An Improved Algorithm and Its Parallel Implementation for Solving a General Blood-Tissue Transport and Metabolism Model

Dexuan Xie[a,*], Ranjan K. Dash[b], and Daniel A. Beard[b]

[a]*Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201*
[b]*Biotechnology and Bioengineering Center, Department of Physiology, Medical College of Wisconsin, Milwaukee, WI 53226*

Abstract

Fast algorithms for simulating mathematical models of coupled blood-tissue transport and metabolism are critical for the analysis of data on transport and reaction in tissues. Here, by combining the method of characteristics with the standard grid discretization technique, a novel algorithm is introduced for solving a general blood-tissue transport and metabolism model governed by a large system of one-dimensional semilinear first order partial differential equations. The key part of the algorithm is to approximate the model as a group of independent ordinary differential equation (ODE) systems such that each ODE system has the same size as the model and can be integrated independently. Thus the method can be easily implemented in parallel on a large scale multiprocessor computer. The accuracy of the algorithm is demonstrated for solving a simple blood-tissue exchange model introduced by Sangren and Sheppard (Bull. Math. Biophys. 15:387-394, 1953), which has an analytical solution. Numerical experiments made on a distributed-memory parallel computer (an HP Linux cluster) and a shared-memory parallel computer (a SGI Origin 2000) demonstrate the parallel efficiency of the algorithm.

Key words: Blood-tissue transport and metabolism, BTEX algorithm, Parallel computation, System of first order semilinear PDEs, Initial-boundary value problems
2000 MSC: 5Y05, 93A30

1. Introduction

Oxidative energy metabolism in cells, including ATP synthesis from carbohydrate and fatty acid substrates, is coupled to the delivery of oxygen by the microcirculation. Since oxygen is highly extracted from the blood and is consumed rapidly in many tissues, oxygen tension in cells varies spatially—from approximately 100 mmHg near the inflow into capillaries to as low as 10-20 mmHg near the outflow in metabolically active tissues such as the myocardium [3, 24]. Therefore, the transport of oxygen, and other key solutes, to tissue is an inherently spatially distributed process, captured by partial differential transport equations. Simulation of blood-tissue solute exchange finds applications in analyzing experimental data on cell and tissue/organ function during changing physiological conditions such as ischemia (low blood flow), hypoxia (low oxygen supply), and exercise (high energy demand) [4, 6, 5, 7, 10]. Such modeling is also useful in the analysis of experimental data from tracer studies involving tracer-labeled ^{15}O- and ^{17}O-oxygen and other substrates. Understanding the transport of tracer ^{15}O oxygen, and its metabolic byproduct tracer ^{15}O water, is important in interpreting results from positron emission tomography imaging used to discover functional information relating to local perfusion and metabolism in the heart [11, 20]. Similarly, understanding the effects of changing blood flow on local bulk oxygen concentration is important in interpreting the physiological significance of blood-oxygen-level-dependent contrast magnetic resonance imaging [17] and ^{17}O nuclear magnetic resonance imaging [23] of the working brain.

*Corresponding author. E-mail: dxie@uwm.edu; Telephone: (414) 229-3528; Fax: (414) 229-4907

The classical Krogh cylinder model [18] (a single cylindrical tissue supplied by a single cylindrical capillary) has been the basis for most of the theoretical studies on microcirculatory oxygen transport [7, 10, 11, 20, 24] in which the transport of oxygen is governed by a system of one-dimensional nonlinear partial differential equations (PDEs). The governing PDEs can be parabolic or hyperbolic depending on whether molecular diffusion in the direction of blood flow is or is not considered. Furthermore, when oxygen transport is coupled with the cellular metabolic processes, the governing PDEs can be highly nonlinear and the number of governing PDEs can increase to the order of tens or even hundreds depending on the number of chemical species modeled, resulting in a computationally expensive model [4, 10, 31, 32, 33]. Therefore, fast efficient numerical solutions of the governing PDEs are important for the analysis of experimental data, which requires repeated computation of solutions of the governing PDEs and comparison of the experimental data to the corresponding model outputs.

Currently, the most useful numerical procedure for solving the one-dimensional microcirculatory advection-diffusion-reaction models is the Lagrangian sliding fluid element algorithm (also known as the blood-tissue exchange or BTEX algorithm) of Bassingthwaighte et al. [2]. This algorithm works well for small-scale systems involving only few chemical species, as in the models of Bassingthwaighte and co-workers [7, 10, 11, 20]. However, this algorithm works considerably slowly for large-scale systems such as for the Beard model of coupled blood-tissue transport and metabolism [4], and is difficult to parallelize. Consequently, the algorithm is computationally expensive for the analysis of experimental data with complex transport-metabolism models. The present work was motivated in part to improve and generalize this BTEX algorithm such that the resulting generalized BTEX algorithm can be easily implemented in parallel on a large scale multiprocessor computer, making it suitable for rapid numerical solutions of large-scale one-dimensional microcirculatory advection-diffusion-reaction models. Specifically, rapid numerical solutions of the recently developed coupled blood-tissue transport-metabolism model of Beard [4] are sought, which simulates advective oxygen transport and oxidative energy metabolism by a system of 30 semilinear first order PDEs. This model does not account for the axial diffusion of chemical species, has the constant coefficient of the spatial derivative term (i.e., a constant axial blood velocity) and the nonlinear reaction terms that arise from the contribution of cellular metabolic processes (the source or sink terms in the PDEs), and does not produce any singularity problem that may occur in a system of semi-linear PDE of first order [21].

Here, such coupled blood-tissue transport-metabolism models are formulated as a general blood-tissue exchange (BTEX) model governed by a system of semilinear PDEs of first order, along with the boundary and initial conditions. We then prove that this general BTEX model has a unique solution under the assumptions that both the boundary and initial value functions are continuous and bounded, and the nonlinear reaction terms are continuous and satisfy a Lipschitz condition in the solution. Based on this analysis, a novel numerical algorithm is developed to numerically solve this general BTEX model using the characteristic-line and standard mesh discretization techniques. The key part of this algorithm is to approximate the model as a group of independent ordinary differential equation (ODE) systems at the discretized spatial mesh points such that each ODE system has the same size as the model and can be solved independently by a standard ODE numerical integrator. Due to this, this algorithm can be simply implemented on a parallel computer by a domain decomposition strategy [30]. For clarity, this new algorithm will be called the generalized BTEX algorithm since it is designed particularly for numerically solving large-scale general BTEX models involving transport and metabolism, and is more efficient than the original BTEX algorithm of Bassingthwaighte et al. [2].

Semilinear PDE problems of the sort that arise from coupled blood-tissue transport-metabolism modeling can also be solved by the method of lines, a general procedure for solving time-dependent one-dimensional PDEs [15, 27, 28]. In the method of lines, the model is approximated as only one system of ODEs in which the size becomes nN, where N is the number of differential equations in the BTEX model, and n is the number of spatial grid points required for spatial approximation. The number n may need to be large to achieve sufficient accuracy in the spatial approximation. Since most stiff ODE solvers are implicit, requiring solving large-scale nonlinear algebraic systems at each time step, solving such large ODE system may become computationally costly and challenging in parallel implementation. In contrast, the generalized BTEX algorithm approximates the model as n independent small ODE systems with size N (e.g., $N = 30$ in the Beard model [4]), it is a natural parallel algorithm, and each small ODE system can be efficiently solved by using a high-order stiff ODE integrator such as an one-step implicit Runge-Kutta method [13, 14] or a multi-step BDF (backward differentiation formula) method [9].

To study the accuracy of the generalized BTEX algorithm, a simple BTEX model introduced by Sangren and Sheppard [26], which can be solved analytically to obtain the exact solution, is analyzed here. Both the generalized

BTEX algorithm and the method of lines were applied to solve the Sangren-Sheppard model and the numerical solutions were compared to the analytical solution. Results show that the numerical solutions from the generalized BTEX algorithm match the analytical solution with a much higher accuracy than those obtained from the method of lines.

Finally, a parallel program package was developed in Fortran 77 based on the Livermore ODE Solver DLSODES [16, 25] and MPI (Message Passing Interface) library [1]. This package was applied to Beard model simulating advective oxygen transport and oxidative energy metabolism by a system of 30 one-dimensional semilinear first order PDE equations [4]. The program package was tested on a distributed-memory parallel computer (an HP Linux cluster at the Medical College of Wisconsin) and a shared-memory parallel computer (an SGI Origin 2000 at the University of Wisconsin-Milwaukee) with up to 16 processors. Numerical results demonstrate good parallel performance on both computers.

The remainder of the paper is organized as follows. Section 2 introduces the general BTEX model. Section 3 describes the generalized BTEX algorithm. Section 4 discusses its parallel implementation. Section 5 studies the accuracy of the numerical solutions produced by the method. Section 6 discusses the parallel performance of the algorithm. Finally, conclusions are made in Section 7.

2. A General Blood-Tissue Exchange Model

In this section, we define a general blood-tissue exchange (BTEX) model involving transport and metabolism based on the following basic assumptions: (1) concentrations of chemical species vary primarily along the capillary length within the capillaries, interstitial fluid (ISF) space, and parenchymal cells domain; (2) advective transport happens only in the capillary region; and (3) transport between regions (capillaries, ISF, and parenchymal cells) and cellular metabolism is governed by suitable nonlinear functions of species concentrations.

Let $\{u_i(x,t)\}_{i=1}^{N_1}$ and $\{u_j(x,t)\}_{j=N_1+1}^{N}$ be the functions representing the concentrations of chemical species in the capillary and tissue (ISF or parenchymal cells) regions, respectively. They are defined on the domain $D = \{(x,t)|0 < x < L, 0 < t < T\}$, where L is the length of capillary, and T is the length of time. Then the general blood-tissue exchange model is defined by

$$\begin{cases} \frac{\partial u_i(x,t)}{\partial t} + a\frac{\partial u_i(x,t)}{\partial x} &= f_i(t,x,u), \quad i = 1,2,\ldots,N_1, \\ \frac{\partial u_j(x,t)}{\partial t} &= f_j(t,x,u), \quad j = N_1+1, N_1+2,\ldots,N, \\ u_k(x,0) &= \varphi_k(x), \quad k = 1,2,\ldots,N \text{ (initial conditions)}, \\ u_i(0,t) &= g_i(t), \quad i = 1,2,\ldots,N_1 \text{ (boundary conditions)}, \end{cases} \tag{1}$$

where $a > 0$ is the speed of blood transport in the capillary, $u = (u_1, u_2, \ldots, u_N)$, f_j is a nonlinear function representing either a passive permeation between the domains or a flux through a biochemical reaction in the cellular domain, or a combination of both, φ_k is a given initial function, and g_i is a given boundary function at $x = 0$. In small blood vessels and capillaries (microcirculation), the blood is usually squeezed through the system, in which the flow can be approximated as plug flow. Therefore, the speed of blood transport in such systems is usually considered constant [2, 3]. For example, in [4], the speed a of blood transport was estimated by the formula $a = \frac{\rho FL}{V_1}$, where F is the blood flow in the capillary expressed in the units of volume per unit time per unit mass of the tissue, ρ is the tissue density in mass per unit volume, and V_1 is the volume of the capillary region.

The model proposed in [4] is a special case of Equation (1) with $N_1 = 1$ and $N = 30$, considering only oxygen transport in the capillary region. In that model, three equations describe oxygen transport within three-regions of the cardiac tissue — the capillary, the ISF region, and the cellular (myocyte) region, while the other 27 equations describe cellular energy metabolism. In this model, the transport of other species (such as glucose, fatty acids, and amino acids) was not considered within the capillary blood region. Additional species in the advecting (blood) region can be studied with models with $N_1 > 1$.

The independent variables x and t usually do not appear explicitly in the functions f_j, and f_j may be an implicit function of some unknown functions u_i. For example, in [4], the nonlinear term f_1 is given by

$$f_1(u_1, u_2) = -c_1 P - c_2 u_2, \tag{2}$$

where c_1 and c_2 are two given constants, and P is a function of u_1 defined implicitly by the nonlinear equation

$$u_1 = \alpha P + \beta \frac{P^\sigma}{P^\sigma + \gamma^\sigma}, \tag{3}$$

where α is the oxygen solubility coefficient in blood, β is the product of the hematocrit with the concentration of oxygen binding sites in red blood cells (hemoglobin), γ is the partial pressure of oxygen for half saturation of hemoglobin, and σ is the Hill exponent. Here P indicates the oxygen partial pressure in the capillary. In [4], the constants were set as $\alpha = 1.3 \times 10^{-6}$ M mmHg^{-1}, $\gamma = 30.0$ mmHg, $\beta = 0.45 \times 0.0213$ M, and $\sigma = 2.5$.

To solve Equation (3) for P, we rewrite it in the standard form

$$\alpha P^{\sigma+1} + (\beta - u_1)P^\sigma + \alpha\gamma^\sigma P - \gamma^\sigma u_1 = 0, \tag{4}$$

and then solve it by the Newton-Raphson iterative method:

$$P^{(k+1)} = P^{(k)} - \frac{\psi(P^{(k)})}{\psi'(P^{(k)})}, \quad k = 0, 1, 2, \ldots,$$

where $P^{(0)}$ is an initial guess (e.g., $P^{(0)} = 0$), $\psi(P)$ is a function of P defined by the left-hand side of Equation (4), and $\psi'(P)$ denotes its derivative. The kth iterate $P^{(k)}$ is selected as a value of P for calculating f_1 in Equation (2) provided that it satisfies the termination rule

$$|\psi(P^{(k)})| < \epsilon_r |\psi(P^{(0)})| + \epsilon_a,$$

where ϵ_r and ϵ_a are the relative and absolute error tolerances, respectively (e.g., $\epsilon_a = \epsilon_r = 10^{-6}$).

The general BTEX model actually consists of N_1 semilinear hyperbolic equations and $N - N_1$ semilinear first order PDEs containing only the derivatives with respect to time t. Its solution existence and uniqueness can be proved similarly as the case of semilinear first order hyperbolic systems [21] under the following assumptions: $g_k(t)$ and $\varphi_k(x)$ are continuous and bounded on $[0, T]$ and $[0, L]$, respectively, and f_k is continuous on the domain $D = \{(x, t, u) | 0 \le t \le T, 0 \le x \le L, u \in R^N\}$ and satisfies the following Lipschitz condition

$$|f_k(x, t, u) - f_k(x, t, v)| \le \mathcal{L}\|u - v\| \quad \text{for all } x \in [0, L], t \in [0, T], u, v \in R^N, k = 1, 2, \ldots, N, \tag{5}$$

where $\mathcal{L}$ is Lipschitz constant, $u = u(x, t)$, $v = v(x, t)$, and $\|\cdot\|$ is a fixed norm on R^N. The key step of the proof is to use the method of characteristics to reduce the boundary initial value problem (1) to integral equations and then use the Picard method to define sequences of functions that converge to the solutions. The details of the proof is beyond the scope of the present paper and will be reported in a subsequent paper.

3. Generalized BTEX Algorithm for Solving the General BTEX Model

This section describes the generalized BTEX algorithm for solving the general BTEX model of Equation (1). As described in Section 2, the blood speed a is considered as a constant and the source functions f_j are expressed as $f_j(u)$.

The generalized BTEX numerical algorithm is defined on a special grid mesh of the domain D with the spatial and time step sizes h and τ satisfying $\tau = h/a$. By using $h = L/n$, the grid points (x_j, t_i) of this grid mesh are defined by $x_j = jh$ and $t_i = i\tau$ such that

$$0 = t_0 < t_1 < t_2 < \ldots < t_m \le T, \quad 0 = x_0 < x_1 < x_2 < \ldots < x_n = L, \tag{6}$$

where n is a given positive integer, and m is the integer closest to the real number $(aTn)/L$. See Figure 1 for an illustration of the grid mesh with $n = 4$ and $m = 6$.

Clearly, for $0 \le t \le t_1$, a characteristic line that starts at $x = x_k$ can be expressed by a linear function of t in the form $x = at + x_k$ for $k = 0, 1, 2, \ldots, n - 1$. When the functions $u_i(x, t)$ for $i = 1, 2, \ldots, N_1$ are restricted to these characteristic lines, we have that

$$\frac{du_i(at + x_k, t)}{dt} = a\frac{\partial u_i(at + x_k, t)}{\partial x} + \frac{\partial u_i(at + x_k, t)}{\partial t}.$$

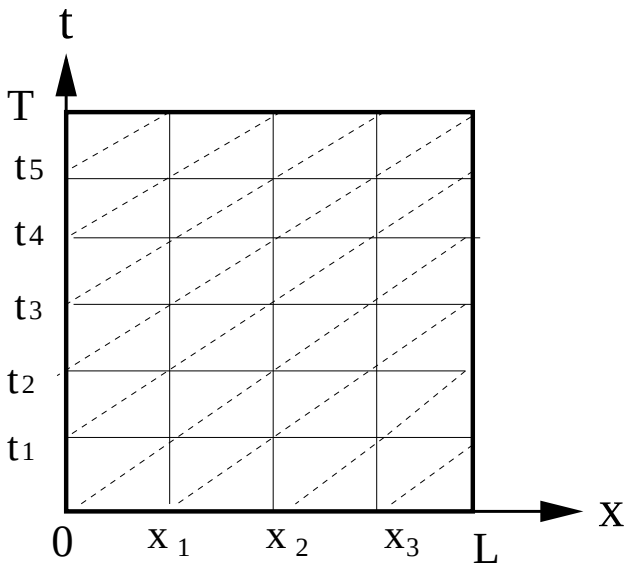

Figure 1: An example of the grid mesh ($n = 4$ and $m = 6$) used by the traditional BTEX method (Lagrangian sliding fluid element algorithm of Bassingthwaighte et al. [2].

Thus, the first N_1 equations of (1) are converted to the ODEs without any errors:

$$\frac{du_i(at + x_k, t)}{dt} = f_i(u_1(at + x_k, t), u_2(at + x_k, t), \ldots, u_N(at + x_k, t)), \tag{7}$$

where $i = 1, 2, \ldots, N_1$ and $0 \le t \le t_1$. Here we have taken an advantage of the characteristic line approach for solving one-dimensional hyperbolic PDEs [29].

The other equations of (1) do not involve any partial derivatives with respect to the spatial variable x. They can be converted to ODEs along the vertical mesh lines $x = x_k$:

$$\frac{du_j(x_k, t)}{dt} = f_j(u_1(x_k, t), u_2(x_k, t), \ldots, u_N(x_k, t)) \tag{8}$$

for $0 \le t \le t_1$, where $j = N_1 + 1, N_1 + 2, \ldots, N$.

Next, we can obtain the following fact: If $u_j(x, t)$ is Lipschitz-continuous in the sense that there exists a constant $\mathcal{L} > 0$ such that $|u_j(x, t) - u_j(y, t)| \le \mathcal{L}|x - y|$ for all $x, y \in (0, L)$, then for any $t \in [t_i, t_{i+1}]$,

$$|u_j(x_k, t) - u_j(a(t - t_i) + x_k, t)| \le \mathcal{L}\, a|t - t_i| \le \mathcal{L}\, a\tau = \mathcal{L}\, h,$$

where $j = 1, 2, \ldots, N$, and $\tau = h/a$ has been used.

Based on the above fact, we claim that $u_j(a(t - t_i) + x_k, t)$ can be well approximated by $u_j(x_k, t)$ for $t_i \le t \le t_{i+1}$ if h is sufficiently small. Hence, for $i = 0$, $t_0 = 0$, and $u_j(at + x_k, t)$ can be approximated as $u_j(x_k, t)$ for $j = N_1 + 1, N_1 + 2, \ldots, N$; thus, Equation (7) is modified as

$$\frac{du_i(at + x_k, t)}{dt} = f_i(u_1(at + x_k, t), \ldots, u_{N_1}(at + x_k, t), u_{N_1+1}(x_k, t), \ldots, u_N(x_k, t)) \tag{9}$$

for $0 \le t \le t_1$. Naturally, the functions $u_i(at + x_k, t)$ for $i = 1, 2, \ldots, N_1$ defined in Equation (9) can be employed to modify Equation (8) as

$$\frac{du_j(x_k, t)}{dt} = f_j(u_1(at + x_k, t), \ldots, u_{N_1}(at + x_k, t), u_{N_1+1}(x_k, t), \ldots, u_N(x_k, t)). \tag{10}$$

Consequently, a combination of Equations (9) and (10) with the initial conditions $u_j(x_k, 0) = \varphi(x_k)$ gives the n initial value problems that provide a numerical approximation to the general model of Equation (1) on $0 \le t \le t_1$ and $0 \le x \le L$. These n initial value problems can be expressed in the vector form: For $k = 0, 1, 2, \ldots, n-1$,

$$\begin{cases} \dfrac{d\hat{u}_k(t)}{dt} = f(\hat{u}_k), & \text{for } 0 < t \le t_1, \\ \hat{u}_k(0) = \varphi(x_k), \end{cases} \tag{11}$$

where $\varphi(x) = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_N(x))$, $\hat{u}_k(t) = (u_1(at+x_k, t), \ldots, u_{N_1}(at+x_k, t), u_{N_1+1}(x_k, t), \ldots, u_N(x_k, t))$, and $f(\hat{u}_k) = (f_1(\hat{u}_k), f_2(\hat{u}_k), \ldots, f_N(\hat{u}_k))$. From the construction process of Equation (11) it is easy to see that these n initial value problems are independent of each other. Hence, they can be solved independently by a numerical ODE algorithm, yielding an approximation of $\hat{u}_k(t)$ at $t = t_1$.This completes the description of the first time step of the generalized BTEX method.

To illustrate the independence of these ODE systems, we display an example of (11) with $N = 2$, $N_1 = 1$, and the grid mesh with $h = L/4$ (i.e., $n = 4$) as shown in Figure 1. The general BTEX model problem is now approximated as the following four ODE systems: For $k = 0, 1, 2, 3$,

$$\begin{cases} \dfrac{du_1(at + x_k, t)}{dt} = f_1(u_1(at + x_k, t), u_2(x_k, t)), \\ \dfrac{du_2(x_k, t)}{dt} = f_2(u_1(at + x_k, t), u_2(x_k, t)), & 0 < t \le t_1, \\ u_1(x_k, 0) = \varphi_1(x_k), \\ u_2(x_k, 0) = \varphi_2(x_k), \end{cases}$$

where $x_k = kh$ for $k = 0, 1, 2, 3$. Clearly, these four ODE systems are defined on four different sets of line segments with each set containing two different line segments. For example, in the ODE system with $k = 0$, the first equation is defined on the line segment $x = at$ for $0 \le t \le t_1$ while the second one on the vertical line segment $x = 0$ for $0 \le t \le t_1$, solving them gives numerical solutions $u_1(x_1, t_1)$ and $u_2(0, t_1)$. For the ODE system with $k = 1$, the first equation is defined on the line segment $x = at + x_1$ for $0 \le t \le t_1$ while the second one on the vertical line segment $x = x_1$ for $0 \le t \le t_1$. Clearly, they are completely independent from the equations of the first ODE system. Similarly, we can see that they are independent from other two ODE systems too. Hence, they can be solved independently to get numerical solutions $u_1(x_2, t_1)$ and $u_2(x_1, t_1)$.

In the i-th time step of the generalized BTEX method for $i = 1, 2, \ldots, m-1$, the general BTEX model of Equation (1) is solved approximately on $t_i \le t \le t_{i+1}$ and $0 \le x \le L$ as n independent ODE initial value problems. Let $U^i_{j,k}$ denote a numerical value of $u_j(x_k, t_i)$, which has been computed in the previous time step over $t_{i-1} \le t \le t_i$ with $i \ge 1$, and the vector function $\hat{u}_k(t)$ be defined by

$$\hat{u}_k(t) = (u_1(a(t - t_i) + x_k, t), \ldots, u_{N_1}(a(t - t_i) + x_k, t), u_{N_1+1}(x_k, t), \ldots, u_N(x_k, t)). \tag{12}$$

Using $U^i_{j,k}$ and the boundary conditions at $x = 0$ of (1), we can construct the initial conditions at $t = t_i$ as below:

$$\hat{u}_k(t_i) = \omega^i_k,$$

where ω^i_k are determined by the general formula:

$$\omega^i_k = \begin{cases} \varphi(x_k) & \text{if } i = 0 \\ (g_1(t_i), g_2(t_i), \ldots, g_{N_1}(t_i), U^i_{N_1+1,0}, U^i_{N_1+2,0}, \ldots, U^i_{N,0}) & \text{if } k = 0 \text{ and } i \ge 1 \\ (U^i_{1,k}, U^i_{2,k}, U^i_{3,k}, \ldots, U^i_{N,k}) & \text{if } k \ge 1 \text{ and } i \ge 1. \end{cases} \tag{13}$$

Similar to the construction of Equation (11), we then can obtain the n independent ODE initial value problems in vector form as follows: For $k = 0, 1, 2, \ldots, n-1$,

$$\begin{cases} \dfrac{d\hat{u}_k(t)}{dt} = f(\hat{u}_k) & \text{for } t_i < t \le t_{i+1}, \\ \hat{u}_k(t_i) = \omega^i_k, \end{cases} \tag{14}$$

where $i = 0, 1, 2, \ldots, m-1$, and the initial values ω_k^i are defined in (13).

Since $a(t_{i+1} - t_i) + x_k = a\tau + x_k = h + x_k = x_{k+1}$, the value of the vector function $\hat{u}_k(t)$ defined in Equation (12) at $t = t_{i+1}$ becomes

$$\hat{u}_k(t_{i+1}) = (u_1(x_{k+1}, t_{i+1}), \ldots, u_{N_1}(x_{k+1}, t_{i+1}), u_{N_1+1}(x_k, t_{i+1}), \ldots, u_N(x_k, t_{i+1})).$$

Thus, solving Equation (14) produces the following numerical values:

$$U_{j,k+1}^{i+1} \text{ for } j = 1, 2, \ldots, N_1, \quad \text{and} \quad U_{j,k}^{i+1} \text{ for } j = N_1 + 1, N_1 + 2, \ldots, N. \tag{15}$$

In other words, the first N_1 numerical values have been slid one step forward from x_k to x_{k+1} while the time is changed from t_i to t_{i+1}. This phenomenon was used in [2] and referred to as the lagrangian sliding fluid element.

4. Parallel Implementation of Generalized BTEX Algorithm

Computer implementation of the generalized BTEX algorithm is straightforward. On a sequential computer, we assume that one global two dimensional array, $U[1 : N, 0 : n-1]$, is used to store both input and output numerical values in solving the ODE initial value problems of Equation (14) at each time step $i = 0, 1, 2, \ldots, m-1$. Most ODE solvers are designed in this way. In the input step at $t = t_{i-1}$, we set $U[j,k] := U_{j,k}^{i-1}$ for $j = 1, 2, \ldots, N$ and $k = 0, 1, 2, \ldots, n-1$. Here the symbol ":=" denotes the assignment operation in the computer program. Because of Equation (15), in the output step at $t = t_i$, U will hold the following numerical values for $k = 0, 1, 2, \ldots, n-1$:

$$U[j,k] := U_{j,k+1}^{i} \text{ for } j = 1, 2, \ldots, N_1, \quad \text{and} \quad U[j,k] := U_{j,k}^{i} \text{ for } j = N_1 + 1, N_1 + 2, \ldots, N. \tag{16}$$

Hence, an adjustment of the array U has to be done to get the initial values ω_k^i as given in Equation (13) for solving Equation (14) from t_i to t_{i+1}. As the result of (16), in the input step at $t = t_i$, only need the first N_1 rows of array U be updated:

$$U[j,0] := g_j(t_i) \text{ and } U[j, 1 : n-1] := U[j, 0 : n-2] \quad \text{for } j = 1, 2, \ldots, N_1. \tag{17}$$

The generalized BTEX method can be easily implemented in parallel based on either a shared-memory programming model in OpenMP [8] or a distributed-memory programming model in MPI [1, 12]. With OpenMP, a sequential program of the generalized BTEX method can be easily parallelized by only adding a compiler directive before the "parallelizable" spatial step loop of k for $k = 0, 1, 2, \ldots, n-1$, which involves the numerical solutions of n independent ODE initial value problems. To efficiently implement the generalized BTEX method in MPI on a distributed-memory parallel computer (including a cluster of PCs), we need to explore its data-distribution and data-communication explicitly as shown below. Such a MPI program works well on a shared-memory parallel computer too.

Let p be the number of processors to be used in the parallel implementation. In each time step of i for $i = 0, 1, 2, \ldots, m-1$, the n independent ODE problems of the k-loop can be partitioned into p independent groups such that each group contains almost the same number of ODE problems based on a simple domain decomposition strategy. That is, the $(n+1)$ spatial grid points $\{x_j\}_{j=0}^n$ of the spatial domain $[0, L]$ are partitioned into p equally-sized subsets, $\Omega_k = \{x_j | j = n_{k-1} + 1, n_{k-1} + 2, \ldots, n_k\}$ for $k = 1, 2, \ldots, p$, such that the ODE initial value problems related to Ω_k are assigned to processor k for calculation. Here $n_k = k(n+1)/p - 1$ is assumed to be a positive integer. To reduce the memory cost, the global array $U[1 : N, 0 : n]$ is also split into p sub-arrays such that processor k contains only sub-array $U[1 : N, n_{k-1} + 1 : n_k]$ for storing the local numerical values. Furthermore, the array updating job of Equation (17) is also split into p parts as given below:

Part 1: do $U[j,0] := g_j(t_i)$ and $U[j, 1 : n_1] := U[j, 0 : n_1 - 1]$ for $j = 1, 2, \ldots, N_1$.

Part k: do $U[1 : N_1, n_{k-1} + 1 : n_k] := U[1 : N_1, n_{k-1} : n_k - 1]$ for $k = 2, 3, \ldots, p$.

However, the updating job of Part k on processor k involves the entries $U[1 : N_1, n_{k-1}]$ that are located on processor $k-1$ for $k = 2, 3, \ldots, p$ since processor k contains only the local array $U[1 : N, n_{k-1} + 1 : n_k]$. The update cannot be completed until the entries $U[1 : N_1, n_{k-1}]$ are calculated and sent out by processor $k-1$ and received by processor k. This is the only place that requires interprocessor data communication operations in the parallel implementation of the generalized BTEX method. Hence, the total amount of data required for interprocessor communications is small. But,

since the work required to solve the different ODE initial value problems may be different, different processors may produce different CPU times to finish the job even though they are assigned the same number of ODE initial value problems. This potentially unbalanced work load problem may affect the parallel performance of the generalized BTEX method. One possible solution is to solve each ODE system by a high order one-step stiff ODE solver, such as the Radau IIA method [13, 14], in a fixed number of steps.

For clarity, a piece of pseudo code in MPI is given in Algorithm 1 to illustrate the parallel implementation of the generalized BTEX algorithm. It is written in the SPMD (single-program, multiple-date) style. That is, the same program code is implemented on all available processors while each processor has different data and thread of controls. The SPMD style is widely used in practice for shared or distributed memory. In Algorithm 1, we mention the names of MPI functions that we used in our MPI program but ignore the details since the details of their usages can be found in [1, 12] or on the MPI forum web at *http://www.mpi-forum.org*.

Algorithm 1 (Parallel BTEX Pseudo-program in MPI). *Let the p processors be labeled from 0 to* $p-1$*, n be selected to make the ratio* $N_b = (n+1)/p$ *a positive integer, and* ODEsolver(Y, t_{in}, t_{out}) *denote an ODE solver program routine. Here* $Y = Y[1:N]$ *is a real array with N entries, which holds the initial solution* $\hat{u}_k(t_{in})$ *in the input and the final solution* $\hat{u}_k(t_{out})$ *in the output of the ODE solver program routine.*

1. *Create a new communicator for the p processors with the one-dimensional Cartesian topology by MPI function MPI_CART_CREATE, get the rank* M_a *of calling processor in the new communicator by MPI_COMM_RANK, and define the left and right neighboring processors of calling processor by MPI_CART_SHIFT.*
2. *Set* $i = 0$*, define the k-loop for* $k = N_s$ *to* N_e*, and input the initial values by* $U[1:N,k] := \varphi(x_k)$ *for* $k = N_s$ *to* N_e*, where* $N_s = 1 + M_e N_b, N_e = (M_e+1)N_b$*, and* $N_e = n$ *if* $M_e = p-1$.
3. *Do the following k-loop to solve the* N_b *ODE initial value problems of (14) on time interval* $[t_i, t_{i+1}]$*:*
   for $k = N_s$ *to* N_e*,*
   Set $t_{in} := t_i, t_{out} := t_{i+1}$*, and* $Y[1:N] := U[1:N,k]$*;*
   Call ODEsolver(Y, t_{in}, t_{out}) *to get solution Y at* $t = t_{out}$*;*
   Save the solution by $U[1:N,k] := Y[1:N]$.
   end for (k)
4. *Adjust the value of* $U[1:N, N_s:N_e]$ *according to Parts 1 and 2 by doing the following steps:*
   (1) Send $U[1:N_1, N_e]$ *to the right neighboring processor by MPI_SEND;*
   (2) Receive $U[1:N_1, N_e]$ *from the left neighboring processor and save it to a buffer array* $B(1:N_1)$ *by MPI_RECV;*
   (3) Adjust the value of U (i.e., sliding one entry) by $U[1:N_1, N_s+1:N_e] := U[1:N_1, N_s:N_e-1]$*;*
   (4) Redefine the value of $U[1:N_1, N_s]$ *by setting* $U[j,1] := g_j(t_{out})$ *for* $j = 1$ *to* N_1 *if* $M_e = 0$*, and* $U[1:N_1, N_s] := B[1:N_1]$ *if* $M_e \geq 1$.
5. *Increase i by 1 and go to Step 3 if* $i \leq m-1$*; otherwise, exit the i-loop of time.*

In summary, the generalized BTEX method is a general procedure for solving the general BTEX model of Equation (1). It is defined on a particular grid mesh of (6), and solves the BTEX model of Equation (1) in m time steps. At each time step of the generalized BTEX method, n independent ODE initial value problems, as defined in Equation (14), are produced as an approximation to the BTEX model. Here each ODE system has the same size as the BTEX model and can be solved independently by a numerical ODE solver. Moreover, the method can be easily and efficiently implemented in parallel on a multiprocessor computer. Based on the spatial grid point partition, a good load balance can be simply achieved by assigning the same number of ODEs to each processor, and the parallel implementation of the generalized BTEX method only involves relatively little interprocessor communication. Since the most computing costs come from solving the ODEs, the performance of the method is strongly related to the efficiency of the numerical ODE solver.

5. Accuracy Testing of the Generalized BTEX Method

To verify the accuracy of the numerical solutions produced by the generalized BTEX method, we made numerical experiments on the BTEX method for solving the Sangren-Sheppard model [26], which is a special case of the general

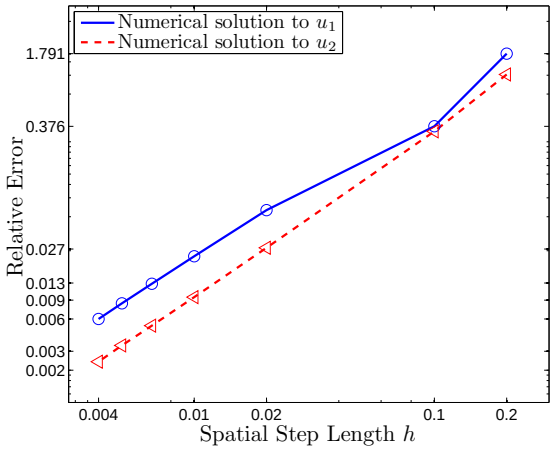

Figure 2: The relative error of the generalized BTEX method for solving the Sangren-Sheppard model is a increasing function of the spatial step length h.

BTEX model of Equation (1) with $N = 2$ and $N_1 = 1$. The Sangren-Sheppard model has the analytical solution in terms of the first-order modified Bessel function of the first kind. A comparison of our numerical solutions with those obtained from the method of lines was also made since the method of lines is a popular procedure for solving time-dependent PDE problems [15, 28, 27].

The Sangren-Sheppard model can be described as follows:

$$\begin{cases} \frac{\partial u_1}{\partial t} &= -a\frac{\partial u_1}{\partial x} - \frac{P_s}{V_1}(u_1 - u_2), \\ \frac{\partial u_2}{\partial t} &= +\frac{P_s}{V_2}(u_1 - u_2) \end{cases} \tag{18}$$

where $u_1(x,t)$ and $u_2(x,t)$ are the concentrations in the capillary blood and interstitial fluid (ISF) spaces, a is the blood velocity in the capillary, P_s is the permeability-surface area product for the exchange between the capillary and ISF, and V_1 and V_2 are the volumes of distribution of the given solute in the blood and ISF.

With the initial conditions $u_1(x,0) = 0$, $u_2(x,0) = 0$ and the boundary conditiion $u_1(0,t) = \frac{q_o}{F}\delta(t)$, the analytical solution of Equation (18) can be found as below:

$$u_1(x,t) = \begin{cases} \frac{q_o}{F}e^{-\frac{P_s}{aV_1}x}\delta\left(t - \frac{x}{a}\right) + \frac{q_o}{F}e^{-\frac{P_s}{V_2}\left[t-\left(1-\frac{V_2}{V_1}\right)\right]}\frac{P_s}{\sqrt{V_1V_2(vt/x-1)}}I_1(\beta), & t \geq x/a, \\ 0, & t < x/a, \end{cases}$$

and

$$u_2(x,t) = \begin{cases} \frac{q_o}{F}\frac{P_s}{V_2}e^{-\frac{P_s}{V_2}\left[t-\left(1-\frac{V_2}{V_1}\right)\right]}I_0(\beta), & t \geq x/a \\ 0, & t < x/a, \end{cases}$$

where $0 \leq x \leq L$, $0 \leq t \leq T$, I_0 and I_1 are the zeroth-order and first-order modified Bessel functions of the first kind [22], q_o is the finite mass injected into the capillary, $F = aV_1/L$, $\delta(t)$ is the Dirac delta function, and $\beta = 2P_s\left[\frac{(t-x/a)x}{aV_1V_2}\right]^{1/2}$. Here F is the blood flow to the tissue, and the boundary condition at $x = 0$, and the initial and boundary conditions simulate that a spike of finite mass q_o is injected into the capillary at position $x = 0$ at time $t = 0$.

If flow F and volumes V_1 and V_2 are expressed relative to total mass of tissue (for example, in units of ml·min^{-1}·g^{-1} and ml·g^{-1}, respectively), then the injected mass q_o is expressed in units of moles per mass of tissue. The finite mass injected at $x = 0$ results in an infinitely high concentration in an infinitely small volume. In the numerical tests reported in Figures 2 and 3, we set $L = 1$, $T = 1.5$, $a = 0.33$, $F = 1/60$, $V_1 = 0.05$, $V_2 = 0.2$, $P_s = 0.1$, $q_0 = 1.0$, and $h = L/n$ for $n = 10, 50, 100, 150, 200$ and 250.

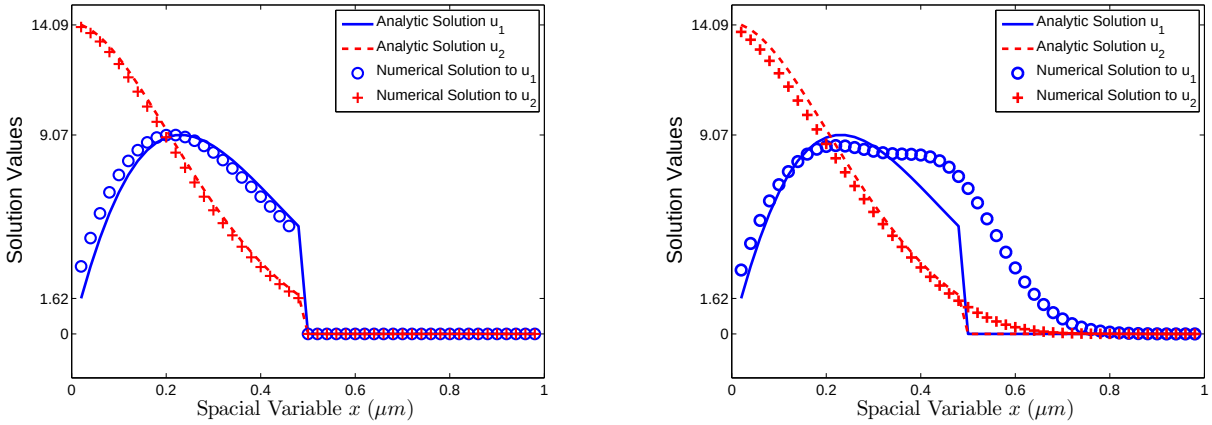

Figure 3: Comparison of the numerical solution of the generalized BTEX method (left figure) with that of the method of lines (right figure) for solving the Sangren-Sheppard model, a special example of (1) with $N_1 = 1$ and $N = 2$. Here the spatial grid size $h = 1/50$, $L = 1$ and $t = 1.5$.

Figure 2 displays the relative errors of the generalized BTEX method for solving the Sangren-Sheppard model as an increasing function of the spatial step length h. The accuracy of the method is second-order with respect to the step length h (or τ since $\tau = h/a$). The relative errors displayed in the figure were computed by using the formulas

$$\frac{\sqrt{\sum_{j=1}^{n}[U_{i,h}(j) - u_i(x_j,t)]^2}}{\sqrt{\sum_{j=1}^{n}[u_i(x_j,t)]^2}} \quad \text{for } i = 1, 2,$$

where $U_{i,h}(j)$ denotes the approximate value of $u_i(x_j, t)$, $h = L/n$, $x_j = jh$, and $t = 1.5$.

Figure 3 compares the numerical solutions produced by the generalized BTEX method with the analytical solutions and the numerical solutions produced by the method of lines. Here $h = L/50$. From the figures we can see that the numerical solutions by the generalized BTEX method match the analytical solutions, and that the generalized BTEX method has higher accuracy in approximating the Sangren-Sheppard model than the method of lines. In fact, at $t = 1.5$, both $u_1(x, t)$ and $u_2(x, t)$ have a jump discontinuous point and $u_1(x, t)$ has a impulse at $x = 0.5$. The generalized BTEX method resolves the impulse as a single point with total mass corresponding to the impulse part of $u_1(x, t)$. The impulse is not shown in the figure for the axis scale used. In the method of lines, numerical diffusion spreads the impulse and the method does a poor job in approximating the jump discontinuity.

6. Parallel Performance of the Generalized BTEX Method

We developed a parallel program package for the generalized BTEX method in FORTRAN 77 and MPI (Message Passing Interface) library [1], and tested it for solving the coupled blood-tissue transport and metabolism model proposed in [4]. In this package, the Livermore ODE Solver DLSODES [16, 25] is called for solving each ODE system associated with each spatial grid point. Since the source functions f_j of the model are defined implicitly in several levels of relation identities, it is difficult to obtain their explicit analytical expressions for partial derivatives. Hence, we used the method flag $MF = 222$ in DLSODES, the BDF (backward differentiation formula) method, which is also known as Gear's method or a variable order and multistep method [9] and a numerical Jacobian matrix estimated by finite difference formulas for solving each ODE. In addition, the following optional values were set in DLSODES for our numerical tests:

- Relative tolerance is set $rtol = 10^{-8}$ and the absolute tolerance is set $atol = 10^{-10}$ such that the local error of u_k is less than $rtol|U_k| + atol$, where U_k is a numerical solution in the kth step.
- The time step size dt of the ODE numerical solver is set to satisfy $10^{-3} < dt < 10^{-10}$.

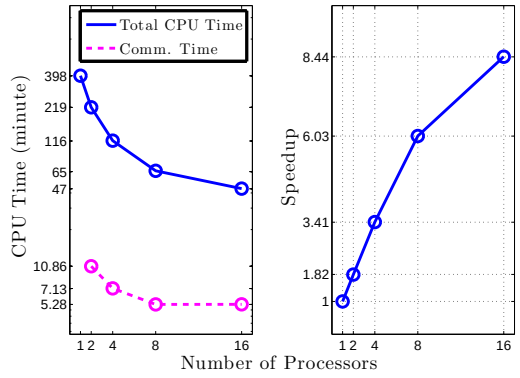

Figure 4: Parallel performance of the generalized BTEX algorithm on an SGI Origin-2000 machine.

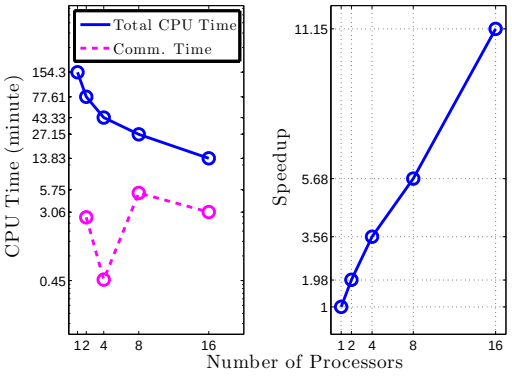

Figure 5: Parallel performance of the generalized BTEX algorithm on an HP Linux Cluster.

- The initial timestep is set to $dt = 10^{-10}$ in the BDF.

Numerical experiments were made on a distributed-memory parallel computer (the HP Linux Integrity Superdome Cluster at the Medical College of Wisconsin) and a shared-memory parallel computer (the SGI Origin 2000 at the University of Wisconsin-Milwaukee). Each node of the HP cluster is one HP R2600 Integrity server, which has two 1.3GHz Intel Itanium 2 processors, and 4Gbytes main memory. The cluster private interconnect is 1Gb Ethernet HP Procurve 2824 J4903A switch. Each processor of SGI Origin 2000 is R12000 with 400 MHz. The total main memory is 8Gbytes. Here we used the same boundary and initial value functions and the same data as the ones from [4] to evaluate the functions f_j. We set $N_1 = 1$, $N = 30$, $L = 550$, $T = 20$, and $a = 144.7875$. The grid mesh was made by $h = L/n$ with $n = 2048$ and $\tau = h/a = 1.8548 \times 10^{-3}$. This resulted in a total number of 10782 time steps ($m = 10782$) for the time interval $[0, 20]$. Hence, the total number of the ODE initial value problems to be solved in the generalized BTEX method was 22,081,536.

The parallel performances of the generalized BTEX method on the SGI Origin 2000 and HP Linux Cluster are shown in Figures 4 and 5, respectively. Total CPU time, total interprocessor data communication time, and associated speedups are shown in the figures. From the figures we see that the HP Cluster ran about 3 times faster than the Origin 2000, and had a better parallel performance; it only took about 13 minutes for the BTEX method to solve the model on 16 processors. It is interesting to note that the communication time is fluctuant on the HP Cluster while decreasing on the SGI Origin 2000. Although the total number of ODEs assigned to each processor was almost the same, the total CPU time and the total interprocessor communication time required to solve them on each processor were different. In the figures, the largest of them are reported. Also, we note that total communication time becomes a serious factor affecting the parallel efficiency when the number of processors is increased. Although interprocessor communication involves only one real number to be sent and received at each time step of generalized BTEX method, its cost becomes large after carrying out 10782 time steps due to the overhead costs of the sending and receiving operations. Even so, a speedup of 11.15 on 16 processors represents a good parallel performance for the generalized BTEX method. A larger speedup is possible by using a smaller value of h since more ODE systems can be generated such that each processor has more jobs to do to reduce the overhead of interprocessor communication.

A representative model simulation of steady-state oxygen profile is illustrated in Figure 6, which compares the oxygen partial pressures P_1 and P_3 in the capillary and the cell region calculated on a large grid ($n = 2048$) with that on a small grid ($n = 32$) as a function of the scaled distance along the capillary length (in x/L). The model simulations were performed using the parameter values representative of normal physiological conditions in the working cardiomycytes at moderate work rate (see [4] for details). Specifically, for these simulations, the total pool of exchangeable phosphate in the myocytes was set to TPP = 15 mM, blood flow was set to $F = 0.75$ ml min^{-1} (g tissue)$^{-1}$, and the rate of ATP consumption was set at $J_{AtC} = 0.45$ mmol sec^{-1} (l cell)$^{-1}$. The input arterial P_{O2} was assumed to be 100 mmHg. Other model parameters values representing cellular energy metabolism are set as in Beard [4]. The steady-state corresponds to $T = 20$ minutes.

These model solutions match well with the simulations of Beard [4] (see Figure 1B in [4]). These solutions suggest that oxygen partial pressures decrease from the arterial ($x = 0$) to the venous ($x = L$) end of the capillary, and decrease from the capillary region to the cellular region. The rate of cellular oxygen consumption and oxygen extraction at the venous end corresponding to these simulations were found to be 5.3 mol min^{-1} (g tissue)$^{-1}$ and 76%, respectively, values consistent with a moderate rate of cardiac work [4]. The model simulations in Figure 6 show that the solutions with $n = 2048$ do not differ significantly from the solutions with $n = 32$, suggesting that the solutions are accurate enough for mesh grids with $n = 32$. As shown in Figures 4 and 5, a typical steady-state simulation of Beard model of coupled blood-tissue transport and metabolism [4] using our generalized BTEX algorithm is significantly faster than that obtained using the traditional BTEX algorithm [2] with comparable grid sizes and accuracy criteria. Therefore, this new generalized BTEX algorithm will be critical for the analysis of experimental data on blood-tissue transport and reactions, which requires repeated computation of solutions of the governing PDEs and comparison of the experimental data to the corresponding model outputs.

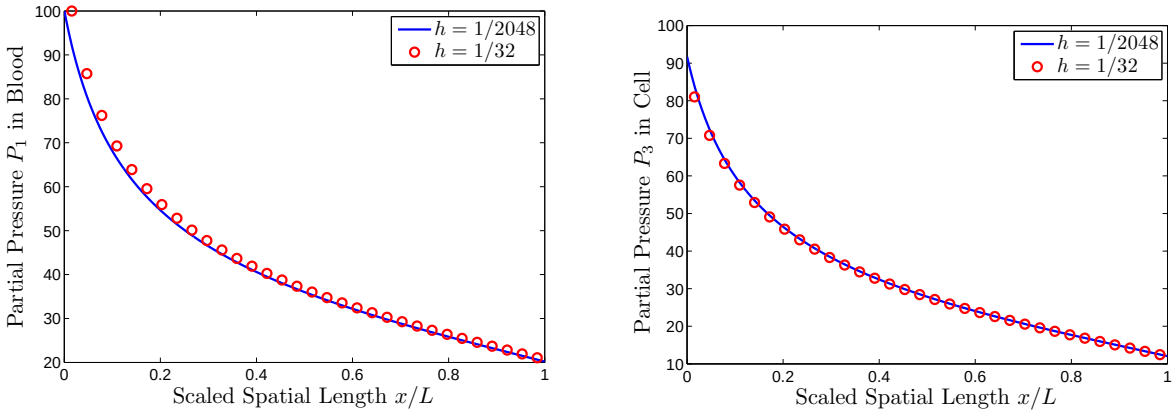

Figure 6: Comparison of the partial pressures P_1 and P_3 of oxygen in the capillary and the cell region as functions of the scaled distance along the capillary length. The generalized BTEX method is to solve the Beard model of coupled blood-tissue oxygen transport and oxidative energy metabolism [4]. The model parameters are as in [4].

7. Discussions and Conclusions

In this paper, we have presented an improved algorithm called the generalized BTEX (blood-tissue exchange) algorithm for solving a general blood-tissue transport and metabolism model governed by a system of one-dimensional semilinear first order partial differential equations (PDEs). We have also discussed its parallel implementation and parallel performance on both distributed-memory and shared-memory computers. Our method is a general procedure to convert the model problem approximately into many ODE initial value problems based on a special grid mesh with the time and spatial step sizes τ and h satisfying $\tau h = a$, where a is a speed of blood transport in the capillary. All these ODE problems have been shown to be independent of each other and to retain the size of the original model problem. Hence, they can be solved independently by an efficient ODE solver such as DLSODES. In this sense, the generalized BTEX algorithm is naturally a parallel algorithm, which can be easily implemented on a multiprocessor computer.

Note that most blood-tissue transport and metabolism models possess the stiffness spanning over multiple time scales since different species interact with each other between different blood-tissue regions. Hence, one favorable approach is to convert such a model to one or many ODE initial value problems so that a proper stiff ODE solver can be used to treat the stiff properties of the model problem. In fact, many stiff ODE algorithms and software packages have been developed in the numerical ODE research field over the years [9, 16, 14, 13, 19, 25]. Several such approaches have been available in the current literature (for solving the PDEs by converting them into ODEs), and the most popular one is the method of lines [15, 27, 28]. Most of these approaches approximate the model problem as only one large scale ODE initial value problem, which may be difficult to solve since most stiff ODE algorithms are

implicit, which involve numerical solutions of nonlinear algebraic systems at each time step, may produce a prohibited computing expenses, and may be very difficult to be implemented in parallel on a parallel computer. In contrast, all the ODE systems produced from our generalized BTEX method have the same size as that of the model problem, which is relatively very small (e.g., order of tens so far). Hence, our generalized BTEX method provides a better way to solve large-scale coupled blood-tissue transport and metabolism models than the method of lines [15, 27, 28] or even the traditional Lagrangian sliding fluid element algorithm [2]. The suggested generalized BTEX algorithm is accurate and robust which is demonstrated by comparing the analytical solutions of the Sangren and Sheppard model [26] with the numerical solutions obtained from the generalized BTEX algorithm as well as from the method of lines, showing the priority of the generalized BTEX algorithm over the method of lines.

The generalized BTEX algorithm is motivated by the Lagrangian sliding fluid element algorithm of Bassingthwaighte et al. [2] which was proposed in 1992. Compared to the Lagrangian sliding fluid element algorithm, our generalized BTEX algorithm is mathematically formalized and has been demonstrated to be suitable for fast numerical solutions of large-scale coupled blood-tissue transport and metabolism models, governed by one-dimensional semilinear PDEs, which is important for physiological applications in the analysis of experimental data (e.g., see [4, 31, 32, 33]). As a computational algorithm, the generalized BTEX algorithm needs to be further developed and analyzed on its error estimations and convergence properties. This will be taken up in our subsequent papers. Furthermore, the generalized BTEX algorithm will need to be extended to handle diffusive terms which will be suitable for numerical solutions of large-scale one-dimensional microcirculatory advection-diffusion-reaction models.

Acknowledgements

This work was supported by NIH grants R01-EB005825 and R01-HL072011. The authors thank the anonymous referees for their valuable comments.

References

[1] *MPI: A Message-Passing Interface Standard*. Knoxville, Tennessee, June 2008.

[2] J. B. Bassingthwaighte, I. S. J. Chan, and C. Y. Wang. Computationally efficient algorithms for convection-permeation-diffusion models for blood-tissue exchange. *Annals of Biomedical Engineering*, 20:687–725, 1992.

[3] J. B. Bassingthwaighte and C. A. Goresky. Modeling in the analysis of solute and water exchange in the microvasculature. In E. M. Renkin and C. C. Michel, editors, *Handbook of Physiology. Section 2, The Cardiovascular System*, volume IV, The Microcirculation, pages 549–626. American Physiological Society, Bethesda, MD, 1984.

[4] D. A. Beard. Modeling of oxygen transport and cellular energetics explains observations on in vivo cardiac energy metabolism. *PLoS Computational Biology*, 2:1093–1106, 2006.

[5] D. A. Beard and J. B. Bassingthwaighte. Modeling advection and diffusion of oxygen in complex vascular networks. *Annals of Biomedical Engineering*, 29:298–310, 2001.

[6] D. A. Beard, K. A. Schenkman, and E. O. Feigl. Myocardial oxygenation in isolated hearts predicted by an anatomically realistic microvascular transport model. *American Journal of Physiology Heart Circulation Physiology*, 285(5):H1826–1836, 2003.

[7] R. P. Beyer, J. B. Bassingthwaighte, and A. J. Deussen. A computational model of oxygen transport from red blood cells to mitochondria. *Computer Methods and Programs in Biomedicine*, 67:39–54, 2002.

[8] B. Chapman, G. Jost, and R. van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. MIT Press, Cambridge, MA, USA, 2007.

[9] C. F. Curtiss and J. O. Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Science*, 38:235–243, 1952.

[10] R. K. Dash and J. B. Bassingthwaighte. Simultaneous blood-tissue exchange of oxygen, carbon dioxide, bicarbonate and hydrogen ion. *Annals of Biomedical Engineering*, 34(7):1129–1148, 2006.

[11] A. Deussen and J. B. Bassingthwaighte. Modeling [15o]oxygen tracer data for estimating oxygen consumption. *American Journal of Physiology Heart Circulation Physiology*, 270:H1115–H1130, 1996.

[12] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* MIT Press, Cambridge, MA, USA, 2nd edition, 1999.

[13] E. Hairer and G. Wanner. Stiff differential equations solved by Radau methods. *Journal of Computational Applied Mathematics*, 111:93–111, 1999.

[14] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, Berlin, 2nd edition, 2002.

[15] S. Hamdi, W. E. Schiesser, and G. W. Griffiths. Method of lines. *Scholarpedia,* article 2859, 2007.

[16] A. Hindmarsh. Odepack: A systemized collection of ODEsolvers. In R. Stepleman, editor, *Scientific Computing*. American Physiological Society, North-Holland: Amsterdam, 1983.

[17] M. I. Kettunen, O. H. Grhn, M. J. Silvennoinen, M. Penttonen, and R. A. Kauppinen. Quantitative assessment of the balance between oxygen delivery and consumption in the rat brain after transient ischemia with t2 -bold magnetic resonance imaging. *Journal of Cerebral Blood Flow and Metabolism*, 22(3):262–270, 2002.

[18] A. Krogh. The number and distribution of capillaries in muscle with calculations of the oxygen pressure head necessary for supplying the tissue. *Journal of Physiology*, 52:409–415, 1919.

[19] S. Li and L. R. Petzold. Software and algorithms for sensitivity analysis of large-scale differential-algebraic systems. *Journal of Computational Applied Mathematics*, 125:131145, 2000.

[20] Z. Li, T. Yipintsoi, and J. B. Bassingthwaighte. Nonlinear model for capillary-tissue oxygen transport and metabolism. *Annals of Biomedical Engineering*, 25:604–619, 1997.

[21] Robert C. McOwen. *Partial Differential Equations: Methods and Applications*. Prentice Hall, Upper Saddle River, NJ, second edition, 2003.

[22] F. W. J. Olver. Bessel functions of integer order. In M. Abramowitz and I. A. Stegun, editors, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, chapter 9, pages 355–436. National Bureau of Standards, Washington DC, 1964.

[23] J. Pekar, T. Sinnwell, L. Ligeti, A. S. Chesnick, J. A. Frank, and A. C. McLaughlin. Simultaneous measurement of cerebral oxygen consumption and blood flow using ^{17}O and ^{19}F magnetic resonance imaging. *Journal of Cerebral Blood Flow and Metabolism*, 15(2):312–320, 1995.

[24] A. S. Popel. Theory of oxygen transport to tissue. *Critical Review in Biomedical Engineering*, 17:257–321, 1989.

[25] K. Radhakrishnan and A. C. Hindmarsh. Description and use of LSODE, the Livermore solver for ordinary differential equations. Technical Report 1327, NASA, NASA, Washington DC, 1993.

[26] W. C. Sangren and C. W. Sheppard. A mathematical derivation of the exchange of a labeled substance between a liquid flowing in a vessel and an external compartment. *Bulletin in Mathematical Biology*, 15(4):387–394, 1953.

[27] W. E. Schiesser. *The Numerical Method of Lines*. Academic Press, San Diego, 1991.

[28] W. E. Schiesser and C. A. Silebi. *Computational Transport Phenomena*. Cambridge University Press, New York, 1997.

[29] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. SIAM, Philadelphia, 2nd edition, 1994.

[30] E. F. van De Velde. *Concurrent Scientific Computing*. Springer-Verlag, New York, 1994.

[31] F. Wu, J. Zhang, and D. A. Beard. *Experimentally observed phenomena on cardiac energetics in heart failure emerge from simulations of cardiac metabolism*. *Proc Natl Acad Sci U S A* 106:7143-7148, 2009.

[32] F. Wu and D. A. Beard. *Roles of the creatine kinase system and myoglobin in maintaining energetic state in the working heart*. *BMC Syst Biol* 3:22, 2009.

[33] F. Wu, E. Y. Zhang, J. Zhang, R. J. Bache, and D. A. Beard. *Phosphate metabolite concentrations and ATP hydrolysis potential in normal and ischaemic hearts*. *J Physiol Lond* 586:4193-4208, 2008.