Analysis of a Domain Decomposition Method for Linear Transport Problems on Networks

J.B. Collins^{1,*}

Department of Mathematics, West Texas A&M University Canyon, TX 79016, USA

P.A. Gremaud¹

Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA and Statistical and Applied Mathematical Sciences Institute, Research Triangle Park, NC 27709-4006, USA

Abstract

In this paper we analyze the convergence of the domain decomposition method applied to transport problems on networks. In particular, we derive estimates for the number of required iterations for linear problems. These estimates can be used to determine when the implementation of domain decomposition methods would be beneficial for this type of problems.

Keywords: Transport, network, domain decomposition

1. Introduction

Many physical processes can be, in first approximation, modeled as transport problems on networks; vehicular traffic flows [9, 10], blood flow [6, 14, 15, 16] or gas flow through pipe networks [2, 5] are but three examples. Large networks can have thousands or even millions of edges, as in vascular systems for instance. The choice of efficient numerical strategies is thus of paramount importance. The goal of the paper is to analyze under what circumstances the use of domain decomposition (DD) methods is advantageous.

*Principal corresponding author

Email addresses: jbcoll12@gmail.com (J.B. Collins), gremaud@ncsu.edu (P.A. Gremaud)

¹Partially supported by the National Science Foundation (NSF) through grant DMS-0811150 and DMS-1522765.

For simplicity, we consider *linear* transport problems on networks

$$\frac{\partial}{\partial t} \mathbf{q}_{e_i} + A_{e_i} \frac{\partial}{\partial x} \mathbf{q}_{e_i} = 0, \quad i = 1, \dots, M, \quad (1)$$

where e_i is the i -th edge of an M -edge network and where $\mathbf{q}_{e_i} \in \mathbb{R}^N$ stands for the state variables on e_i , $A_{e_i} \in \mathbb{R}^{N \times N}$ being a constant matrix associated to e_i . By assumption, our problem is one of transport; in other words, it is hyperbolic: each matrix A_{e_i} is diagonalizable with real eigenvalues. Junction conditions, usually algebraic constraints, have to be prescribed at the nodes of the network. These conditions can be quite complicated depending on the eigenstructure of A_{e_i} . To simplify our problem and focus on the analysis of the domain decomposition method, we consider the linear acoustics equations, which has symmetric eigenvalues and therefore relatively simple junction conditions can be imposed. While the details and numerical examples are given for these equations only, we show how the theory can be generalized to a linear constant coefficient system of the form (1), as long as appropriate junction conditions are given, which is very complicated in general, see Section 4.

DD methods were originally developed for steady state problems. In this work, the spatial dimension is decomposed into the various edges of the network. When domain decomposition is applied to time dependent problems, at least two approaches are possible. First, DD can be applied spatially at each time step, see for instance [1, 17]. Second, it can be applied in a spatio-temporal way, i.e., on cylinders corresponding to products of spatial subdomains with the entire computational time interval. This approach was introduced in [3, 4] and later applied to the one-dimensional wave equation [7, 8]. As detailed in Section 2, the second approach is used in this work, where the domains are the edges of the network.

Considerable work has been done in analyzing DD methods for transport problems defined on networks, see [11, 12, 13]. In this work, we use characteristics analysis to estimate the number of iterations necessary for the DD method to converge, disregarding numerical error. We show that this estimate depends upon on propagation speeds along each edge as well as on the length T of the considered time interval during which the problem is solved. This estimate is a key point in the cost study of the DD method for the present network problems. This analysis is carried out in Sections 3. Finally, Section 5 compares numerical results to the theoretical estimates.

2. Linear Acoustics Equations on a Network Domain

Consider a network $\mathcal{G} = (V, E)$, where V is the set of vertices (nodes) of the network, and E is the set of edges. Let M denote the total number of edges in the network. Each edge e_i is modeled by an interval $[a_{e_i}, b_{e_i}]$, $a_{e_i} < b_{e_i}$, possibly with either $a_{e_i} = -\infty$ or $b_{e_i} = \infty$. A semi-infinite interval implies that the edge is an outlier, i.e., it is connected to the network on only one side.

We consider the particular transport problem given by the acoustics equations. We focus on this problem because of its simplicity, especially where junction conditions are concerned. The symmetric nature of the problem allows for easily implemented junction conditions and simplifies the convergence analysis of Section 3.

The linear acoustics equation on an arbitrary edge $e_i \in E$ of the network $\mathcal{G}$ reads

$$\left(\frac{\partial}{\partial t} + \begin{bmatrix} 0 & K_{e_i} \\ 1/\rho_{e_i} & 0 \end{bmatrix} \frac{\partial}{\partial x} \right) \begin{bmatrix} p_{e_i} \\ u_{e_i} \end{bmatrix} (x, t) = 0, \quad x \in (a_{e_i}, b_{e_i}), t > 0, \quad (2)$$

where p_{e_i} is the pressure and u_{e_i} is the velocity. The physical parameters $K_{e_i}, \rho_{e_i} > 0$ are assumed constant on the domain (a_{e_i}, b_{e_i}) . Initial conditions are given by

$$p_{e_i}(x, 0) = \phi_{e_i}(x), \quad x \in [a_{e_i}, b_{e_i}], \quad (3)$$

$$u_{e_i}(x, 0) = \psi_{e_i}(x), \quad x \in [a_{e_i}, b_{e_i}]. \quad (4)$$

We assume $\phi_{e_i}(x) \in L^2([a_{e_i}, b_{e_i}])$, $\psi_{e_i}(x) \in L^2([a_{e_i}, b_{e_i}])$. Boundary conditions are introduced below when we consider the junction conditions for linear acoustics.

The eigenvalues of the matrix $\begin{bmatrix} 0 & K_{e_i} \\ 1/\rho_{e_i} & 0 \end{bmatrix}$ from (2) are given by

$$\lambda_1 = -c_{e_i} \quad \lambda_2 = c_{e_i}$$

where $c_{e_i} = \sqrt{K_{e_i}/\rho_{e_i}} > 0$ is the speed of sound. The problem is thus strictly hyperbolic. Through a change of variables,

$$w_{e_i} = \frac{1}{2Z_{e_i}}(-p_{e_i} + z_{e_i}u_{e_i}) \quad z_{e_i} = \frac{1}{2Z_{e_i}}(p_{e_i} + z_{e_i}u_{e_i})$$

where $Z_{e_i} = \rho_{e_i}c_{e_i}$, we can rewrite (2) as

$$\frac{\partial}{\partial t} \begin{bmatrix} w_{e_i} \\ z_{e_i} \end{bmatrix} (x, t) + \begin{bmatrix} -c_{e_i} & 0 \\ 0 & c_{e_i} \end{bmatrix} \frac{\partial}{\partial x} \begin{bmatrix} w_{e_i} \\ z_{e_i} \end{bmatrix} (x, t) = 0, \quad (5)$$

which is a decoupled system in terms of the characteristic variables w_{e_i} , and z_{e_i} . This form of the equations is used in the analysis below because of its direct connection to the speed and direction of propagation.

Given an edge e_i parameterized by the interval $[a_{e_i}, b_{e_i}]$, we denote the vertex corresponding a_{e_i} by e_i^a and the vertex corresponding to b_{e_i} by e_i^b . The various adjacent edges to e_i are also characterized in terms of their own orientation. More precisely, we set

$$\begin{aligned} \mathcal{A}_{e_i} &= \{e_j \in E : e_j^b = e_i^a\}, & \mathcal{B}_{e_i} &= \{e_j \in E : e_j^a = e_i^a\}, \\ \mathcal{C}_{e_i} &= \{e_j \in E : e_j^b = e_i^b\}, & \mathcal{D}_{e_i} &= \{e_j \in E : e_j^a = e_i^b\}. \end{aligned}$$

These adjacent edge sets are represented graphically in Figure 1.

Figure 1: The incoming and outgoing edges to edge e_i , denoted by the sets $\mathcal{A}_{e_i}, \mathcal{B}_{e_i}, \mathcal{C}_{e_i}, \mathcal{D}_{e_i}$.

Since, for the above system, one eigenvalue is always positive and one is always negative, the resolution of the problem on edge e_i requires one boundary condition at each end. Alternatively, and equivalently, if we consider a vertex $v \in V$ along with $\mathcal{G}|_v$, the set of edges connected to v ,

$$\mathcal{G}|_v = \{e_i \in E | e_i^a = v \text{ or } e_i^b = v\},$$

then the number of conditions, i.e., *junction conditions*, needed at v is equal to the number N_v of edges in $\mathcal{G}|_v$. The precise nature of these conditions

depends on the physics of the considered problem. Here, we impose (i) continuity of pressure at v and (ii) conservation of momentum at v . While not central to the purpose of the paper, this specific choice is both physically reasonable and compatible with the above considerations. When written in an edge based way, i.e., as boundary conditions for variables on edge e_i , these conditions take the form

$$\left. \begin{aligned} p_{e_i}(a_{e_i}, t) &= p_{e_j}(b_{e_j}, t) & \forall e_j \in \mathcal{A}_{e_i} \\ p_{e_i}(a_{e_i}, t) &= p_{e_\ell}(a_{e_\ell}, t) & \forall e_\ell \in \mathcal{B}_{e_i} \\ p_{e_i}(b_{e_i}, t) &= p_{e_m}(b_{e_m}, t) & \forall e_m \in \mathcal{C}_{e_i} \\ p_{e_i}(b_{e_i}, t) &= p_{e_n}(a_{e_n}, t) & \forall e_n \in \mathcal{D}_{e_i} \end{aligned} \right\} \quad \forall t > 0, \quad i = 1, \dots, N, \quad (6)$$

for continuity of pressure and,

$$\left. \begin{aligned} u_{e_i}(a_{e_i}, t) &= \sum_{e_j \in \mathcal{A}_{e_i}} u_{e_j}(b_{e_j}, t) - \sum_{e_\ell \in \mathcal{B}_{e_i}} u_{e_\ell}(a_{e_\ell}, t) \\ u_{e_i}(b_{e_i}, t) &= \sum_{e_n \in \mathcal{D}_{e_i}} u_{e_n}(a_{e_n}, t) - \sum_{e_m \in \mathcal{C}_{e_i}} u_{e_m}(b_{e_m}, t) \end{aligned} \right\} \quad \begin{aligned} \forall t > 0, \\ i = 1, \dots, N, \end{aligned} \quad (7)$$

for conservation of momentum.

For the sake of simplicity, we assume that outlier edges are semi-infinite to avoid having to impose additional boundary conditions. Such conditions could however easily be incorporated in the analysis below.

Analysis of hyperbolic network problems has been done extensively [11, 12, 13]. In particular, the general theorems in [11] include the problem defined by (2)-(4) together with the junction conditions (6)-(7). These theorems give existence and uniqueness results showing that this problem is well-posed.

2.1. Example Network Problem

A simple example is given to illustrate that the junction conditions (6)-(7) are sufficient boundary conditions to solve the network problem. Consider the network shown in Figure 2. Let us define the point in the domain of each edge that coincides with the vertex to be v , i.e. edge e_1 is parameterized by $(-\infty, v]$ and so on. We then write the junction condition as,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} p_{e_1}(v, t) \\ u_{e_1}(v, t) \\ p_{e_2}(v, t) \\ u_{e_2}(v, t) \\ p_{e_3}(v, t) \\ u_{e_3}(v, t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

Figure 2: A simple example network to demonstrate the junction conditions.

We note this system is underdetermined, however for hyperbolic problems, boundary conditions are set using the characteristic variables. Consider the equations relating the state variables to characteristic variables,

$$p_{e_i}(v, t) = Z_{e_i}(z_{e_i}(v, t) - w_{e_i}(v, t)) \quad (9)$$

$$u_{e_i}(v, t) = w_{e_i}(v, t) + z_{e_i}(v, t) \quad (10)$$

where $Z_{e_i} = \rho_{e_i} c_{e_i}$ and $i = 1, 2, 3$. Using (9)-(10), we can write the junction conditions in terms of characteristic variables. The outgoing variables, z_{e_1} , z_{e_2} , and w_{e_3} , that is those whose direction of propagation is away from the vertex, can be determined at the boundary by following characteristics back to previously computed or known data. Therefore we must determine the remaining three variables using the three junction conditions, which gives a well-determined system. Rearranging to put the known variables on one side, we obtain,

$$\begin{bmatrix} -Z_{e_1} & 0 & 0 \\ 0 & -Z_{e_2} & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} w_{e_1}(v, t) \\ w_{e_2}(v, t) \\ z_{e_3}(v, t) \end{bmatrix} = \begin{bmatrix} Z_{e_3}(z_{e_3}(v, t) - w_{e_3}(v, t)) - Z_{e_1}z_{e_1}(v, t) \\ Z_{e_3}(z_{e_3}(v, t) - w_{e_3}(v, t)) - Z_{e_2}z_{e_2}(v, t) \\ w_{e_3}(v, t) - z_{e_1}(v, t) - z_{e_2}(v, t) \end{bmatrix} \quad (11)$$

This linear system is invertible, so boundary conditions for the characteristic variables are well defined for all three edges in this network for all time.

The boundary conditions for the state variables are then computed from the boundary conditions of the characteristic variables. For general networks, a linear system similar to (11) is solved at each vertex.

2.2. Domain Decomposition Algorithm

For the domain decomposition algorithm we use an additive Schwarz [17]. We briefly define this algorithm for the linear acoustics problem in the following steps. First, we define the vector $\mathbf{q}_{e_i}^k(x, t) = [p_{e_i}^k(x, t), u_{e_i}^k(x, t)]^T$.

1. Define an initial iterate for the solution on all edges.

$$\mathbf{q}_{e_i}^0(x, t) = \mathbf{h}_{e_i}(x, t) \quad \forall e_i \in E, x \in (a_{e_i}, b_{e_i}), 0 < t \leq T \quad (12)$$

2. Partition the network into S disjoint connected subdomains $\mathcal{H}_j$,

$$\begin{aligned} \mathcal{G} &= \mathcal{H}_1 \cup \mathcal{H}_2 \cup \dots \cup \mathcal{H}_S \\ \mathcal{H}_j &= (V_j, E_j) \quad \forall j = 1, \dots, S \end{aligned}$$

3. For each subdomain $\mathcal{H}_j$, solve the following problem on each edge $e_i \in \mathcal{H}_j$ to final time $t = T$.

$$\begin{aligned} \frac{\partial}{\partial t}(\mathbf{q}_{e_i}^k) + \begin{bmatrix} 0 & K_{e_i} \\ 1/\rho_{e_i} & 0 \end{bmatrix} \frac{\partial}{\partial x}(\mathbf{q}_{e_i}^k) &= 0, \quad x \in (a_{e_i}, b_{e_i}), t \in (0, T], \\ \mathbf{q}_{e_i}^k(x, 0) &= \mathbf{\Phi}_{e_i}(x), \quad x \in [a_{e_i}, b_{e_i}]. \end{aligned}$$

where $\mathbf{\Phi}_{e_i}(x)$ is the initial condition on edge e_i . Note that we require the initial conditions to satisfy the junction conditions at each vertex. The boundary conditions are determined by the linear system of junction

conditions at each vertex,

$$J_{e_i^a} \begin{bmatrix} \mathbf{q}_{\hat{e}_1}^k(a_{e_i}, t) \\ \vdots \\ \mathbf{q}_{\hat{e}_L}^k(a_{e_i}, t) \\ \mathbf{q}_{\tilde{e}_1}^{k-1}(a_{e_i}, t) \\ \vdots \\ \mathbf{q}_{\tilde{e}_N}^{k-1}(a_{e_i}, t) \end{bmatrix} = 0 \quad \hat{e}_\ell \in \mathcal{G}|_{e_i^a} \cap \mathcal{H}_j, \tilde{e}_\ell \in \mathcal{G}|_{e_i^a} \setminus \mathcal{H}_j, \quad (13)$$

$$J_{e_i^b} \begin{bmatrix} \mathbf{q}_{\hat{e}_1}^k(b_{e_i}, t) \\ \vdots \\ \mathbf{q}_{\hat{e}_L}^k(b_{e_i}, t) \\ \mathbf{q}_{\tilde{e}_1}^{k-1}(b_{e_i}, t) \\ \vdots \\ \mathbf{q}_{\tilde{e}_N}^{k-1}(b_{e_i}, t) \end{bmatrix} = 0 \quad \hat{e}_\ell \in \mathcal{G}|_{e_i^b} \cap \mathcal{H}_j, \tilde{e}_\ell \in \mathcal{G}|_{e_i^b} \setminus \mathcal{H}_j. \quad (14)$$

Here the matrices $J_{e_i^a}$ and $J_{e_i^b}$ are similar to the matrix in (8) given in Section 2.1. The more general junction conditions (13)-(14) relate the state variables at iteration k to the ones at iterate $k-1$.

The general junction conditions (13)-(14) are given in terms of state variables to simplify notation, however to actually determine the boundary values we first convert to characteristic variables. To better explain, we consider the example in Section 2.1. First, we partition into two subdomains,

$$\begin{aligned} \mathcal{H}_1 &= e_1 \cup e_2 \\ \mathcal{H}_2 &= e_3 \end{aligned}$$

To determine $\mathbf{q}_{e_1}^k(v, t)$ and $\mathbf{q}_{e_2}^k(v, t)$, the boundary conditions for the state variables in $\mathcal{H}_1$ at the vertex v for the k^{th} iteration, we solve

$$\begin{bmatrix} -Z_{e_1} & 0 & 0 \\ 0 & -Z_{e_2} & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} w_{e_1}^k(v, t) \\ w_{e_2}^k(v, t) \\ z_{e_3}^{k-1}(v, t) \end{bmatrix} = \begin{bmatrix} Z_{e_3}(z_{e_3}^{k-1}(v, t) - w_{e_3}^{k-1}(v, t)) - Z_{e_1}z_{e_1}^k(v, t) \\ Z_{e_3}(z_{e_3}^{k-1}(v, t) - w_{e_3}^{k-1}(v, t)) - Z_{e_2}z_{e_2}^k(v, t) \\ w_{e_3}^{k-1}(v, t) - z_{e_1}^k(v, t) - z_{e_2}^k(v, t) \end{bmatrix}. \quad (15)$$

Therefore to determine the boundary conditions for $\mathbf{q}_{e_1}^k$ and $\mathbf{q}_{e_2}^k$ at the vertex v we solve (15) for $w_{e_1}^k(v, t)$ and $w_{e_2}^k(v, t)$. Then using initial

data we know $z_{e_1}^k(v, t)$ and $z_{e_2}^k(v, t)$ and can change from characteristic variables to state variables. This solves for the boundary condition for all $t > 0$. When we discretize, this step must be performed for each temporal node in the discretization.

Solving (15) we obtain $w_{e_1}^k(v, t)$ and $w_{e_2}^k(v, t)$ which are the boundary conditions for those two variables at the vertex v , and the boundary conditions $z_{e_1}^k(v, t)$ and $z_{e_2}^k(v, t)$ are known from initial data. Thus we can determine boundary conditions for $\mathbf{q}_{e_i}^k$, for $i = 1, 2$ by changing back to state variables. We note that we solve for one more variable than needed, namely $z_{e_3}^{k-1}(v, t)$. This is simply ignored when determining boundary conditions.

4. Repeat process until the solution converges in some meaningful way.

3. Convergence study

To simplify the convergence analysis, we work with exact solutions rather than with numerical approximations. Therefore, the only “error” that is introduced to the process comes from the initial iterate of the DD method, see (12). Our numerical results, see below, clearly show the effects of the discretization process on the convergence properties to be minimal.

We start by qualitatively demonstrating how the DD method converges for the acoustic equation given on networks. This is done on a simple network with many simplifying assumptions and is meant to give a feel for how convergence occurs. We then more formally analyze convergence on general networks with more general acoustics problems.

3.1. Convergence Example

We begin with a simple example to illustrate how the DD method converges for this problem. The network is a two edge loop network shown in Figure 3. We assume that the same acoustics equation is solved on both edges, and that the speed of sound is constant and given by $c_0 > 0$. Both edges are assumed to be of unit length and their parameterizations are $[0, 1]$. Two subdomains are used for the DD method, and one edge is assigned to each of them.

Examining the x - t plots of the solution on both edges after the first and second iterations, one can gain an idea of how the domain decomposition algorithm converges. We consider the characteristic variables w_{e_i} and z_{e_i} .

The convergence analysis is done using characteristic variables to easily determine the propagation of the error. The exact solution for z_{e_i} is reached in the domain $\{(x, t) : c_0 t < x\} \subset e_i \times [0, T]$. Likewise, for w_{e_i} , the exact solution is reached in $\{(x, t) : c_0 t < 1 - x\} \subset e_i \times [0, T]$, see Figure 4. We observe that after two iterations the method has converged up to time $t = 1/c_0$ for all x . Each repeated iteration increases the convergence domain and the process continues until the final time T is reached.

Figure 3: A 2 edge loop network.

Figure 4: The shaded area represents the part of the domain where convergence has been reached after one iteration, left, and two iterations, right. The green line represents characteristics for z , the red lines represent characteristics for w .

3.2. Convergence of General Problems

We consider the general network problem introduced in Section 2. In particular, the network $\mathcal{G}$ is any connected graph, with edges of varying length L_{e_i} . The sound speeds c_{e_i} can vary on each edge, while still being constant along the edge. To simplify the analysis we wish to reformulate our problem so that all interior edges have unit length and are parameterized by $[0, 1]$. This can be done via a change of variables $\hat{x} = \frac{x - a_{e_i}}{L_{e_i}}$. This change of variables transforms the speed of sound on edge e_i from c_{e_i} to $\hat{c}_{e_i} = \frac{c_{e_i}}{L_{e_i}}$. The sound speed $\hat{c}_{e_i}$ is used in the following work, with the “hat” removed for notational simplicity. By this change of variables, we encapsulate the effect of edge length within the sound speed. Finally, we partition the network so that each subdomain contains exactly one edge.

To more systematically analyze convergence, we introduce

$$\tau_{e_i}^k = \sup\{t > 0 : z_{e_i}(x, t) - z_{e_i}^k(x, t) = 0, \quad \forall x \in (a_{e_i}, b_{e_i})\}, \quad (16)$$

$$\sigma_{e_i}^k = \sup\{t > 0 : w_{e_i}(x, t) - w_{e_i}^k(x, t) = 0, \quad \forall x \in (a_{e_i}, b_{e_i})\}. \quad (17)$$

These convergence variables correspond to the largest time at which convergence has been reached on edge e_i after k iterations for each of the characteristic variables w_{e_i} and z_{e_i} . Through a simple analysis, one can then describe how these critical times increase with each additional iteration.

Theorem 1. *Convergence Variable Update Formula: Given a connected network $\mathcal{G}$ with the acoustic network problem as defined above, if we solve it with the additive Schwarz domain decomposition algorithm with the network partitioned such that each subdomain contains exactly one edge, the convergence variables satisfy the update formula,*

$$\tau_{e_i}^{k+1} \geq \min_{\substack{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i} \\ e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}}} \left\{ \tau_{e_j}^k + \frac{1}{c_{e_j}}, \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}}, \tau_{e_m}^k + \frac{1}{c_{e_m}} + \frac{1}{c_{e_i}}, \sigma_{e_n}^k + \frac{1}{c_{e_n}} + \frac{1}{c_{e_i}} \right\} \quad (18)$$

$$\sigma_{e_i}^{k+1} \geq \min_{\substack{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i} \\ e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}}} \left\{ \tau_{e_m}^k + \frac{1}{c_{e_m}}, \sigma_{e_n}^k + \frac{1}{c_{e_n}}, \tau_{e_j}^k + \frac{1}{c_{e_j}} + \frac{1}{c_{e_i}}, \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}} + \frac{1}{c_{e_i}} \right\} \quad (19)$$

To prove Theorem 1 we need the following lemma,

Lemma 1. *Let $\mathcal{E}(x, t)$ satisfy the following differential equation,*

$$\begin{cases} \frac{\partial}{\partial t} \mathcal{E}(x, t) + \alpha \frac{\partial}{\partial x} \mathcal{E}(x, t) = 0 & \forall x \in [0, 1], t > 0, \\ \mathcal{E}(x, 0) = 0, & \forall x \in [0, 1], \\ \mathcal{E}(0, t) = \psi_E(t), & t > 0, \end{cases} \quad (20)$$

where $\alpha > 0$. If $\mathcal{E}(x, t) = 0 \quad \forall x \in [0, 1]$ and $t < \omega$, then $\mathcal{E}(1, t) = 0 \quad \forall t < \omega + \frac{1}{\alpha}$.

PROOF. We can solve (20) to obtain,

$$\mathcal{E}(x, t) = \begin{cases} 0 & t \leq \frac{x}{\alpha} \\ \psi_E(t - \frac{x}{\alpha}) & t > \frac{x}{\alpha} \end{cases}.$$

Obviously, if $\mathcal{E}(x, t) = 0 \quad \forall x \in [0, 1]$ and $t < \omega$, then $\psi_E(s) = 0 \quad \forall s < \omega$. This then implies $\mathcal{E}(1, t) = 0$ for $t < \omega + \frac{1}{\alpha}$.

A similar lemma can be proved for negative speed of propagation.

Lemma 2. Let $\mathcal{F}(x, t)$ satisfy the following differential equation,

$$\begin{cases} \frac{\partial}{\partial t} \mathcal{F}(x, t) + \beta \frac{\partial}{\partial x} \mathcal{F}(x, t) = 0 & \forall x \in [0, 1], t > 0, \\ \mathcal{F}(x, 0) = 0, & \forall x \in [0, 1], \\ \mathcal{F}(1, t) = \psi_F(t), & t > 0, \end{cases} \quad (21)$$

where $\beta < 0$. If $\mathcal{F}(x, t) = 0 \quad \forall x \in [0, 1]$ and $t < \omega$, then $\mathcal{F}(0, t) = 0 \quad \forall t < \omega + \frac{1}{|\beta|}$.

We are now able to prove the update formula theorem.

PROOF. We begin by defining variables $\mathcal{E}_{e_i}^k = z_{e_i} - z_{e_i}^k$ and $\mathcal{F}_{e_i}^k = w_{e_i} - w_{e_i}^k$. We can then rewrite the convergence variables as

$$\tau_{e_i}^k = \sup\{t > 0 : \mathcal{E}_{e_i}^k = 0, \quad \forall x \in (a_{e_i}, b_{e_i})\}, \quad (22)$$

$$\sigma_{e_i}^k = \sup\{t > 0 : \mathcal{F}_{e_i}^k = 0, \quad \forall x \in (a_{e_i}, b_{e_i})\}. \quad (23)$$

Note that $\mathcal{E}_{e_i}^k$ and $\mathcal{F}_{e_i}^k$ satisfy (20) and (21) respectively, with boundary conditions $\psi_E^k(t)$ and $\psi_F^k(t)$. In particular, we consider $\mathcal{E}_{e_i}^{k+1}$. The boundary condition $\psi_E^{k+1}(t)$ is obtained by solving the linear system of junction conditions, similar to equation (15), and so depends on the incoming variables to that vertex, i.e. the set $S_z = \{\mathcal{E}_{e_j}^k(1, t) : e_j \in \mathcal{A}_{e_i}\} \cup \{\mathcal{F}_{e_\ell}^k(0, t) : e_\ell \in B_{e_i}\} \cup \mathcal{F}_{e_i}^{k+1}(0, t)$.

We assume that $\tau_{e_i}^k$ and $\sigma_{e_i}^k$ are known for $i = 1, \dots, N$ and observe that they act as ω in the previous lemmas for the variables $\mathcal{E}_{e_i}^k$ and $\mathcal{F}_{e_i}^k$ respectively.

Using the fact that $\tau_{e_i}^{k+1} \geq \sup\{t : \psi_E^{k+1}(t) = 0\}$ and the previous lemmas applied to the variables in the set S_z , we have,

$$\begin{aligned} \tau_{e_i}^{k+1} &\geq \sup\{t : \psi_E^{k+1}(t) = 0\} \\ &\geq \min_{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i}} \left\{ \tau_{e_j}^k + \frac{1}{c_{e_j}}, \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}}, \sigma_{e_i}^{k+1} + \frac{1}{c_{e_i}} \right\}. \end{aligned} \quad (24)$$

Through similar arguments with $\mathcal{F}_{e_i}^{k+1}$ we also have,

$$\sigma_{e_i}^{k+1} \geq \min_{e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}} \left\{ \tau_{e_m}^k + \frac{1}{c_{e_m}}, \sigma_{e_n}^k + \frac{1}{c_{e_n}}, \tau_{e_i}^{k+1} + \frac{1}{c_{e_i}} \right\}. \quad (25)$$

We note that (24) and (25) are coupled. Therefore by substituting we obtain,

$$\begin{aligned} \tau_{e_i}^{k+1} &\geq \min_{\substack{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i} \\ e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}}} \left\{ \tau_{e_j}^k + \frac{1}{c_{e_j}}, \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}}, \tau_{e_m}^k + \frac{1}{c_{e_m}} + \frac{1}{c_{e_i}}, \right. \\ &\quad \left. \sigma_{e_n}^k + \frac{1}{c_{e_n}} + \frac{1}{c_{e_i}}, \tau_{e_i}^{k+1} + \frac{2}{c_{e_i}} \right\} \\ &= \min_{\substack{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i} \\ e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}}} \left\{ \tau_{e_j}^k + \frac{1}{c_{e_j}}, \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}}, \tau_{e_m}^k + \frac{1}{c_{e_m}} + \frac{1}{c_{e_i}}, \sigma_{e_n}^k + \frac{1}{c_{e_n}} + \frac{1}{c_{e_i}} \right\}, \end{aligned} \quad (26)$$

and,

$$\begin{aligned} \sigma_{e_i}^{k+1} &\geq \min_{\substack{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i} \\ e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}}} \left\{ \tau_{e_m}^k + \frac{1}{c_{e_m}}, \sigma_{e_n}^k + \frac{1}{c_{e_n}}, \tau_{e_j}^k + \frac{1}{c_{e_j}} + \frac{1}{c_{e_i}}, \right. \\ &\quad \left. \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}} + \frac{1}{c_{e_i}}, \sigma_{e_i}^{k+1} + \frac{2}{c_{e_i}} \right\} \\ &= \min_{\substack{e_j \in \mathcal{A}_{e_i}, e_\ell \in \mathcal{B}_{e_i} \\ e_m \in \mathcal{C}_{e_i}, e_n \in \mathcal{D}_{e_i}}} \left\{ \tau_{e_m}^k + \frac{1}{c_{e_m}}, \sigma_{e_n}^k + \frac{1}{c_{e_n}}, \tau_{e_j}^k + \frac{1}{c_{e_j}} + \frac{1}{c_{e_i}}, \sigma_{e_\ell}^k + \frac{1}{c_{e_\ell}} + \frac{1}{c_{e_i}} \right\}. \end{aligned} \quad (27)$$

The last term is dropped from each min function because it can never be selected for $c_{e_i} < \infty$. ■

These update formulas for the convergence variables are useful for determining how many iterations are needed for the solution to converge to a certain final time. While theoretical results cannot be proven for nonuniform sound speed, the convergence variables can be inexpensively computed and an estimate of the number of iterations required can be obtained for specific problems. Examples of this are given in Section 5.

Finally, we give a theoretical result for a specific class of problems, where all edges have equal sound speed c_0 . In this case we have the following theorem.

Theorem 2. *Given an acoustic network problem as defined in Theorem 1, where the sound speed on all edges is the constant value $c_0 > 0$, then the total number of iterations needed for convergence is bounded by,*

$$k \leq Tc_0 + 1, \quad (28)$$

where T is the final time.

PROOF. We first prove by induction that the convergence variables satisfy,

$$\tau_{e_i}^k \geq \frac{k-1}{c_0}, \quad (29)$$

$$\sigma_{e_i}^k \geq \frac{k-1}{c_0}, \quad (30)$$

for all edges e_i and $k > 0$. We know that $\tau_{e_i}^1 = \sigma_{e_i}^1 = 0$ for all edges, so (29)-(30) hold for $k = 1$. Assuming these equations hold for $k - 1$, and using (18)-(19) from Theorem 1, we obtain,

$$\tau_{e_i}^k \geq \min \left\{ \frac{k-2}{c_0} + \frac{1}{c_0}, \frac{k-1}{c_0} + \frac{1}{c_0} \right\} = \frac{k-1}{c_0}, \quad (31)$$

$$\sigma_{e_i}^k \geq \min \left\{ \frac{k-2}{c_0} + \frac{1}{c_0}, \frac{k-1}{c_0} + \frac{1}{c_0} \right\} = \frac{k-1}{c_0}. \quad (32)$$

Setting $\tau_{e_i}^k = \sigma_{e_i}^k = T$ and solving for k we obtain (28). ■

Remark 1. We note that Theorem 2 can be extended to a network with edges of uniform length L . In this case the sound speed on each edge is $\frac{c_0}{L}$ and the iteration estimate is,

$$k \leq \frac{Tc_0}{L} + 1.$$

3.3. Multi-Edge Subdomains

Let's consider now grouping edges in more general subdomains. We continue to utilize convergence variables to track the convergence of the method as the iterations progress. A heuristic update formula similar to (26)-(27) is given but not proven. However, numerical experiments verify this update formula is valid.

For simplicity, we consider a single network, shown in Figure 5, with constant speed $c_0 > 0$ on all edges and unit edge length. This network is an example of the a “caveman” network in which a group of edges have high intraconnectivity among themselves and low interconnectivity to the rest of the network. We partition this network into these clusters of high intraconnectivity.

Figure 5: A network split into three domains, shown by the different line types.

By observing how this problem is solved with the DD method, we obtain

the following update formulas for the convergence variables $\tau_{e_i}^k$ and $\sigma_{e_i}^k$.

$$\tau_{e_i}^{k+1} \geq \min_{\substack{e_j \in \mathcal{A}_{e_i}^O, e_\ell \in \mathcal{B}_{e_i}^O \\ e_m \in \mathcal{C}_{e_i}^O, e_n \in \mathcal{D}_{e_i}^O}} \left\{ \tau_{e_j}^k + \frac{\mathcal{L}_{ij} + 1}{c_0}, \sigma_{e_\ell}^k + \frac{\mathcal{L}_{il} + 1}{c_0}, \right. \quad (33)$$

$$\left. \tau_{e_m}^k + \frac{\mathcal{N}_{im} + 2}{c_0}, \sigma_{e_n}^k + \frac{\mathcal{N}_{in} + 2}{c_0} \right\},$$

$$\sigma_{e_i}^{k+1} \geq \min_{\substack{e_j \in \mathcal{A}_{e_i}^O, e_\ell \in \mathcal{B}_{e_i}^O \\ e_m \in \mathcal{C}_{e_i}^O, e_n \in \mathcal{D}_{e_i}^O}} \left\{ \tau_{e_m}^k + \frac{\mathcal{N}_{im} + 1}{c_0}, \sigma_{e_n}^k + \frac{\mathcal{N}_{in} + 1}{c_0}, \right. \quad (34)$$

$$\left. \tau_{e_j}^k + \frac{\mathcal{L}_{ij} + 2}{c_0}, \sigma_{e_\ell}^k + \frac{\mathcal{L}_{il} + 2}{c_0} \right\}.$$

We define the sets $\mathcal{A}_{e_i}^O, \mathcal{B}_{e_i}^O, \mathcal{C}_{e_i}^O, \mathcal{D}_{e_i}^O$ by,

$$\mathcal{A}_{e_i}^O = \{e_j \in E \setminus \mathcal{H}_{e_i} : e_j^b = e_\ell^a \text{ for some } e_\ell \in \mathcal{H}_{e_i}\} \quad (35)$$

$$\mathcal{B}_{e_i}^O = \{e_j \in E \setminus \mathcal{H}_{e_i} : e_j^a = e_\ell^a \text{ for some } e_\ell \in \mathcal{H}_{e_i}\} \quad (36)$$

$$\mathcal{C}_{e_i}^O = \{e_j \in E \setminus \mathcal{H}_{e_i} : e_j^b = e_\ell^b \text{ for some } e_\ell \in \mathcal{H}_{e_i}\} \quad (37)$$

$$\mathcal{D}_{e_i}^O = \{e_j \in E \setminus \mathcal{H}_{e_i} : e_j^a = e_\ell^b \text{ for some } e_\ell \in \mathcal{H}_{e_i}\}. \quad (38)$$

where $\mathcal{H}_{e_i}$ is the subdomain containing edge e_i . The sets defined in (35)-(38) are the multi-edge subdomain counterparts to $\mathcal{A}_{e_i}, \mathcal{B}_{e_i}, \mathcal{C}_{e_i}, \mathcal{D}_{e_i}$ from Section 2, and are in fact identical to them when only one edge is assigned to each subdomain. The numbers $\mathcal{L}_{ij}$ and $\mathcal{N}_{ij}$ represent the number of edges in the shortest path from the beginning and end of edge e_j to $\mathcal{H}_{e_i}$. These numbers take into account the fact that some edges are not directly connected to the adjacent subdomains.

For the network given in Figure 5, the subnetworks contained in each of the subdomains are identical. Consequently, the convergence properties are identical in each of these subdomains. By observation and induction, we can reduce the update formulas (33)-(34) to the following (considering the first subdomain for instance),

$$\tau_a^{k+1} = \sigma_b^{k+1} = \sigma_c^{k+1} = \frac{2k-1}{c_0} \quad (39)$$

$$\sigma_a^{k+1} = \tau_b^{k+1} = \tau_c^{k+1} = \frac{2k}{c_0} \quad (40)$$

Note that the two different formulas correspond to two different classes of characteristic variables. The first formula corresponds to variables that obtain their boundary data from a node directly connected to an adjacent subdomain, while the variables corresponding to the second formula obtain their boundary data from a node completely interior to the subdomain. This is why the second formula converges faster.

Using this result, we can estimate the number of iterations needed to solve the entire network to final time T by,

$$k = \left\lceil \frac{Tc_0 + 3}{2} \right\rceil. \quad (41)$$

By using larger domains convergence is achieved in approximately half the iterations for this example network. The solution converges faster on the large subdomains because the error from the initial iterate is not introduced directly on both sides of the edge. Instead, portions of the error must travel through the subdomain to reach certain edges. This allows the solution to be solved exactly for longer times before error is introduced.

4. General Linear Transport Network Problems

While all of the above work has been done exclusively for the linear acoustics equations, these results can be generalized for any system of the form (1), as long as appropriate junction conditions are imposed. We note two things about generalizing beyond the linear acoustics problem. First, we continue to consider constant coefficient problems on each edge. Variable coefficients can significantly complicate the analysis for the DD convergence. Second, obtaining well-posed junction conditions for general problems is quite difficult and questions of existence and uniqueness must be considered before DD analysis can be done. We assume that junction conditions are imposed in such a way that the network problem is well-posed. We ignore these issues in order to focus on the analysis of the DD method, but note that they are significant and must be addressed before this analysis can be applied. We also require that the characteristic structure along each edge must be the same. This amounts to the number of positive and negative eigenvalues being the same for each A_{e_i} .

To accommodate general problems of the form (1), one convergence variable must be defined for each characteristic variable. We can define these

by $\chi_{e_i,j}$ where e_i represents the edge and j represents the characteristic variable. Using this notation for the linear acoustics problem we would have $\sigma_{e_i} = \chi_{e_i,1}$ and $\tau_{e_i} = \chi_{e_i,2}$. Let us order the characteristic variables by their corresponding eigenvalues such that

$$\begin{aligned}\lambda_j &< 0, & 1 \leq j \leq p \\ \lambda_j &> 0, & p+1 \leq j \leq M\end{aligned}$$

where M is the number of state variable. Assuming uniform speed and edge length, that is $A_{e_i} = A_0$ for any $e_i \in E$, the update formula for the convergence variables is given by,

$$\chi_{e_i,j}^{k+1} \geq \begin{cases} \min\{\chi_{e_m,q}^k, \chi_{e_n,r}^k \mid m \in \mathcal{C}_i, n \in \mathcal{D}_i, p+1 \leq q \leq N, 1 \leq r \leq p\} + \frac{1}{\lambda_j}, \\ \text{if } 1 \leq j \leq p, \\ \min\{\chi_{e_m,q}^k, \chi_{e_n,r}^k \mid m \in \mathcal{A}_i, n \in \mathcal{B}_i, p+1 \leq q \leq N, 1 \leq r \leq p\} + \frac{1}{\lambda_j}, \\ \text{if } p+1 \leq j \leq N. \end{cases} \quad (42)$$

There are obvious parallels between the update formulas (18)-(19) and the update formula (42). Similar generalizations can be made for non-uniform constant A_{e_i} and for multi-edge subdomains. Difficulties occur when the coefficient matrix depends upon space and time, i.e., $A_{e_i} = A_{e_i}(x, t)$ although a more complicated update formula could account for the change in propagation speed over time and space.

5. Numerical Results

The above analysis is based on properties of the exact solutions to our problems. Do similar convergence results for DD still hold after discretization?

This question is studied here through the use of a second order central scheme [18] and a second order explicit TVD Runge-Kutta scheme in time. A uniform mesh of 160 nodes per edge and a CLF number of $\lambda = \frac{3}{4}$ are used to compute the problem. The initial conditions are chosen as compatible with the boundary conditions (6)-(7). The initial iterate is a constant function for all x and t . This initial iterate is neither compatible with the boundary conditions, nor at all close to the exact solution. This initial iterate is chosen

to illustrate that the convergence analysis does not depend on an accurate initial iterate. The final time is set to $T = \frac{6}{c_0}$ for those results using uniform speed of propagation. For variable speed of propagation, the final time used is $T = 6/\max(c_{e_i})$. The domain decomposition iteration is terminated when the solutions from each iteration begin to get closer together in the L^1 norm.

The junction conditions are implemented exactly, as described in Section 2. That is, the characteristic variables are set at each vertex. The incoming characteristic variables are set using data from the previous time step, while the outgoing variables are set by solving a linear system obtained from the junction conditions.

We consider the three cases from Section 3: uniform speed and edge length, variable speed, and multi-edge subdomains. For variable speed, a random speed is chosen on each edge. While it is possible in the case of uniform propagation speed to obtain general theoretical results, the more general cases are too algebraically involved. However, even then, the update formulas (18) and (19) can still be computed for a specific problem and an estimate of the number of iterations can be obtained.

Figure 6: The three generation binary tree network. When using multi edge subdomains, the network is partitioned as shown.

5.1. Three Generation Binary Tree

Table 1 presents the number of iterations needed for convergence of the DD method for the three generation binary tree shown in Figure 6. The

analytical estimates for uniform speed obtained using the update formulas agree with Theorem 2. The actual number of iterations is about two more than expected. This is due to numerical error introduced by the discretization. These extra iterations are required to remove the numerical error that is introduced from the discretization. Note that when large subdomains are used, the iteration number is roughly cut in half.

	uniform speed	variable speed	multi edge subdomain
estimated	7	5	3
actual	9	7	5

Table 1: Comparison of analytical bounds to iteration count for the 3 generation binary tree from Figure 6.

5.2. Three Clusters Network

In Table 2, we show results for the three clusters network in Figure 5. The analytical bounds for multi edge subdomains agree with the bounds presented above. Again, two to three extra iterations are required to filter out the numerical error.

	uniform speed	variable speed	multi edge subdomain
estimated	7	7	5
actual	10	9	6

Table 2: Comparison of analytical bounds to iteration count for the 3 Clusters Network from Figure 5.

5.3. Bird Island Network

Finally, in Table 3 we consider the road network on Bird Island, NC as seen in Figure 7. While we still solve the acoustics equations, we use a larger network to demonstrate that the results continue to hold as the size of the network increases. The update formulas still estimate the iteration count within a couple of iterations.

Figure 7: The Bird Island, NC traffic network. When using multi edge subdomains, the network is partitioned into 6 sections as shown.

	uniform speed	variable speed	multi edge subdomain
estimated	7	6	6
actual	11	8	8

Table 3: Comparison of analytical bounds to iteration count for the Bird Island Network from Figure 7.

6. Conclusion

Network problems lend themselves well to domain decomposition. However, there are some situations where solving the problem serially is faster. By using the convergence theory developed in this paper, one can determine a priori which method is the fastest for a given problem. If we assume that solving the problem serially take $\mathcal{O}(1)$ work, then solving the problem in parallel will take roughly $\mathcal{O}(\frac{L}{P})$, where L is the number of iterations of the Schwarz method estimated as above and P is the number of subdomains used. While it ignores communication overhead, this simplified calculation indicates that DD is potentially advantageous only if $L < P$. This also raises interesting questions of scalability since L depends not only on the number of

subdomains, but also on how those subdomains are organized. An interesting avenue of future work would be to investigate how the number of iterations changes as the number of subdomains increases.

- [1] A. Bamberger, R. Glowinski and Q.H. Tran, *A Domain Decomposition Method for the Acoustic Wave Equations with Discontinuous Coefficients and Grid Change*, SIAM J. Numer. Anal., 34 (1997), pp. 603-639.
- [2] M.K. Banda, M. Herty and A. Klar, *Coupling Conditions for Gas Networks Governed by the Isothermal Euler Equations*, Networks and Heterogeneous Media, 1 (2006), pp. 275-294.
- [3] M. Bjørhus, *A Note on the Convergence of Discretized Dynamic Iteration*, BIT 35 (1995) pp. 291-296.
- [4] M. Bjørhus, *On Domain Decomposition, Subdomain Iteration and Waveform Relaxation*, Ph.D. thesis, University of Trondheim, Trondheim, Norway, 1995.
- [5] R.M. Colombo and M. Garavello, *A Well Posed Riemann Problem for the p -system at a Junction*, Networks and Heterogeneous Media, 1 (2006), pp. 495-511.
- [6] K.J. DeVault, P.A. Gremaud, V. Novak, M.S. Olufson, G. Vernieres and P. Zhao, *Blood Flow in the Circle of Willis: Modeling and Calibration*, Multiscale Model. Simul., 7 (2008), pp. 888-909.
- [7] M. J. Gander, L. Halpern, and F. Nataf, *Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation*, in Eleventh International Conference of Domain Decomposition Methods, C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, eds., ddm.org, 1999.
- [8] M. Gander, L. Halpern, and F. Nataf, *Optimal Schwarz Waveform Relaxation for the One-Dimensional Wave Equation*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1643-1681.
- [9] M. Garavello and B. Piccoli, *Traffic Flow on a Road network Using the Aw-Rascle Model*, Comm. in Partial Differential Equations, 31 (2006), pp. 243-275.

- [10] H. Holden and N. Risebro, *A Mathematical Model of Traffic Flow on a Network of Unidirectional Roads*, SIAM J. Math. Anal., 26 (1995), pp. 999-1017.
- [11] Lagnese, John E., E. J. Schmidt, and Gunter Leugering. Modeling, analysis and control of dynamic elastic multi-link structures. Birkhauser Boston, 1994.
- [12] Leugnerin, Günder, *Dynamic Domain Decomposition of Optimal Control Problems for Networks of Strings and Timoshenko Beams*, SIAM J. Control Optim., 37(6), 1649-1675.
- [13] Leugering, Gnter. Domain in Decomposition Methods in Optimal Control of Partial Differential Equations. No. 148. Springer, 2004.
- [14] K.S. Matthys, J. Alastruey, J. Peiró, A.W. Khir, P. Segers, P.R. Verdonck, K.H. Parker and S.J. Sherwin, *Pulse wave propagation in a model human arterial network: assessment of 1-D numerical simulations against in vitro measurements*, J. Biomechanics, 40 (2007), pp. 3476-3486.
- [15] S.M. Moore, K.T. Moorhead, J.G. Chase, T. David and J. Fink, *One-dimensional and three dimensional models of cerebrovascular flow*, Trans. ASME, **127** (2005), pp. 440-449.
- [16] T. Passerini, M. de Luca, L. Formaggia, A. Quarteroni and A. Veneziani, *A 3D/1D geometrical multiscale model of cerebral vasculature*, J. Engrg. Math., 64 (2009), pp. 319-330.
- [17] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Oxford University Press, New York, 1999.
- [18] A. Kurganov, E. Tadmor, *New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations*, Journal of Computational Physics 160 (2000), pp. 241-282.