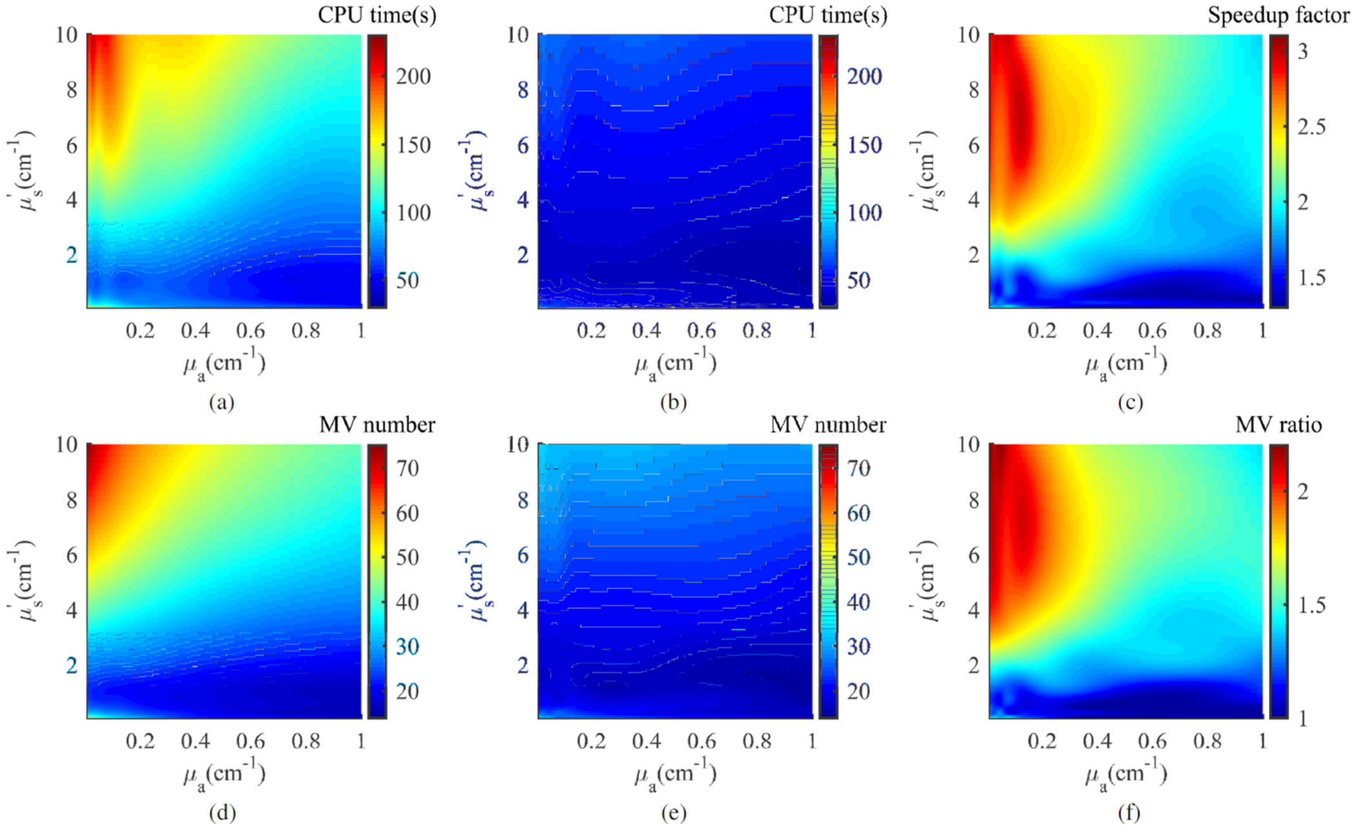


Fig 2.

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).

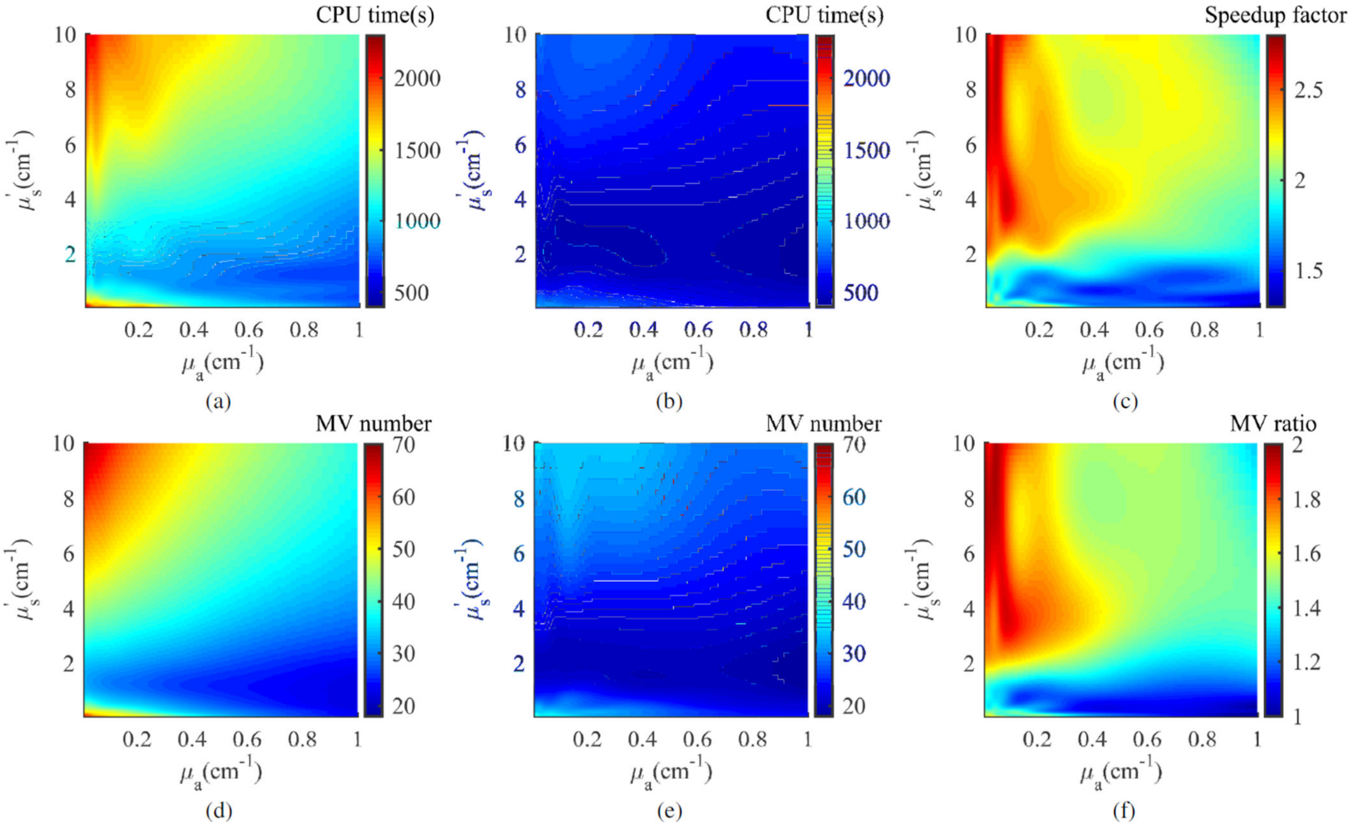


Fig 3.

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).

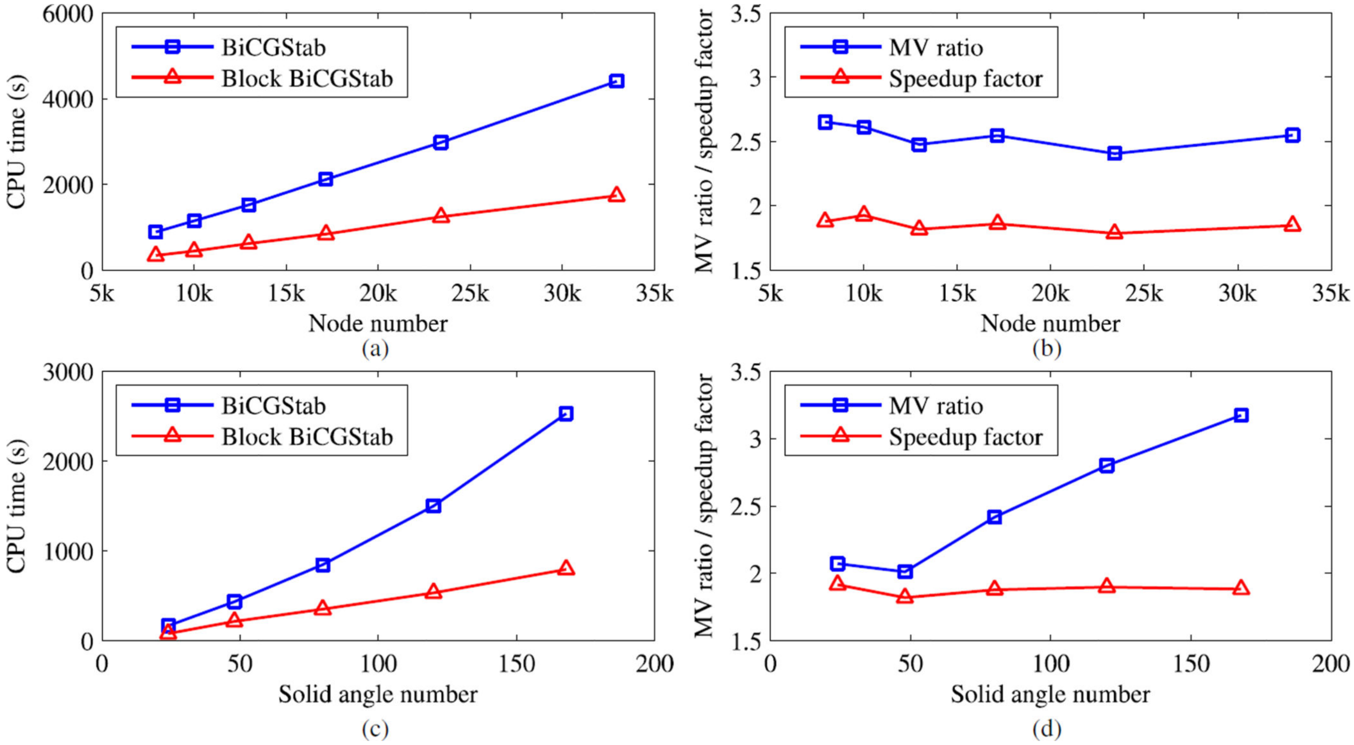
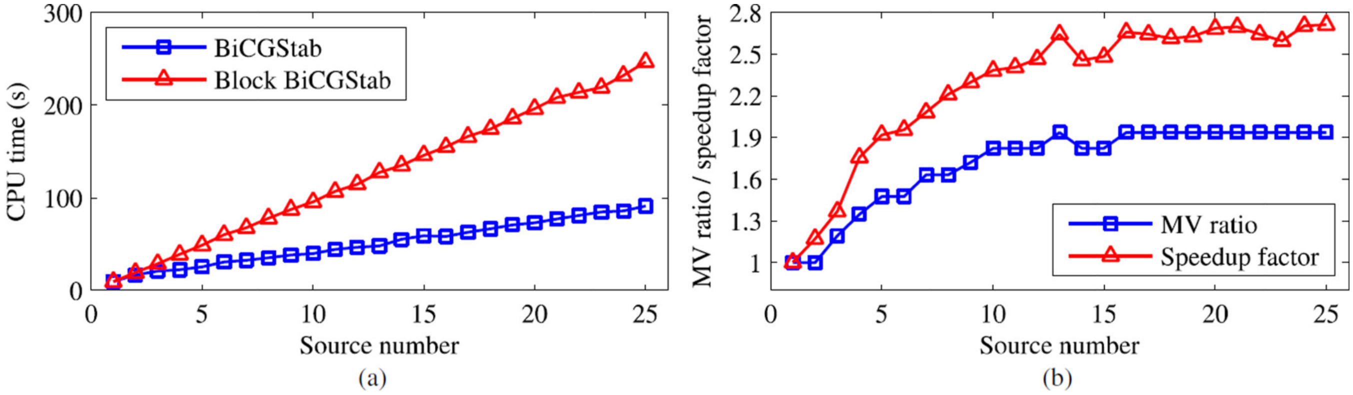


Fig 4.

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 ($S_4 \sim S_{12}$) with the fixed spatial mesh.



(a)

(b)

Fig 5.

Performance of the block BiCGstab algorithm with different number of sources: (a) Average CPU per source required in the block BiCGstab algorithm 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.

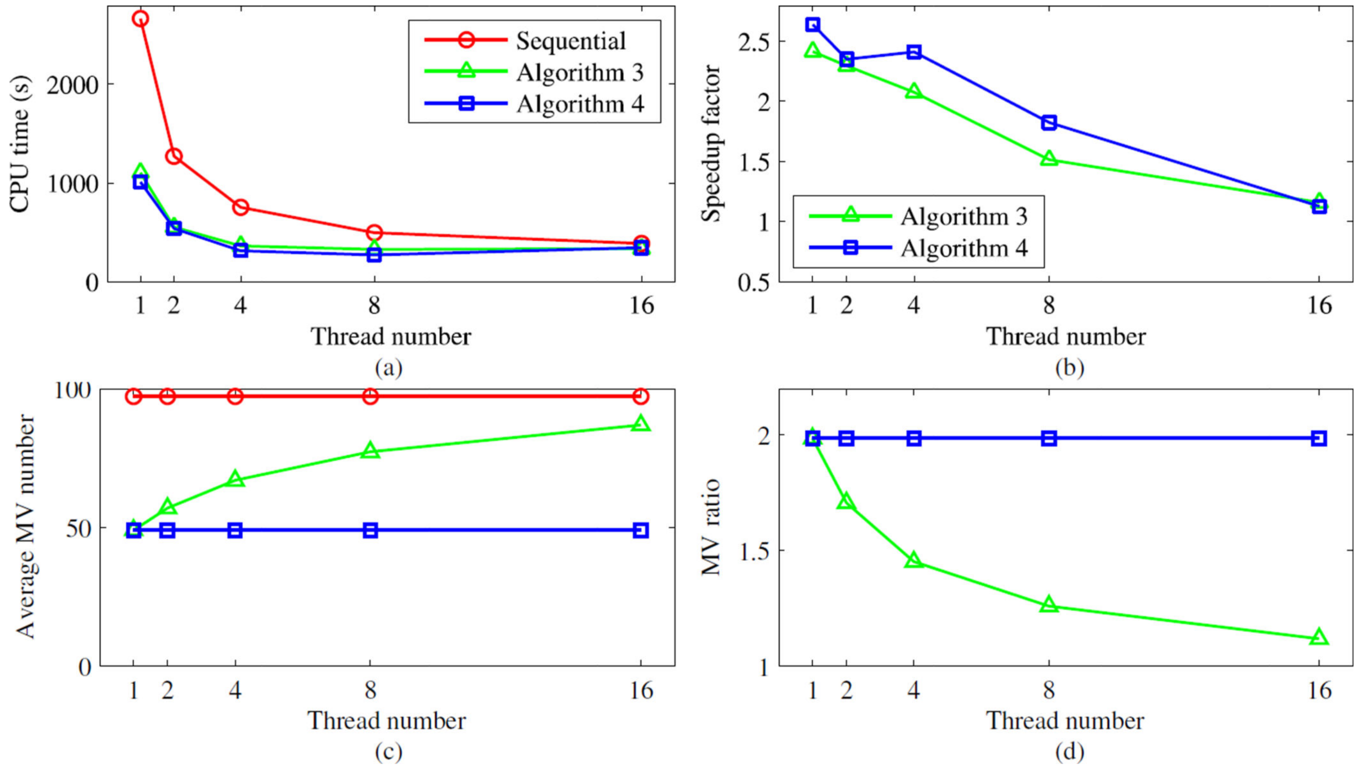


Fig 6.

CPU time, speedup factor, MV number and MV ratio comparison respect to thread number in normal tissue case (μ_a = 0.1cm^{-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.

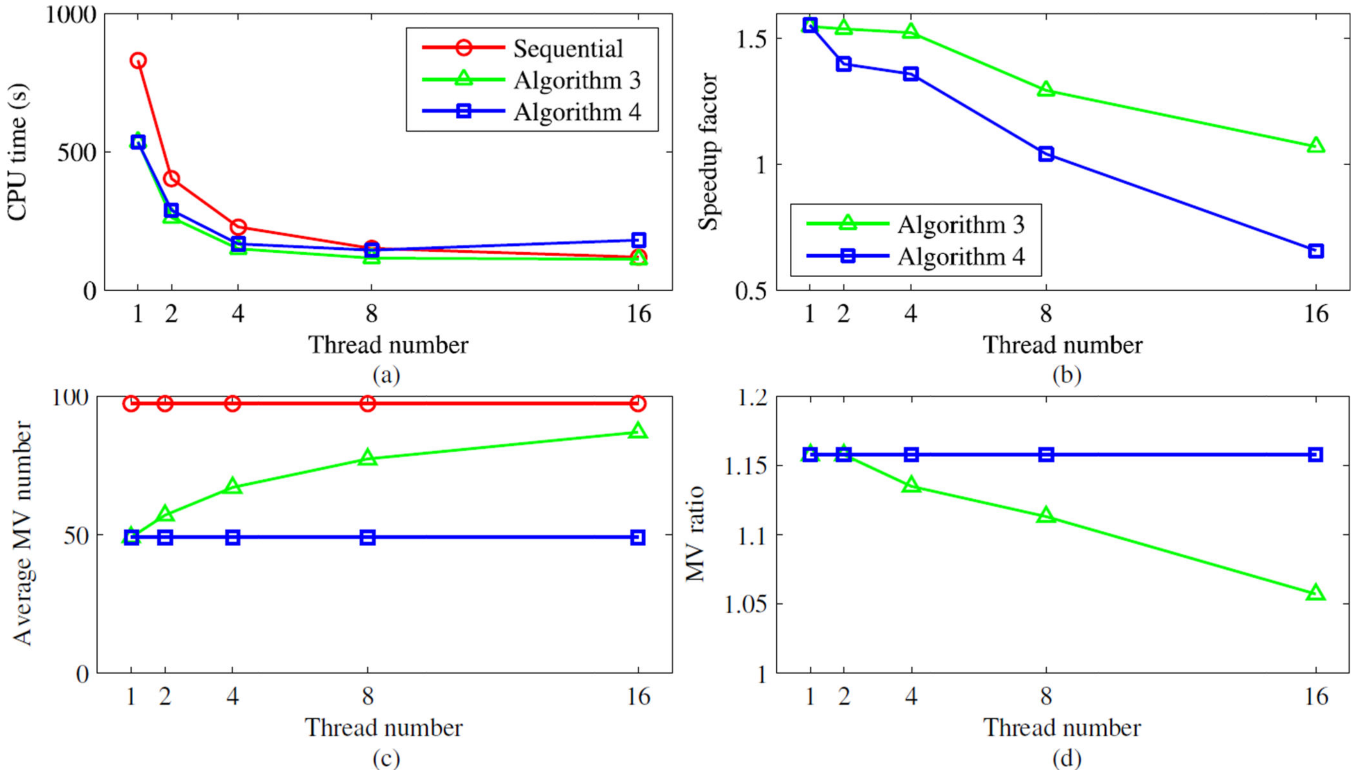


Fig 7.

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.

Table 1

The phantom settings of the 2D disk and the 3D cylinder

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

Table 2

Performance comparison between traditional and block BiCGStab algorithms on various combinations of (*g*, $\mu_s^{'}$)

$\mu_s^{'}\left[\mathrm{cm}^{-1}\right]$	***g***	**μ_s[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

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 4

Discrete ordinates employed to study the effects of angular discretization

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