Numerical Model to Predict Hemolysis and Transport in a Membrane-Based Microfluidic Device

Matthew D. Poskus
mdp5882@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Numerical Model to Predict Hemolysis and Transport in a Membrane-Based Microfluidic Device

By:

Matthew D. Poskus

A Thesis Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Approved By:

Dr. Steven Day – Thesis Advisor
Department of Biomedical Engineering

Dr. Kathleen Lamkin-Kennard
Department of Mechanical Engineering

Dr. Thomas Gaborski
Department of Biomedical Engineering

Dr. Alan Nye – Department Representative
Department of Mechanical Engineering

Department of Mechanical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
July, 2019
Copyrights

Copyright 2019 © Matthew D. Poskus

The use of this publication, as a whole or in part including data, for public education efforts and noncommercial purposes, is strongly encouraged and requires no expressed authorization. It is requested, however, that such use is accompanied by a citation as specified in a recognized style manual.
Acknowledgements

I would like to express my gratitude toward my advisor, Dr. Steven Day, for his mentorship and guidance throughout time in his lab. I would like to thank my other committee members, Dr. Kathleen Lamkin-Kennard and Dr. Thomas Gaborski for providing their support and expertise as well. I would also like to thank my fellow lab members: James Krisher, Matt Simonetty, Chance Kelly, Stacy Diaz, Saniya Attar, and James Rivera for their generous help and input throughout this thesis.

I would like to thank SiMPore and Jared Carter for providing CAD models and the microfluidic dialyzers that were tested in this thesis.

I would also like to thank the Bioengineering department staff members Renee Milliken and Michelle Horan as well as Rob Kraynik from Industrial & Systems Engineering department and Jan Manett from the Mechanical Engineering department. I would also like to thank The Construct @ RIT.

Finally, I would like to thank my family: my parents Frank and Debbie and my brother Michael. They provided endless support and encouragement from the very beginning of this project and I could not have completed this thesis without their help in troubleshooting experiments, providing feedback on my writing, and frequent phone calls.
Abstract

Microfluidics has become an increasingly popular tool in the design and development of medical devices and artificial organs. Two promising applications of microfluidics are dialyzers and oxygenators. As a step toward portable dialysis treatment, continuous microfluidic dialysis may resolve many clinical issues with current dialysis treatments. Additionally, commercially available oxygenators exceed the blood volume of neonatal patients; low-volume microfluidic devices may safely deliver oxygen to these patients. Two critical parameters in the development of these devices is mechanical hemolysis and membrane diffusion, which are intricately connected to the geometry, flow rate, properties of the membrane, and each other. A computational model is developed to elucidate the connection between these phenomena to guide the design and optimization of these devices. In vitro experiments are conducted to validate the model. Importantly, a subset of hemolysis models agrees with experimental data, which is consistent with the literature. Additionally, the effect of microfluidic mixing elements that perturb flow near the membrane interface are studied in silico and in vitro. These data reveal that herringbone mixing elements increase hemolysis by 10% and flux across the membrane interface by 38% in silico and a statistically significant difference between smooth and herringbone devices is observed for a subset of devices tested. Furthermore, 10 of 18 computational models of hemolysis are shown to be statistically similar to experimental data. The agreement of these results suggest that finite element analysis may be able to quantitively model important factors in the design of microfluidic oxygenators and dialyzers.
Table of Contents

Copyrights .............................................................................................................. i
Acknowledgements .................................................................................................. ii
Abstract .................................................................................................................... iii
Nomenclature ............................................................................................................. vi
1.0 Introduction ......................................................................................................... 1
  1.1 Flow in Microfluidic Devices ............................................................................. 1
  1.2 Membrane-Based Microfluidic Devices .............................................................. 2
  1.3 Quantifying Permeability .................................................................................. 4
  1.3 Staggered Herringbone Mixers ......................................................................... 6
  1.4 Blood and Hemolysis ....................................................................................... 7
  1.5 Quantitating Hemolysis ..................................................................................... 8
  1.6 Hemolysis Modeling .......................................................................................... 9
  1.7 Computational Fluid Dynamics ......................................................................... 11
  1.8 Blood Damage Computational Models ............................................................... 12
2.0 Motivation ............................................................................................................ 14
3.0 Objectives ........................................................................................................... 14
4.0 Methods and Materials ...................................................................................... 14
  4.1 Hemolysis Model ............................................................................................. 15
  4.2 Implementation of Hemolysis Models ............................................................... 18
  4.3 Membrane Transport Model ............................................................................ 21
  4.4 Model Verification ............................................................................................. 23
  4.5 Grid Size Study ................................................................................................ 24
  4.6 Hemolysis Experimentation ............................................................................. 26
  4.7 Flow Calibration ............................................................................................... 28
5.0 Results and Discussion ....................................................................................... 29
6.0 Conclusions ........................................................................................................ 38
  6.1 Relevance to Clinical Dialysis ......................................................................... 40
A – Appendices ....................................................................................................... 43
  MATLAB Code ....................................................................................................... 43
BD_1X3_100mL-min.c ................................................................. 47
Membrane Transport.c ............................................................ 64
References ......................................................................................... 68
Nomenclature

Re – Reynolds Number

$\rho$ – fluid density $[\text{kg/m}^3]$

$V$ = fluid velocity $[\text{m/s}]$

$\mu$ = fluid dynamic viscosity $[\text{kg/m s}]$

$V$ = velocity $[\text{m/s}]$

$D$ = diameter of pipe [m]

$D_0$ = original diffusivity $[\text{m}^2\text{s}^{-1}]$

$R_p$ = pore radius [m]

$N$ = Pore density $[\text{pores/m}^2]$

$\phi$ = porosity [-]

$P_m$ = membrane permeability $[\text{m/s}]$

$P_d$ = pore discovery permeability $[\text{m/s}]$

$P_t$ = transmembrane permeability $[\text{m/s}]$

$D_{t_m}$ = transmembrane diffusivity $[\text{m}^2\text{s}^{-1}]$

$D_m$ = complete membrane diffusivity $[\text{m}^2\text{s}^{-1}]$

$\xi$ = size ratio [-]

$d$ = membrane thickness [m]

$\Delta C$ = concentration difference between adjacent cells [mg/dL]
\[ fHb = free \ hemoglobin \ concentration \ \left[ \frac{mg}{dL} \right] \]

\[ Hb = Hemoglobin \ concentration \ \left[ \frac{mg}{dL} \right] \]

\[ J_i = flux \ across \ membrane \ for \ cell \ pair \ i, \ \left[ \frac{m^2 \times mg}{s \times dL} \right] \]
1.0 Introduction

1.1 Flow in Microfluidic Devices

Microfluidic devices are sub-millimeter scale devices that manipulate small volumes of fluid through electrical or mechanical forces. These devices are fabricated from glass, silica, polymers, or metals [1] and are commonly used in physical separations [2]–[5], cell manipulation [6], [7], and bioanalytics. Microfluidic channels typically have long, tortuous lengths that are several times greater than the dimensions of the cross section. Devices typically operate at flow rates in the $\mu$L/min range, leading to extremely low Reynolds Numbers ($Re < 1$). Reynolds number is defined as the ratio of inertial forces to viscous forces, and is defined in Equation (1-1).

$$Re = \frac{\rho V D}{\mu}$$

Together, the small channels and long lengths create unique flow characteristics in microfluidic channels. The no-slip condition, which states that the fluid velocity must equal zero at wall boundaries, has an exaggerated influence on the fluid velocity profile because of the short distance between walls, which create large velocity gradients and shear stresses in the device. Additionally, the high surface area to volume ratio in these devices allows for significantly higher heat and mass transfer rates and chemical reaction rates at interfaces than macroscale devices. This is often exploited in bioanalytical microfluidics to reduce the amount of reagent required and improve analysis time. Microfluidic flow is characteristically laminar due to the small hydraulic diameter, and therefore low Reynolds number, of microchannels. Poiseuille flow is the pressure-driven, viscous, laminar flow that develops in long channels of constant cross section, such as those of a microfluidic device. For a circular cross-section tube, this regime has
a parabolic velocity profile when fully developed and shear stress is proportional to the distance from the center axis of flow. Figure 1 shows the velocity and shear stress profiles of Poiseuille flow.

Figure 1: Fully-developed Poiseuille flow has a parabolic velocity profile and linear shear stress [8].

1.2 Membrane-Based Microfluidic Devices

This thesis focuses on microfluidic devices that contain blood-contacting, semipermeable membranes, such as oxygenators and hemodialyzers that deliver oxygen or remove wastes and toxins from the blood, respectively. Oxygenators, also known as lung assist devices, deliver oxygen to the blood of patients with respiratory failure. Hemodialyzers eliminate waste and toxins from the body, such as urea, creatinine, and phosphorus in patients with kidney failure [9].

The membranes function as a size-based separator in which the porosity, pore size, and thickness of the membrane govern the membrane permeability and affect the overall efficiency of the device. Membranes are commonly made of silicones [10], [11]; polymers [6], [7]; or silicon [2]–[4], [7], [12] materials. The membrane pores are large enough to permit the diffusion of small molecules across the membrane but too small to permit the leakage of blood across the membrane, which could lead to blood contamination and infection.
Historically, dialyzer membranes are classified as high- or low-flux, which describes the physical properties of the membrane that filters toxins from the blood [13], [14]. High-flux membranes have larger pores that allow for more effective filtration of small toxins (urea); however, they also remove desirable proteins (serum albumin) from the blood. Low-flux membranes can selectively filter toxins from the blood; however, the small pores inhibit diffusion through the membrane, resulting in longer dialysis treatments. The planar geometry of microfluidics serves as a platform for an emerging class of membranes that combines the high selectivity of low-flux membranes with the high permeability of high-flux membranes.

Microfluidic devices are also an important step toward portable dialyzers [12]; however, there are still many challenges that must be overcome before these devices could be used clinically. Continuous dialysis via portable dialyzers may resolve a critical issue with current dialysis treatment. Patients are typically treated 3-4 times per week; between treatments toxins accumulate in the body. Patient mortality has been shown to correlate with period of high toxin accumulation and is therefore undesirable [15]. Microfluidic oxygenators address a need for small-volume oxygenators for neonatal applications in which the volume of the oxygenator must be appropriately small due to the low blood volume of patients [16].

The characteristically small dimensions and sharp edges of a microfluidic device may induce nonphysiological shear stresses on the blood at certain flow rates. This may induce hemolysis, or damage and lysis of red blood cells (RBCs). Acute hemolysis, although rare in current dialysis treatments, can cause anemia, pain, headaches, tachycardia, and changes in blood pressure [17], [18]. Furthermore, consequences of damaging RBCs are exacerbated by the decreased erythropoietin production in dialysis patients, which controls the production of RBCs
[17], [19], [20]. While a clinical target for dialysis-induced hemolysis has not been set, it is important to minimize hemolysis, as the effects can aggregate with frequent treatment [17].

1.3 Quantifying Permeability

Particles that are much smaller than the pores will quickly diffuse through the membrane, whereas particles that are nearly the same size as the pores will diffuse more slowly. This diffusion resistance, or inverse of permeability, has been quantified by Snyder et al [4]. as the sum of the pore discovery and transmembrane permeability [4]. The pore discovery (Equation 1-2) permeability is the ability of a molecule to locate and diffuse into the entrance of a pore, whereas the transmembrane permeability is the steric and frictional resistance a molecule experiences as it travels through a membrane pore. The permeabilities can be conceptualized as two conductances in series, and each can be calculated as shown in Equation 1-3.

\[ P_d = 2D_0NR_p \]  \hspace{1cm} (1 - 2)

\[ P_m = \frac{1}{\frac{1}{P_d} + \frac{1}{P_t}} \]  \hspace{1cm} (1 - 3)

The pore density \( N \) (Equation 1-4) can be calculated based on the porosity and radius of the pore. Porosity is the amount of membrane area occupied by pores per total membrane area. For a uniform pore size and uniform porosity, the membrane area occupied by pores can be converted into the number of pores per total membrane area.

\[ N = \frac{\phi}{\pi R_p^2} (1 - 4) \]
Conceptually, the transmembrane permeability (Equation 1-5) is a function of the pore and molecule diameter. Molecules that are similar in size to the pore will experience greater steric resistance than molecules that are much smaller than the pores. The equation for transmembrane permeability is:

\[ P_t = \frac{N D_{t_m} \pi R_p^2}{d} \]  

(1 - 5)

\( D_{t_m} \) (Equation 1-6) is the transmembrane diffusion coefficient of the molecule through the membrane and is calculated using the cross-sectionally averaged hinderance [21]. It is based on the ratio of molecule to pore radius. \( \xi \) (Equation 1-7) is the ratio of molecule size to pore size. As the molecule size approaches the size of the pore, \( \xi \to 1, D_m \to 0 \), which physically represents great steric and frictional forces that prohibit diffusion through the membrane. When the size of the molecule is negligible compared to the size of the pore, \( \xi \to 0, D_m \to 1 \), which represents unhindered diffusion through the pore.

\[ D_{t_m} = D_0 \left( 1 + \frac{9}{8} \xi \ln \xi - 1.56 \xi + 0.53 \xi^2 + 1.92 \xi^3 - 2.82 \xi^4 + 0.27 \xi^5 + 1.10 \xi^6 - 0.44 \xi^7 \right) (1 - 6) \]

\[ \xi = \frac{R_s}{R_p} \]  

(1 - 7)

The membrane diffusivity \( D_m \) (Equation 1-8) is determined by considering the sum of the transmembrane and pore discovery permeabilities. The flux (Equation 1-9) through the follows the equations of Fick’s laws. In the discretized form of Fick’s first law, the flux is calculated through adjacent cells on opposite sides of the membrane in the computational model. The
membrane is mathematically represented through calculating the flux in each pair of cells for each timestep. The flux in a cell pair $i$ is calculated through Fick’s First Law.

$$D_m = P_m \times d \quad (1 - 8)$$

$$J_i = D_m \Delta C \quad (1 - 9)$$

The membrane diffusion model computes the concentration gradient between cells paired on opposite sides of the membrane interface. In this way, the flux through the membrane is localized to a specific cell pair, which better captures the spatial distribution of flux across the membrane compared to a lumped model, which averages the flux across a region or entire membrane surface.

1.3 **Staggered Herringbone Mixers**

Mixing in microfluidic systems is slow due to characteristically low Re in channels and lack of turbulent eddies to churn and mix flow. In membrane-based devices, near equal concentrations of solute can develop at the surface of the membrane, limiting diffusion across the membrane. Stirring flow near the membrane can disrupt the boundary layer and improve transport across the membrane. Passive mixers can improve mixing by introducing chaotic advection to improve homogenization without consuming power. These mixing elements were originally designed to mix two parallel streams for applications such as bioanalytics, cell separations, and reactors [22]. Tortuous path mixers are a type of passive mixer with a long, winding zigzag or serpentine path designed for the mixing of two parallel flows. The curves and sharp turns in the channels cause streamlines of the two flows to cross, which promotes mixing.
Another type of passive mixer is the staggered herringbone mixer. Herringbone mixers are chevron-shaped steps in the device that induce countercurrent vortices in the flow, as shown in Figure 2[11], [22]–[25]. The staggered herringbone mixer is a well-characterized mixing element that has been demonstrated to improve mixing for a wide range of flow rates (1 < Re < 100) [22], [24]. Two cycles of asymmetric chevrons induce different size vortices in the flow. The transition between asymmetric designs reverses the direction of the vortices near the center to further improve flow. Consequently, herringbones are an effective feature to promote diffusion in microfluidics. Herringbone mixers could improve mixing near the membrane interface to increase flux across the membrane.

1.4 Blood and Hemolysis

Blood consists of red blood cells (RBCs), white blood cells, and platelets suspended in a solution of plasma. RBCs contain hemoglobin, a molecule that binds to oxygen and carbon dioxide, that allows the cell to deliver oxygen throughout the body and return carbon dioxide to the lungs [19].
RBCs have a biconcave shape with a flattened center and viscoelastic cell membrane; this elongated shape and flexible membrane allow cells to deform to pass through small capillaries in the body, but also causes them to deform due to shear stress [26]. At rest, the cells assemble into a stacked structure called a rouleaux. At shear stresses above 0.1 [Pa], the stacks break down into individual cells which assume a biconcave shape (Figure 3). Shear stresses above 1 [Pa] deform RBCs into an ellipsoid and cause tank treading. Tank treading occurs when the cell membrane rotates independently of the contents of the cell. Greater shear stresses can open pores in the membrane and allow hemoglobin to leak into the plasma. The leakage of hemoglobin through these pores or the rupture of cells are known as hemolysis [26]. Although chemical, thermal, and osmotic stress may also induce hemolysis [18], mechanical stress due to non-physiologic flow may be a source of hemolysis in microfluidic hemodialyzers and oxygenators.

1.5 Quantitating Hemolysis

Hemoglobin that leaks out of cells remains in the plasma and is known as plasma free hemoglobin (fHb). Plasma that is physically separated from RBCs, such as through centrifugation, will retain the fHb, which dyes the plasma red. The intensity of the red color is related to the concentration of free hemoglobin and therefore the number of damaged cells. The
intensity of the redness can be measured optically using a spectrophotometer. Several methods have been developed to convert the absorbance at various wavelengths and weightings into the concentration of free hemoglobin [27].

The number of RBCs in the blood can affect the amount of hemoglobin released into the plasma. The change in concentration of free hemoglobin as a result of blood damage can be converted into the index of hemolysis (IH), which is normalized by the hematocrit, or percentage of RBCs in the blood [28], [29]. IH% (Equation 1-10) is expressed as a percentage from 0-100%, where 100% IH corresponds complete rupture of all RBCs.

\[
IH = \frac{\Delta fHb(1 - Ht)}{Hb} \times 100 \tag{1 - 10}
\]

1.6 Hemolysis Modeling

While hemolysis can also be induced by chemical, thermal, and radiation stresses [18], mechanical shear is the biggest contribution to hemolysis in microfluidic hemodialyzers and oxygenators. The characteristically high spatial velocity gradients and sharp edges due to the planar substrate generate non-physiologic shear stresses that induce blood damage. Over the past several decades, many groups have attempted to mathematically predict blood damage.

Giersiepen et al. was one of the first to publish a generalized power-law equation (Equation 1-11) that expresses blood damage as a function of the shear stress applied to the blood and exposure time, or the duration for which the stress is applied [20], [30]. Regression analysis of experimental data is applied to determine the constants of \( C, \alpha, \) and \( \beta \).

\[
D = \frac{\Delta fHb}{fHb} = C\tau^\alpha t^\beta \tag{1 - 11}
\]
Many experiments have been conducted to estimate the parameters of this equation. In these experiments, the experimental setup was designed to closely mimic the turbulent, rotational flow of a ventricular assist device. Shear stress was applied to blood using a Couette-flow shearing device consisting of a concentric rotor and housing connected through mechanical bearings. The rotor was tapered on both ends with a narrow passage in the center, known as the shear gap. The rotor induced shear stresses proportional to the speed of the rotor. The flow rate through the device determined the exposure time; higher flow rates yielded lower exposure times. This allows both the applied shear stress and exposure time to be independently and precisely controlled throughout the experiment. Data collected from this device was used to determine the constants of the power-law equation through regression analysis. Many groups have estimated the parameters of this model using experimental data from couette-flow shearing experiments [8], [28], [29], [31]–[33]. Differences in experimental design has resulted in parameters that yield vastly different hemolysis predictions. Of all models, three published parameter sets have been extensively used in hemolysis prediction models: the coefficients of Giesiepen (Equation 1-12), Heuser and Opitz (Equation 1-13), and Zhang (Equation 1-14) [34]. The respective models are listed below.

\[
D_{GW} = 3.63 \times 10^{-7} \tau^{2.416} t^{0.785} \quad (1 - 12)
\]

\[
D_{HO} = 1.8 \times 10^{-8} \tau^{1.991} t^{0.765} \quad (1 - 13)
\]

\[
D_{ZT} = 1.228 \times 10^{-7} \tau^{1.9918} t^{0.6606} \quad (1 - 14)
\]

The discrepancy between groups can be attributed to several factors. In Couette-flow shearing devices, it is assumed that blood damage occurs only in the shearing gap, which has a controlled
shear stress and exposure time. Secondary damage sources such as damage due to mechanical bearings have been cited as additional sources of blood damage [28], [31], and may contribute to the overestimation of blood damage with the Giersiepen constants compared to other models [20], [28]. Blood preparation, source, and species may also influence blood damage by introducing biological variability. Although a universally accepted hemolysis does not exist yet, the three established damage models are all used in the assessment of blood-contacting devices to develop a range of hemolytic potential in the context of ventricular assist devices [34]–[36].

1.7 Computational Fluid Dynamics

Computational fluid dynamics (CFD) is a method of numerical analysis to solve fluid flow problems. The results provide spatial and temporal information about the velocity, stress, and pressure fields. Modern CFD packages allow the user to embed user-defined equations to be solved concurrently with the flow equations; this allows the generic power-law equation to be applied to complex, 3-dimensional flow, such blood flow through a ventricular assist device to predict hemolysis [26], [35]–[37].

In vitro blood-testing can be expensive, time consuming, and inconsistent, which can slow the development of blood-contacting devices [36], [38]. Simulations eliminate variation between donor species, collection method, and handling methods. Supplementing experimentation with numerical modeling allows the designer to better optimize the product by performing more design iterations between experiments. In addition, the spatial and temporal simulation results highlight hemolytic areas of the device, which allows the designer to focus on these areas. Therefore, it has been of great interest to develop an accurate computational blood damage model.
1.8 **Blood Damage Computational Models**

Computational modeling of hemolysis is commonly used to guide the design of ventricular assist devices (VADs) and other artificial organs [28], [30], [34]–[37], [39]–[42]. While *in vitro* testing of VADs is important to assess hemolysis within the device, computational modeling can supplement experimental testing to further improve device design. Testing with animal blood can be an expensive and time-consuming. Additionally, intra- and interspecies biological variability in blood fragility can mask improvements in device design. A validated numerical model can predict the differences between device designs with a high amount of precision. Importantly, a spatiotemporal model can determine areas of the device which induce the greatest amount of hemolysis, unlike experimental testing in which regional hemolysis cannot be easily determined. This allows the engineer to target specific areas of the design for improvement.

A wide range of blood damage models exist that vary in complexity and predictiveness [34]–[36]. Blood damage is commonly modeled as a function of shear stress \( \tau_s \) and exposure time \( t \) with parameters fitted to experimental data, commonly known as the power-law model [24], [40]. This type of model loosely models the viscoelastic cell membrane deformation that permits the leakage of hemoglobin out of the cell. This model can be integrated directly into the framework of a computational fluid dynamics solver, minimizing model development and verification time.

Several other damage models have been developed with varying levels of complexity and computational cost. The majority of the models are based on the power-law equation and are implemented in either Lagrangian or Eulerian approaches. Lagrangian methods follow the path of an individual particle, whereas Eulerian methods follow a volume of fluid. Lagrangian models range in complexity from simple pathline integrals to RBC membrane strain models [26],
Pathline models follow the trajectory of massless particles, typically from the inlet to the outlet of the device. After the flow field is solved, stress is computed at discrete points along the path of the particle and damage is calculated.

While the pathline method is the simplest way to calculate blood damage, it has several issues that may affect its accuracy. Pathlines may not pass through the entire domain, so areas of high shear stress, such as sharp corners, or recirculations may be ignored in the hemolysis calculation [34]. Additionally, the number of pathlines and starting location may affect the calculation. While it is possible to account for these additional factors, it significantly increases the complexity of calculating hemolysis. Two models have been developed and are commonly used in hemolysis prediction for their simplicity and agreement with experimental data: the power-law (PL) (Equation 1-15) and time-history models (TH) (Equation 1-16).

\[
D(\tau, t) = \left[ \frac{1}{C^\beta \tau^\alpha t} \right]^{\beta} \tag{1-15}
\]

\[
D(\tau, t) = \sum_{i=\text{inlet}}^{\text{outlet}} C\beta \left[ \sum_{j=1}^{i} \tau(t_j)^\alpha \Delta t_j + D_b(t_0) \right]^{\beta-1} \tau(t_i)^\alpha \Delta t_i \tag{1-16}
\]

The time history model differs from the power law model in that it considers an RBC’s “history” of hemolysis; RBCs that have previously experienced shear stress will release less hemoglobin upon subsequent exposure to the same shear, which better mimics physiologic observations.
2.0 Motivation
The use of microfluidics is growing increasingly popular in the development of medical devices and artificial organs. Among the applications of this technology is the improvement of dialyzers and oxygenators, which bear a membrane that filters or oxygenates the blood respectively. These devices may address several clinical needs as discussed earlier.

3.0 Objectives
The purpose of this thesis was to develop a computational model of two key parameters in the design of dialyzers/oxygenators: hemolysis induced by the flow of blood through the device and diffusion of molecules across the membrane.

Five distinct devices were simulated in silico and in vitro to explore the model’s predictiveness for a variety of device configurations. All of the devices have an inlet and outlet manifold that transitions from the circular port to the channels at the membrane interface. Information about the properties of each device can be found in Table 1. The computational and experimental results are compared. Furthermore, two devices are equipped with herringbone mixers to explore their impact on hemolysis and membrane flux.

4.0 Methods and Materials
A computational model of hemolysis and membrane flux within a microfluidic dialyzer/oxygenator was developed in the ANSYS Fluent computational fluid dynamics package. The governing equations of hemolysis and membrane flux were converted to the built-in User Defined Function architecture to implement source/sink and convective terms. The fluid domain was extracted from CAD models of five prototype devices (Table 1). Flow was simulated along a computational grid at a rate of 100 [mL/min]. The simulation was validated against in vitro experiments of recirculating flow in a prototype device. The five device configurations are highlighted in Table X.
Table 1: Properties of the devices tested in silico and in vitro.

<table>
<thead>
<tr>
<th>Device ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Herringbone Mixers</strong></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong># of Channels</strong></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td><strong>Channel Width (mm)</strong></td>
<td>1.66</td>
<td>1.66</td>
<td>1.66</td>
<td>1.66</td>
<td>1.66</td>
</tr>
<tr>
<td><strong>Channel Height (mm)</strong></td>
<td>2.8</td>
<td>2.8</td>
<td>2.8</td>
<td>2.8</td>
<td>0.25</td>
</tr>
<tr>
<td><strong>Port Size (mm)</strong></td>
<td>1</td>
<td>1</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Figure 4: (Left) Simulation setup for the hemolysis model Only the blood-side of the device is modeled. (Right) Simulation setup for the membrane transport model.

4.1 Hemolysis Model

A breadth of hemolysis models were found in the literature that ranged in complexity from simple post-processing of the shear stress field to resolving the deformation of the red blood cell membrane under specific shear stresses [34]. Two blood damage models were selected from the literature based on their minimal complexity and relatively high agreement with experimental results in the literature: the power-law and time-history models. In these models, RBCs are assumed to be uniformly distributed within each computational cell, as the computational cells are much larger (10-fold) than individual RBCs. All RBCs in a
computational cell are assumed to experience the same shear stresses. For both hemolysis models, the hemoglobin released by RBCs within a computational cell is based on the scalar shear stress at the center of the computational cell. The hemoglobin released by cells is convected to the outlet and measured as the mass-averaged concentration. The time history model differs from the power law model in that it considers an RBC’s “history” of hemolysis; RBCs that have previously experienced shear stress will release less hemoglobin upon subsequent exposure to the same shear, which better mimics physiologic observations. Within each class of hemolysis models, three distinct parameter sets and three methods of computing the scalar shear stress were selected from literature for a total of eighteen damage models. The equations and parameters for each model are summarized in Table S2. All models were solved as second-order upwind or third-order when applicable. This achieved greater convergence order \((p)\) than that of a first-order solution, thus decreasing the grid-induced numerical error. This model pairs directly with the built-in flow solver in Fluent. After the computational domain was prepared, the flow solver calculates the flow variables (pressure, velocity) at each timestep. Concurrently, the shear stress tensor in each cell serves as an input to the hemolysis model, which converts the stress tensor into a scalar shear stress.

Hemolysis was modeled as the leakage of free hemoglobin into the computational domain. Because computational model is agnostic to hematocrit, or the percentage of red blood cells of a volume of blood, the hemolysis ratio is calculated and convected through the domain. The blood damage per pass through the device is calculated as the average hemolysis ratio at the device outlet. The resultant hemolysis ratio (Equation 4-1) calculated in the simulation must be converted to IH\% (Equation 4-2) to compare the simulation and experiment results, where \(Ht\) is the hematocrit of the experimental data.
\[
\text{Hemolysis Ratio} = \frac{\Delta fHb}{Hb} \quad (4 - 1)
\]

\[
IH\% = (\text{Hemolysis Ratio})(1 - Ht) \times 100 \quad (4 - 2)
\]

Figure 5 describes the simulation workflow and how the flow solver interfaces with the membrane transport and hemolysis models.

---

**Figure 5: Flow chart of the hemolysis and membrane transport models.**
4.2 Implementation of Hemolysis Models

The hemolysis equations are converted from their original forms to that required by the UDF. The UDF requires the hemolysis to be expressed as a source term in the transport equations for an arbitrary quantity in Fluent (Equation 4-3). The transport equation is solved for each computational cell in the domain.

\[
\frac{\partial}{\partial x_i} \left( \rho u_i \phi_k - \Gamma \frac{\partial \phi_k}{\partial x_i} \right) = S_{\phi_k}, k = 1, 2, 3
\]  

(4 - 3)

The two terms on the left side of the equation represent the advective and diffusive terms. The source term \( S_{\phi_k} \), defined by the UDF, and is used to describe the hemolysis (generation of free hemoglobin). The power-law and time-history models must be converted to this form. Importantly, the models must be transformed such that the time variable is raised to the first power (Equation 4-4).

\[
D(\tau, t) = \left[ \frac{1}{C^\beta \tau^\beta} t \right]^\beta
\]  

(4 - 4)

This form can be subsequently implemented in the transport equation as a source term (Equation 4-5). The density term is added to the source term to satisfy resolve the proper units for the source term.

\[
\frac{\partial}{\partial x_i} \left( \rho u_i H - \Gamma \frac{\partial H}{\partial x_i} \right) = S_{PL} = C^\beta \tau^\beta \times \rho
\]  

(4 - 5)
Due to the complexity of the time-history model, it must be separated into two equations (Equations 4-6 and 4-7), and therefore two transported species, are required to implement into the UDF. The first transport equation is solved across the domain at each time step. The resulting value at each computational cell serves as an input to the second equation, which represents the convected hemoglobin as a result of hemolysis.

\[
\frac{\partial}{\partial x_i} \left( \rho Db - \Gamma \frac{\partial Db}{\partial x_i} \right) = S_{TH_1} = \tau^\alpha \times \rho \quad (4 - 6)
\]

\[
\frac{\partial}{\partial x_i} \left( \rho u_i H - \Gamma \frac{\partial H}{\partial x_i} \right) = S_{TH_2} = C\beta \left(Db^\beta - 1\tau^\alpha\right) \rho \quad (4 - 7)
\]

Importantly, this implementation of the time-history model does not require exponentiating by $\beta$ as a post-processing step. In this way, these equations are then implemented into Fluent as a User Defined Scalar, which handles the convective terms. The eighteen damage models and the corresponding parameters are summarized in Table 2.
Table 2: Properties of each hemolysis model tested in silico.

A parabolic velocity profile is specified at the inlet to avoid hemolysis induced by boundary conditions. Figure 4 shows the simulation setup for the hemolysis model. The
hemoglobin that leaked from cells was measured as the average hemolysis ratio at the outlet. This quantity represents the amount of hemolysis accumulated in a single pass through the device. The simulation boundary conditions are described in Figure 4.

4.3 Membrane Transport Model

The diffusion of a molecule through a porous medium was modeled based on the molecule-specific permeability through the membrane; larger molecules experience greater resistance to passing through pores than smaller molecules due to steric interactions with the membrane. The diffusivity of urea is calculated as a function of the properties of the molecule and membrane, such as the diameter of the molecule and pore, thickness of the membrane, and porosity.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>15</td>
<td>nm</td>
<td>[15]</td>
</tr>
<tr>
<td>R_s</td>
<td>0.36</td>
<td>nm</td>
<td>[44]</td>
</tr>
<tr>
<td>R_p</td>
<td>5</td>
<td>nm</td>
<td>[15]</td>
</tr>
<tr>
<td>D_0</td>
<td>1.16E-09</td>
<td>m^2/s</td>
<td>[43]</td>
</tr>
<tr>
<td>φ</td>
<td>0.05</td>
<td>-</td>
<td>[15]</td>
</tr>
<tr>
<td>D_m</td>
<td>2.90E-11</td>
<td>m^2/s</td>
<td>[15]</td>
</tr>
</tbody>
</table>

Table 3: Properties of the molecule (urea) and membrane tested in the validation step[15], [43], [44].

The diffusivity follows Fick’s laws of diffusion. Urea was selected as a physiologically relevant molecule, as it is the main target in dialysis. The properties of urea and the membrane used in this work are listed in Table S3. The flux through an adjacent cell pair \(i\) at the membrane is described by Fick’s First Law for each timestep, where \(D_m\) represents the calculated membrane diffusivity and \(\Delta C\) is the concentration gradient across the paired cells.
\[ J_i = D_m \Delta C \]  \quad (4 - 8)

The membrane diffusion model computes the concentration gradient between cells paired on opposite sides of the membrane interface. In this way, the flux through the membrane is localized to a specific cell pair, which better captures the spatial distribution of flux across the membrane compared to a lumped model, which averages the flux across a region or entire membrane surface. The membrane UDF is only applied to cells adjacent to the membrane. All cells, including those adjacent to the membrane allow the flow of urea through diffusion and advection to all neighboring cells. The concentration of urea is normalized to a value of 0 to 1, where 1 corresponds to the maximum concentration of urea that is present at the inlet of the blood-side of the device. The concentration of urea is computed as the mass-average of the outlet concentration. The change in concentration from the inlet to outlet is then calculated.

*Figure 6: Verification the power-law model by comparing to a MATLAB numerical solution in the sinusoidal shear stress case.*
To verify that the equations (Equations 4-5, 4-6, and 4-7) were correctly implemented in the UDF and solver, numerical solutions were reproduced from the literature and compared to analytical solutions when applicable. To this end, the hemolysis model was validated in two scenarios: static shear and sinusoidal shear stress. The analytical solution to the power-law model was used to validate the power-law model in the constant shear stress case. No analytical solution exists for the other model, so a MATLAB code developed and verified against the literature. This MATLAB numerical solution was used to verify the computational model of hemolysis. The results of this verification are shown in Figure 6 and Figure 7. The membrane model is verified by reproducing numerical results found in the literature. Conditions described in Burgin et al.[15] were recreated and shown to sufficiently agree with the published experimental and analytical solution as shown in Figure 8.

Figure 7: Verification the time-history model by comparing to a MATLAB numerical solution in the sinusoidal shear stress case.
Figure 8: Verification and validation of the membrane transport model with data published in the literature (Burgin et al.)

4.5 Grid Size Study

For each simulation case, a grid convergence study was performed using Richardson Extrapolation to assess mesh discretization error. This error arises from the estimation of a continuum using finite elements. As the size of mesh elements decreases, it is expected that the discretization error will approach zero and the solution will asymptotically converge toward the “exact” solution. While this “exact” value is not known, the error from this value can be estimated by obtaining results for three mesh sizes via Richardson extrapolation. This error can be used to assess whether the mesh size may be significantly influencing the results.

For each simulation condition, the solution was resolved for three mesh sizes: 80, 113, and 160 μm elements (fine, medium, and coarse respectively). Mesh refinement regions proportional to the base mesh size were specified for areas with high velocity gradients to improve solution accuracy and convergence. For all simulations except the high shear device, the high convergence order suggests that grid-induced error is sufficiently low (<5% average for all models). The error in the high shear device could not be estimated because the solution does not
asymptotically approach a value, indicating that the grid is not sufficiently refined. In this case, the “exact” solution cannot be estimated; however, the error was minimized by solving these simulations at the highest resolution permitted by the resources available.

Figure 9: (Left) Grid study results for the Smooth (1.0mm) devices. (Right) Estimated error from the “exact” solution determined by Richardson Extrapolation.

Figure 10: Results for the Restricted device. The results do not asymptotically approach the "exact" solution, so the error cannot be determined.
4.6 Hemolysis Experimentation

An experimental system to measure hemolysis in the prototype devices was developed as shown in Figure 11. Whole bovine calf blood with ACD anticoagulant was acquired through venipuncture (Lampire Biological Laboratories, PA) from distinct animals, each considered separate biological replicates.

Blood is drawn and shipped in a refrigerated container and used within 12 hours of receipt. Blood is filtered through a 250 micron pore mesh (McMaster Carr) to remove any thrombus formation. The circuits are constructed of polycarbonate fittings (VWR) and silicone tubing (MasterFlex). Flow through the circuit was driven by a peristaltic pump (MasterFlex 7518) with multiple drive heads attached to permit parallel testing of multiple devices with the same blood donor. The pump speed is calibrated to 100 mL/min to match simulations. The circuits are filled with 1X PBS and circulated for 20 minutes following ASTM F1841-97 [45]. The PBS is removed and 75 mL blood is immediately infused into the circuits via syringe pump.

Figure 11: (Left) schematic of the blood circuit in used in experimentation and multiple circuits connected in parallel with multiple peristaltic pump heads. (Right) Devices fabricated and used in experimentation.
Figure 12: (Left) Samples collected are placed in microcentrifuge tubes for processing. (Right) Isolated plasma is plated in 96-well plates to measure the absorbance.

Two replicate samples are aspirated from the circuit every 30 minutes for 10-12 hours. These samples are centrifuged at 3000xg for 10 minutes to separate plasma from RBCs. The plasma is separated and centrifuged at 13000xg for 10 minutes to remove any remaining cells. The resulting plasma is plated into a 96-well plate and the absorbance is measured at 560, 576, and 593 nm wavelengths (Figure 12). Hemolysis is determined from the absorbance measurements according to the Cripps method. The index of hemolysis is computed by accounting for the hematocrit (Equation 4-9).

\[
IH\% = \frac{\Delta fHB}{fHB}(1 - Hct) \times 100
\]  

(4 - 9)

The hemolysis per pass is calculated to directly compare the experimental data to simulation results. Regression analysis is performed on the curve of IH% vs. number of passes to determine this value. The number of passes during a time interval \(i\) is based on the space velocity in the system (Equation 4-10).

\[
Passes_i = \frac{Flow \ Rate}{Circuit \ Volume}
\]  

(4 - 10)
Since the circuit volume changes significantly throughout the experiment due to sampling (75 mL to 12.5 mL), the change in volume after each sampling is considered. Hemolysis induced by the device was separated from other damage sources (peristaltic pump, autohemolysis) by testing a “control” blood circuit lacking a microfluidic device simultaneously with the other circuits. Thus, the device-induced hemolysis is the difference between the device loop hemolysis per pass and the control circuit hemolysis per pass. Model agreement was assessed using Student’s one-sample T test was used to identify any cases in which the experimental and simulation results were not statistically different.

4.7 Flow Calibration

The flow rate of the experimental setup must match the 100 mL/min selected for the computational model. A flow meter was first calibrated against a syringe pump because it is more accurate than the error of the flow meter according to its published specifications. The flow meter was then used to measure the flow rate of each individual pump head for a variety of pump head speeds. 110 RPM was selected to yield, on average, 100 mL/min. The true flow rate was determined from this data and used to in the conversion of sample measurement time to passes. The flow rate curves for each of the pumps is shown in Figure 13.
5.0 Results and Discussion
To determine whether a computational model can predict hemolysis and membrane in an arbitrary membrane-based device, eighteen variations of the hemolysis model were simulated to determine which computational models were statistically similar (p<0.05) to experimental data. Five devices were simulated and *in vitro* hemolysis testing was conducted with whole bovine blood (N=4).
Figure 14: Example computational model outputs. (Left) contour slices through the smooth device. Hemolysis is notably greater at the outlet compared to the inlet. (Right) Simulation results of the membrane model. Diffusion occurs at the membrane interface.

In Figure 14, an example of the solution outputs is shown. Information about the spatial distribution of hemoglobin can be observed through contour plot cross-sections through the domain. Contour plots of the concentration of urea at the membrane interface are also shown.

Figure 15 shows the results of the experiment for a single device. The red bar in the bottom left corner shows the number of passes through a dialyzer that occur during clinical hemodialysis.

Figure 15: Experimental results of the IH% measured at points throughout the experiment. Regression analysis is performed to extract the hemolysis per pass through the device.
Figure 16: The damage per pass of the experimental data is compared to the simulation results.

The two classes of models: the power-law and time-history models both predicted similar hemolysis results for each parameter set (Figure 16). This is expected because the time-history effect should be small since the magnitude of hemolysis predicted is also low. The models predicted hemolysis values across several orders of magnitude, which is consistent with the literature. Furthermore, models that differ only by the scalar shear stress calculation (1-3, 4-6, 7-9) predict similar levels of hemolysis, suggesting that the prediction is less sensitive to the scalar shear stress model compared to the other model parameters ($C, \tau, t$). Furthermore, comparing the hemolysis results between devices reveals that some models consistently predict higher or lower hemolysis than others for a variety of devices geometries. Importantly, comparing the prediction between two devices reveals that all models predict either greater or less hemolysis.

![Figure 16: The damage per pass of the experimental data is compared to the simulation results.](image)

Table 4: Student’s T tests are performed for each replicate for each device (listed in cells). Values highlighted in green indicate no significant difference between experiment and simulation hemolysis (95% confidence interval).
This observation is consistent with the literature and is potentially an artifact of secondary damage sources in Giersiepen model calibration dataset. The Heuser and Optiz coefficients (PL-7, PL-9, TH-7, TH-9) agreed most frequently (Table 4), which is consistent with the literature. For the Smooth (Large Ports) device, no simulations agree with experimental data. This may be attributed to the negative value of hemolysis per pass for this device from the experimentation. For this device, the control damage was consistently greater than the device damage, thus yielding a negative hemolysis value. This is likely due to variation in the magnitude of secondary hemolysis (peristaltic pump, autohemolysis) and low hemolysis induced by the device. This low signal-to-noise ratio may explain the discrepancy in the data.

The Student’s T test only gives information about whether the experimental and simulation are statistically similar based on the signal-to-noise ratio of the data. Since there was a high amount of biological variability, it is possible that the data may be statistically similar but the models may vastly overpredict or underpredict hemolysis compared to the experimental data. In Table 5, the percent difference in hemolysis magnitude is shown for the models that are statistically similar to the experimental data (p<0.05). Positive percent difference indicates that the simulation overpredicts compared to the experimental data and negative values indicate that the model underpredicts hemolysis. Interestingly, some models (PL-4, TL-4) nearly match the experimental data, whereas other models overpredict (PL-7, PL-9, TH-7, TH-9) or underpredict (PL-5, TH-5). Interestingly, the hemolysis models for the Smooth High Shear device overpredict by an order of magnitude; however, the experimental and simulation data are still statistically similar. This is due to the high biological variability in the Smooth High Shear devices.
Table 5: Percent Difference between simulation and experimental hemolysis. Positive values indicate the simulation overpredicts hemolysis, whereas negative values indicate the simulation underpredicts hemolysis.

<table>
<thead>
<tr>
<th>Device</th>
<th>PL-1</th>
<th>PL-2</th>
<th>PL-3</th>
<th>PL-4</th>
<th>PL-5</th>
<th>PL-6</th>
<th>PL-7</th>
<th>PL-8</th>
<th>PL-9</th>
<th>TH-1</th>
<th>TH-2</th>
<th>TH-3</th>
<th>TH-4</th>
<th>TH-5</th>
<th>TH-6</th>
<th>TH-7</th>
<th>TH-8</th>
<th>TH-9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Herringbone (Large Ports)</td>
<td>7%</td>
<td>-50%</td>
<td>48%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smooth High Shear</td>
<td>105%</td>
<td>133%</td>
<td>94%</td>
<td>94%</td>
<td>120%</td>
<td>620%</td>
<td>150%</td>
<td>375%</td>
<td>820%</td>
<td>1200%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Next, the effect of staggered herringbone mixers on the hemolysis and membrane flux was explored. The difference in predicted hemolysis between the smooth and herringbone devices was quantitated by comparing the average difference across all models.

Interestingly, in the devices with smaller ports, hemolysis in the herringbone device is predicted to be lower than the smooth device. However, the opposite prediction is observed in the devices with larger ports; hemolysis is predicted to be 10% greater in the herringbone device compared to the smooth device. The greater hemolysis in the 1.6mm device may be due to the increased shear stress in the channels induced by the folding of streamlines by the herringbones.
Figure 17: Comparison of in silico hemolysis results for the PL-1 model. The 1.0mm devices and high shear devices are shown to induce 10-fold more hemolysis than the baseline model.

Additionally, the sharp edges of the herringbones may also induce greater shear in the device. In the devices with smaller ports, the majority of the hemolysis is induced by the high shear stress in the inlet and outlet. This is supported by the nearly 20-fold increase in hemolysis when comparing a 1.6mm port to a 1.0mm port device (Figure 17). Since the majority of the hemolysis in the 1.0mm devices is caused by the inlet and outlet ports, which are identical in the smooth and herringbone devices, the slightly lesser prediction may be a numerical artifact and may not represent any effect caused by the herringbones.
Figure 18: Comparison of experimental hemolysis for both the small- and large-port devices. Statistical significance is determined for $p<0.05$.

Next the *in vitro* results for the small- and large-port devices were compared (Figure 18). Based on the simulation results, the 1.6mm devices should induce less hemolysis than the 1.0mm devices because less hemolysis is induced at the inlet and outlet ports. There is no statistical difference between the smooth and herringbone 1.0mm port devices. In some cases, the circuit with only tubing (no devices) exceeded the hemolysis of circuits with devices at all time points. This may be due to biological variability and variability in the experimental system. Thus, when the device damage is calculated by subtracting the tubing damage, the result is a negative number, which is physically impossible. In all cases, the circuits with the Smooth 1.6mm devices had lesser damage than the tubing. A Student’s T test was performed to determine whether the hemolysis per pass of the tubing and Smooth 1.6mm device circuits was statistically different (Figure 20). Since the data was not statistically different, the Smooth 1.6mm device damage was subtracted from all other circuits instead of the tubing damage to ensure that the device-only damage is positive.
Figure 19: Comparison of the raw hemolysis per pass of the tubing-only circuit and the circuit with the Smooth 1.6mm. devices (p<0.05).

Figure 20: Comparison of the in silico urea transport results.

The difference in membrane flux was quantified by comparing the change in concentration of urea from the inlet to the outlet of the blood-side of the device (Figure 20). A higher change in concentration indicates that diffusion across the membrane is greater. The herringbone devices
achieved a greater change in urea concentration as a result of greater membrane flux at the membrane interface (18% to 25%).

![Streamlines through the smooth device center channels. (Right) Streamlines through the herringbone device channels. The chaotic streamlines suggest the flow is perturbed by the herringbones.](image)

Figure 21: (Left) Streamlines through the smooth device center channels. (Right) Streamlines through the herringbone device channels. The chaotic streamlines suggest the flow is perturbed by the herringbones.

This difference is attributed to the folding of streamlines induced by the herringbones. To explore this theory further, streamlines were examined in the membrane channels. As shown in Figure 21, the streamlines in the smooth device are all nearly straight and normal to the membrane channel. The streamlines in the herringbone device appear chaotic and deviate from the straight path observed in the smooth devices. This observation is further evidence that the herringbone mixers disturb the boundary layer near the membrane to increase diffusion across the membrane.

The benefit of the herringbone may be improved in devices with a lesser channel height. The herringbones perturb flow nearest the herringbone structures and less streamline folding and vorticity is noted further away from the herringbones. Decreasing the channel height may increase the effectiveness of the herringbones by their ability to disturb the boundary layer and restore the concentration gradient.
6.0 Conclusions

A computational model of the hemolysis and membrane transport was developed and validated with in vitro testing in this work. These two parameters were selected because they are critical to the development of microfluidic dialyzers and oxygenators, which show promise in resolving challenges in their respective areas. The model quantitatively agreed with results published in literature and qualitative observations in the accuracy of certain hemolysis models in the context of VAD development. Specifically, the Giersiepen parameters have been cited as significantly overpredicting hemolysis and the Heuser/Opitz parameters have shown the best agreement with experimental data. Agreement with these literature findings is impressive given the context in which these models were developed. The experiments to define the parameters of each model were designed to mimic the turbulent, rotational flow of a VAD as closely as possible. The ability of these models to predict hemolysis in laminar flow through a microfluidic device suggests that these models could predict hemolysis in other types of blood-contacting devices. Furthermore, while only three models consistently agreed with experimental data, all models showed the same trend in blood damage over- or under-prediction, suggesting that although some models may not predict the magnitude of hemolysis induced by the device, any model could predict the trend in blood damage between device iterations, making the computational model still beneficial in device design.

The experiments also revealed an interesting characteristic of the tested devices. Although these devices contain sharp edges and small features that induce non-physiological shear stresses at the tested flow rates, these stresses are insufficient to induce significant hemolysis in the devices. A relatively low amount of hemolysis was induced by the device.
compared to the peristaltic pump. While this results in a low signal-to-noise ratio in experimentation, the low hemolysis is promising for the future development of microfluidic blood-contacting devices where hemolytic potential is a consideration. Furthermore, it provides additional evidence that the hemolysis models capture key mechanisms of hemolysis that are independent of the flow regime.

The implementation of herringbones near the membrane interface increased the change in concentration from the inlet of the device to the outlet (18% to 25%) at the tested flow rate in silico. These results show promise in the implementation of mixing elements to improve diffusion across the membrane in these applications, increasing the effectiveness of the device. Improving diffusion across the membrane may also decrease the required flow rate to achieve a target absorption/oxygenation rate. This may subsequently decrease shear stresses and therefore hemolysis in a dialyzer/oxygenator. The proposed herringbone geometry in this work was adapted from the literature. Optimization of the herringbone features and channel geometry may further improve the benefit of the herringbone mixers. The herringbone features have been shown in silico to increase hemolysis by up to 10%; however, no statistical significance was shown experimentally. Further testing is necessary to conclude whether a difference in hemolysis is measurable; however, optimization of the herringbone features may also minimize this effect.

This work demonstrates a tool that may be used in the development of microfluidic dialyzers and oxygenators. Three device designs and five distinct devices were examined with eighteen hemolysis models. This work suggests that computational modeling can predict blood damage in these devices, which may be paramount in device design. Additional experimentation is necessary to determine the parameter space in which the model may predict accurately. To this
end, multiple flow rates and a range of device geometries may be tested to further examine this predictive range. The benefits of modeling have been observed in VAD design, in which computational modeling plays a significant role in the iterative development of blood pumps. While optimization of these devices is not explored in this work, it is a powerful tool that may drive the optimization and clinical adoption of microfluidic dialyzers and oxygenators.

6.1 Relevance to Clinical Dialysis

The experimental conditions (flow rate: 100 [mL/min], reservoir volume: 75 [mL], experiment length: 10-12 hours) were specified to maximize the hemolysis signal from the devices and minimize secondary hemolysis due to the tubing and peristaltic pump. These conditions generate 1000+ passes through the devices by the end of the experiment to achieve ~1% hemolysis. Comparatively, clinical dialysis occurs at vastly different conditions. Hemodialysis occurs at flow rates of (300+ [mL/min])[18] and at patient blood volumes of several liters. Furthermore, hemodialysis treatment is typically 3-4 hours in length [18]. Thus, the number of passes through the device in these experiments greatly exceed that of a clinical hemodialysis treatment (Figure 22). In this way, the experiments can be interpreted as “fatigue testing” of the microfluidic devices in which the number of passes at the end of an experiment may be of similar order to the total passes of a patient’s blood through the device throughout the many treatments they receive in a lifetime. Because free hemoglobin is filtered by the body and RBCs are constantly replaced, the amount of hemolysis after the lifetime of the device may not be representative of actual patient hemolysis levels. Nevertheless, this “lifetime” hemolysis can be used to communicate to clinicians whether a device may cause dangerously hemolytic. In this case, the total hemolysis induced by the smooth device is near 1%, which is less than the 2-5% at which symptoms of toxicity occur [18].
Figure 22: Comparison of the conditions in these experiments to an in vivo experiment with similar conditions to clinical hemodialysis.

Interestingly, the contrast in the hemolysis per pass between the experimental data and those found in the literature (Sakota et al.) highlights the need to consider other source of blood damage when considering hemolysis at the patient level. These experiments focused on the hemolysis induced by the dialysis device. Sakota et al. studied hemodialysis-induced hemolysis by measuring hemolysis in a blood circuit at clinically-relevant flow rates and volumes. Their experimental setup also includes cannulas that connect the patient’s blood vessels. These cannulas have studied experimentally and computationally as another source of hemolysis in hemodialysis [17], [20], [30], [34], [46]. The diameter and flow rate determine the shear stresses and thus hemolysis induced by the cannula. Multiple cannulas (double cannula hemodialysis) may be used [18] to reduce the flow rate through each cannula to induce less hemolysis overall. Fundamentally, hemolysis is caused by the high flow rates required to remove toxins from the blood in a relatively short treatment time (4 hours). Longer treatments at lesser flow rates may
reduce this effect. This is ultimately the motivation for continuous, portable hemodialysis: lower flow rates may be afforded by continuously filtering the blood at a lesser rate compared to filtering several days of toxin accumulation in the span of a few hours.
A – Appendices

MATLAB Code

%CrippsMethodCalculationModified.m

%Aauthored by: James Krisher, 2018

%This document assumes the use of a 96-well plate at 576,560, and 593nm
%wavelengths for use in calculation of Cripps Method of Hemolysis.
%**XML file generated from use of SpectraMax iD3 Multi-Mode Microplate
%Reader in basement lab must be converted to excel.** One excel file
%per reading. It is also assumed that wells H11 and H12 are PBS
%controls.

%To use this file, simply enter the file path (line 14) and the names of
%each file to be read separated by semicolons under filenamearr
%(line 17). The processed hemolysis data will be exported as a
%spreadsheet in the same file directory along with .png image files of
%each plate as read.'

%Modified by Matthew Poskus, 2019
%Modified by Matthew Simonetty, 2019
%Must have BioInformatics Toolbox (v4.11 for this file)

clc; clear;
load microplateassay
colormap(whiteToRed)
filepath = '/Volumes/GoogleDrive/My Drive/School/M. Poskus Thesis/Raw Experiment Data/8-13-19/';

cd(filepath)

filenamearr = {'Poskus_8-13-19_Plate5_R1.xlsx';
               'Poskus_8-13-19_Plate5_R2.xlsx';
               'Poskus_8-13-19_Plate5_R3.xlsx';};

crippmethK = 177.6;

%%Read by Read Cripps Method
findat = zeros(96,length(filenamearr)+2);
header = {'Index','Cell Label'};
diluteArr = zeros(3,8,3);

for r = 1:length(filenamearr)
    rawdat = zeros(24,12); adjdat = zeros(24,12); hemdat = zeros(8,12);
    background = zeros(24,12);
    filename = char(filenamearr(r));
    fullpath = strcat(filepath,filename);
    xlsread(fullpath,1,'C52:N59','basic');
    xlsread(fullpath,1,'C62:N69','basic');
    xlsread(fullpath,1,'C72:N79','basic');
end
c=0;
for a=1:8
    for b=1:12
        rawdat(a+c,b) = rawdat1(a,b);
        rawdat(a+c+1,b) = rawdat2(a,b);
        rawdat(a+c+2,b) = rawdat3(a,b);
    end
    c=c+2;
end

% Adjusted data takes raw data and subtracts out absorbance value of PBS background samples
q=1;
while q < 25
    background(q,1:12) = (rawdat(22,11)+rawdat(22,12))/2;
    background(q+1,1:12) = (rawdat(23,11)+rawdat(23,12))/2;
    background(q+2,1:12) = (rawdat(24,11)+rawdat(24,12))/2;
    q=q+3;
end
adjdat = rawdat - background;

%DiluteArr is an array that indicates which wells must be diluted in order for the Cripp’s Method to work properly
for i=1:8
    for j=1:12

if adjdat(3*i-2,j)>2 || adjdat(3*i-1,j)>2 || adjdat(3*i,j)>2
    diluteArr(i,j,r) = 1;
end

hemdat(i,j) = crippmethK*...
    (adjdat(3*i-2,j)-(adjdat(3*i-1,j)+adjdat(3*i,j))/2);
end
end

figure(1)

microplateplot(hemdat)
title(filename(1:end-5), 'Interpreter', 'none')

fignam = strcat(filename(1:end-5),'_PlateDiagram.png');
saveas(figure(1),fignam)

findat(:,r+2) = reshape(transpose(hemdat),[],1);
header(r+2) = {filename(1:end-5)};
end

findat(:,1) = 1:96;
labels = {'A1';'A2';'A3';'A4';'A5';'A6';'A7';'A8';'A9';'A10';'A11';'A12'; ...
    'B1';'B2';'B3';'B4';'B5';'B6';'B7';'B8';'B9';'B10';'B11';'B12'; ...
    'C1';'C2';'C3';'C4';'C5';'C6';'C7';'C8';'C9';'C10';'C11';'C12'; ...
    'D1';'D2';'D3';'D4';'D5';'D6';'D7';'D8';'D9';'D10';'D11';'D12'; ...
    'E1';'E2';'E3';'E4';'E5';'E6';'E7';'E8';'E9';'E10';'E11';'E12'; ...
    'F1';'F2';'F3';'F4';'F5';'F6';'F7';'F8';'F9';'F10';'F11';'F12'; ...
    'G1';'G2';'G3';'G4';'G5';'G6';'G7';'G8';'G9';'G10';'G11';'G12'; ...
    'H1';'H2';'H3';'H4';'H5';'H6';'H7';'H8';'H9';'H10';'H11';'H12';};
% Average and standard deviation of the three readings are calculated for each well

findatavg=zeros(size(findat,1),1);
findatstd=zeros(size(findat,1),1);
findatnum=zeros(size(findat,1),3);
for d=1:size(findat,1)
    findatnum(d,1)=findat(d,3);
    findatnum(d,2)=findat(d,4);
    findatnum(d,3)=findat(d,5);
    findatavg=mean(findatnum,2);
    findatstd=std(findatnum,0,2);
end
findat=[findat,findatavg,findatstd];
header=[header,'Average of three readings','Standard deviation'];

findatcell = num2cell(findat);
findatcell(:,2) = labels;
findatcell = cell2table([header; findatcell]);
exportnam = strcat(filename(1:end-15),'_ProcessedHemolysis_Plate5.xlsx');
writetable(findatcell,exportnam)

BD_1X3_100mL-min.c

#include "udf.h"
#include<stdio.h>
#include<stdlib.h>
#define C_GW 0.000000363
#define alpha_GW 2.416
#define beta_GW 0.785

#define C_HO 0.000000018
#define alpha_HO 1.991
#define beta_HO 0.765

#define C_ZT 0.0000001228
#define alpha_ZT 1.9918
#define beta_ZT 0.6606

#define FlowRate_mL_Min 100.
#define R 0.0005

DEFINE_PROFILE(inlet_velocity, thread, position)
{
    real vec[ND_ND];
    real y, z, r, Rsq, volumetricFlowRate, Vavg;
    face_t f;

    volumetricFlowRate = FlowRate_mL_Min/1000/1000/60; /* Convert from mL/min to m3/s */
    Vavg = volumetricFlowRate/(M_PI*pow(R,2));
begin_f_loop(f,thread);
{
    F_CENTROID(vec, f, thread);
    y = vec[1];
    z = vec[2];
    r = pow(pow(y,2)+pow(z,2),0.5);
    F_PROFILE(f, thread, position) = 2*Vavg*(1-pow(r,2)/pow(R,2));
}
end_f_loop(f,thread)

DEFINE_ADJUST(BloodDamage,d)
{
    Thread *c_t;
    cell_t c;
    real maxShearRate, freq, t, mu;
    real second_invariant;
    real tau11, tau12, tau13, tau21, tau22, tau23, tau31, tau32, tau33;
    real tau_vm, tau_p, tau_b;
    mu = 0.0035;
    thread_loop_c(c_t, d)
    {
        begin_c_loop(c,c_t)
\[
\begin{align*}
\tau_{11} &= 2(C_{\text{DUDX}}(c,c_t))\mu; \\
\tau_{12} &= (C_{\text{DUDY}}(c,c_t)+C_{\text{DVDX}}(c,c_t))\mu; \\
\tau_{13} &= (C_{\text{DUDZ}}(c,c_t)+C_{\text{DWDX}}(c,c_t))\mu; \\
\tau_{21} &= \tau_{12}; \\
\tau_{22} &= 2(C_{\text{DVDY}}(c,c_t))\mu; \\
\tau_{23} &= (C_{\text{DVDZ}}(c,c_t)+C_{\text{DWDY}}(c,c_t))\mu; \\
\tau_{31} &= \tau_{13}; \\
\tau_{32} &= \tau_{23}; \\
\tau_{33} &= 2C_{\text{DWDZ}}(c,c_t)\mu; \\
\text{second_invariant} &= \text{fabs}((\tau_{11}+\tau_{22}+\tau_{33})(\tau_{11}+\tau_{22}+\tau_{33}) - (\tau_{11}\tau_{11} + \tau_{22}\tau_{22} + \tau_{33}\tau_{33} + 2\tau_{12}\tau_{21} + 2\tau_{23}\tau_{32} + 2\tau_{13}\tau_{31}))/2; \\
\tau_{\text{vm}} &= \sqrt{3\text{second_invariant}}; \\
\tau_{\text{p}} &= \sqrt{\text{second_invariant}}; \\
\tau_{\text{b}} &= \sqrt{\text{fabs}((\tau_{11}-\tau_{22})(\tau_{11}-\tau_{22}) + (\tau_{11}-\tau_{33})(\tau_{11}-\tau_{33}) + (\tau_{22}-\tau_{33})(\tau_{22}-\tau_{33}) + \tau_{12}\tau_{12} + \tau_{13}\tau_{13} + \tau_{21}\tau_{21} + \tau_{23}\tau_{23} + \tau_{31}\tau_{31} + \tau_{32}\tau_{32})}; \\
C_{\text{UDMI}}(c,c_t,0) &= \tau_{\text{vm}}; \\
C_{\text{UDMI}}(c,c_t,1) &= \tau_{\text{p}}; \\
C_{\text{UDMI}}(c,c_t,2) &= \tau_{\text{b}}; \\
\text{end_c_loop}(c,c_t)
\end{align*}
\]
DEFINE_SOURCE(A_GW_vm,c,t,dS,eqn)
{
    real C_Mod, Tau_mod, source;
    C_Mod = pow(C_GW,1/beta_GW);
    Tau_mod = pow(C_UDMI(c,t,0),alpha_GW/beta_GW);
    source = C_Mod*Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(A_GW_p,c,t,dS,eqn)
{
    real C_Mod, Tau_mod, source;
    C_Mod = pow(C_GW,1/beta_GW);
    Tau_mod = pow(C_UDMI(c,t,1),alpha_GW/beta_GW);
    source = C_Mod*Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(A_GW_b,c,t,dS,eqn)
{
    real C_Mod, Tau_mod, source;
    C_Mod = pow(C_GW,1/beta_GW);
\[
\text{ Tau\_mod} = \text{pow}(\text{C\_UDMI}(c,t,2),\text{alpha\_GW}/\text{beta\_GW});
\]

\[
\text{source} = \text{C\_Mod}\times\text{Tau\_mod}\times\text{C\_R}(c,t);
\]

\[
dS[\text{eqn}] = 0;
\]

\[
\text{return source};
\]

} 

DEFINE\_SOURCE(A\_HO\_vm,c,t,dS,eqn) 

{ 

\text{real C\_Mod, Tau\_mod, source;}

\text{C\_Mod} = \text{pow}(\text{C\_HO},1/\text{beta\_HO});

\text{Tau\_mod} = \text{pow}(\text{C\_UDMI}(c,t,0),\text{alpha\_HO}/\text{beta\_HO});

\text{source} = \text{C\_Mod}\times\text{Tau\_mod}\times\text{C\_R}(c,t);

\text{dS[eqn]} = 0;

\text{return source;}

} 

DEFINE\_SOURCE(A\_HO\_p,c,t,dS,eqn) 

{ 

\text{real C\_Mod, Tau\_mod, source;}

\text{C\_Mod} = \text{pow}(\text{C\_HO},1/\text{beta\_HO});

\text{Tau\_mod} = \text{pow}(\text{C\_UDMI}(c,t,1),\text{alpha\_HO}/\text{beta\_HO});

\text{source} = \text{C\_Mod}\times\text{Tau\_mod}\times\text{C\_R}(c,t);

\text{dS[eqn]} = 0;

\text{return source;}

}
DEFINE_SOURCE(A_HO_b,c,t,dS,eqn)
{
    real C_Mod, Tau_mod, source;
    C_Mod = pow(C_HO,1/beta_HO);
    Tau_mod = pow(C_UDMI(c,t,2),alpha_HO/beta_HO);
    source = C_Mod*Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(A_ZT_vm,c,t,dS,eqn)
{
    real C_Mod, Tau_mod, source;
    C_Mod = pow(C_ZT,1/beta_ZT);
    Tau_mod = pow(C_UDMI(c,t,0),alpha_ZT/beta_ZT);
    source = C_Mod*Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(A_ZT_p,c,t,dS,eqn)
{
    real C_Mod, Tau_mod, source;
    C_Mod = pow(C_ZT,1/beta_ZT);
    Tau_mod = pow(C_UDMI(c,t,1),alpha_ZT/beta_ZT);
source = C_Mod*tau_mod*C_R(c,t);

dS[eqn] = 0;

return source;

}

DEFINE_SOURCE(A_ZT_b,c,t,dS,eqn)
{
real C_Mod, tau_mod, source;

C_Mod = pow(C_ZT,1/beta_ZT);

tau_mod = pow(C_UDMI(c,t,2),alpha_ZT/beta_ZT);

source = C_Mod*tau_mod*C_R(c,t);

dS[eqn] = 0;

return source;

}

/* Model C Equations */

/* UDS 9*/

DEFINE_SOURCE(C_GW_vm_db,c,t,dS,eqn)
{
real tau_mod, source;

tau_mod = pow(C_UDMI(c,t,0),alpha_GW/beta_GW);

source = tau_mod*C_R(c,t);

dS[eqn] = 0;
return source;
}

DEFINE_SOURCE(C_GW_vm,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    Db = C_UDSI(c,t,9);
    Tau_mod = pow(C_UDMI(c,t,0),alpha_GW/beta_GW);
    if (Db <=0)
    {
        source = 0;
    }
    else
    {
        source = C_GW*beta_GW*pow(Db,beta_GW-1)*Tau_mod*C_R(c,t);
    }
    dS[eqn] = 0;
    return source;
}

/* UDS 11*/

DEFINE_SOURCE(C_GW_p_db,c,t,dS,eqn)
{
    real Tau_mod, source;
    Tau_mod = pow(C_UDMI(c,t,1),alpha_GW/beta_GW);

source = Tau_mod*C_R(c,t);
dS[eqn] = 0;
return source;
}

DEFINE_SOURCE(C_GW_p,c,t,dS,eqn)
{
real Db, Tau_mod, source;
Db = C_UDSI(c,t,11);
Tau_mod = pow(C_UDMI(c,t,1),alpha_GW/beta_GW);
if (Db <=0)
{
    source = 0;
}
else
{
    source = C_GW*beta_GW*pow(Db,beta_GW-1)*Tau_mod*C_R(c,t);
}
dS[eqn] = 0;
return source;
}

/* UDS 13*/
DEFINE_SOURCE(C_GW_b_db,c,t,dS,eqn)
{

real Tau_mod, source;

Tau_mod = pow(C_UDMI(c,t,2),alpha_GW/beta_GW);

source = Tau_mod*C_R(c,t);

dS[eqn] = 0;

return source;

}

DEFINE_SOURCE(C_GW_b,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    
    Db = C_UDSI(c,t,13);
    
    Tau_mod = pow(C_UDMI(c,t,2),alpha_GW/beta_GW);
        
    if (Db <=0)
        
        {

            source = 0;
        
        }
    
    else
    
    {

            source = C_GW*beta_GW*pow(Db,beta_GW-1)*Tau_mod*C_R(c,t);

    }

    dS[eqn] = 0;

    return source;

}

/* USD 15 */

DEFINE_SOURCE(C_HO_vm_db,c,t,dS,eqn)
{ 
real Tau_mod, source;

Tau_mod = pow(C_UDMI(c,t,0),alpha_HO/beta_HO);

source = Tau_mod*C_R(c,t);

dS[eqn] = 0;

return source;
}

DEFINE_SOURCE(C_HO_vm,c,t,dS,eqn)
{
real Db, Tau_mod, source;

Db = C_UDSI(c,t,15);

Tau_mod = pow(C_UDMI(c,t,0),alpha_HO/beta_HO);

if (Db <=0)
{

source = 0;

}
else
{

source = C_HO*beta_HO*pow(Db,beta_HO-1)*Tau_mod*C_R(c,t);

}

dS[eqn] = 0;

return source;
}
/* UDS 17 */

DEFINE_SOURCE(C_HO_p_db,c,t,dS,eqn)
{
    real Tau_mod, source;
    Tau_mod = pow(C_UDMI(c,t,1),alpha_HO/beta_HO);
    source = Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(C_HO_p,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    Db = C_UDSI(c,t,17);
    Tau_mod = pow(C_UDMI(c,t,1),alpha_HO/beta_HO);
    if (Db <=0)
    {
        source = 0;
    }
    else
    {
        source = C_HO*beta_HO*pow(Db,beta_HO-1)*Tau_mod*C_R(c,t);
    }
    dS[eqn] = 0;
    return source;
/* UDS 19 */

DEFINE_SOURCE(C_HO_b_db,c,t,dS,eqn)
{
    real Tau_mod, source;
    Tau_mod = pow(C_UDMI(c,t,2),alpha_HO/beta_HO);
    source = Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(C_HO_b,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    Db = C_UDSI(c,t,19);
    Tau_mod = pow(C_UDMI(c,t,2),alpha_HO/beta_HO);
    if (Db <=0)
    {
        source = 0;
    }
    else
    {
        source = C_HO*beta_HO*pow(Db,beta_HO-1)*Tau_mod*C_R(c,t);
    }
}
\[ dS[\text{eqn}] = 0; \]

return source;

} /* USD 21 */

DEFINE_SOURCE(C_ZT_vm_db,c,t,dS,eqn)
{
    real Tau_mod, source;
    Tau_mod = pow(C_UDMI(c,t,0),alpha_ZT/beta_ZT);
    source = Tau_mod*C_R(c,t);
    dS[\text{eqn}] = 0;
    return source;
}

DEFINE_SOURCE(C_ZT_vm,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    Db = C_UDSI(c,t,21);
    Tau_mod = pow(C_UDMI(c,t,0),alpha_ZT/beta_ZT);
    if (Db \leq 0)
    {
        source = 0;
    }
    else
    {
        source = C_ZT*beta_ZT*pow(Db,beta_ZT-1)*Tau_mod*C_R(c,t);
    }
}
dS[eqn] = 0;
return source;
}

/* UDS 23 */
DEFINE_SOURCE(C_ZT_p_db,c,t,dS,eqn)
{
    real Tau_mod, source;
    Tau_mod = pow(C_UDMI(c,t,1),alpha_ZT/beta_ZT);
    source = Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(C_ZT_p,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    Db = C_UDSI(c,t,23);
    Tau_mod = pow(C_UDMI(c,t,1),alpha_ZT/beta_ZT);
    if (Db <=0)
    {
        source = 0;
    }
    else
{ 
    source = C_ZT*beta_ZT*pow(Db,beta_ZT-1)*Tau_mod*C_R(c,t);
}
dS[eqn] = 0;
return source;
}

/* UDS 25*/
DEFINE_SOURCE(C_ZT_b_db,c,t,dS,eqn)
{
    real Tau_mod, source;
    Tau_mod = pow(C_UDMI(c,t,2),alpha_ZT/beta_ZT);
    source = Tau_mod*C_R(c,t);
    dS[eqn] = 0;
    return source;
}

DEFINE_SOURCE(C_ZT_b,c,t,dS,eqn)
{
    real Db, Tau_mod, source;
    Db = C_UDSI(c,t,25);
    Tau_mod = pow(C_UDMI(c,t,2),alpha_ZT/beta_ZT);
    if (Db <=0)
    {
        source = 0;
    }
}
else
{
    source = C_ZT*beta_ZT*pow(Db,beta_ZT-1)*Tau_mod*C_R(c,t);
}
dS[eqn] = 0;
return source;
}

Membrane Transport.c

#include "udf.h"
#include<stdio.h>
#include<stdlib.h>

#define ZONE_ID 5

DEFINEPROFILE(inlet_velocity, thread, position)
{
    real vec[ND_ND];
    real y, z, r, R, Rsq, volumetricFlowRate, Vavg;
    face_t f;

    R = 0.0005;

    volumetricFlowRate = 100./1000/1000/60; /* Convert from mL/min to m3/s */
    Vavg = volumetricFlowRate/(M_PI*pow(R,2));
begin_f_loop(f, thread);
{
    F_CENTROID(vec, f, thread);
    y = vec[1];
    z = vec[2];
    r = pow(pow(y, 2) + pow(z, 2), 0.5);
    FPROFILE(f, thread, position) = 2*Vavg*(1-pow(r, 2)/pow(R, 2));
}
end_f_loop(f, thread)
}

DEFINE_ADJUST(transport, d)
{
    cell_t c0, c1 = -1;
    face_t f;
    Thread *f_thread;
    Thread *t0, *t1 = NULL;
    real CellConc0, CellConc1;
    real NV_VEC(A0);
    real NV_VEC(A1);
    real Resistance;
    Resistance = 517.;

    f_thread = Lookup_Thread(d, ZONE_ID);
    begin_f_loop(f, f_thread)
    {

t0 = THREAD_T0(f_thread);
t1 = THREAD_T1(f_thread);
c0 = F_C0(f,f_thread);
c1 = F_C1(f,f_thread);
F_AREA(A0,f,f_thread);

CellConc0 = C_UDSI(c0,t0,0);
CellConc1 = C_UDSI(c1,t1,0);

C_UDMI(c0,t0,0) = (CellConc1 - CellConc0)*C_R(c0,t0)*NV_MAG(A0)/C_VOLUME(c0,t0)/Resistance;
C_UDMI(c0,t0,1) = -C_R(c0,t0)*NV_MAG(A0)/C_VOLUME(c0,t0)/Resistance;
C_UDMI(c1,t1,2) = (CellConc0 - CellConc1)*C_R(c1,t1)*NV_MAG(A0)/C_VOLUME(c1,t1)/Resistance;
C_UDMI(c1,t1,3) = -C_R(c1,t1)*NV_MAG(A0)/C_VOLUME(c1,t1)/Resistance;

end_f_loop(f,f_thread)
}

DEFINE_SOURCE(membrane_adjacent,c,t,dS,eqn)
{
    real source;
    source = C_UDMI(c,t,0);
    dS[eqn] = C_UDMI(c,t,1);
}
DEFINE_SOURCE(membrane_shadow_adjacent,c,t,dS,eqn)
{
    real source;
    source = C_UDMI(c,t,2);
    dS[eqn] = C_UDMI(c,t,3);
    return source;
}

return source;
References


[16] M. Dabaghi et al., “An artificial placenta type microfluidic blood oxygenator with double-
sided gas transfer microchannels and its integration as a neonatal lung assist device,”

*Biomicrofluidics*, vol. 12, no. 4, p. 44101, 2018.


