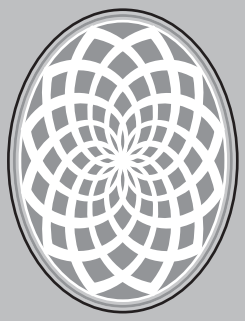

Industrial Mathematics Institute

2001:11

An Eulerian-Lagrangian substruturing domain decomposition method for unsteady-state advection-diffusion equations

H. Wang and M. Al-Lawatia

IMI

Preprint Series

Department of Mathematics
University of South Carolina

An Eulerian-Lagrangian Substructuring Domain Decomposition Method for Unsteady-State Advection-Diffusion Equations *

H. Wang † Mohamed Al-Lawatia ‡

April 22, 2001

Abstract

We develop an Eulerian-Lagrangian substructuring domain decomposition method for the solution of unsteady-state advection-diffusion transport equations. This method reduces to an Eulerian-Lagrangian scheme within each subdomain and to a type of Dirichlet-Neumann algorithm at subdomain interfaces. The method generates accurate and stable solutions that are free of artifacts even if large time steps are used in the simulation. Numerical experiments are presented to show the strong potential of the method.

1 Introduction

Advection-diffusion partial differential equations (PDEs) arise in petroleum reservoir simulation, subsurface contaminant transport and remediation, and many other applications. These problems typically possess solutions with moving steep fronts within some relatively small regions, where important chemistry and physics take place. Furthermore, an identifying feature of these applications is the presence of extremely large scale fluid flows coupled with transient transport of physical quantities such as pollutants, chemical species, radionuclides, and temperature. Consequently, an extremely refined global mesh is not feasible due to the excessive computational and storage cost. Domain decomposition techniques prove to be a feasible and powerful approach for the solution of these problems, because they allow a significant reduction of the size of the problems and the use of different physical or numerical models on different subdomains to model fluid flows more accurately. Furthermore, they are easily parallelizable. Therefore, a very important issue in the numerical simulation of subsurface fluid flow problems is the development of domain decomposition techniques for unsteady-state advection-diffusion transport PDEs.

Many domain decomposition methods that work well for elliptic and parabolic PDEs [20] could perform less promising for unsteady-state advection-diffusion transport PDEs. For example, the well-known Dirichlet-Neumann algorithm [5, 20], which was originally proposed by Bjørstad and Widlund in the development of domain decomposition methods for elliptic equations, assigns a Dirichlet condition to one subdomain and a Neumann boundary condition to its adjacent subdomain in the domain decomposition method and has been successfully applied to solve elliptic and parabolic problems. However, the Dirichlet-Neumann algorithm

*This research was supported in part by generous awards from Mobil Technology Company and ExxonMobil Upstream Research Company. In addition, the first author would like to acknowledge the support from the South Carolina State Commission of Higher Education: South Carolina Research Initiative Grant 13060-GA00 and NSF Grant DMS-0079549.

†Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208

‡Department of Mathematics and Statistics, Sultan Qaboos University, P.O. Box 36, Al-Khod Postal Code 123, Muscat, Sultanate of Oman

tends to generate nonphysical layers near subdomain interfaces for advection-diffusion transport PDEs, which are well known to arise for certain types of boundary conditions. The fundamental reason is that these methods do not necessarily respect the hyperbolic nature of advection-diffusion transport PDEs. In particular, even though the Dirichlet-Neumann matching conditions could be imposed for advection-diffusion PDEs with a nondegenerate diffusion, they might not be the hyperbolic limit of the advection-diffusion problems and so could become numerically unstable as the diffusion becomes smaller and smaller. The errors generated at subdomain interfaces are then propagated into the interior domain and could destroy the accuracy and the stability of the solutions on the entire domain.

Extensive research has been carried out on developing domain decomposition methods for advection-diffusion transport equations. One of the earliest developments to these problems was the work of Rannacher and Zhou [18]. Cai [6, 7] developed multilevel additive and multiplicative Schwartz preconditioners for parabolic and unsteady-state advection-diffusion PDEs. The Adaptive Dirichlet-Neumann (ADN) and Adaptive Robin-Neumann (ARN) nonoverlapping domain decomposition methods introduced in [10, 13, 22] choose the interface matching conditions to be adapted to the local flow direction. These methods prevent the occurance of artificial layers at subdomain interfaces as the advection becomes dominant. However, the underlying numerical methods used in these domain decomposition algorithms are standard finite difference, finite element, or finite volume methods possibly with some types of upwinding, and carry out temporal discretization in time. These methods tend to generate solutions with serious nonphysical oscillations or excessive numerical dispersion and grid orientation effect, or a combination of both, unless the spatial grids and time steps are extremely refined. This leads to significantly increased computational cost and computer storage [12, 25, 26]. Furthermore, these methods generate discrete algebraic systems with strongly nonsymmetric coefficient matrices at each time step, and thus require extra effort in solving the discrete systems.

Eulerian-Lagrangian methods [3, 11, 16, 17] carry out the temporal discretization of the advection-diffusion transport PDEs along the streamlines by combining the advective component with the accumulation term through a characteristic tracking algorithm and treat the diffusive term separately. These methods symmetrize the governing PDEs and generate accurate numerical solutions even if large time steps are used. Thus, they solve advection-diffusion transport PDEs effectively and efficiently. Unfortunately, many characteristic methods fail to conserve mass and have difficulty in treating boundary fluxes when characteristics intersect the boundary of the domain. This is one of the reasons why the few characteristic domain decomposition methods developed for advection-diffusion transport PDEs so far are overlapping domain decomposition methods [18, 21]. The Eulerian-Lagrangian localized adjoint method (ELLAM) provides a general characteristic solution procedure for the solution of advection-diffusion PDEs with general boundary conditions in a mass-conservative manner. Thus, ELLAM overcomes the principle shortcomings of previous characteristic methods while maintaining their numerical advantages. Our previous work [25, 26] also show that ELLAM schemes generate accurate solutions even if very large time steps and very coarse spatial grids are used, and often outperform many well received and widely used numerical methods.

In this paper we base on the ELLAM formulation to develop an Eulerian-Lagrangian nonoverlapping substructuring domain decomposition method for unsteady-state advection-diffusion transport equations. The developed method reduces to an ELLAM scheme within each subdomain and turns out to be a type of Dirichlet-Neumann domain decomposition al-

gorithm at subdomain boundaries. This method symmetrizes the governing transport PDEs and generates accurate and stable solutions that are free of artifacts even if large time steps and spatial grids are used in the simulation. This is in contrast to the facts that many Dirichlet-Neumann domain decomposition methods are known to generate poor solutions for advection-diffusion PDEs and that a Neumann boundary condition is not necessarily a natural interface matching condition for advection-diffusion PDEs.

The rest of the paper is organized as follows. In Section II we outline an ELLAM scheme for linear advection-diffusion transport equations. In Section III we develop an Eulerian-Lagrangian substructuring domain decomposition method. In Section IV we present numerical experiments to show the strong potential of the method. In Section V we draw conclusions and have some discussions.

2 An Underlying Numerical Method

In this paper we intend to carry out a principal study on the development of an effective and efficient domain decomposition method for unsteady-state advection-diffusion transport PDEs. For simplicity of exposition, we primarily focus on one-dimensional advection-diffusion problems to illustrate the ideas of the development. We then briefly outline the extension to multidimensional problems and discuss additional technical difficulties. We follow the development in [8, 27] to present an ELLAM scheme for the following initial-boundary value problem for the one-dimensional linear advection-diffusion transport PDE

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(V(x,t)u - D(x,t)\frac{\partial u}{\partial x}\right) &= f(x,t), \quad x \in (a,b), \ \ t \in [0,T], \\ u(x,0) &= u_o(x), \qquad x \in [a,b], \\ u(a,t) &= g(t), \qquad t \in (0,T], \\ -\left(D\frac{\partial u}{\partial x}\right)(b,t) &= h(t), \qquad t \in (0,T]. \end{aligned} \tag{2.1}$$

Here u_o is a prescribed initial condition. $V(x,t)$ and $D(x,t)$ represent the velocity field and the diffusion coefficient, respectively; both of which are assumed to be positive throughout the domain. The ELLAM method can accommodate any combination of Dirichlet, Neumann, or Robin conditions at the two boundary points $x = a$ and $x = b$ naturally in its formulation [8, 27]. In this section, we only present an ELLAM formulation for the advection-diffusion PDE with an inflow Dirichlet and an outflow Neumann boundary conditions, since this is what we need in the development of our domain decomposition algorithm in the next section.

2.1 Partition and Characteristic Curves

Let I and N be two positive integers, we define a partition of the domain $[a,b] \times [0,T]$ as follows:

$$\begin{aligned} x_i &= a + i\Delta x, \qquad i = 0,1,\ldots,I, \qquad \Delta x = \frac{b-a}{I}, \\ t^n &= n\Delta t, \qquad\quad n = 0,1,\ldots,N, \qquad \Delta t = \frac{T}{N}. \end{aligned} \tag{2.2}$$

A time-stepping procedure is used in the scheme, thus we only need to focus on the current time interval $(t^n, t^{n+1}]$. While this method can accommodate nonuniform meshes and also different spatial meshes on different time intervals, for simplicity of presentation we use a uniform spatial grid given in (2.2).

The space-time weak formulation for equation (2.1), obtained by multiplying a test function w that vanishes outside $\Omega_n = [a, b] \times (t^n, t^{n+1}]$ (described in more detail below) and integrating by parts, is given by

$$\begin{aligned} &\int_a^b u(x, t^{n+1}) w(x, t^{n+1})\, dx + \int_{t^n}^{t^{n+1}} \int_a^b D \frac{\partial u}{\partial x} \frac{\partial w}{\partial x}\, dx dt \\ &\quad + \int_{t^n}^{t^{n+1}} \left(Vu - D \frac{\partial u}{\partial x} \right) w \Big|_a^b dt - \int_{t^n}^{t^{n+1}} \int_a^b u \left(\frac{\partial w}{\partial t} + V \frac{\partial w}{\partial x} \right) dx dt \\ &= \int_a^b u(x, t^n) w(x, t_+^n)\, dx + \int_{t^n}^{t^{n+1}} \int_a^b f w\, dx dt, \end{aligned} \tag{2.3}$$

where, due to the discontinuity of $w(x, t)$ at time t^n, we use the notation $w(x, t_+^n) = \lim_{t \to t_+^n} w(x, t)$.

In accordance with the principle of the localized adjoint method [8], the test functions w should be selected from the solution space of the homogeneous adjoint equation

$$-\frac{\partial w}{\partial t} - V(x, t) \frac{\partial w}{\partial x} + \frac{\partial}{\partial x} \left(D(x, t) \frac{\partial w}{\partial x} \right) = 0. \tag{2.4}$$

Because the solution space of equation (2.4) is infinite-dimensional and only a finite number of test functions should be used in a numerical scheme, the test functions are chosen to be the (approximate) solutions of the following splitted system [8, 27]

$$\begin{aligned} \frac{\partial w}{\partial t} + V(x, t) \frac{\partial w}{\partial x} &= 0, \\ \frac{\partial}{\partial x} \left(D(x, t) \frac{\partial w}{\partial x} \right) &= 0. \end{aligned} \tag{2.5}$$

The first equation in (2.5) can be written as the ordinary differential equation

$$\begin{aligned} -\frac{d}{d\theta} w(X(\theta; \bar{x}, \bar{t}), \theta) &= 0, \\ w(X(\theta; \bar{x}, \bar{t}), \theta)|_{\theta = \bar{t}} &= w(\bar{x}, \bar{t}), \end{aligned} \tag{2.6}$$

along the characteristic $X(\theta; \bar{x}, \bar{t})$ which is defined for any given point $(\bar{x}, \bar{t})$ with $\bar{t} \in [t^n, t^{n+1}]$ as the solution of

$$\begin{aligned} \frac{dX}{d\theta} &= V(X, \theta) \\ X(\theta; \bar{x}, \bar{t})|_{\theta = \bar{t}} &= \bar{x}. \end{aligned} \tag{2.7}$$

From equation (2.6) we observe that inside the domain Ω_n, the test functions w are constant along the characteristics, i.e.

$$w(X(\theta; \bar{x}, \bar{t}), \theta) = w(\bar{x}, \bar{t}). \tag{2.8}$$

In addition, we introduce the following notations

$$\begin{aligned} \tilde{x} &= X(t^{n+1}; x, t^n), & b &= X(t; b^*(t), t^n), \\ x &= X(t^{n+1}; x^*, t^n), & b &= X(\tilde{t}(x); x, t^n), \\ x &= X(t^{n+1}; a, t^*(x)). \end{aligned} \tag{2.9}$$

We define $\tilde{x}$ to be the spatial point which the characteristic curve emanating from (x, t^n) forward tracks to at time t^{n+1}. Similarly we let x^* at time t^n denote the point which forward tracks to (x, t^{n+1}). For the characteristic that intersects the outflow boundary $\Gamma_n^{(O)} := \{(b, s) : s \in (t^n, t^{n+1})\}$, we define $b^*(t)$ as the spatial point at time t^n which forward tracks to (b, t) (with $b^* := b^*(t^{n+1})$). We also define $\tilde{t}(x)$ as the time instance that the characteristic curve emanating from (x, t^n) intersects the outflow boundary. Similarly for the characteristic originating on the inflow boundary $\Gamma_n^{(I)} := \{(a, s) : s \in (t^n, t^{n+1})\}$, we define $t^*(x)$ to be the time instance such that $(a, t^*(x))$ forward tracks to x at time t^{n+1}. We extend the definition of both $\tilde{t}(x)$ and $t^*(x)$ to be t^{n+1} and t^n, respectively, for characteristics not intersecting the corresponding boundary.

Furthermore, we observe that if the spatial nodes x_i near the outflow boundary $x = b$ are tracked forward from time t^n to time t^{n+1}, the number of spatial degrees of freedom crossing the outflow boundary $\Gamma_n^{(O)}$ is essentially the Courant number at the outflow boundary. To preserve the information, we should discretize $\Gamma_n^{(O)}$ in time with about the same number of degrees of freedom. More precisely, we partition the outflow boundary as follows

$$t_{n,i} = t^{n+1} - \frac{i\Delta t}{IC}, \qquad i = 0, 1, \ldots, IC, \tag{2.10}$$

where $IC = \lceil Cr \rceil$ is the ceiling of the Courant number Cr at the outflow boundary. This partition introduces some unknowns on the outflow boundary. However, it is used to insure that the scheme developed is suitable even for large values of the Courant number.

2.2 Numerical Scheme

In developing the ELLAM scheme, we notice that the second equation in the splitting (2.5) is a second-order variable-coefficient elliptic equation, which defines the spatial variation of the test functions. However, it is impossible to find the analytical solutions in closed form for this equation, in general. From equation (2.8), we only need to specify the test functions on $[a, b]$ at time t^{n+1} and at the space-time outflow boundary $\Gamma_n^{(O)}$. Furthermore, standard finite element shape functions provide accurate approximations to elliptic PDEs. Hence, we naturally choose the trial and test functions to be piecewise-linear functions at time t^{n+1} and at the outflow boundary $\Gamma_n^{(O)}$ as follows: At time t^{n+1}, we define the test function $w_i(x, t^{n+1})$ $(i = 1, \ldots, I-1)$ to be the hat functions

$$w_i(x, t^{n+1}) = \begin{cases} \dfrac{x - x_{i-1}}{\Delta x}, & x \in [x_{i-1}, x_i], \\ \dfrac{x_{i+1} - x}{\Delta x}, & x \in [x_i, x_{i+1}], \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

At the outflow boundary, the test functions are given by,

$$w_{I+i}(b, t) = \begin{cases} \dfrac{t_{n,i-1} - t}{\Delta t_f}, & t \in [t_{i-1}, t_i], \\ \dfrac{t - t_{n,i+1}}{\Delta t_f}, & t \in [t_i, t_{i+1}], \\ 0, & \text{otherwise,} \end{cases} \tag{2.12}$$

for $i = 1, \ldots, IC-1$, where $\Delta t_f = \frac{\Delta t}{IC}$. The test function $w_0(x, t^{n+1})$ is defined only on $[x_0, x_1]$ by (2.11) and similarly w_{I+IC} is defined only on $[t^n, t_{n,IC-1}]$, while w_I is so that $w_I(x, t^{n+1})$ is defined by (2.11) for the interval $[x_{I-1}, x_I]$, and $w_I(b, t)$ is defined by (2.12) on the interval $[t_{n,I+1}, t^{n+1}]$. For the interior of the domain Ω_n, we extend these test functions to be constant along the characteristics as described in equation (2.8).

The ELLAM scheme can be formulated by evaluating the space-time integrals in equation (2.3) along the characteristics. The second (source) term can be evaluated using a backward Euler quadrature as follows

$$
\begin{aligned}
&\int_{t^n}^{t^{n+1}} \int_a^b f\, w\, dx\, dt \\
&\quad = -\int_{t^n}^{t^{n+1}} \int_t^{t^{n+1}} f(X(s;a,t),s)\, w(X(s;a,t),s)\, \frac{\partial X}{\partial t}(s;a,t)\, ds dt \\
&\qquad + \int_a^{b^*} \int_{t^n}^{t^{n+1}} f(X(s;x,t^n),s)\, w(X(s;x,t^n),s)\, \frac{\partial X}{\partial x}(s;x,t^n)\, ds dx \\
&\qquad + \int_{b^*}^{b} \int_{t^n}^{\tilde{t}(x)} f(X(s;x,t^n),s)\, w(X(s;x,t^n),s)\, \frac{\partial X}{\partial x}(s;x,t^n)\, ds dx \\
&\quad = \int_a^{\tilde{a}} (t^{n+1} - t^*(x))\, f(x,t^{n+1})\, w(x,t^{n+1})\, dx \\
&\qquad + \int_{\tilde{a}}^{b} \Delta t\, f(x,t^{n+1})\, w(x,t^{n+1})\, dx \\
&\qquad - \int_{t^n}^{t^{n+1}} (t - t^n)\, \frac{\partial X}{\partial t}(t, b^*(t), t^n)\, f(b,t)\, w(b,t)\, dt + E_f(w),
\end{aligned}
\tag{2.13}
$$

where $E_f(w)$ is the truncation error due to the application of the backward Euler quadrature.

Next we evaluate the second (diffusion) integral on the left hand side of equation (2.3). This term can be approximated in a similar manner to the source term as follows

$$
\begin{aligned}
&\int_{t^n} t^{n+1} \int_a^b D\, \frac{\partial u}{\partial x}\, \frac{\partial w}{\partial x}\, dx\, dt \\
&\quad = -\int_{t^n}^{t^{n+1}} \int_t^{t^{n+1}} \left(D \frac{\partial u}{\partial X} \right) (X(s;a,t),s)\, \frac{\partial w}{\partial X}(X(s;a,t),s) \\
&\qquad \frac{\partial X}{\partial t}(s;a,t)\, ds dt \\
&\qquad + \int_a^{b^*} \int_{t^n}^{t^{n+1}} \left(D \frac{\partial u}{\partial X} \right) (X(s;x,t^n),s)\, \frac{\partial w}{\partial X}(X(s;x,t^n),s) \\
&\qquad\quad \frac{\partial X}{\partial x}(s;x,t^n)\, ds dx \\
&\qquad + \int_{b^*}^{b} \int_{t^n}^{\tilde{t}(x)} \left(D \frac{\partial u}{\partial X} \right) (X(s;x,t^n),s)\, \frac{\partial w}{\partial X}(X(s;x,t^n),s) \\
&\qquad\quad \frac{\partial X}{\partial x}(s;x,t^n)\, ds dx
\end{aligned}
$$

$$
\begin{aligned}
&= \int_a^{\tilde{a}} (t^{n+1} - t^*(x))\, D(x, t^{n+1})\, \frac{\partial u}{\partial x}(x, t^{n+1})\, w(x, t^{n+1})\, dx \\
&\quad + \int_{\tilde{a}}^{b} \Delta t\, D(x, t^{n+1})\, \frac{\partial u}{\partial x}(x, t^{n+1})\, \frac{\partial w}{\partial x}(x, t^{n+1})\, dx \\
&\quad - \int_{t^n}^{t^{n+1}} (t - t^n)\, D(x, t^{n+1})\, \frac{\partial u}{\partial x}(b, t)\, \frac{\partial w}{\partial t}(b, t)\, dt + E_D(w).
\end{aligned} \tag{2.14}
$$

where we have used a backward Euler quadrature in the second equality, with a truncation error denoted by $E_D(w)$. However, in order to avoid difficulties that would result from having the diffusive flux integral on the inflow boundary (in the third term of equation (2.3)), we make a slight adjustment to the variational formulation (2.3). Instead of applying our approximation procedure to the second term on the inflow boundary term, we approximate the term

$$
\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial x} \left(D \frac{\partial u}{\partial x} \right) (x, t) w(x, t)\, dx dt
$$

by a backward-Euler time integration along the characteristics and then integrate by parts. This results in the term

$$
- \int_a^{\tilde{a}} \frac{d\, t^*(x)}{dt} D \frac{\partial u}{\partial x}(x, t^{n+1}) w(x, t^{n+1})\, dx \tag{2.15}
$$

instead of the diffusive boundary flux integral in the third term of the left side of equation (2.3).

In deriving the ELLAM scheme, we use equations (2.13)–(2.15) for the appropriate terms in the weak form (2.3) and drop the error terms $E_f(w)$ and $E_D(w)$. We then incorporate the prescribed inflow and outflow boundary conditions in these terms. Secondly, we approximate the solution u of equation (2.1) by a piecewise-linear trial function U on $[a, b]$ at time t^{n+1} with respect to the spatial partition in (2.2), and at the outflow boundary $\Gamma_n^{(O)}$. Finally we drop the fourth (adjoint) term on the left hand side of equation (2.3). We obtain the ELLAM scheme

$$
\begin{aligned}
&\int_a^b u(x, t^{n+1})\, w(x, t^{n+1})\, dx + \int_{t^n}^{t^{n+1}} V(b, t)\, u(b, t)\, w(b, t)\, dt \\
&\quad + \int_a^b (t^{n+1} - t^*(x))\, D(x, t^{n+1})\, \frac{\partial U}{\partial x}(x, t^{n+1})\, \frac{\partial w}{\partial x}(x, t^{n+1})\, dx \\
&\quad - \int_a^{\tilde{a}} \frac{d\, t^*(x)}{dt}\, D(x, t^{n+1}) \frac{\partial U}{\partial x}(x, t^{n+1})\, w(x, t^{n+1})\, dx \\
&= \int_a^b u(x, t^n)\, w(x, t_+^n)\, dx \;+\; \int_{t^n}^{t^{n+1}} V(a, t)\, g(t)\, w(a, t)\, dt \\
&\quad - \int_{t^n}^{t^{n+1}} h(t)\, w(b, t)\, dt \;-\; \int_{t^n}^{t^{n+1}} t^{n+1}(t - t^n) h(t)\, \frac{\partial w}{\partial t}(b, t)\, dt \\
&\quad - \int_{t^n}^{t^{n+1}} t^{n+1}(t - t^n)\, \frac{\partial Y}{\partial t}(t; b^*(t), t^n)\, f(b, t)\, w(b, t)\, dt \\
&\quad + \int_a^b (t^{n+1} - t^*(x))\, f(x, t^{n+1})\, w(x, t^{n+1})\, dx
\end{aligned} \tag{2.16}
$$

Remark 1 *With the prescribed inflow and outflow boundary conditions as well as the solution $u(x,t^n)$ which is known from the computations at the previous time level (or $u_o(x)$ for $n=0$), the derived ELLAM scheme solves for $u(x,t^{n+1})$ on $[a,b]$ and $u(b,t)$ on $\Gamma_n^{(O)}$. This scheme symmetrizes the governing equation (2.1), generates accurate numerical solutions even if large time steps are used, naturally incorporates the boundary conditions, and conserves mass. The numerical performance of scheme (2.16) and its comparison with many other schemes can be found in [1, 23, 27]. The theoretically proven optimal-order error estimates for ELLAM schemes (2.16) were derived in [23, 27].*

Remark 2 *For simplicity of illustrating the ideas of the ELLAM schemes and the proposed domain decomposition methods, we outlined an ELLAM scheme for one-dimensional advection-diffusion PDE in this section. For multidimensional advection-diffusion PDEs, in principle the ELLAM schemes can be derived in parallel although additional technical difficulties could arise in the development. The reason is that the ELLAM method, like many other characteristic methods, combines the time derivative term and the advection term in multidimensional advection-diffusion PDEs into a one-dimensional ordinary differential equation along the characteristics that are spatial curves in the multidimensional spaces. The approximation of the characteristics curves in multiple spatial dimensions has virtually the same accuracy and complexities as that for one-dimensional problems. The fact that the ELLAM approaches give a good approximation for the critical initial-boundary value problems of he domains in addition to giving a more symmetric problem can be well addressed in the context of one-dimensional problems. In fact, the principal issues could be drown in the technically more difficult multidimensional problems. The strength of the ELLAM method is its consistent handling of boundary conditions, which in a domain decomposition setting means that the global transport of information is accurately taken care of.*

Remark 3 *We refer readers to [14, 19, 25] for the detailed algorithm development for multidimensional ELLAM schemes and the discussions of the related technical difficulties. The performance of multidimensional ELLAM schemes and their comparison with other methods were investigated in [14, 25, 26]. The associated optimal-order error estimate was obtained in [24]. The application of ELLAM schemes for multidimensional coupled systems can be found in [4, 28].*

3 An Eulerian-Lagrangian Nonoverlapping Substructuring Domain Decomposition Method

In this section we develop an Eulerian-Lagrangian nonoverlapping substructuring domain decomposition method for unsteady-state advection-diffusion transport PDEs. For simplicity of exposition, we present the development of the method for the one-dimensional problem (2.1) based on the ELLAM scheme (2.16) presented in the previous section. Hence, the presentation in this section should be looked upon as a principal investigation of the method. The extension of the method to multidimensional unsteady-state advection-diffusion transport PDEs could pose additional technical difficulties and will be discussed briefly at the end of this section.

As we mentioned in Section I, many domain decomposition methods work very well for elliptic and parabolic equations, but fail to work for advection-diffusion PDEs due to the advection-dominance. In order for the underlying finite difference, finite element, or finite

volume methods used in these domain decomposition algorithms to generate physically reasonable solutions without nonphysical oscillations, the size of the mesh Peclet number Pe must satisfy

$$Pe = \frac{V\Delta x}{D} = \mathcal{O}(1). \tag{3.1}$$

Namely, the spatial grids have to be refined excessively for advection-dominated PDEs. Consequently, the size of time steps has to be chosen very small too. Thus, the underlying numerical methods generate extremely large size discrete algebraic systems to solve, leading to extremely expensive computational cost and excessive computer storage. One approach to stabilize these numerical schemes is to use some types of upstream weighting techniques [2, 9]. This corresponds to increasing the size of the diffusion D in (3.1) by means of numerical diffusion. This explains why upwind methods stabilize numerical schemes for advection-diffusion equations. While the domain decomposition methods using the upwind techniques could stabilize the numerical procedures, the excessive numerical diffusion introduced in upwind schemes tend to significantly smear the steep fronts of the solutions. To alleviate the smearing, one again has to use very fine spatial grids and time steps.

Another approach to stabilize the numerical schemes for advection-diffusion equations is to use characteristic methods [3, 11, 16, 17], which directly apply to these equations in a nonconservative form

$$\frac{\partial u}{\partial t} + V(x,t)\frac{\partial u}{\partial x} - \frac{\partial}{\partial x}\left(D(x,t)\frac{\partial u}{\partial x}\right) + \frac{\partial V}{\partial x}(x,t)u = f(x,t). \tag{3.2}$$

By combining the advective component with the accumulation term in equation (3.2) to formulate a directional derivative along the characteristic $X(\theta;\bar{x},\bar{t})$ given by (2.7), one can rewrite equation (3.2) as the following parabolic equation

$$\frac{du}{d\theta} - \frac{\partial}{\partial x}\left(D(x,t)\frac{\partial u}{\partial x}\right) + \frac{\partial V}{\partial x}(x,t)u = f(x,t) \tag{3.3}$$

without an advection term. This is why characteristic methods symmetrize the governing advection-diffusion equations and stabilize the corresponding numerical schemes. For example, if one discretizes the first term on the left hand side of equation (3.3) along the characteristics emanating backward from (x,t^{n+1}) by a first-order difference quotient and apply the finite element method to the resulting equation at time t^{n+1}, one obtains the modified method of characteristics (MMOC) that was originally developed in [11]

$$\begin{aligned}\int_\Omega \frac{u(x,t^{n+1}) - u(x^*,t^n)}{\Delta t} w(x,t^{n+1})dx \\ + \int_\Omega \frac{\partial V}{\partial x}(x,t^{n+1})u(x,t^{n+1})w(x,t^{n+1})\,dx \\ = \int_\Omega f(x,t^{n+1})w(x,t^{n+1})dx.\end{aligned} \tag{3.4}$$

As one can see, the MMOC scheme carries out the temporal discretization along the characteristics through a characteristic tracking algorithm and treats the diffusive term separately on the fixed spatial grids (2.2) on $[a,b]$ at time t^{n+1}. Since there is no advection term in equation (3.3) and the MMOC scheme (3.4), the restriction (3.1) is satisfied automatically.

However, equation (3.3) is defined along the characteristics, which could intersect the boundary of the domain. It is not clear how to define a characteristic scheme (e.g. the MMOC scheme) near the boundary; many characteristic methods have treated general flux boundary conditions in an *ad hoc* manner. This is why in the development of many characteristic methods in the literature, only the initial-value problem for advection-diffusion PDEs is considered or a periodic assumption has been imposed on the advection-diffusion PDEs. Consequently, despite the advantages of characteristic methods, the few characteristic domain decomposition methods developed for advection-diffusion transport PDEs so far are overlapping domain decomposition methods [18, 21] that avoid the difficulties in treating boundary conditions.

The ELLAM formalism outlined in Section II starts from a space-time weak formulation on the current space-time strip $[a,b] \times [t^n, t^{n+1}]$. In addition to possessing all the numerical advantages of many previous characteristic methods, the resulting ELLAM scheme (2.16) is defined on $[a,b]$ at time t^{n+1} and on the space-time inflow and outflow boundaries $\Gamma_n^{(I)}$ and $\Gamma_n^{(O)}$. It accommodates all the boundary conditions into its formulation. This enables us to develop an Eulerian-Lagrangian nonoverlapping substructuring domain decomposition method for unsteady-state advection-diffusion equations. However, we need to choose the subdomain interface matching conditions very carefully for the method. At the first glance, equation (3.3) is formally a parabolic equation, so the standard Dirichlet-Neumann algorithm should work well. However, equation (3.3) is oriented along the characteristics. Hence, if we want to use the standard Dirichlet-Neumann algorithm, we would have to specify the interface matching conditions on the space-time subdomains oriented along the characteristics. This is equivalent to specifying the interface matching conditions on the moving boundary in the space domain and is apparently not computationally feasible! In practice, we want to carry out the domain decomposition algorithm on subdomains with their boundaries being oriented in time. Hence, even though we utilize a characteristic method to solve the governing transport equation, we still have to respect their advection-dominance in the space-time domain. Notice that in the case of elliptic or parabolic equations that are virtually homogeneous in all the directions, the choice of the Dirichlet or Neumann interface matching condition is symmetric with respect to an interface boundary. Namely, if a Dirichlet condition is assigned to one subdomain, then a Neumann condition should be assigned to its adjacent subdomain or vice versa. However, advection-diffusion transport equations are strongly direction-dependent. Their solutions are much smoother along the characteristics than they are in any other directions. It is well known that in the context of advection-diffusion transport equations a Dirichlet boundary condition is well posed on the inflow boundary while a Neumann boundary condition is well posed on the outflow boundary, but could be completely numerically unstable vice versa [27]. Therefore, we have to very carefully choose the proper type of interface matching conditions at the proper side of the subdomain. This leads to the following Eulerian-Lagrangian nonoverlapping substructuring domain decomposition method:

Partition of the Domain

Decompose the spatial domain $\Omega = [a,b]$ into a union of $2M$ subdomains

$$\Omega = \bigcup_{i=1}^{2M} \Omega^{(i)}, \qquad \Omega^{(i)} = [d_{i-1}, d_i], \qquad i = 1, 2, \ldots, 2M \tag{3.5}$$

with

$$a = d_0 < d_1 < d_2 < \ldots < d_{2M} = b. \tag{3.6}$$

for $n = 0, 1, \ldots, N-1$ **do**

Initialize the iteration parameter $l = 0$.

if ERROR $>$ TOLERANCE **then**

$l = l + 1$.

L1. **for** $i = 1, 3, \ldots, 2M-1$ **do**

A. Use the prescribed inflow boundary condition at $x = a$ for $\Omega^{(1)}$ and the following relation

$$u^{(l)}(d_{i-1}^+, t) = u^{(l-1)}(d_{i-1}^-, t), \qquad i = 3, 5, \ldots, 2M-1 \tag{3.7}$$

to define an artificial inflow Dirichlet boundary condition for the subdomain $\Omega^{(i)}$ $(i = 3, 5, \ldots, 2M-1)$. Here $u(d_{i-1}^-, t)$ and $u(d_{i-1}^+, t)$ represent the left- and right-limits of u at the point d_{i-1}, respectively, and $u^{(0)}(d_{i-1}^-, t)$ is defined by

$$u^{(0)}(d_{i-1}^-, t) = u(d_{i-1}^*(t), t^n), \qquad i = 3, 5, \ldots, 2M-1, \tag{3.8}$$

where $d_{i-1}^*(t) \in \Omega^{(i-1)}$ is the point at time level t^n which the point (d_{i-1}, t) backtracks to.

B. Use the relation

$$\frac{\partial u}{\partial x}^{(l)}(d_i^-, t) = \frac{\partial u}{\partial x}^{(l-1)}(d_i^+, t), \qquad i = 1, 3, \ldots, 2M-1 \tag{3.9}$$

to define an artificial outflow Neumann boundary condition for the subdomain $\Omega^{(i)}$ $(i = 1, 3, \ldots, 2M-1)$. Here $\frac{\partial u}{\partial x}^{(l-1)}(d_i^+, t)$ is defined by

$$\frac{\partial u}{\partial x}^{(l-1)}(d_i^+, t) = \begin{cases} \dfrac{\partial u}{\partial x}(d_i^*(t), t^n), & \text{if } l = 1, \\ \dfrac{\partial u}{\partial x}(d_i^*(t), t^n)\dfrac{t^{n+1} - t}{\Delta t} & \\ \quad + \dfrac{\partial u}{\partial x}^{(l-1)}(\tilde{d}_i(t), t^{n+1})\dfrac{t - t^n}{\Delta t}, & \text{if } l \geq 2, \end{cases} \tag{3.10}$$
$$i = 1, 3, \ldots, 2M-1,$$

where $d_i^*(t) \in \Omega^{(i)}$ is the point at time level t^n which the point (d_i, t) backtracks to and $\tilde{d}_i(t) \in \Omega^{(i+1)}$ is the point at time level t^{n+1} which the point (d_i, t) tracks forward to.

C. With the inflow and outflow boundary conditions introduced in Steps L1.A and L1.B, use the ELLAM scheme presented in Section II to solve equation (2.1) on the *odd numbered* subdomains $\Omega^{(i)}$ $(i = 1, 3, \ldots, 2M-1)$ in parallel, yielding the l-th iterative solution $u^{(l)}(x, t^{n+1})$ on $\Omega^{(i)}$ and $u^{(l)}(d_i^-, t)$ on the outflow boundary of $\Omega^{(i)}$ for all the *odd numbered* subdomains $\Omega^{(i)}$ $(i = 1, 3, \ldots, 2M-1)$.

end

L2. **for** $i = 2, 4, \ldots, 2M$ **do**

A. Use the following relation

$$u^{(l)}(d_{i-1}^+, t) = u^{(l)}(d_{i-1}^-, t), \qquad i = 2, 4, \ldots, 2M \tag{3.11}$$

to define an artificial inflow Dirichlet boundary condition for the subdomain $\Omega^{(i)}$ $(i = 2, 4, \ldots, 2M)$.

B. Use the prescribed outflow boundary condition at $x = b$ for the subdomain $\Omega^{(2M)}$ and the relation

$$\frac{\partial u}{\partial x}^{(l)}(d_i^-, t) = \frac{\partial u}{\partial x}^{(l)}(d_i^+, t), \qquad i = 2, 4, \ldots, 2M-2 \tag{3.12}$$

to define an artificial outflow Neumann boundary condition for the subdomain $\Omega^{(i)}$ $(i = 2, 4, \ldots, 2M-2)$. Here

$$\frac{\partial u}{\partial x}^{(l)}(d_i^+, t) = \frac{\partial u}{\partial x}(d_i^*(t), t^n)\frac{t^{n+1}-t}{\Delta t} + \frac{\partial u}{\partial x}^{(l)}(\tilde{d}_i(t), t^{n+1})\frac{t-t^n}{\Delta t}. \tag{3.13}$$

C. With the inflow and outflow boundary conditions introduced in Steps L2.A and L2.B, use the ELLAM scheme presented in Section II to solve equation (2.1) on the *even numbered* subdomains $\Omega^{(i)}$ $(i = 2, 4, \ldots, 2M)$ in parallel, yielding the l-th iterative solution $u^{(l)}(x, t^{n+1})$ on $\Omega^{(i)}$ and $u^{(l)}(d_i^-, t)$ on the outflow boundary of $\Omega^{(i)}$ for all the *even numbered* subdomains $\Omega^{(i)}$ $(i = 2, 4, \ldots, 2M)$.

end

else

Define $u(x, t^{n+1}) = u^{(l)}(x, t^{n+1})$ on $\Omega^{(i)}$ for $i = 1, 2, \ldots, 2M$.

endif

end

Remark 4 *The domain decomposition method reduces to an Eulerian-Lagrangian method on each subdomain and to a type of Dirichlet-Neumann domain decomposition algorithm between the subdomain interfaces. However, the interface Dirichlet-Neumann algorithm is different from the standard Dirichlet-Neumann algorithm [5, 20] which is symmetric in the choice of Dirichlet and Neumann interface conditions. In fact, the current domain decomposition method is related to the Adaptive Dirichlet-Neumann (ADN) and the Adaptive Robin-Neumann (ARN) nonoverlapping domain decomposition methods introduced in [10, 13, 22] which choose the interface matching conditions to be adapted to the local flow direction. But in the ADN and ARN methods, the information on the flow direction is only used to select the correct type of interface conditions to be imposed. A standard numerical method is still used to solve the governing transport PDE on each subdomain. In the current method, we use the flow information both on selecting the correct interface matching condition and on setting up an ELLAM scheme to solve the advection-diffusion PDE.*

Remark 5 *For simplicity of exposition, we have presented the domain decomposition method for the one-dimensional unsteady-state advection-diffusion transport PDE (2.1) in this section. As pointed out in Remarks 2.2 and 2.3 in Section II, ELLAM schemes have been developed and analyzed for multidimensional advection-diffusion transport PDEs. In principle, the domain*

decomposition method developed in this paper can be extended to multidimensional analogues of advection-diffusion PDE (2.1). For example, instead of partitioning the spatial domain as a union of a sequence of intervals as in (3.5) for the one-dimensional problem (2.1), we should order the subdomains in a red-black order in the domain decomposition method for a two-dimensional advection-diffusion transport PDEs. Of course, varieties of new technical difficulties could arise and need to be taken care of. These issues will be addressed in subsequent papers.

Remark 6 *As pointed out in Remark 2.3 in Section II, optimal-order asymptotic error estimates have been proven theoretically for ELLAM schemes for multidimensional advection-diffusion transport PDEs. Although the theoretical estimates for Dirichlet-Neumann algorithms for elliptic PDEs were already proved, there is no corresponding estimate for the domain decomposition method presented in this paper. These estimates will be studied in subsequent papers.*

4 Numerical Experiments

In this section we describe numerical experiments which show the performance of the domain decomposition method developed.

Example 1. This example considers the transport of a Gaussian pulse over the spatial domain $[a, b] = [0, 3.2]$. The initial configuration of the hill is given by

$$u_o(x) = \exp\left(-\frac{(x - x_c)^2}{2\sigma^2}\right) \tag{4.1}$$

where the center $x_c = 0.0$ and the standard deviation $\sigma = 0.0316$ (which gives $2\sigma^2 = 0.002$). We use the variable velocity field given by $V(x, t) = 1 + 0.01x$, and a diffusion coefficient of $D(x, t) = 10^{-4}$. The corresponding analytical solution to equation (2.1) with the initial condition (4.1) is given by,

$$u(x, t) = \frac{\sigma}{\sqrt{\sigma^2 + 2Dt}} \exp\left(-\frac{(x - x_c - Vt)^2}{2\sigma^2 + 4Dt}\right). \tag{4.2}$$

The right hand side $f(x, t)$ is computed accordingly.

In this example, we use a mesh of size $\Delta x = \frac{1}{100}$, $\Delta t = \frac{1}{30}$, and four subdomains (i.e. $M = 2$) as follows

$$[0, 3.2] = [0, 0.8] \cup [0.8, 1.6] \cup [1.6, 2.4] \cup [2.4, 3.2]. \tag{4.3}$$

We carry several runs with different number of iterations in the domain decomposition algorithm. Table 1 contains the results of several of these runs including the L_2 and the L_1 norms of the error at time $T = 2.5$ and the consumed CPU time (in seconds) measured on a Pentium III-500MHZ Personal computer. Moreover, we present in Figure 1 the ELLAM solution (plotted using a solid line) verses the analytical solution (plotted using a dotted line) at various times when two iterations are used in the algorithm.

From Figure 1 we observe that the numerical solution generated is very accurate and coincides with the analytical solution. Moreover, it moves across the interfaces in a fairly

Table 1: Results of Experiment I at time $T = 2.5$ (with $M = 2$).

Iterations	L_2 Error	L_1 Error	CPU in seconds	Figure#
0	3.729710×10^{-3}	1.484552×10^{-3}	5.3	-
1	3.653351×10^{-3}	1.462222×10^{-3}	8.4	-
2	3.648984×10^{-3}	1.459424×10^{-3}	11.3	1
3	3.649800×10^{-3}	1.459893×10^{-3}	15.0	-
4	3.649687×10^{-3}	1.459830×10^{-3}	18.2	-
5	3.649702×10^{-3}	1.459838×10^{-3}	21.5	-
10	3.649700×10^{-3}	1.459837×10^{-3}	37.6	-

smooth manner without introducing any artifacts. In terms of the number of iterations used, we observe that two iterations were enough in the algorithm, with a minimal cost of 11.3 CPU seconds.

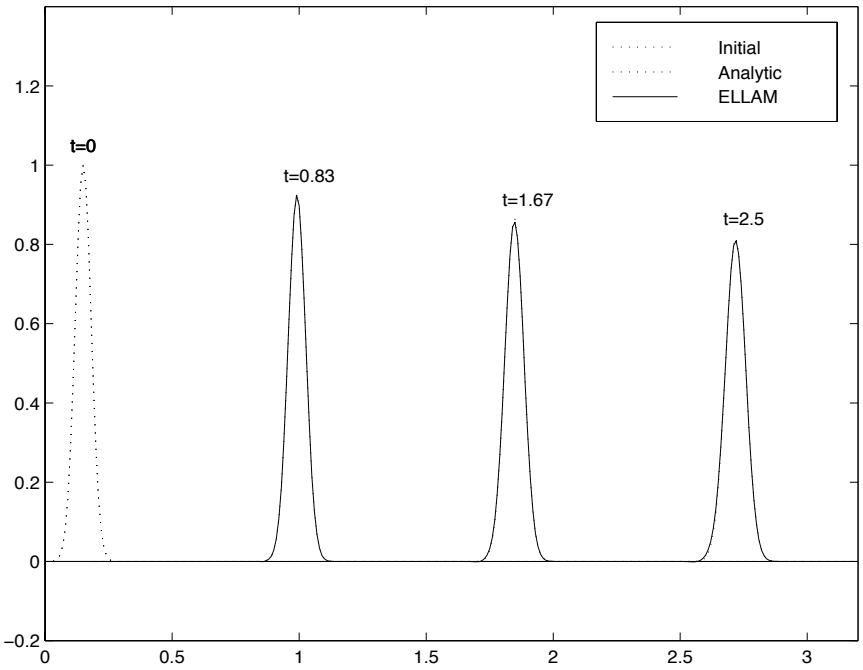

Figure 1: Solutions of Experiment I at various times

Example 2. To observe the performance of the method for problems with discontinuous initial conditions and steep analytical solutions, we consider in this example the transport of a diffused step function over a spatial domain of $[0, 6]$. The initial condition is a box function of a unit height supported on the interval $[0.35, 0.65]$

$$u_o(x) = \begin{cases} 1, & \text{if } x \in [0.35, 0.65] \subset (0, 6) \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

We simulate using a constant velocity field of $V(x, t) = 1$ and the relatively small diffusion of $D(x, t) = 10^{-4}$ over a time interval of $[0, 5]$. The analytical solution for equation (2.1) with initial condition (4.4) is given by

$$u(x, t) = \frac{1}{2}\left[\text{erf}\left(\frac{x - Vt - 0.35}{\sqrt{4Dt}}\right) - \text{erf}\left(\frac{x - Vt - 0.65}{\sqrt{4Dt}}\right)\right] \tag{4.5}$$

where

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-s^2) ds$$

denotes that standard error function. The source and sink term is zero.

Table 2: Results of Experiment II at time $T = 5.0$ (with $M = 3$).

Iterations	L_2 Error	L_1 Error	CPU in seconds	Figure#
0	2.453164×10^{-3}	2.983214×10^{-3}	8.1	-
1	2.261420×10^{-3}	2.503061×10^{-3}	12.0	-
2	2.248619×10^{-3}	2.472792×10^{-3}	15.9	2
3	2.250118×10^{-3}	2.475962×10^{-3}	20.1	-
4	2.249953×10^{-3}	2.475616×10^{-3}	24.0	-
5	2.249973×10^{-3}	2.475655×10^{-3}	26.8	-
10	2.249968×10^{-3}	2.475649×10^{-3}	45.3	-

In the test runs we decompose the domain into six equally spaced subdomains (i.e. $M = 3$) each of unit length. Then using a mesh of sizes $\Delta x = \frac{1}{100}$ and $\Delta t = \frac{1}{30}$ we carry several runs with different number of iterations in the method. The results of several of these runs including the L_2 and the L_1 norms of the error and the CPU time used are presented in Table II. Furthermore, we present in Figure 2 the ELLAM solution (solid line) and the analytical solution (dotted line) at several time instances of the simulation. In this figure we observe that the ELLAM solution accurately captures the solution and its steep fronts without introducing any numerical artifacts. From Table II we observe that the solution stabilizes as we increase the number of iterations, while only two iterations were enough to provide the solution with the least error (measured by the L_2 and L_1 norms). The results in this example conform to the results of the previous example.

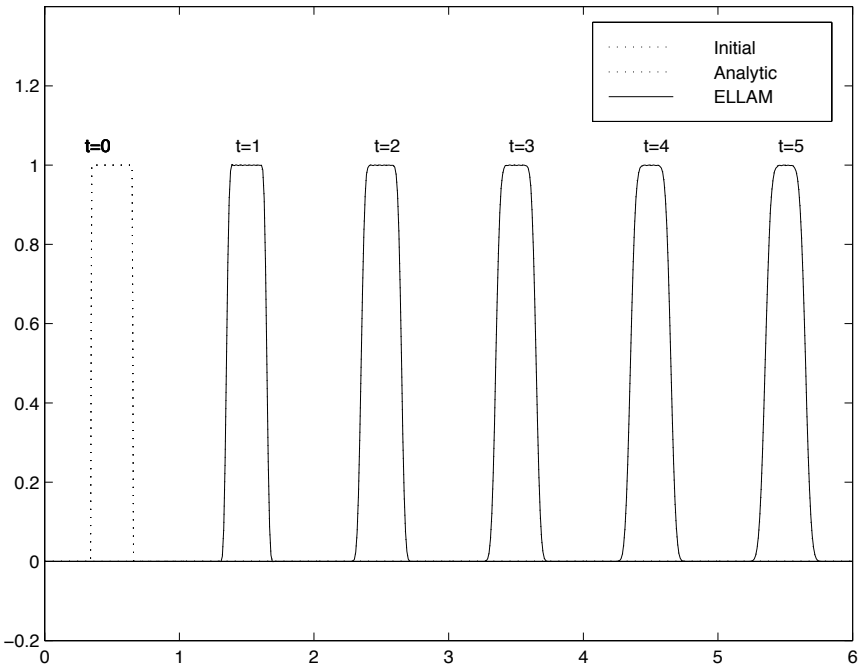

Figure 2: Solutions of experiment II at various times using 6 subdomains

Example 3. In an attempt to cover the variety of standard test problems used widely to test advection-diffusion codes, we consider in this example the transport of diffused step and triangle functions over a spatial domain of $[0, 8]$. The initial condition we use is

$$u_o(x) = \begin{cases} 1, & \text{if } 0.1 \le x \le 0.4, \\ \left| \dfrac{x - 0.55}{0.05} \right| - 1, & \text{if } 0.5 \le x \le 0.6, \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

We use a velocity field of $V(x,t) = 1$ and a diffusion coefficient of $D(x,t) = 10^{-4}$. The analytical solution to equation (2.1) using these parameter and the initial condition (4.6) is found assuming a homogeneous source and sink term.

Table 3: Results of Experiment III at time T =6.8 (with $M = 2$).

Iterations	L_2 Error	L_1 Error	CPU in seconds	Figure #
0	1.787892×10^{-3}	2.738425×10^{-3}	10.1	-
1	1.724342×10^{-3}	2.646041×10^{-3}	14.6	-
2	1.719773×10^{-3}	2.636059×10^{-3}	19.1	3
3	1.720241×10^{-3}	2.637064×10^{-3}	23.9	-
4	1.720202×10^{-3}	2.636994×10^{-3}	27.8	-
5	1.720206×10^{-3}	2.637002×10^{-3}	33.0	-
10	1.720205×10^{-3}	2.636999×10^{-3}	55.7	-

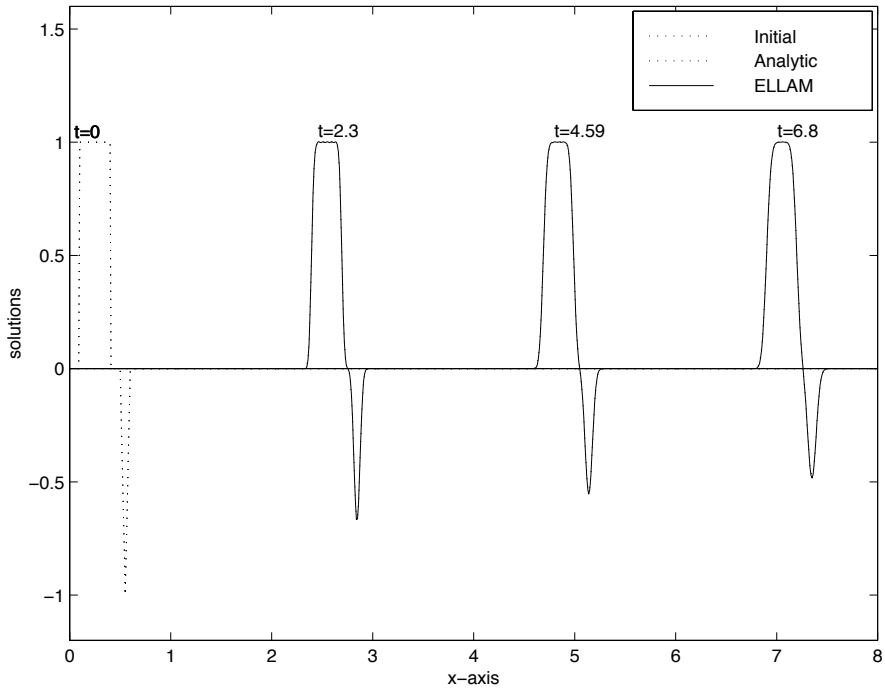

Figure 3: Solutions of experiment III at various times using 4 subdomains

In a similar manner to the earlier examples, we decompose the spatial domain $[0, 8]$ into four equally spaced subdomains (i.e. each of two units length). Then using a mesh of sizes $\Delta x = 0.01$ and $\Delta t = .0425$ we carry several runs varying the number of iterations. We present in Figure 3 the ELLAM and the analytical solutions generated at several time instances while in Table 4 we present the results of some of these runs at the final time of the simulation $T = 6.8$. We observe that the ELLAM solutions generated using two iterations in the algorithm are very accurate and capture all the details of the solution in an accurate manner without any artifacts.

5 Summary and Discussion

In this paper we base on an Eulerian-Lagrangian localized adjoint method (ELLAM) to develop an Eulerian-Lagrangian nonoverlapping substructuring domain decomposition method for the solution of unsteady-state advection-diffusion transport equations. The method turns out to be a type of Dirichlet-Neumann domain decomposition algorithms, but is different from the standard Dirichlet-Neumann algorithm which was originally proposed by Bjørstad and Widlund [5, 20] in the development of nonoverlapping domain decomposition methods for elliptic equations. Rather, the method is related to the Adaptive Dirichlet-Neumann (ADN) and the Adaptive Robin-Neumann (ARN) nonoverlapping domain decomposition methods in [10, 13, 15, 22] in that they all choose the interface matching conditions based on the the local flow direction. But in the ADN and ARN methods a standard finite difference, finite element, or finite volume method is still used to solve the governing transport equation on each subdomain. In the current method, we use the flow information both on selecting the correct interface matching condition and on setting up an ELLAM scheme to solve the advection-diffusion equation. Numerical results show that the developed domain decomposition method generates accurate numerical solutions without noticeable artifacts.

References

[1] M. Al-Lawatia, R.C. Sharpley, and H. Wang, (1999). Second-order characteristic methods for advection-diffusion equations and comparison to other schemes. *Advances in Water Resources*, 22: 741–768.

[2] J.W. Barrett and K.W. Morton, (1984). Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems. *Comp. Meth. Appl. Mech. Engrg.*, 45:97–122.

[3] J.P. Benque and J. Ronat, (1982). Quelques difficulties des modeles numeriques en hydraulique. In Glowinski and Lions, editors, *Computing Methods in Applied Sciences and Engineering*, pages 471–494. North-Holland.

[4] P.J. Binning and M.A. Celia, (1996). A finite volume Eulerian-Lagrangian localized adjoint method for solution of the contaminant transport equations in two-dimensional multi-phase flow systems. *Water Resour. Res.*, 32: 103–114.

[5] Bjørstad P. and Widlund O., (1986). Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.* 23: 1093–1120.

[6] Cai X.-C., (1991). Additive Schwarz algorithms for parabolic convection-diffusion equations. *Numer. Math.* 60: 41–61.

[7] Cai, X.-C., (1994). Multiplicative Schwarz methods for parabolic problems. *SIAM J. Sci Comput.* 15: 587–603, 1994.

[8] M.A. Celia, T.F. Russell, I. Herrera, and R.E. Ewing, (1990). An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Advances in Water Resources*, 13:187–206.

[9] I. Christie, D.F. Griffiths, A.R. Mitchell, and O.C. Zienkiewicz, (1976). Finite element methods for second order differential equations with significant first derivatives. *Int. J. Num. Engrg.*, 10:1389–1396.

[10] M. Ciccoli, (1996). Adaptive domain decomposition algorithms and finite volume/finite element approximation for advection-diffusion equations. *J. Sci. Comput.* 11: 299–341.

[11] J. Douglas, Jr. and T.F. Russell, (1982). Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Num. Anal.*, 19:871–885.

[12] B.A. Finlayson, (1992). *Numerical methods for problems with moving fronts.* Ravenna Park Publishing, Seattle.

[13] F. Gastaldi, L. Gastaldi, and A. Quarteroni, (1996). Adaptive domain decomposition methods for advection dominated equations. *East-West J. Numer. Math.* 4: 165–206.

[14] R.W. Healy and T.F. Russell, (1998). Solution of the advection-dispersion equation in two dimensions by a finite-volume Eulerian-Lagrangian localized adjoint method. *Adv. Water Res.*, 21: 11–26.

[15] L. Marini and A. Quarteroni, (1989). A relaxation procedure for domain decomposition methods using finite elements. *Numer. Math.* 55: 575–598.

[16] G.F. Pinder and H.H. Cooper, (1970). A numerical technique for calculating the transient position of the saltwater front. *Water Resources Research*, 6: 875–882.

[17] O. Pironneau, (1982). On the transport-diffusion algorithm and its application to the Navier-Stokes equations. *Num. Math.*, 38:309–332.

[18] R. Rannacher and G. Zhou, (1994). Analysis of a domain-splitting method for nonstationary convection-diffusion problems. *East-West J. Num. Math.*, 2:151–172.

[19] T.F. Russell and R.V. Trujillo, (1990). Eulerian-Lagrangian localized adjoint methods with variable coefficients in multiple dimensions. In Gambolati *et al.* editors, *Computational Methods in Surface Hydrology*, pages 357–363. Springer-Verlag, Berlin.

[20] B.F. Smith, P.E. Bjørstad, and W.D. Gropp, (1996). *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press.

[21] X. Tai, T. Johansen, H. Dahle, and M. Espedal, (1997). A characteristic domain splitting method. In Glowinski *et al.*, editors, Domain Decomposition Methods in Science and Engineering, *Proceedings of Eighth International Conference on Domain Decomposition Methods*, pages 317–323, John-Wiley, New York.

[22] R. Trotta, (1996). Multidomain finite elements for advection-diffusion equations. *Appl. Numer. Math.* 21: 91–118.

[23] H. Wang, (1998). A family of ELLAM schemes for advection-diffusion-reaction equations and their convergence analyses. *Numerical Methods for PDEs*, 14: 739–780.

[24] H. Wang, (2000). An optimal-order error estimate for an ELLAM scheme for two-dimensional linear advection-diffusion equations. *SIAM J. Numer. Anal.*, 37: 1338–1368.

[25] H. Wang, H.K. Dahle, R.E. Ewing, M.S. Espedal, R.C. Sharpley, and S. Man, (1999). An ELLAM Scheme for advection-diffusion equations in two dimensions. *SIAM J. Sci. Comput.*, 20: 2160–2194.

[26] H. Wang, R.E. Ewing, G. Qin, S.L. Lyons, M. Al-Lawatia, and S. Man, (1999). A family of Eulerian-Lagrangian localized adjoint methods for multi-dimensional advection-reaction equations. *J. Comput. Phys.*, 152: 120–163.

[27] H. Wang, R. Ewing, and T. Russel, (1995). Eulerian-Lagrangian localized adjoint methods for convection-diffusion equations and their convergence analysis. *IMA Journal of Numerical Analysis* 15:405-459.

[28] H. Wang, D. Liang, R.E. Ewing, S.L. Lyons, and G. Qin, (2000) An ELLAM-MFEM solution technique for compressible fluid flows in porous media with point sources and sinks. *J. Comput. Phys.*, 159: 344–376.