5-28-2019

Computational Feasibility of Simulating Whole-Organ Vascular Networks and Their Response to Injury

William Patrick Donahue
Louisiana State University and Agricultural and Mechanical College

COMPUTATIONAL FEASIBILITY OF SIMULATING WHOLE-ORGAN VASCULAR NETWORKS AND THEIR RESPONSE TO INJURY

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Physics & Astronomy

by
William Patrick Donahue
B.S. in Physics, University of Connecticut, 2012
August 2019

Dedicated to

Meagan Victoria Cairns

for her support and patience

Acknowledgments

I thank my dissertation supervisory committee (Wayne Newhauser, Joyoni Dey, Juana Moreno, Harris Wong, Xin Li, and Vincent Wilson) for their guidance and commitment to my dissertation research. In particular, I thank Dr. Newhauser for his mentorship and support during my time at Louisiana State University. Working together over the past 7 years has taught me many important aspects of being a successful researcher. I thank Dr. Wilson for motivating me to study more biology than I ever planned on in my lifetime.

I thank the Louisiana State University High-Performance Computing Center for their excellent staff and resources. Without their support in terms of computation time, classes, and discussions, this project would not have been possible. I specifically thank Wei Feinstein and Feng Chen for their comments and discussions, I learned a great deal from them.

To my fellow students, thank you for your efforts and friendship during my doctoral studies. I want to specifically acknowledge the fellow members of my research group (Lydia Wilson, Chris Schneider, and Suman Shrestha) for their participation, feedback, and support during this long process. I also want to thank Paul Maggi and Andrew Halloran for insightful discussions about research and general topics.

I thank the organizations and individuals who provided funding to support this research project. This included The Bella Bowman Foundation, The United States Nuclear Regulatory Commission (NRC; award NRT-HQ-84-15-G-0017), and The United States Naval Academy (Contract No N00189-13-P-0786). The story of Bella Bowman inspired this line of inquiry.

Finally, I thank my parents for their support of my education and life pursuits. Without their help, none of this would have been possible.

Table of Contents

Abstract

Healthy vasculature is critical to sustaining the function of normal tissues in the human body. Radiation therapy for cancer causes injury of the vasculature of non-cancerous tissues. These changes have been associated with potentially deadly conditions such as necrosis of the brain tissue. There are currently no computational methods to study the effects of radiation vascular injury in whole-organ vasculatures because of the large number of vessels involved. The goal of this work was to test the feasibility of simulating radiation damage to whole-brain vascular networks and calculate the resulting change in blood flow. To accomplish this, we developed algorithms to create a fractal-like geometry with 17 billion vessels, simulate the radiation dose to those vessels, and calculate the resulting change in blood flow. Computational performance metrics were measured for each algorithm individually and as a complete pipeline to determine the computational feasibility. Using a modular system containing these algorithms, we demonstrated that it is computationally feasible to predict the effects of radiation on blood flow in whole-organ vasculature. The system required 90 hours to perform the simulation for 2 million protons incident on an 8.5 billion vessel network using 128 compute nodes. Furthermore, the dose calculations were determined to be the most time consuming part of this system. The vessel-geometry algorithm and blood-flow algorithm both demonstrated the ability to reach 17 billion vessels. With future improvements, whole-organ simulations of vascular injury have the potential to elucidate the importance of vasculature to the development of radiation late effects.

1. Introduction

1.1. Context

It is estimated that in 2019 approximately 1.8 million people will be diagnosed with cancer in the United States [1]. Improvements in cancer treatment have improved the 5-year survival rate from 50% in 1975 to 69% in 2019 for the US population [1]. Because people are living longer after a cancer diagnosis, it is increasing important to consider the detrimental long-term effects of their treatment [2]. Radiation therapy is a widely-used treatment modality due to its ability to non-invasively and accurately target a tumor [3]. The long-term side effects of radiation therapy include second cancers [4], fibrosis [5], and white-matter necrosis [6] to name just a few. Necrosis in the human brain can significantly impact the quality of life and possibly cause death in up to 25% of patients who receive heavy ion therapy[6].

Necrosis is defined as a swelling of cells followed by a catastrophic rupturing of the cell membrane, releasing inflammatory factors into the tissue [7, 8]. These inflammatory factors can cause other cells to enter necrotic-death pathways, leading to the formation of a large necrotic region in tissue. Necrosis can be triggered by ischemia, hypoxia, and other effects that disrupt cellular equilibrium [9]. There are two hypotheses regarding the initiating cause of necrosis in the brain. The first is the death of glial cells that support the function of neurons [10]. This causes a cascade of cell death that leads to the formation of a necrotic lesion. The second hypothesis is radiation-induced vascular injury leads to the formation of necrosis, involving regions of ischemia, hypoxia, and changes in the blood brain barrier. [10, 11]. These hypotheses can, in principle, be tested in part by considering the physics of the delivery of oxygen and nutrient delivery by the vasculature. Hence, simulations of radiation-induced injury on blood flow could be used to probe the physical determinants of radiation necrosis.

1.2. State of Knowledge

The human brain contains up 9 billion blood vessels in the densest vascular network in the human body [12, 13]. Many of these blood vessels are capillaries, which are responsible for transferring oxygen and nutrients to the glial cells and neurons. It has been demonstrated that accurately modeling the blood flow in the cortex depends on accurately modeling the blood flow in the capillaries [14]. The average brain capillary has a length of 55 µm and the smallest radius is 2.5 µm [12]. In a first principles approach to simulating the effects of vascular injury from radiation on the blood flow in the brain, it is necessary to model each vessel, the dose it receives, and the flow of blood through it.

The modeling of vascular geometry has progressed during recent decades. One method is to reconstruct vascular geometry from high resolution imaging techniques [15–18]. These techniques are typically limited to small volumes and destroy the sample in most cases. Recent advances have enabled researchers to construct realistic geometries using computational methods. One widely-used computational technique is constrained constructive optimization (CCO) [19]. CCO uses a stochastic approach to fill the target volume with vasculature, while optimizing the vascular density, sheer stress, and blood flow [20]. Linninger *et al.* [21] coupled CCO with a Voronoi mesh technique to computationally create a 27 mm^3 segment of the human brain. This small volume contained only 256,000 vessels but accurately reproduced the morphometric measurements of the brain. Another technique is the application of fractal methods to create vascular trees [22–24]. These models have seen extensive use in the modeling of arterial trees for organs [22, 25]. The natural simplicity of a fractal enables an efficient computation of vascular geometry, however it is limited in it is ability to produce a realistic, 3-dimensional geometry [26]. The largest vascular geometry previously constructed was limited to 360,000 vessels [27], 10,000 times fewer than in the human brain.

Radiation transport to small targets, such as the capillaries, necessitates calculating the interactions of each particle track with each individual vessel. Monte Carlo methods simulate the deposition of energy in a geometry by using probabilistic methods guided by

interaction cross-sections [28]. These techniques have been used to great effect for simulating the radiation chemistry of DNA damage [29], treatment planning [30, 31], and whole-body dose reconstructions [32, 33]. Monte Carlo techniques, while accurate, are computationally expensive, especially as the level of detail increases. This prevents the simulation of large volumes with many small targets, *e.g.* the vascular network of a human brain. One way to reduce the computations is to not transport the individual secondary particles, *e.g.*, delta rays, that are created by the primary radiation. An alternative is to aggregate the transport of delta rays using amorphous-track structure models [34–36]. These models replace the transport of the individual secondary particles with an analytical approximation of the mean dose deposited by all the delta rays from a track. This reduces the computational complexity and expense. Track structure models are commonly used to model biological responses, *e.g.*, they are using in treatment planning systems for heavy ion therapies [35].

The biological modeling of radiation damage is typically limited to cellular responses and mechanisms [37]. Using closed form equations, the dose is converted to a cell survival fraction [37]. The cell survival fraction has been used to model the clinical response of normal tissues and tumors due to radiotherapy treatments [38, 39]. The study of radiation's effect on vasculature using computational methods has been limited. The most comprehensive work on vascular network changes due to radiation studied rabbit ear vasculature [40, 41]. These studies provided details on the relative change in vessel length, radius, surface area, and volume versus the dose deposited, but was limited to population averages and assumed a uniform dose. More recent studies have focused on the effects of radiation on vascular function and correlate the damage with changes is tissue function [42–45]. All of the previous studies on the impacts of radiation on vasculature have ignored the effects that changes in vessel structure and function have on blood flow through an organ.

The simulation of blood flow is a mature field. The most advanced and accurate technique for simulating blood flow through vessels is finite-element modeling. In this approach, a vascular network is broken into a finite number of small volumes. The Naiver-Stokes equa-

tions are applied to each element, creating a system of partial differential equations that can be solved to determine the blood flow [46]. Finite-element models have been used to simulate the blood flow for research on stents [47–49], strokes [50, 51], and aneurysms [52, 53]. Each of these simulations is computationally complex and has been limited in the number of vessels that can be simulated, *i.e.*, less than 50 vessels [54, 55]. Simplifying the blood flow theory can increase the maximum number of vessels modeled in a simulation, but this comes at the cost of reduced physical accuracy [21, 56, 57]. Linninger *et al.* [21] successfully used a steady-state approach to simulate the blood flow through 256,000 vessels and quantify the local tissue oxygenation in a 27 mm^3 section of human brain. This is the largest study of blood flow in individually modeled vessels to date.

To summarize, simulations of vasculature have been limited to small volumes with high-levels of detail or large volumes with less vascular details. However, the radiation delivered during a therapeutic treatment has the potential to damage blood vessels through out the human brain. These changes in individual vessels, while small, can collective impact the blood flow in a significant way. Currently, there are no no proven methods that can construct a geometric description of, calculate the dose to, or simulate the blood flow through a vascular network containing 9 billion individual vessels. It is currently unknown if it is feasible to perform each of these tasks using current high-performance computing resources.

1.3. Objective

The objective of this work was to test the feasibility of simulating radiation damage to whole-brain vascular networks and calculate the resulting change in blood flow. To accomplish this, we first developed and evaluated an algorithm to create a reproducible vascular geometry with billions of individual vessels (Chapter 2). Next, we tested the feasibility of calculating blood-flow rates using a steady-state approach (Chapter 3) and evaluated the accuracy of the model (Chapter 4). To facilitate the dose calculations, we developed an analytical formula for stopping powers of heavy ions spanning five orders of magnitude in kinetic energy (Chapter 5). Finally, we developed a recursive dose algorithm to score the

dose from protons to individual blood vessels and tested the feasibility of simulating the radiation-induced changes in blood flow for whole-brain vasculatures (Chapter 6).

2. A Simple Computational Model of the Vasculature of the Whole Human Brain

2.1. Introduction

The brain controls and coordinates the cognitive and critical motor processes of all vertebrates. In humans it is the second largest internal organ, with a volume of 1.4 L [58]. The human brains contains approximately 200 billion cells, over half of which are neurons [59]. The neurons play roles in decision making, memory, and effecting action. The cells that support the neurons can be broadly classified as endothelial cells (of the vasculature) and glial cells. Glial cells provide necessary support functions to neurons, including neurotransmitter uptake and reprocessing [60] and maintenance of the extracellular environment [61]. Both neurons and glial cells require energy and oxygen, which are supplied by blood that is transported through the vasculature. It is estimated that up to 9 billion blood vessels supply the human brain, in the densest vascular network of the human body [12, 62]. Although vasculature has been extensively studied [63–66], its relationship to disease and injury is not completely understood [67–71]. Vascular injury has been correlated with many major illnesses and conditions, including Alzheimer's Disease [72], traumatic brain injury [73], dementia [74], and radiation-induced complications, such as white-matter necrosis [11] and fibrosis [5].

The literature is replete with studies on the vasculature of the central nervous system (CNS) and the gross anatomy of the brain. These are well understood, primarily through pathologic and morphometric studies, such as those performed by Duvernoy *et al.* [75] and Gray and Lewis [76], whose works mapped the positions of larger blood vessels in the brain. Additionally, we know about the structure of the larger vessels; *e.g.*, arteries have walls that are thicker and lined with more muscle than veins [76]. High resolution imaging and histopathology have elucidated physical aspects of microcirculatory anatomy, *e.g.*, the density of the vascular network [12, 13, 62, 77, 78]. An extensive literature describes the func-

tions of vasculature in normal tissues [79–82] and damaged states [83–87]. Most knowledge concerning vasculature comes directly from observational studies and experiments.

Relatively less attention has been paid in the literature to computationally modeling vascular geometry. Much progress has been made in recent decades toward creating models of vessels and small vascular networks; we briefly review several examples here. One method is to extract vascular geometries from high-resolution images. Imaging capillaries requires a technique with a spatial resolution greater than their diameter (*i.e.*, 4 µm) [17]. Such techniques include confocal and optical microscopy [15, 16], synchrotron radiation micro-computed tomography [17], and knife-edge scanning microscopy [18]. These techniques are limited to small volumes (<1 cm^3) due to long data acquisition times. Another method simulates vascular angiogenesis, where vessels "grow" according to measured or theoretical distributions of vascular endothelial growth factor [88]. These models are used frequently in research studies on cancer, vascular injury, and embryonic development [89–91]. The angiogenesis simulations require modeling of cellular interactions and vascular growth, both of which are computationally expensive, thus limiting the simulations to small volumes (<2 mm^3). Another method is constrained constructive optimization (CCO) [19], which fills the modeled volume with vessels using a stochastic approach which is constrained by geometrical and physical quantities like vascular density, sheer stress, and blood flow [20]. Due to its random nature, the method constructs networks that are geometrically similar to those of the human body [92]. Recently, Linninger *et al.* [21] coupled CCO with a method to create mesh-like micro-vascular networks in order to completely model 27 mm^3 of the human brain with a high degree of completeness and realism. Yet another method uses self-symmetric fractals, which can accurately reproduce the statistical distributions of network properties [22–24], but are limited in their ability to create a realistic 3-d network [26]. These models have typically been applied to the arterial tree of an organ [22, 25]. While each of the aforementioned methods is theoretically capable of creating a whole-organ vasculatures, no

attempt has been reported to computationally model the vasculature of any whole organ of a mammal.

The objective of this study was to determine if it is computationally feasible to create a model of the vasculature of the whole human brain containing 9 billion vessels. Using a simple fractal-based geometry algorithm and high-performance computing systems, we estimated the minimum computational requirements to create vasculature with 8 to 17 billion vessels. In particular, we determined the computation speeds, electronic memory requirements, and persistent storage requirements.

2.2. Methods

2.2.1. Vascular Model

We developed an algorithm to create arbitrary-sized vasculatures, comprising a 3-dimensional (3-d) network containing from just 6 vessels up to 17 billion vessels, *i.e.* the number in the human brain. A fractal-based approach was adopted for its algorithmic simplicity and its inherent applicability across multiple dimensional scales.

The network model comprises two symmetric halves, an arterial tree and a corresponding venous tree, each with an equal number of vessels. In this model, each successive bifurcation of vessels is called a generation and occurs at vessel junctions (Figure 2.1). In each tree, a parent vessel branches into two child vessels that are of smaller diameter than the parent. The trees grow toward one another, meeting at the midplane. This technique produces a mirror-symmetric network that begins with one artery and ends with one vein.

The model includes descriptors of each vessel's geometry and its connections to other vessels in the network (Figure 2.2). The geometric descriptors include the start and end locations of a vessel and its inner (lumen) radius. The descriptor of a connection of vessels includes the identity of each vessel in that connection and the labels for the junctions at both ends. Junctions in this work refer to the location of abutments between two vessels (*i.e.*, where a capillary from the venous tree contents to one in the arterial tree), bifurcation points (*i.e.*, where three vessels connect), and the start and end points of the vascular network.

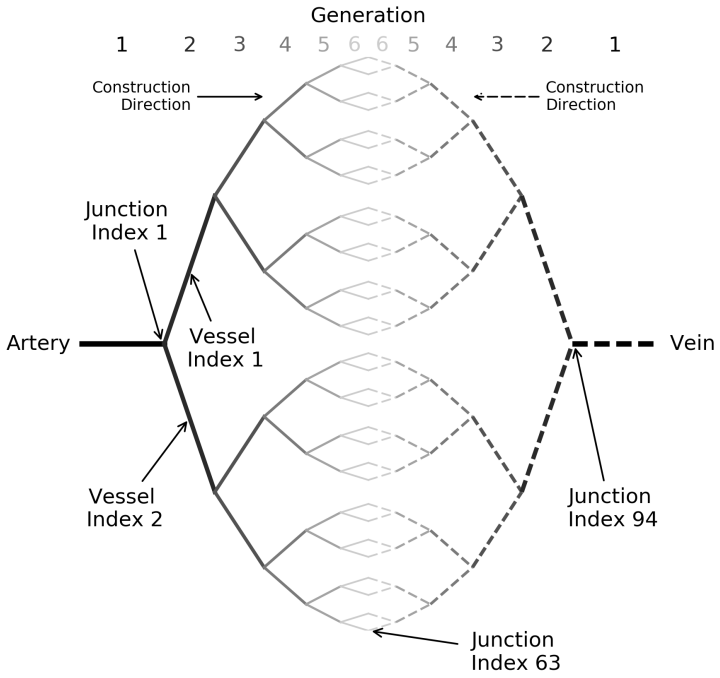


Figure 2.1. An illustrative example of a 2-dimensional vascular network generated with fractal methods. The arterial tree is represented with solid lines, while the venous tree is represented with dashed lines. The direction of construction in each tree proceeds from the largest to smallest vessel.

2.2.2. Algorithm to Construct Simple Vascular Networks

2.2.2.1. Algorithm Overview

The first step in creating a 2-d or 3-d vascular network was to calculate the start and end locations of the network (Figure 2.3a). The start location corresponds to the source of the arterial tree and the end location to the sink of the venous tree. Next, the first vessel in each tree was constructed (Figure 2.3b), which was followed by bifurcations. For each successive generation, we reduced the vessel radius, bifurcation angle, and length using scaling factors selected to avoid geometric overlapping of vessels. Then child vessels were constructed for each parent (Figure 2.3c-f). This branching process occurs for N_{G} generations of vessels in both trees, *i.e.*, until the two trees meet at the midplane. After all generations were constructed (in both trees), the vessel descriptors were updated with information on vessel connections at the midplane. We constructed the arterial and venous vessels in the trees

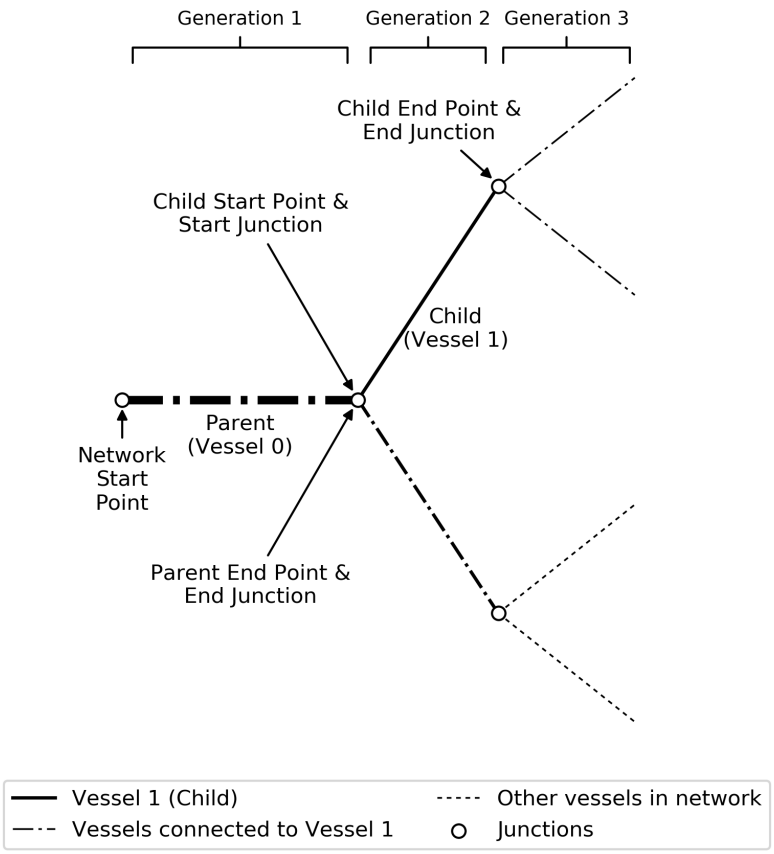


Figure 2.2. An illustration of the types of descriptors used in the geometric model of the arterial tree. Specifically, this highlights the vessels that are connected to a Child vessel (Vessel 1), including its parent. In this example, the two vessels at the bottom are not associated with the Child but are still part of the same network. The network start location is used to specify boundary conditions for blood flow calculations.

simultaneously for two reasons. First, because of the mirror symmetry, most of the geometric computations were shared, which reduced the calculation time required. Second, doing so guaranteed that the trees met at the midplane, thus simplifying their connections to one another. The number of vessels in a completed network is

$$N_{\mathrm{V}} = 2^{N_{\mathrm{G}}+1} - 2 \tag{2.1}$$

where N_{G} is the number of generations in the network. 3-d networks were created by alternating whether the bifurcations occurred in the x-z or y-z planes. Figure 2.4 shows a flowchart of key aspects of network construction.

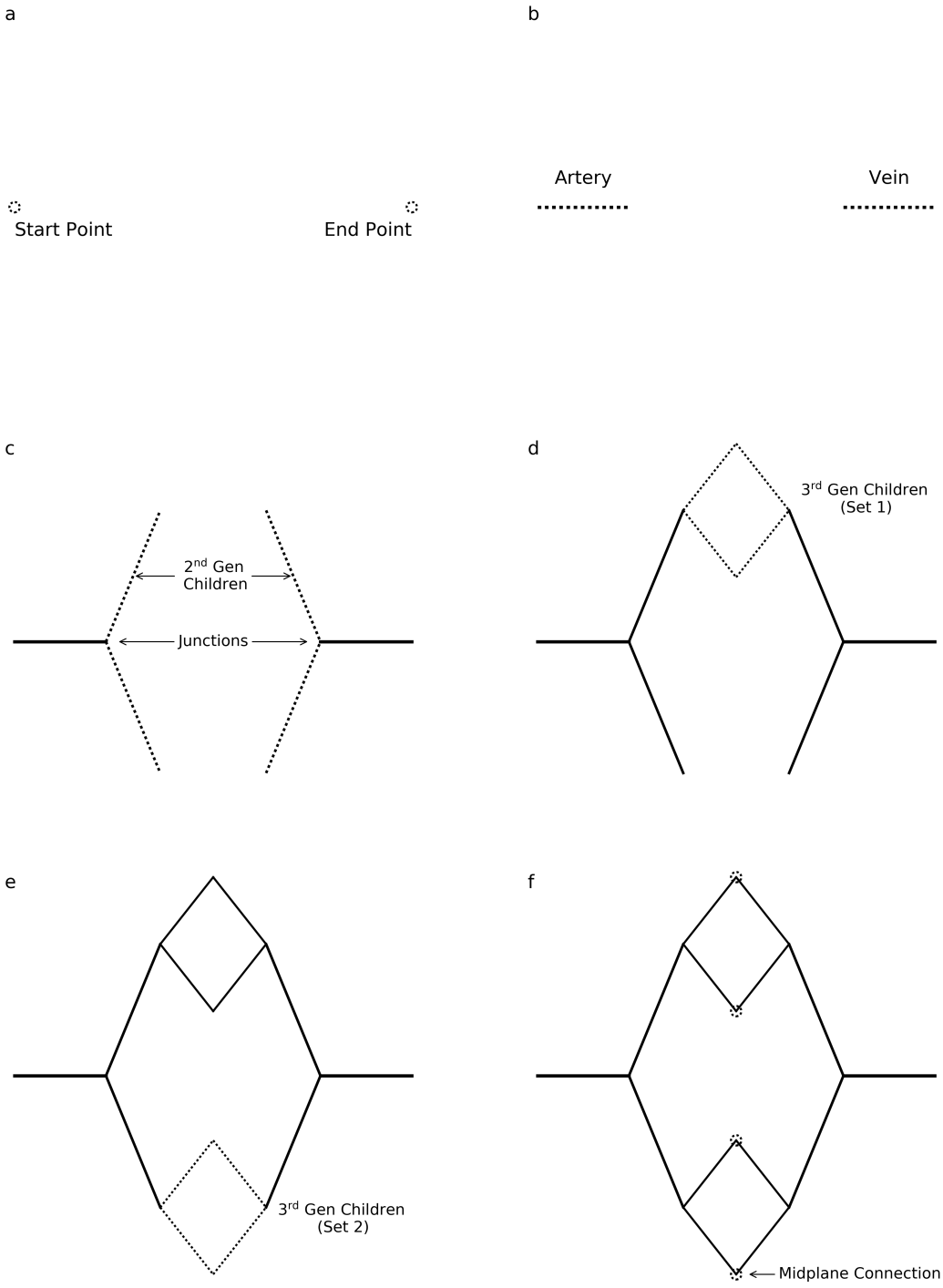


Figure 2.3. An overview of the sequence of steps in constructing the vascular network. Panel (a) depicts the network start and end points, (b) the first generation of vessels in the network, (c) the second generation of vessels, (d) the first set of vessels in the third generation, (e) the second set of vessels in the third generation, and (f) midplane connections and the completed network. Solid lines represent previously constructed vessels. Dotted lines represent newly constructed vessels that are not yet fully connected to the network.

Table 2.1. The user defined parameters that govern the construction of 2-d and 3-d vascular network models.

	Value		
Parameter (units)	2-d	3-d	Source
Number of Generations	$3 \leq N_G \leq 32$		This work
Geometry Dimensions	2	3	This work
Radius of the Smallest Radius (µm)	2.5	2.5	Linninger *et al.*[21]
Radius Scaling Factor	$2^{-1/3}$	$2^{-1/3}$	Adam [93]
Length of the Smallest Vessel (µm)	55	55	Lauwers *et al.* [12]
Length Scaling Factor	0.8	0.8	This work
Initial Bifurcation Angle (°)	70	33.31	Adam [93]
Bifurcation Scaling Factor	0.6	0.88	This work

To construct a new pair of child vessels at a bifurcation, we used the end point of the parent vessel as the starting point of each child vessel (Figure 2.2). We calculated end points for each child vessel based on the bifurcation angle and vessel length of that generation. Next, the connection descriptors were computed. Each new record of a child vessel was assigned its parent's end bifurcation identifier and a new, unique identifier for the new child's end junction. Figure 2.4 shows a flowchart of the algorithm for implementation on a single computer.

Table 2.1 lists the parameters that govern the geometry of the vessel network. These user-supplied parameters included the number of vessel generations, selection of 2-d or 3-d network geometry, radius of the smallest vessel, length of the smallest vessel, the bifurcation angle of the first generation, and scaling constants describing how these quantities change with each successive vessel generation. We selected the initial bifurcation angle for 3-d networks from Adam [93]. For 2-d networks, the initial bifurcation angle was selected to avoid overlap of the vessels in order to enhance visual clarity. The bifurcation-scaling factor ratio and length-scaling factor were selected to eliminate geometric overlapping of vessels in 2-d and 3-d networks.

After constructing the geometric description of the vascular network, the data was saved to persistent storage. This enabled the data to be used in other applications, such as programs that calculate blood flow and radiation transport.

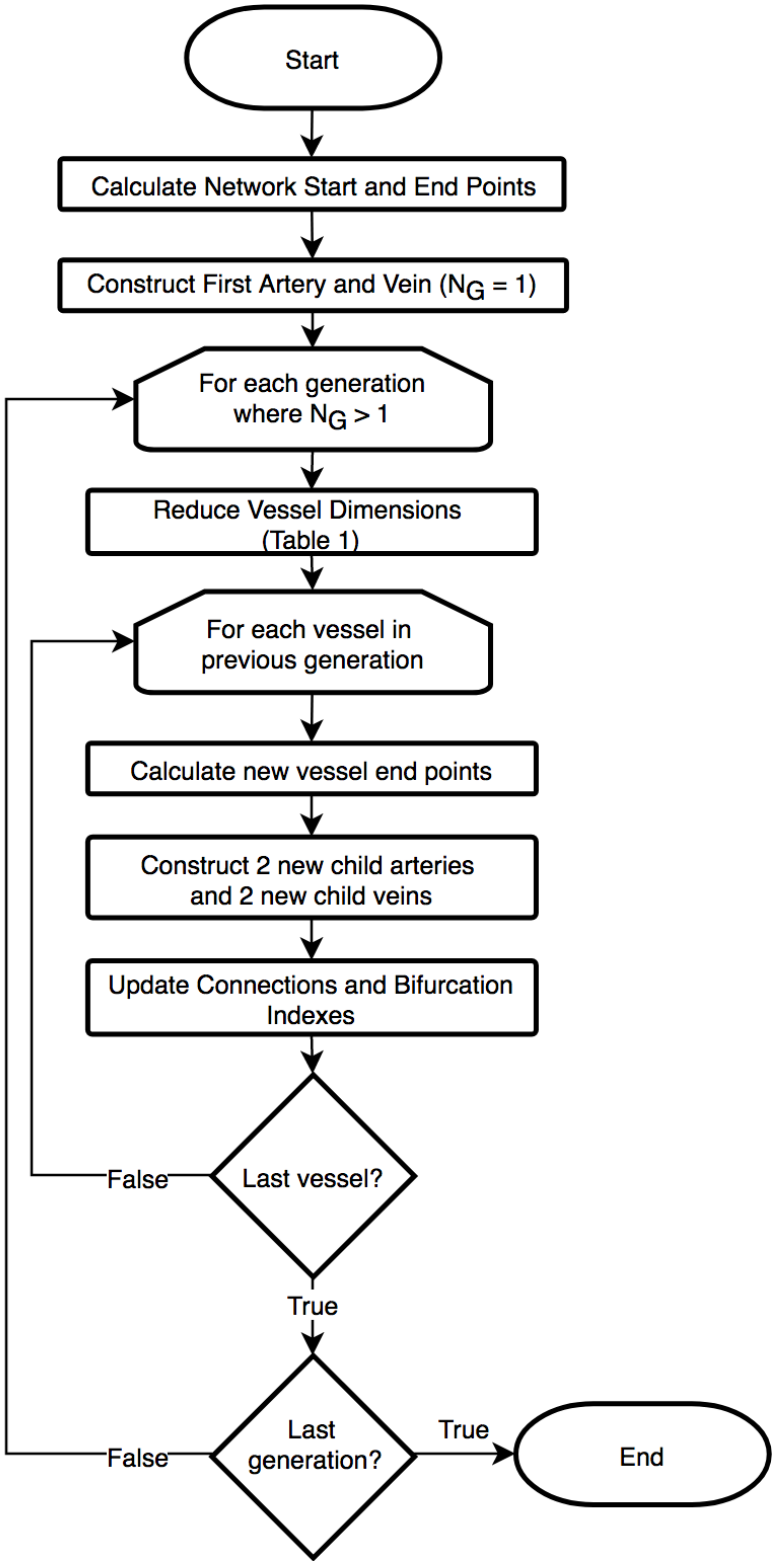


Figure 2.4. Flowchart of the algorithm to create a vessel network when implemented to run on a single computer.

2.2.2.2. Distributed Parallel Algorithm to Construct Simple Vascular Networks

Whole-organ vascular networks occupy vastly more electronic memory than is available on a typical contemporary compute node. To overcome this obstacle, we implemented a divide-and-conquer approach. This enabled parallelization across multiple compute nodes, which facilitated the construction of networks with more than 26 generations, corresponding to approximately 135 million vessels.

In the first stage of the parallel algorithm, each compute node constructed the root trees (Figure 2.5a) of the arterial and venous networks. The root trees were constructed following the methods in Section 2.2.2.1, however construction was stopped after a specified number of generations (N_{root}). Specifically, the number of vessels in the final generation of the root trees was equal to the number of compute nodes being used (N_{n}). The number of generations in a root tree (Figure 2.5a) is

$$N_{\text{root}} = \log_2 (N_{\text{n}}) + 1 \ . \tag{2.2}$$

The second stage connected the root trees (Figure 2.5a) by constructing sub-networks (Figure 2.5c). To accomplish this, each compute node was assigned a pair of vessels to connect (Figure 2.5b) and then used the algorithm discussed in Section 2.2.2.1 to connect the assigned vessels. Each sub-network (Figure 2.5c) was constructed independently on a single compute node.

2.2.2.3. Implementation of the Algorithm

Two versions of the vascular construction algorithm were implemented: one for a single compute node and another for a distributed-parallel computing system. We implemented the algorithms in the C++ language and compiled them with commercial tools (Intel C++ Compiler 2018, Intel Corporation, Santa Clara, CA, United States). The algorithms were designed with shared-memory parallelism using OpenMP [94] and distributed memory par-

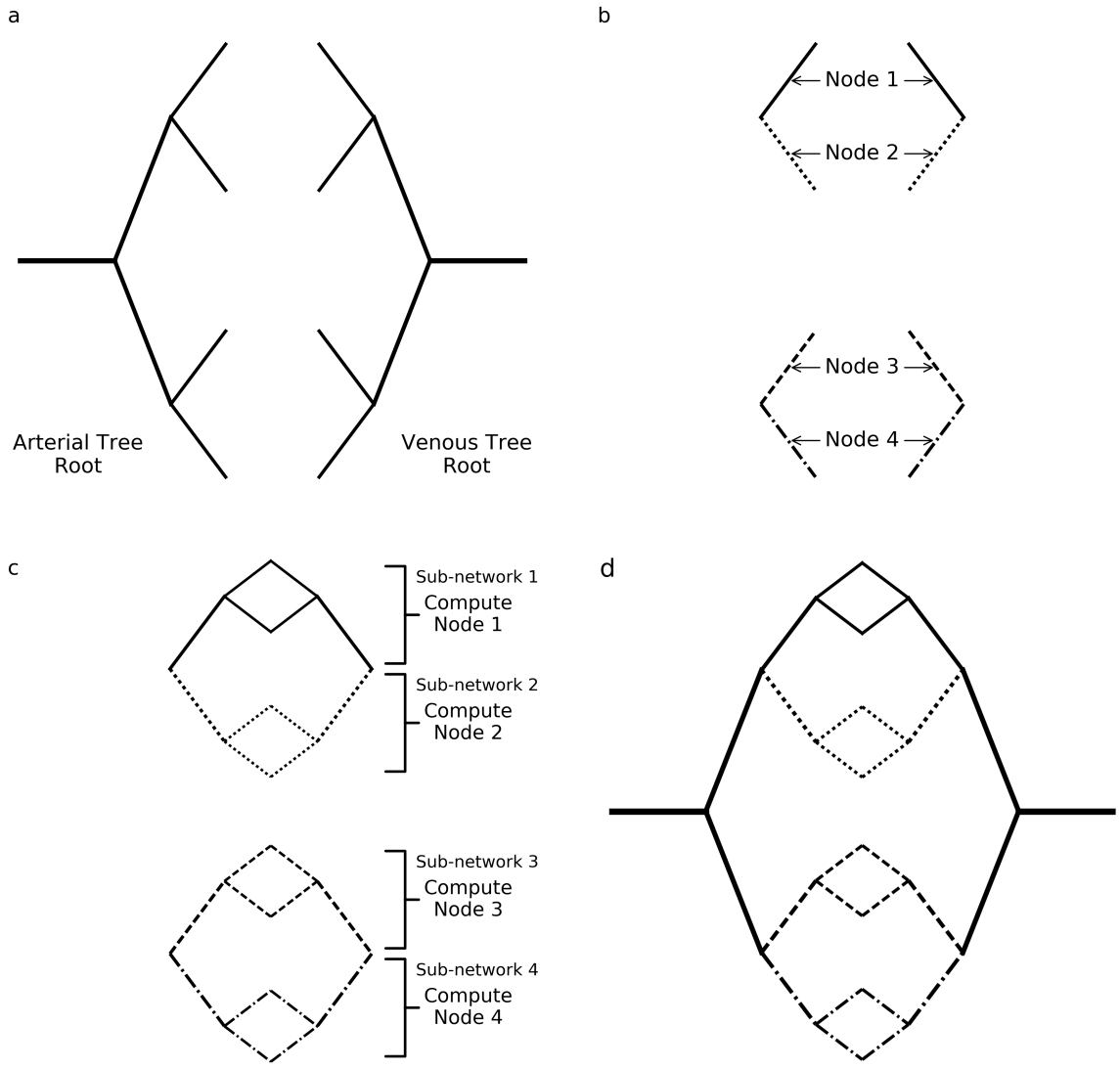


Figure 2.5. Illustration of the basic scheme for the division of work in the distributed-parallel version of the vascular construction algorithm. This example network has four generations and was created by four compute nodes. Panel (a) shows the root networks that are constructed on all the compute nodes. In panel (b) the vessels assigned to each compute node are shown, while the rest of the root trees are written to persistent storage. In panel (c), each line style represents the sub-network constructed on each compute node. The complete network, as it was described in persistent storage, is shown in panel (d).

allelism using the Message Passing Interface (MPI) [95]. We used the implementations of these parallel constructs provided with the commercial tools.

Persistent storage operations were handled with the parallel Hierarchical Data Format version 5 (HDF5) library [96]. A single file stored all the data for the vascular network. In

the single-node version of the algorithm, all output (I/O) was performed after the entire network was constructed. In the parallel algorithm, the I/O occured twice. First, the head node stored the root trees (Figure 2.5a) prior to beginning construction of its assigned sub-network. Second, following the construction of all the sub-networks all the compute nodes simultaneously stored the sub-networks (Figure 2.5c) to the same file using the collective facilities of HDF5. This process was tuned following the techniques of Howison *et al.* [97] to maximize performance.

The vessel descriptors were stored in memory using C++ classes. Signed 64-bit integers stored the vessel and junction identifiers. 64-bit floating point variables limited the impact of rounding and truncation errors associated with the large dynamic range of the vessel dimensions, *i.e.*, from micrometers to centimeters. The vessel classes were stored in an array-based heap to leverage the constant lookup time while resident in electronic memory. One instance of the vessel class requires only 136 bytes of memory. This corresponds to a total of 17 GB for a 135 million vessel network, which is available on most contemporary computers and individual compute nodes of clusters in research settings.

2.2.3. Feasibility Testing

2.2.3.1. Test Environment

We performed the feasibility tests on the SuperMIC cluster at Louisiana State University's High-Performance Computing Center. This cluster comprises 360 nodes, each with two 10-core processors (2.8 GHz Xeon Ivy Bridge-EP E5-2680, Intel Corporation, Santa Clara, CA, United States) and 64 GB of shared electronic memory. The cluster is networked with an InfiniBand Interconnect [98]. A parallel file system consisting of 840 TB of Lustre disks [99] provided persistent storage, enabling up to 16 compute nodes to operate on the file simultaneously.

2.2.3.2. Computation Speed and Scalability

To determine the computational speed and scalability of the algorithm, we created vessel networks ranging in size from 14 vessels to 17 billion vessels (3 to 33 generations). The number of compute nodes (N_{n}) used to create a network with N_{G} generations was determined by

$$N_{\mathrm{n}} = \begin{cases} 1 & N_{\mathrm{G}} \leq 26 \\ 2^{(N_{\mathrm{G}}-26)} & N_{\mathrm{G}} \geq 27 \end{cases}. \tag{2.3}$$

Recall that the divide-and-conquer approach was used to meet the sizable memory requirements of large vascular networks (*e.g.*, N_{G}>26). For each of the networks constructed, recorded the I/O time, vessel network construction time, and the total elapsed wall-clock time. These measurements were repeated five times in order to characterize variations in the execution times. We normalized all times to core-hours to reduce dependence on number of compute nodes to complete the calculations in this study.

To assess the algorithm's scalability, we calculated the speedup factor (S), or

$$S = \frac{T_1}{T_n} \tag{2.4}$$

where T_1 is the time it takes a single compute node, and T_{n} is the time it takes n compute nodes to complete the same task. The task was to create a 26-generation vessel network. This size was selected because it was the largest that could fit in the electronic memory of a single compute node. We recorded the times for 1, 2, 4, 8, 16, 32, 64, and 128 compute nodes to create the 26-generation vessel network.

2.2.3.3. Requirements on Electronic Memory and Persistent Storage

To quantify the amount of electronic memory required by the algorithm, we profiled the application while constructing vascular networks with N_{G} equal to 1, 13, 19, 26, 27, 29, 31, and 33 generations using an open-source memory profiling tool (Valgrind Massif

Table 2.2. The number of vessels (N_V) in a network for select number of generations (N_G). The third column (N_n) shows the number of compute nodes required to construct a network of size (N_V) and (N_G).

N_G	N_V	N_n	Comment
3	14	1	Smallest Network
9	1,022	1	
17	262,142	1	Approximate Largest Previously Reported [21]
20	2,097,150	1	
21	4,194,302	1	Approximate Mouse Brain [102]
26	134,217,726	1	Largest on Single Compute Node
27	268,435,454	2	
28	536,870,910	4	
29	1,073,741,822	8	
30	2,147,483,646	16	
31	4,294,967,294	32	
32	8,589,934,588	64	Approximate Human Brain [12]
33	17,179,869,176	128	Approximate Human Body[58]

[100]). These networks sizes were selected because they span the current operational range of our algorithm. We recorded the average and peak amounts of memory usage. For networks requiring more than one compute node ($N_G \geq 27$ generations), we recorded the peak memory requirement and average memory requirement per node.

Persistent storage requirements of the complete vessel network were measured with operating system utilities. For vessel networks with $3 \leq N_G \leq 32$, the resulting uncompressed and compressed HDF5 file sizes were recorded. Compression was performed *post-hoc* using the GZIP algorithm [101].

2.3. Results

2.3.1. Computation Speed and Scalability

Figure 2.6 plots the total time to construct and store vessel networks as a function of network size (Equation 2.1), revealing that a network with 8.5 billion vessels (approximately the number in the human brain) was created in 425 core-hours, corresponding to only 22 minutes of wall-clock time. The algorithm constructed a network of 17 billion vessels, *i.e.*, doubling the number of vessels in the human brain, in 48 wall-clock minutes. This demon-

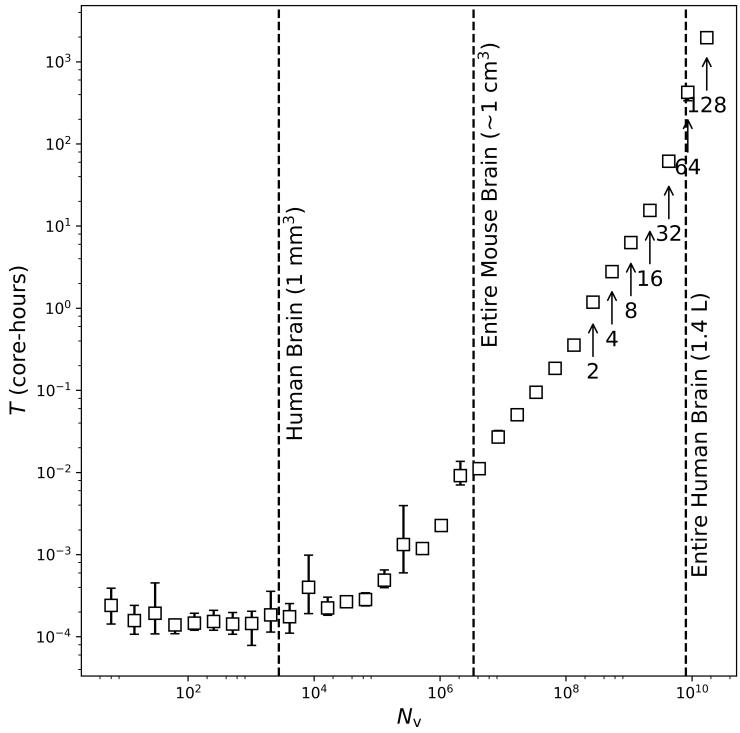


Figure 2.6. The time (T) versus the number of vessels N_{V} created by the algorithm. The error bars represent the maxima and minima of the 5 trials run for each value. The data labels specify the number of compute nodes (N_{n}) used to create the network. Points without a label were run on a single compute node. For context, the vertical lines represent estimates of the numbers of vessels in the human [12] and mouse [102] brains.

strates that our algorithm is capable of constructing network with the number of vessels for any organ in small animals and humans, and is potentially capable of modeling the 20 billion vessels contained in the whole human body [58]. Our algorithm created a whole-brain vascular network of a rodent (approximately 3 million vessels [102]) in less than 15 milliseconds using a single compute node.

Figure 2.7 plots the relative portion of time spent on network construction. I/O time becomes more important as the network size increases. In the smallest networks, most of the execution time is spent performing the network construction. In the largest network, the I/O operations comprise on average 99% of the compute time. Deviation from monotonic behavior was caused by contention on the I/O servers due to competition from unrelated compute jobs on the shared resources of the cluster.

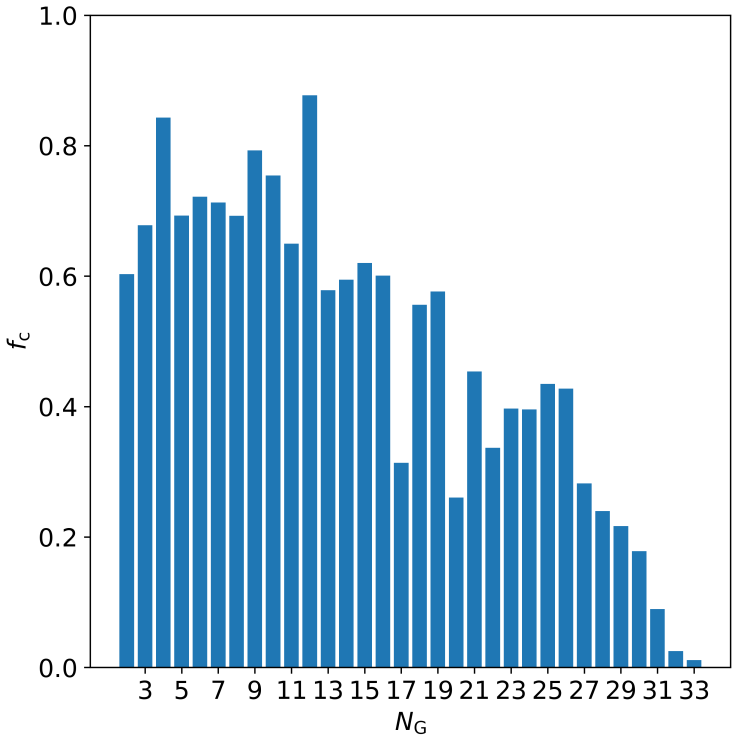


Figure 2.7. The relative fraction of time spent performing network construction (f_c) *versus* the number of generations in the network (N_G). In small networks ($N_G \leq 16$), the vascular construction algorithm dominated the total computation time. For larger networks, the I/O (writing to disk) predominated.

We performed scalability measurements on our algorithm by varying the number of compute nodes used to construct a 26-generation network. Figure 2.8 plots the overall speedup, as well as the speedup of construction and I/O operations individually. This plot reveals nearly-perfect scalability of the vessel network construction algorithm. Additionally, it highlights how I/O operations limit the scalability and speed of the algorithm, in its current form.

2.3.2. Requirements of Electronic Memory and Persistent Storage

Profiling confirmed a linear scaling of the electronic memory required as a function of the number of vessels (Figure 2.9). The average memory required per node was constant for networks that utilized more than one compute node. The peak memory usage always occurred during the I/O operations; this was caused by the conversion from data structures

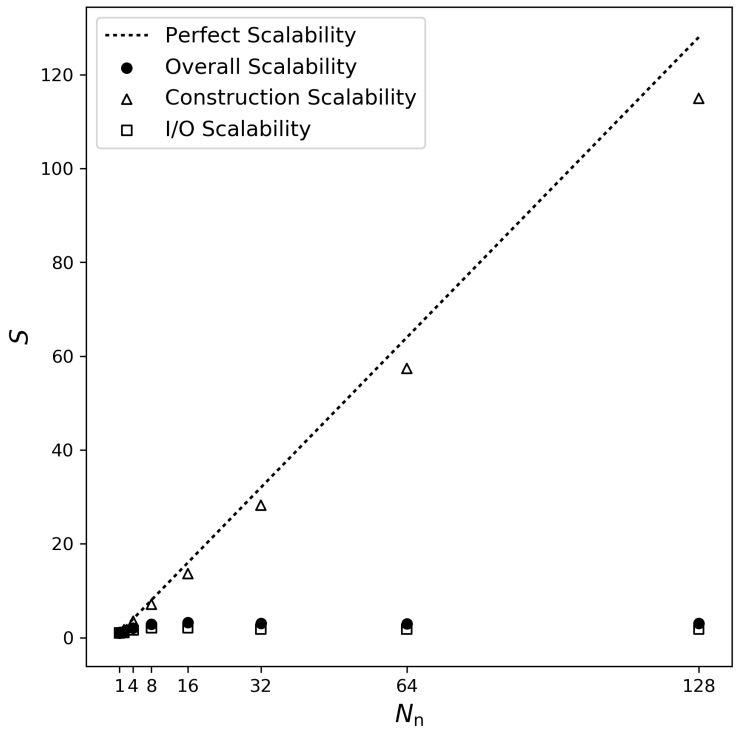


Figure 2.8. Speedup (S) versus the number of compute nodes (N_n). The construction algorithm shows strong scalability. I/O operations demonstrated minimal scalability, which limited the overall speedup.

in electronic memory to persistent storage. This conversion process is slightly different in the non-parallel and parallel versions of the algorithm, as evidenced by the slight difference in the peak memory used between generations 26 and 27 (Figure 2.9).

The persistent-storage file size increased linearly with the number of vessels in the network (Figure 2.10), as expected. For the largest network size (17 billion vessels), lossless compression reduced the file size from 1.6 TB to 182 GB. On average, output file sizes were compressed by a factor of approximately 8.

2.4. Discussion

In this study, we used fractal-based geometry to construct, for the first time, a simple vascular network containing 17 billion fully-connected vessels. We evaluated the algorithm's computational speed, scalability, electronic memory requirements, and persistent-storage requirements using high-performance computing hardware. The major finding of this work

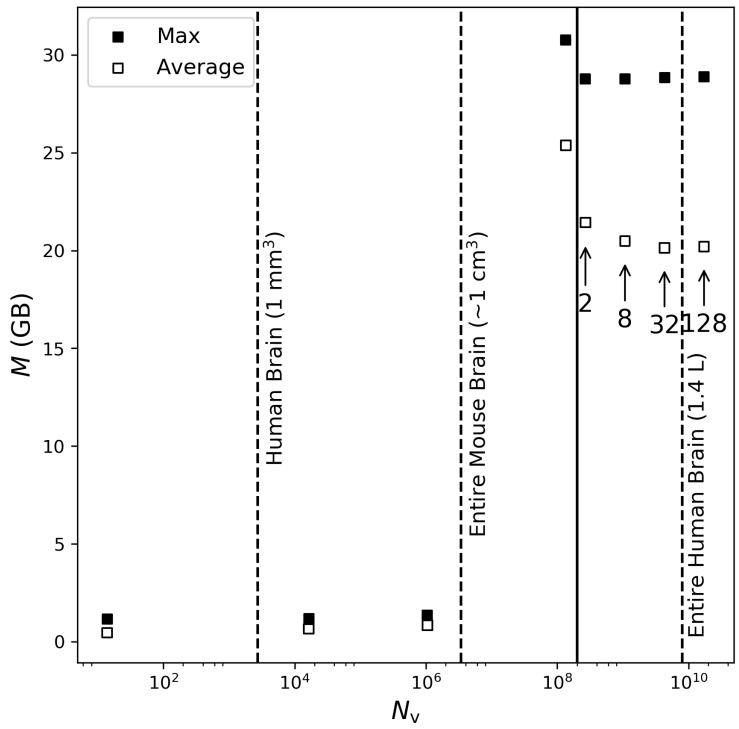


Figure 2.9. Memory (M) per node versus number of vessels (N_{V}). Squares represent maximum memory use, while circles represent the average use during the run. The solid line represents the transition from the single node algorithm to the distributed parallel algorithm. The data labels specify the number of compute nodes (N_{n}) used to create the network. For context, the vertical dashed lines represent the estimates for the numbers of vessels vessels in the human [12] and mouse [102] brains from the literature.

is that it is computationally feasible to construct and store whole-organ vascular networks. Additionally, we found that the algorithm is fast, scales strongly, and is easily accommodated by contemporary computers.

The implication of this work is that modeling the vasculature of whole organs may open new lines of inquiry that were previously thought impossible. For example, modeling of whole tissue perfusion at the capillary level may find applications in research on drug transportation, circulating tumor cells, and vascular injury. Additionally, it could simplify the implementation of blood flow algorithms by eliminating complex boundary conditions at the edges of simulation domains [21, 46]. That said, much additional research and development will be needed to achieve the full potential of this approach. For instance the relative simplicity of our fractal geometry could be enhanced or replaced with more anatomically-

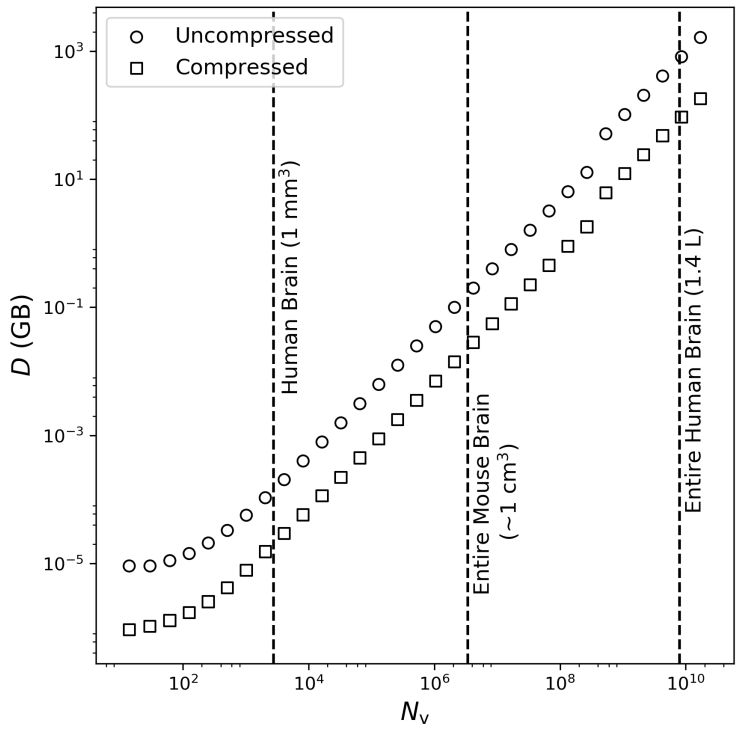


Figure 2.10. Output file size (D) versus number of vessels (N_V). The circles show the uncompressed size, while the squares show the size when compressed. The tail for small networks ($N_\mathrm{V} < 256$) is due to a fixed header size in our HDF file. For context, the vertical dashed lines represent the literature estimates of the number of vessels in the human (Lauwers et al. 2008) and mouse (David Mayerich, Kwon, and Choe 2008) brains.

realistic geometry. The divide-and-conquer technique could be coupled with more advanced algorithms, such as constrained constructive optimization [103], improving the geometric realism of the vasculature. More broadly, the geometric model and algorithm proposed in this work provides a simple, scalable geometry that can be used to study the performance of blood flow simulations, vascular visualization algorithms, and other computational research applications.

The capabilities of computational models may be broadly characterized according to the number of vessels in the network, geometric realism, and algorithmic qualities. In previous studies, the largest achieved network sizes were less than 300,000 vessels [21, 104, 105]. The capacity of our model, in its current form, was 17 billion vessels, or approximately 56,000 times greater than the largest previously-reported network size. Most studies in the literature

sought to achieve high geometric accuracy of the constructed vascular networks. These used various techniques, including image segmentation [55, 104, 105], constrained constructive optimization [16, 103], global constructive optimization [106], and self-symmetric fractals [25]. The goal of these algorithms was to provide realistic geometries for simulations, typically of blood flow. In contradistinction, the goal of this study was to determine the largest achievable network size. We selected a fractal-based approach because it allowed us to probe the hitherto uncharted computational aspects of whole-organ and whole-organism vasculatures, *e.g.*, feasibility, scalability, and minimum requirements to create large vascular networks. Of the previous literature available on computational vascular network modeling, only one work [106] discussed the computation time required by the algorithm, and it reported the times to construct only two comparatively small networks. Our study reported performance information for 30 different network sizes ($14 \leq N_\mathrm{V} \leq 17 \times 10^9$), yielding the first set of data on scalability. In addition, our study characterized the benefits of parallel computing and the divide-and-conquer approach.

The major strength of this work is its use of generally-applicable methods and widely-available computational tools and hardware, which facilitates replication and future extensions. Additionally, the algorithm creates a reproducible and non-stochastic geometry that is potentially well suited to benchmark problems of relevance to computer-science aspects of vascular modeling. Benchmark problems will be needed in the future, *e.g.*, when comparing the performance of algorithms to predict blood flow. Using a benchmark geometry across research studies would facilitate meaningful comparisons. Another strength of this work is its focus on the computational performance of the vessel construction algorithm, which may inform future work to develop more realistic, whole-organ vasculatures. Any algorithms with increased geometric realism will require more computational operations to create the same number of vessels, thus requiring more computer resources. Therefore, our model provides estimates of the minimum computational requirements, which may inform the design of future studies in this area.

This work has several limitations, most notably, the modest anatomical realism of the vessel geometry. For example, we assumed a 1:1 ratio of arteries to veins in our network. This ratio is approximately 3:1 in normal human brain tissue [75]. Furthermore, we used a symmetric fractal network whereas real vascular networks are partly stochastic in nature. These are not serious limitations, however, because we believe the divide-and-conquer strategy could be adapted to stochastic methods, such as constrained constructive optimization [103], in the future. Another limitation of this and previous studies is that the number of blood vessels in the human brain is poorly known, which makes it difficult to know, with confidence, if the algorithm can indeed model the number of vessels in the brain. Our estimate of 8.5 billion vessels is in the range 3 to 9 billion vessels estimated from the sparse literature [12, 13, 58, 76]. This is not a serious limitation because the proposed algorithm was able to create a vascular network with 17 billion vessels, or approximately double the best estimate of the number of vessels in the brain. Additionally, the findings on scalability suggest that, with additional computational resources, which already exist at other institutions, the algorithm could construct even larger vascular networks. Finally, one minor limitation is that we did not calculate vessel wall thickness in this study. Analytical methods of calculating the wall thickness are available in the literature Avolio [107]. We implemented these methods in a companion study with negligible increase in execution time (Chapter 6).

Further investigations are currently underway in our laboratory to demonstrate the feasibility of blood flow simulations in whole-organ vascular networks, using the vascular model from this work. Additionally, the geometric model is being used to benchmark an algorithm that simulates radiation dose to vascular networks in whole organs. With the ability to create geometries with billions of vessels, more detailed studies involving individual microvasculature will be possible.

3. Feasibility of Calculating the Steady-State Blood Flow Rate Through the Vasculature of the Entire Human Body

3.1. Introduction

All mammalian life depends on a circulatory system to transport oxygen and nutrients to organs such as the central nervous system (CNS), liver, and skin. The circulatory system includes the heart, lungs, arteries, capillaries, and veins [76]. To ensure tissues receive adequate oxygen and nutrients, the CNS and local cellular environments monitor and modulate blood flow rates throughout the circulatory system [108, 109]. Blood flow may be altered by disease processes [110], internal and external environmental conditions [111, 112], and stress and injury [113, 114]. In the United States, cardiovascular and cerebrovascular diseases resulted in approximately 775,000 deaths in 2015 [115] and are expected to cost $1 trillion per year by 2030 [116].

A great deal is known about the physics of blood flow. The canonical Naiver-Stokes Equations, derived from Newton's equations of motion, describe the time-dependent changes in fluid flow. To solve these equations computationally, the vascular geometry may be broken into a finite element mesh resulting in a system of partial differential equations [117], which is a well understood and widely used technique. Researchers have used this technique to simulate strokes [50, 51] and stent safety and effectiveness [49, 118] to a high degree of detail. While it is theoretically possible to solve the three-dimensional (3-D) Naiver-Stokes equations for all the vessels in the human body, simulations using these techniques are typically limited to less than 50 vessels [54, 55].

The Naiver-Stokes Equations can be simplified using a variety of assumptions that reduce the dimensionality, enabling the simulation of larger vascular networks. One popular choice is the one-dimensional (1-D) time-dependent approximation where the blood is transported axially along the vessel while assuming angular symmetry of the flow in the vessel [46]. This approach has been used to model the effects of pulsatile flow on blood vessels [57, 119], large portions of the arterial tree [56, 120–122], and to predict changes in blood flow due

to different physiological conditions [25]. Time-independent, or steady-state, models further reduce the dimensionality allowing larger vascular networks to be simulated [123]. This is an attractive approach because it reduces the computational requirements and simplifies implementation [124, 125]. Biologically, blood flow is steady in the capillaries and veins of the human vascular network [119, 121]. This approach has been used to simulate blood flow through the microvasculature of the brain [21] and heart [126] and to determine the relative viscosity of blood in vessels [127]. Finally, lumped-parameter or compartmental modeling groups vessels together into compartments and, using a simple parameterisation, models the transfer of blood between the compartments [27, 46]. This is used extensively in blood tracer studies to model the distribution of a drug in different organs and tissue compartments [128–130]. With this approach, it is possible to model the blood distribution throughout the entire human body but it provides no information on individual vessels [131].

Quarteroni *et al.* [46] provided an excellent review of blood flow modeling techniques and described approaches for coupling compartment, 1-D, and 3-D Naiver-Stokes models to simulate whole-body blood flow. To summarize the literature reports, blood flow can be simulated with high-levels of detail for a few blood vessels or low-levels of detail for entire organism. Previous studies performed detailed simulations in small numbers of vessels and then used simplified models to incorporate the influence of other vessels in the organ or human body. Hitherto, it is currently unknown if computing the blood flow rate through each and every individual vessel in the human body was feasible.

The goal of this work is to test the feasibility of calculating the blood flow rates through 17 billion individual vessels, *i.e.*, approximately the number in the human body [58]. To accomplish this task, we calculated the blood flow rates in a network of vessels that was modeled with fractal geometry. We used a steady-state approach to calculate the blood flow rates. The network was geometrically partitioned using graph analysis to reduce the amount of communication necessary for the blood flow rate calculations, increasing the computational efficiency.

3.2. Methods

3.2.1. Modeling Steady-State Blood Flow

3.2.1.1. Geometric Model

The vascular geometry used in this study was constructed with a symmetric, fractal approach. Vessels were represented as rigid, cylindrical tubes that were connected at junctions to form a network. The term junction denotes any location where vessels meet, or the start and end points of the network, where boundary conditions are applied. The fractal model was conveniently scalable to billions of vessels and had only two boundary conditions, namely the pressure at the inlet and the blood flow rate at the outlet. The number of vessels (N_{v}) in a network was related to the number of bifurcations or generations, denoted by N_{g}, by

$$N_{\mathrm{v}} = 2\left(2^{N_{\mathrm{g}}} - 1\right) = 2^{N_{\mathrm{g}}+1} - 2. \tag{3.1}$$

In the model, each parent vessel branches into two child vessels. The smallest vessels in the network were 2.5 µm in radius and 55 µm in length [12]. Between each generation, the vessel radius was reduced by $2^{-1/3}$ following Murray's Law for bifurcation [93] and the length was reduced by 0.8 to reduce vessel overlap for visualization [132]. Figure 3.1 plots an illustrative example of a 2-dimensional network created using this algorithm. The algorithm used to construct this network geometry was described previously (Chapter 2) [132]. Each vessel was modeled with a start location, end location, and inner radius, as well as junction indexes, and a list of immediately connecting vessels.

3.2.1.2. Fluid Dynamics Model

The Naiver-Stokes Equations are well-established in the field of blood flow simulation. At a point $\boldsymbol{x} = (x_1, x_2, x_3)$ in an arbitrary 3-D coordinate system, the time dependent form

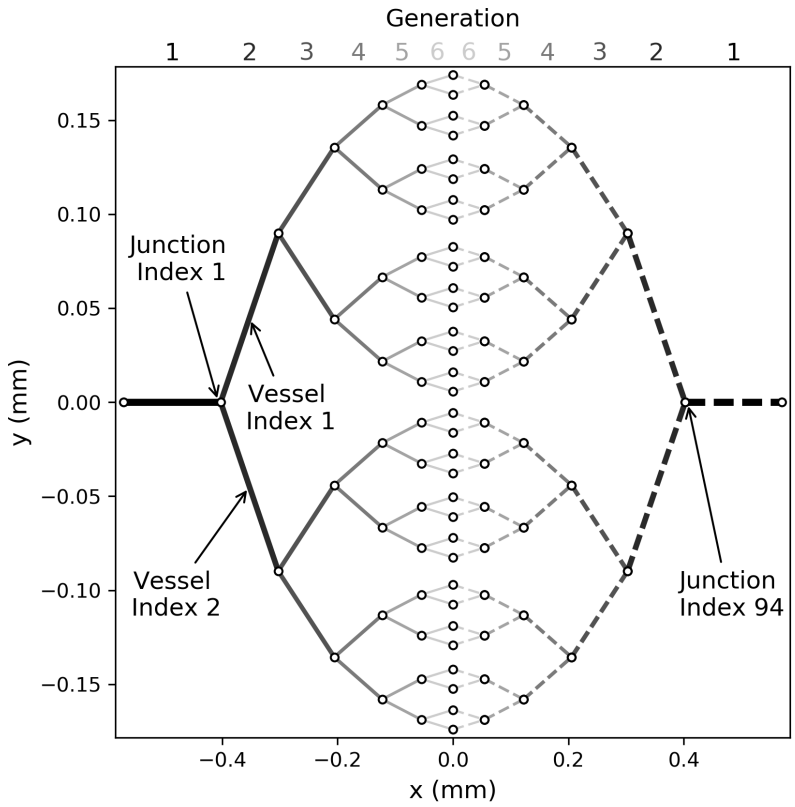


Figure 3.1. A simple illustrative example of a 2-dimesional vascular network with 6 generations. The solid lines represent the arteries, while the dashed lines represent the veins. The circles depict junctions in the network.

of the Naiver-Stokes equations are

$$\rho \frac{\mathrm{d}\boldsymbol{v}\left(\boldsymbol{x},t\right)}{\mathrm{d}t} - \nabla \cdot \boldsymbol{T} = \rho \boldsymbol{f}\left(\boldsymbol{x},t\right) \tag{3.2}$$

and

$$\nabla \cdot v = 0, \tag{3.3}$$

where ρ is the fluid density, $\boldsymbol{v}\left(\boldsymbol{x},t\right)$ is the fluid velocity at time t, $\boldsymbol{T}$ is the fluid stress tensor, and $\boldsymbol{f}\left(\boldsymbol{x},t\right)$ represents the net change in fluid volume at time t [46]. Equation 3.3 describes the continuity of the blood flow through the vascular network, primarily that the fluid is incompressible.

In this work, we simplified the Naiver-Stokes Equations by assuming the flow to be steady, laminar, and fully developed. These assumptions were originally proposed by Poiseuille in 1838 to calculate the pressure drop of an incompressible Newtonian fluid through a cylindrical tube [123]. The solution gives the volumetric blood flow rate (Q) in a vessel of radius r and length L as

$$\begin{aligned} Q &= \frac{\pi r^4}{8\eta L}\left(P_{\text{in}} - P_{\text{out}}\right) \\ &= C \cdot \Delta P \end{aligned} \tag{3.4}$$

where η is the blood viscosity, C is the conductance of the vessel, and P_{in} and P_{out} are the pressures at the inlet and outlet of the vessel, respectively [21]. To specify the viscosity, η, of the blood, we used an empirical model for *in-vitro* blood viscosity adapted from Pries *et al.* [127] for a constant hematocrit of 0.45, or

$$\eta = \eta_0 \left(220 \cdot e^{-2.6r} + 3.2 - 2.44 \cdot e^{-0.06(2r)^{0.645}}\right) \tag{3.5}$$

where r is the radius of the vessel in millimeters and the value of η_0 was $35\,\text{mPa} \cdot \text{s}$ [133].

Because blood is an incompressible fluid, the net blood flow rate at each junction must be zero, or, mathematically,

$$\sum_{m=0}^{n} Q_m = 0 \tag{3.6}$$

where n is the total number of vessels at the junction and Q_m is the blood flow rate of the m^{th} vessel at that junction (by convention the sign of Q_m is positive if blood is entering the junction and negative if leaving). Using this constraint and Equation 3.4, we constructed a system of linear equations that described the net flow rates at each junction to determine the corresponding fluid pressures. For example, in a network with 6 vessels (Figure 3.2) the

system of equations was

$$\begin{aligned} P_0 &= P_{\text{in}} \\ C_A\,(P_0 - P_1) - C_B\,(P_1 - P_2) - C_C\,(P_1 - P_3) &= 0 \\ C_B\,(P_1 - P_2) - C_D\,(P_2 - P_4) &= 0 \\ C_C\,(P_1 - P_3) - C_E\,(P_3 - P_4) &= 0 \\ C_D\,(P_2 - P_4) + C_E\,(P_3 - P_4) - C_F\,(P_4 - P_5) &= 0 \\ C_F\,(P_4 - P_5) &= Q_{\text{out}} \end{aligned} \tag{3.7}$$

where P_{in} in the inlet pressure boundary condition and Q_{out} is the outlet blood flow rate. We specified one pressure and one flow rate boundary condition to assure the problem is well defined. These six equations are used to determine the six unknown pressures. This system was cast in matrix form, or

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ C_A & -C_A\text{-}C_B\text{-}C_C & C_B & C_C & 0 & 0 \\ 0 & C_B & -C_B\text{-}C_D & 0 & C_D & 0 \\ 0 & C_C & 0 & -C_C\text{-}C_E & C_E & 0 \\ 0 & 0 & C_D & C_E & -C_D\text{-}C_E\text{-}C_F & C_F \\ 0 & 0 & 0 & 0 & C_F & -C_F \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} P_{\text{in}} \\ 0 \\ 0 \\ 0 \\ 0 \\ Q_{\text{Out}} \end{bmatrix}. \tag{3.8}$$

In this form, the 2-dimensional matrix describes the conductance values of the vessels that connect junctions while the pressure vector describes the unknown pressures at each junction (*e.g.*, P_i's) and the other vector specifies the net flow rate at each junction. For brevity, the 2-dimensional matrix will be referred to as the conductance matrix for the remainder of the manuscript. We solved this system of equations using iterative methods [134] to determine the pressures. With the pressures calculated at all the junctions in the network, we calculated the blood flow rate, Q, through each vessel using Equation 3.4.

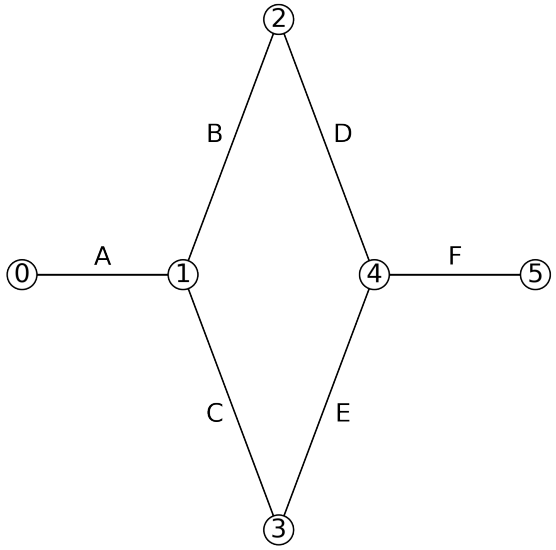


Figure 3.2. Illustration of a simple vascular geometry showing the geometric relationship between the blood vessels and junctions. Junctions are labeled with numbers and vessels labeled letters.

3.2.2. Algorithms

3.2.2.1. General Overview

We designed a two-step algorithm to calculate blood flow rate in a vascular network of arbitrary size. The first step partitioned the vascular network's geometry and created a computationally-efficient division of data using graph partitioning algorithms (see Section 3.2.2.2). The second step calculated blood flow rates through the vessels using matrix inversion techniques (see Section 3.2.2.3). Figure 3.3 is a flowchart of this algorithm.

We called the first step Network Preprocessing and the second step Blood Flow Rate Calculations. We developed separate modules for each step to facilitate validation and testing. Additionally, this modular design simplifies future extensions by allowing additional modules to manipulate the vascular data prior to calculating the blood flow.

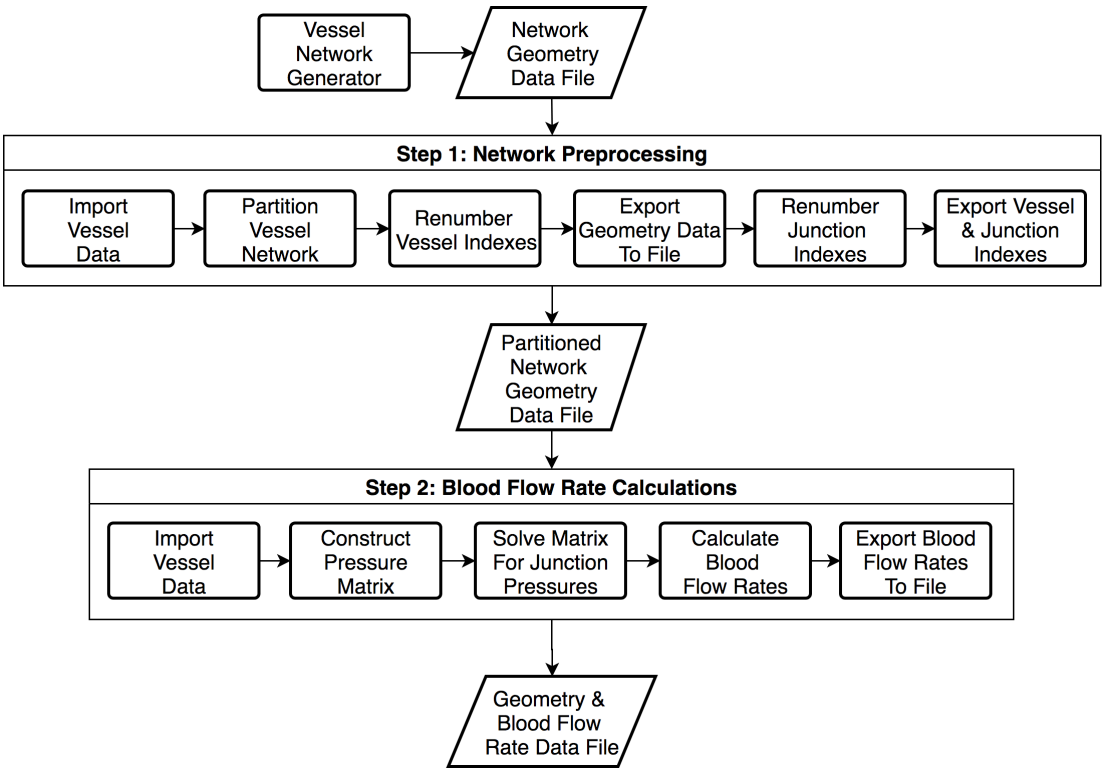


Figure 3.3. Flowchart mapping the two major steps and selected sub-tasks of calculating the blood flow in a vascular network.

3.2.2.2. Preprocessing of the Vessel Network's Geometry

Limitations on the memory available in a single compute node, as well as matrix solver requirements motivated the first step of our algorithm. For networks with greater than 67 million vessels, the required amount of memory to describe the vessel network geometry and the conductance matrix exceeded 64 GB available on a single compute node. Our solution was to distribute the rows of the conductance matrix across multiple compute nodes. The matrix solver used in this study stipulated that rows of the conductance matrix be distributed in contiguous groups among the compute nodes [135], *e.g.*, rows 1 -n on compute node 1, rows n+1 to 2n on compute node 2, *etc.* To minimize inter-node communication, the network geometry was partitioned to reduce the number of compute nodes where a given junction appeared.

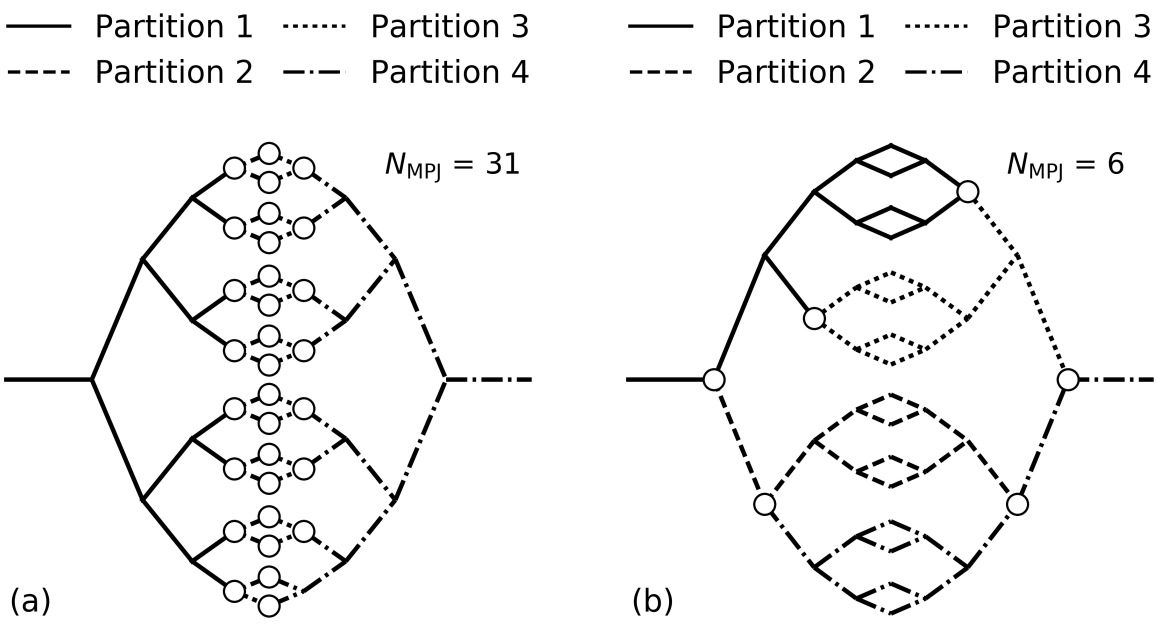


Figure 3.4. Schematic illustration of the effects of graph partitioning on the network data distribution. Panel (a) shows the distribution of data among the compute nodes after sequentially and contiguously importing the data. Panel (b) shows the distribution of data after the graph partitioning and renumbering of junction and vessel indexes. In both panels, the open circles represent multi-partition junctions where communication is necessary for the construction of the conductance matrix. The number of multi-partition junctions (N_{MPJ}) is significantly reduced in Panel (b), greatly reducing the inter-node communication.

The algorithm constructing the network [132] stored the vascular data by level of bifurcation. If the data was read and distributed among compute between nodes sequentially and contiguously, it resulted in many junctions referenced on multiple compute nodes, reducing computational efficiency (See Figure 3.4a). We used a graph partitioning algorithm [136] to group and sort the vessels to minimize the number of multi-partition junctions (Figure 3.4b). A multi-partition junction is a junction where vessels belonging to multiple partitions meet (Figure 3.4). In practice, each partition is assigned to a different compute node during the Blood Flow Rate Calculations. Thus, to construct the row in the matrix related to the multi-partition junction, the conductances of all the vessels must be communicated to the compute node that contains that row of the matrix. To facilitate this communication, one of the partitions was selected as the master partition. The master partition was used to

decide which compute node was responsible for collecting the vessel conductance values and creating the associated row in the conductance matrix. The lists of immediately connected vessels and the junction indexes were renumbered for consistency with the new partitions. The newly partitioned and renumbered data was written to the data file for subsequent processing by the Blood Flow Rate Calculation algorithm.

3.2.2.3. Blood Flow Rate Calculations

Calculating the blood flow rates in the preprocessed network comprised three components: constructing the conductance matrix, solving for the blood pressure at each junction, and calculating the blood flow rate through each individual vessel. To begin, each compute node was assigned one of the partitions created by the Network Preprocessing application. The compute node then imported the relevant junction indexes, boundary conditions, and vessel geometry from the data file.

We calculated the pressures at each junction by constructing the conductance matrix and solving for the unknown pressures. First, the conductance matrix (Section 3.2.1.2) was constructed in compressed sparse row format. This required the communication of conductance values for vessels that met at a multi-partition junction. The system of linear equations was solved using an iterative Krylov method [134]. We applied a boundary condition at the network inlet of $P_{\text{in}} = 133.3$ Pa, which is the average blood pressure of the cardiac cycle [137], and a boundary condition was assigned to the blood flow rate at the outlet using

$$Q_{\text{out}}(N_G) = 2^{N_{\text{G}}-1} \cdot v_{\text{c}} \cdot \pi r_{\text{c}}^2 \tag{3.9}$$

where Q_{out} is the outlet boundary condition, N_{G} is the number of generations in the network, v_{c} is the blood velocity in a capillary, and r_{c} is the radius of a capillary. In our geometric model, the radius of the smallest capillary (r_{c}) was 2.5 µm [12, 13, 75]. Ivanov *et al.*

[138] measured the blood flow velocities in capillaries at approximately $1\,\mathrm{mm\,s^{-1}}$, which we assigned to v_c.

With the pressures at each junction computed, the blood flow rates in the individual vessels were calculated according to the following procedure. First, the pressure for each of the multi-partition junctions was communicated to all the compute nodes. This resulted in each compute node having the required pressure values to calculate the blood flow rates for the vessels in its assigned partition. Then, the blood flow rate of each vessel was calculated using Equation 3.4. To optimize this procedure for computational speed and efficiency, we converted the problem into a matrix-vector multiplication operation.

In order to verify the calculations were free of blunders, each compute node calculated the blood flow rate using the following secondary method. For the specific-case of symmetric vascular networks used in this study and the assumption of steady-state physics, we calculated the blood flow rate (Q) of each vessel of network according to

$$Q = \frac{Q_{\mathrm{out}}(N_G)}{2^{l-1}} \tag{3.10}$$

where l is the generation number of the vessel. This simple procedure allowed us to verify the calculations for networks with up to 17 billion vessels. Following this verification, the blood flow rates were written to persistent storage media.

3.2.3. Computational Feasibility Measurements

3.2.3.1. Testing Environment

We measured the performance of the algorithm on the QB2 cluster hosted by the Louisiana Optical Network Initiative. This shared resource has 504 compute nodes. Each compute node contains 64 GB of memory and two 10-core processors (2.8 GHz Ivy Bridge-EP E5-2680 Intel Xeon 64-bit). An Infiniband Interconnect [98] provides internode communication and access to persistent storage, comprising a 2.8-PB parallel file system. The file

system uses a Lustre architecture [99], enabling up to 16 compute nodes to operate on a single file simultaneously.

3.2.3.2. Software Environment

We compiled all software developed in this study with a commercial C++ compiler (Intel C++ Compiler Cluster Edition v18.0, Intel Corporation, Santa Clara, CA, United States). The applications utilized shared-memory parallelism, distributed-memory parallelism, and linear algebra routines. The compiler provided an implementation of OpenMP [94] for shared-memory parallelism, an implementation of the message passing interface (MPI) for distributed memory communication (Intel MPI v18.0), and the Math Kernel Library (MKL v18.0) of optimized linear algebra routines.

All input and output (I/O) operations to persistent storage for the applications utilized the Hierarchical Data Format version 5 (HDF5) library [139]. Our applications used the parallel I/O capabilities of the HDF5 library and the Lustre file system to increase speed. We tuned the I/O operations following the suggestions of Howison *et al.* [97].

Graph partitioning in the Network Preprocessing application utilized a k-way partitioning algorithm implemented in the parallel METIS library (ParMETIS version 4.0.3) [136, 140]. We used the k-way partitioning routine on an edge graph of our vascular network. The ParMETIS library was compiled with support for large indexes (64-bit) and double-precision real (64-bit) values to accommodate networks with up to 2.3×10^{18} vessels.

Matrix solving was completed with the Loose Generalized Minimum Residual algorithm (LGMRES) [141], a Krylov-based iterative solver. We used the BoomerAMG solver [142] as a matrix preconditioner to accelerate the convergence of the iterative solution to the system of equations. Implementations of these algorithms were provided by the Hypre Solver Library [135, 143]. We compiled the library with support for OpenMP and large (64-bit) indexes. Table 3.1 list the user-specified parameters and options for the matrix solver and preconditioner. These parameters were determined through an iterative tuning process.

Table 3.1. Configuration parameters for the LGMRES [141] and BoomerAMG [142] solvers used in this work.

LGMRES Matrix Solver Settings	
CoarsenType	10
StrongThreshold	0.25
MeasureType	1
KDim	35
Tol	1×10^{-6}
AbsTol	2×10^{-6}
MaxIter	150
BoomerAMG Matrix Solver Settings	
KeepTranspose	True
MaxLevels	20
MaxIter	1
Tol	0
RAP2	True
InterpType	7
PMaxElmts	6
AggNumLevels	2
AggInterpType	1
AggP12MaxElmts	6
AggPMaxElmts	6
RelaxOrder	1
Cycle 1 NumSweeps	1
Cycle 2 NumSweeps	2
Cycle 3 NumSweeps	2

3.2.3.3. Computation Speed

One of the primary metrics for computational feasibility and utility is execution time. We measured the execution times to calculate the blood flow rate in networks containing between 14 vessels and 17 billion vessels (3 and 33 generations, respectively). Each calculation was repeated 5 times to quantify system variation in execution times caused by resource contention in the cluster of parallel compute nodes. For large vessel networks (*i.e.* $N_{\mathrm{G}} > 25$ and $N_{\mathrm{V}} > 67,000,000$), the volume of data required to describe the network and perform the calculations exceeded the memory of a single compute node (64 GB). The number of

compute nodes, N_{p}, used was

$$N_{\mathrm{p}} = \begin{cases} 1, & \text{for } N_{\mathrm{G}} \leq 25 \\ 2^{(N_{\mathrm{G}}-25)}, & \text{for } N_{\mathrm{G}} > 25 \ . \end{cases} \tag{3.11}$$

The Network Preprocessing application was only required for large networks ($N_{\mathrm{G}} > 25$). Computational execution times were recorded for the Blood Flow Rate Calculations and the Network Preprocessing independently. The total time, vessel-network-partitioning time, vessel-index-renumbering time, junction-renumbering time, and I/O time were recorded for the Network Preprocessing application. For the Blood Flow Rate Calculations, the time to complete the I/O operations, conductance matrix construction, conductance matrix solution, blood flow rate calculation, and total times were recorded. We recorded all times in units of core-hours.

3.2.3.4. Computational Scalability

Weak and strong scaling are two metrics used to evaluate the scalability of an algorithm in high-performance computing [144]. Weak scaling characterizes the execution time as the number of processors increases when the workload per processor is fixed, *e.g.*, as the number of processors doubles so does the workload. Ideally, as the number of processors increases the wall-clock execution time remains constant. We measured the weak scaling for both Network Preprocessing and Blood Flow Rate Calculations. For the Blood Flow Rate Calculations, the smallest size used was $N_{\mathrm{G}} = 25$ on a single compute node and doubled the number of vessels and compute nodes up to $N_{\mathrm{G}} = 33$ on 256 compute nodes. Because the Network Preprocessing was only used for more than one compute node, we started with a problem size of $N_{\mathrm{G}} = 26$ on 2 compute nodes and scaled the application to $N_{\mathrm{G}} = 33$ on 256 compute nodes.

Strong scaling characterizes how adding more processors reduces the computation time of a fixed size problem. This is typically measured using the speedup factor. Ideally, the speedup factor is equal to the ratio of the new number of processors to the original number used for the problem [144]. The strong scaling was determined for Network Preprocessing and Blood Flow Rate Calculations by constructing a fixed-size vessel network ($N_{\mathrm{G}} = 25$) and varying the numbers of compute nodes ($1 \leq N_{\mathrm{p}} \leq 256$). Strong scaling was quantified by calculating the speedup factor (S) according to

$$S = \frac{T_q}{q \cdot T_n} \tag{3.12}$$

where T_q is the time it takes a reference number of processors, q, to complete the calculation and T_n is the time it takes n processors to perform the same calculation. For the Blood Flow Rate Calculations, $q = 1$. The Network Preprocessing codes had a reference number of processors of $q = 2$.

3.2.3.5. Memory Requirements

Memory requirements for Network Preprocessing and Blood Flow Rate Calculations were collected separately using a heap memory profiling tool (Valgrind Massif version 3.13 [100]). We measured the peak and average memory usage for vessel networks with 3, 13, 19, 25, 26 28, 30, and 33 generations. For network sizes requiring two or more compute nodes, the memory used per node was recorded.

3.3. Results

3.3.1. Computational Feasibility

After implementation, the algorithm illustrated in Figure 3.3 successfully calculated the blood flow rates in networks containing up to 17 billion vessels. The blood flow rates calculated using our algorithm (Section 3.2.1.2) matched the results calculated using the validation

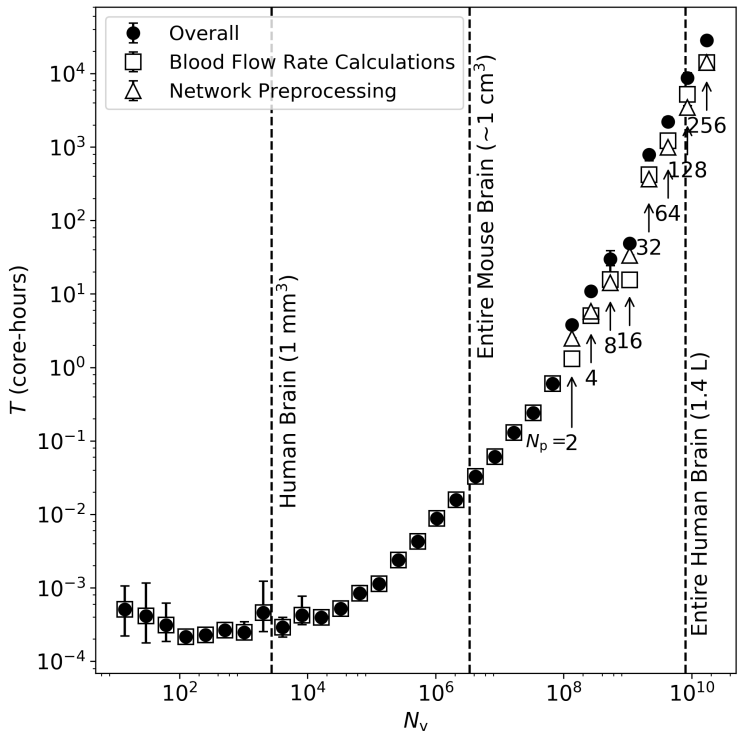


Figure 3.5. Plot of execution time in core-hours (T) versus the number of vessels in a network (N_{V}). The arrows signify the number of compute nodes, (N_{p}), that performed the calculations ($N_{\mathrm{p}} = 1$ if omitted). The error bars represent the range of compute times of the 5 timing runs. For context, the dashed lines represent literature estimates for the number of vessels in the human brain [12] and the mouse brain[102].

method (Equation 3.10). The algorithm completed all computations without errors for all network sizes tested demonstrating that the computations are feasible.

3.3.2. Computational Speed

We measured the execution times of the Network Preprocessing and Blood Flow Rate Calculation applications (Figure 3.5). A 17-billion vessel network (*i.e.*, approximately the number in the human body [58]) required 6.5 wall-clock hours to complete using 256 compute nodes. A 67-million vessel network (*i.e.*, the maximum size that would fit on a single compute node and more than a whole mouse brain) took 1.8 wall-clock minutes on a single compute node. Finally, 4-million vessels (*i.e.*, approximately the number of vessels in a mouse brain [102]) required only 36 wall-clock seconds, also on a single compute node.

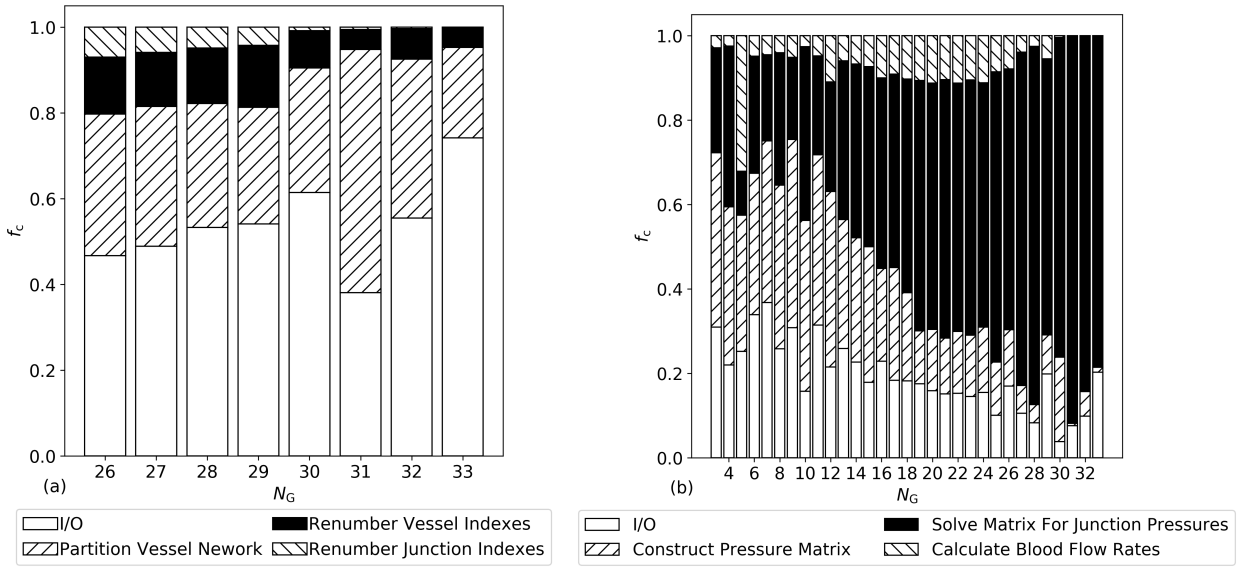


Figure 3.6. Plots of the relative fraction of time (f_c) spent on each task listed in Figure 3.3 verses the number of generations (N_G) in the network. The data for the Network Preprocessing and Blood Flow Rate Calculations are in a and b, respectively. The fraction of time is normalized to each application's total compute time, not the overall algorithm time.

To determine which tasks in the applications were consuming the most amount of time, we computed the relative fraction of time spent performing each task (Figure 3.6). The time of the Network Preprocessing application was predominated by I/O operations at all vessel network sizes. This was attributed to our cluster's high ratio of compute nodes (1-128) to file-system access points (16), leading to communication bottlenecks. Theoretically, a lower-ratio architecture will increase the I/O speed. Figure 3.6 also reveals that solving for the pressure values predominated the execution time in the Blood Flow Rate Calculations, highlighting the large computational expense of solving a sparse matrix. The fraction of time spent on solving the system of equations increased dramatically with the number of vessels. This was partially caused by memory bandwidth limitation inherent to solving sparse matrices. Additionally, the amount of time spent on inter-node communication required to solve the system of equations increased with matrix size.

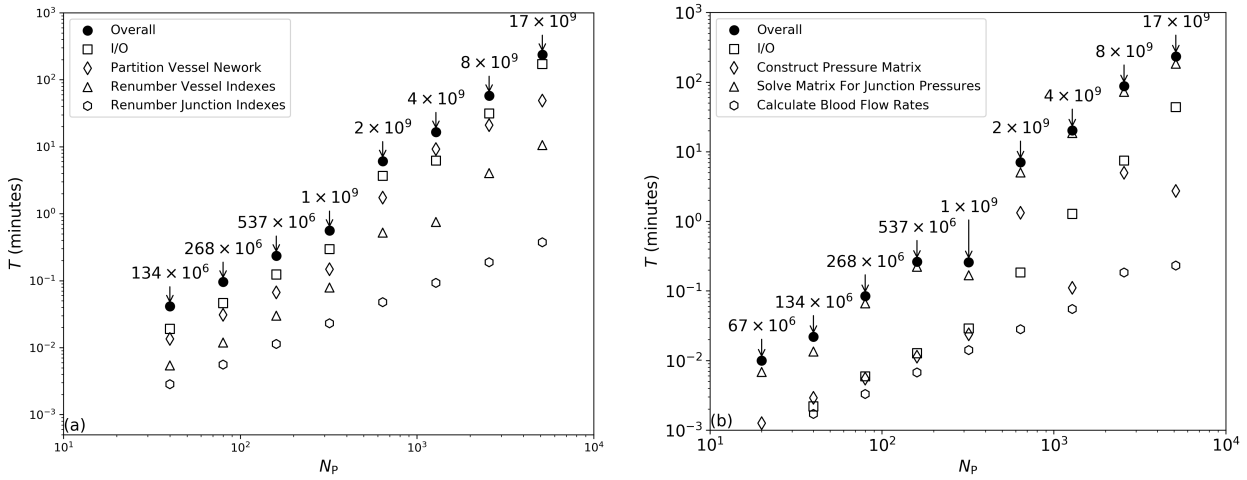


Figure 3.7. Plots of the wall-clock time (T) versus the number of processors (N_p) for Network Preprocessing (a) and Blood Flow Rate Calculations (b). The dashed line represents perfect scalability. The dark circles represent the overall scalability of Network Preprocessing. Arrows specify the number of vessels in the problem. The labels denote the number of vessels in the vascular network.

3.3.3. Computational Scalability

To quantify the scalability of the algorithm, we measured both the weak (Figure 3.7) and strong scaling (Figure 3.8). The weak scaling data shows that both applications scaled well when the number of processors was below 640 (32 compute nodes). The reasons each application started performing poorly after 32 compute nodes differed. The Network Preprocessing was limited by the cluster architecture. The application was restricted to 16 I/O channels leading to increasing contention as more processors are added, and thus the execution time increased. This conclusion is further supported by the strong scaling data because the computational portions of the algorithm scale extremely well with the number of processors for the fixed problem size used.

The Blood Flow solving algorithm is limited by the matrix solving algorithm. The lack of scalability in the matrix solver can be attributed to two primary culprits, increasing inter-process communication with increasing compute nodes and memory use as the problem size increased. The effects of inter-process communication are evident in the strong scaling data (Figure 3.8b) because as more processors are utilized the ratio of computation to communi-

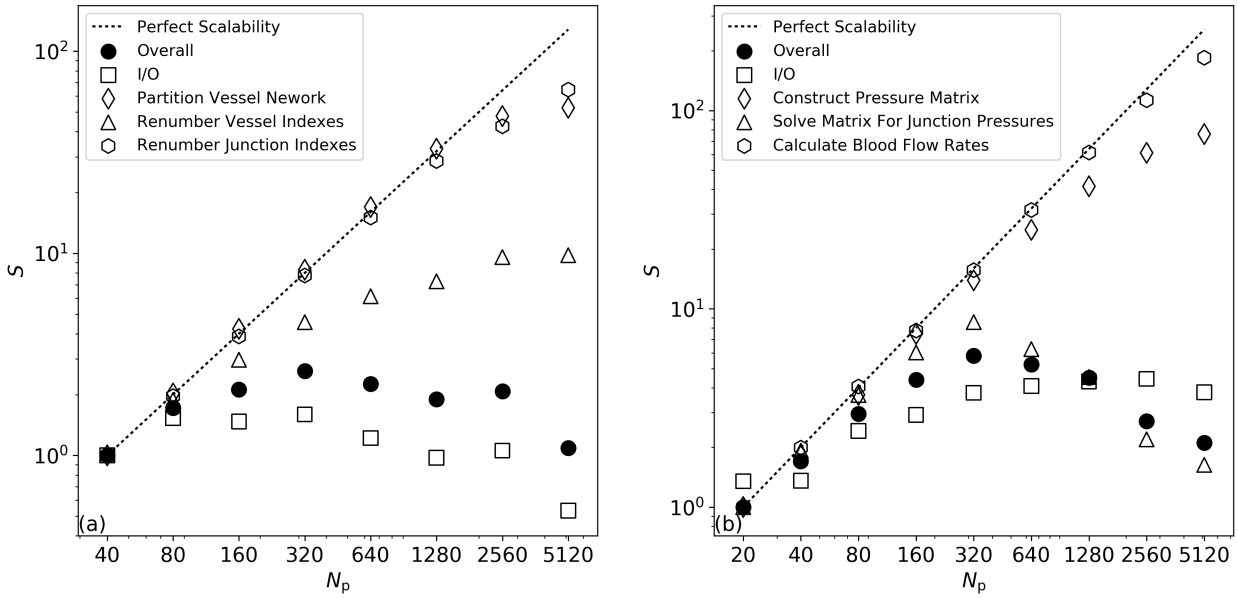


Figure 3.8. Plots of the speedup factor (S) versus the number of processors (N_p) for Network Preprocessing (a) and Blood Flow Rate Calculations (b). The dashed line represents perfect scalability. The dark circles represent the overall scalability of Network Preprocessing (a) or Blood Flow Rate Calculations (b).

cation decreases. In the weak scaling measurements (Figure 3.7b), the problem size increased as the number of processors was increased, reducing the impact of this effect. However, we see that a majority of the time was still spent on solving the matrix. This is because solving sparse matrices is a memory bandwidth limited problem. As the problem size grew, more memory bandwidth was utilized to calculate the matrix elements on each compute node.

3.3.4. Memory Usage

Figure 3.9 plots peak and average memory usage per node for the applications. The Network Preprocessing application remained well below the 62 GB available on each node. The Blood Flow Calculation required a peak of about 68 GB which occured during the matrix solving. This peak is above the available memory per node and could result in performance degradation due to the use of the hard disk as a virtual memory device.

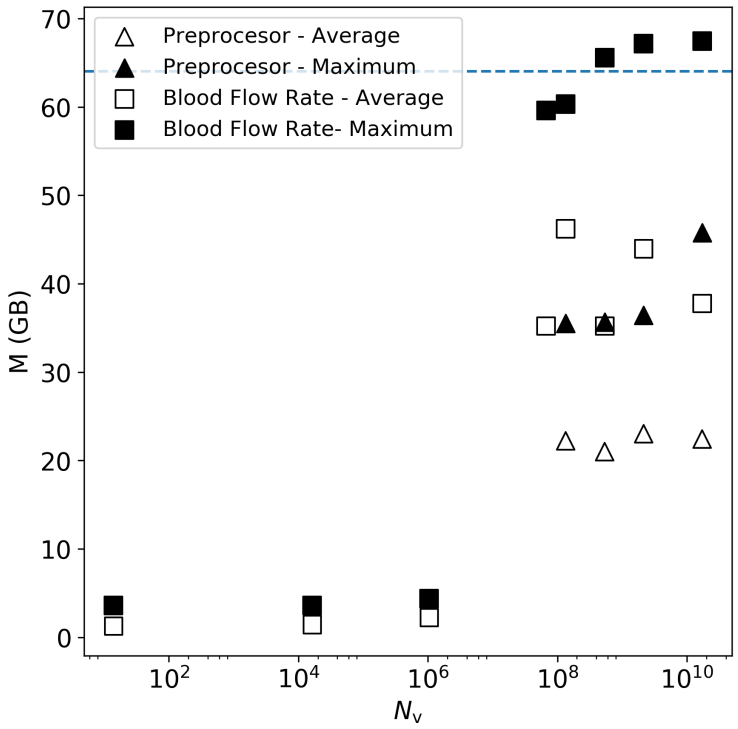


Figure 3.9. Plot of memory usage (M) versus the number of vessels (N_V). The dashed line marks the maximum available memory on a single compute node (64 GB).

3.4. Discussion

In this study, we developed a two-step algorithm to compute the blood flow rate through a vascular network corresponding in size to that of the human body. The computational characteristics of the algorithm were measured, including execution time, scalability, and memory usage, were measured. We successfully demonstrated, for the first time, that whole-body blood-flow rate calculations are feasible with a contemporary high-performance computing system. To simulate a network with 17 billion vessels, 6.5 hours of wall-clock time was required using 256 compute nodes (5120 processors). The algorithm scaled modestly as the number of processors increased, primarily due to limitations of the matrix solver and I/O operations. In both stages of the algorithm, the average memory consumption per node was 56 GB, *i.e.* less than the 64 GB available. Finally, the results of this work show that 4 million vessels, the approximated size of a mouse brain [102], executed in only 36 seconds on a single compute node.

The primary implication of this study is that new avenues of inquiry are possible that could elucidate blood-flow problems that were hitherto considered intractable. Several areas of medical research may find applications for this new capability, including the simulation and evaluation of injury, healing, disease, and drug delivery. In all of these applications, knowledge of the blood flow in individual vessels has the potential to improve the accuracy of physiological simulations, provide new insights, and, ultimately, to improve outcomes for patients. For example, many drugs are delivered to patients intravenously and the bloodstream transports these drugs throughout the body. A whole-body model of vasculature could, in theory, enable more accurate localization of the drug within each organ than would be possible with compartmental modeling for example. Our model may also be used to validate the findings of compartmental models, as in Hyde *et al.* [126], but in the much larger and complete vascular networks of whole organs. Another implication is that a whole-body model, even a simple one, could be used to improve boundary condition estimations needed for more advanced blood-flow simulation techniques. This would alleviate the need for cyclic boundary conditions or relax simplifying assumptions currently used. Finally, using a whole-body model of blood flow could facilitate a more complete interpretation of data from existing diagnostic imaging and nuclear medicine procedures. For example, blood flow plays a critical role in generating the signal in functional MRI [145]. This imaging procedure measures the changes in blood flow in the brain and correlates those changes in different regions to brain activity. Being able to model blood flow though the entire brain may enable researchers to draw more robust conclusions from their data by reducing the impact of confounding changes in blood flow caused by the connected nature of a vascular network.

There are three important characteristics of blood flow models, namely the extent of the vascular domain simulated, biophysical realism, and computational performance. In the previous literature, the largest network used in a blood flow simulation was approximately 256,000 vessels, representing a $27\,\mathrm{mm}^3$ functional sub-unit of the human brain [21]. Our model is capable of calculating the blood flow rates through networks containing up to 17

billion vessels, but with a much simpler geometry. Realism of a blood flow simulation can be broken down into geometric and physical categories. Our model is comparable in physical realism to many steady-state models of blood flow [21, 126, 146, 147], which are used for their simplicity and scalability. 3-D finite element simulations provide superior geometric and physical realism [27, 56, 104, 119, 122]. One important feature of finite element simulations is the modeling of the blood's momentum, enabling accurate modeling of the second-order effects of bifurcations on blood flow rate. Another important characteristic of time-dependent blood flow rates in the arterial network is cardiac pulsation. Our model intentionally omitted these features so that we could expand the performance envelope to 17 billion vessels, where previously reported time-dependent and time-independent models reached less than 300,000 vessels. Similarly, our geometry was less realistic than that modeled in the finite-element and steady-state studies previously performed. This was done to probe the performance envelope for billions of vessels. Finally, our work revealed important insights regarding the computational-performance aspects of blood flow rate calculations. Our literature search found only two previous studies that provided any details on execution times [104, 119], and none reported memory requirements. Neither study provided sufficient details to allow a direct comparison of execution times. One reason for this is that the previous studies used time-dependent models of the blood flow rate, but neglected to provide the number of iterations required to solve the system of partial differential equations. The work performed by our algorithm is similar to the inner-most loop of a time-dependent simulation, where a matrix must be constructed and solved. Thus, the timing data reported in our study could be used to estimate a lower bound of the compute time for whole-body blood flow simulations using time-dependent models.

To our knowledge, this is the first study to systematically explore the computational requirements to calculate the blood flow rate through each vessel in the entire human body. Using the simplified vessel geometry and a steady-state blood flow model, we were able to provide an estimate of the minimum computational resources needed to perform such

calculations. This information can be used to inform future research study designs. Another strength was the use of the Poiseuille equation in representing the blood flow. It minimized the computational complexity of the problem, making the calculations feasible and results reproducible. A further strength of this work was the use of realistic lengths and radii for the vessel lumens in our vascular geometry. This provided a suitable test of the matrix solver's ability for resolving dimensions spanning 5 orders of magnitude. Finally, the modular design of our algorithm allows for the integration of more accurate geometries with the steady-state blood flow model. In theory, our algorithm could readily accommodate more realistic vessel geometries, such as those taken from imaging studies [17, 77, 148].

One limitation of this work is that our vessel geometry comprised straight vessels that were rigid and connected with fixed bifurcation angles. This vessel network geometry is the only currently known geometry that can be scaled to billions of vessels; a requirement for charting the performance envelope at the whole-body scale. It was more important to model the correct number of vessels for the computational feasibility, than it is to model an accurate geometry for this study. The differences in the accuracy of this geometry will be addressed as part of a future research study. Another limitation of this work is that the blood flow model neglected second-order effects, such as blood momentum, red blood cell skimming, and vessel compliance [46]. This limitation could be addressed by implementing a 1-D time dependent blood flow model, which is currently being investigated by our group.

This study has shown that, for the first time, it is computationally feasible to compute blood flow rates through each vessel in the entire human body with current computing technology. These results could enable improvements in many fields of research in the life sciences and provide critical computational insights about the blood flow in whole organisms and organs.

4. Accuracy of a Steady-State Model of Blood Flow Through a Vasculature Containing 126 Vessels

4.1. Introduction

Approximately 775,000 deaths in the United States were caused by cardiovascular and cerebrovascular diseases during 2015 [115]. It is expected that the treatment of these diseases will cost over $1 trillion per year by 2030 [116]. The study of vascular diseases is complicated by the connected nature of the human circulatory system. Because it is a closed system, localized changes to a vessel can cause non-local changes in blood flow [149, 150]. While multiple diagnostic techniques exist to measure changes in blood flow, they are limited to macroscopic domains [151–153] Simulations, in contrast, can predict macroscopic and microscopic blood flow.

The simulation of fluid dynamics is a mature field, with an extensive body of literature [154–157]. When simulating blood flow, the gold-standard method is solving the time-dependent 3-Dimensional (3-D) Naiver-Stokes Equations [46]. They can model interactions of blood with vessel walls, red-blood-cell transport, [158], stroke processes [50, 51], and stent safety and effectiveness [49, 118]. Theoretically, the Naiver-Stokes Equations are applied to all points in a vascular geometry, resulting in a system of differential equations. In practice however, the vascular geometry is descretized using a finite-element mesh [159]. As the size of the simulation domain grows, the number of elements increases exponentially. This typically limits the size of the vascular domain that can be simulated to approximately 50 vessels [54, 55]). By assuming the blood flow is time-independent and laminar, the Naiver-Stokes Equations may be simplified, which increases the number of vessels that can be simulated. This technique has been used to model whole hearts [126], a 27 mm^3 region of the human brain [21], and 17 billion vessels (Chapter 3) [160], which approaches the number in the human body [58]. The latter work focused on computational feasibility of the algorithm however and the accuracy of the blood-flow calculations was not reported.

The goal of this work was to quantify the accuracy of the steady-state blood flow model used to calculate blood flow through 17 billion vessels. We achieved this by comparing the results of the LSU algorithm of Donahue *et al.* [160] (Chapter 3) to that of a previously-validated computational-fluid-dynamics (CFD) package. We compared blood flow rates through networks with 6 to 126 blood vessels. We investigated the model's ability to simulate the effects of vessel occlusion on blood flow by blocking a single vessel in a 126-vessel network and comparing the results to those of the commercial package.

4.2. Methods

4.2.1. LSU Steady-State Model for Blood Flow

The LSU model used steady-state fluid dynamics to calculate the blood flow rate in each vessel using the methods of Donahue *et al.* [160] (Chapter 3). For the reader's convenience, we will summarize the methods here. The governing equation in this model was the Poiseuille's equation, or

$$Q = \frac{\pi r^4}{8\eta L}\left(P_{\text{in}} - P_{\text{out}}\right) \tag{4.1}$$

where Q is the blood flow rate, r is the vessel's radius, L is the vessel's length, η is the viscosity of the fluid, and P_{in} and P_{out} are the pressures at the input and output of the vessel, respectively [21]. Blood was assumed to be incompressibile. Therefore, at any point the blood entering must equal the blood leaving, or

$$\sum_{i=0}^{3} Q_i = 0 \tag{4.2}$$

where Q_i is the blood flow rate in the i^{th} vessel. To model the effect of red blood cells on the viscosity of the blood, the LSU Model used an empirical model for *in-vitro* viscosity for a constant hematocrit of 0.45, or

$$\eta_{0.45} = \eta_{\text{abs}}\left(220 \cdot e^{-2.6r} + 3.2 - 2.44 \cdot e^{-0.06(2r)^{0.645}}\right) \tag{4.3}$$

where r is the radius of the vessel in mm [127]. The absolute viscosity (η_{abs}) of the blood was taken as 36 mPa s [133].

We calculated the blood flow rate of blood in in two steps. First, Equation 4.1 was applied to the vessels and Equation 4.2 to the bifurcations. The resulting system of linear equations was solved using iterative Krylov methods to determine the pressure at each bifurcation [134, 135, 141, 142]. Lastly, we calculated the blood flow rate in each vessel using Equation 4.1.

4.2.2. Computational Fluid Dynamics Model

To test the accuracy of the LSU model we compared its calculated blood flow rates to those of a commercial computational-fluid-dynamics (CFD) package (Autodesk CFD 2018, San Rafael, CA, USA). This package was previously validated [161, 162]. The simulated vessels contained blood and were embedded in human tissue. The physical properties, such as density, elasticity, and viscosity, of these materials were specified in the material library provided with the CFD package. Automated tools, included in the CFD package, were used to create the mesh for each geometry. Table 4.1 lists the settings we used for mesh generation.

We used the default solver in the CFD package to perform the blood-flow-rate calculations. We required that the solver perform a laminar, steady-state simulation, in accordance with the assumptions in the LSU-model calculation and Equation 4.1 (see Section 4.2.1). The Reynolds number [154] was less than 100 for all networks simulated according to calculations performed by the CFD package. Table 4.2 lists the solver parameters we selected for the CFD package.

4.2.3. Vascular Geometries

4.2.3.1. General Overview

We used two different vascular network configurations to evaluate the accuracy of the LSU blood flow rate model. The first configuration consisted of doubly, (*i.e.* healthy) net-

Table 4.1. Settings used for mesh generation in the CFD package. Settings not listed were set to the default values.

Diagnostics	
Minimum Refinement Length	0.0025 mm
Wall Layer	
Number of Layers	3
Layer Factor	0.2
Layer gradation	Auto
Advanced	
Resolution Factor	0.9
Edge Growth Rate	2
Minimum Points on Edge	3
Points on Longest Edge	10
Volume Growth Rate	2
Surface Growth Rate	2
Enhancement Growth Rate	2
Refinement length	0 or minimum
Fluid Gap Elements	3

works. This allowed us to evaluate the accuracy of the LSU model under simple conditions. The second configuration was an asymmetric network (*i.e.*, a single vessel occluded). This allowed testing the accuracy of the LSU model in slightly-more complex geometries. Both configurations were created in a 2-D plane to simplify visualization of the resulting blood flow rates.

4.2.3.2. Vascular Network Configurations

The healthy-network configurations comprised between 6 and 126 vessels (2-6 generations of vessels) in a fractal pattern that was symmetric about the x and y axes (Figure 4.1). Each vessel in a network was a right cylindrical tube connecting one or two junctions. The network had a single inlet and a single outlet, which resulted in two boundary conditions. The smallest vessels were 2.5 µm in radius and 55 µm long, representing nominal capillary dimensions [12]. Between each generation, the vessel radius was reduced by $2^{-1/3}$ following

Table 4.2. Settings for the CFD solver. Default values were used for any settings not explicitly listed.

Automatic Convergence Control	
Instantaneous Convergence Curve Slope	1×10^{-13}
Time-Average Convergence Curve Slope	1×10^{-13}
Time-Average Convergence Curve Concavity	1×10^{-13}
Field Fluctuations	1×10^{-13}
Maximum Iterations	1500
Physics	
Compressibility	Incompressible
Turbulence	Laminar
Automatic Mesh Adaptation	
Number of Cycles	3
Growth Rate	2
Boundary Layer Growth	1.1
Refinement Limit	0.001
Resolution Factor	0.9

Murray's Law for bifurcation [93] and the length was reduced by 0.8 to avoid vessel overlap for visualization [132].

The damaged-network configuration was based upon the 126-vessel network (Figure 4.1) but with a single vessel occluded. Three different network geometries were created, each with a different vessel occluded (Figure 4.2).

4.2.3.3. Creating the Geometric Descriptions

The description of the vascular geometry for the LSU algorithm was constructed with a fractal-based algorithm (Chapter 2) [132]. Each blood vessels was defined by its start point, end point, and inner radius. The geometric description also included information about connections between vessels and bifurcations in the network. Occluded vessels were represented by setting the radius to zero. The resulting geometric description of the network was directly imported into the LSU model of blood flow.

The CFD package required that the geometry created by the fractal method be converted into a discrete 3-D model, where each vessel was represented as a tube. This was done

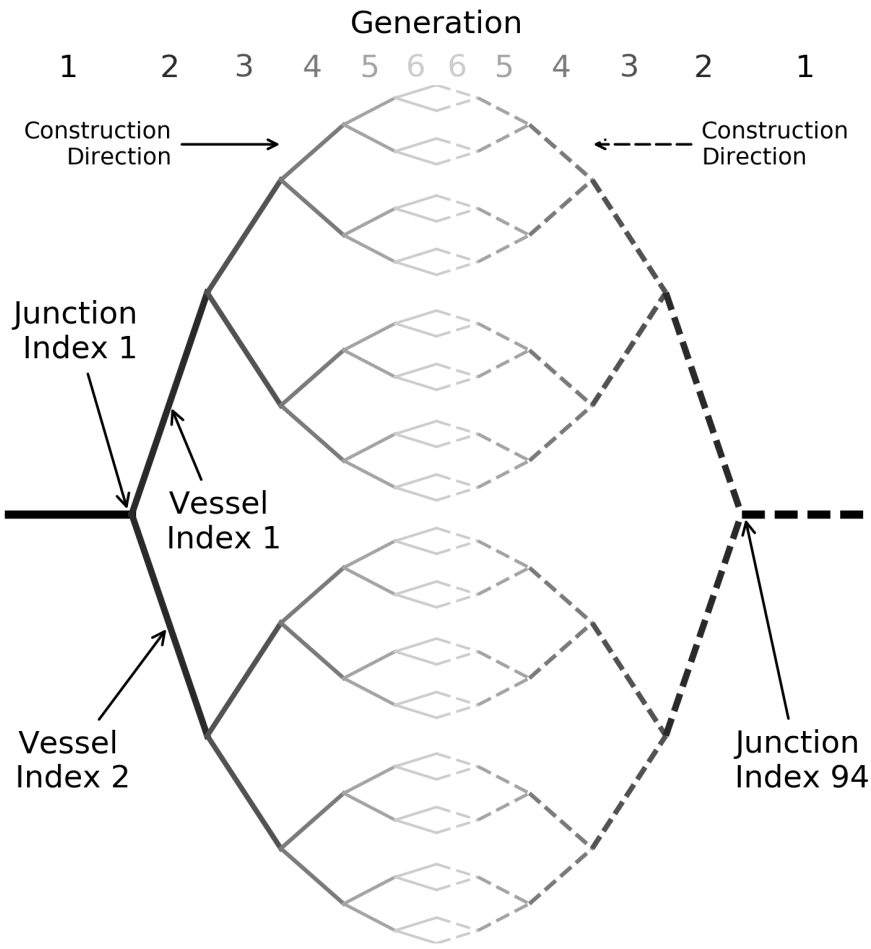


Figure 4.1. An example of a 2-D vascular network containing 126 vessels, created with fractal methods. The left half is considered the arterial tree (solid), while the right side is the venous tree (dashed).

using a commercial computer-aided design (CAD) package (Inventor 2018, Autodesk, San Rafael, CA, USA). First, we imported the bifurcation locations and vessel radii from the LSU model geometry. Next, linearly-tapered vessels were constructed between junctions. Tapering vessels avoided nonphysical sharp edges at the bifurcations in the CFD model. The initial radius of each vessel matched the radius of the corresponding generation in the LSU model while the final radius was the radius of the next generation. The vessels in the final generation had a constant radius of 2.5 µm. We performed this conversion manually for

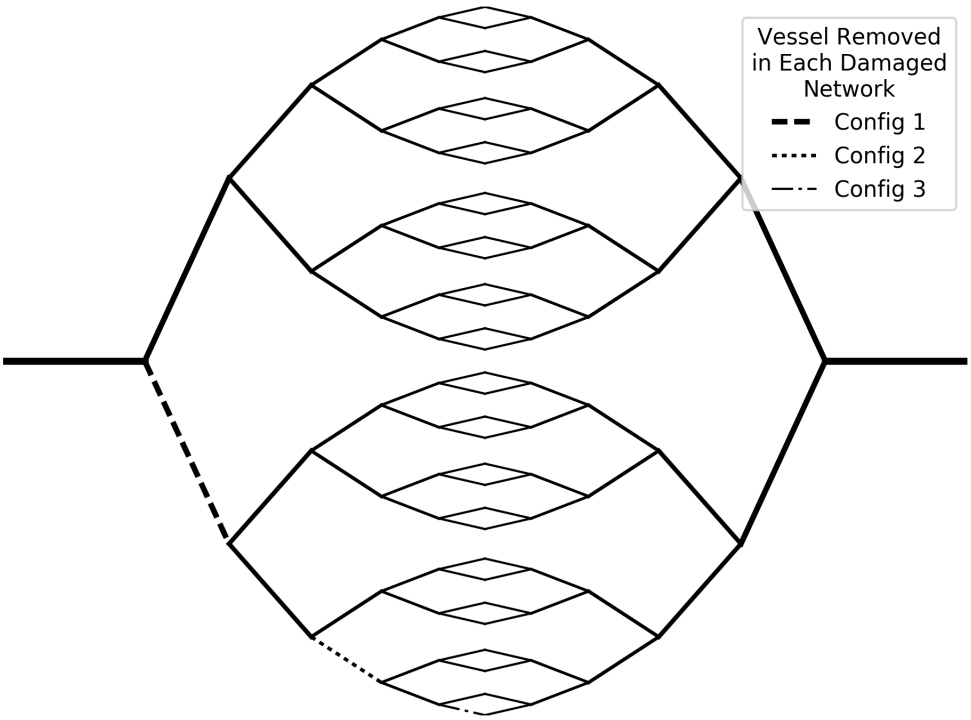


Figure 4.2. Illustration of vessels that were removed from the 126-vessel network (Figure 4.1)to create the 3 different damaged-network configurations.

all of the healthy and damaged networks. Occluded vessels were completely removed from the geometric representation in the CAD model.

4.2.4. Computational Environment

All calculations performed in the study were completed using a personal computer. The system had a 4-core 4.1 GHz CPU (Core i5-3570K, Intel Corporation, Intel Corporation, Santa Clara, CA, United States) and 32 GB of memory. The operating system was Windows 10 (Microsoft, Redmond, Washington, United States).

4.2.5. Evaluation of the LSU Blood-Flow Model Accuracy

To quantify the accuracy of the LSU model we compared its calculations of blood flow rate to that predicted by the CFD package. The LSU model calculated a single value of the

blood flow rate through each vessel For the CFD package, we recorded the blood flow rate at the at the middle of each vessel perpendicular to the blood flow.

The boundary condition at the inlet was 133 kPa blood pressure and the outlet boundary condition was

$$Q_0(N_\mathrm{G}) = \frac{N_\mathrm{V} + 2}{4} \cdot v_\mathrm{c} \cdot \pi r_\mathrm{c}^2 \tag{4.4}$$

where $Q_0(N_\mathrm{G})$ is the blood flow rate at the outlet, N_V is the number of vessels in the healthy network, v_c is the blood velocity in a capillary, and r_c is the minimum capillary radius (see Section 4.2.3.2). We assigned v_c the value of $1\,\mathrm{mm\,s^{-1}}$ based of the findings of Ivanov *et al.* [138].

Theoretically, blood vessels directly downstream of an occluded vessel have a zero flow rate. The iterative solvers used in both the LSU model and the CFD package reported negligible but non-zero values in these vessels that were caused by limitations in machine precision (*e.g.*, rounding and truncation errors). These non-zero values produced artifacts in the analysis. To overcome this, we set any blood flow rate less than $1\,\mathrm{\mu m^3\,s^{-1}}$ to zero. This value is is almost 20,000 times less than the theoretical minimum flow rate in the network ($1.9 \times 10^4\ \mathrm{\mu m^3\,s^{-1}}$).

We compared the blood flow rates calculated by the LSU model and the CFD package in each vessel. We recorded the the unsigned absolute ($|\Delta|$) and relative ($|\Delta_r|$) differences in blood flow reported by the two calculation methods.

4.3. Results

4.3.1. Evaluation of the LSU Blood-Flow Model Accuracy

We compared the blood flow rates predicted by the LSU model and CFD package for small networks (6-126 vessels). Figure 4.3 plots a representative example of the differences in blood flow rates through both healthy and damaged networks calculated by the LSU model. The CFD package produced similar qualitative results. This shows blood being redistributed through the network after a vessel had been removed from the 4^{th} generation.

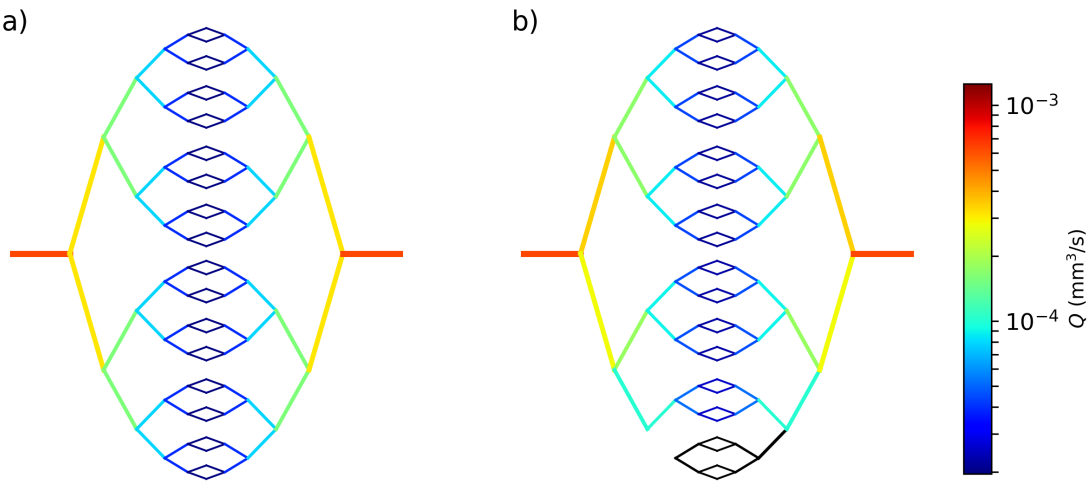
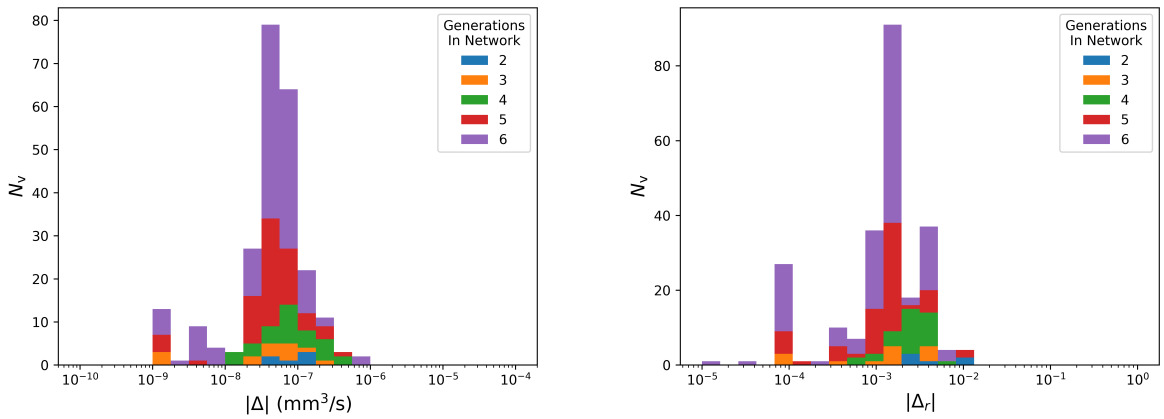


Figure 4.3. Plot of the blood flow rates in a 6 generation network before (a) and after (b) removal of a vessel in the 4^{th} generation was removed. The lines represent the vessels, with the color denoting the flow rate and line weight their relative radii. The black lines represent vessels that were occluded or have no flow.

Figure 4.4a shows a histogram of the absolute differences in blood flow rate reported by the LSU model compared to the CFD package for the healthy networks. This figure shows that the distribution for all networks considered is centered around 2×10^{-8} mm^3 s^{-1}, with a maximum error less than 7×10^{-7} mm^3 s^{-1}. Figure 4.4b plots the relative difference in blood flow rate between the CFD package and the LSU model for healthy networks. The relative discrepancy was less than 0.85%. This provides confidence in the LSU model's implementation and the accuracy of the calculated steady-state laminar blood flow rates.

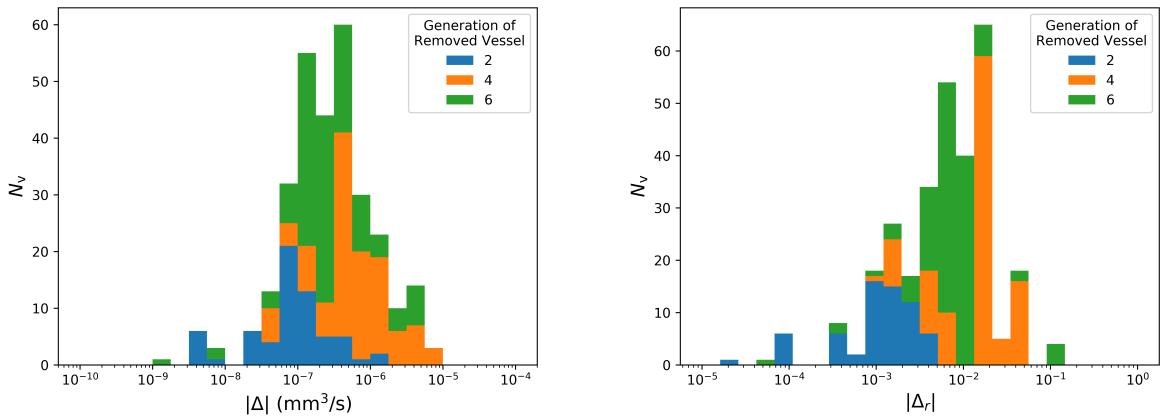
Figure 4.5a shows a histogram of the absolute differences between the LSU model and the CFD package for the damaged networks. Overall, the LSU model predicted the blood flow rates with moderate accuracy. The maximum unsigned difference for all three networks analyzed was less than 7×10^{-6} mm^3 s^{-1}, while the average signed difference was 5.3×10^{-7} mm^3 s^{-1}. Figure 4.5b plots a histogram of the relative difference between the CFD package and the LSU model for damaged networks. Here, the relative difference was less than 13% and the root-mean-square error was 1.9%. This relative accuracy is acceptable based on the assumptions made in the steady-state model.



(a) (b)

Figure 4.4. Histogram of the number of vessels (N_V) versus (a) unsigned absolute ($|\Delta|$) and (b) relative ($|\Delta_r|$) differences between the commercial CFD package and the LSU model for healthy networks. The colors break down the data by the number of generations in each network.

There are two main reasons for the larger differences in the damaged networks compared to the healthy networks. First, the LSU model under-predicted the blood flow rate near the occluded vessel and the amount of under-prediction decreased with increasing distance from the occluded vessel. This was caused in part by the tapered vessels in the CFD model, as opposed to the constant radius of the vessels in the LSU model. This resulted in a smaller average radius for each vessel than in the LSU model and a decreased flow (see Equation 4.1). Second, the CFD package included more physical realism than the LSU model, especially when modeling the effects of junctions and sharp turns on the blood flow rate. This enabled the CFD model to more accurately handle these effects, making it more accurate in the redistribution of blood flow. This effect is more pronounced in the damaged networks than the healthy networks because the vessel occlusion resulted in a sharp bend in the network geometry. Figure 4.6 illustrates the differences in blood flow rate compared to healthy networks.



(a) (b)

Figure 4.5. Histogram of the number of vessels (N_V) versus (a) unsigned absolute ($|\Delta|$) and (b) relative ($|\Delta_r|$) between the commercial CFD package and the LSU model for damaged networks. The colors break down the data by generation of the removed vessel.

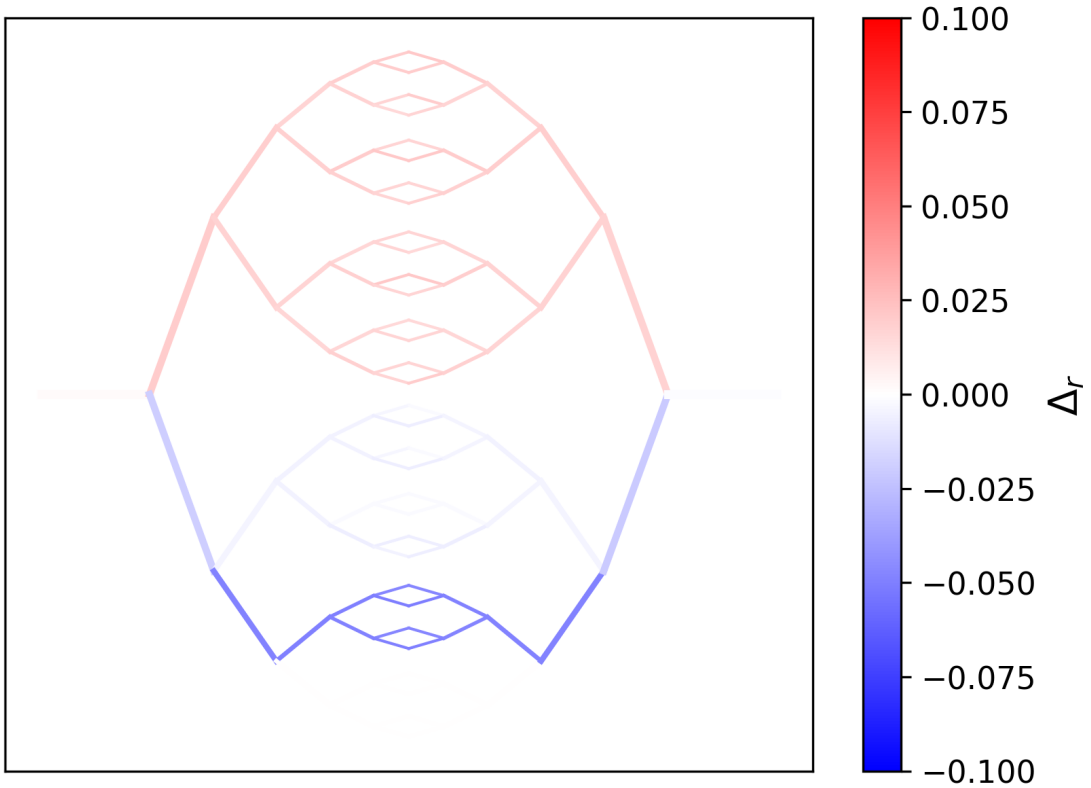


Figure 4.6. Relative difference Δ_r between the LSU model and the CFD package calculations for the network with a vessel removed in the 4th generation.

4.4. Discussion

We compared the blood flow rate calculations of the LSU model to those of a previously validated, commercial CFD package to quantify the accuracy of the LSU model. The major finding was that the methods calculated blood flow rates with a maximum difference less than 1% for symmetric networks, suggesting that the LSU model has comparable accuracy to the CFD package under the conditions considered. Unlike the CFD package, however, the LSU model can be used to calculate the blood flow in billions of vessels (*i.e.,* in vasculatures similar in size to that of the human body). Furthermore, in networks with an occluded vessel, the LSU model agreed with the CFD package within 13% and an RMS error of 2%. These results provide a quasi-independent verification of the LSU model, strongly suggesting that it is free of major systematic errors. Together with previous results (Chapter 3) [160], these works suggest that the simulation of blood flow through the entire body is not only computationally feasible, but also sufficiently accurate for present intents and purposes.

The LSU model of blood flow studied in this work could open new avenues of research in the study of vascular damage. One potential application of this is in the study of the systemic impact of localized or whole-organ vascular changes, *e.g.*, those caused by trauma, radiation, or surgery. Many studies on blood flow focus on the pulsatile arterial flow and truncate their networks before the capillary level. This leads to the application of boundary conditions at the unconnected ends of the vessel that are based on assumptions regarding the status of vasculature not modeled. The steady-state flow simulated in the LSU model is a good approximation of the blood flow through capillaries and veins [119, 121]. The LSU model could be used to simulate the vascular flow through these, reducing the dependence on the assumed value of boundary conditions. Another potential application is to investigate the kinetics of systemic therapies, which are typically transported by the blood. A whole-body model of blood flow could be used to investigate how a drug passes through the various organs of the body in vessel-by-vessel detail. This could provide statistical approaches to predict the distribution of a drug throughout a tissue at a microscopic level.

The accuracy of blood flow simulations is a well studied topic. Sun *et al.* [163] reported that their 3-D Navier-Stokes model of blood flow rate differed from experimental measurements by less than 11%, which is comparable to the maximum relative difference observed in this work. Another study observed a 1% average difference between their 3-D Navier-Stokes model and experimental results [164], which is on the order of the maximum relative difference seen for the healthy networks in this study. Our work is also comparable to the results published by Radaelli *et al.* [165], who compared 7 different models of blood flow through stents, each created by independent research groups. They reported a 5% difference in the predicted blood flow rate across all the models tested. This is similar to RMS error observed between the LSU model and the CFD package (1.9%), lending confidence to the calculations performed by the LSU model. Many works compare the results blood flow algorithms to laboratory or clinical measurements [121, 126, 163, 166]. This differs from our computational approach to validation, making a direct comparison of results difficult.

The primary strength of this work was the use of a commercially available and validated CFD package. This enabled us to evaluate the accuracy of the LSU model using a systematic method that was reproducible. This allowed us to investigate the causes of the difference between the two models, such as the rounding and truncation errors found in the damaged networks. An additional strength was the calculation of accuracy in 6 different sizes of symmetric networks. This allowed us to quanitify how the error changed as the number of vessels in the network increased. Another strength of this work was the testing of both symmetric and asymmetric networks. Vascular networks in the human body are not symmetric in nature. By studying the extreme case of a completely occluded of vessel, we were able to put an upper bound on the error of asymmetries in the network geometry.

The major limitation of this work was the modest sizes of vasculatures studied. This limitation was caused by the computational complexity of 3-D finite element solutions to the Naiver-Stokes equations used in the CFD package. An additional limitation was the simplified physics implemented in the LSU model, which enables calculations of blood flow

in 17 billion vessels. In principle, these major limitations can be overcome with additional research and computing power.

The accuracy of the LSU model was 1% for symmetric (healthy) networks and 13% for damaged (asymmetric) networks. The LSU model appears adequate to study the distant effects of network damage in individual vessels. These techniques may open up new fields of research in systemic therapy and trauma response simulations.

5. Analytical Model for Ion Stopping Power and Range in the Therapeutic Energy Interval for Beams of Hydrogen and Heavier Ions

5.1. Introduction

Coulomb force interactions are the primary energy loss mechanisms for hydrogen and heavier ions penetrating matter. The total stopping power, or an ion's energy loss per unit pathlength, comprises energy loss from coulombic interactions of the ion with the atomic nucleus of the target (nuclear stopping power) and with the target's atomic electrons (electronic stopping power). The average pathlength traveled by a beam of ions coming to rest is known as its penetration range. Stopping power and range data are commonly needed for applications such as radiation therapy, radiation protection, ion implantation, and accelerator physics, as well as in basic sciences such as nuclear and particle physics.

The literature on stopping powers and ranges for hydrogen and heavier ions is vast, including measurements, theory, evaluations and tabulations, and software algorithms and programs [168–171]. Electronic stopping powers are typically calculated with either the Bethe-Bloch equation or empirical functions, depending on the energy regime of the ion. Nuclear stopping powers are typically calculated using a theoretical approach. Ranges are commonly calculated numerically using pathlength integration, stopping power data and the continuous slowing down approximation (CSDA). The Stopping and Ranges of Ions in Matter (SRIM) by Ziegler and Biersack [172], which provides these and other capabilities, exemplifies the integration of highly-realistic theories, evaluated data, and radiation transport capabilities. In applications where speed and simplicity are paramount considerations, analytical calculations of range offer considerable advantages over numerical methods.

Range has long been calculated analytically using a simple power law known as the range-energy relationship, or Bragg-Kleeman Rule [173]. This method is accurate for a range of ion

Adapted with permission from: W. Donahue *et al.*, "Analytical model for ion stopping power and range in the therapeutic energy interval for beams of hydrogen and heavier ions", Physics in Medicine and Biology **61**, 6570–6584 (2016)

species (e.g. hydrogen, carbon), ion energies (e.g. 10-200 MeV u^{-1}), absorber compositions (e.g. plastic, water, metal), and thicknesses (e.g. 0.5-110 mm) [174, 175]. The major limitation of formulas derived from a simple power law relation was their limited energy interval of applicability, i.e., 10-200 MeV for hydrogen ions in water. Thus, for a projectile in a target large enough to completely stop it, such as a cancer patient, the simple power law model for stopping power breaks down in the biologically important interval below 10 MeV. This energy corresponds to the last 1 mm of a proton's range in water, which is typically not a concern in current standard-of-care treatment planning, because the treatment plan is calculated on a spatial grid similar to the planning image resolution (1 mm). However, there is considerable interest in performing multi-scale calculations of ion transport e.g., organs (cm), image voxels (mm), and cells (m). Thus, there is an increasing need for new fast and simple formulae to calculate range and stopping power values over wider intervals of ion energy. It was not known if simple analytical equations would meet this need and, if so, the limits of their applicability.

The aim of this work was to develop a simple, broadly applicable continuous model of total stopping power which could be analytically integrated to provide a simple formula for calculating range. More specifically, we sought to develop models with accuracy of 10% in stopping power and 1 mm in range. To accomplish this, we extended the simple power-law approach, yielding new analytical formulae for range and stopping power that is applicable to a wider energy interval. The accuracy was determined by comparison with the best-available evaluated stopping power and range data available from the literature for hydrogen, carbon, iron, and uranium projectile ions and target materials including water, carbon, aluminum, lead, copper, iron, gadolinium, and gold.

5.2. Methods

5.2.1. State-of-the-Art Theories for Stopping Power and Projected Range

The most widely used method for calculating stopping power and range was developed by Ziegler and colleagues over the last four decades. They wrote a computer program to

conveniently calculate the Stopping and Range of Ions in Matter (SRIM) [172]. SRIM is actively supported by Ziegler and colleagues, who provide annual updates to the program based on evaluations of new measurement data and advances in stopping theory. In this section, for the reader's convenience, we will provide a brief overview of the methods used in the SRIM code, which is representative of methods used for stopping power and calculations in many basic and applied fields.

Total stopping power, S, is given by

$$S = S_n + S_e \tag{5.1}$$

where S_n is the nuclear stopping power and S_e is the electronic stopping power. Ziegler and Biersack [172] developed a universal nuclear stopping theory for the calculation of the nuclear stopping power for various ion species, ion energies, and targets. It is given by

$$S_n = \frac{d\ Z_2\ z_1\ M_1\ S_n(\epsilon)}{(M_1 + M_2)(z_1^{0.23} + Z_2^{0.23})} \tag{5.2}$$

where, the function $S_n(\epsilon)$ is the reduced nuclear stopping power, ϵ is the unitless reduced energy, and d is a scaling coefficient [172]. z_1 is the charge of the ion and M_1 is the mass of the ion. Z_2 and M_2 are the corresponding quantities for the target. These nuclear stopping powers have been shown to match closely with measurements [172].

Electronic stopping powers are calculated differently depending on the energy regime. For hydrogen ions with energies above 1 MeV/nucleon (defined as high energy), SRIM employs the Bethe-Bloch formula with corrections or

$$S_{e,high} = \frac{4\pi r_0^2\ N_A\ \rho\ m\ e\ c^2\ z_1^2\ Z_2}{M_2} \frac{1}{\beta^2} \left[\ln \frac{2m_e v^2}{1-\beta^2} - \ln \langle I \rangle - \beta^2 - \frac{C}{Z_2} + L_n \right] \tag{5.3}$$

where r_0 is the classical electron radius, N_A is Avogadro's number, ρ is the mass density of the target material, m_e is the mass of the electron, c is the speed of light, Z_2, z_1, and M_2

are as before, $\beta = v/c$, v is the ion velocity,$\langle I \rangle$ is the mean ionization potential, C is the shell correction, and L_n is the sum of all the higher order corrections [172]. The higher order corrections include the density correction, the Barkas correction, and the Bethe correction [170]. C and $\langle I \rangle$ are normally treated as fitting parameters, allowing for adjustment of the equation to match experimental data or alternative approaches to calculating stopping powers [172].

Electronic stopping power in the low energy regime is a simple power law, or

$$S_{e,low} = AE^n, \tag{5.4}$$

where A and n are fitting parameters. $S_{e,low}$ is used at ion energies below the target conduction band [172].

At intermediate energies a physical theory is not available. Instead, a "bridging theory" is used, or

$$S_e = \frac{S_{e,low}\ S_{e,mid}}{S_{e,low} + S_{e,mid}} \tag{5.5}$$

where $S_{e,low}$ is the power law from equation 5.4 and $S_{e,mid}$ is a function used to fit the intermediate energies between the low energy formula and the Bethe-Bloch Equation. The intermediate energy function is selected to provide optimal fitting of this region of the stopping power curve. ICRU Report 49 used a logarithmic function for fitting $S_{e,mid}$ [168]. The functional form of this "bridging function" used in SRIM is proprietary. It is also necessary to interpolate between the Bethe-Bloch equation, 5.3, and the "bridging theory" to produce a continuous function, spanning a short energy interval between the intermediate to high energy regime. Example boundaries of ion energy regimes for the ICRU formalism are shown in Figures 5.1 and 5.2.

All of the electronic stopping power equations above are combined and then fit to experimental data. SRIM uses published measurements and evaluations of stopping powers in various materials from many authors, where at least 9 free parameters are allowed to vary

to match the theory to the experimental results [172]. An effective charge scaling method is used to convert hydrogen stopping powers to helium stopping powers [172] and, for ions heavier than helium, hydrogen stopping powers are scaled using Brandt-Kitagawa Theory [176].

Range is defined as

$$R = \int_0^{E_0} S^{-1} dE \tag{5.6}$$

where S is the stopping power and dE is the differential energy. Evaluation of the definite integral in equation 5.6 provides the total pathlength traveled by the particle, i.e., using the straight-ahead approximation. Projected range additionally takes into account the effect multiple coulomb scattering and range straggling. SRIM uses the Projected Range Algorithm (PRAL), developed by Biersack [177] to calculate projected range.

5.2.2. Proposed Analytical Model for Ion Stopping Powers and Projected Range

The proposed model is based on the Bragg-Kleeman (BK) Rule [173], or

$$R = \alpha E^p \tag{5.7}$$

where α and p are empirical constants and E is the particle energy in MeV. This empirical formula has been shown to fit to an energy range of about 1-200 MeV for hydrogen ions in water [178, 179] and has been used extensively in proton therapy for determining beam energy requirements and water equivalent thicknesses [174, 175].

Bortfeld (1997) showed that the stopping power can be approximated from the Bragg-Kleeman rule using:

$$\frac{dE}{dx}\frac{1}{\alpha p E^{p-1}}. \tag{5.8}$$

The parameters α or p are obtained by fitting equations 5.7 or 5.8 to data from measurements, theory, or evaluations [174] .

In this work, we propose a new expression for stopping powers given by

$$S_p = \frac{\kappa}{G(\hat{E}) + H(\hat{E}) - J(\hat{E})} \tag{5.9}$$

where κ is a constant, $G(\hat{E})$ describes the low energy regime, $H(\hat{E})$ describes the high energy regime, and $J(\hat{E})$ is a high energy correction, and $\hat{E}$ is defined as energy per nucleon, MeV u^{-1}. κ is

$$\kappa = 2r_0^2 n_e m_e c^2 z_1^2 \quad \text{MeV cm}^{-1} \tag{5.10}$$

where r_0 is the classical electron radius, m_e is the mass of the electron, b is the speed of light, and z_1 is the charge of the projectile. n_e is the electron density of the target material, or $N_a \rho Z_2 / M_2$. $G(\hat{E})$ corresponds to the low energy regime and is given by

$$G(\hat{E}) = \beta q \hat{E}^{q-1} \tag{5.11}$$

where β and q are empirical fitting parameters.$H(\hat{E})$ corresponds to the high-energy regime, and is expressed as

$$H(\hat{E}) = \alpha p \hat{E}^{p-1} \tag{5.12}$$

where α and p are additional fitting parameters. Finally, $J(\hat{E})$ corresponds to the ultra-high energy regime and may be considered as a correction to $H(\hat{E})$ (e.g., for the density effect and other higher order corrections) that helps fit stopping powers above 100 MeV u^{-1}. It is expressed as

$$J(\hat{E}) = h\left(e^{-g\hat{E}} - 1\right) \tag{5.13}$$

where h and g are fitting parameters. By substituting equation 5.9 into Eq. 5.6 and integrating we obtain an expression for range given by

$$R = \int_0^{E_0} \frac{1}{S} = \frac{1}{\kappa}\left[\beta \hat{E}^q + \alpha \hat{E}^p + \frac{h}{g}\left(e^{-g\hat{E}} + g\hat{E} - 1\right)\right] u \tag{5.14}$$

where κ and the fitting parameters are as defined previously and u is the atomic mass number of the projectile. The factor of u arises from substitution of variables required for the integration of $\hat{E}$ in equation 5.6.

5.2.3. Stopping Power and Projected Range data for Fitting

To test the performance equations 5.9 and 5.14 we selected hydrogen, carbon, iron, and uranium ions due to clinical interest in radiation therapy, space radiation protection, and basic physics. For hydrogen ions, the target materials were water, carbon, aluminum, lead, copper, iron, gadolinium, and gold. For the other projectiles, the target materials were water, carbon, aluminum, lead, and copper. For all ions and materials, we generated the total stopping power and projected range data using SRIM 2012. The SRIM software provides 26 energies per decade for its stopping power and range data output by default. The lower bound of the energy interval for each ion and target combination was calculated using

$$E_L(\text{keV}) = 0.922\ Z_1\ Z_2\ \left(\frac{M_1}{M_2} + 1\right)\left(Z_1^{0.23} + z_2^{0.23}\right), \tag{5.15}$$

where, M_1, Z_1, M_2, and z_2 are as previously defined. This approach follows from Equation 2-88 in Ziegler et al. (2012). A fixed upper energy bound was used for all targets of a particular ion. Table 5.1 lists the energy intervals for all ions.

We generated data at 10 energy bins per decade within the energy intervals defined by the minimum and maximum energies (table 5.1). Data was generated with variable logarithmic spacing according to

$$\hat{E} = 0.001\ (\text{MeV u}^{-1}) \cdot 10^{i/B} \tag{5.16}$$

where B is the number of energies per decade and i is the index of the ith data point. The stopping power and range values at each ion energy were then extracted from the SRIM data using a piecewise cubic Hermite polynomial interpolation, implemented in MATLAB® (2012, The MathWorks Inc., Natick, Massachusetts, United States)

Table 5.1. Energy cutoffs for each ion investigated in this study, as well as, where this cutoff occurs in range of the ion in water.

Projectile Ion	Lower Energy Bound (MeV u^{-1})	Upper Energy Bound (MeV u^{-1})	Upper Range (cm)
Hydrogen	0.002 – 0.029	450	116
Carbon	0.017 – 0.171	600	53
Iron	0.053 – 0.217	1400	43
Uranium	0.193 – 0.351	2500	30

Note. The upper range column was rounded to the nearest centimeter. Lower energy limits are rounded to the nearest keV.

Uncertainties in the stopping power and range data from SRIM were estimated from the ratio of data from PSTAR [180] to data from SRIM [172] . The resulting ratios were 5% or less for $\hat{E} \leq 1$ MeV and 2% or less for $\hat{E} > 1$ MeV. These ratios provide a good estimate for the lower limit of the uncertainties because PSTAR and SRIM use different calculation methods, fitting techniques, and source data. The reduced uncertainty seen above 1 MeV is due to the fact that the Bethe-Bloch Equation is used by both SRIM and PSTAR to generate high-energy stopping power values.

5.2.4. Fitting Procedure

Fitting was performed using a hybrid genetic algorithm [181] implemented in commercial software (MATLAB Global Optimization Toolbox, The MathWorks Inc., Natick, Massachusetts, United States), to locate global minima. The algorithm was configured with a population size of 1000, which was broken into 10 sub-populations, and was allowed to use up to 1200 generations to converge on a solution. A constrained minimization (MATLAB Optimization Toolbox) was used to finalize the fit parameters. The objective function minimized is given by

$$FOM_i = \sqrt{\left(\frac{|R_{fit,i} - R|}{\Delta_R}\right)^2 + 0.97\left(\frac{|S_{fit,i} - S|/S}{\Delta_S}\right)^2 + 0.03\left(\frac{\max\left(|S_{fit,i} - S|/S\right)}{\Delta_S}\right)^2} \quad (5.17)$$

Table 5.2. Constraints on fitting parameters provided to the hybrid genetic algorithm.

Bound	β	r	α	p	g	h
Lower	0	0	0	1	0	0
Upper	0.8	1	0.8	2.1	0.5	0.15

where $|R_{fit,i}-R|$ represents the absolute value of range difference, $|S_{fit,i}-S|/S$ represents the absolute value of the relative stopping power difference, $\max\left(|S_{fit,i}-S|/S\right)$ is the maximum absolute value of the relative difference in stopping power, and Δ_R and Δ_S are criteria parameters that govern the relative importance of the three terms in equation 5.17. The values of criteria used were $\Delta_R = 0.1$ cm, and $\Delta_S = 10\%$. The first two terms in the objective function minimize the global deviations in both range and stopping power, while the third term reduced local hotspots. The relative weight of the two stopping power terms, in equation 5.17, was determined empirically to balance the effects of the third term. We applied constraints (table 5.2) to the fitting parameters to facilitate reliable optimization.

Due to the stochastic nature of the start points and the hybrid genetic algorithm, final solutions occasionally converged on a solution that was outside of the expected results. To avoid the impact of these outliers on final fitting parameters, each dataset was fit 35 times and outliers were excluded using Chauvenet's criteria [182, 183] .

5.2.5. Robustness to Sparse Data

We tested the model and fitting procedure for robustness to sparse data by varying the number of energies per decade (B=1,2,4,6,…,26 in equation 5.16) for all datasets. Each set of data points was fit 40 times and all the trials were run sequentially. As before, outliers were rejected using Chauvenet's criteria.

5.2.6. Average Time Required to Compute Stopping Power and Range

Stopping power calculation times were recorded for the proposed analytical model (Eq. 5.9) and an in-house full theory model (Eqs. 5.2-5.5) to generate lookup tables of stopping power. The lookup table was generated with energies defined by equation 5.16 using 28 energies per decade and 160 total energies.

Range calculation times were recorded for the analytical formula (Eq. 5.14) and, for comparison, the numerical integration (NI) of equation 5.6. NI was performed using stopping powers calculated on the fly with either the proposed analytical model or the full theory model. Additionally, NI was performed using nearest neighbor interpolation from a fast lookup table, which used the same energy grid as the stopping power tables.

Timing information was determined by using the minimum time of 10 trials, each consisting of 10,000 averages to overcome system timing limitations. Comparisons were performed using in-house software developed in C++ and compiled with the Intel 2016 Compilers (Intel Corporation, Santa Clara, CA). Full Optimization (-O3) was utilized during compiling and linking and the automatic parallelization was disabled. A personal computer running Windows 10 Pro (build 14342, Microsoft Corporation, Redmond Washington) with a 4.05 GHz processor was used to collect timing information.

5.3. Results

5.3.1. Fitting Parameters for New Model

The parameters of the analytical model for each ion-target combination are listed in tables 5.3 and 5.4. These parameters can be used in equation 5.9 to calculate stopping powers or in equation 5.14 to calculate the range of the ion. Using these parameter values with all of the significant digits listed in these tables will allow replication of our results to within 0.1% for stopping powers or 7 m for range. The uncertainty in individual parameter values due to the fitting procedure was less than 4% in all cases.

Table 5.3. Parameters from fitting hydrogen and carbon stopping powers and ranges.

Hydrogen						
Material	$\beta\,(\times 10^{-5})$	q	$\alpha\,(\times 10^{-4})$	p	$g\,(\times 10^{-2})$	$h\,(\times 10^{-3})$
H_2O	3.1764	0.4583	1.3127	1.6493	2.1239	4.1618
Carbon	3.3753	0.5050	1.3509	1.6591	2.0452	4.2046
Aluminum	4.2267	0.4619	1.9859	1.5648	2.7479	3.8267
Lead	17.0464	0.2332	5.2020	1.4714	3.2305	3.6440
Copper	12.8511	0.5381	2.6283	1.5724	2.6525	3.9419
Iron	7.6905	0.3141	2.5231	1.5759	2.6049	3.8929
Gold	19.5433	0.2635	5.2962	1.4539	3.3555	3.7148
Gadolinium	11.9556	0.4262	4.2390	1.4789	3.2748	3.6800
Carbon						
Material	$\beta\,(\times 10^{-5})$	q	$\alpha\,(\times 10^{-4})$	p	$g\,(\times 10^{-2})$	$h\,(\times 10^{-3})$
H_2O	26.1662	0.346	1.2682	1.6391	2.4017	3.966
Carbon	26.5006	0.5221	1.3069	1.6375	2.4551	3.8986
Aluminum	41.1891	0.6422	1.3374	1.6714	2.2982	4.5151
Lead	103.8596	0.5369	3.9312	1.5368	3.0039	4.0902
Copper	71.5322	0.4574	2.3035	1.593	2.7591	4.0453

Note. All significant digits provided should be used to create accurate stopping powers and ranges.

Table 5.4. Parameters from fitting iron and uranium stopping powers and ranges.

Iron						
Material	$\beta\,(\times 10^{-5})$	q	$\alpha\,(\times 10^{-4})$	p	$g\,(\times 10^{-2})$	$h\,(\times 10^{-3})$
H_2O	667.2047	0.0461	6.0204	1.2719	3.5983	2.8844
Carbon	539.2538	0.0503	8.8684	1.0907	4.6477	2.7597
Aluminum	253.2571	0.4511	3.7556	1.4396	3.3158	3.2088
Lead	651.7461	0.3451	11.2829	1.2839	4.351	3.1556
Copper	817.2919	0.1272	10.6575	1.2179	4.2673	3.14970
Uranium						
Material	$\beta\,(\times 10^{-5})$	q	$\alpha\,(\times 10^{-4})$	p	$g\,(\times 10^{-2})$	$h\,(\times 10^{-3})$
H_2O	2186.154	0.12	21.7563	1.1526	3.391	2.8531
Carbon	1734.845	0.2086	19.8765	1.1803	3.4502	2.7267
Aluminum	2417.656	0.226	19.1826	1.2094	3.5016	2.9827
Lead	5866.977	0.1961	45.5524	1.0922	4.4667	3.1018
Copper	8817.095	0.0741	39.6814	1.0807	4.2152	3.04543

Note. All significant digits provided should be used to create accurate stopping powers and ranges.

Table 5.5. Average and maximum relative stopping power difference between SRIM and the proposed model for all ion/target combinations considered.

Ion	Hydrogen		Carbon		Iron		Uranium	
Material	Average (%)	Max (%)	Average (%)	Max (%)	Average (%)	Max (%)	Average (%)	Max (%)
H_2O	-0.44	8.10	-0.13	-5.00	-0.28	-6.93	-0.41	-6.16
Carbon	0.40	-4.73	0.02	-2.93	0.29	-2.28	0.17	-3.25
Aluminum	0.06	-3.13	-0.39	-5.80	0.14	-4.34	0.10	1.35
Lead	0.37	-3.91	-0.03	2.47	0.13	-2.37	0.07	-2.42
Copper	0.01	0.62	-0.10	-1.52	0.72	3.81	0.38	-3.72
Iron	-0.02	2.26	-	-	-	-	-	-
Gold	0.37	2.95	-	-	-	-	-	-
Gadolinium	0.45	1.71	-	-	-	-	-	-

Table 5.6. Average and maximum range difference between SRIM and the proposed model for all ion/target combinations considered.

Ion	Hydrogen		Carbon		Iron		Uranium	
Material	Average (mm)	Max (mm)	Average (mm)	Max (mm)	Average (mm)	Max (mm)	Average (mm)	Max (mm)
H_2O	0.018	0.091	0.044	0.508	0.105	-0.392	0.070	-0.206
Carbon	0.010	0.109	0.006	0.044	0.054	-0.530	0.049	-0.249
Aluminum	0.021	-0.194	0.009	-0.046	0.015	-0.113	0.022	-0.085
Lead	0.021	-0.086	0.005	-0.023	0.011	-0.086	0.021	-0.088
Copper	0.002	-0.019	0.004	0.028	0.044	-0.332	0.080	-0.295
Iron	0.004	-0.02	-	-	-	-	-	-
Gold	0.020	-0.113	-	-	-	-	-	-
Gadolinium	0.016	-0.0981	-	-	-	-	-	-

5.3.2. Difference in Stopping Power and Range Values from SRIM and Proposed Model

Tables 5.5 and 5.6 list the average and maximum errors for all ion and target combinations. The largest uncertainties in stopping power are associated with hydrogen ions in water. The reasons for this will be addressed in the discussion section.

Figure 5.1 plots $S(\hat{E})$ and the inverse of $J(\hat{E})$, $H(\hat{E})$, and $G(\hat{E})$ for hydrogen ions incident on carbon. The low-energy term closely matches the training data. The high-energy BK rule is offset from the training data but it is clear that the sum with the exponential term reduces

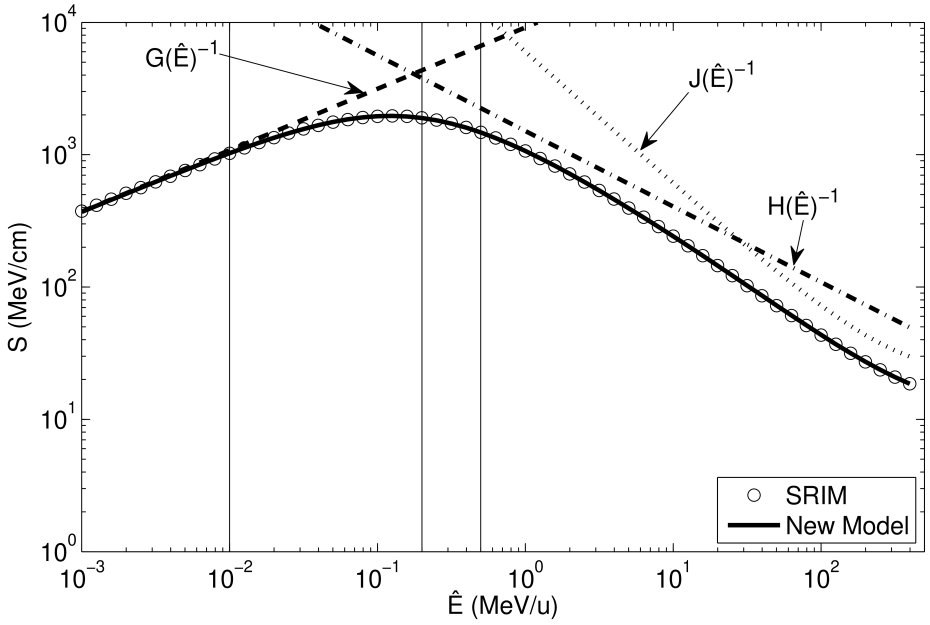


Figure 5.1. Components of the proposed stopping power equation, 5.9, for the stopping power, S, of hydrogen ions in carbon as a function of ion energy, $\hat{E}$. The figure is divided into the four regimes, as defined by the vertical bars, whose energy values were taken from ICRU Report 49 (1994) to construct their stopping power curves. The solid line represents the complete fit model. $H(\hat{E})$ is represented by the dashed line. The dash-dot line is the high energy term, $G(\hat{E})$. The exponential term, $J(\hat{E})$, is shown as the dotted line. The data presented here is for hydrogen ions in copper.

the stopping power to match the base data. Similarly, figure 5.2 reveals good agreement in range values and how each component of the model contributes to the projected range calculation.

Figures 5.3(a) and (b) plot of stopping power and range values for hydrogen ions in water. This was the data with the worst overall agreement in stopping power value. The maximum relative difference in stopping power values was 8.1%. Interestingly, this difference occurred at the ion energy interval where SRIM interpolates between intermediate and high energy stopping theories, which may introduce small, non-physical artifacts in the curve. This worst-case result for stopping power values is still less than the stated goal of 10%. Figures 5.3(c) and (d) plot stopping power and range for carbon ions in copper. These figures reveal excellent agreement between data from the analytical model and SRIM, with differences in stopping power values of less than 2% and range differences of 3 m. Figures 5.4(a) and (b)

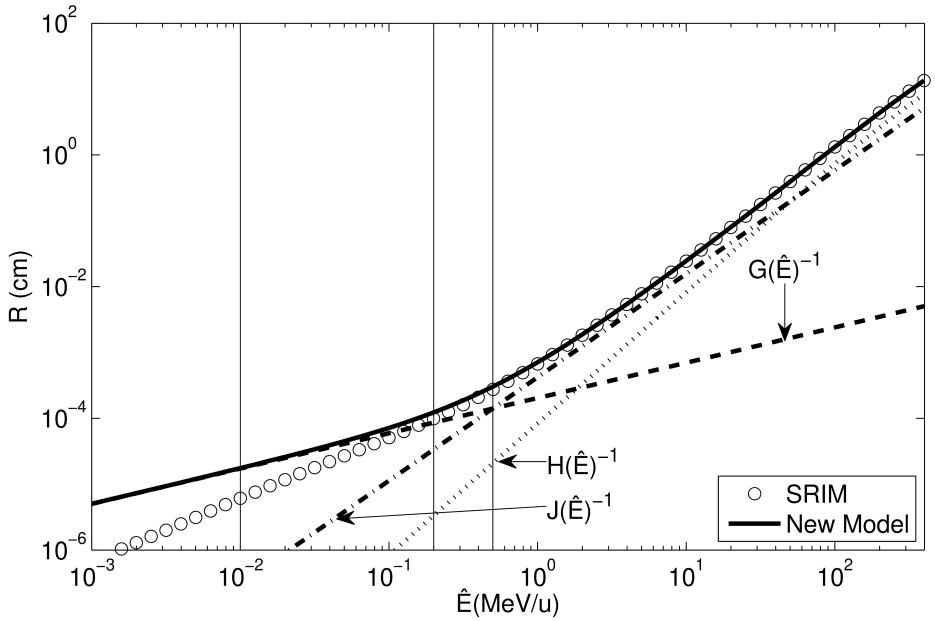


Figure 5.2. Components of the proposed range equation, 5.14, for the range, R, of hydrogen ions in carbon as a function of ion energy, $\hat{E}$. The figure is divided into the four regimes, as defined by the vertical bars, whose energy values were taken from ICRU Report 49 (1994) to construct their stopping power curves. The solid line represents the complete fit model. $H(\hat{E})$ is represented by the dashed line. The dash-dot line is the high energy term,$G(\hat{E})$. The exponential term, $J(\hat{E})$, is shown as the dotted line. The data presented here is for hydrogen ions in copper.

plot stopping power and range values for iron in carbon. This worst-case result for range values (0.53 mm maximum difference) is still less than the 1 mm goal of this work. Finally, Figures 5.4(c) and (d) plot stopping power and range for uranium ions in aluminum. These data once again show excellent agreement between data from the analytical model and the SRIM data, with stopping power differences less than 1.5% and range differences less than 0.1 mm.

5.3.3. Robustness to Sparse Data

Figure 5.5 plots the maximum difference in range and stopping power versus the number of energy values per decade for a few representative ion-target combinations. The data reveals that with 4 or more energies per decade, the criteria of 10% stopping power difference and 1 mm range difference were met. The differences also were stable between 4 and

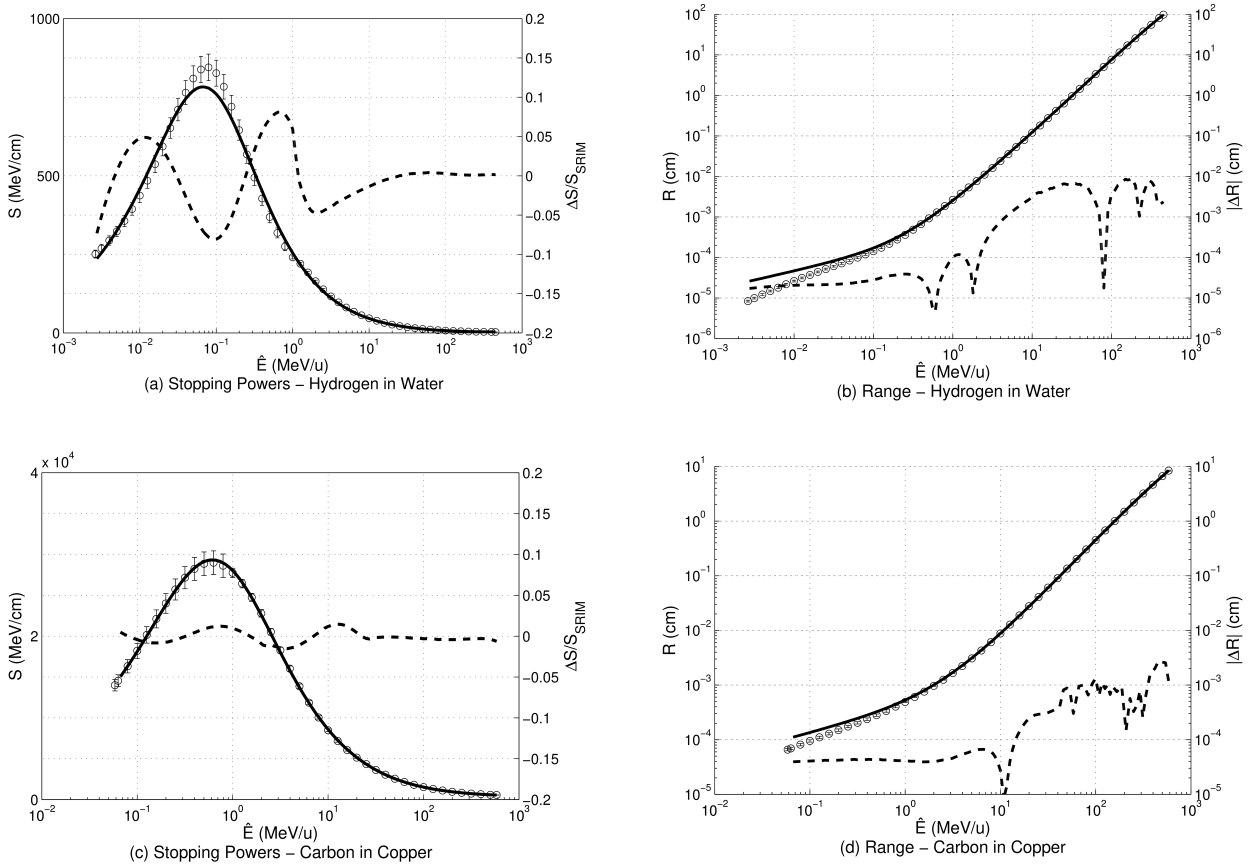


Figure 5.3. Stopping powers, S, and ranges, R, versus energy, $\hat{E}$, for hydrogen ions in water ((a) and (b)) and carbon ions in copper ((c) and (d)). The solid line represents the proposed model. The dashed line is the absolute value of the difference in range, $|\Delta R|$, or the relative difference in stopping powers, $\Delta S/S_{SRIM}$, and are associated with the axis on the right. The circles represent the data points used in the fitting.

26 energies per decade, never varying by more than 1% in stopping power or 0.25 mm in range.

5.3.4. Average Time Requires to Compute Range and Stopping Power

The proposed analytical model for stopping power was faster than a full theory based calculation. The calculations of 160 values took 10.7 s with the proposed analytical model versus 15.1 s with the full theory, corresponding to a 28% reduction in computation time.

The proposed analytical range model was the fastest approach to calculate range. Table 5.7 shows the timing results for all of the numerical integration tests and the analytical

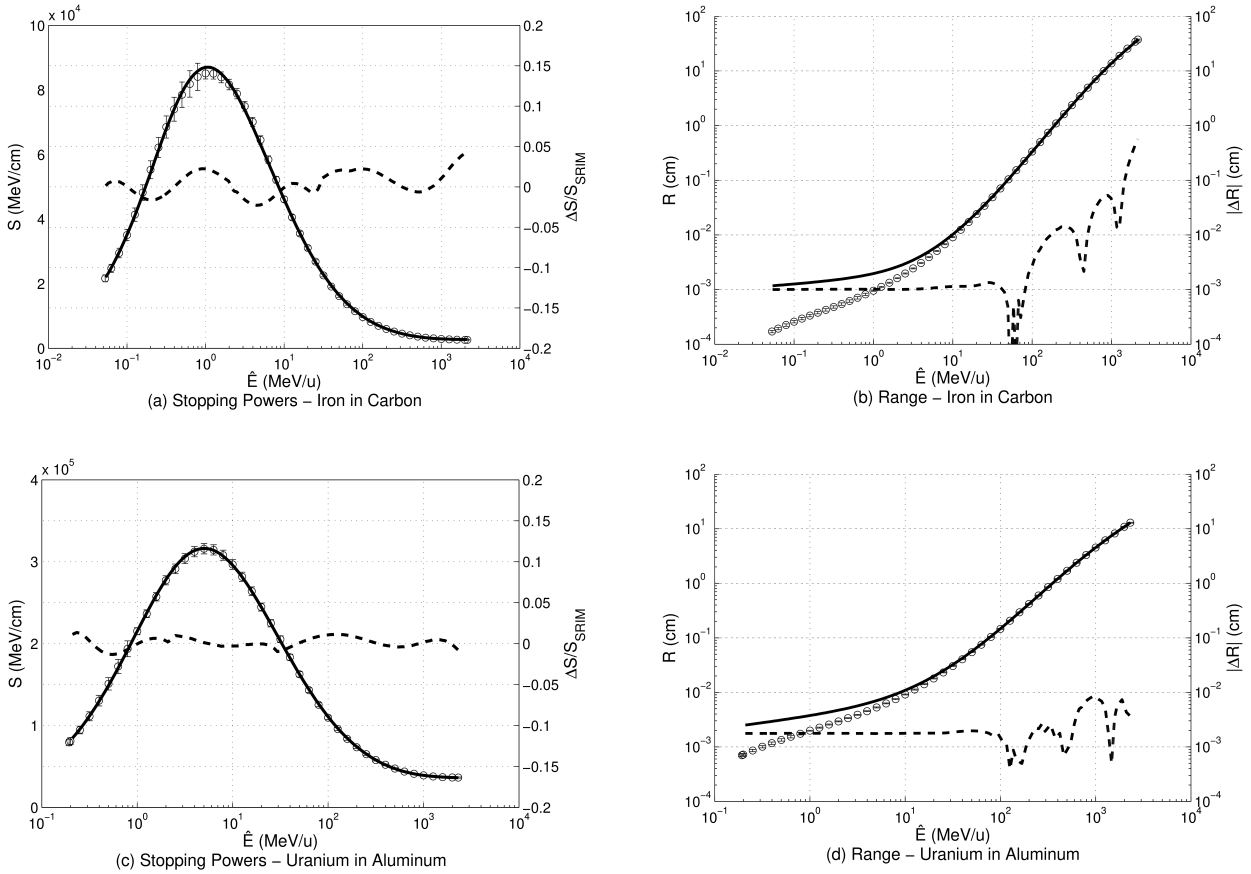


Figure 5.4. Stopping powers, S, and ranges, R, versus energy, $\hat{E}$, for iron ions in carbon ((a) and (b)) and uranium ions in aluminum ((c) and (d)). The solid line represents the proposed model. The dashed line is the absolute value of the difference in range, $|\Delta R|$, or the relative difference in stopping powers, $\Delta S/S_{SRIM}$, and are associated with the axis on the right. The circles represent the data points used in the fitting.

model. This data demonstrates that the analytical range model performs calculations at least 945 times faster than any numerical integration (NI) approach tested in this work. The data also demonstrates a fast lookup table is only 34% faster than on-the-fly calculations using the proposed model. Using the proposed model to calculate stopping powers is 38% faster than using the full theory model when performing NI.

5.4. Discussion

In this study, we developed a simple analytical expression for calculating stopping power and projected range. We evaluated the new model's ability to fit evaluated stopping powers and projected ranges, robustness to sparse data, and computational speed. The model was

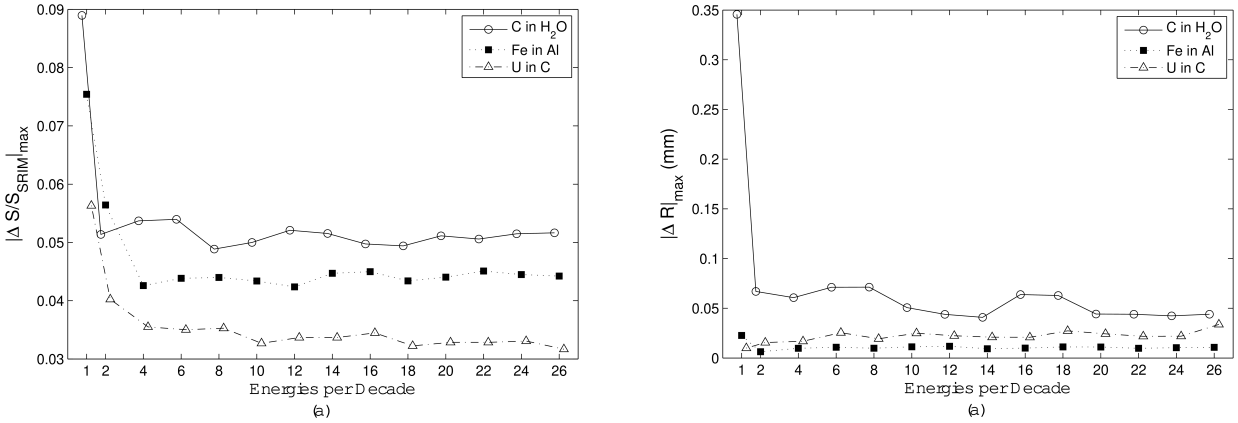


Figure 5.5. The two panels above show the maximum stopping and range errors versus the number of energy points per decade for a representative collection of ions and target materials. The data for carbon ions in water, iron ions in aluminum, and uranium in carbon are represented by circles, squares, and diamonds respectively. Error bars represent one standard deviation of the 40 trials run to collect each data point. Different features of these datasets are discussed in the text.

Table 5.7. Computation time and speedup of range calculations.

Model	Time (s)	Speedup
Analytical range	0.048	1
NI: Full theory stopping power	107.859	2267
NI: Analytical stopping sower	67.390	1416
NI: Lookup table stopping power	45.000	946

Note. The first row shows the results of using the analytical model to find the range, which is used as the timing standard. Rows representing numerical integration (NI) specify the stopping power model used in the integration. Speedup is defined as T_i/T_A where T_i is the time of the method and$T_A$is the time using the analytical range model.

fit to evaluated stopping powers and ranges with better than 8.1% accuracy in stopping powers and less than 0.6 mm in range for hydrogen, carbon, iron, and uranium projectiles in water, carbon, aluminum, lead, copper, iron, gadolinium, and gold for all ion energies investigated (an interval of at least 351 keVu^{-1} to 450 MeV u^{-1}). The model and fitting procedure are robust to sparse data, requiring only 4 energies per decade to produce results with less than 10% relative difference in stopping power or 0.53 mm in range. The proposed range model demonstrated a reduced computation time, when compared with numerical

integration. Additionally, it was shown that the analytical stopping power was nearly as fast as the approach of using pre-calculated values and fast lookup tables.

The implication of the findings of this study include the possibility to increase speed and/or accuracy of charged particle transport calculations. One application is the calculation of clinical water equivalent thickness and stopping power ratios, as demonstrated by Zhang and Newhauser [174], with an expanded interval of ion-energy applicability. The proposed model overcomes many of the limitations presented in their study by improving the accuracy of the model over a wider range of energies. Memory-constrained GPU-based dose calculation algorithms may also benefit from the proposed model. The proposed stopping power model used only 2% of the memory required to store a lookup table. This approach could enable GPU-based algorithms to increase calculation accuracy by increasing the number of materials that can be stored in the unit memory and speed by reducing the amount of information that needs to be transferred to the GPU. Additionally, the model parameters can be used to calculate the range of the particles at no additional memory cost.

The results from our study are consistent with those from Zhang and Newhauser [174] and Zhang *et al.* [175]. In their work, they fit a single Bragg-Kleeman Rule to hydrogen, helium, carbon, and iron stopping powers for energies between 10 and 250 MeV u^{-1}. Using their model with the fitting procedure and data used in this study in that energy interval, the maximum relative difference produced was 9%. In this work, the maximum relative difference was reduced to less than 0.3%. Outside of the 10 to 250 MeV u^{-1} interval, a single BK rule is not applicable whereas our model performed well in the energy range from 1 keV to 450 MeV u^{-1}. Konac *et al.* [184] proposed another parameterization of electronic stopping powers for heavy ions. In their work they fitted stopping powers of ions ranging from hydrogen to bismuth in carbon and silicon targets in the energy interval from 0.01 to 100 MeV u^{-1}. They were able to achieve uncertainties of less than 10% for all of their tested ions with only 6 free parameters. In this work, our model had a maximum stopping power uncertainty of less than 9% in the ions and targets investigated, with a larger range of applicability of target

materials. Of all of the stopping power models reviewed [184–189], none of them provided an analytical equation for calculating range using the same parameters provided by their stopping power model.

The major strength of this study was the wide variety of projectile ions and target materials used in training and testing the model. The four ions selected span the periodic table demonstrating the broad applicability of the model. A second strength of this study was the use of a stochastic fitting algorithm to reliably find the global minimum values during optimization.

A limitation of this work is our model did not explicitly attempt to take into account Bragg additivity or molecular bonding effects when modeling compounds and mixtures. This limitation is not serious because these effects may be taken into account in the input data, e.g. implicitly in measured data or explicitly when the using molecular bonding theory contained in SRIM, used in the fitting procedure. Additionally, the proposed model does not explicitly calculate losses due to radiative processes, e.g. bremsstrahlung. This is not a significant limitation because the model is not applicable in the energy regime where radiative losses become important, i.e. ~100-1000 $GeVu^{-1}$ [169, 190]. This is due to the radiative losses being inversely proportional to the square of the ion mass. Finally, this work did not perform exhaustive fitting of ion species and target materials. This is not a severe limitation because the cases considered are representative, demonstrate broad applicability, and users may easily apply our method to arbitrary cases.

A possible future extension could be to develop a parameterization of the fitting parameters (α, p, β, q, g, and h) based on target material and ion properties. If the fitting parameters could be calculated using physical quantities, this would improve convenience and applicability. In addition, the model could be further extended to increase the energy interval of applicability, especially in the high-energy regime. Finally, further investigation into the computational performance and optimization of the model should be performed.

This study has shown that a simple, continuous analytical model can be fit to evaluated data to provide stopping power and range data of reasonable accuracy for various applications. The faster computation speed compared to a full theoretical model and the smaller memory footprint compared to lookup tables may enable improvements in limited memory parallel computing applications.

6. Computational Feasibility of Simulating Changes in Blood Flow through Whole-Organ Vascular Networks from Radiation Injury

6.1. Introduction

The circulatory system comprises the heart, lungs, arteries, capillaries, and veins, which delivers blood to most tissues in the body [76]. Capillaries, which facilitate the transfer of oxygen and nutrients from the blood to the surrounding tissue, determine the local and regional blood flow of organs [14]. Radiation injury of vessels can lead to beneficial and detrimental responses [5, 11, 191]. For example, radiation therapy prunes tumor blood vessels, reducing the amount of oxygen and nutrients deliver, effectively causing the tumor to stave. However, radiation necrosis, a potentially fatal side-effect which occurs in approximately 5% to 25% of patients [6], has been correlated with increased vascular abnormalities caused by radiation injury [10, 11]. Histologic and clinical techniques have been used to study necrosis, with limited success toward understanding the complex sequence of events between vascular injury and the onset of necrosis [10, 11]. Computer simulations could, in principle, elucidate the key determinants of the response to vascular injury, including reduced blood flow, modification of the tumor micro-environment and induction of necrosis. Such simulations would, by necessity, entail modeling physical aspects, such as vascular geometry, radiation dosimetry, and blood flow, as well as radiation biology.

Much is known about each of these modeling methods and progress in recent years has been remarkable. The human brain contains up to 9 billion vessels, forming the densest vascular network of the body [12, 62]. The feasibility of simulating the vascular geometry (Chapter 2) [132] and blood flow (Chapter 3) [160] through networks with up to 17 billion vessels was recently demonstrated. Radiation injury of the capillaries, which are only micrometers in size [12, 77], requires radiation transport modeling on the same scale. Monte Carlo simulations are the most realistic way to simulate radiation dose deposition at microscopic or macroscopic scales [32, 192–194], yet it remains computationally prohibitive to

simulate all relevant dimensional scales simultaneously. This limitation if frequently overcome by using amorphous track-structure models, which provide a faster but slightly less physically realistic approach to simulating radiation dose [34, 35, 195]. Irradiation of a blood vessel causes a biologic response in the vessel's function and structure, including increased permeability, loss of elasticity, and pruning of the vascular tree [40–42, 196]. Recently, some progress has been made in understanding the mechanistic processes behind these manifestations of injury [197–199], primarily regarding the cellular mechanisms and local changes in blood flow. Computational techniques for modeling both dose deposition and biologic response have been applied in small volumes of tissue to investigate the effects of radiation on the tumor micro-environment [200, 201]. However, little attention has been paid in the literature to simulations of the effects radiation has on the vascular networks of whole organs. It is currently unknown if it is computationally feasible to simulate the radiation dose to individual blood vessels and model their biological changes in a network the size of the human brain.

The goal of this work was to test the the computational feasibility of simulating the dose deposition to a whole-organ vascular network and the resulting change in blood flow. To do this, we developed a new amorphous track-structure model to transport radiation and combined this with previously described methods to model the vasculature and blood flow rates (Chapters 2-4). We assessed the algorithm's execution time, computational scalability, and memory usage.

6.2. Methods

6.2.1. Geometry Description

The vascular geometry was created using fractal methods (Chapter 2) [132] and then partitioned to create a computationally efficient data distribution (Chapter 3) [160]. The geometric model described each vessel's start point, end point, and inner radius. In the model, each parent vessel branches into two child vessels. The smallest vessels in the network were 2.5 µm in radius and 55 µm in length [12]. Between each generation, the vessel radius

was reduced by $2^{-1/3}$ following Murray's Law for bifurcation [93] and the length was reduced by 0.8 to reduce vessel overlap for visualization [132]. The vascular networks were created in 2-D (for visualization) and 3-D for use in the radiation dose calculation (Figure 6.1). An bounding box that describes the maximum extents of the vascular network was also included in the model.

6.2.2. Dose Calculations

6.2.2.1. Radiation Transport

To test the computational feasibility of the dose calculations, we simulated proton tracks traversing a vascular geometry. The source comprised a beam of protons propagating in the -z direction from a rectangular field above the network (Figure 6.1). We limited the beam laterally to exclude the six largest vessels in the network (Figure 6.2). This was done to exclude large regions of the simulation volume where no vessels exist, which decreased simulation times. The start position of each proton was determined stochastically by uniform sampling from within the field (see Figure 6.2). The initial direction of proton travel was

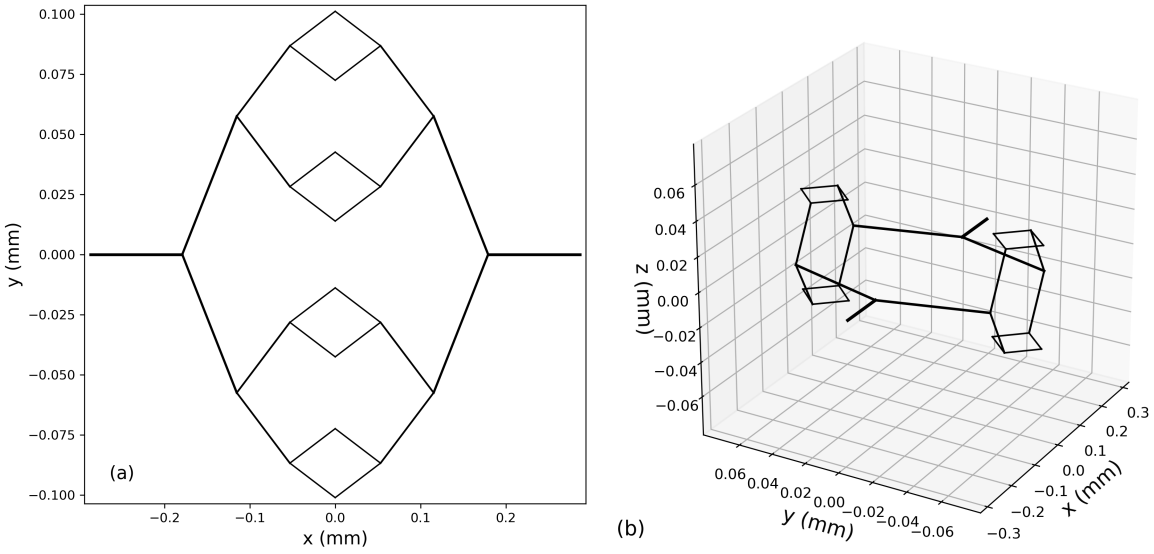


Figure 6.1. Illustration of 2-dimensional (a) and 3-Dimensional (b) versions of a 30-vessel network created using the methods of Donahue and Newhauser [132] (Chapter 2)

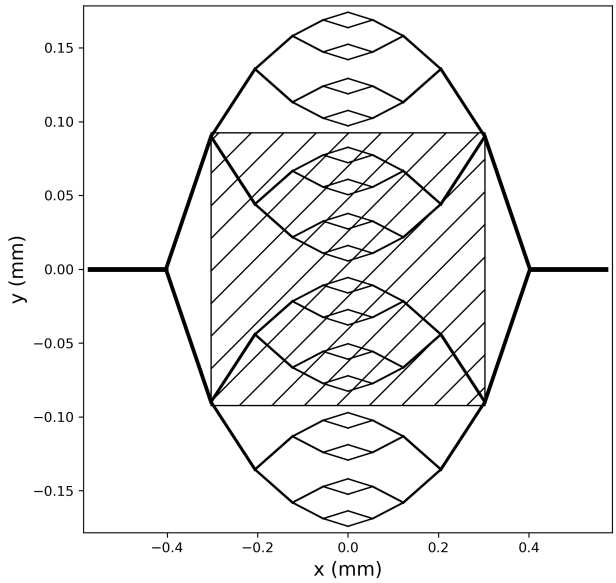


Figure 6.2. An example of the field size (crosshatched area) used in the dose calculation algorithm for an example network with 126 vessels from beams eye view. The 2-dimensional network was used for visual clarity.

stochastically sampled uniformly from a cone with an half-angle of 4 degrees. The initial starting energies of the protons were stochastically sampled from a Gaussian with a width (σ) of 1 MeV and mean value equal to the nominal beam energy (E_0).

We transported the protons using a simple analytic model. Proton energy loss was modeled with the continuous slowing down approximation. Lateral scattering, energy straggling, and nuclear reactions of the protons were neglected to decrease simulation times. Stopping powers were calculated using the methods of Donahue *et al.* [167] (Chapter 5) on the assumption that the vascular networks and surrounding tissue were water. Each proton track was represented by a list of points along a linear trajectory, where the proton kinetic energy and coordinate were recorded for each point. Using the mass stopping power, S, at the current point, the distance to the next point was calculated using an adaptive step size of

$$\Delta x = \frac{0.05E}{\rho S} \tag{6.1}$$

where E is the current proton energy and ρ is the material mass density. Each proton was transported until the particle track exited the simulation volume or its energy fell below a cutoff of 5 keV, at which point the residual energy was deposited locally. Each proton track was surrounded by a bounding cone with a base radius equal to the maximum δ-ray range of the initial proton energy. To calculate the maximum range of the δ-rays in water, we fit an emperical formula to the calculated results from Waligórski *et al.* [34] for proton energies from 1 keV to 350 MeV, or

$$r_{\max} = 9.9420 \times 10^{-4} \cdot \left(E^{1.0225}\right) + 1.4479 \times 10^{-3} \cdot \left(E^{1.7792}\right) \tag{6.2}$$

where E is the proton energy in MeV.

We modeled the dose deposition of a proton track and its secondary particles using an amorphous track-structure model by Elsässer *et al.* [35]. This model considers the dose deposition from the proton core and, separately, from the halo of δ-rays. The dose deposited by the δ-rays is approximated to the radially symmetric (Figure 6.3). The model used was

$$D(r) = \begin{cases} \lambda S / r_{\min}^2 & r \leq r_{\min} \\ \lambda S / r^2 & r_{\min} < r \leq r_{\max} \\ 0 & r > r_{\max} \end{cases} \tag{6.3}$$

where r is the radius from the proton track to the dose calculation point, $r_{\min}$ is the core radius, $r_{\max}$ is the maximum δ-ray range (Equation 6.2), S is the mass stopping power of the proton, and λ is an empirical normalization factor. The normalization factor forces the total energy lost by the proton to be equal to the dose absorbed in the medium from the track's core and halo. It was calculated following the method of Elsässer *et al.* [35], or

$$\lambda = \left(\pi \left[1 + 2 \ln \left(r_{\max} / r_{\min}\right)\right]\right)^{-1} \ . \tag{6.4}$$

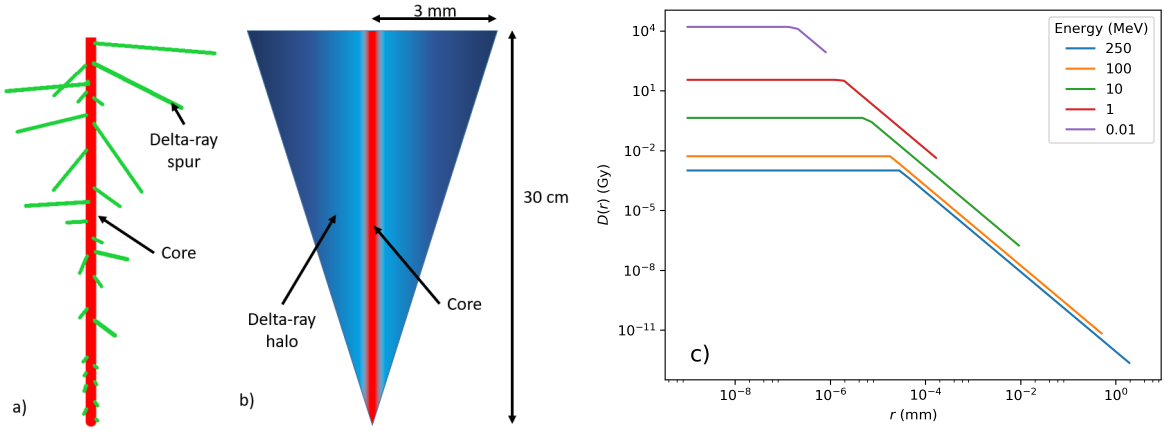


Figure 6.3. An illustration of proton track structure (a) and an amorphous track structure model (b) for a proton traveling toward the bottom of the page. Panel (c) plots the actual track structure model used in this work for various proton energies.

The core radius was

$$r_{\min} = 1.8428 \times 10^{-6} \cdot \left(E^{0.4984}\right) \tag{6.5}$$

where E is the proton energy in MeV [35].

Finally, a k-d tree was constructed for each proton track [202, 203]. The k-d tree provided an efficient method for finding the nearest point in the line of points to any arbitrary point in the geometry. This improved the computation speed of the dose calculations.

6.2.2.2. Dose Scoring

We scored the radiation dose to the endothelial cell layer of the vessel walls. We split the tube-shaped geometry of each vessel into 10 axial segments. For each of these segments, eight dose points were evenly spaced around the midline, forming a ring. Radiation absorbed doses were scored to each of the 80 points.

To efficiently score dose from a proton track to the non-rectilinear geometry of a blood vessel, we used a recursive approach (Figure 6.4). First, the bounding cone of the particle track and the bounding box of the vessel were checked for any intersection. If they overlapped, the dose was calculated at all eight corners of the vessel bounding box and its centroid. The dose at each corner was compared to the dose at the centroid to determine uni-

formity. If all the corner doses were equal to the centroid dose (within ϵ), then the centroid dose was scored for all sub-volumes in the vessel (Figure 6.4b). Otherwise, the doses were calculated to determine if the dose was uniform in each axial segment in the vessel (Figure 6.4c). The bounding shape for each axial segment was a 2-D square that encompassed the circular cross-section of the vessel and contained 8 dose points. If the dose was not uniform (Figure 6.4e), the dose was calculated to all 8 dose points in the segment individually. We selected ϵ to be 50% for this work to limit recursion and the associated simulation time.

6.2.2.3. Algorithm Implementation

The calculation of radiation dose from a proton to a blood vessel is an embarrassingly parallel problem. We separated the algorithm into three applications to facilitate efficient parallel computations (Figure 6.5). These applications together calculated the dose to the vascular network contained in the vessel data file. Details of each of these applications is discussed in the remainder of this section.

The first application (Dose Initialization) prepared the vessel data file for the dose calculations by adding beam parameters to file and dividing up the problem for distributed parallel computations. For distributed parallel computations, we divided the blood vessels into N_{b} blocks, where each block consisted of n_{v} vessels selected contiguously from the vessel data file. The number of blocks in a computation depended on the number of compute nodes, N_{c}, used, or

$$N_{\mathrm{b}} = 16N_{\mathrm{c}} \ . \tag{6.6}$$

Each vessel block was processed by a separate instance of the Dose Calculation application, where each instance was run simultaneously on a separate computer. We used 16 blocks of vessels per compute node to facilitate load balancing across a distributed-computing environment. Because the geometry data was partitioned geometrically, each vessel block contained vessels that were spatially clustered together. Additionally, the Dose Initialization application reserved the cumulative dose-output storage structures in the vessel data file and created

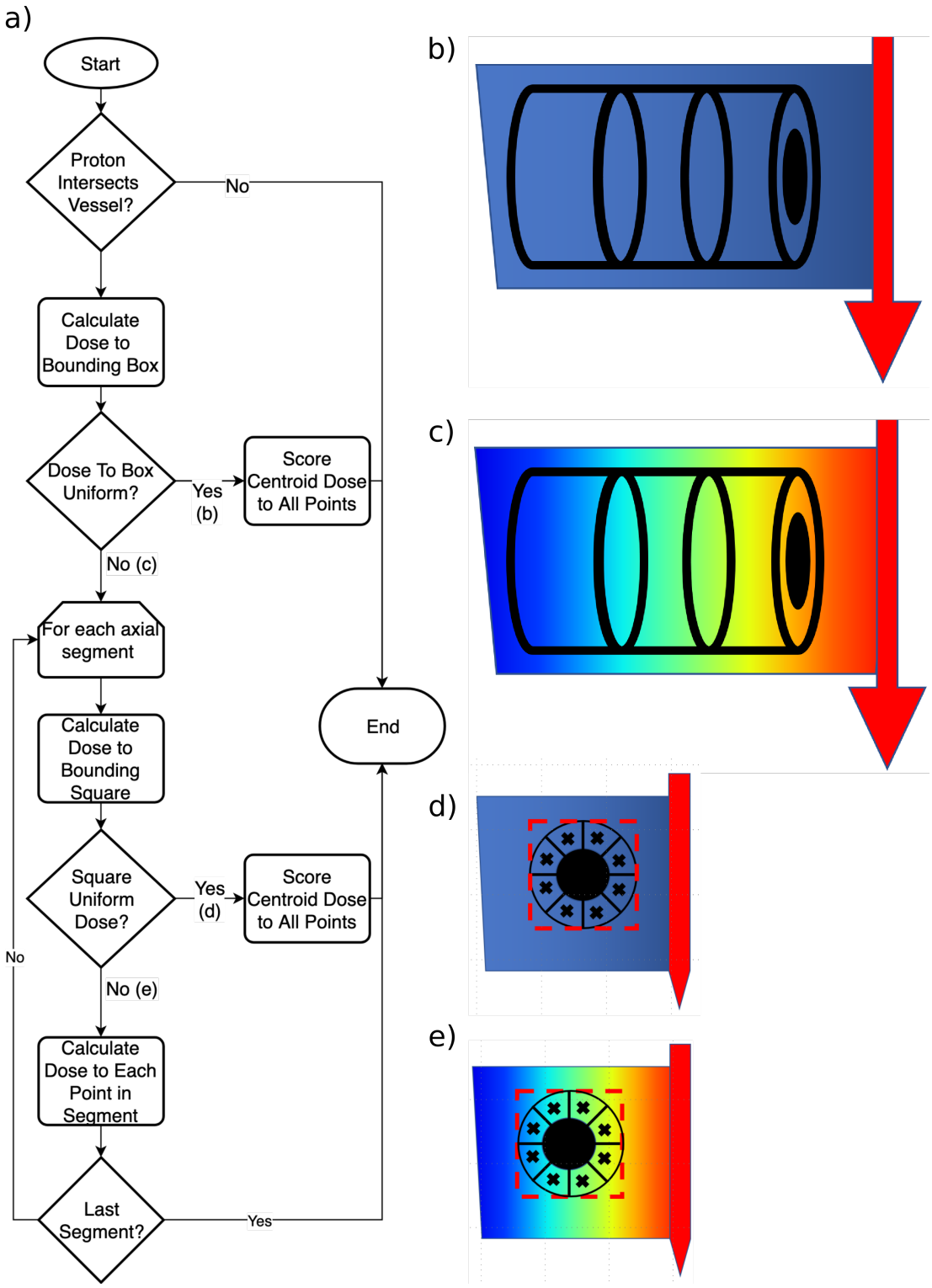


Figure 6.4. Flow chart of the recursive dose algorithm (a). Panels b-e illustrate specific branch cases of flow chart. The red arrows represent the core of a proton track and the color washes represent dose to the vessel (from δ-rays) or segment. Panels d and e represent the cross-section of an axial segment of a vessel.

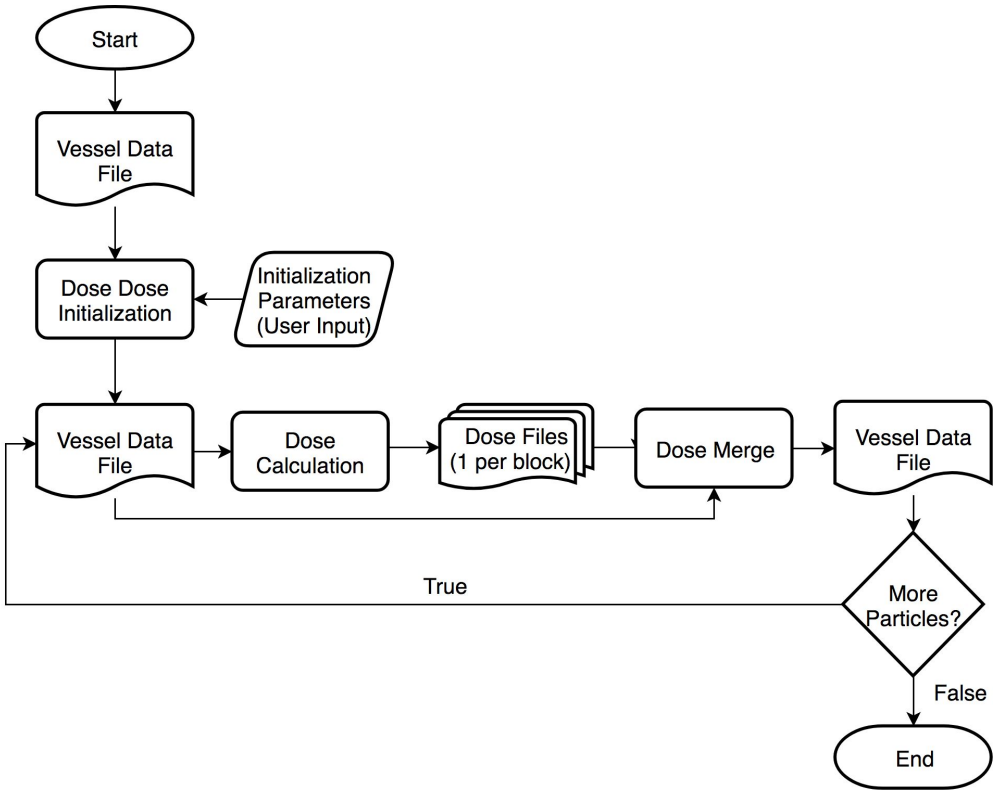


Figure 6.5. A flowchart of the dose calculation algorithm.

the simulation volume. The simulation volume was constructed by extended the vascular network bounding box to include the specified radiation field. The Dose Initialization application accepted command line arguments to specify the beam parameters and the size of vessel blocks (listed in Table 6.1). All proton beam and vessel block parameters were stored in the vessel data file.

The Dose Calculation application computed the dose from proton tracks to each block of vessels. The inputs for this application included the vessel data file to be processed, index of the vessel block to be processed, and the number of proton tracks to be simulated. To begin, the Dose Calculation application loaded the geometric description of its assigned vessel block from the vessel data file. For each vessel, we calculated the thickness of the layer of endothelial cells, t_{W}, by adapting the the equation published by Blanco *et al.* [104] to include a lower limit to the thickness, thus better modeling the the wall thickness of

Table 6.1. List of input parameters and their acceptable domains of the Dose Initialization.

Parameter	Acceptable Range	Units	Description
field_size_width	0 to maximum extent of network	mm	Specify field width
field_size_height	0 to maximum extent of network	mm	Specify field height
field_dir_x	0 to 1	arb.	x component of direction unit vector
field_dir_y	0 to 1	arb.	y component of direction unit vector
field_dir_z	0 to 1	arb.	z component of direction unit vector
field_center_x	x extents of network	mm	x component of field center
field_center_y	y extents of network	mm	y component of field center
field_center_z	z extents of network	mm	z component of field center
energy	1 to 350	MeV	Nominal beam energy
num_chunks	1 to 32,000	arb.	Number of vessel blocks

capillaries [12, 77], or

$$t_{\mathrm{W}}\,(\mathrm{mm}) = \begin{cases} 0.001 & r_{\mathrm{V}} \leq 0.1 \text{ mm} \\ r_{\mathrm{v}}\left(0.2802 r_{\mathrm{v}}^{-0.5053} + 0.1324 r_{\mathrm{v}}^{-0.01114}\right) & r_{\mathrm{v}} > 0.1 \text{ mm} \end{cases} \tag{6.7}$$

where r_{v} is the inner radius of the vessel. Each vessel was surrounded by an oriented bounding box that encompassed the vessel's maximum dimensions. This was necessary to perform efficient calculations of the interactions of particle tracks and blood vessels. A bounding box was constructed to describe the maximum extents of the vessel block. Proton transport was performed using the methods described in Section 6.2.2.1. To minimize the memory usage of the application, the transport was stopped after 100,000 proton tracks intersected the bounding box of the vessel block or all proton tracks were simulated. Dose from the block of protons was calculated to each vessel in the vessel block (Section 6.2.2.2). If more protons required transport, the algorithm created another proton block and repeated the dose scoring process until all protons were transported. The doses calculated to the blood vessels were written a new temporary dose file for the processed vessel block. The dose calculations and radiation transport for each proton block were performed using shared-memory parallelism.

We used an open-source utility [204] to distribute the invocations of the Dose Calculation application across N_c compute nodes, facilitating the parallel processing of the vessel blocks. The utility provided load-balancing capabilities to efficiently use all compute nodes.

The third application (Dose Merge) combined the results of the Dose Calculation application's output files into the vessel data file. This was done by accumulating the doses from each temporary dose file into the vessel data file for the entire vascular network. This step ensured that the calculated doses were correctly combined during the parallel computations. After calling this application, the vessel data file contained all dose information and the temporary dose files were deleted.

6.2.3. Biological Response Modeling

We modeled the radiation-induced injury to a blood vessel as a uniform reduction in radius throughout an entire vessel. For simplicity, we assumed that the length and wall thickness of each vessel remained constant. The relationship between the healthy and damaged radii of a vessel was

$$r_d = r_h Q(D) \tag{6.8}$$

where r_d is the radius of a damaged vessel, r_h is the radius of a healthy vessel, and $Q(D)$ is the damage factor at dose D. D was the mean vessel dose.

To determine the damage factor, we fit experimental data for the relative change in vessel surface area of rabbit ear vasculature (Figure 6.6) [40]. The data were stratified into capillary and larger vessel strata, and each strata was fit separately The damage factor, $Q(D)$, was

$$Q(D) = \begin{cases} 0.601 + 0.399e^{-0.00256D} & \text{for } r_h \leq 10 \text{ µm} \\ \left(-2.461 \times 10^{-11}\right) D^3 + \left(1.306 \times 10^{-7}\right) D^2 & \\ \qquad - \left(1.605 \times 10^{-4}\right) D + 0.988 & \text{for } r_h > 10 \text{ µm} \end{cases} \tag{6.9}$$

where D was the mean dose to the blood vessel in cGy.

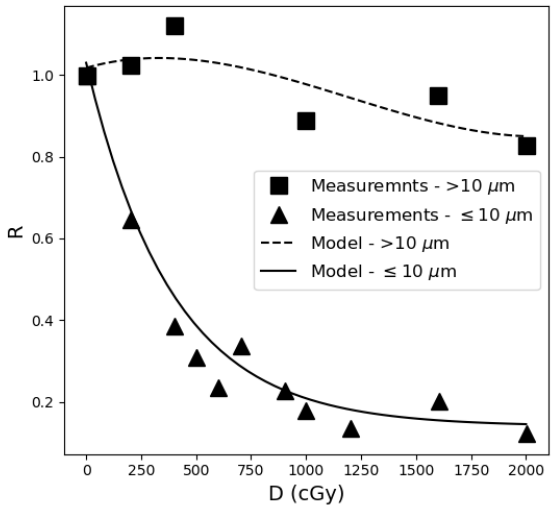


Figure 6.6. A plot of the dose response model (lines) and the experimental measurements (markers) from Dimitrievich *et al.* [40] versus dose. The solid line and triangles correspond to the capillary data, while the dashed line and squares correspond to the larger vessel data.

Equations 6.8 and 6.9 were implemented in the Dose Merge application. The damaged vessel radii were stored in the vessel data file for subsequent use in the blood-flow rate calculations.

6.2.4. Measurements of Computational Feasibility

6.2.4.1. Computational Resources

We used the SuperMIC cluster at Louisiana State University to perform all the computations in this work. This shared resource consists of 360 compute nodes, each containing two 10-core processors (2.8 GHz Xeon E5-2680, Intel Corporation, Santa Clara, CA, United States) and 64 GB of electronic memory. A high-performance network system (Infiniband Interconnect, [98]) connects the compute nodes and provides access to a persistent storage system (Lustre file system [99]), that enables up to 16 compute nodes to simultaneously read from and write to the same file.

We compiled all software used in this study using commercial C++ compilers (Intel C++ Compiler Cluster Edition v18.0, Intel Corporation, Santa Clara, CA, United States). The

compiler provided an implementation of OpenMP [94] for use in shared-memory parallelism and implementation of the message passing interface (MPI) for all distributed memory communication (Intel MPI v18.0, Intel Corporation, Santa Clara, CA, United States).

All input and output (I/O) operations to persistent storage utilized the Hierarchical Data Format version 5 (HDF5) library [139]. Our applications used the parallel I/O capabilities of the HDF5 library and the Lustre file system to increase speed. The library was tuned to the local file system following the suggestions of Howison *et al.* [97].

6.2.4.2. Execution Time

Execution time is an important performance characteristic of computationally demanding applications. We developed an empirical formula to predict the execution times of the dose calculation algorithm (T) based on three simulation parameters: number of vessels (N_v), number of particles (N_p), and nominal beam energy (E_0). To estimate the execution time we used

$$T = T_0 \cdot V(N_\mathrm{v}) \cdot P(N_\mathrm{p}) \cdot F(E_0, n_s(E_0, N_\mathrm{v})) \tag{6.10}$$

where T_0 is the baseline execution time in core hours, $V(N_\mathrm{v})$ is the vessel scaling function, $P(N_\mathrm{p})$ is the particle flux scaling function, and $F(E_0, n_s(E_0, N_\mathrm{v}))$ is the particle energy scaling function. T_0 was the execution time to perform a simulation with 2 million vessels and 100,000 100-MeV protons. The scaling factors adjust the baseline time for different simulation-parameter combinations. The vessel scaling function, $V(N_\mathrm{v})$, adjusted the time for the number of vessels and was

$$V(N_v) = aN_\mathrm{v}^b + c \tag{6.11}$$

where a, b, and c are empirical fitting parameters. The particle flux scaling function, $P(N_\mathrm{p})$, adjusted the time for the number of protons simulated. The equation was

$$P(N_\mathrm{p}) = fN_\mathrm{p}^g + h \tag{6.12}$$

where f, g, and h are empirical fitting parameters. $F(E_0, n_\mathrm{s})$ scales the baseline execution time with the initial proton energy. We empirically determined the form of this scaling function to be

$$F(E_0, n_\mathrm{s}) = (kE_0^m + jE_0^q) \cdot \log(n_s(E_0, N_\mathrm{v})) \tag{6.13}$$

where n_s is the average number of steps in a particle track and k, m, j, and q are empirical fitting parameters. n_s was determined by counting the number of steps a nominal energy proton needed to cross the simulation volume or run out of energy.

Each scaling function was trained independently to quantify its effect on the execution time. Data was collected for each function by changing only the corresponding dependent variable from its baseline condition. We collected each measurement three times and computed the average to account for system variations. Fits were performed using an open-source least-squares fitting algorithm (curve_fit, Numpy v1.13, [205]). The vessel scaling data was collected for $14 \leq N_\mathrm{V} \leq 4 \times 10^9$. Multiple compute nodes were required for calculations with $N_\mathrm{V} > 33.5 \times 10^6$. The particle flux scaling data was obtained for $100 \leq N_\mathrm{P} \leq 10^6$ on a single compute node. The initial proton energy data was collected on a single compute node for nominal proton energies of $1 \leq E_0 \leq 350$ MeV.

We validated equation 6.10 against an independent set of execution time trials. These measurements were selected to probe a wide range of parameter combinations (Table 6.2). We calculated absolute and relative difference between the predicted and observed execution times to determine the accuracy of the execution time model.

Table 6.2. List of input parameters for the validation trials. N_v is the number of vessels in the network, N_p is the number of protons tracks, E_0 is the nominal beam energy of the protons, N_c is the number of compute nodes, and N_b is the number of vessel blocks.

Trial No.	N_v	N_p	E_0 (MeV)	N_c	N_b
1	126	100	10	1	16
2	1022	16,000	350	4	64
3	4094	100,000,000	75	32	512
4	16,382	300,000	350	1	16
5	262,142	10	250	2	32
6	2,097,152	1,000	50	1	16
7	4,194,302	200,000	1	1	16
8	4,194,302	1,000,000,000	140	128	2048
9	4,294,967,294	2,000,000	180	128	2048
10	8,589,934,590	500,000	180	128	4096

6.2.4.3. Computational Scalability

Computational scalability is an important performance characteristic of relevance to applications that utilize parallel computations. Computational scalability was characterized by two widely-used metrics known as weak and strong scaling [144].

Weak scaling characterizes the execution time as the number of processors increases and the amount of work per processor remains constant. In this work, this means that the number of vessels doubled as the number of processors doubled. In the ideal case, the execution time remains constant. This metric provides information on how the application scales with problem size.

Strong scaling is a metric to characterize the ability for additional compute nodes to decrease the execution time of a fixed problem size. The primary metric of strong scalability is speedup (S), or

$$S = \frac{T_1}{T_n} \tag{6.14}$$

where T_1 and T_n are the execution times of one and n compute nodes, respectively. We used 1 to 128 compute nodes to simulate the dose to the blood vessels for two different networks sizes: 2 million and 33 million vessels. The 2-million-vessel network was used to probe the

effects of problem size on strong scaling. A 33-million-vessel network was the maximum size that could be processed on a single compute node. If the speedup factor is equal to the ratio of the number of compute nodes (n) then the algorithm is considered to scale strongly.

Scalability of the Dose Calculation application was assessed because it predominated the overall execution time. The scalability was only measured relative to the number of vessels simulated, *i.e.*, the number of particles and initial energy remained constant for each measurement.

6.2.4.4. Memory Usage

Memory usage in a computer program is an important factor to understanding the performance of an application on a single compute node. Ideally, applications use as little memory as possible and in the worst case as much as is available on the system. Memory usage was measured for all three applications using an open-source profiling tool [100] for multiple trials (Table 6.3). For the Dose Calculation application, we recorded the peak and mean memory usage per vessel block processed. The peak and mean memory usage per compute node was recorded for the Dose Initialization and Dose Merge applications. The peak memory usage was used to determine if the application was ever limited by the amount of system memory. The mean memory usage was used as a measure of how much memory was needed to run the application.

Table 6.3. List of input parameters for the memory profiling trials. N_{v} is the number of vessels in the network, N_{p} is the number of protons tracks, E_0 is the nominal beam energy of the protons, N_{c} is the number of compute nodes, and N_{b} is the number of vessel blocks.

Trial No.	N_{v}	N_{p}	E_0 (MeV)	N_{c}	N_{b}
1	32,768	10,000	350	1	16
2	32,768	200,000	10	2	32
3	33,554,432	1,000	250	1	16
4	33,554,432	10,000	150	2	32
5	2,147,483,648	10,000	150	128	2048

6.2.5. Simulation of Radiogenic Changes of Blood Flow in an Entire Organ

To evaluate the potential utility of the dose algorithm in, for example, a radiobiological study, we calculated the changes to blood flow in an organ-size vasculature following partial-organ irradiation. To achieve this, we assembled a modular workflow of algorithms (Figure 6.7). This workflow combined methods to construct vascular geometry (Chapter 2) [132], calculate dose and change in radius (Section 6.2.2), and calculate the blood flow rates through whole-organ vascular networks (Chapter 3) [160]. The details of the geometry algorithm were reviewed briefly in Section 6.2.1.

For the convenience of the reader, a brief review of the blood flow algorithm is presented here. Blood-flow rates in the vascular network were calculated using a steady-state blood-flow model. A steady-state model ignores the time-dependent evolution of blood flow and assumes laminar flow. The steady-state blood flow is described by Pouiseuille's equation, or

$$Q = \frac{\pi r^4}{8\eta l} \left(P_{\text{in}} - P_{\text{out}} \right) \tag{6.15}$$

where Q is the volumetric flow rate, r is the vessel's inner radius, η is the viscosity of the blood, l is the length of the vessel, and P_{in} and P_{out} are the pressures at the input and output of the vessel, respectively [21]. Applying this equation to all blood vessels in a network resulted in a system of linear equations, which we solved using numerical techniques [134]. We used this model to calculate the blood-flow rates through unirradiated and irradiated vessels.

We evaluated the mean execution time to simulate changes in blood flow rates for whole human brains and whole rodent brains. Table 6.4 lists the relevant simulation parameters used in the study. To complete the computations in a network with 8.5 billion vessels, the number of blocks per compute node was set to 32 in equation 6.6. This was necessary to overcome the limitations in the I/O library used. We measured the overall execution time of the workflow three times for each trial configuration.

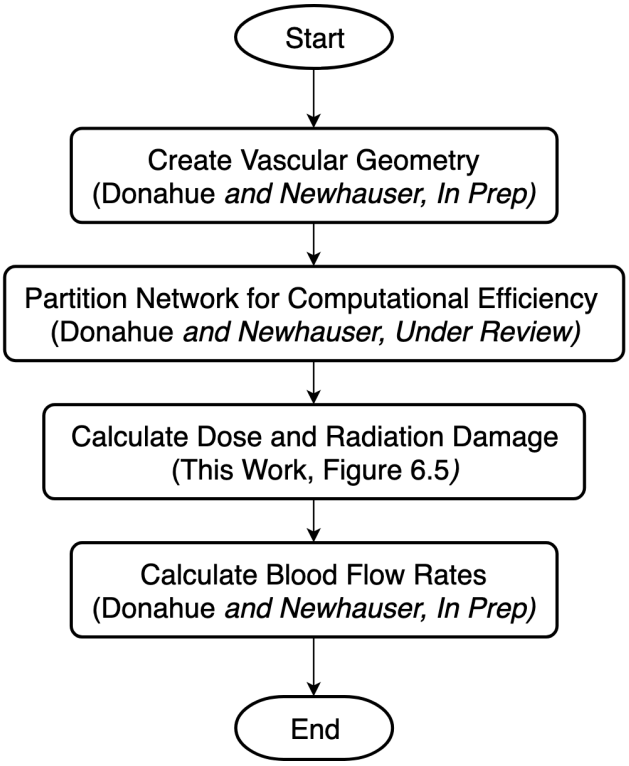


Figure 6.7. A flowchart of the workflow to simulate changes in blood flow due to radiation damage.

6.3. Results

6.3.1. Dose Calculation Feasibility

6.3.1.1. Execution Time

We fit the scaling function (Equations (6.10) to (6.13)) to the training data to determine the empirical parameters of the model of execution time (Table 6.5 and Figure 6.8a-c) with

Table 6.4. List of input parameters for the workflow trials. N_{v} is the number of vessels in the network, N_{p} is the number of protons tracks, E_0 is the nominal beam energy of the protons, N_{c} is the number of compute nodes, and N_{b} is the number of vessel blocks.

Trial No.	Model	N_{v}	N_{p}	E_0 (MeV)	N_{c}	N_{b}
Visualization Only		126	10000	80	1	16
1	Mouse Brain	4,194,302	1,000,000,000	140	128	2048
2	Human Brain	4,294,967,294	2,000,000	180	128	2048
3	Human Brain	8,589,934,590	2,000,000	180	128	4096

Table 6.5. Fitting parameters for the timing model to predict dose calculation times (See Equations (6.10) to (6.13))

Parameter	Scaling Factor	Value
T_0	Baseline Time	9.96 core-hours
a	$V(N_{\mathrm{v}})$	1.113×10^{-5}
b	$V(N_{\mathrm{v}})$	0.716
c	$V(N_{\mathrm{v}})$	2.5×10^{-2}
f	$P(N_{\mathrm{p}})$	9.642×10^{-6}
g	$P(N_{\mathrm{p}})$	1.004
h	$P(N_{\mathrm{p}})$	1.113×10^{-2}
k	$F(E_0, n_s)$	5.315×10^{-2}
m	$F(E_0, n_s)$	7.973×10^{-1}
j	$F(E_0, n_s)$	3.160×10^{-5}
q	$F(E_0, n_s)$	2.125

good agreement. More specifically, the model agreed with measured times from the validation trials (Figure 6.8d) within a factor of 3.3 across approximately 5 decades of execution times. The discrepancies can be attributed, in part, to the correlations between the various scaling functions that were not taken into account, *e.g.*, vascular density and maximum δ-ray range. The agreement suggests the model to estimate the execution time is sufficiently accurate for designing computational experiments.

6.3.1.2. Computational Scalability

We characterized the strong and weak scaling of the Dose Calculation application (Figure 6.9). The weak scaling data (Figure 6.9a) shows a decreasing trend in execution time. Thus, for a fixed number of protons the application ran faster as the number of vessels and the number of processors increased. These results imply that the algorithm could efficiently use additional computational resources to decrease the execution time.

We studied strong scaling for two network sizes (2 million and 33 million vessels). Results revealed that strong scalability of the algorithm increased with problem size (Figure 6.9b). This can be attributed to the smaller fraction of time spent on I/O as the number of vessels per block increases. The data also shows that the speedup factors were supra-linear for

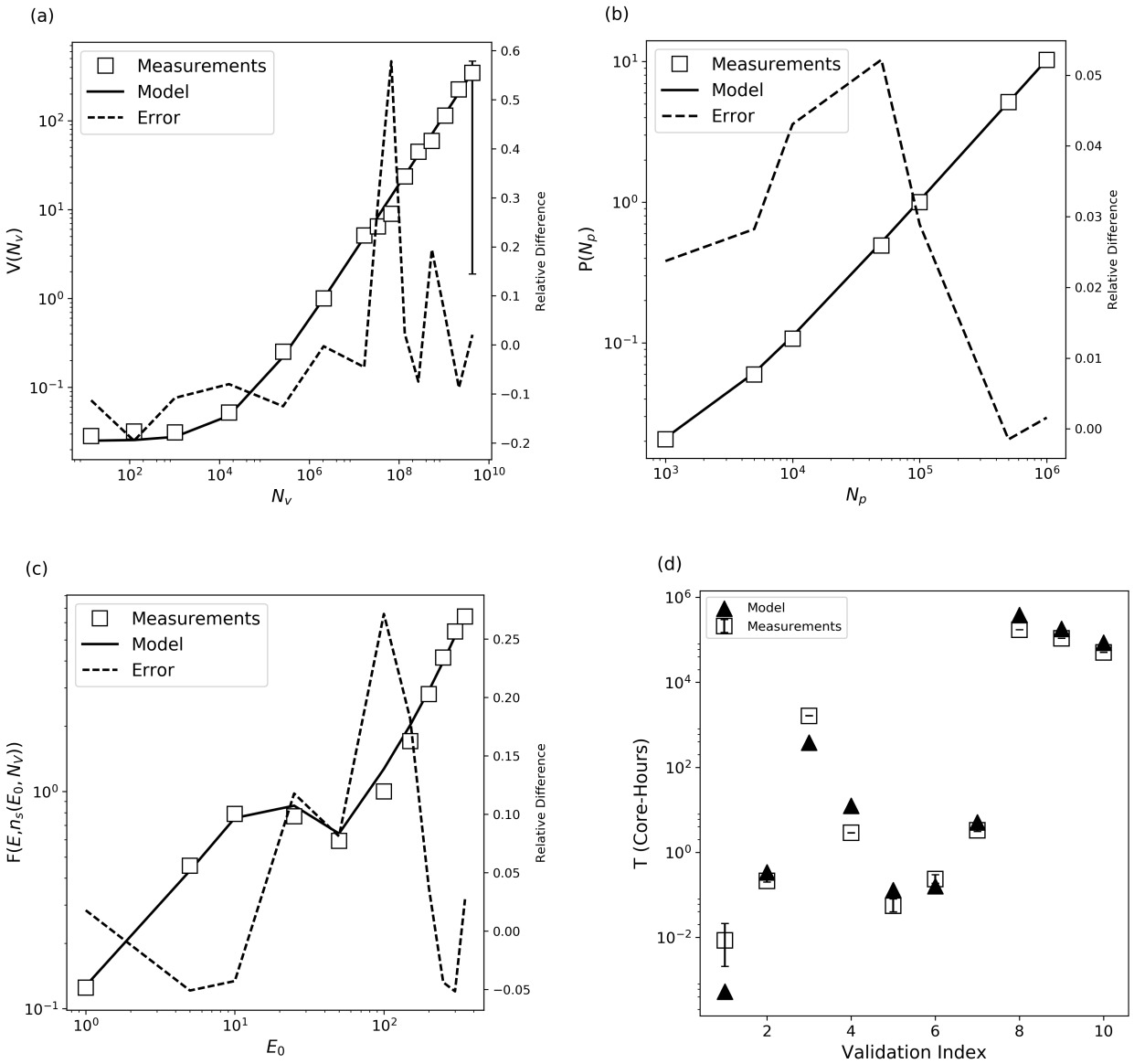


Figure 6.8. Plots of the three scaling functions along with the fitting data (a-c) and the predicted and observed execution times for the validation trials (d).

portions of the scalability curve. This is caused by the reduction in the number of interactions between proton tracks and the vessel block as the number of compute nodes increased. The downturn in scalability is due to a tradeoff between I/O time and the number of floating point operations required for each vessel block. This is indicative of an algorithm that efficiently uses its computational resources and is limited by the mathematical and logical operations and not by memory movement in the computer system.

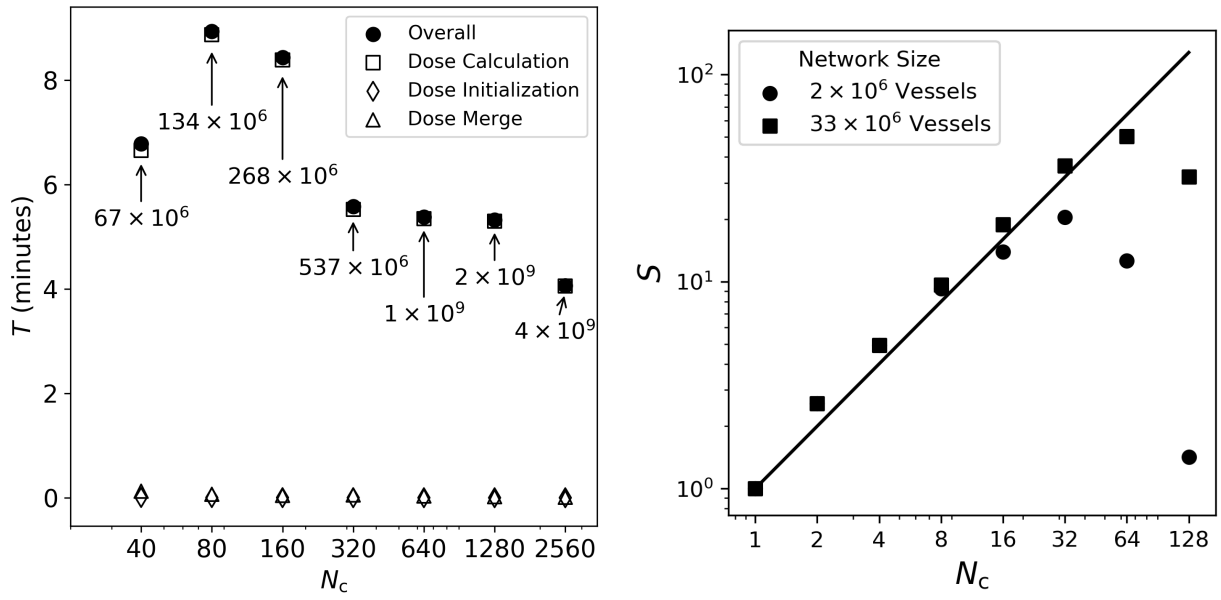


Figure 6.9. Plots of the weak scaling (a) and strong scaling (b) of the Dose Calculation application. Panel (a) plots the wall-clock time T in minutes versus the number of compute nodes. Panel (b) plots the speedup (S) verses the number of compute nodes (N_c). The solid line in Panel (b) represents linear scalability.

6.3.1.3. Memory usage

We quantified the memory usage for five trials (Table 6.6). The peak memory usage was 13.8 GB for the case of the largest vascular network tested which is significantly less than the 64 GB available on the compute nodes. This peak occurred during the initialization of the dose data structures in the vessel data file. For the Dose Calculation application, the peak memory usage was 6 GB for the largest vessel block. This low memory utilization can be attributed to the use of vessel blocks and particle blocks, which reduced the requirements of memory for every compute node (Section 6.2.2.3). It also suggests that the algorithm could be ported to GPU architectures for faster parallel computations. The average memory usage for all three applications was less than 5 GB, suggesting that the long computation times were caused by the arithmetic and logical operations and not memory movement between system memory and the processor cache.

Table 6.6. Memory profiling measurements for Dose Calculation algorithms.

	Memory usage (GB)					
	DoseInitialization[a]		DoseCalculation[b]		DoseMerge[a]	
Trail No.	Mean	Peak	Mean	Peak	Mean	Peak
1	0.50	2.25	0.15	0.28	0.80	1.19
2	0.32	2.32	0.20	1.87	0.62	1.28
3	4.45	4.64	2.23	5.95	4.40	5.22
4	3.04	5.13	1.28	3.10	3.00	3.33
5	3.76	13.76	1.23	3.20	2.77	3.33

[a]: Memory usage per compute node
[b]: Memory usage per vessel block

6.3.2. Simulation of Radiogenic Changes of Blood Flow in an Entire Organ

We measured the computation time for the complete workflow of algorithms (Figure 6.7) for 3 different network sizes (Table 6.4). Qualitatively, the results of the computations may be visualized in Figure 6.10, which plots the dose, change in vessel radius, and change in blood-flow rate. The dose was deposited in a highly stochastic pattern in the vascular network (Figure 6.10a). The highest observed dose was cause by a proton track directly crossing through a blood vessel. This stochastic dose deposition led to highly variable changes in the vessel radii, from complete closure to a 5% increase in the radius (Figure 6.10b). The changes in vessel radii resulted in changes in blood flow both inside and outside the radiation field (Figure 6.10c). In particular, the high doses received in some segments of a few vessel led to large changes in blood flow throughout the vascular network. This highlights the need to account for radiation damage at the microscopic level.

Table 6.7 lists the execution times for the mouse and human brain trials. It required 68.6 wall-clock hours to simulate the changes in blood flow from 1 billion protons ($E_0 = 140$ MeV) impinging on a mouse brain. The the trials of human brain networks executed in 49 hours for 4 billion vessels, or 88 hours for 8.5 billion vessels when simulating 2,000,000 protons ($E_0 = 180$ MeV). The execution time was predominated by the dose calculations in all three trials (98% of the execution time).

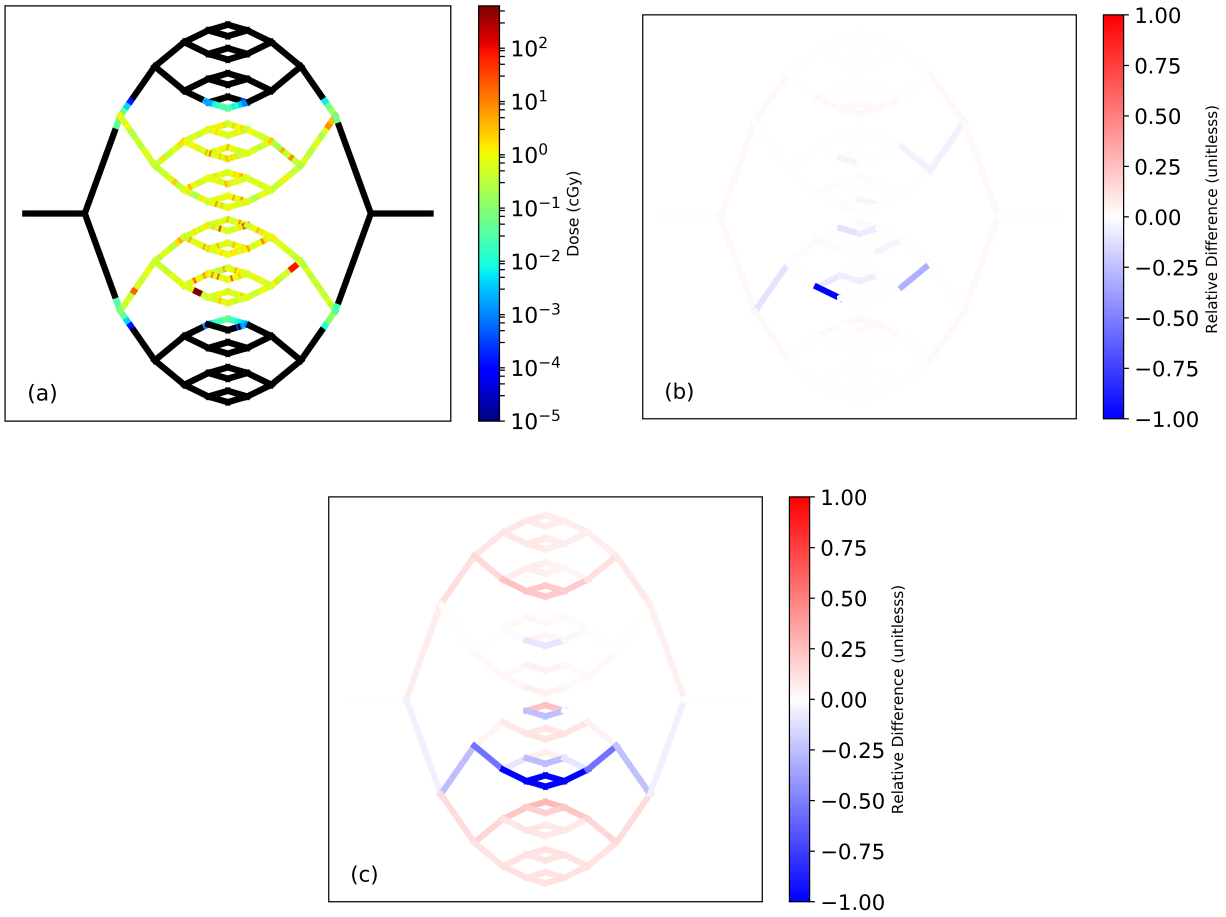


Figure 6.10. The qualitative results of the simulated dose distribution (a), the change in vessel radii (b), and the relative change in blood flow (c). The dose (a) and the dose response function (Figure 6.6) result in the changes in vessel radii (b). At doses less than 500 cGy and in vessels larger than 10 micrometers in radius, there is a slight increase in the vessel radius. The changes in vessel radii results in the blood flow patterns observed in (c).

We compared the observed execution time of the workflow with the predicted time for using our model of execution time for the dose calculations (see Section 6.2.4.2). The predicted Dose Calculation time over estimated the observed workflow time by 43% to 120%. We expected the times to be comparable, as the dose calculations predominated the execution time of the workflow. This suggests that the timing model can be used as a conservative estimate of the run time for the workflow and not just the dose calculations. This provides a useful tool when designing computational experiments and determining the amount of computational resources required.

Table 6.7. Measured and predicted executions times for workflow trials (Table 6.4).

Trial	Predicted Dose Calculation Time (core-h)	Measured Workflow Time (core-h)	Measured Workflow Time (wall-h)	$\frac{T_p - T_m}{T_m} \times 100$
1	384,322	175,616	68.6	118%
2	180,897	126,720	49.4	43%
3	344,845	223,328	87.2	54%

6.4. Discussion

We developed a recursive algorithm to calculate the radiation absorbed dose to blood vessels and characterized its computational performance. Combining the recursive dose model with previously published techniques for vascular geometry and blood flow modeling, we evaluated the utility of the recursive dose calculation algorithm to compute the changes in blood flow in vasculature of mouse and human brains. The major finding of this work is computing the radiation dose to whole-organ vascular networks is computationally feasible. The results of this work also show that it is computationally feasible to simulate the change in blood flow resulting from radiation injury. Computing the changes of blood flow in a human brain from 2 million protons required 2 to 4 days on a modest-sized cluster. Additionally, the dose distributions, changes in vessel radius, and changes in blood-flow rates demonstrate the effects local damage can have on regional blood flow. Using the model of execution time developed in this work, we estimate whole-organ dose calculations with up to a billion protons would require 10^8 core-hours or 58 days on a 4,000-node cluster.

The implication of this study is that new frontiers in computational research related to vascular injury and radiation biology appear to be opening. In the field of radiation therapy, this model could eventually find applications in routine treatment planning. Contemporary treatment planning methods calculate the dose to voxels, which are typically considered to be independent functional sub-units. Studying the dose to the vascular network enables consideration of connections in the tissue. For example, it could provide useful insights to necrosis [11], fibrosis [5], and cardiac toxicity [191], which are all linked to vascular damage. Additionally, this algorithm could enable the investigation of the implications of vascular

damage using statistical methods. The large number of blood vessels in a complete organ may naturally lend themselves to statistical measures. The workflow used in this study could be applied to the study of vascular injury from any cause. This was demonstrated for radiation in this work but the modular design of the workflow enables the use of other models of damage, such as blunt-force trauma [206] or surgical procedures. Additionally, the workflow could be extended beyond the blood-flow calculations to study diseases, such as Alzheimer's or Traumatic Brain Injury [72], both of which are dependent upon vascular function. That said, it must be understood that the present work demonstrated only the computational feasibility; the road to practical applications will be long and difficult.

The computational characterization of the dose calculation algorithm in this work are comparable to those of previous studies. Using Monte Carlo techniques, researchers have modeled the track-by-track dose deposition in voxelized tissues with a high degree of accuracy [30, 207, 208]. The use of parallel computing in these algorithms has made whole body calculations of dose achievable [32] and allowed for the routine clinical use of these algorithms [207, 208]. It is difficult to compare our work to these studies because of the drastic differences in anatomical geometry and proton transport used in this work. For example, Vadapalli *et al.* [30] calculated the dose from 25 million protons into a voxelized phantom with 2400 effective dose points in approximately 812 core-hours. Our model calculated the dose from 25 million protons to a 30-vessel network (2400 dose points) in 168 core-hours while simulating 25 million protons through 8.5 billion vessels using our model takes 800 million core-hours. This can be explained by the competing factors of a simplified proton transport model used in the dose calculations of our simulation and a more complex (vascular) geometry.

The major strengths of this work were the emphasis on the computational aspects of the simulations and the modular design of the workflow. This enabled the algorithms to be separated by function and it simplified the implementation. Additionally, it provides the opportunity for future extension. For example, simultaneous improvements to the dose and radiobioloigcal models could be developed in tandem by experts in the respective fields.

Another strength was the use of simplified models for vascular geometry and particle transport. In addition to enabling the algorithm to complete computations containing billions of individual vessels and particles, this approach enabled us to quantify the minimum computational requirements of the dose algorithm and workflow.

Our study had three major limitations. First, the calculations of the absorbed dose and blood flow rate were not validated using new experimental data. This is not a serious limitation because the purpose of this study was to explore questions surrounding computational feasibility using simple models, and not the ultimate degree of detail and accuracy that is achievable. Furthermore, the amorphous track structure model used in this work has been validated previously [35]. The blood flow calculations were validated against a commercially-available computational fluid dynamics package [209]. Future work will be needed to address the simple algorithms used in this study. For example, the radiation transport algorithm ignored lateral scatter, energy straggling, and nuclear interactions. This was one of the simplifications necessary to reduce the computation time of the dose calculations, enabling the study of dose distribution in whole-organ vascular networks. Another example is the biological model, which used a simplified damage model that only looked at a fixed end point. These limitations are the focus on ongoing work in our group. Finally, this study only used convetional CPUs for the computations. The parallel nature of the Dose Calculation application and its scalability suggest that it might benefit from the use of accelerators, *e.g.*, GPUs. This could be addressed in future implementations of the algorithm.

This study demonstrated, for the first time, that it is feasible to simulate the changes in blood flow through the whole-brain caused by vascular injury from localized irradiation. This new capability could eventually lead to models that improve the prediction and treatment of radiation-induced necrosis and other deleterious radiation damage.

7. Conclusion

In this work, we developed and tested algorithms to simulate the changes in blood flow due to vascular injury caused by irradiation in of an organ's vasculature. The algorithms included the capabilities to construct vascular geometries (Chapter 2), calculate the radiation absorbed dose (Chapter 6), and determine the blood flow rates through individual vessels (Chapter 3 & 4). The major finding of this work was that it is computationally feasible to simulate the effects of radiation on blood flow in a whole-organ vasculature. The algorithms required 89.5 hours to simulate 2 million protons incident on an 8.5 billion vessel network using 128 compute nodes. Importantly, these studies revealed the dose calculations were the most time consuming task. The vessel geometry algorithm and blood flow calculation algorithm both demonstrated the ability to reach 17 billion vessels, *i.e.* approximately the number in the human body [58].

This work has the potential open new lines of inquiry on a wide range of related topics including radiation late effects, circulating tumor cells, and the tumor microenvironment. For the first time, we are able to consider the cumulative impact of vascular injury to capillary beds on an organ's blood flow. This could lead to improvements in modeling drug distributions, which are dependent on blood flow for delivery. For example, chemotherapy is often given in conjunction with radiation therapy. The effects of chemotherapy assume that the vascular network is intact and the drug is able to make it to the tumor. However, radiation is damaging the vasculature during treatment, and may cause the drug to be redirected to a healthy region of the organ. The work presented here is a major step toward understanding the impact and extent of this effect on cancer treatment. This work could also be used to investigate the relationship between radiation injury of the vasculature, tumor control, and normal tissue toxicities. The algorithms in this work could be used to generate hypotheses prior to embarking on experimental research on biological models. With future improvements, the algorithms could also be used to study different types of damage to the vasculature.

It is difficult to draw direct comparisons of this work as a whole with the existing literature. However, each component of this work can be directly compared to literature from its associated field. We built our vascular geometry on existing fractal-based techniques [22–24], which are less realistic than more advanced techniques [103, 105, 106], but are scalable far beyond the 360,000 vessels previously achieved [21]. Because goal of this work was to study the computational aspects of modeling whole-organ vasculature, here it is more important to model the correct number of vessels than to achieve a highly-accurate geometry. The blood flow modeling component used in this work is the well-established steady-state approach for calculations in whole-organ vascular networks [21, 126, 146, 147]. This technique enabled us to take advantage of improvements in numerical techniques for solving matrices to scale to billions of vessels [134]. While more advanced fluid dynamics methods exist (*i.e.* Naiver-Stokes equations coupled with finite element meshing), their computational expense makes their implementation at whole-organ scales unfeasible [56, 104, 119]. Our validation of the blood flow model was coherent with existing literature. We achieved an average accuracy of 5% compared to the 5% difference observed for 7 different models of of aneurysm blood flow [165]. Our stopping power model was able to fit stopping power and range data with 10%, which is consistent with previous attempts [174, 175, 184]. Furthermore, it is the only model available to provide a directly integrable relation to calculate range from stopping power. The algorithm for determining the dose to the vessels was inspired by the local effects model (LEM) [35]. LEM determines the biological effect of radiation in normal tissues by calculating the dose to a random set of cells in each dosimetric voxel using radial dose models. Our model used the same radial dose model, which supports its use in multi-dimensional simulations. Previous work on tumor modeling has studied the effect of radiation on blood flow in small vascular networks [210–215]. Most of the work has been on characterizing the blood flow using imaging techniques for determining clinical prognoses. While some computational modeling on tumor blood flow was done in these studies, none simulated the radiation dose deposited to the vessels from individual particle tracks, but instead applied a uniform dose

to all vessels in a volume. Additionally, none provided the ability to model the blood flow in all of the tissue surrounding the tumor.

The primary strength of this work was the modular design of the workflow. This simplified the implementation of each component by separating the responsibilities among the applications. Another benefit of this approach is the ability for future modifications of the algorithms to happen in parallel. For example, the radiation transport algorithm could be improved to incorporate scattered protons while others improve the blood flow model. In this way all researchers using the workflow can benefit from improvements made by experts in each field. A second strength was the use of simple computational methods within each algorithm. This provides an estimate of the minimum performance requirements to simulate a problem of a given size, more realistic simulations will require more computational resources. This information is critical to designing future studies for simulating whole-organ or whole-organism biological processes. Using simplified methods also increases the reproducibility and transparency of the algorithms.

Future studies are necessary to address the limitations of this work. The primarily limitation is that only the blood flow component was validated in this work. This limitation had a diminished impact on the current work because until an algorithm was designed and tested for computational feasibility, it was unknown what level of approximations needed to be validated in the model. Additionally, the blood flow and dose algorithms used well-established and validated approaches from the literature [21, 35]. The second major limitation was the simplistic model of vasculature, *e.g.*, vessels as straight tubes and a symmetric branching ratios. This model was necessary because it is currently the only model demonstrably capable of reaching 9 billion vessels. The simplicity was not a major limitation for the blood flow calculations because the steady-state model neglected the effects of vessel shape on blood flow. Finally, to simulate blood flow and dose deposition in multi-billion vessel networks, it was necessary to use many simplifying assumptions, *e.g.*, steady-state blood flow

and the straight-ahead approximation. This limitation will be overcome in future studies by developing new algorithms and optimized code to perform the computations.

This work demonstrated that it is feasible to simulate the effects of vascular injury on whole-organ blood flow with current computational technologies, although simplifications are necessary. Using the techniques developed in this work, it should now be possible to synthesize new hypotheses pertaining to vascular damage and sequelae, such as necrosis. In the future, this may eventually lead to safer treatments and better outcomes for patients.

References

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2019", CA: A Cancer Journal for Clinicians **69**, 7–34 (2019).

[2] G. T. Armstrong, T. Kawashima, W. Leisenring, K. Stratton, M. Stovall, M. M. Hudson, C. A. Sklar, L. L. Robison, and K. C. Oeffinger, "Aging and risk of severe, disabling, life-threatening, and fatal events in the childhood cancer survivor study", Journal of Clinical Oncology **32**, 1218–1227 (2014).

[3] D. R. Smart, *Physician Characteristics and Distribution in the US 2010* (Amer Medical Assn, 2009).

[4] W. D. Newhauser and M. Durante, "Assessing the risk of second malignancies after modern radiotherapy.", Nature reviews. Cancer **11**, 438–48 (2011).

[5] J. Yarnold and M. C. Vozenin-Brotons, "Pathogenetic mechanisms in radiation fibrosis", Radiotherapy and Oncology **97**, 149–161 (2010).

[6] S. T. Chao, M. S. Ahluwalia, G. H. Barnett, G. H. J. Stevens, E. S. Murphy, A. L. Stockham, K. Shiue, and J. H. Suh, "Challenges with the diagnosis and treatment of cerebral radiation necrosis", International Journal of Radiation Oncology Biology Physics **87**, 449–457 (2013).

[7] N. Festjens, T. Vanden Berghe, and P. Vandenabeele, "Necrosis, a well-orchestrated form of cell demise: Signalling cascades, important mediators and concomitant immune response", Biochimica et Biophysica Acta - Bioenergetics **1757**, 1371–1387 (2006).

[8] G. Kroemer, L. Galluzzi, P. Vandenabeele, J. Abrams, E. S. Alnemri, E. H. Baehrecke, M. V. Blagosklonny, W. S. El-Deiry, P. Golstein, D. R. Green, M. Hengartner, R. A. Knight, S. Kumar, S. A. Lipton, W. Malorni, G. Nuñez, M. E. Peter, J. Tschopp, J. Yuan, M. Piacentini, B. Zhivotovsky, G. Melino, and Nomenclature Committee on Cell Death 2009, "Classification of cell death: recommendations of the Nomenclature Committee on Cell Death 2009.", Cell death and differentiation **16**, 3–11 (2009).

[9] S. L. Fink and B. T. Cookson, "Apoptosis, pyroptosis, and necrosis: Mechanistic description of dead and dying eukaryotic cells", Infection and Immunity **73**, 1907–1916 (2005).

[10] G. Rahmathulla, N. F. Marko, and R. J. Weil, "Cerebral radiation necrosis: A review of the pathobiology, diagnosis and management considerations", Journal of Clinical Neuroscience **20**, 485–502 (2013).

[11] N. Lyubimova and J. W. Hopewell, "Experimental evidence to support the hypothesis that damage to vascular endothelium plays the primary role in the development of late radiation-induced CNS injury", British Journal of Radiology **77**, 488–492 (2004).

[12] F. Lauwers, F. Cassot, V. Lauwers-Cances, P. Puwanarajah, and H. Duvernoy, "Morphometry of the human cerebral cortex microcirculation: General characteristics and space-related profiles", NeuroImage **39**, 936–948 (2008).

[13] F. Cassot, F. Lauwers, S. Lorthois, P. Puwanarajah, V. Cances-Lauwers, and H. Duvernoy, "Branching patterns for arterioles and venules of the human cerebral cortex", Brain Research **1313**, 62–78 (2010).

[14] M. Peyrounette, Y. Davit, M. Quintard, and S. Lorthois, "Multiscale modelling of blood flow in cerebral microcirculation: Details at capillary scale control accuracy at the level of the cortex", PLoS ONE **13**, edited by J. Boltze (2018) `10.1371/journal.pone.0189474`.

[15] P. S. Tsai, B. Friedman, A. I. Ifarraguerri, B. D. Thompson, V. Lev-Ram, C. B. Schaffer, Q. Xiong, R. Y. Tsien, J. A. Squier, and D. Kleinfeld, "All-optical histology using ultrashort laser pulses", Neuron **39**, 27–41 (2003).

[16] I. G. Gould, P. Tsai, D. Kleinfeld, and A. Linninger, "The capillary bed offers the largest hemodynamic resistance to the cortical blood supply", Journal of cerebral blood flow and metabolism **37**, 52–68 (2016).

[17] S. Heinzer, T. Krucker, M. Stampanoni, R. Abela, E. P. Meyer, A. Schuler, P. Schneider, and R. Müller, "Hierarchical microimaging for multiscale analysis of large vascular networks", NeuroImage **32**, 626–636 (2006).

[18] D. Mayerich, L. Abbott, and B. McCormick, "Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain", Journal of Microscopy **231**, 134–143 (2008).

[19] W. Schreiner, M. Neumann, F. Neumann, S. M. Roedler, A. End, P. Buxbaum, M. R. Müller, and P. Spieckermann, "The branching angles in computer-generated optimized models of arterial trees", The Journal of general physiology **103**, 975–989 (1994).

[20] R. Karch, F. Neumann, M. Neumann, and W. Schreiner, "A three-dimensional model for arterial tree representation, generated by constrained constructive optimization", Computers in Biology and Medicine **29**, 19–38 (1999).

[21] A. A. Linninger, I. G. Gould, T. Marinnan, C.-Y. Y. Hsu, M. Chojecki, and A. Alaraj, "Cerebral microcirculation and oxygen tension in the human secondary cortex", Annals of Biomedical Engineering **41**, 2264–2284 (2013).

[22] M. S. Olufsen, "Structured tree outflow condition for blood flow in larger systemic arteries.", American Journal of Physiology-Heart and Circulatory Physiology **276**, H257–H268 (1999).

[23] J. C. Parker, C. B. Cave, J. L. Ardell, C. R. Hamm, and S. G. Williams, "Vascular tree structure affects lung blood flow heterogeneity simulated in three dimensions", Journal of Applied Physiology **83**, 1370–1382 (1997).

[24] P. Perdikaris, L. Grinberg, and G. E. Karniadakis, "An effective fractal-tree closure model for simulating blood flow in large arterial networks", Annals of Biomedical Engineering **43**, 1432–1442 (2015).

[25] B. N. Steele, M. S. Olufsen, and C. A. Taylor, "Fractal network model for simulating abdominal and lower extremity blood flow during resting and exercise conditions", Computer Methods in Biomechanics and Biomedical Engineering **10**, 39–51 (2007).

[26] W. Schreiner, R. Karch, M. Neumann, F. Neumann, P. Szawlowski, and S. Roedler, "Optimized arterial trees supplying hollow organs", Medical Engineering and Physics **28**, 416–429 (2006).

[27] M. Zagzoule and J. P. Marc-Vergnes, "A global mathematical model of the cerebral circulation in man.", Journal of biomechanics **19**, 1015–1022 (1986).

[28] P. Andreo, "The physics of small megavoltage photon beam dosimetry", Radiotherapy and Oncology **126**, 205–213 (2017).

[29] N. T. Henthorn, J. W. Warmenhoven, M. Sotiropoulos, R. I. Mackay, K. J. Kirkby, and M. J. Merchant, "Nanodosimetric simulation of direct ion-induced DNA damage using different chromatin geometry models", Radiation Research **188**, 690–703 (2017).

[30] R. Vadapalli, P. Yepes, W. Newhauser, and R. Lichti, "Grid-enabled treatment planning for proton therapy using monte carlo simulations", Nuclear technology **175**, 16–21 (2011).

[31] N. Qin, C. Shen, M.-Y. Tsai, M. Pinto, Z. Tian, G. Dedes, A. Pompos, S. B. Jiang, K. Parodi, and X. Jia, "Full Monte Carlo-based biologic treatment plan optimization system for intensity modulated carbon ion therapy on graphics processing unit", International Journal of Radiation Oncology Biology Physics **100**, 235–243 (2018).

[32] W. D. Newhauser, J. D. Fontenot, A. Mahajan, D. Kornguth, M. Stovall, Y. Zheng, P. J. Taddei, D. Mirkovic, R. Mohan, J. D. Cox, and S. Woo, "The risk of developing a second cancer after receiving craniospinal proton irradiation", Physics in Medicine and Biology **54**, 2277–2291 (2009).

[33] J. D. Fontenot, A. K. Lee, and W. D. Newhauser, "Risk of secondary malignant neoplasms from proton therapy and intensity-modulated x-ray therapy for early-stage prostate cancer", International Journal of Radiation Oncology Biology Physics **74**, 616–622 (2009).

[34] M. P. R. Waligórski, R. N. Hamm, and R. Katz, "The radial distribution of dose around the path of a heavy ion in liquid water", International Journal of Radiation Applications and Instrumentation. Part **11**, 309–319 (1986).

[35] T. Elsässer, R. Cunrath, M. Krämer, M. Scholz, E. Thilo, C. Richard, K. Michael, and S. Michael, "Impact of track structure calculations on biological treatment planning in ion radiotherapy", New Journal of Physics **10**, 75005 (2008).

[36] H. Wang and O. N. Vassiliev, "Radial dose distributions from protons of therapeutic energies calculated with Geant4-DNA", Physics in Medicine and Biology **59**, 3657–3668 (2014).

[37] J. P. Williams and W. Newhauser, "Normal tissue damage: its importance, history and challenges for the future", The British Journal of Radiology **92** (2018) `10.1259/bjr.20180048`.

[38] M. Guerrero and D. J. Carlson, "A radiobiological model of reoxygenation and fractionation effects", Medical Physics **44**, 2002–2010 (2017).

[39] R. D. Stewart, D. J. Carlson, M. P. Butkus, R. Hawkins, T. Friedrich, and M. Scholz, "A comparison of mechanism-inspired models for particle relative biological effectiveness (RBE)", in Medical physics, Vol. 45, 11 (2018), e925–e952.

[40] G. S. Dimitrievich, K. Fischer-Dzoga, and M. L. Griem, "Radiosensitivity of vascular tissue. I. Differential radiosensitivity of capillaries: a quantitative in vivo study.", Radiation research **99**, 511–35 (1984).

[41] K. Fischer-Dzoga, G. S. Dimitrievich, and M. L. Griem, "Radiosensitivity of vascular tissue. II. Differential radiosensitivity of aortic cells in vitro", Radiat Res **99**, 536–546 (1984).

[42] N. M. Roth, M. R. Sontag, and M. F. Kiani, "Early effects of ionizing radiation on the microvascular networks in normal tissue", Radiation Research **151**, 270–7 (1999).

[43] V. Nguyen, M. W. Gaber, M. R. Sontag, and M. F. Kiani, "Late effects of ionizing radiation on the microvascular networks in normal tissue.", Radiation research **154**, 531–6 (2000).

[44] J. a. Coderre, G. M. Morris, P. L. Micca, J. W. Hopewell, I. Verhagen, B. J. Kleiboer, and A. J. van der Kogel, "Late effects of radiation on the central nervous system: role

of vascular endothelial damage and glial stem cell survival.", Radiation research **166**, 495–503 (2006).

[45] A. Mizrachi, A. P. Cotrim, N. Katabi, J. B. Mitchell, M. Verheij, and A. Haimovitz-Friedman, "Radiation-induced microvascular injury as a mechanism of salivary gland hypofunction and potential target for radioprotectors", Radiation Research **186**, 189–195 (2016).

[46] A. Quarteroni, A. Veneziani, and C. Vergara, "Geometric multiscale modeling of the cardiovascular system, between theory and practice", Computer Methods in Applied Mechanics and Engineering **302**, 193–252 (2016).

[47] M. Imani, A. M. Goudarzi, D. D. Ganji, and A. L. Aghili, "The comprehensive finite element model for stenting: the influence of stent design on the outcome after coronary stent placement", Journal of Theoretical and Applied Mechanincs **51**, 639–648 (2013).

[48] G. S. Karanasiou, E. E. Tripoliti, E. R. Edelman, L. K. Michalis, and D. I. Fotiadis, "Stent deployment computer based simulations for health care treatment of diseased arteries", in *Concepts and trends in healthcare information systems*, Vol. 16, edited by D.-D. Koutsouris and A. A. Lazakidou (Springer International Publishing, 2014) Chap. 10, pp. 143–167.

[49] I. V. Gomes, H. Puga, J. L. Alves, and J. C. Claro, "Finite element analysis of stent expansion: Influence of stent geometry on performance parameters", in Enbeng 2017 - 5th portuguese meeting on bioengineering, proceedings (2017), pp. 1–4.

[50] D. Tang, C. Yang, J. Zheng, P. K. Woodard, J. E. Saffitz, G. A. Sicard, T. K. Pilgram, and C. Yuan, "Quantifying effects of plaque structure and material properties on stress distributions in human atherosclerotic plaques using 3D FSI models", Journal of Biomechanical Engineering **127**, 1185 (2005).

[51] X. Leng, F. Scalzo, H. L. Ip, M. Johnson, A. K. Fong, F. S. Fan, X. Chen, Y. O. Soo, Z. Miao, L. Liu, E. Feldmann, T. W. Leung, D. S. Liebeskind, and K. S. Wong, "Computational fluid dynamics modeling of symptomatic intracranial atherosclerosis may predict risk of stroke recurrence", PLoS ONE **9**, edited by J. Hendrikse, e97531 (2014).

[52] A. Chien, F. Vinuela, and G. Duckwiler, "Blood flow changes induced by flow diverting stent in a large wide-neck intracranial aneurysm", Journal of Vascular and Interventional Radiology **26**, 148 (2015).

[53] A. C. O. Tsang, S. S. M. Lai, W. C. Chung, A. Y. S. Tang, G. K. K. Leung, A. K. K. Poon, A. C. H. Yu, and K. W. Chow, "Blood flow in intracranial aneurysms treated with Pipeline embolization devices: computational simulation and verification with Doppler ultrasonography on phantom models.", Ultrasonography **34**, 98–108 (2015).

[54] S. Moore, T. David, J. G. Chase, J. Arnold, and J. Fink, "3D models of blood flow in the cerebral vasculature", Journal of Biomechanics **39**, 1454–1463 (2006).

[55] N. Xiao, J. D. Humphrey, and C. A. Figueroa, "Multi-scale computational model of three-dimensional hemodynamics within a deformable full-body arterial network", Journal of Computational Physics **244**, 22–40 (2013).

[56] L. O. Müller and E. F. Toro, "A global multiscale mathematical model for the human circulation with emphasis on the venous system", International Journal for Numerical Methods in Biomedical Engineering **30**, 681–725 (2014).

[57] H. M. Hasan, A. Coccarelli, and P. Nithiarasu, "Novel semi-implicit, locally conservative Galerkin (SILCG) methods: Application to blood flow in a systemic circulation", Computer Methods in Applied Mechanics and Engineering **332**, 217–233 (2018).

[58] I. P. Herman, *Physics of the human body*, 2nd ed., Biological and medical physics, biomedical engineering (Springer, Berlin; New York, 2016), p. 953.

[59] F. A. C. Azevedo, L. R. B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel, "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain", Journal of Comparative Neurology **513**, 532–541 (2009).

[60] F. A. Henn and A. Hamberger, "Glial cell function: uptake of transmitter substances", Proceedings of the National Academy of Sciences of the United States of America **68**, 2686–2690 (1971).

[61] F. A. Henn, H. Haljamäe, and A. Hamberger, "Glial cell function: Active control of extracellular K+ concentration", Brain Research **43**, 437–443 (1972).

[62] F. Cassot, F. Lauwers, S. Lorthois, P. Puwanarajah, and H. Duvernoy, "Scaling laws for branching vessels of human cerebral cortex", Microcirculation **16**, 331–344 (2009).

[63] E. Papp-Szabó, P. D. Josephy, and B. L. Coomber, "Microenvironmental influences on mutagenesis in mammary epithelial cells", International Journal of Cancer **116**, 679–685 (2005).

[64] P. Rubin and L. Fajardo, "Biophysiopathology of the microvasculature and microcirculation", in *Alert - adverse late effects of cancer treatment*, edited by P. Rubin, L. S. Constine, and L. B. Marks (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014) Chap. 3, pp. 27–39.

[65] L. Poston, *An introduction to vascular biology: from basic science to clinical practice*, edited by B. J. Hunt, L. Poston, M. Schachter, and A. W. Hallidat, 2nd ed. (Cambridge University Press, Cambridge, UK, 2002).

[66] M. Cavaglia, S. M. Dombrowski, J. Drazba, A. Vasanji, P. M. Bokesch, and D. Janigro, "Regional variation in brain capillary density and vascular response to ischemia", Brain Research **910**, 81–93 (2001).

[67] B. Li, H. Wang, G. Qiu, X. Su, and Z. Wu, "Synergistic effects of vascular endothelial growth factor on bone morphogenetic proteins induced bone formation in vivo: influencing factors and future research directions", BioMed Research International **2016**, 11 (2016).

[68] F. N. E. Gavins and K. Y. Stokes, eds., *Vascular responses to pathogens*, 1st ed. (Academic Press, Cambridge, MA, 2015), p. 252.

[69] M. A. Matthay, G. A. Zimmerman, C. Esmon, J. Bhattacharya, B. Coller, C. M. Doerschuk, J. Floros, M. A. Gimbrone, E. Hoffman, R. D. Hubmayr, M. Leppert, S. Matalon, R. Munford, P. Parsons, A. S. Slutsky, K. J. Tracey, P. Ward, D. B. Gail, and A. L. Harabin, "Future research directions in acute lung injury: Summary of a National Heart, Lung, and Blood Institute Working Group", American Journal of Respiratory and Critical Care Medicine **167**, 1027–1035 (2003).

[70] S. Hauser, F. Jung, and J. Pietzsch, "Human endothelial cell models in biomaterial research", Trends in Biotechnology **35**, 265–277 (2017).

[71] M. G. Clemente, C. Mandato, M. Poeta, and P. Vajro, "Pediatric non-alcoholic fatty liver disease: Recent solutions, unresolved issues, and future research directions", World Journal of Gastroenterology **22**, 8078–8093 (2016).

[72] M. Franzblau, C. Gonzales-Portillo, G. S. Gonzales-Portillo, T. Diamandis, M. C. Borlongan, N. Tajiri, and C. V. Borlongan, "Vascular damage: A persisting pathology common to Alzheimer's disease and traumatic brain injury", Medical Hypotheses **81**, 842–845 (2013).

[73] C. Werner and K. Engelhard, "Pathophysiology of traumatic brain injury", British Journal of Anaesthesia **99**, 4–9 (2007).

[74] W. R. Brown, R. M. Blair, D. M. Moody, C. R. Thore, S. Ahmed, M. E. Robbins, and K. T. Wheeler, "Capillary loss precedes the cognitive impairment induced by fractionated whole-brain irradiation: A potential rat model of vascular dementia", Journal of the Neurological Sciences **257**, 67–71 (2007).

[75] H. M. Duvernoy, S. Delon, and J. L. Vannson, "Cortical blood vessels of the human brain", Brain Research Bulletin **7**, 519–579 (1981).

[76] H. Gray and W. H. Lewis, *Anatomy of the human body*, 20th ed. (Lea & Febiger, Philadelphia, 1918), p. 1396.

[77] D. Mayerich, J. Kwon, C. Sung, L. Abbott, J. Keyser, and Y. Choe, "Fast macro-scale transmission imaging of microvascular networks using KESM", Biomedical Optics Express **2**, 2888 (2011).

[78] F. Cassot, F. Lauwers, C. Fouard, S. Prohaska, and V. Lauwers-Cances, "A novel three-dimensional computer-assisted method for a quantitative study of microvascular networks of the human cerebral cortex", Microcirculation **13**, 1–18 (2006).

[79] I. G. Gould and A. A. Linninger, "Hematocrit distribution and tissue oxygenation in large microcirculatory networks", Microcirculation **22**, 1–18 (2015).

[80] J. Jurcovicova, "Glucose transport in brain - effect of inflammation", Endocrine regulations **48**, 35–48 (2014).

[81] K. Masamoto and K. Tanishita, "Oxygen transport in brain tissue", Journal of Biomechanical Engineering **131**, 74002–74006 (2009).

[82] N. J. Abbott, L. Rönnbäck, and E. Hansson, "Astrocyte–endothelial interactions at the blood–brain barrier", Nature Reviews Neuroscience **7**, 41–53 (2006).

[83] C. Davis, J. Fischer, K. Ley, and I. J. Sarembock, "The role of inflammation in vascular injury and repair", Journal of thrombosis and haemostasis **1**, 1699–1709 (2003).

[84] G. K. Hansson, J. Holm, S. Holm, Z. Fotev, H. J. Hedrich, and J. Fingerle, "T lymphocytes inhibit the vascular response to injury", Proceedings of the National Academy of Sciences **88**, 10530–10534 (1991).

[85] J. D. Catravas, A. D. Callow, and U. S. Ryan, *Vascular endothelium: Responses to injury*, edited by J. D. Catravas, A. D. Callow, and U. S. Ryan, 1st ed., NATO ASI series (Springer, New York, 1996).

[86] L. C. P. Azevedo, M. D. A. Pedro, L. C. Souza, H. P. De Souza, M. Janiszewski, P. L. Da Luz, and F. R. M. Laurindo, "Oxidative stress as a signaling mechanism of the vascular response to injury: The redox hypothesis of restenosis", Cardiovascular Research **47**, 436–445 (2000).

[87] A. C. Newby, "An overview of the vascular response to injury: A tribute to the late Russell Ross", Toxicology Letters **112-113**, 519–529 (2000).

[88] M. Scianna, C. G. Bell, and L. Preziosi, "A review of mathematical models for the formation of vascular networks", Journal of Theoretical Biology **333**, 174–209 (2013).

[89] M. R. Owen, T. Alarcón, P. K. Maini, and H. M. Byrne, "Angiogenesis and vascular remodelling in normal and cancerous tissues", Journal of Mathematical Biology **58**, 689–721 (2009).

[90] A. A. Qutub, G. Liu, P. Vempati, and A. S. Popel, "Integration of angiogenesis modules at multiple scales: from molecular to tissue", in Pacific symposium on biocomputing, Vol. 327 (2009), pp. 316–27.

[91] S. M. Peirce and T. C. Skalak, "Microvascular remodeling: a complex continuum spanning angiogenesis to arteriogenesis", Microcirculation **10**, 99–111 (2003).

[92] R. Karch, F. Neumann, B. K. Podesser, M. Neumann, P. Szawlowski, and W. Schreiner, "Fractal properties of perfusion heterogeneity in optimized arterial trees: a model study", The Journal of General Physiology **122**, 307–321 (2003).

[93] J. A. Adam, "Blood vessel branching: beyond the standard calculus problem", Mathematics Magazine **84**, 196–207 (2011).

[94] L. Dagum and R. Menon, "OpenMP: An industry-standard API for shared-memory programming", Computing in Science & Engineering **5**, 46–55 (1998).

[95] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A high-performance, portable implementation of the MPI message passing interface standard", Parallel Computing **22**, 789–828 (1996).

[96] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, "An overview of the HDF5 technology suite and its applications", in Proceedings of the edbt/icdt 2011 workshop on array databases (ACM, 2011), pp. 36–47.

[97] M. Howison, Q. Koziol, D. Knaak, J. Mainzer, and J. Shalf, "Tuning HDF5 for Lustre file systems", in Iasds '10 proceedings of the workshop on interfaces and abstractions for scientific data storage (2010).

[98] G. F. Pfister, "An introduction to the infiniband architecture", High Performance Mass Storage and Parallel I/O **42**, 617–632 (2001).

[99] P. Schwan, "Lustre: Building a file system for 1000-node clusters", in Proceedings of the 2003 linux symposium, Vol. 2003 (2003), pp. 380–386.

[100] N. Nethercote and J. Seward, "Valgrind", Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation - PLDI '07 **42**, 89–100 (2007).

[101] P. Deutsch, *GZIP file format specification version 4.3*, United States, 1996.

[102] D. Mayerich, J. Kwon, and Y. Choe, "Constructing high resolution microvascular models", in Proceedings of the 3rd workshop on microscopic image analysis with applications in biology (2008).

[103] W. Schreiner, R. Karch, F. Neumann, and M. Neumann, "Constrained constructive optimization of arterial tree models", in *Scaling in biology*, edited by J. Brown and G. West, 1st ed. (Oxford University Press, Oxford, UK, 2000), pp. 145–165.

[104] P. J. Blanco, S. M. Watanabe, M. A. R. Passos, P. A. Lemos, and R. A. Feijóo, "An anatomically detailed arterial network model for one-dimensional computational hemodynamics", IEEE Transactions on Biomedical Engineering **62**, 736–753 (2014).

[105] M. Kociński, A. Materka, A. Deistung, J. Reichenbach, and A. Lundervold, "Towards multi-scale personalized modeling of brain vasculature based on magnetic resonance image processing", in International conference on systems, signals, and image processing (2017), pp. 1–5.

[106] H. K. Hahn, M. Georg, and H.-O. Peitgen, "Fractal Aspects of Three-Dimensional Vascular Constructive Optimization", in *Fractals in biology and medicine*, edited by G. A. Losa, D. Merlini, T. F. Nonnenmacher, and E. R. Weibel (Birkhäuser-Verlag, Basel, 2005), pp. 55–66.

[107] A. Avolio, "Multi-branched model of the human arterial system", Medical and Biological Engineering and Computing **18**, 709–718 (1980).

[108] S. Strandgaard and O. B. Paulson, "Cerebral autoregulation", Stroke **15**, 413–416 (1984).

[109] D. Attwell, A. M. Buchan, S. Charpak, M. Lauritzen, B. A. MacVicar, and E. A. Newman, "Glial and neuronal control of brain blood flow", Nature **468**, 232–243 (2010).

[110] D. H. Ingvar and G. Franzén, "Abnormalities of cerebral blood flow distribution in patients with chronic schizophrenia", Acta Psychiatrica Scandinavica **50**, 425–462 (1974).

[111] F. J. Haddy and J. B. Scott, "Metabolically linked vasoactive chemicals in local regulation of blood flow.", Physiological reviews **48**, 688–707 (1968).

[112] A. Hauge, G. Nicolaysen, and M. Thoresen, "Acute effects of acetazolamide on cerebral blood flow in man", Acta Physiol Scand **117**, 233–239 (1983).

[113] D. W. Marion, J. Darby, and H. Yonas, "Acute regional cerebral blood flow changes caused by severe head injuries", J Neurosurg **74**, 407–414 (1991).

[114] K. Fassbender, B. Hodapp, S. Rossol, T. Bertsch, J. Schmeck, S. Schütt, M. Fritzinger, P. Horn, P. Vajkoczy, S. Kreisel, J. Brunner, P. Schmiedek, and M. Hennerici, "Inflammatory cytokines in subarachnoid haemorrhage: Association with abnormal blood

flow velocities in basal cerebral arteries", Journal of Neurology Neurosurgery and Psychiatry **70**, 534–537 (2001).

[115] S. L. Murphy, J. Xu, K. D. Kochanek, S. C. Curtin, and E. Arias, *Deaths: Final data for 2015*, tech. rep. 6 (National Center for Health Statistics, 2017), p. 75.

[116] P. A. Heidenreich, J. G. Trogdon, O. A. Khavjou, J. Butler, K. Dracup, M. D. Ezekowitz, E. A. Finkelstein, Y. Hong, S. C. Johnston, A. Khera, D. M. Lloyd-Jones, S. A. Nelson, G. Nichol, D. Orenstein, P. W. Wilson, and Y. J. Woo, "Forecasting the future of cardiovascular disease in the United States: A policy statement from the American Heart Association", Circulation **123**, 933–944 (2011).

[117] J. Volker, *Finite element methods for incompressible flow problems*, Vol. 51, Springer Series in Computational Mathematics (Springer International Publishing, Cham, Switzerland, 2016), p. 812.

[118] M. Conti, M. Marconi, G. Campanile, A. Reali, D. Adami, R. Berchiolli, and F. Auricchio, "Patient-specific finite element analysis of popliteal stenting", Meccanica **52**, 633–644 (2017).

[119] Q. Pan, R. Wang, B. Reglin, G. Cai, J. Yan, A. R. Pries, and G. Ning, "A one-dimensional mathematical model for studying the pulsatile flow in microvascular networks not exhibiting vascular tone", Journal of Biomechanical Engineering **136**, 011009 (2013).

[120] L. Formaggia, F. Nobile, A. Quarteroni, and A. Veneziani, "Multiscale modelling of the circulatory system: a preliminary analysis", Computing and Visualization in Science **2**, 75–83 (1999).

[121] M. S. Olufsen, C. S. Peskin, W. Y. Kim, E. M. Pedersen, A. Nadim, and J. Larsen, "Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions", Annals of Biomedical Engineering **28**, 1281–1299 (2000).

[122] S. J. Sherwin, L. Formaggia, J. Peiró, and V. Franke, "Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system", International Journal for Numerical Methods in Fluids **43**, 673–700 (2003).

[123] S. P. Sutera and R. Skalak, "The History of Poiseuille's Law", Annual Review of Fluid Mechanics **25**, 1–20 (1993).

[124] A. R. Pries and T. W. Secomb, "Blood flow in microvascular networks", Circulation Research **67**, 826–824 (1990).

[125] N. M. Roth and M. F. Kiani, "A "geographic information systems" based technique for the study of microvascular networks", Annals of Biomedical Engineering **27**, 42–47 (1999).

[126] E. R. Hyde, A. N. Cookson, J. Lee, C. Michler, A. Goyal, T. Sochi, R. Chabiniok, M. Sinclair, D. A. Nordsletten, J. Spaan, J. P. H. M. Van Den Wijngaard, M. Siebes, and N. P. Smith, "Multi-scale parameterisation of a myocardial perfusion model using whole-organ arterial networks", Annals of Biomedical Engineering **42**, 797–811 (2014).

[127] A. R. Pries, T. W. Secomb, and P. Gaehtgens, "Biophysical aspects of blood flow in the microvasculature", Cardiovascular Research **32**, 654–667 (1996).

[128] G. L. Atkins, *Multicompartment models for biological systems*, 2nd (Methuen, New York, 1974), p. 153.

[129] D. McGowan, R. Macpherson, S. Hackett, D. Liu, F. Gleeson, W. McKenna, G. Higgins, and J. Fenwick, "18F-fluoromisonidazole uptake in advanced stage non-small cell lung cancer: A voxel-by-voxel PET kinetics study: A", Medical Physics **44**, 4665–4676 (2017).

[130] M. H. Cherkaoui-Rbati, S. W. Paine, P. Littlewood, and C. Rauch, "A quantitative systems pharmacology approach, incorporating a novel liver model, for predicting pharmacokinetic drug-drug interactions", PLoS ONE **12**, edited by J.-M. Yang, e0183794 (2017).

[131] H. G. Holzhütter, D. Drasdo, T. Preusser, J. Lippert, and A. M. Henney, *The virtual liver: A multidisciplinary, multilevel challenge for systems biology*, Vol. 4, 3 (Wiley-Blackwell, 2012), pp. 221–235.

[132] W. Donahue and W. D. Newhauser, "A simple computational model of the vasculature of the whole human brain", (In Prep.).

[133] S. S. Shibeshi and W. E. Collins, "The rheology of blood flow in a branced arterial system", Applied Rheology **15**, 398–405 (2005).

[134] H. A. van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge Monographs on Applied and Computational Mathematics (Cambridge University Press, Cambridge, UK; New York, USA, 2003), p. 221.

[135] R. D. Falgout and U. M. Yang, "hypre: A Library of High Performance Preconditioners", in Computational science - iccs 2002, edited by P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra (2002), pp. 632–641.

[136] G. Karypis, "Parallel multilevel k-way partitioning scheme for irregular graphs", SIAM Review **41**, 278–300 (1999).

[137] B. S. Aribisala, Z. Morris, E. Eadie, A. Thomas, A. Gow, M. C. Valdés Hernández, N. A. Royle, M. E. Bastin, J. Starr, I. J. Deary, and J. M. Wardlaw, "Blood Pressure, Internal Carotid Artery Flow Parameters, and Age-Related White Matter Hyperintensities", Hypertension **63**, 1011–1018 (2014).

[138] K. P. Ivanov, M. K. Kalinina, and Y. I. Levkovich, "Blood flow velocity in capillaries of brain and muscles and its physiological significance", Microvascular Research **22**, 143–155 (1981).

[139] M. Folk, A. Cheng, and K. Yates, "HDF5: A file format and I/O library for high performance computing applications", Proceedings of Supercomputing **99**, 5–33 (1999).

[140] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs", SIAM Journal on Scientific Computing **20**, 359–392 (1998).

[141] A. H. Baker, E. R. Jessup, and T. Manteuffel, "A technique for accelerating the convergence of Restarted GMRES", SIAM Journal on Matrix Analysis and Applications **26**, 962–984 (2005).

[142] V. E. Henson and U. M. Yang, "BoomerAMG: A parallel algebraic multigrid solver and preconditioner", in Applied numerical mathematics, Vol. 41, 1 (2002), pp. 155–177.

[143] K. Stüben, "A review of algebraic multigrid", Journal of Computational and Applied Mathematics **128**, 281–309 (2001).

[144] J. L. Gustafson, "Reevaluating Amdahl's law", Communications of the ACM **31**, 532–533 (1988).

[145] G. H. Glover, "Overview of functional magnetic resonance imaging", Neurosurgery clinics of North America **22**, 133–vii (2011).

[146] D. A. Boas, S. R. Jones, A. Devor, T. J. Huppert, and A. M. Dale, "A vascular anatomical network model of the spatio-temporal response to brain activation", NeuroImage **40**, 1116–1129 (2008).

[147] J. Reichold, M. Stampanoni, A. Lena Keller, A. Buck, P. Jenny, and B. Weber, "Vascular graph model to simulate the cerebral blood flow in realistic vascular networks", Journal of Cerebral Blood Flow and Metabolism **29**, 1429–1443 (2009).

[148] Y.-Y. I. Shih, E. R. Muir, G. Li, B. H. De La Garza, and T. Q. Duong, "High-resolution 3D MR microangiography of the rat ocular circulation", Radiology **264**, 234–241 (2012).

[149] E. A. Carter, R. G. Tompkins, M. L. Yarmush, W. A. Walker, and J. F. Burke, "Redistribution of blood flow after thermal injury and hemorrhagic shock", J Appl Physiol **65**, 1782–1788 (1988).

[150] D. J. Barker and M. A. Hanson, "Altered regional blood flow in the fetus: The origins of cardiovascular disease?", Acta Paediatrica, International Journal of Paediatrics **93**, 1559–1560 (2004).

[151] R. W. Gill, "Measurement of blood flow by ultrasound: Accuracy and sources of error", Ultrasound in Med. & Biol **11**, 625–641 (1985).

[152] E. G. Hoeffner, I. Case, R. Jain, S. K. Gujar, G. V. Shah, J. P. Deveikis, R. C. Carlos, B. G. Thompson, M. R. Harrigan, and S. K. Mukherji, "Cerebral perfusion CT: technique and clinical applications.", Radiology **231**, 632–644 (2004).

[153] M. M. Islam, T. Tsujikawa, T. Mori, Y. Kiyono, and H. Okazawa, "Estimation of arterial input by a noninvasive image derived method in brain $H_2{}^{15}O$ PET study: confirmation of arterial location using MR angiography", Physics in Medicine and Biology **62**, 4514–4524 (2017).

[154] B. R. Munson, T. H. Okiishi, W. W. Huebsch, and A. P. Rothmayer, *Fundementals of Fluid Mechanics*, 7th (John Wiley & Sons, Hoboken, NJ, 2013), p. 795.

[155] H. T. Huynh, Z. J. Wang, and P. E. Vincent, "High-order methods for computational fluid dynamics : A brief review of compact differential formulation on unstructured grids", Computers and Fluids **98**, 209–220 (2014).

[156] F. Moukalled, L. Mangani, M. Darwish, *et al.*, *The finite volume method in computational fluid dynamics*, Vol. 113 (Springer, Switzerland, 2016), p. 500.

[157] J. Tu, G.-H. Yeoh, and C. Liu, *Computational fluid dynamics: a practical approach* (Butterwork-Heinemann, 2018), p. 498.

[158] K.-i. Tsubota, S. Wada, and T. Yamaguchi, "Particle method for computer simulation of red blood cell motion in blood flow", Computer Methods and Programs in Biomedicine **83**, 139–146 (2006).

[159] K. Ho-Le, "Finite element mesh generation methods: a review and classification", Computer-Aided Design **20**, 27–38 (1988).

[160] W. P. Donahue, H. Wong, J. Moreno, V. Wilson, J. Dey, and W. D. Newhauser, "Calculations of steady-state blood flow in entire body: A computational feasibility study", (In Prep.).

[161] V. Alexa, I. Kiss, and S. Raţiu, "Verofocation of Bernoulli Law using the Software Autodesk Simulation CFD", Analecta Technica Szegedinensia **8**, 120–127 (2014).

[162] Autodesk, *Verification*, 2015.

[163] Q. Sun, A. Groth, and T. Aach, "Comprehensive validation of computational fluid dynamics simulations of in-vivo blood flow in patient-specific cerebral aneurysms", Medical Physics **39**, 742–754 (2012).

[164] T. Passerini, A. Quaini, U. Villa, A. Veneziani, and S. Canic, "Validation of an open source framework for the simulation of blood flow in rigid and deformable vessels", International Journal for Numerical Methods in Biomedical Engineering **29**, 1192–1213 (2013).

[165] A. G. Radaelli, L. Augsburger, J. R. Cebral, M. Ohta, D. A. Rüfenacht, R. Balossino, G. Benndorf, D. R. Hose, A. Marzo, R. Metcalfe, P. Mortier, F. Mut, P. Reymond, L. Socci, B. Verhegghe, and A. F. Frangi, "Reproducibility of haemodynamical simulations in a subject-specific stented aneurysm model - A report on the Virtual Intracranial Stenting Challenge 2007", Journal of Biomechanics **41**, 2069–2081 (2008).

[166] C. C. Botar, á. Tóth, O. R. Klisurić, D. D. Nićiforović, V. A. Vučaj Ćirilović, and V. E. Till, "Dynamic simulation and Doppler Ultrasonography validation of blood flow behavior in Abdominal Aortic Aneurysm", Physica Medica **37**, 1–8 (2017).

[167] W. Donahue, W. D. Newhauser, and J. F. Ziegler, "Analytical model for ion stopping power and range in the therapeutic energy interval for beams of hydrogen and heavier ions", Physics in Medicine and Biology **61**, 6570–6584 (2016).

[168] ICRU, *Stopping Powers and Ranges for Protons and Alpha Particles* (ICRU, Betheda, MD, 1993).

[169] ICRU, "Stopping of ions heavier than helium", ICRU Report 73 **5**, edited by R. Bimbot, H. Geissel, H. Paul, A. Shinner, and P. Sigmund (2005).

[170] J. F. Ziegler, "The Stopping of Energetic Light Ions in Elemental Matter", Journal of Applied Physics **85**, 1249–1272 (1999).

[171] P. Sigmund, "Errata and Addenda for ICRU Report 73 , Stopping of Ions Heavier than Helium 1 Errors and misprints in ICRU Report 73 (2005) 2 Revised tables", Journal of the ICRU **5**, 1–10 (2009).

[172] J. F. Ziegler and J. P. Biersack, *The stopping and range of ions in matter* (SRIM Co., Chester, Md., 1985), pp. 93–129.

[173] W. H. Bragg and R. Kleeman, "On the α particles of radium, and their loss of range in passing through various atoms and molecules", Philosophical Magazine Series 6 **10**, 318–340 (1905).

[174] R. Zhang and W. D. Newhauser, "Calculation of water equivalent thickness of materials of arbitrary density, elemental composition and thickness in proton beam irradiation.", Physics in medicine and biology **54**, 1383–95 (2009).

[175] R. Zhang, P. J. Taddei, M. M. Fitzek, and W. D. Newhauser, "Water equivalent thickness values of materials used in beams of protons, helium, carbon and iron ions", NIH Public Access **55**, 2481–2493 (2010).

[176] W. Brandt and M. Kitagawa, "Effective stopping-power charges of swift ions in condensed matter", Physical Review B **25**, 5631–5637 (1982).

[177] J. P. Biersack, "New projected range algorithm as derived from transport equations", Zeitschrift fur Physik A Atoms and Nuclei **305**, 95–101 (1982).

[178] W. Ulmer and E. Matsinos, "Theoretical methods for the calculation of Bragg curves and 3D distributions of proton beams", European Physical Journal: Special Topics **190**, 1–81 (2010).

[179] T. Bortfeld, "An analytical approximation of the Bragg curve for therapeutic proton beams.", Medical Physics **24**, 2024 (1997).

[180] M. J. Berger, J. S. Coursey, M. A. Zucker, and J. Chang, *ESTAR, PSTAR, and ASTAR: Computer Programs for Calculating Stopping-Power and Range Tables for Electrons, Protons, and Helium Ions (version 1.2.3).* Gaithersburg, MD, 2005.

[181] D. Goldberge and Addison-Wesley, *Genetic Algorithms in Search, Optimization & Machine Learning* (Addison-Wesley Longman Publishing Co., Inc., 1989), p. 372.

[182] W. Chauvenet, *A manual of spherical and practical astronomy* (J.B. Lippincott Company., Philadelphia, 1863), p. 723.

[183] N. M. Halfina, "Applications of the chauvenet test to detection of overliers in observations connected into a homogeneous Markov Chain", Journal of Mathematical Sciences **118**, 5667–5672 (2003).

[184] G. Konac, C. Klatt, and S. Kalbitzer, "Universal fit formula for electronic stopping of all ions in carbon and silicon", Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms **146**, 106–113 (1998).

[185] A. K. M. M. Haque, A. Mohammadi, and H. Nikjoo, "Study of the stopping power and straggling for alpha particles and protons in organic solids, liquids and gases", Radiation protection dosimetry **13**, 71 (1985).

[186] H. Paul and A. Schinner, "Judging the reliability of stopping power tables and programs for heavy ions", Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms **209**, 252–258 (2003).

[187] S. Kalbitzer, H. Oetzmann, H. Grahmann, and A. Feuerstein, "A simple universal fit formula to experimental nuclear stopping power data", Zeitschrift f??r Physik A: Atoms and Nuclei **278**, 223–224 (1976).

[188] D. O. Gericke, "Stopping power for strong beam- plasma coupling", Laser and Particle Beams **20**, 471–474 (2002).

[189] M. Li, W. Guo, B. Verma, and H. Lee, "A neural networks-based fitting to high energy stopping power data for heavy ions in solid matter", Proceedings of the International Joint Conference on Neural Networks, 1 (2012).

[190] W. R. Leo, *Techniques for Nuclear and Particle Physics Experiments*, 2nd, Vol. 58 (U.S. Government Printing Office, 1990), p. 1216.

[191] F. a. Stewart, S. Hoving, and N. S. Russell, "Vascular damage as an underlying mechanism of cardiac and cerebral toxicity in irradiated cancer patients.", Radiation research **174**, 865–869 (2010).

[192] B. Bednarz and X. G. Xu, "Monte Carlo modeling of a 6 and 18 MV Varian Clinac medical accelerator for in-field and out-of-field dose calculations: Development and validation", Physics in Medicine and Biology **54**, N43–N57 (2009).

[193] M. Douglass, E. Bezak, and S. Penfold, "Development of a randomized 3D cell model for Monte Carlo microdosimetry simulations", Medical Physics **39**, 3509–3519 (2012).

[194] J. Brown, M. Bernal, S. Incerti, H. Tran, S. Okada, M. Davídková, E. Delage, Z. Francis, Q. Pham, S. Guatelli, V. Ivanchenko, Y. Perrot, S. Enger, V. Štěpán, A. Ristic-Fira, H. Payno, Z. El Bitar, M. Bordage, I. Kyriakou, C. Villagrasa, I. Petrovic, T. Sasaki, M. Karamitros, L. Maigne, S. Meylan, and K. Murakami, "Track structure modeling in liquid water: A review of the Geant4-DNA very low energy extension of the Geant4 Monte Carlo simulation toolkit", Physica Medica **31**, 861–874 (2015).

[195] C. Champion, M. A. Quinto, J. M. Monti, M. E. Galassi, P. F. Weck, O. A. Fojón, J. Hanssen, and R. D. Rivarola, "Water versus DNA: new insights into proton track-structure modelling in radiobiology and radiotherapy", Phys. Med. Biol **60**, 8–9 (2015).

[196] A. Bouchet, M. Potez, N. Coquery, C. Rome, B. Lemasson, E. Bräuer-Krisch, C. Rémy, J. Laissue, E. L. Barbier, V. Djonov, and R. Serduc, "Permeability of brain tumor vessels induced by uniform or spatially micro-fractionated synchrotron radiation therapies", International Journal of Radiation Oncology*Biology*Physics **98**, 1174–1182 (2017).

[197] M. S. Mendonca, H. Chin-Sinex, R. Dhaemers, L. E. Mead, M. C. Yoder, and D. A. Ingram, "Differential mechanisms of x-ray-induced cell death in human endothelial progenitor cells isolated from cord blood and adults.", Radiation research **176**, 208–16 (2011).

[198] M.-O. Lee, S.-H. Song, S. Jung, S. Hur, T. Asahara, H. Kim, S.-M. Kwon, and H.-J. Cha, "Effect of ionizing radiation induced damage of endothelial progenitor cells in vascular regeneration.", Arteriosclerosis, thrombosis, and vascular biology **32**, 343–52 (2012).

[199] B. P. Venkatesulu, L. S. Mahadevan, M. L. Aliru, X. Yang, M. H. Bodd, P. K. Singh, S. W. Yusuf, J.-I. Abe, and S. Krishnan, "Radiation-induced endothelial vascular injury: A review of possible mechanisms", JACC. Basic to translational science **3**, 563–572 (2018).

[200] J. L. Gevertz, "Computational modeling of tumor response to vascular-targeting therapies–part I: validation.", Computational and mathematical methods in medicine **2011**, 830515 (2011).

[201] A. Merrem, S. Bartzsch, J. Laissue, and U. Oelfke, "Computational modelling of the cerebral cortical microvasculature: effect of x-ray microbeams versus broad beam irradiation", Physics in Medicine and Biology **62**, 3902–3922 (2017).

[202] J. L. Bentley, "Multidimensional binary search trees used for associative searching", Communications of the ACM **18**, 509–517 (1975).

[203] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data", Pattern Analysis and Machine Intelligence, IEEE Transactions on **36** (2014).

[204] O. Tange, "GNU Parallel: the command-line power tool", ;login: The USENIX Magazine **36**, 42–47 (2011).

[205] T. E. Oliphant, *A guide to NumPy*, Vol. 1 (Trelgol Publishing USA, 2006).

[206] A. C. McKee, T. D. Stein, C. J. Nowinski, R. A. Stern, D. H. Daneshvar, V. E. Alvarez, H. S. Lee, G. Hall, S. M. Wojtowicz, C. M. Baugh, D. O. Riley, C. A. Kubilus, K. A. Cormier, M. A. Jacobs, B. R. Martin, C. R. Abraham, T. Ikezu, R. R. Reichard, B. L. Wolozin, A. E. Budson, L. E. Goldstein, N. W. Kowall, and R. C. Cantu, "The spectrum of disease in chronic traumatic encephalopathy", Brain **136**, 43–64 (2013).

[207] X. Gu, D. Choi, C. Men, H. Pan, A. Majumdar, and S. B. Jiang, "GPU-based ultra-fast dose calculation using a finite pencil beam model", Physics in Medicine and Biology **54**, 6287–6297 (2009).

[208] K. Souris, J. A. Lee, and E. Sterpin, "Fast multipurpose Monte Carlo simulation for proton therapy using multi- and many-core CPU architectures", Medical Physics **43**, 1700–1712 (2016).

[209] W. P. Donahue, H. Wong, J. Dey, and W. D. Newhauser, "Accuracy of a steady-state model of blood flow though a vasculature of 126 vessels", (In Prep.).

[210] M. Mantyla, J. Kuikka, and A. Rekonen, "Regional blood flow in human tumours with special reference to the effect of radiotherapy", British Journal of Radiology **49**, 335–338 (1976).

[211] J. P. Pirhonen, S. A. Grenman, Å. B. Bredbacka, R. O. Bahado-Singh, and T. A. Salmi, "Effects of external radiotherapy on uterine blood flow in patients with advanced cervical carcinoma assessed by Color Doppler ultrasonography", Cancer **76**, 67–71 (1995).

[212] T. W. Secomb, R. Hsu, R. D. Braun, J. R. Ross, J. F. Gross, and M. W. Dewhirst, "Theoretical simulation of oxygen transport to tumors by three-dimensional networks of microvessels.", Advances in Experimental Medicine and Biology **454**, 629–34 (1998).

[213] C. Pozrikidis and D. A. Farrow, "A model of fluid flow in solid tumors", Annals of Biomedical Engineering **31**, 181–194 (2003).

[214] D. J. Rohrbach, E. C. Tracy, J. Walker, H. Baumann, and U. Sunar, "Blood flow dynamics during local photoreaction in a head and neck tumor model", Frontiers in Physics **3**, 13 (2015).

[215] A. L. Zwingenberger, R. E. Pollard, S. L. Taylor, R. X. Chen, J. Nunley, and M. S. Kent, "Perfusion and volume response of canine brain tumors to stereotactic radiosurgery and radiotherapy", Journal of Veterinary Internal Medicine **30**, 827–835 (2016).

Appendix A. Permissions to Reproduce Published Works

Permission Document for Chapter 5

Re: Request for permission to reuse article material in dissertation

Permissions <permissions@iop.org>
Thu 9/13/2018 5:40 AM
To: William P Donahue <wdonah2@lsu.edu>

Dear William Donahue,

Thank you for your email and for taking the time to seek this permission.

When you transferred the copyright in your article to Institute of Physics and Engineering in Medicine, you were granted back certain rights, including the right to include the Final Published Version of the article within any thesis or dissertation. Please note you may need to obtain separate permission for any third party content you included within your article.

Please include citation details, "© Institute of Physics and Engineering in Medicine. Reproduced with permission. All rights reserved" and for online use, a link to the Version of Record.

The only restriction is that if, at a later date, you wanted your thesis/dissertation to be published commercially, further permission would be required.

I wish you the best of luck with the completion of your thesis/dissertation.

Kind regards,
Christina

Copyright & Permissions Team
Gemma Alaway – Senior Rights & Permissions Adviser
Christina Colwell - Rights & Permissions Assistant

Contact Details
E-mail: permissions@iop.org
For further information about copyright and how to request permission:
https://publishingsupport.iopscience.iop.org/copyright-journals/

See also: https://publishingsupport.iopscience.iop.org/

Please see our Author Rights Policy https://publishingsupport.iopscience.iop.org/author-rights-policies/
Please note: We do not provide signed permission forms as a separate attachment. Please print this email and provide it to your publisher as proof of permission.

Please note: Any statements made by IOP Publishing to the effect that authors do not need to get permission to use any content where IOP Publishing is not the publisher is not intended to constitute any sort of legal advice. Authors must make their own decisions as to the suitability of the content they are using and whether they require permission for it to be published within their article.

From: William Donahue <wdonah2@lsu.edu>
Sent: Tuesday, August 28, 2018 21:19
To: Permissions
Subject: Request for permission to reuse article material in dissertation

To Whom It May Concern:
I am in the process of preparing my PhD dissertation, and would greatly appreciate permission to reuse the material from my first author publication in Physics in Medicine and Biology. The information regarding the article I am requesting permission to reuse is as follows:
Journal: Physics in Medicine and Biology
Article Title: Analytical model for ion stopping power and range in the therapeutic energy interval for beams of
hydrogen and heavier ions
Authors: William Donahue, Wayne D Newhauser and James F Ziegler
Volume: 17
Issue: 61
Page Numbers: 6570 - 6584
DOI: https://doi.org/10.1088/0031-9155/61/17/6570
The Information regarding the new document I am preparing is as follows:
Article/Chapter Title: Dissertation Chapter
Author: William Patrick Donahue
Publisher: Louisiana State University and Agricultural and Mechanical College

I will include a citation to the original work on the first page of the chapter. Would it be possible to get a letter or email stating that IOP grants permission for the use this material in my dissertation?
I did see the link to the CCC, however none of the options seemed to match my criteria. If I need to use that process, which options do I need to select?
Thank you for you time,
William Donahue
Louisiana State University - Ph.D. Candidate

Vita

William Donahue, born in Hartford Connecticut, became interested in the field of medical physics during high school. He pursued a Bachelor's degree in Physics from the University of Connecticut to begin his journey. During that time, he participated in internships and research related to the clinical aspects of medical physics. His undergraduate experiences motivated him to pursue a doctorate in medical physics, to improve the quality of care given to patients. William focused his research of the basic science aspects of medical physics and the interactions of radiation and normal tissues. Following completing his doctorate, William plans to enter a residency program with the goal of becoming a board-certified medical physicist.