

An Isogeometric Analysis Computational Platform for Material Transport Simulation in Complex Neurite Networks

Angran Li¹, Xiaoqi Chai², Ge Yang^{2,3} and Yongjie Jessica Zhang^{1,2,*}

¹Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States.

²Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States.

³Department of Computational Biology, Carnegie Mellon University, Pittsburgh, PA 15213, United States.

*Corresponding Author: Yongjie Jessica Zhang. Email: jessicaz@andrew.cmu.edu.

Abstract: Neurons exhibit remarkably complex geometry in their neurite networks. So far, how materials are transported in the complex geometry for survival and function of neurons remains an unanswered question. Answering this question is fundamental to understanding the physiology and disease of neurons. Here, we have developed an isogeometric analysis (IGA) based platform for material transport simulation in neurite networks. We modeled the transport process by reaction-diffusion-transport equations and represented geometry of the networks using truncated hierarchical tricubic B-splines (THB-spline3D). We solved the Navier-Stokes equations to obtain the velocity field of material transport in the networks. We then solved the transport equations using the streamline upwind/Petrov-Galerkin (SU/PG) method. Using our IGA solver, we simulated material transport in three basic models of the network geometry: a single neurite, a neurite bifurcation, and a neurite tree with three bifurcations. In addition, the robustness of our solver is illustrated by simulating material transport in three representative and complex neurite networks. From the simulation we discovered several spatial patterns of the transport process. Together, our simulation provides key insights into how material transport in neurite networks is mediated by their complex geometry.

Keywords: Material transport; neurite network; isogeometric analysis; streamline upwind/Petrov-Galerkin method; variational multiscale method

1 Introduction

Neurons exhibit striking complexity and diversity in their geometry, which is defined predominantly by the morphology of their highly branched networks of neurites, i.e., axons and dendrites (Fig. 1) [1-3]. The geometry is essential to function of individual neurons and formation of neural circuits [4,5] but poses a significant transportation challenge because material synthesis and degradation in neurons are carried out mainly in the cell body [6,7]. To survive and function, neurons must constantly transport a wide variety of essential materials throughout their complex neurite networks to meet their metabolic and functional needs [8,9]. Currently, we cannot answer the question of how materials are transported within the complex neurite networks so that the right material is delivered to the right destination without oversupply, e.g., traffic jams, or shortage within individual neurite branches. Answering this question is fundamental to understanding the physiology and pathophysiology of neurons because they depend critically on the material transport process [8,9]. Answering this question is also fundamental to controlling the transport process for applications such as targeted delivery [10].

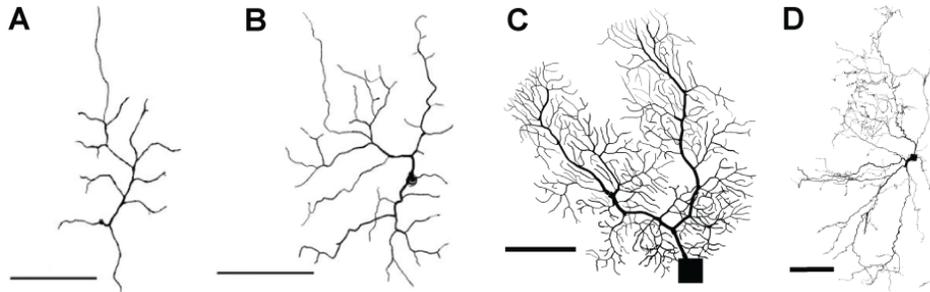


Figure 1: Examples of complex neuronal geometry. (A, B) Two zebrafish retina neurons. (C) A mouse cerebellum Purkinje neuron. (D) A rat olfactory short-axon neuron. Scale bars: $50 \mu\text{m}$. All data were downloaded from the NeuroMorpho database [1] and illustrated using custom software. NeuroMorpho.Org ID: (A) NMO_66731; (B) NMO_66748; (C) NMO_00865 and (D) NMO_06171

To answer the question of how materials are transported in neurite networks, computer simulation is essential because it is the foundation for mathematical modeling and analysis of the transport process. In this study, we modeled the transport process using a group of reaction-diffusion-transport equations [11-13]. However, the complex neurite network geometry posed significant challenges to solving these partial differential equations (PDEs). Conventional finite element based numerical techniques are known to have significant challenges in their numerical accuracy and efficiency when solving PDEs within the long, thin, and branched geometry of the neurite networks [14-16]. In particular, those techniques cannot handle large complex neurite networks (Fig. 1) because they cannot reconstruct the branched geometry with high accuracy without excessive computational costs. To overcome these challenges, we have developed a method to solve PDEs within the neurite network geometry using isogeometric analysis (IGA). IGA is an advanced finite element technique that differs from conventional finite element analysis (FEA) for its direct integration of geometrical modeling with numerical solution [17,18]. With the same smooth spline basis functions utilized as the basis for both geometrical modeling and numerical solution, IGA can accurately represent a wide range of complex geometry with high-order continuity [18,19] while offering superior performance over FEA in numerical accuracy and robustness [18,20,21]. Due to its many performance advantages, IGA has been adopted in a wide variety of research areas, such as biomedical applications [20,22,23], shell structures [24,25] and fluid-structure interaction [26-28].

In this study, we have developed an IGA-based computational platform for material transport simulation within the complex geometry of neurite networks. To derive physically realistic velocity field required for solving the generalized 3D transport model, we solved the Navier-Stokes equations using the variational multiscale residual based approach [29]. To overcome the oscillation of solution caused by the convection term in the transport equations, we used the streamline upwind/Petrov-Galerkin (SUPG) formulation [30] to avoid numerical instability and to improve numerical accuracy. To verify our solver, we first compared its output with the theoretical solution of the transport equations in a single infinitely long straight pipe. We found that the output matched the analytical solution with less than 5% error. To further verify the solver, we simulated material transport in three basic models of the complex network geometry: a single neurite, a neurite bifurcation, and a neurite tree with three bifurcations. We then applied it to transport simulation in three complex neurite networks and discovered several patterns of the transport process. In summary, our simulation revealed that the geometry of neurite network plays an important role in routing of material transport at junctions of neurite branches and in distributing the transported materials throughout the networks. It provides key insights into how material transport in neurite networks is mediated by their complex geometry. Our IGA solver is transformative and can be extended to solve other PDE models of cellular processes in the neurite network geometry.

The remainder of the paper is organized as follows. Section 2 overviews the computational platform and introduces the construction of neurite geometry. Section 3 presents 3D generalization of the motor-assisted material transport model and how to solve the equations. Several representative neurite geometries

are tested in Section 4 to demonstrate the applicability and robustness of our platform. Section 5 draws the conclusion and suggests future work.

2 Platform Overview and Neurite Geometry Construction

The overall workflow of our platform is summarized in Fig. 2. The platform consists of both modeling and analysis. In the modeling part, we adopt a skeleton-based sweeping method [20] to generate hexahedral control meshes for the neurite networks. Truncated hierarchical tricubic B-splines (THB-spline3D) [31,32] are then constructed on these control meshes to represent the geometry for simulation. In addition, we collect simulation parameters like the diffusion coefficient, transport velocity and reaction rate based on experiments and related literature. In the analysis part, we develop an IGA kernel solver by generalizing the one dimensional motor-assisted transport model [11] to 3D geometry. We first solve the Navier-Stokes equations to obtain the velocity field inside the constructed neurite geometry, and then take the velocity field as the input to analyze 3D motor-assisted transport. Both the velocity field and the dynamic material distribution results are extracted and visualized in ParaView [33].

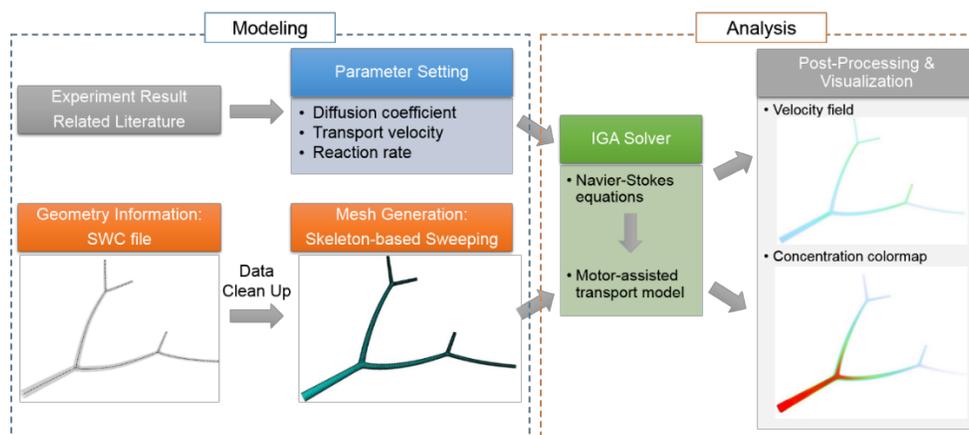


Figure 2: An overview of our IGA-based platform workflow

2.1 Control Mesh Generation

To represent the complex and branched geometry of neurite networks, we import geometry (vertices on the centerline with directions and associated diameters) of neurite networks recorded in the SWC file from the NeuroMorpho database [1] and visualize the geometry using the TREES toolbox [34]. Before we import the SWC file for mesh generation, we first clean up the data to make sure there is no duplicated vertices or overlapping centerlines inside the geometry. We then decompose the geometry of the neurite network structure into many segments. We refer to the connected centerlines of the segments as a *skeleton* (Fig. 3(A)) and the splitting of a neurite into two branches as a *bifurcation*. We assume each segment as a pipe so that we can describe its location and shape with two end points and their associated diameters. We generate the hexahedral control mesh for each segment by sweeping the quadrilateral mesh of the cross section along the skeleton. The control mesh is required to guide the generation of splines for more accurate and smooth representation of the neurite network geometry. One-to-one sweeping requires that the source and target surfaces have the same topology. However, bifurcations do not satisfy this requirement at the branching points. To resolve this issue, we construct a template for each bifurcation by inserting a half plane [20] (Figs. 3(B) & 3(C)). The control mesh is generated to satisfy two requirements to achieve a G^1 -continuous surface around each branching point: (a) the boundary vertex shared by two branches in the control mesh must be collinear with its two neighboring vertices along the axial direction; and (b) the boundary vertex shared by three branches must be coplanar with all of its neighboring boundary vertices. Using the skeleton-based sweeping method, we generate control meshes for three simple models of the neurite network geometry: a single straight neurite (Fig. 3(D)), a neurite bifurcation (Fig. 3(E)), and a

neurite tree with three bifurcations (Fig. 3(F)). The hexahedral control meshes are then used to construct THB-spline3D for final representation of these three models.

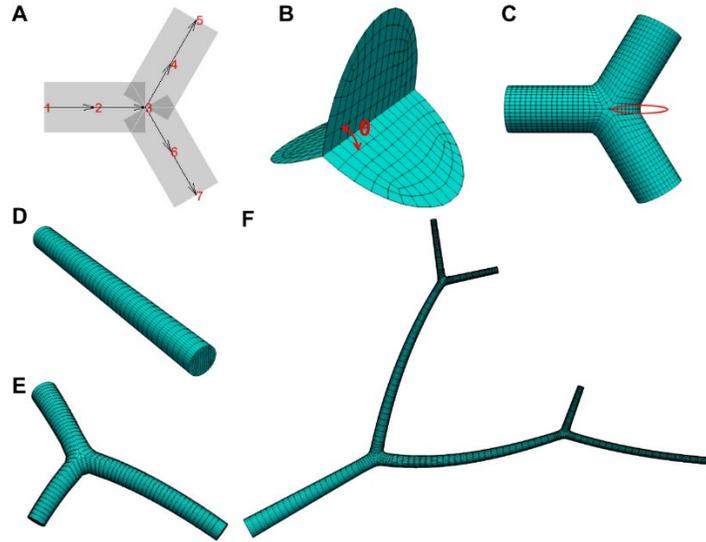


Figure 3: Mesh generation for neurite geometry. (A-C) Generation of hexahedral control mesh for a bifurcation structure by inserting a half plane at the bifurcation (red ellipsoid in C). (D-F) Hexahedral control meshes for a single straight neurite, a neurite bifurcation and a neurite tree with three bifurcations, respectively

2.2 A Review of THB-Spline3D

THB-spline3D [31,32] was developed by extending bicubic blending functions over quadrilateral meshes to tricubic blending functions over hexahedral meshes. The hierarchical structure and the truncation mechanism are further developed on tricubic blending functions for highly localized refinement. Let us briefly review the definition of tricubic blending functions using Bernstein polynomial. We first introduce some terminologies for explanation. In a hexahedral mesh, different types of element have different definition of blending functions. We sort all the elements into three groups: boundary elements, interior regular elements and interior irregular elements. An interior face is shared by two elements while a boundary face is owned by only one element. Edges and vertices of a boundary face are *boundary edges* and *boundary vertices*, respectively. An interior edge is regular if it is shared by four elements. Otherwise it is called an *extraordinary edge*. An interior element is regular if all its edges are regular; otherwise it is irregular. The blending functions on the irregular elements are defined with Bézier extraction matrix \mathbf{C} that can transform a local spline control mesh to a Bézier control mesh [35,36]. For a given hexahedral element Π_e in Fig. 4(A), the corresponding 64 Bézier points \mathbf{R}^e can be calculated using the vertices \mathbf{P}^e of its local control mesh as follows:

- Body points \mathbf{R}_{body}^e (red points in Fig. 4(B)): We compute the body points as the convex combination of 8 vertices \mathbf{P}_{body}^e :

$$\mathbf{R}_{body}^e = \mathbf{C}_{body}^e \mathbf{P}_{body}^e, \quad (1)$$

where \mathbf{C}_{body} converts 8 corner points of the hexahedral element to 8 Bézier body points \mathbf{R}_{body} and it is a

tensor product of 1D coefficients $\left\{ \frac{2}{3}, \frac{1}{3} \right\}$.

- Non-body points $\mathbf{R}_{non-body}^e$: Once we get the body Bézier points for all the elements, we evaluate the Euclidean distance from non-body points to the body points. For each non-body point, we can find its nearest body point in each element sharing this point and create a set $\Omega(e, i)$ in which the index pair (k, j) represents the j^{th} index in element Π_k . Then we can further calculate all the non-body points $\mathbf{R}_{non-body}^e$ by averaging the nearest body points:

$$R_{non-body}^{e,i} = \frac{1}{n_i} \sum_{(k,j) \in \Omega(e,i)} R_{body}^{k,j}, \quad (2)$$

where n_i is the number of elements sharing the point $R_{non-body}^{e,i}$.

After these two steps, we derive all the 64 Bézier points as a convex combination of vertices in the local spline control mesh, which is $\mathbf{R}^e = \mathbf{C}\mathbf{P}$. Then the blending functions on Π_e are defined using the transpose of \mathbf{C} ,

$$\mathbf{N}^e = \mathbf{C}^T \mathbf{b}, \quad (3)$$

and

$$\mathbf{b} = [g_0(\xi)g_0(\eta)g_0(\zeta), g_1(\xi)g_0(\eta)g_0(\zeta), \dots, g_3(\xi)g_0(\eta)g_0(\zeta), g_0(\xi)g_1(\eta)g_0(\zeta), g_1(\xi)g_1(\eta)g_0(\zeta), \dots, g_3(\xi)g_1(\eta)g_0(\zeta), g_0(\xi)g_0(\eta)g_1(\zeta), \dots, g_1(\xi)g_0(\eta)g_1(\zeta), \dots, g_3(\xi)g_0(\eta)g_1(\zeta), \dots, g_3(\xi)g_3(\eta)g_3(\zeta)]^T, \quad (4)$$

where $g_i(\omega) = \binom{3}{i} (1-\omega)^{3-i} \omega^i$ for $i = 0, 1, 2, 3$. The blending functions on a regular interior element can also be derived following the same procedure with fixed n_i for edge point ($n_i = 4$) and corner point ($n_i = 6$) in Eq. (2). Regarding to the boundary hexahedral element, we use the same idea to compute the interior Bézier points and Bézier points on the boundary are defined by bicubic blending functions on quadrilateral surface mesh [37].

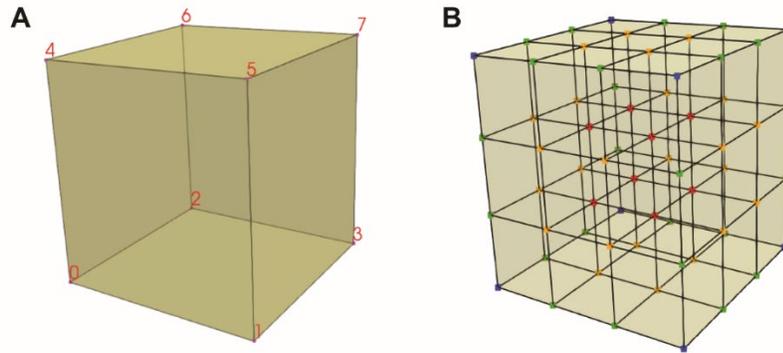


Figure 4: A hexahedral element with local indices of 8 vertices (A) and its corresponding 64 Bézier points (B). The red, yellow, green and blue dots represent body points, face points, edge, points and corner points, respectively

Based on the blending functions in the given hexahedral mesh, we can construct the THB-spline 3D volume by

$$V = \sum_{k=1}^{n_p-1} P_k N_k, \quad (5)$$

where n_p is the number of control points. P_k represents the k^{th} control points. N_k stands for the blending function defined on P_k . The blending functions are further used as basis functions in our simulation.

3 IGA Solver

In this section, we first explain 3D generalization of the motor-assisted transport model for intracellular particles used in our IGA solver. Then, we introduce the implementation of stabilization formulation to solve the equations.

3.1 Modeling Motor-Assisted Material Transport in 3D Geometry

A one-dimensional model [11] was developed to describe the macroscopic intracellular transport of vesicles and organelles, which we referred to henceforth simply as “particles”, by molecular motors on a network of aligned intracellular filaments, specifically microtubules or actin filaments. The model consists of a group of “reaction-diffusion-transport” equations that use simple kinetics to describe the interaction of particles with filaments and allow free diffusion of unattached particles and directed motion of attached particles (Fig. 5).

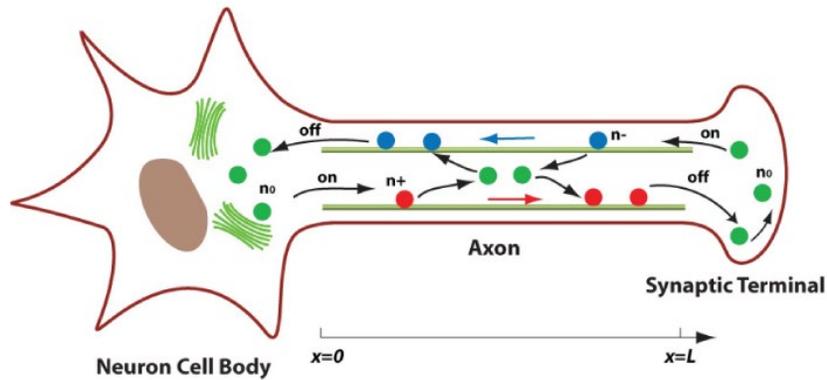


Figure 5: A cartoon of motor-assisted transport model in a single neurite (an axon in this case). Green: particles in free diffusion state with concentration n_0 ; Red: particles in anterograde transport state on microtubule with concentration n_+ ; and Blue: particles in retrograde transport state on microtubule with concentration n_-

To accurately account for the actual morphology of a single neurite, we generalize the transport model to 3D geometry and obtain

$$\begin{cases} \frac{\partial n_0}{\partial t} - D\nabla^2 n_0 = -(k_+ + k_-)n_0 + k'_+ n_+ + k'_- n_- & \text{in } \Omega, \\ \frac{\partial n_+}{\partial t} + v_+ \cdot \nabla n_+ = k_+ n_0 - k'_+ n_+ & \text{in } \Omega, \\ \frac{\partial n_-}{\partial t} + v_- \cdot \nabla n_- = k_- n_0 - k'_- n_- & \text{in } \Omega, \end{cases} \quad (6)$$

where the open set $\Omega \subset R^3$ represents the internal space of the single neurite; n_0 , n_+ and n_- are the spatial concentrations of free, incoming (relative to the cell body; retrograde), and outgoing (anterograde) material, respectively; D is the diffusion coefficient of free material; v_+ and v_- are velocities of incoming and outgoing material, respectively; and k_{\pm} and k'_{\pm} are rates of cytoskeletal filament attachment and detachment of incoming and outgoing materials, respectively. Note that n_0 , n_+ and n_- represents the concentration of the material in three different states, therefore the real concentration distribution of the material is evaluated by the summation of these three fields.

Appropriate boundary conditions are required to simulate material transport in a single neurite by solving Eq. (6). Here, we assume stable concentrations of free and incoming materials at both the incoming end and the outgoing end. We have

$$\begin{cases} n_0 = n, & n_+ = \lambda n \text{ at incoming end,} \\ n_0 = \tilde{n}, & n_- = \tilde{\lambda} \tilde{n} \text{ at outgoing end,} \end{cases} \quad (7)$$

where $\lambda, \tilde{\lambda}$ represent the degree of filament attachment at both ends and are also referred to as the “degree of loading” [11]. In the 1D transport model [11], the velocities of particle transport were assumed to be uniform. We could no longer make this assumption for physically realistic simulation of the transport process in the 3D geometry. Inspired by the kinetic theory of traffic flow [38,39], we apply a hydrodynamic description to describe the motion of traffic flow inside neurite. We assume that the flow of transport is incompressible while the diffusion coefficient is a constant. We solve the steady-state Navier-Stokes equations to derive a physically realistic velocity field inside the single neurite

$$\begin{cases} \nabla \cdot u = 0 & \text{in } \Omega, \\ \nabla \cdot (u \otimes u) + \nabla p = \nu \Delta u + f & \text{in } \Omega, \end{cases} \quad (8)$$

where the open set $\Omega \subset R^3$ represents the incompressible fluid domain, u is the flow velocity, p is the pressure, f is the given body force per unit volume, ν is the kinematic viscosity, and \otimes denotes the tensor product. Regarding boundary conditions, we impose non-slip condition at the neurite wall and apply a parabolic profile inlet velocity for each point on the circular cross section as,

$$v(r) = v_t \left(1 - (r/R)^2\right), \quad (9)$$

where v_t is the inlet transport velocity defined in our model, r is the distance from the center of the circular cross section to the point, and R is the radius of the circular cross section. The direction of the velocity is perpendicular to the cross section. In particular, the velocity result u is used in Eq. (6) with $v_+ = u$ and $v_- = -u$ defining a symmetric bidirectional transport. To overcome the oscillation of solution caused by the convection term in the transport equations, we use the streamline upwind/Petrov-Galerkin (SUPG) formulation [30] to avoid numerical instability and to improve numerical accuracy.

3.2 The Stabilized Formulation

In Section 3.1, we present a “reaction-diffusion-transport” model for material transport in neurite networks. Steady-state Navier Stokes equations are coupled with the model to derive the velocity field. We employ IGA in space discretization and a fully implicit method in time discretization. However, there is still a problem we need to consider before solving Eq. (6). The convection term in Eqs. (6) and (8) makes the stiffness matrix in the Galerkin method nonsymmetric, resulting in oscillations in solutions.

To resolve the instability issue in Eq. (8), we implement the variational multiscale (VMS) residual-based method [29]. Let V denote both trial solution and weighting function spaces. The VMS method decomposes V into ‘‘coarse-scale’’ and ‘‘fine-scale’’ subspaces, \bar{V} and V' , respectively. We have

$$V = \bar{V} + V'. \quad (10)$$

\bar{V} is assumed to be a finite-dimensional space defined by the basis functions we actually compute and V' is the infinite-dimensional space with unknown exact mathematic expression. Then the fine-scale V' can be represented as a functional F' of the coarse-scale \bar{V} and the residual $\mathbf{Res}(\bar{V})$, and we have

$$V' = F'(\bar{V}, \mathbf{Res}(\bar{V})). \quad (11)$$

In Eq. (8), we split the velocity and pressure into two components, namely, $u = \bar{u} + u'$ and $p = \bar{p} + p'$. Then we multiply Eq. (8) by the weighting functions $\bar{\omega}$ and $\bar{\psi}$, respectively, integrate over the domain, and perform some mathematical operations [29]. Following these steps, we obtain the weak formulation

$$B(\{\bar{\omega}, \bar{\psi}\}, \{\bar{u} + u', \bar{p} + p'\}) - L(\bar{\omega}) = 0, \quad (12)$$

with

$$\begin{aligned} B(\{\bar{\omega}, \bar{\psi}\}, \{\bar{u} + u', \bar{p} + p'\}) = & (\bar{\omega}, \bar{u} \cdot \nabla \bar{u})_{\Omega} - (\nabla \cdot \bar{\omega}, \bar{p})_{\Omega} + (\nabla^{sym} \bar{\omega}, 2\nu \nabla^{sym} \bar{u})_{\Omega} \\ & + (\bar{\psi}, \nabla \cdot \bar{u})_{\Omega} + (\bar{\omega}, u' \cdot \nabla \bar{u})_{\Omega} - (\nabla \bar{\omega}, (\bar{u} + u') \otimes u')_{\Omega} \\ & - (\nabla \cdot \bar{\omega}, p')_{\Omega} - (\nabla \bar{\psi}, \nabla \cdot \bar{u})_{\Omega} \end{aligned} \quad (13)$$

and

$$L(\bar{\omega}) = (\bar{\omega}, f)_{\Omega}, \quad (14)$$

where $\nabla^{sym}(\cdot)$ is the symmetric gradient operator given by $\nabla^{sym} \bar{u} = (\nabla \bar{u} + (\nabla \bar{u})^T)/2$.

The velocity and pressure solutions are derived by coupling the coarse-scale and approximated fine-scale together. Benchmark problems such as the laminar pipe flow were tested using our solver. Then we derive the steady-state velocity field for the given geometry and couple the velocity field solution with the transport equations (Eq. (6)) to obtain the concentration solution.

To handle the convection term in Eq. (6), we implement the stabilized formulations to prevent numerical oscillations and other instabilities problems. Several powerful stabilized methods have been proposed for flow simulation and modeling, such as the streamline upwind/Petrov-Galerkin (SUPG) method [30], the Galerkin/least-squares (GLS) method [40], and the pressure-stabilizing/Petrov-Galerkin (PSPG) method [41]. In these methods, the stabilization parameter τ was introduced and the selection of τ attracted great interest and study. In our solver, we implement the method introduced by Tezduyar et al. [42] to compute the stabilization parameter τ . First, let ω^h be the weighting function, n_0^h and n_{\pm}^h be the trial solutions for n_0 and n_{\pm} , respectively. Then, the stabilized formulation for Eq. (6) can be written as

$$\int_{\Omega} \omega^h \frac{\partial n_0^h}{\partial t} d\Omega + \int_{\Omega} \nabla \omega^h \cdot D \nabla n_0^h d\Omega + \int_{\Omega} \omega^h (k_+ + k_-) n_0^h d\Omega - \int_{\Omega} \omega^h (k'_+ n_+^h + k'_- n_-^h) d\Omega = 0, \quad (15)$$

$$\int_{\Omega} \omega^h \left(\frac{\partial n_+^h}{\partial t} + v_+^h \cdot \nabla n_+^h - k_+ n_0^h + k'_+ n_+^h \right) d\Omega + \sum_{i=1}^{n_{ele}} \int_{\Omega_i} \tau_{SUPG} v_+^h \cdot \nabla \omega^h \left(\frac{\partial n_+^h}{\partial t} + v_+ \cdot \nabla n_+^h - k_+ n_0^h + k'_+ n_+^h \right) d\Omega = 0 \quad (16)$$

and

$$\int_{\Omega} \omega^h \left(\frac{\partial n_-^h}{\partial t} + v_-^h \cdot \nabla n_-^h - k_- n_0^h + k'_- n_-^h \right) d\Omega + \sum_{i=1}^{n_{ele}} \int_{\Omega_i} \tau_{SUPG} v_-^h \cdot \nabla \omega^h \left(\frac{\partial n_-^h}{\partial t} + v_- \cdot \nabla n_-^h - k_- n_0^h + k'_- n_-^h \right) d\Omega = 0, \quad (17)$$

where n_{ele} is the number of elements, Ω_i is the domain for the i^{th} element, and τ_{SUPG} is the SUPG stabilization parameter. We define the length scale h as

$$h = 2 \|v_{\pm}^h\| \left(\sum_{k=1}^{n_{en}} |v_{\pm}^h \cdot \nabla N_k| \right)^{-1}, \quad (18)$$

where N_k is the basis function corresponding to node k and n_{en} is the number of nodes in this element. Then we defined the stabilization parameters as

$$\tau_1 = \frac{h}{2 \|v_{\pm}\|}, \quad (19)$$

$$\tau_2 = \frac{\Delta t}{2} \quad (20)$$

and

$$\tau_{SUPG} = \left(\frac{1}{\tau_1^2} + \frac{1}{\tau_2^2} \right)^{-1/2}, \quad (21)$$

where Δt stands for the time step size.

4 Numerical Examples

In this section, we start with a 1D benchmark problem, and then simulate the transport process in three simple nerite networks to study spatial patterns of the material. We extend our simulation to three neurite networks with complex geometry to further study the transport patterns and test the robustness of our solver.

4.1 1D Benchmark Problem

In order to verify the generalized 3D model, a straight pipe geometry (Fig. 3(D)) was first designed and used to compare with the analytical solution of the 1D transport equations in a single infinitely long straight pipe [11]. Specifically, to match with the settings of the analytical solutions, we assumed the same initial point distribution of the material as in Smith et al. [11] on the centerline of the pipe, with a length of $12 \mu m$ and a uniform diameter of $1 \mu m$ (the ratio of length over diameter is set to be 12 in order to approximate an infinitely long pipe). The transport process was assumed to be unidirectional, with the diffusion coefficient $D = 0.1 \mu m^2/s$, the filament attachment rate $k = 1.0 s^{-1}$ and the detachment rate $k' = 0.5 s^{-1}$. Four different material transport velocities, $v = 0.1, 0.2, 0.5$ and $1.0 \mu m/s$, were selected to verify that the SUPG stabilization method works for any velocity within a reasonable range. To compare our 3D simulation result with the 1D analytical solution [11], we conducted the simulation by solving Eq. (8) then Eq. (6) using IGA and extracted the average concentration distribution along the pipe (Fig. 6). We found that the distribution profile calculated from the simulation matched well with the analytical solution. Specifically, we compared the location of the maximum concentration for each velocity setting and found that the relative error was below 5% in each case (Tab. 1). Taken together, these results verified our 3D transport model and IGA solver under the single straight pipe geometry.

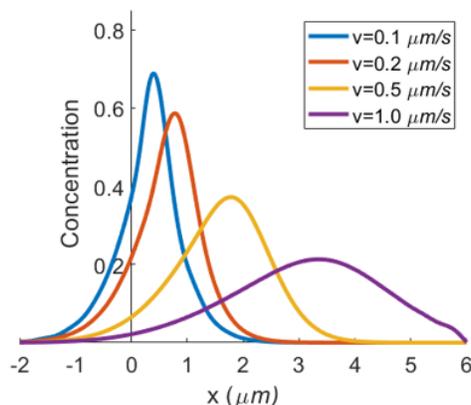


Figure 6: Verification of the generalized 3D model. Spatial distribution of unidirectionally transported particles was obtained in a single pipe at 5 s under different motor velocities at $v = 0.1, 0.2, 0.5$ and $1.0 \mu\text{m/s}$, respectively

Table 1: Comparison of locations of the maximum concentration under different velocities

Velocity ($\mu\text{m/s}$)	0.1	0.2	0.5	1.0
1D result [11]	0.390	0.854	1.854	3.537
3D result (Fig. 5(B))	0.372	0.852	1.836	3.552
Relative error	4.62%	0.23%	0.97%	0.42%

4.2 Simulation of Material Transport in Three Simple Neurite Networks

After generating geometric representations of three simple models, we simulated material transport within them using our IGA solver. According to [11], we assumed a diffusion coefficient $D = 0.4 \mu\text{m}^2/\text{s}$ derived from the Einstein-Stokes relation for a $1 \mu\text{m}$ diameter sphere in water; a typical molecular motor velocity $v_t = 1.0 \mu\text{m/s}$ as the inlet transport velocity; and an attachment rate $k = 1.0 \text{ s}^{-1}$. Boundary conditions in Eq. (7) were set using $n = 2.0 \text{ mol}/\mu\text{m}^3$, $\tilde{n} = 0 \text{ mol}/\mu\text{m}^3$, $\lambda = 0.5$ and $\tilde{\lambda} = 0$ to define a unidirectional material transport. We first calculated the steady-state velocity field by solving the Navier-Stokes equations; see the results in Figs. 7(A)-7(C). For each model, the inlet velocity was assumed to have a parabolic profile with a fixed maximum velocity magnitude. For the straight neurite geometry, the uniform velocity along the cylinder was consistent with the analytical solution of laminar flow in a straight pipe (Fig. 7(A)). Sudden increase in the velocity magnitude was observed near the branching point in both bifurcation models (Figs. 7(B) & 7(C)) and we found that the velocity magnitude was higher in shorter branch. The velocity difference is caused by the “do-nothing” boundary condition applied to all outlets which prescribes the same mean pressure at each outlet. Therefore, in short length pipes the pressure gradient is higher and leads to larger velocity magnitude. This indicated that geometry of neurite networks could substantially influence the interior velocity field.

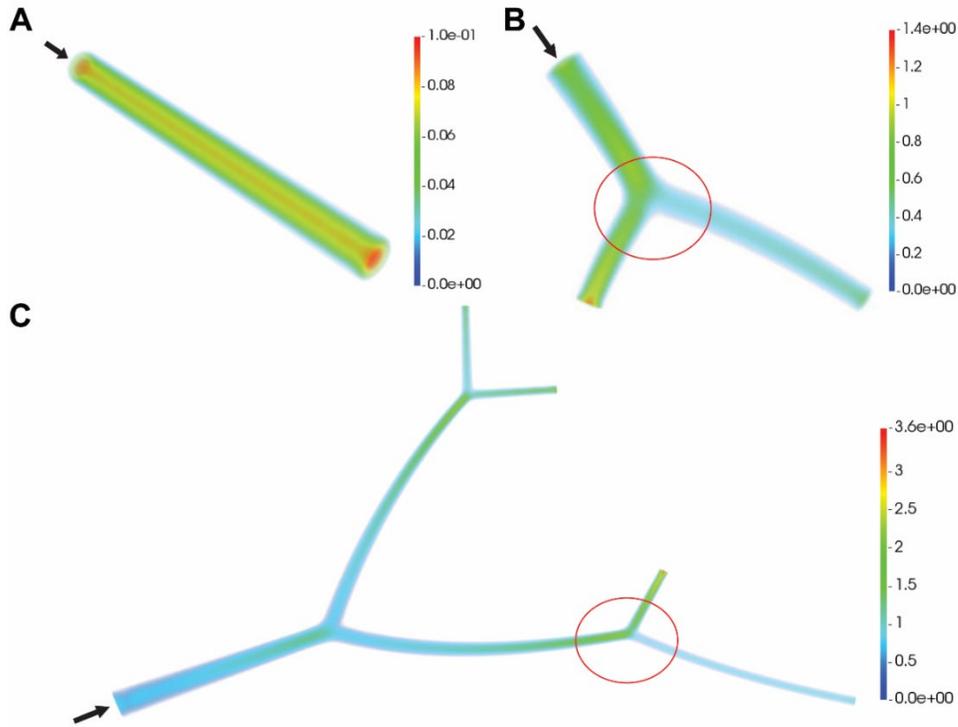


Figure 7: Steady-state velocity field of material transport in (A) a single straight neurite; (B) a neurite bifurcation; and (C) a neurite tree with three bifurcations. Black arrows point to inlets of the material. Unit for color bars: $\mu m/s$

After calculating the velocity field, we simulated the dynamic distribution of materials for these three models by solving the transport equations. The results are shown in Fig. 8. In each model, two different filament detachment rates $k' = 0.1 s^{-1}$ and $k' = 0.5 s^{-1}$ were selected to examine how the detachment from filaments affects the transport efficiency in neurites. In principle, higher detachment rate should yield a shorter movement of particles along filaments and therefore result in lower transport efficiency. Indeed, when we compared the propagation of transported particle population under the lower detachment rate of $0.1 s^{-1}$ vs. under the higher detachment rate of $0.5 s^{-1}$, we found that the propagation was faster under the lower detachment rate. Furthermore, we found that geometry affected the velocity field inside neurites and in turn affected the spatial distribution of transported particle population. In particular, for the one-bifurcation model (Figs. 8(C) & 8(D)), transport exhibited faster propagation in the left branch than in the right branch. Similar observation was obtained in the neurite tree of three bifurcations (Figs. 8(E) & 8(F)). Taken together, our simulation reveals that geometry of neurite networks plays an important role in mediating material transport with them.

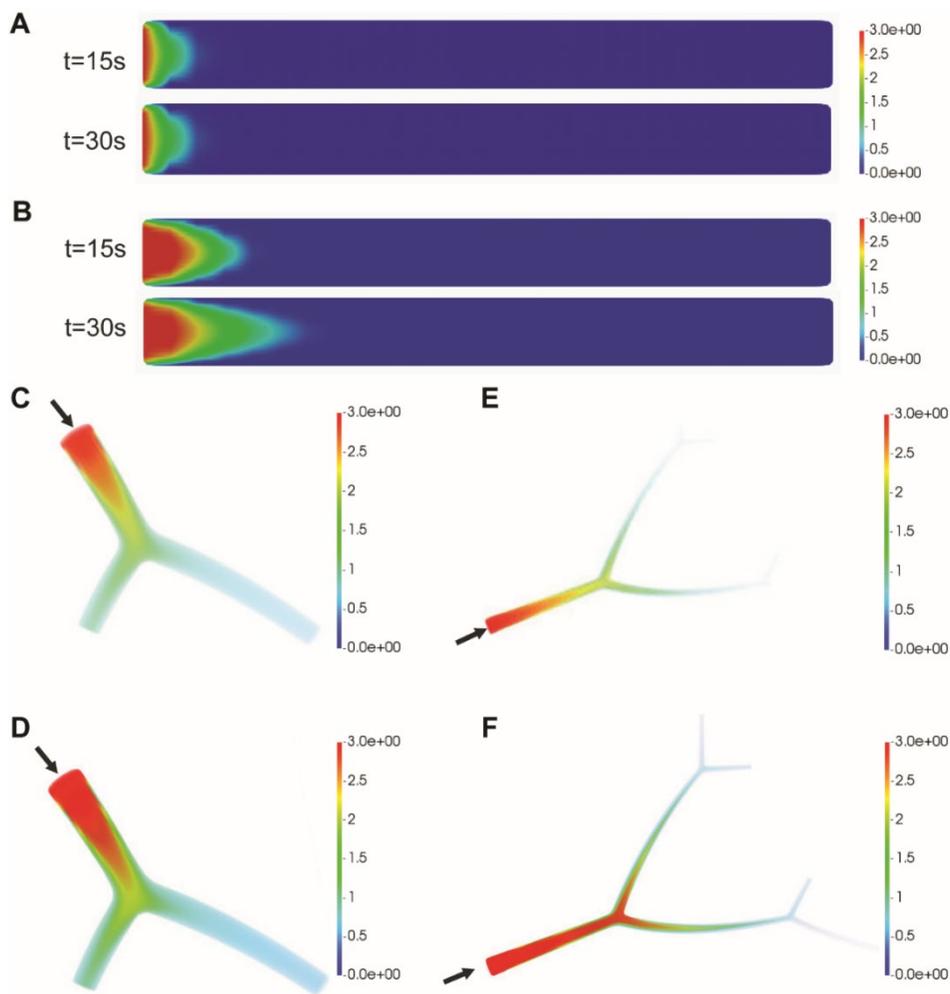


Figure 8: Simulation results for three simple neuron geometries with the filament detachment rate of (A, C, E) $k' = 0.5 \text{ s}^{-1}$ and (B, D, F) $k' = 0.1 \text{ s}^{-1}$. (A, B) The distribution of materials at time 15 s and 30 s on the longitudinal section of the single neurite. Steady-state distribution of materials for (C, D) the single bifurcation and (E, F) the neurite tree with three bifurcations. Black arrows point to inlets of material. Unit for color bars: $\text{mol}/\mu\text{m}^3$

4.3 Simulation of Material Transport in Complex Geometry of Neurite Networks

To test the robustness of our IGA solver, we simulated material transport in three complex neurite networks shown in Figs. 1(A)-1(C). We ran our simulation on the XSEDE (Extreme Science and Engineering Discovery Environment) supercomputer Bridges [43] in the Pittsburgh Supercomputer Center. The computation details were summarized in Tab. 2. Specifically, we assumed a diffusion coefficient $D = 1.0 \mu\text{m}^2/\text{s}$; an inlet transport velocity $v_i = 1.0 \mu\text{m}/\text{s}$; an attachment rate $k = 1.0 \text{ s}^{-1}$; and a detachment rate $k' = 0.5 \text{ s}^{-1}$ based on the standard parameter settings in [11]. Boundary conditions in Eq. (7) were set using $n = 2.0 \text{ mol}/\mu\text{m}^3$, $\tilde{n} = 1.0 \text{ mol}/\mu\text{m}^3$, $\lambda = 0.5$ and $\tilde{\lambda} = 0.2$. In simulations, n and λ are assumed high since materials are being synthesized at the neuron cell body, while \tilde{n} and $\tilde{\lambda}$ are assumed low since materials are being utilized during the transport. The steady-state velocity fields for these three networks are shown in Figs. 9(B), 10(B) and 11(B), respectively. Their distributions of transported materials are shown in Figs. 9(C), 10(C) and 11(C).

Although it may be possible to intuitively predict the distribution of transported materials under simple neurite geometry, such prediction would not be feasible under complex geometry of neurite networks in our simulation. Overall, our simulation shows how material transport is mediated by the geometry of neurite networks. In particular, we found that concentration clearly varied at different neuron terminals (Figs. 10(C) & 11(C)). In addition, we also observed that the concentration shows obvious difference in a small segment along the branch (see the red ellipse in Fig. 11(C)), and the material tends to accumulate in the region with high density of branches. The simulated spatial distribution of transported materials indicates that the neuron may control the delivery of material through its complex geometry.

Table 2: Statistics of three complex geometries

NeuroMorpho.Org ID	Mesh (vertices, elements)	Navier-Stokes computation (nodes, time (hrs))	Transport computation (nodes, time (hrs))
NMO_66731 (Fig. 1(A))	(127,221, 112,500)	(8, 1.5)	(8, 6.3)
NMO_66748 (Fig. 1(B))	(282,150, 249,660)	(10, 2.1)	(10, 9.1)
NMO_00865 (Fig. 1(C))	(1,350,864, 1,179,900)	(40, 6.2)	(20, 26.1)

Note: Each node in Bridges has 28 cores.

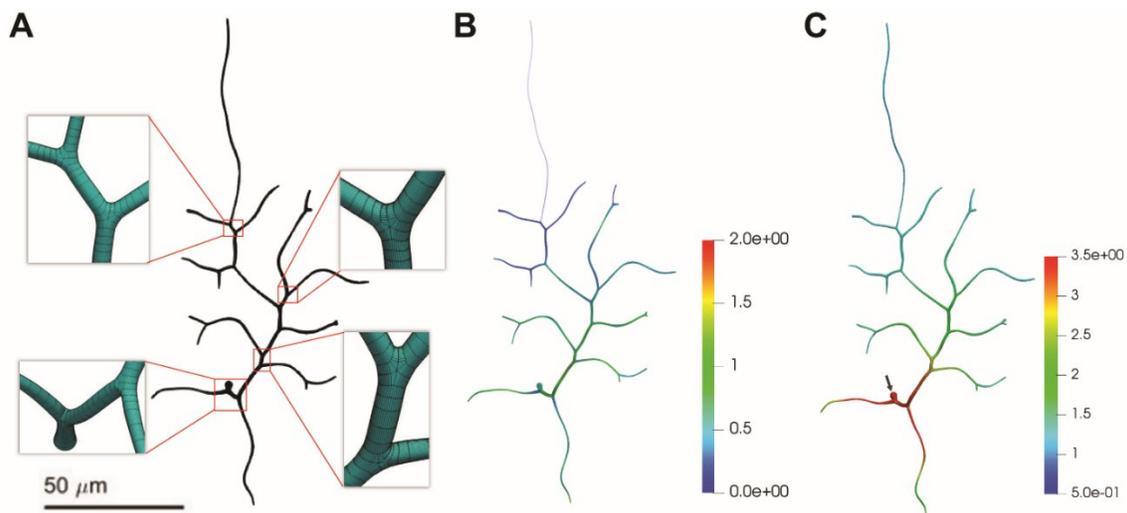


Figure 9: Simulation of material transport in the neurite network geometry of NMO_66731. (A) Hexahedral meshes of the network with zoom-in details. (B) Velocity field of the network. (C) Distribution of transport materials at 10 s, and the black arrow points to the inlet of material. Scale bar: 50 μm. Unit for color bars: (B) μm/s and (C) mol/μm³

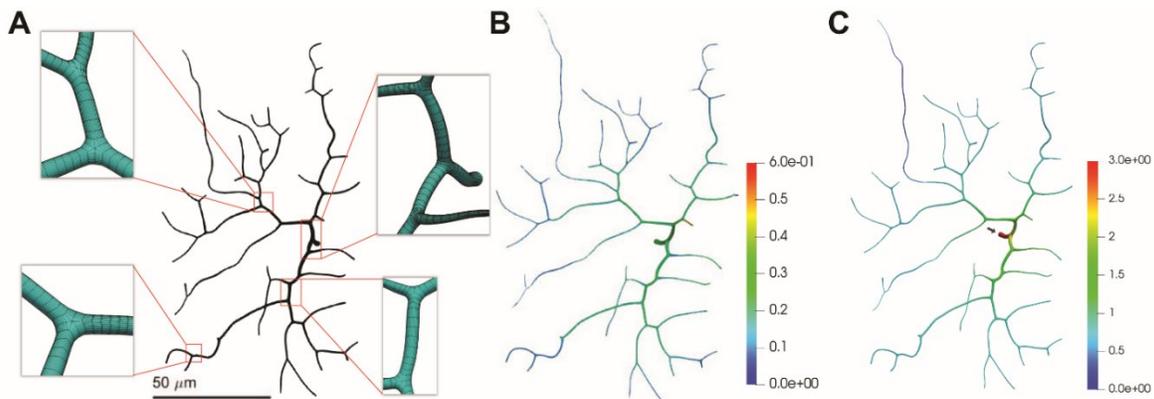


Figure 10: Simulation of material transport in the neurite network geometry of NMO_66748. (A) Hexahedral meshes of the network with zoom-in details. (B) Velocity field of the network. (C) Distribution of transport materials at 10 s, and the black arrow points to the inlet of material. Scale bar: 50 μm . Unit for color bars: (B) $\mu\text{m}/\text{s}$ and (C) $\text{mol}/\mu\text{m}^3$

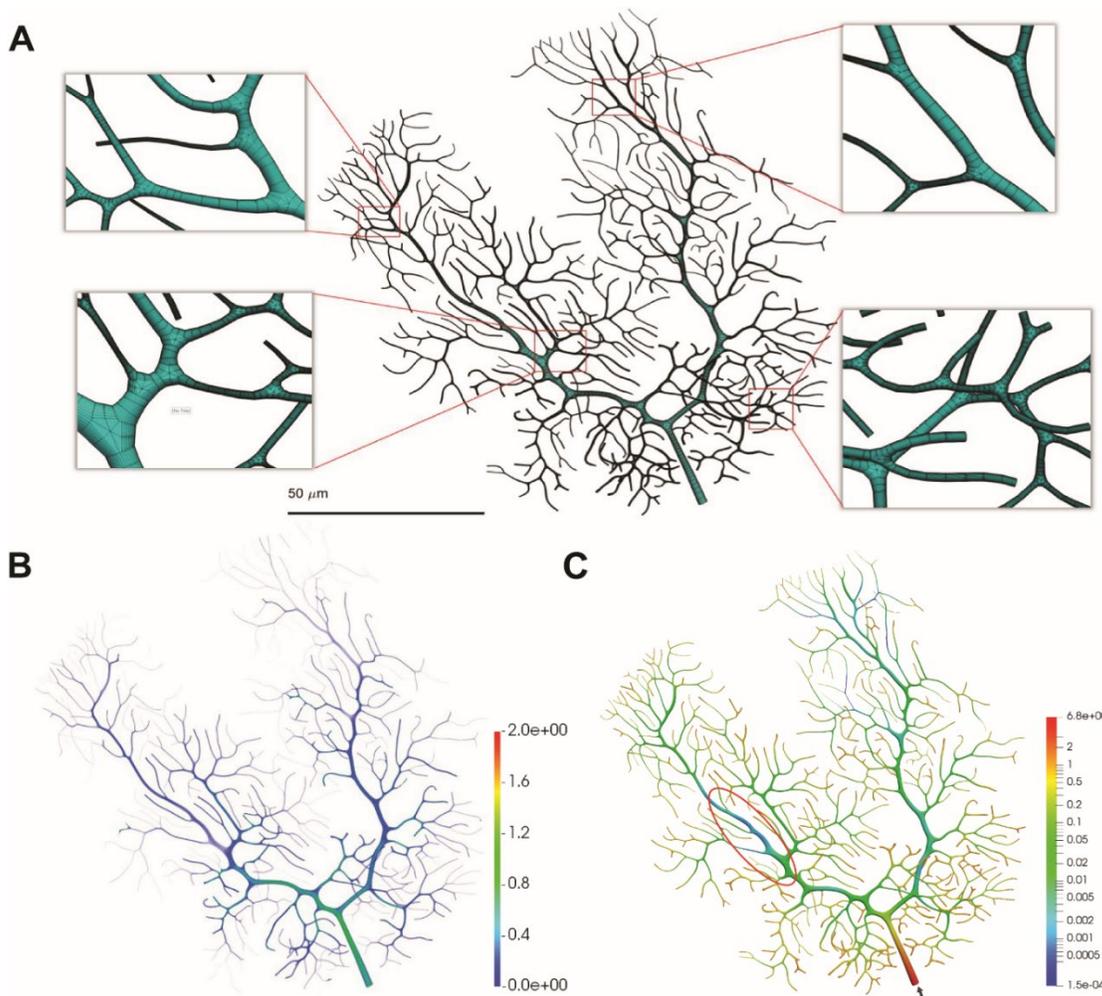


Figure 11: Simulation of material transport in the neurite network geometry of NMO_00865. (A) Hexahedral meshes of the network with zoom-in details. (B) Velocity field of the network. (C) Distribution of transport materials at 10 s (logarithmic scale is used to highlight the distribution pattern), and the black arrow points to the inlet. Scale bar: 50 μm . Unit for color bars: (B) $\mu\text{m}/\text{s}$ and (C) $\text{mol}/\mu\text{m}^3$

5 Discussion

In this study, we developed an IGA-based computational platform for material transport simulation in complex geometry of neurons. Specifically, we developed mathematical models, geometric representations, and numerical solvers for simulating material transport in complex neurite networks. Using our simulation, we examined the transport process in three representative yet complex neurite networks and discovered several spatial patterns of the transport process. First, the distribution of transported materials in neurite networks generally exhibits a spatial gradient. Neurites closer to the neuronal cell body usually are thicker than those farther away from the cell body, and thicker neurites usually exhibit higher concentrations of transported materials (Figs. 9(C), 10(C) & 11(C)). Moreover, the density of branches also influences the material distribution and the region with high density of branches tends to have high concentration, which may cause traffic jam over there. Second, terminal branches of neurite networks usually are thinner than the other branches, and they tend to have lower spatial concentrations of transported materials. Furthermore, terminal branches in different regions of neurite networks may have different concentrations of transported materials (Figs. 9(C) & 10(C)). Third, asymmetry in geometry of neurite branches from the same junction causes asymmetry in transport velocities, which in turn causes asymmetry in concentrations of transported materials (Figs. 7(B) & 8(D)). Sudden increase of velocity in short branches leads to the accumulation of material, which may explain the high demand of material for the growth of short branches. Overall, our results show how the complex network geometry mediates spatial patterns of transport velocities at neurite junctions and within different branches. The spatial patterns of transport velocities in turn drive different distributions of transported material in different regions of neurite networks, especially at their terminals. In this way, our study provides key insights into how material transport in neurite networks is mediated by their complex geometry.

Complex and diverse geometry of neurons has long been thought to have important functional implications. So far, studies of the functional significance of neuronal geometry have focused on how it mediates the transmission of electrical signals. Here we show that neuronal geometry also plays critical roles in mediating distribution of transported materials, including biochemical signals. In this way, our study provides insights into relations between the form and the function of neurons by bringing in a new material transport perspective.

Our study also developed an IGA solver for solving PDEs in complex geometry of neurite networks. Specifically, mesh generation for representing neuron geometry was performed using our skeleton-based sweeping method, which is very suitable for the tree structures of neurite networks. When provided with the skeleton and branch diameters of neurite networks, our method can automatically reconstruct the network geometry with high accuracy and high order of continuity for IGA computation. Overall, our automatic IGA solver provides an efficient computation tool for studies of material transport in complex neurite networks. In this study, we used the solver to solve both the Navier-Stokes equations and the transport equations. It can also be extended to solve other PDE models of cellular processes in complex neurite network geometry.

Our study also has its limitations, which we are addressing in our ongoing work. First, the reaction-diffusion-transport equations only provided a simplified model of the actual transport process. It did not, for example, take into account active regulations of the transport process. Nor did it take into account biochemical reactions of transported materials with other intracellular components. Second, our simulation was based on the assumption that the velocity field of the material transport follows the steady state Navier-Stokes equations and the dynamic change of the velocity field was not considered in the current model. We also lack the experiment data to verify our assumption. Third, although IGA offered great advantages in accurately simulating material transport in neurite networks with complex geometry, the computational cost of simulating transport in large-scale neurite networks remained very high. This may limit the application of this technique. Despite these limitations, our simulation directly showed how the geometry of neurite networks could greatly influence the material traffic within them. The simulation results provide references for the comparison with actual transport in real neurons to further answer the question of how neurons

deliver the right material to the right destination in a balanced fashion with their complex neurite networks and how the transport may be affected by disease conditions.

Source code availability

The source code for our simulation pipeline is available for download from a public software repository located at <https://github.com/truthlive/NeuronTransportIGA>. The User's Guide, describing how to compile and run the code, may also be found in this link. All the input files to generate results in the paper are available from the GitHub link. Other datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgements: We would like to thank Xiaodong Wei for providing the code to create volumetric spline models and Hugo Casquero for providing the Navier-Stokes solver. The authors acknowledge support of the NSF grant CBET-1804929 and a collaborative research seed grant from Department of Mechanical Engineering, Carnegie Mellon University. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the NSF grant ACI-1548562. Specifically, it used the Bridges system, which is supported by the NSF grant ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

References

1. Parekh R, Ascoli GA. Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron* **2013**, 77(6): 1017-1038.
2. Dong X, Shen K, Bülow HE. Intrinsic and extrinsic mechanisms of dendritic morphogenesis. *Annual Review of Physiology* **2015**, 77: 271-300.
3. Kalil K, Dent EW. Branch management: mechanisms of axon branching in the developing vertebrate CNS. *Nature Reviews Neuroscience* **2014**, 15(1): 7-18.
4. London M, Häusser M. Dendritic computation. *Annual Review of Neuroscience* **2005**, 28: 503-532.
5. Segev I, London M. Untangling dendrites with quantitative models. *Science* **2000**, 290(5492): 744-750.
6. Swanger SA, Bassell GJ. Dendritic protein synthesis in the normal and diseased brain. *Neuroscience* **2013**, 232: 106-127.
7. Steward O, Schuman EM. Compartmentalized synthesis and degradation of proteins in neurons. *Neuron* **2003**, 40(2): 347-359.
8. Hirokawa N, Takemura R. Molecular motors and mechanisms of directional transport in neurons. *Nature Reviews Neuroscience* **2005**, 6(3): 201-214.
9. Maeder CI, Shen K, Hoogenraad CC. Axon and dendritic trafficking. *Current Opinion in Neurobiology* **2014**, 27: 165-170.
10. Ruozi B, Belletti D, Bondioli L, De Vita A, Forni F et al. Neurotrophic factors and neurodegenerative diseases: a delivery issue. *International Review of Neurobiology* **2012**, 102: 207-247.
11. Smith D, Simmons R. Models of motor-assisted transport of intracellular particles. *Biophysical Journal* **2001**, 80(1): 45-68.
12. Bird RB, Stewart WE, Lightfoot EN. Transport Phenomena. A Wiley-Interscience Publication, John Wiley & Sons. **2007**.
13. Truskey GA, Yuan F, Katz DF. Transport Phenomena in Biological Systems. Pearson Education, Inc. **2004**.
14. Lazarewicz MT, Boer-Iwema S, Ascoli GA. Practical aspects in anatomically accurate simulations of neuronal electrophysiology. *Computational Neuroanatomy*. Springer **2002**, 127-148.
15. Slepchenko BM, Loew LM. Chapter One-Use of Virtual Cell in Studies of Cellular Dynamics. Academic Press. **2010**, 1-56.
16. Brown SA, Moraru II, Schaff JC, Loew LM. Virtual NEURON: a strategy for merged biochemical and electrophysiological modeling. *Journal of Computational Neuroscience* **2011**, 31(2): 385-400.
17. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact

- geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* **2005**, 194(39): 4135-4195.
18. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons. **2009**.
 19. Lai Y, Zhang YJ, Liu L, Wei X, Fang E et al. Integrating CAD with Abaqus: a practical isogeometric analysis software platform for industrial applications. *Computers & Mathematics with Applications* **2017**, 74(7): 1648-1660.
 20. Zhang Y, Bazilevs Y, Goswami S, Bajaj CL, Hughes TJR. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering* **2007**, 196(29): 2943-2959.
 21. Bazilevs Y, Akkerman I, Benson D, Scovazzi G, Shashkov MJ. Isogeometric analysis of Lagrangian hydrodynamics. *Journal of Computational Physics* **2013**, 243: 224-243.
 22. Lorenzo G, Scott MA, Tew K, Hughes TJ, Zhang YJ et al. Tissue-scale, personalized modeling and simulation of prostate cancer growth. *Proceedings of the National Academy of Sciences* **2016**, 113(48): 7663-7671.
 23. Zhang YJ. *Geometric modeling and mesh generation from scanned images*. Chapman & Hall/CRC Mathematical and Computational Imaging Sciences Series. CRC Press, Taylor & Francis Group. **2016**.
 24. Kiendl J, Bletzinger KU, Linhard J, Wuchner R. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering* **2009**, 198(49-52): 3902-3914.
 25. Casquero H, Liu L, Zhang Y, Reali A, Kiendl J et al. Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff-Love shells. *Computer Aided Design* **2017**, 82: 140-153.
 26. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics* **2006**, 38(4-5): 310-322.
 27. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics* **2008**, 43(1): 3-37.
 28. Casquero H, Zhang YJ, Bona-Casas C, Dalcin L, Gomez H. Non-body-fitted fluid-structure interaction: Divergence-conforming B-splines, fully-implicit dynamics, and variational formulation. *Journal of Computational Physics* **2018**, 374: 625-653.
 29. Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A et al. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* **2007**, 197(1): 173-201.
 30. Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* **1982**, 32(1): 199-259.
 31. Wei X, Zhang YJ, Hughes TJR. Truncated hierarchical tricubic C^0 spline construction on unstructured hexahedral meshes for isogeometric analysis applications. *Computers & Mathematics with Applications* **2017**, 74(9): 2203-2220.
 32. Wei X, Zhang YJ, Toshniwal D, Speleers H, Li X et al. Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* **2018**, 341: 609-639.
 33. Ahrens J, Geveci B, Law C. Paraview: an end-user tool for large data visualization. *The Visualization Handbook* **2005**, 717.
 34. Cuntz H, Forstner F, Borst A, Husser M. One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS Computational Biology* **2010**, 6(8): 1-14.
 35. Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering* **2011**, 87(1-5): 15-47.
 36. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering* **2011**, 88(2): 126-156.
 37. Scott MA, Simpson RN, Evans JA, Lipton S, Bordas SP et al. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* **2013**, 254: 197-221.

38. Glover F, Laguna M. Tabu search. Handbook of Combinatorial Optimization. Springer. **1998**, 2093-2229.
39. Kerner BS, Konhäuser P. Cluster effect in initially homogeneous traffic flow. *Physical Review E* **1993**, 48(4): 2335.
40. Hughes TJR, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering* **1989**, 73(2): 173-189.
41. Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics* **1991**, 28: 1-44.
42. Tezduyar T, Park Y. Discontinuity-capturing finite element formulations for non-linear convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering* **1986**, 59(3): 307-325.
43. Towns J, Cockerill T, Dahan M, Foster I, Gaither K et al. XSEDE: accelerating scientific discovery. *Computing in Science & Engineering* **2014**, 16(5): 62-74.