

Implementation of 1-D Vascular Model using Structured Tree Outflow Conditions

Alex Baelde, Adam Updegrave, Debanjan Mukherjee

May 16, 2014

1 Introduction

Using the paper reviewed in Journal Club, we attempted to implement the Structured Tree Outflow conditions for a 1-D Vascular Model presented by Olufsen [?]. The 1-D model presented by Olufsen outlines a finite difference scheme for solving the Navier-Stokes equations; as such, we used this approach for implementation of our 1-D model. We will briefly outline the purpose of using a Structured Tree model and the theory behind this implementation. We will then describe the process taken in implementing this code along with some issues encountered along the way. We will discuss the current state of our code as well as display some preliminary results of our implementation. Lastly, we will talk about a small study investigated using Olufsen's code.

2 Theory

2.1 Why Structured Tree

Modeling the human body, in the current state of being, is a complex and difficult process. Large 3-D models that account for large portions of the human vasculature take immense amounts of computation time to solve for the flow within the body. On the other hand, smaller models that take into account the rest of the body a set resistance or circuit (lumped parameter models) cannot accurately resolve the physiological wave reflections in the human vasculature [?]. The Structured Tree 1-D model lies somewhere in between the two complexities. It is a 1-D model, so computational time is significantly decreases; in addition, the wave reflections of the human vasculature are captured because the impedance of the human body is modeled as an impedance based on frequency of the downstream vessels (The Structured Tree). It is important to note that the formation of this tree, specifically the

root impedance can have an effect on the solution. The effect of a change in root impedance is investigated later in the paper.

2.2 Inflow

For the inflow into the 1-D model, a mathematical model fit to measured data was used to represent the pulses of the human cardiac cycle. This mathematical model is described in Olufsen's thesis [?]. This is imposed as an initial condition in the first cardiac period. The following plot is produced to describe the velocity for the inlet.

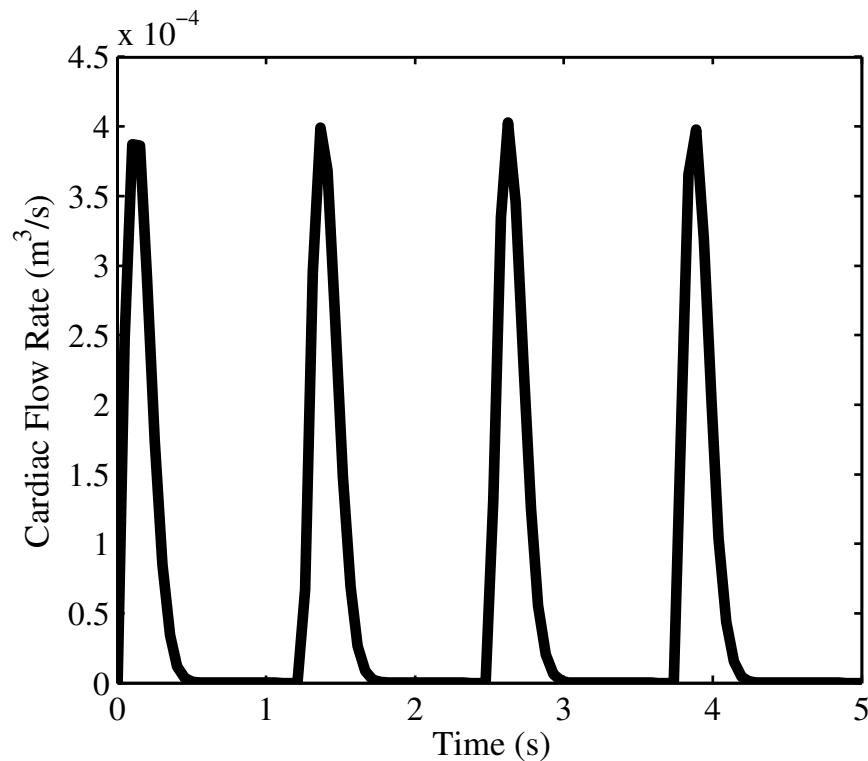


Figure 1: The inflow profile from measured data in m^3/s . The curve is measured at the upper ascending aorta using magnetic resonance.

2.3 Interior

As is commonly the case, blood is modeled as an incompressible, newtonian fluid. The navier-stokes equation of motion are therefore applied to the fluid. Starting with a simplified version of continuity, we get Equation (??), and the simplified conservation of momentum in the x direction we get Equation (??).

$$\frac{\partial A}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad (1)$$

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q^2}{A} \right) + \frac{A}{p} \frac{\partial p}{\partial x} = - \frac{2R\pi\nu q}{\delta A} \quad (2)$$

For implementation of these non-linear equations, a **Two-Step** Lax-Wendroff (Richtmyer) scheme is used. Equations ?? and ?? can be written in the following form:

$$\frac{\partial \bar{U}}{\partial t} + \frac{\partial \bar{R}(\bar{U})}{\partial x} = 0 \quad (3)$$

This equation is then discretized into the two equations to be solved over the interior domain.

1.

$$\bar{U}_j^{T+1} = \bar{U}_j^T - \frac{k}{h} \left(\bar{R}_{j+1/2}^{T+1/2} - \bar{R}_{j-1/2}^{T+1/2} \right) + \frac{k}{2} \left(\bar{S}_{j+1/2}^{T+1/2} + \bar{S}_{j-1/2}^{T+1/2} \right) \quad (4)$$

2.

$$\bar{U}_j^{T+1/2} = \frac{\bar{U}_{j+1/2}^T + \bar{U}_{j-1/2}^T}{2} + \frac{k}{2h} \left(-\bar{R}_{j+1/2}^T - \bar{R}_{j-1/2}^T \right) + \frac{k}{4} \left(\bar{S}_{j+1/2}^T + \bar{S}_{j-1/2}^T \right) \quad (5)$$

In terms of implementing the equation, the very broad algorithm can be viewed to help describe the process

2.4 Outflow

Our implementation of the structured tree actually varies slightly from Olufsen's implementation. In her code, Olufsen runs through each vessel individually and calculates the radius of the vessel based on Equation (??). She retains no structure of the vessels within the tree and recalculates all this information for each structured tree.

$$(r_0)(n, k) = R_{root} \alpha^k \beta^{n-k} \quad (6)$$

where α and β are constants that describe the reduction and asymmetry of the vascular tree. n is the number of total generations of the tree and k is the current generation. She then

Algorithm 1 Implementation

```
1: Set up problem based on user input.
2: Solve for impedance tree.
3: for  $i = 1$  : Number of Cardiac Periods
4:   if 1st Cardiac Period
5:     Apply the inflow condition.
6:   else
7:     Use the flow from previous period.
8:   end if
9:   for  $j = 1$  : Time in Cardiac Period
10:    for  $k = 1$  : Number of Nodes in Domain
11:      Solve equations ?? and ?? for each component of the  $\bar{U}$ ,  $\bar{R}$ , and  $\bar{S}$  vectors.
12:    end for
13:    Apply the outflow condition.
14:  end for
15:  Save the Period Information.
16: end for
```

calculates the corresponding impedance in the same recursive step and returns the total tree impedance at each corresponding frequency.

Our algorithm, however, is based off of a class that we have defined in order to retain information about the tree. We iterate through each generation of the tree and develop two daughters from each branch by applying α and β separately to each member. In the end, we have an array that describes the breakdown of this entire tree with the following format:

$$[Daughter\ 1 \quad Daughter\ 2 \quad Parent \quad Radius] \quad (7)$$

The Inverse Fourier Transform of the impedance is taken in order to get the impedances as a function of time. These impedances are then used to couple the outflow condition to the solution by either using another Lax-Wendroff discretization or a method of Positive Characteristics. This was actually something that is extremely different between Olufsen's thesis and her code, and actually became a big turning point in getting our code working. In her thesis, she states that a Lax-Wendroff method must also be used at the outflow in order to guarantee 2nd order accuracy; however, she doesn't use this method and instead uses a method of positive characteristics. We implemented the method of positive characteristics as an alternative to the Lax-Wendroff discretization to provide an alternate, more reduced approach.

$$\Gamma_{+/-} : A_Q - A_{R/S} + \frac{Q_Q - Q_{R/S}}{-(Q_{R/S}/A_{R/S}) + C_R} = H_{R/S}^{+/-} \Delta t \quad (8)$$

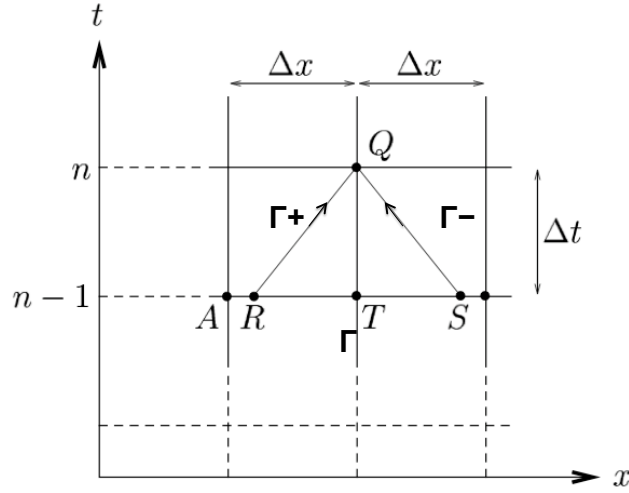


Figure 2: At the right boundary, or the outlet, the source vector \bar{S} , and the \bar{R} converge at point Q using the method of positive characteristics. [?]

3 Methods and Results

3.1 Our Implementation [?]

For future ease of use, Matlab was used as the coding interface. This will allow future students and lab members to easily use the code and either experiment or make modifications to the code. The only down side to this is the fact that the code runs much slower than if it were implemented in fortran or c. The drastic effect of computation time is seen by the mere seconds it takes for Olufsen's code to finish compared to the ten minutes or so it takes for the Matlab implementation. With our code, we are at a point where the code runs, but only for problems of very carefully specified parameters. In addition, for debugging purposes, we set up the code to be able to use a resistance boundary condition to try to isolate whether the problem was in the interior of the domain or while implementing the boundary conditions. This helped to figure out separate issues in the code, as there seemed to be problems with both. The code now works for certain test cases using a resistance boundary condition, but *interesting* things are happening for other problems. The following problem is run on a tube with a length of 100 cm and a radius of 1 cm. This case was run in order to try to compare with a test case run in Olufsen's paper. Olufsen runs a test case using a resistance boundary condition on a tube of equal length, but with decreasing radius. Her radius starts at 0.5 cm and decreases to 0.25 cm. The reason we have not run this exact test size is because the code cannot seem to handle the change in radius at this point. This is likely the reason for the discrepancy in both flow and pressure values. The profiles, however, look fairly similar and can be compared. The results of two problems are shown below:

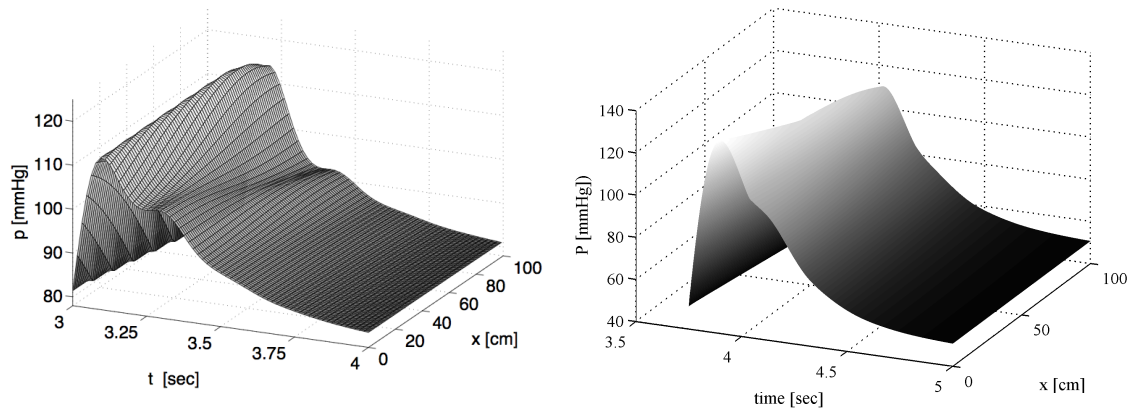



Figure 3: Surface plot of the pressure along the tube for one cardiac period. **Left:** Plot from Olufsen's thesis during fourth cardiac cycle. **Right:** Plot from our implementation during fourth cardiac cycle. Our implementation seems to have less of a dicrotic notch after the peak pressure in the cardiac cycle. 

3.2 Olufsen Implementation

As you know, we were able to get in touch with Olufsen, and she generously sent us her code via e-mail. We were able to compile, run, and post process her code. Because our code is not to the point where we can investigate some of the things we were hoping to, we modified her code in certain ways to display some interesting things. We investigated the following things:

- Effect of decreasing the cardiac period
- Blocking flow in certain arteries (carotid)
- Imposing a root impedance in certain arteries

We were able to see from these...

4 Conclusion and Discussion

We have learned an immense amount from this exercise, both from a development point of view and a cardiovascular biomechanics view. We now understand some complications that may arise when attempting to imitate another person's work. We made the decision to go ahead and use the schemes that Olufsen was using assuming they were consistent, well documented, and working. This may not always be the case, and looking back, we realize

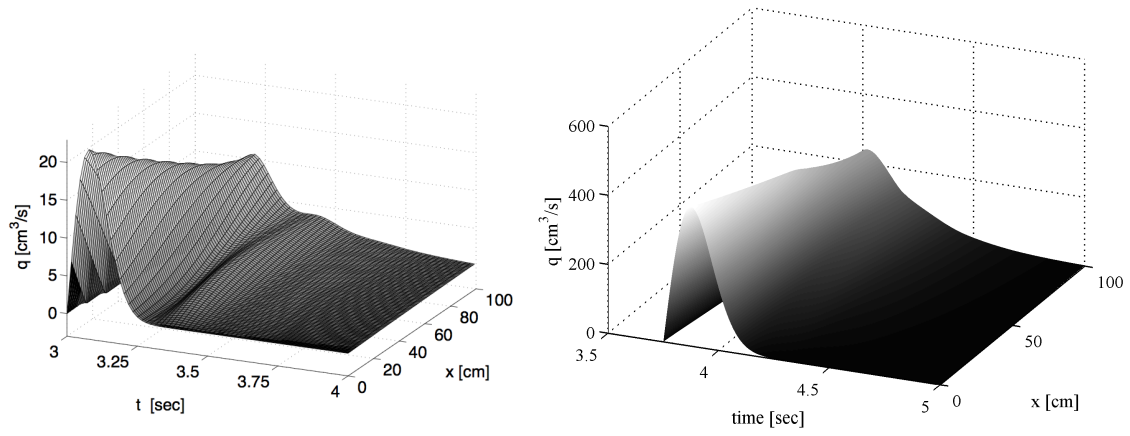


Figure 4: Surface plot of the flow in cm^3/s along the tube for one cardiac period. **Left:** Plot from Olufsen's thesis during fourth cardiac cycle. **Right:** Plot from our implementation during fourth cardiac cycle. Our implementation retains the similar profile, but again, the slight notch in the curve is less visible in our implementation.

it may have been a lot simpler and easier to bug/write our code had we developed our own discretization scheme or method.

Also talk about what we learn from Olufsen's code