

Toolbox for Diffusion-weighted MRI simulations of Realistic White Matter Microstructures

Bioengineering Group Project Final Report

Group Name: Gradient Explode

Group Member:

Ho Pak Hei, Bob Qian, Catherine Ni, Jane Ren, Jiayi Bai, Victor Kam, Kian Daneshi

Supervisor:

Dr. Amy Howard

14 April 2025

Word count: 4981

Abstract

Diffusion-weighted Magnetic Resonance Imaging (dwMRI) is a medical imaging technique that provides enhanced sensitivity to tissue microstructure by tracking water diffusion at microscopic scales. In human brain white matter (hWM), where neurons form dense axonal bundles, dwMRI captures diffusion patterns that reflect tissue microstructure and potential pathologies. While existing dwMRI methods successfully image many in-vivo and ex-vivo scenarios, signal acquisition is dependent on idealizations, limiting signal accuracy and ability to infer tissue features. Simulation-based approaches can overcome these issues by generating biomimetic meshes, subsequently processed by dwMRI simulators. Yet, existing tools are difficult to adopt due to low accessibility and flexibility.

To address these challenges, we present a mesh generation toolbox based on Blender to produce realistic microstructural meshes of hWM. Meshes are made customizable to allow investigation of how cellular complexity influences observed signals simulated by dwMRI simulator (e.g. MCMR). To simplify usage for researchers without advanced programming expertise, we developed user-friendly wrapper functions and integrated a generative AI guide for setup and simulation. Furthermore, this pipeline also supports various advanced diffusion models to enable detailed investigations on hWM tissue organisation. This is assessed against analytical models, followed by a neurodegenerative case study to demonstrate the pipeline's ability to model pathological conditions.

By providing a flexible, accessible, and accurate simulation pipeline, this project aims to enhance the development of dwMRI acquisition protocols and advance our understanding of brain microstructure in health and disease.

Acknowledgements

We would like to express sincere gratitude to our supervisor Dr. Amy Howard, for her continued guidance, feedback and support throughout the project. We would also like to thank Dr. Michiel Cottaar for providing full support and feedback on our developed code, and for the development of MCMR, which greatly contributed to the backbone and progress of our research.

Contents

1. Introduction

- 1.1 Introduction to dwMRI**
- 1.2 Problem Identification & Justification of Aims**
- 1.3 Project Aims**

2. Methods

- 2.1 Axon Fibre Mesh Generation**
- 2.2 Neuron and Glia Mesh Generation**
- 2.3 Mesh Optimization**
- 2.4 MRI & MCMR Packages and Wrappers**
- 2.5 Agentic AI Coder**
- 2.6 Ethics**

3. Results

- 3.1 Digital Reconstruction of Human White Matter Microstructures**
- 3.2 Parametric Control**
- 3.3 Comparative Analysis of Healthy and Diseased Models**
- 3.4 MCMR Results**
- 3.5 ND Case Study: Amyotrophic Lateral Sclerosis (ALS)**
- 3.6 MCMR Wrapper Usability Results**
- 3.7 Agentic AI Coder Results**

4. Discussion

- 4.1 Blender Mesh Discussion**
- 4.2 MCMR Results Discussion**
- 4.3 ALS Case Study Discussion**
- 4.4 MCMR Wrapper and Add-on Usability Discussion**
- 4.5 Agentic AI Coder Discussion**
- 4.6 Conclusion and Future Work**

5. Reference

6. Appendix

- 6.1 Project Planning**
- 6.2 Lessons Learned**
- 6.3 Group Dynamics & RACI Matrix**
- 6.4 Example of “Geometry Nodes” Modifier in Blender**

6.5 Full Parameter table

6.6 Appendix & Test Signals for MCMR Wrapper

6.7 ALS Exponential Fitting Matrix

6.8 MCMR Wrapper Usability Questionnaire and Tutorial

6.9 Code Availability

1. Introduction

1.1 Introduction to dwMRI

DwMRI is a non-invasive technique that has become a pillar of modern medical imaging since its advent in the 1980s.¹ It allows for neuroimaging opportunities and diagnoses of human White Matter (hWM) related pathologies (e.g. Acute Brain Ischemia,² Amyotrophic Lateral Sclerosis (ALS), and other demyelination conditions) at microscopic scales.

While MRI scans the brain macrostructurally, dwMRI has higher sensitivity to microstructural tissue properties by additionally measuring water diffusion within these structures, which causes varying signal decays. dwMRI achieves this through applying a pair of diffusion gradients on top of an MRI sequence before signals are readout. This signal decay is given by:

$$\text{Signal} = S_0 e^{-\frac{TR}{t_1}} e^{-\frac{TE}{t_2}} e^{-bD} \approx S_0 e^{-\frac{TE}{t_2}} e^{-bD}$$

Where S_0 denotes the initial signal strength, Apparent Diffusion Coefficient (D) quantifies diffusivity and B -value (b) quantifies strength of diffusion gradients applied. This added decay term atop MRI signal decay (e^{-bD}) thus gives dwMRI added sensitivity.

hWM is comprised of axon bundles, glial cells, and other obstructions. These surrounding obstructions restrict water diffusion in different directions and extents. Such differences in restricted diffusion are hence detected by dwMRI, leading to unique diffusion patterns that detail the underlying microstructure.

1.2 Problem Identification & Justification of Aims

Nevertheless, several hardware and protocol limitations are associated with both in-vivo and post-mortem dwMRI-based techniques. The former is often motion-sensitive and artifact prone; the latter requires immediate brain tissue fixation to prevent tissue autolysis and bacterial decomposition. However, fixatives (e.g. Formalin) are known to undesirably interfere with MR properties.³

In clinical and preclinical protocols, inverse models aim to infer hWM microstructure from obtained dwMRI signals, while forward models predict those signals from known tissue structures. Current inverse models often rely on simplified forward assumptions that do not reflect in-vivo conditions, limiting the accuracy of interpretations. Therefore, developing biologically-grounded forward models improve inverse inference, making dwMRI signals more interpretable and meaningful.

As above, the compromised fidelity and reproducibility of dwMRI scans, and idealizations made within signal acquisition prompt researchers to adopt simulation-based methods, justifying our simulation-based approach.

Several software modalities, namely analytical and Monte-Carlo (MC) simulators, support dwMRI simulations where geometries are defined to yield an output signal. However, idealizations are made in the former, compromising accuracy.⁴ MC simulators such as the Camino toolkit overcome these issues but is computationally slow.⁵ Newer MC simulators, such as MCMR, run more efficiently⁴ and is our tool of choice, but requires extensive research into documentation and high coding expertise.

The accuracy of simulation outcomes also relies on the fidelity of the underlying mesh. Databases such as NeuroMorpho⁶ provides digitally accurate neuronal structures, including axon fibre tortuosity and soma radius.^{7,8} However, they lack the flexibility required to adjust cellular features, which is essential for exploring characteristics associated with neurodegenerative pathologies that influence dwMRI signals.

Published hWM-mesh generators, such as MEDUSA and CONFiG were shown in research to be capable of generating versatile and realistic hWM meshes.^{9,10} However, at the time of writing, the software had not yet been released as open-source. This limits broader adoption and highlights the need for accessible, efficient simulation toolkits that support more advanced tissue complexity.

1.3 Project Aims

The goal of this project is to develop a flexible dwMRI simulation tool that reduces the complexity of modeling hWM while retaining accuracy in forward-modelling. We aim to create customizable meshes that can reflect both healthy and pathological conditions, by using general modelling software Blender. Corresponding signals will be acquired by establishing streamlined MCMR add-ons, alleviating demand for extensive coding and usage difficulty.

The final tool will be open-source, enabling researchers to efficiently generate realistic tissue meshes and simulate corresponding dwMRI signals while retaining pipeline transparency and accessibility. This contributes to the advancement of both forward and inverse modeling approaches, helping to establish more accurate relationships between dwMRI signals and underlying brain microarchitecture, with potential applications in preclinical and clinical diagnostics.

2. Methods

This section outlines the pipeline for simulating and generating dwMRI simulation signals. Our pipeline integrates 3D mesh-modelling using Blender v4.2, dwMRI simulation using MCMR simulator, and an AI coding assistant to streamline workflows.

2.1 Axon Fibre Mesh Generation

As mentioned, specialised hWM-meshing tools were unavailable, compelling us to turn to general modelling software. Within these, Blender was chosen for its well-documented use in multiple literatures and software maturity.¹¹

In previous research, an existing Blender addon BlenderNEURON was chosen to acquire cell data from NeuroMorpho and model neuron networks in Blender.¹² However, this addon lacks well-defined cell morphological parameters, hindering cell morphology. In addition, its complex integration procedures into Blender user-friendliness. Therefore, we turn to the basic modelling software Blender.

We use Blender's "Geometry Nodes" modifier which builds 3D meshes procedurally using connected node networks ([Appendix 6.4](#)). Modifiers in Blender are computational operations that apply property changes to objects. They are managed in the "Properties" Panel, offering a preview of applied geometric transformations.¹³ Blender's Nodes are building blocks that connect in various ways to achieve the desired visual effects. This workflow allows for flexible manipulation of geometric elements while maintaining dynamic control over the final output ([Appendix 6.4](#)).¹⁴

The modelling pipeline generates neuronal microstructures including axon fibre bundle, neuron and glia. All meshes are optimized for compatibility with the MCMR simulation environment ([Figure 1](#)).

2.1.1 Unit Fibre Mesh Generation

Each axon fibre begins as a Bézier curve and is converted to a tubular mesh using both "Mesh-to-Curve" and "Curve-to-Mesh" nodes. The "Curve Circle" node is used to aid mesh formation, controlling resolution and radii. The recommended range of radius is set to 0.17-4.58µm ([Table 5](#)) to match real axon morphology. Tortuosity is the ratio of an axon's actual length to its straight-line distance.⁷ This property is introduced sequentially through a "Noise Texture" node and math nodes controlled by four key parameters ([Table 1](#)).

Table 1. Functional controls of unit fibre parameters

Parameter	Control
Tortuosity Level	Spatial frequency of bends (does not affect face count)
Tortuosity Detail	Smoothness of bends
Noise Offset	Overall curvature intensity
Noise Seeds	Randomization pattern of bends

For fibre bundle assembly, individual fibres undergo translation, rotation, and scaling via a "Transform Geometry" node ([Section 2.1.2](#)). Adjusting these parameters at the unit fibre level propagates changes to the entire bundle, enabling precise control over fibre dispersion and orientation.

2.1.2 Fibre Bundle Assembly

To construct a 3D axon bundle, we first distribute points on a square plane following the Poisson distribution, achieved by the "Distribute Points on Faces" node. Unit fibres are then projected onto these points. Using the same node, we can control the axonal volume fraction by changing density parameters ([Figure 4e](#)) and minimum distance between axons. The fibre projections are then converted into mesh via "Realize Instance" node to enable further geometric processing.

To enhance biological fidelity, we introduced controlled randomness through a "Random Value" node that simultaneously modulates two key characteristics: angular dispersion for generating natural fibre crossing patterns and diameter scale variation across the bundle.

2.1.3 Confinement Box

In computer graphics, a voxel represents a volumetric pixel. To generate voxel-like structures that accurately represent segmented hWM for simulation inputs, we developed a confinement box tool using Blender's "Geometry Nodes" modifier. This tool operates through a cube that intersects the fibre bundle mesh (generation visualised in [Figure 1](#)) at the desired locations. This is done by resizing, repositioning and rotating the cube to select specific regions of interest for analysis.

This clipping process can occasionally produce topological artifacts. To ensure clean geometry, we implemented a subsequent "Delete Geometry" node to remove all residual points, edges, and faces on the cube's surfaces.

2.2 Neuron and Glia Mesh Generation

In addition to axons, we developed models of neurons, and various glial cells (e.g. astrocytes, microglia), along with their respective cellular features (Figure 1). The geometry is designed with adjustable parameters to accommodate both typical morphologies and pathological variations. ([Appendix 6.5](#) for ALS parameter)

2.3 Mesh Optimization

Prior to simulation, all generated meshes are optimised for computational efficiency. Here, we use the "Decimate" Modifier to achieve a 70-80% decrease in face-count, while preserving all morphological features and minimizing loss of geometric resolution. We then triangularize these faces via the "Triangulate" Modifier, furthering compatibility with subsequent simulations.

The generated meshes are simulated via an automated wrapper for subsequent simulations in [2.4](#). Detailed specifications of these mesh files, along with the full parameter setting, are provided in [Appendix 6.5](#) for research reproducibility and further customisation.

2.4 MRI & MCMR Packages and Wrappers

In accordance with the introduction, MCMR, a MC-based dwMRI simulator was chosen over analytical models due to their ideal assumptions of water molecule behaviour, which contribute to inaccuracies in forward models. Among readily available MC simulators, the Julia-based MCMR is computationally efficient and is highly functionalized, therefore justifying our software choice.

However, its high coding baseline knowledge requirement could hinder researchers. Therefore, our team has developed a wrapper function, allowing detailed analysis and interpretation of signals generated from user-defined meshes (see [Sections 2.1~2.3](#)) while ensuring adequate simplicity and usability. Completing the pipeline, optional Advanced Models can be applied for further microstructural analysis. Below, we outline the necessary steps in achieving this objective.

2.4.1 Signal Acquisition Protocol

The signal acquisition protocol in MCMR follows the procedure outlined in [Figure 1](#). By defining adequate parameters ([Appendix 6.6](#)), the wrapper condenses the mesh-to-signal conversion into a one-step process through calling a single function.

As stated in [2.1~2.3](#), compatible and highly customizable Blender meshes serve as an input to the subsequent steps; alternatively, simpler MCMR-derived geometries could be used to lessen computational load.

After mesh importation, we iteratively generate simulated dwMRI sequences. Each sequence is created separately with adjustable settings. The key parameters of this step consist of the following:

Table 2. **a)** (above) describes core simulation parameters of the dwMRI simulation pipeline. Subsequent adjustable parameters are all made available in [Appendix 6.6](#). **b)** (below) describes the formula to calculate B-value, and the parameters that influence B-value calculations. Details in [Section 3.4](#) and [3.5](#).

Parameter	Description
B-value	Controls the sensitivity of the scan to water diffusion in tissues, hence the extent of signal decay. Can be adjusted post-hoc.
Echo Time (TE)	The time between the excitation pulse and signal measurement. It influences the degree of signal decay.
Repetition Time (TR)	The time between successive simulation excitation pulses. Alternatively, this defines the duration of each scan cycle.
Gradient Orientation (Grad)	Specifies the direction of the diffusion gradient. It can be iterated across multiple directions to capture anisotropic diffusion.

B-value Equation	$\mathbf{B} = \mathbf{G}^2 \delta^2 \gamma^2 (\Delta - \frac{\delta}{3})$
Parameters	Description (units)
Gradient Strength (G)	Strength of Diffusion gradients applied (mT/m)
Gradient Duration (δ)	Duration of Diffusion gradients applied (ms)
Diffusion Time (Δ)	Duration between Diffusion gradients applied (ms)
Gyromagnetic Ratio (γ)	Constant (rad/s/T)

After defining these parameters, they will be applied to the mesh during the simulation step. A timeframe per scan cycle is specified by the user to record signals. Here, users can customize MCMR-based tissue properties and hence their influence on signal decay.

After simulation completion, the user indexes the desired scan output by first specifying the scan cycle of interest, then specifying the B-value and gradient orientation if multiple simulations are being run. The user is also given an option between outputting intracellular and extracellular signals.

Finally, the corresponding output signal is returned to the user. By default, this is a transverse signal time (TS-time) plot, recorded over the specified timeframe. However, by applying optional advanced models, the user can further generate alternative outputs ([Table 3a](#)).

2.4.2 Advanced Models for MCMR

To extract further microstructural information from the above signals generated from the wrapper, users can optionally apply advanced analytical models as we have developed. At the time of writing, we have developed several axon-oriented models for flexibility (parameters detailed in [Table 3b](#)).

Table 3. a) (Above) describes all Advanced Models applicable in the optional final step, to extract further information from the signals generated from the desired meshes. Brief descriptions of their features and expected outputs are listed as follows. **b)** (Below) details all relevant parameters used to produce our results.

Model	Features Observed	Methods	Output
High Angular Resolution Diffusion Imaging (HARDI)	Sensitive to axon crossing and orientation dispersion.	Applies uniformly distributed gradient directions over a spherical shell with respective transverse signal strengths	3D polar plot showing transverse signal strength for each direction.
Axon Diameter Mapping (ADM)	Reveals axon-size dependent water diffusion restrictions.	For every mesh defined, measures TS at a fixed time (TE) per simulated B-value	Plot of transverse signal vs. b-value for each radius.
Diffusion Time-Dependent ADC Analysis (DT-ADC)	Reveals axon-size dependent water diffusion restrictions.	Fixes b-value while linearly increasing diffusion time; estimates ADC from each cycle.	Plot of ADC vs. diffusion time for each radius.

Analysis Model	Independent Variable	Independent Value (units)	Control Variables	Control Value (units)
ADM	Gradient Strength (G)	0~8.5 (mT/m)	Gradient Direction	[0, 1, 0], y-axis
			Diffusion Time (Δ)	50 (ms)
HARDI	Gradient Direction	Spherically sampled across 500 points	B-value (G, δ , Δ , γ)	B = 0.5 (ms/ μm^2)
DT-ADC	Diffusion Time (Δ), Gradient Strength (G)	$\Delta = 5\sim70$ (ms) G = 224~898 (mT/m)	B-value (δ , γ)	B = 1.0 (ms/ μm^2)

For computational efficiency, regularly-spaced cylinders that mimic axons of radii 1, 3, 5, 7, 9 μm were investigated for the ADM and DT-ADC modalities; while for HARDI, we used axons of radii 1 μm . This follows the consideration that histological studies of human brains have reported axon radii (r_{axon}) ranging from 0.25~9 μm , where most lie $\sim 1\mu\text{m}$.¹⁵ These axons extend in the z-direction, and the xy-plane depicts their cross-sectional areas. TE and TR were fixed respectively as 100 and 500ms respectively for all models.

Additional to these models, we have provided the option to export signal outputs into MATLAB, where fitting models can be applied to approximate the exponential or logarithmic function governing the signal. This provides convenience in later comparisons against analytical models (see [results](#)).

Through applying advanced models and combining all previous steps, we complete the building blocks of the pipeline: from generating accurate brain-microstructural meshes to obtaining and analysing

corresponding output signals from an MC simulator. This serves as a chassis to a biologically-grounded forward model. To demonstrate the combined efficacy of our pipeline and its potential clinical/pre-clinical application, a Case Study shall be discussed at length in [Section 3.5](#).

2.4.3 ND Case Study: Amyotrophic Lateral Sclerosis (ALS)

Here, we unified both MCMR and Blender pipelines in attempts to model progression of a neuro-degenerative condition. As demonstration, we investigated Amyotrophic Lateral Sclerosis (ALS), characterized by muscle and motor neuronal atrophy.^{16,17} ALS is associated with diverse microstructural changes in hWM. Within our pipeline, these alterations were captured by systematically adjusting relevant MCMR tissue parameters against literature.¹⁸⁻²⁰ Additionally, scanner configurations were varied to reflect preclinical and clinical dwMRI devices ([Table 4](#)).

Table 4. a) (above) describes some hallmark pathologies as seen in Amyotrophic Lateral Sclerosis (ALS), and the corresponding changes in tissue they cause. The specific numerical values were set according to literature.¹⁸⁻²⁰ **b)** describes the expected performances of a typical preclinical and clinical scanner, with numerical values.

Reflected Pathology	Tissue Property Changes (MCMR Parameter Adjustments)	Numerical Value (Units) Healthy/Mild-ALS/Severe-ALS
Neuronal Atrophy & Death	↑ Permeability (permeability) ↓ Axon density/repeating frequency (repeats) ↓ Axon radii (radii)	0 / 15 / 30 (%) 4 / 8 / 12 (μm) 1 / 0.5 / 0.25 (μm)
Protein Aggregation	↑ Isotropic magnetic susceptibility (susceptibility_iso) ↑ Off-resonance frequency shift (off_resonance)	0.03 / 0.055 / 0.08 0.01 / 0.11 / 0.21 (kHz)
Perimembrane Changes	↑ Tissue magnetization rates (R1, R2)	0.2 / 0.5 / 0.8 (ppm) 0.02 / 0.06 / 0.10 (ppm)
Demyelination	↑ G-ratio (g_ratio)	0.5 / 0.75 / 1.0

Scanner Types	Parameters	Max Values
Preclinical Scanner (PS)	B-value	4.0 (s/mm ²)
	Slew Rate	200.0 (mT/m/ms)
	Gradient Strength	50.0 (mT/m)
Clinical Scanner (CS)	B-value	10.0 (s/mm ²)
	Slew Rate	1000.0 (mT/m/ms)
	Gradient Strength	500.0 (mT/m)

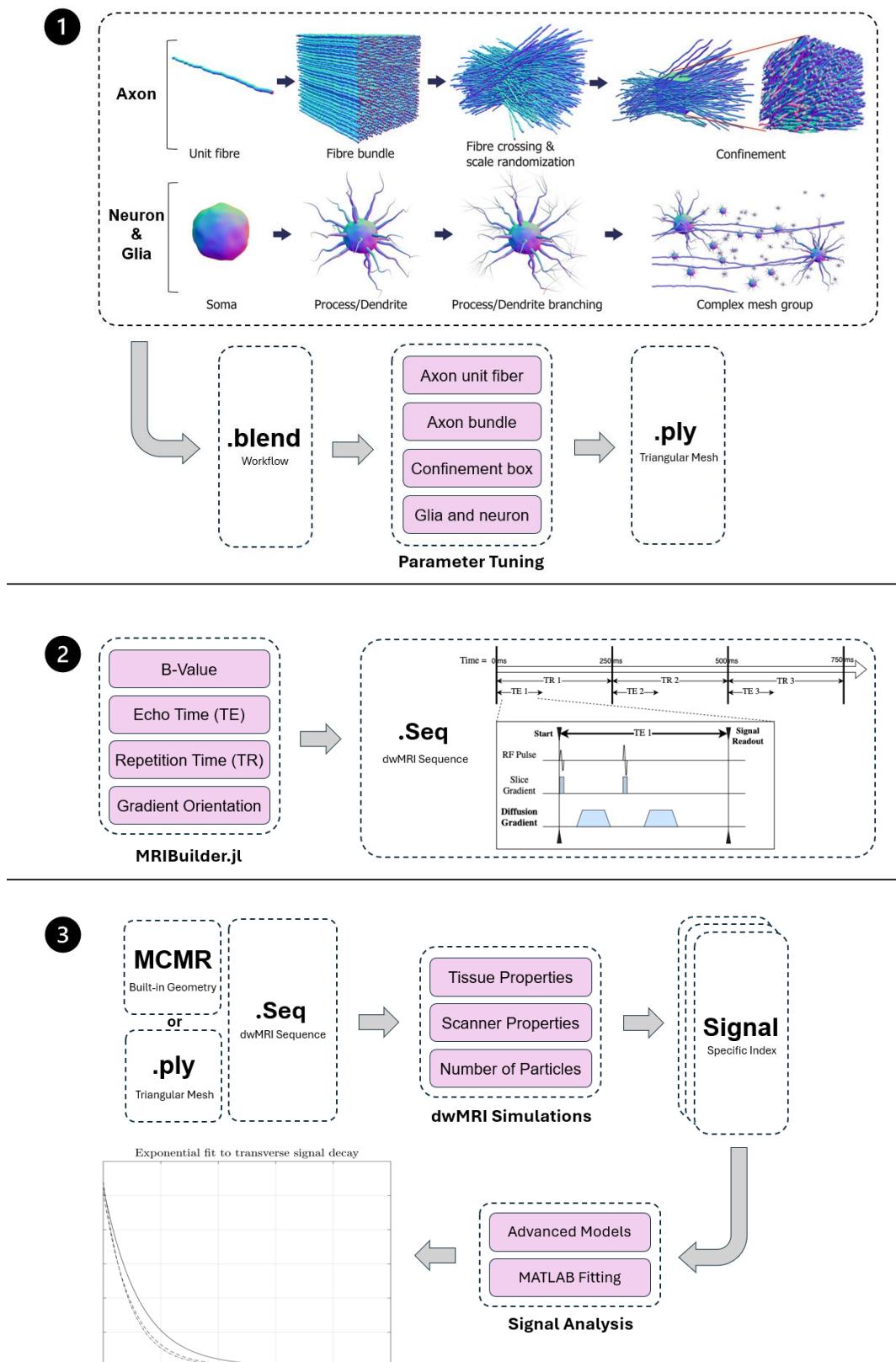
Simulations across all parameters in [Table 4](#) and across three varying states of disease: Healthy, Mild-ALS and Severe-ALS were conducted, yielding $8 \cdot 3 \cdot 2 = 48$ parameter combinations.

We first determined the significance of each tissue parameter by repeating the steps in [2.4.1~2](#) for each simulation. Here, we visualise the TS-B-value plot, similar to the ADM advanced model. These data were exported into MATLAB and an exponential-fitting algorithm was subsequently applied. The normalized intracellular signal strength ($S_0 = \frac{S_{intra}}{S_{0(intra)}}$) and the negative ADC (-D) were quantified relative to healthy control parameters (results summarized in [Appendix 6.7](#)). Simple MCMR geometries are used at this preliminary stage for computational efficiency. For tissue or scanner parameters that led to substantial differences in S_0 and -D across healthy and diseased states, they will be further verified via a secondary test. Here, biomimetic Blender meshes reflecting these parameters changes in healthy and diseased states will be defined and simulated through the MCMR pipeline.

2.4.4 Wrapper Usability

While generating accurate signals is essential, our tools should remain user-friendly as aimed. A combined tutorial–survey approach ([Appendix 6.8](#)) was employed, evaluating usability of the MCMR wrapper. Participants with varying levels of coding experience were invited to work through a tutorial that demonstrates the wrapper’s usage for simulation tasks, after which they completed a survey on their experience. The survey included both rating questions on code clarity and open-ended prompts for improvement suggestions.

Figure 1. Overview of the pipeline for (1) blender mesh generation, (2) MRI sequence setup, and (3) dwMRI simulation and analysis. See Methods [Sections 2.1–2.4](#) for more details.



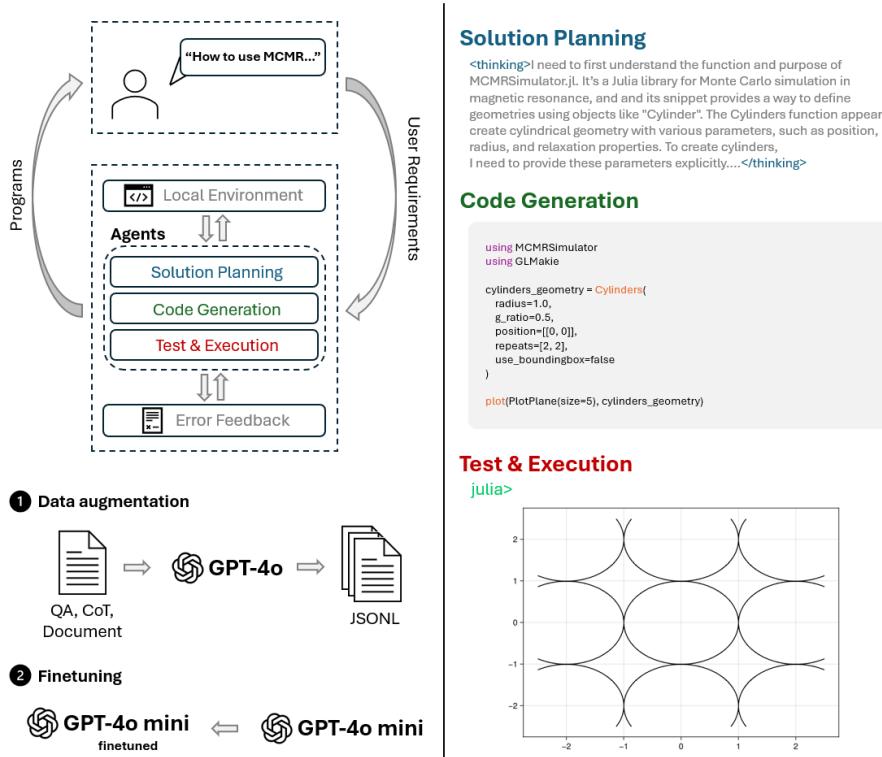
2.5 Agentic AI Coder

Despite efforts to improve usability via a wrapper and tutorial-survey, working with novel, specialized tools with scarce documentations like MCMR can still be challenging. To address this, we built an AI-coding assistant using a method known as agentic coding.²¹ We achieve this through Roo Code, an open-source framework that guides LLMs to read, plan, write, and test code iteratively.

To further facilitate usability, we extended Roo Code²² to support Julia, optimized prompt templates for precision, and fine-tuned the underlying LLM (GPT-4o mini) on MCMR datasets. We fine-tune the model using simple Q&A pairs, annotated code snippets, and structured information extracted from MCMR documentations. To expand upon this dataset, we used GPT-4o to augment and diversify our datasets, hence solving dataset scarcity. All data was converted to the required JSONL format and structured as user-assistant dialogues. The full solution was packaged into a seamless extension available on the Microsoft VSCode, increasing accessibility to users.

Finally, we created a custom evaluation dataset to assess code generation accuracy on both ChatGPT and the Agentic AI-Coder. This includes 22 questions paired with ground-truth answers.

Figure 2. (Left) Overview of our agentic coding framework with iterative agents for planning, code generation, testing, and error correction, powered by a fine-tuned LLM (GPT-4o mini) trained on customized MCMR datasets. (Right) Example showing how the AI assistant can plan, write, and run Julia code to visualize cylindrical geometries using MCMRSimulator.jl. See Methods Sections 2.5 for more details.



2.6 Ethics

All parts of the project, Blender and MCMR pipelines in [Figure 1](#), and training data for the Agentic AI-coder were sourced from MCMR documentations available on GitHub, under an open-source Apache license. Data for the usability survey was collected in accordance with ethical terms as given by the Bioengineering Department and will be promptly deleted after report completion.

3. Results

3.1 Digital Reconstruction of Human White Matter Microstructures

Our Blender-based pipeline generated complex, customizable 3D meshes replicating hWM microarchitecture. The reconstructions encompassed three critical biological elements: axonal networks exhibiting realistic tortuosity and crossing fibre patterns; glial cells (astrocytes and microglia) with anatomically accurate soma geometries and process branching architectures; and composite tissue models simulating both healthy and pathological states as mentioned. We validate our digital reconstructions through referencing and comparing against histological data, where our meshes demonstrated high congruence when compared ([Figure 3c-f](#)).

Microscopic images of hWM confirmed visual alignment across key features including neuronal soma morphology, hierarchical clustering of glia and axonal bundle organisation. This morphological validation established preliminary confidence in the biological plausibility of our models prior to dwMRI simulation.

3.2 Parametric Control

To develop the resulting Blender-based mesh generation tool, we implemented a node workflow that constructs the mesh and allows real-time manipulation of microstructural features. Users can dynamically adjust geometric parameters and observe instant visual feedback during modifications ([Table 5](#)) via an interactive "Properties" panel in Blender, equipped with intuitive sliders ([Figure 3g](#), example of corresponding nodes is referred to [Appendix 6.4](#)).

3.3 Comparative Analysis of Healthy and Diseased Models

The modelling pipeline successfully reproduces both healthy and pathological neural microstructures through integrated assemblies of axons, glial cells and neuronal components, as shown in [Figure 3](#).²³⁻²⁷

[Figure 3a](#) demonstrates physiologically normal hWM organisation, while [Figure 3b](#) presents a neurodegenerative condition featuring three characteristic pathological changes implemented according to histopathological research:²³ axonal demyelination represented by reduced fibre diameters, hierarchical clustering of degenerated neural processes, and hypertrophy (swelling) of reactive glial cells. These structural alterations will be simulated using our dwMRI pipeline and validated in [Section 4.3](#).

Figure 3. Comparative visualization of digitally reconstructed neural microstructures. (a) Healthy WM architecture showing normal axonal organization with astrocytes and microglial cells. (b) Neurodegenerative model featuring pathological hallmarks including demyelination (reduced axon diameters), glia clustering, and Hypertrophy (swelling). (c-f) Microscopic image of (c) neuron cell,²⁴ (d) astrocyte,²⁵ (e) microglia²⁶ and (f) axon fibre²⁷ in hWM for meshing reference.

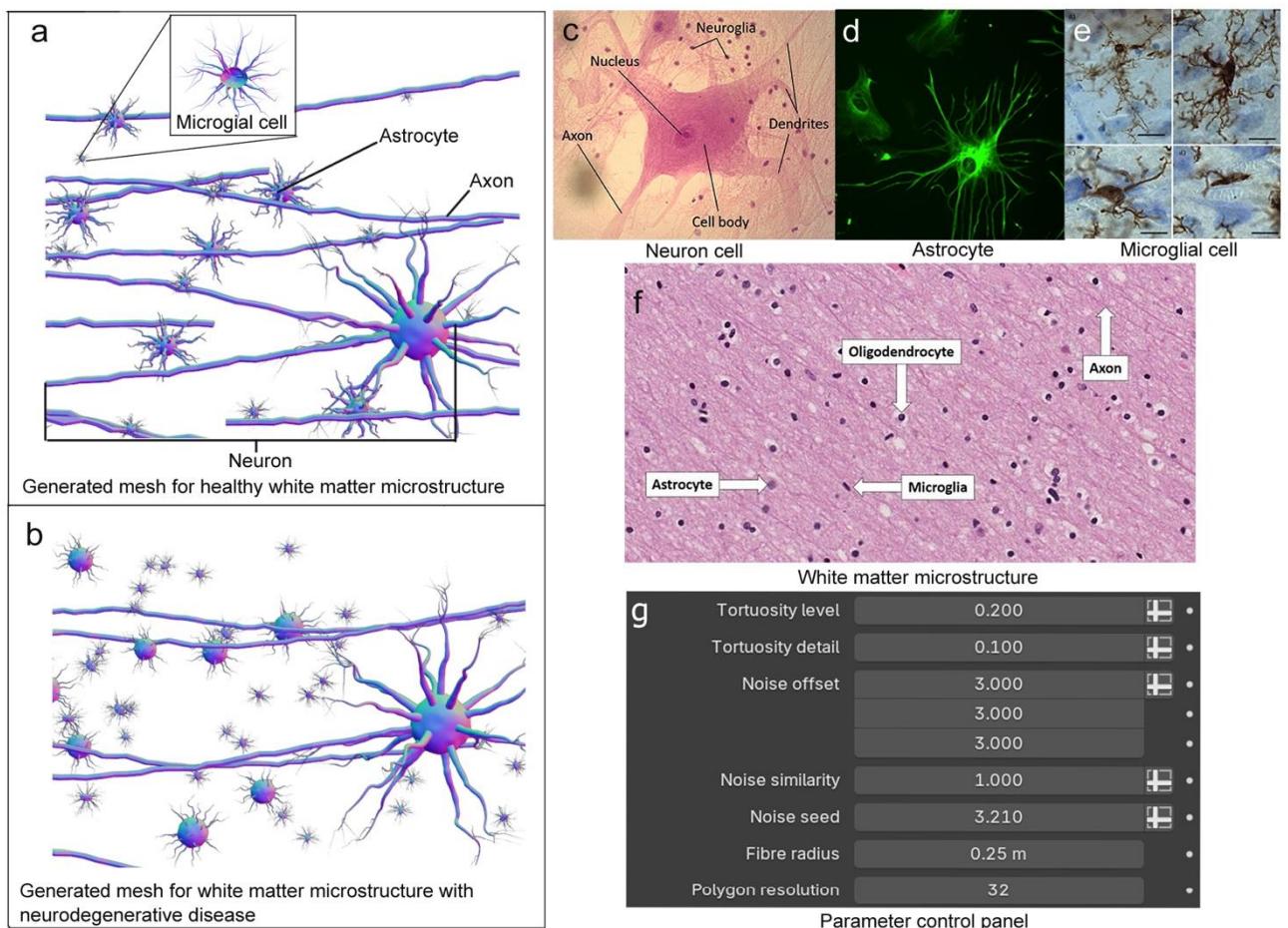


Table 5. Parameter settings for neural mesh generation. Default values and permissible ranges are provided for key geometric parameters of axons, neurons, astrocytes and microglial cells, based on established histological measurements. Angular dispersion values (x,y,z) represent normalised coordinates, while dimensional parameters are given in micrometres (μm).

Mesh Type	Mesh Parameter	Setting Based on Established Research		
		Default	Limit	
Axon ²⁸	Fibre Radius (μm)	0.53	0 - 1	0.17 - 9.00 ¹⁵
	Global Angular Dispersion (Max)	x y z	0.75 0.56 1	
		x y z	0.5 0.52 0	
		x y z	20.65 20.65 20.65	
	Soma Size	100	35.7 - 131.35	
		x y z	10 10 10	
	Dendrite Length	150		0 - 200
Astrocyte (μm)	Soma Size ³¹	x y z	10 10 10	0 - 150
		x y z	3 3 2	
		2	0 - 9	
	Process Length ³²	20	0 - 761.7	
Microglial Cell (μm) ³²	Branching Length	100	0 - 761.7	

3.4 MCMR Results

The following section presents the simulation outputs generated via the MCMR pipeline. As demonstrated previously in [Figure 1](#), we first validate the base wrapper by using both MCMR and Blender-defined microstructures to generate a test signal ([Appendix 6.6](#)). Building upon this foundation, we applied the advanced models implemented in [2.4.2 \(Table 3b\)](#), and benchmarked the results against the analytical predictions provided by Diffusion Models Interactive Viewer (DIVE). These simulations are shown ahead of the neurodegenerative case study.

In [Figure 4a](#), we evaluate the ADM model by comparing exponentially-fitted TS-B-value signals. These were yielded from MCMR simulations and DIVE analytical predictions from parameters in [Figure 1](#). Both demonstrate increasingly pronounced exponential decay with increasing r_{axon} . However, DIVE consistently overestimates attenuation relative to MCMR results especially at high B-values. This overestimation diminishes as axon diameters exceed $5\mu\text{m}$.

Figure 4b illustrates changes in intra-axonal ADC values with diffusion time for both MCMR and DIVE results, logarithmically-fitted for all r_{axon} . As seen, smaller axons exhibit lower ADC values and more rapid attenuation compared to larger axons.

In Figure 4c, results from the HARDI model are presented. Consistent features were observed for DIVE and MCMR models: both displayed oblate, disc-like HARDI profiles with a 'sink' at the centre. Notably, the DIVE HARDI-shell appears less flattened than MCMR results as indicated by the scale.

Additionally, sampled TS were plotted against $\cos^2(\theta)$ for both models, where θ denotes the angle between the axon-extension and diffusion gradient direction. We observe the DIVE model underestimates the exponentially fitted MCMR results for all orientations.

3.5 ND Case Study: Amyotrophic Lateral Sclerosis (ALS)

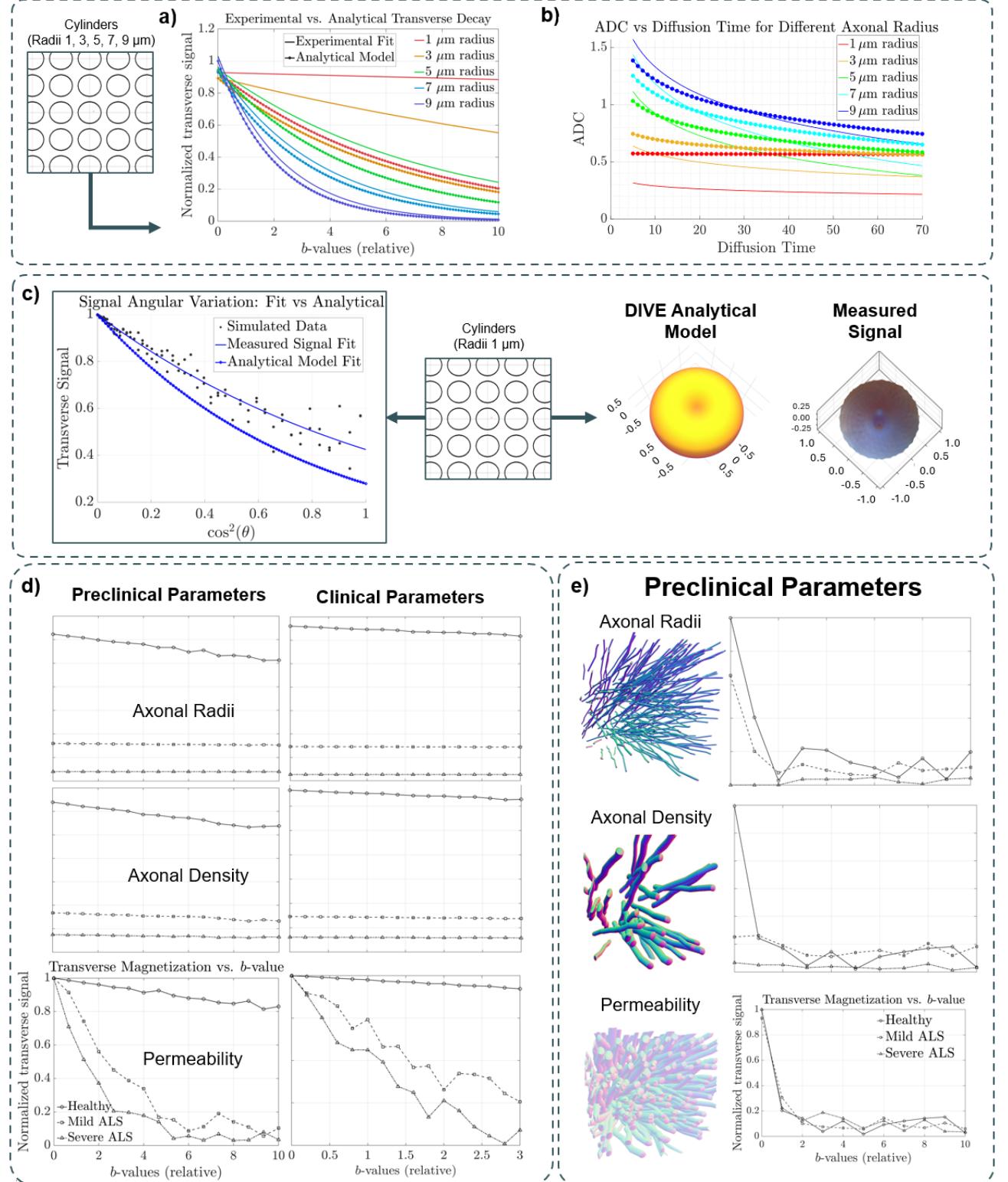
As stated in 2.4.3, for each graph consisting of 3 progressively-diseased statuses (Healthy, Mild-ALS and Severe-ALS), we have compared the S_0 and -D quantities for each tissue and scanner parameter (Appendix 6.7). These are graphically represented as the y-intercept and attenuation, respectively. Out of eight tissue parameters, three preliminarily showed substantial differences across statuses: r_{axon} , ρ_{axon} and permeability and were shown in Figure 4d.

For r_{axon} and ρ_{axon} changes, S_0 lowers as statuses progress, with very little signal attenuation for the healthy state and nearly none for both diseased states. Moreover, this attenuation is more apparent in Preclinical parameters than Clinical.

For permeability, slight attenuation is again observed for the healthy state, with Severe-ALS states exhibiting more decay than Mild-ALS states. We again observe generally more attenuation in all 3 states when comparing Preclinical parameters to Clinical.

Finally, Blender meshes were constructed, reflecting changes of the above three parameters across statuses using Preclinical parameters. These meshes were run using the MCMR pipeline to produce the graphs in Figure 4e. S_0 for all tissue parameters showed identical trends as above. However, across all parameters and all statuses, only 3 showed minimal attenuation: Severe-ALS mesh for r_{axon} and Mild, Severe-ALS meshes for ρ_{axon} . The rest were much more attenuated comparative to the preliminary test round.

Figure 4. All the results yielded from the various Advanced Models offered in the MCMR Pipeline, with the corresponding MCMR or Blender-imported meshes listed. **Subfigure a) to e)** shows the exponentially-fitted results of DIVE analytical model compared against MCMR for: a) the ADM model; b) the ADC-DT model; c) the HARDI model; d) the preliminary round of testing for tissue parameters altered in ALS using MCMR-defined geometries, and e) the secondary round of testing for these parameters using Blender-defined geometries.



3.6 MCMR Wrapper Usability Results

Seven participants responded to the survey. Among them, 71% had some prior coding experience, though none had worked with Julia before.

Table 6. Summary of survey results, with all functionalities ranked for usability.

Success completion rate	86%
Average Clarity Rating (1–10)	7
Function Usability (Ranked)	Defining geometry
	Running simulation
	Outputting test and results
	Defining sequence and simulation

3.7 Agentic AI Coder Results

Table 7. Evaluation of Agentic AI-Coder and ChatGPT (GPT-4o) on custom datasets, Pass@5 means at least one out of the top 5 generated code samples correctly solves the problem.

	Agentic AI Coder	ChatGPT (GPT-4o)
Code Generation Accuracy (Pass@5)	91.0%	4.5%

4. Discussion

We reiterate our aim to develop a mesh-to-signal dwMRI simulation pipeline, investigating signal changes that could reflect underlying microstructures. To ensure accuracy, we crucially rid assumptions made in forward models by first creating a biomimetic mesh-generation tool using Blender, followed by a wrapper function and Advanced Models for MCMR simulations. Finally, we established an agentic AI-coder to enhance usability.

We first evaluated the mesh-generation tool, which demonstrated flexibility through multiple tunable parameters, and found that it accurately captured key features of hWM when compared with histological data. Next, we validated the wrapper function and our Advanced Models by comparing its results with DIVE, concluding partial resemblance of results upon data analysis. Additionally, we compared our agentic AI-coder with GPT-4o, observing significant improvements in resolving troubleshooting. Finally, we presented a case study on ALS to show the pipeline's practical application. A detailed discussion of the above is provided below, followed by a critical evaluation of the shortcomings and potential improvements in our pipeline.

4.1 Blender Mesh Discussion

The Blender generated mesh serves as adjustable base models which can achieve the cellular morphology of hWM.

As a baseline, we model four distinct neural cell types with high resemblance to the corresponding microscopic images, as illustrated in [Figure 3c-f](#). The inclusion of astrocytes and microglial cells, which possess markedly distinct morphology compared to axon fibres, enables investigation into their collective impacts on dwMRI signals. They received limited attention in previous studies.

In addition, the morphological parameter for the model can be customised for individual cells and cell-clusters ([Table 5](#), full list refers to [Appendix 6.5](#)). This parametric pipeline provides the flexibility to simulate diverse hWM architectures, ranging from healthy tissue with uniform and densely packed fibres, to pathological conditions exhibiting elevated axon-fibre dispersion and irregularity ([Figure 3a,b](#)). As for pathological changes seen in other cells (e.g. astrocytes), meshes with swollen cell bodies are generated.²³

Default values for key morphological parameters ([Table 5](#)) are either derived from the most representative morphology.^{15,28–32} This provides guidelines for modelling morphologies according to biological standards. For example, in [Figure 3a](#), the cell bodies of microglia are ellipsoidal. Therefore, the default configuration is elongated along the xy-plane to account for this.

Another important feature of the base model is voxel confinement, which turns meshes into unit voxels, facilitating comparisons between subsequent MCMR simulations ([Figure 1](#)) by eliminating stochastic

volume differences, which is caused by heterogeneity between each cell.

4.2 MCMR Results Discussion

Here, we discuss results produced using the MCMR pipeline and its advanced extensions, seen in Figure 4. All parameters chosen can be referred to Appendix 6.6.

To generate clinically and preclinically interpretable dwMRI signals, the initial intention was to simulate both intracellular and extracellular signals via MCMR. However, project time constraints necessitated focusing our analysis on transverse signals within intracellular compartments only.

Literature demonstrated that diffusion anisotropy in brain tissue is mainly driven by intracellular structures,³³ particularly axons and dendrites rather than extracellular content, and highlighted the impermeability of axonal membranes over typical diffusion timescales.³⁴ These studies justify our emphasis on intracellular signals, which largely allows constructing a forward model rooted in microstructural realism. The measured intracellular signals are normalised by S_0 to eliminate T2-weighted effects within tissue.

Revisiting the ADM modality in Figure 4a, this plot assesses the relationship between r_{axon} and diffusion restriction. As r_{axon} increases, water molecules diffuse more freely along axonal cross-sections, leading to heightened signal attenuation. DIVE and MCMR showed consensus regarding this and high resemblance for $r_{\text{axon}} \geq 5\mu\text{m}$. However, the DIVE signal attenuated more rapidly than its MCMR counterpart for $r_{\text{axon}} < 5\mu\text{m}$, with increasing divergence for higher B-values. This shows that sensitivity to $r_{\text{axon}} < 5\mu\text{m}$ is decreased for MCMR, at which point the analytical model overestimates sensitivity.

These patterns could stem from Gaussian Phase Approximations (GPA) in DIVE, a mathematical simplification that estimates water molecule behaviour when diffusion gradients are applied. While it decreases computational load, it does not account for deflections and interactions of water molecules when approaching boundaries (e.g. axonal membranes), therefore overestimating signal attenuation. As seen, this especially true for small axons, where water molecules frequently interact with the membrane. Meanwhile, MCMR is an MC-based simulator that tracks water molecules every time step, therefore accounting for these interactions.

MCMR therefore depicts restrictive diffusion more accurately, aligning with previous reports (e.g., Veraart et al.³⁵) which indicate that sub-micron axons remain largely undetectable under low B-values (hence G-values) as used here.³⁵ This sensitivity may also be attributed to parameter choice. Preclinical protocols with much stronger G-values (300mT/m) and hence B-values³⁵ or diffusion time-dependent methods could improve sensitivity to small axons. The latter approach is explored below.

Figure 4b shows how increasing diffusion time affects ADC values. Diffusion time here refers to the time window which water molecules diffuse within axons before signal measurement. For smaller

axons, water molecules hit boundaries faster, leading to faster drops in ADC, and vice versa for larger axons.

Comparing ADC values against Mean Diffusivity (MD) obtained from DIVE ([Figure 4b](#)), we observe similar trends overall. However, MD values are consistently higher. This is expected as MD averages radial and axial diffusivities; whereas ADC primarily reflects radial diffusivity only. Axial diffusion more closely resembles free diffusion and is less hindered, hence raising MD. Still, such comparison is inexact, potentially due to the absence of TE and TR specification when making analytical predictions.

Altogether, our results demonstrate two approaches to probe r_{axon} . Through experimental data, we also acknowledge that sensitivity to smaller axons more reflective of those within human brains could be achieved via increasing G-value or shorter diffusion times. Such could potentially be achieved through preclinical acquisition protocols.

[Figure 4c](#) presents HARDI-derived signal shells simulated using $1\mu\text{m}$ circular axons arranged in a perfectly parallel bundle. The shell is generated by a TS-gradient orientation plot. Both DIVE and MCMR-derived shells appear smooth and circular in the xy-plane, reflecting uniform diffusion and hence signal decay in axonal cross-sections and confirming the absence of fibre crossings. The central 'sinks' along both centres indicate minimal signal in directions parallel to the axons, where diffusion is least restricted.

However, the DIVE shell appears smaller overall than the MCMR shell. This was also evident in the TS- $\cos^2(\theta)$ plot where the analytical model consistently underestimated MCMR fitting. (θ denotes the angle between the axon-extension and gradient direction), again suggesting an overestimated signal loss. This supports the hypothesis that GPA-assumed modules may fail to capture diffusion restrictions in small axons.

For the above observations, while GPA explains much of the observed mismatch between DIVE and MCMR data, other factors may contribute. Analytical models like DIVE may assume idealized gradient pulses, meaning diffusion gradients are applied instantaneously and water instantly experiences full gradient strength; while MCMR exposes water molecules to gradually increasing gradient strengths until maximum value, and a gradual decrease afterward. This therefore measures less diffusion compared to the former, hence less signal attenuation. While MCMR bypasses such idealisations, its MC-based nature may still indicate some less significant assumptions, such as total homogeneity of properties (e.g. permeability) within the simulated geometry.

4.3 ALS Case Study Discussion

We first examine the preliminary results and implications of our case study on ALS ([Figure 4d,e](#)). Ranges of values used in experiments are referenced from literature values ([Table 2](#)). As ascertained above, both scanner specifications showed little sensitivity to r_{axon} and ρ_{axon} across statuses due to generally small r_{axon} used, hence restricting diffusion. However, sensitivity is heightened for permeability changes across statuses, due to higher likelihood of water molecules bypassing the axonal membrane, hence leading to less restricted diffusion.

Moreover, S_0 decrements across statuses also fell within expectations, as both lowering r_{axon} without changing ρ_{axon} and vice versa decreases the volume fraction of the mesh, leading to lower probability that a simulated water molecule falls within intracellular compartments. S_0 did not show significant differences across permeability, as volume fractions remained similar throughout.

Finally, we combine Blender-MCMR pipelines for secondary testing ([Figure 4e](#)). Only preclinical parameters were considered due to prior confirmation of their heightened sensitivity against clinical parameters. For all tissue parameters, identical S_0 trends were observed, as explained above. However, much higher sensitivity and attenuation (-D) was generally observed across all tissue parameters and statuses. Considering the $0.25\sim1\mu m$ r_{axon} range used, such results attenuated more than expected. We attribute this to the following:

Table 8. Evaluation and assessment of possible factors that may have individually or collectively contributed to the deviation from expectations for Blender meshes.

Pipeline	Error	Description
Blender	Low mesh watertightness	Results in more water molecules experiencing unrestricted diffusion as they leave the intracellular compartment.
	Fibre crossing overlaps	Mesh overlaps may contribute to complicated simulation calculations and classification between intracellular/extracellular compartments.
MCMR	Post-import mesh processing	After Blender importation, MCMR may individually manipulate mesh configurations, leading to changes in various simulation properties.
	Dimension mismatch	Blender and MCMR length units are not homogeneous, hence leading to conversion errors.
	Ambiguous voxel definition	A confinement box is first defined in Blender, and a 1mm by 1mm-by-1mm cube voxel is additionally defined in MCMR. This repetition may cause errors.
	Ambiguous face normals	The triangular faces on meshes can point inward (intracellularly) or outward (extracellularly), which may not have been clearly defined in MCMR during importation.

4.4 MCMR Wrapper and Add-on Usability Discussion

The overall completion rate and average clarity rating indicated most users successfully ran the wrapper functions, despite limited coding proficiency ([Table 6](#)). Among evaluated tasks, “Defining Geometry” was rated as the easiest, followed by “Running Simulation” and “Outputting Results”. “Defining Sequence and Simulation” was seen as most challenging, suggesting needs for additional instructions.

These observations suggest while the wrapper functions are generally accessible, providing more guidance for sequence configuration would further improve facility.

4.5 Agentic AI Coder Discussion

Our Agentic AI-Coder significantly improves domain-specific code generation accuracy compared to ChatGPT ([Table 7](#)). A key limitation of ChatGPT is hallucination,³⁶ producing incorrect code due to training gaps, especially with obscure packages requiring specialized syntax. Another challenge is limited context window,³⁷ where ChatGPT struggles to retain relevant information in long prompts. Agentic coding mitigates these issues by managing context more effectively and enhancing model self-correction through code execution and autonomous error revision.

4.6 Conclusion and Future Work

To conclude, our pipeline advances forward modelling by removing common idealizations in analytical methods: oversimplified geometries, GPA, and idealized gradient waveforms. We achieve this by combining highly biomimetic brain meshes from our Blender pipeline with a streamlined MCMR workflow with advanced models, supported by AI-coder tools and user-friendly tutorials. Through a case study on ALS, we demonstrate how microstructural neurodegeneration can be visualized using this unified pipeline.

Nonetheless, there are limitations we aim to address. Currently, our Blender toolkit supports only select hWM neuron types. Also, complex, dense axonal geometries may introduce fibre overlap, interfering MCMR simulations. Moreover, despite adequate existing research on cell morphology, more breadth is desired for single-cell models. To improve this, we plan to cross-validate against broader literature and incorporate additional cell types, such as oligodendrocytes.

Our MCMR wrapper simplifies signal generation into a single step for custom geometries and dwMRI sequences. However, it remains a code-based tool with limited interactivity. We propose to develop a graphical interface, enhancing usability and accessibility.

Due to time constraints, we focused on axon-rich structures for computational efficiency. This excluded extracellular compartments and glial/neuronal soma, which may influence dwMRI signals.³⁸ Future work will include multi-cell-type simulations in literature-accurate ratios, alongside performance-optimized MCMR workflows.

Our advanced models and neurodegenerative case studies involved axon-derived geometries, selected for computational efficiency. While many hWM structures (e.g. Corpus Callosum) are axon-rich, trace glial cells and neuronal somas could contribute to signal changes.³⁸ Additionally, extracellular compartments were not accounted for. Future work will include simulations of Blender multi-cell type meshes in literature-accurate ratios, alongside optimized MCMR workflows.

Our axon-oriented advanced models may be refined to better capture microstructural changes. For ADC-DT, while shorter diffusion times increase sensitivity to small axons ([Figure 4b](#)), standard sequences impose diffusion time limits. Specialized sequences, such as Oscillating Gradient spin-echo,

can be implemented to overcome this.³⁹ Additionally, our HARDI implementation has limited resolution for low-angle fibre crossings. Therefore, integrating spherical deconvolution better estimates the fibre orientations,⁴⁰ enhancing angular resolution.

5. Reference

1. Caroli A. Diffusion-Weighted Magnetic Resonance Imaging: Clinical Potential and Applications. *Journal of Clinical Medicine*. 2022;11(12):3339. doi:<https://doi.org/10.3390/jcm11123339>
2. Baliyan V, Das CJ, Sharma R, Gupta AK. Diffusion weighted imaging: Technique and applications. *World Journal of Radiology*. 2016;8(9):785. doi:<https://doi.org/10.4329/wjr.v8.i9.785>
3. Shatil AS, Matsuda KM, Figley CR. A Method for Whole Brain Ex Vivo Magnetic Resonance Imaging with Minimal Susceptibility Artifacts. *Frontiers in Neurology*. 2016;7. doi:<https://doi.org/10.3389/fneur.2016.00208>
4. Hall MG, Alexander DM. Convergence and Parameter Choice for Monte-Carlo Simulations of Diffusion MRI. *IEEE transactions on medical imaging*. 2009;28(9):1354-1364. doi:<https://doi.org/10.1109/tmi.2009.2015756>
5. Palombo M, Alexander DC, Zhang H. A generative model of realistic brain cells with application to numerical simulation of the diffusion-weighted MR signal. *NeuroImage*. 2019;188:391-402. doi:<https://doi.org/10.1016/j.neuroimage.2018.12.025>
6. NeuroMorpho.org - a centrally curated inventory of digitally reconstructed neurons and glia. neuromorpho.org. <https://neuromorpho.org/>
7. Banovac I, Sedmak D, Rojnić Kuzman M, Hladnik A, Petanjek Z. Axon morphology of rapid Golgi-stained pyramidal neurons in the prefrontal cortex in schizophrenia. *Croatian Medical Journal*. 2020;61(4):354-365. doi:<https://doi.org/10.3325/cmj.2020.61.354>
8. Coelho S, Pozo JM, Costantini M, et al. Local volume fraction distributions of axons, astrocytes, and myelin in deep subcortical white matter. *NeuroImage*. 2018;179:275-287. doi:<https://doi.org/10.1016/j.neuroimage.2018.06.040>
9. Callaghan R, Alexander DC, Palombo M, Zhang H. ConFiG: Contextual Fibre Growth to generate realistic axonal packing for diffusion MRI simulation. *NeuroImage*. 2020;220:117107. doi:<https://doi.org/10.1016/j.neuroimage.2020.117107>
10. Kévin Ginsburger, Matuschke F, Fabrice Poupon, Mangin JF, Aixer M, Poupon C. MEDUSA: A GPU-based tool to create realistic phantoms of the brain microstructure using tiny spheres. *NeuroImage*. 2019;193:10-24. doi:<https://doi.org/10.1016/j.neuroimage.2019.02.055>
11. Abdellah M, José J, Foni A, Guerrero R, Boci E, Schürmann F. *Meshing of Spiny Neuronal Morphologies Using Union Operators*. The Eurographics Association; 2022:19-26.
12. Birgiolas J. *Towards Brains in the Cloud: A Biophysically Realistic Computational Model of Olfactory Bulb*. 2019.

13. Selin E. How modifiers work in Blender, an overview - Artisticrender.com. Artisticrender.com. Published February 6, 2020. Accessed April 12, 2025. <https://artisticrender.com/how-modifiers-work-in-blender-an-overview/>
14. Baily J. What Are Geometry Nodes In Blender? - Blender Base Camp. Blender Base Camp. Published March 9, 2022. Accessed April 12, 2025. <https://www.blenderbasecamp.com/what-are-geometry-nodes-in-blender>
15. Liewald D, Miller R, Logothetis N, Wagner HJ, Schüz A. Distribution of axon diameters in cortical white matter: an electron-microscopic study on three human brains and a macaque. *Biological Cybernetics*. 2014;108(5):541-557. doi:<https://doi.org/10.1007/s00422-014-0626-2>
16. Amor S, Nutma E, Marzin M, Puentes F. Imaging immunological processes from blood to brain in amyotrophic lateral sclerosis. *Clinical and Experimental Immunology*. 2021;206(3):301-313. doi:<https://doi.org/10.1111/cei.13660>
17. Fujimori K, Ishikawa M, Otomo A, et al. Modeling sporadic ALS in iPSC-derived motor neurons identifies a potential therapeutic agent. *Nature Medicine*. 2018;24(10):1579-1589. doi:<https://doi.org/10.1038/s41591-018-0140-5>
18. Müller HP, Kassubek J. MRI-Based Mapping of Cerebral Propagation in Amyotrophic Lateral Sclerosis. *Frontiers in Neuroscience*. 2018;12. doi:<https://doi.org/10.3389/fnins.2018.00655>
19. Acosta-Cabronero J, Machts J, Schreiber S, et al. Quantitative Susceptibility MRI to Detect Brain Iron in Amyotrophic Lateral Sclerosis. *Radiology*. 2018;289(1):195-203. doi:<https://doi.org/10.1148/radiol.2018180112>
20. Jin T, Zhao F, Kim S. Sources of functional apparent diffusion coefficient changes investigated by diffusion-weighted spin-echo fMRI. *Magnetic Resonance in Medicine*. 2006;56(6):1283-1292. doi:<https://doi.org/10.1002/mrm.21074>
21. Huang D, Zhang J, Luck M, Bu Q, Qing Y, Cui H. *AgentCoder: Multi-Agent Code Generation with Effective Testing and Self-Optimisation*. <https://arxiv.org/pdf/2312.13010.pdf>
22. RooVetGit. GitHub - RooVetGit/Roo-Code: Roo Code (prev. Roo Cline) gives you a whole dev team of AI agents in your code editor. GitHub. Published April 12, 2025. Accessed April 12, 2025. <https://github.com/RooVetGit/Roo-Code>
23. Schirmer L, Schafer DP, Bartels T, Rowitch DH, Calabresi PA. Diversity and function of glial cell types in multiple sclerosis. *Trends in Immunology*. 2021;42:228-247. doi:<https://doi.org/10.1016/j.it.2021.01.005>
24. Nervous Tissue. SCIENTIST CINDY. Published 2019. <https://www.scientistcindy.com/nervous-tissue.html>

25. Wikipedia Contributors. Astrocyte. Wikipedia. Published April 3, 2025. https://en.wikipedia.org/wiki/Astrocyte#/media/File:Human_astrocyte.png.
26. Torres-Platas SG, Comeau S, Rachalski A, et al. Morphometric characterization of microglial phenotypes in human cerebral cortex. *Journal of Neuroinflammation*. 2014;11(1). doi:<https://doi.org/10.1186/1742-2094-11-12>
27. Cerebrum - Normal Histology. Nus.edu.sg. Published 2025. Accessed April 14, 2025. <https://medicine.nus.edu.sg/pathweb/normal-histology/cerebrum>
28. Moss HG, Chen AA, Jensen JH, Benitez A. Intra-voxel angular dispersion of fibers in corpus callosum decreases with healthy aging. *Imaging Neuroscience*. 2025;3. doi:https://doi.org/10.1162/imag_a_00463
29. Milošević NT, Ristanović D, Marić DL, Rajković K. Morphology and cell classification of large neurons in the adult human dentate nucleus: A quantitative study. *Neuroscience Letters*. 2010;468(1):59-63. doi:<https://doi.org/10.1016/j.neulet.2009.10.063>
30. Meyer G, Wahle P, A. Castaneyra-Perdomo, R. Ferres-Torres. Morphology of neurons in the white matter of the adult human neocortex. *Experimental Brain Research*. 1992;88(1):204-212. doi:<https://doi.org/10.1007/bf02259143>
31. Oberheim NA, Takano T, Han X, et al. Uniquely Hominid Features of Adult Human Astrocytes. *Journal of Neuroscience*. 2009;29(10):3276-3287. doi:<https://doi.org/10.1523/jneurosci.4707-08.2009>
32. Placone AL, McGuigan PM, Bergles DE, Guerrero-Cazares H, Quiñones-Hinojosa A, Searson PC. Human astrocytes develop physiological morphology and remain quiescent in a novel 3D matrix. *Biomaterials*. 2015;42:134-143. doi:<https://doi.org/10.1016/j.biomaterials.2014.11.046>
33. Reveley C, Ye FQ, Mars RB, Matrov D, Chudasama Y, Leopold DA. Diffusion MRI anisotropy in the cerebral cortex is determined by unmyelinated tissue features. *Nature Communications*. 2022;13(1). doi:<https://doi.org/10.1038/s41467-022-34328-z>
34. Yablonskiy DA, Sukstanskii AL. Theoretical models of the diffusion weighted MR signal. *NMR in Biomedicine*. 2010;23(7):661-681. doi:<https://doi.org/10.1002/nbm.1520>
35. Veraart J, Nunes D, Rudrapatna U, et al. Noninvasive quantification of axon radii using diffusion MRI. *eLife*. 2020;9. doi:<https://doi.org/10.7554/elife.49855>
36. Huang L, Yu W, Ma W, et al. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. <https://arxiv.org/pdf/2311.05232.pdf>
37. Chen S, Wong SH, Chen L, Tian Y. Extending Context Window of Large Language Models via Positional Interpolation. *arXiv (Cornell University)*. Published online June 27, 2023. doi:<https://doi.org/10.48550/arxiv.2306.15595>

38. Reyes-Haro D, Mora-Loyola E, Soria-Ortiz B, García-Colunga J. Regional density of glial cells in the rat corpus callosum. *Biological Research*. 2013;46(1):27-32. doi:<https://doi.org/10.4067/s0716-97602013000100004>
39. Xu J. Probing neural tissues at small scales: Recent progress of oscillating gradient spin echo (OGSE) neuroimaging in humans. *Journal of Neuroscience Methods*. 2020;349:109024. doi:<https://doi.org/10.1016/j.jneumeth.2020.109024>
40. Tournier JD, Calamante F, Gadian DG, Connelly A. Direct estimation of the fibre orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage*. 2004;23(3):1176-1185. doi:<https://doi.org/10.1016/j.neuroimage.2004.07.037>

6. Appendix

6.1 Project Planning

Overall, our team successfully adhered to the overall project timeline. Background research was completed within the first term, establishing a solid foundation for developing the Blender mesh, wrapper functions, and the AI coder. However, we underestimated the time required to integrate Blender-generated meshes with the MCMR simulator, as simulations involving complex geometries often took hours or even several days to run. To address this challenge, we simplified the meshes based on ensured balancing accuracy and realism. With this adjustment, simulation times were reduced to just a few hours.

Despite these adjustments, prolonged simulation times still slowed feedback on mesh improvements and delayed subsequent project stages, resulting in a disproportionate amount of time spent verifying results rather than advancing towards our primary goals. We additionally acknowledge that such time could instead have been allocated towards further optimization within the mesh-generation process or the MCMR wrapper code, shortening the required computational time and load for each simulation.

6.2 Lessons Learned

6.2.1 Communication

Slack served effectively as our primary platform for communicating progress, sharing resources, and discussing results. Messages typically received prompt responses, and all team members actively participated in the discussions. Nevertheless, there is room for improvement.

One key takeaway from this project was the importance of regular, consistent progress updates. While focusing on individual tasks is vital, maintaining an updated, holistic view of the project ensures cohesive teamwork and clear recognition of each member's achievements and contributions. Such comprehensive awareness also aids in producing written reports and documentation.

To enhance productivity, team members should proactively track updates, clearly communicate their ongoing tasks, and regularly inform the group of their progress. Equally important is that everyone maintains a clear understanding of the overarching project goals and status, even in aspects beyond their direct responsibilities.

6.2.2 Task Allocation

The second lesson we learnt was the need to manage task allocation. At the start of the project, we formed subgroups and assigned each member to a specific area. However, the time and effort involved for certain tasks were initially underestimated, leading to overloading of some members. While our team successfully completed assigned tasks at the end, more thorough planning and regular reviews of workload distribution would further enhance productivity and balance responsibilities among team

members.

This issue can be improved via several practical measures. First, allocating responsibilities within each subgroup and clearly defining tasks ensures that everyone knows their objectives from the outset. Second, establishing and adhering to a weekly progress-tracking routine helps identify potential overloads early and maintains consistent productivity. This approach ensures that each person has a well-defined task, preventing uneven workloads and supporting a more efficient overall project outcome.

6.2.3 Group Research Completeness

Extensive research has been done throughout the preparatory phase of this project, regarding the investigation of parameters within various glial cells and neurons for Blender modelling, and the documentation and operation of various modules and functions within MCMR during the development of the wrapper. While a vast majority of such were deemed beneficial in allowing us to establish this full pipeline, a more robust foundation ought to have been established before beginning the development of either pipeline, such that this researched knowledge could be utilized further for the benefit of the project. Corresponding to Communication improvements in 5.2.1, better communication of research findings within group meetings may have aided in alleviating this gap, allowing a more effective workflow.

6.3 Group Dynamics & RACI Matrix

In the tale below, a breakdown of the respective roles and contributions of each member is denoted with one of the following colours: Red (Responsible); Yellow (Accountable); Green (Consulted) or Blue (Informed).

Throughout the approximately 20-week term time from the start of this project until present, weekly meetings were held in-person whenever possible at the South Kensington Campus. Internal meetings were held within the group on Mondays, and supervised meetings were held under the guidance of Dr. Amy Howard on Wednesdays at the White City campus.

Minutes were taken and updated on a weekly basis per meeting on OneNote in .pdf format, and a weekly to-do list was established and updated at the start of each academic week. A Gantt Chart, used in gauging group work progress, is updated monthly. All the above documents and relevant updates are done within Slack.

Embarking on a project involving the use of state-of-the-art, newly developed software packages involving dwMRI simulations of meshes allowed the team to fully understand the significance of proper compartmentalisation of research, given that at the time of compiling this report, very few examples were published demonstrating the usage of a few functions essential for simulating dwMRI acquisition protocols. Throughout the duration of this project, the team was further subdivided into subgroups with reference to the RACI Matrix, which can be found below:

RACI Matrix:

	Ho Pak Hei	Bob Qian	Catherine Ni	Jane Ren	Jiayi Bai	Victor Kam	Kian Daneshi
Autumn Term							
Group Allocation							
Background Research							
Meshing Software Viability Tests							
MRI-Simulation Viability Tests							
Group Project Mind Map							
Spring Term							
Blender Modelling Research							
Blender Mesh Function Development							
MCMR Wrapper Creation							
MCMR Advanced Tool Development							
MCMR Signal-Mesh Variations							
AI-Coder Development							
Brain Degenerative and Microstructural							
Group Pitching							
Final Report (Intro)							
Final Report (Methods & Results)							
Final Report Discussion							
Responsible	Team members should execute and complete the assigned task.						
Accountable	Team members should oversee the assigned task, delegating and reviewing the work.						
Consulted	Team members whose opinions are sought, providing input and feedback actively towards the work.						
Informed	Team members who are kept up to date on project progress.						

6.4 Example of “Geometry Nodes” Modifier in Blender

Figure of “Geometry Nodes” workflow built for fibre bundle assembly. The adjustable parameters are listed in the “Group Input” node.

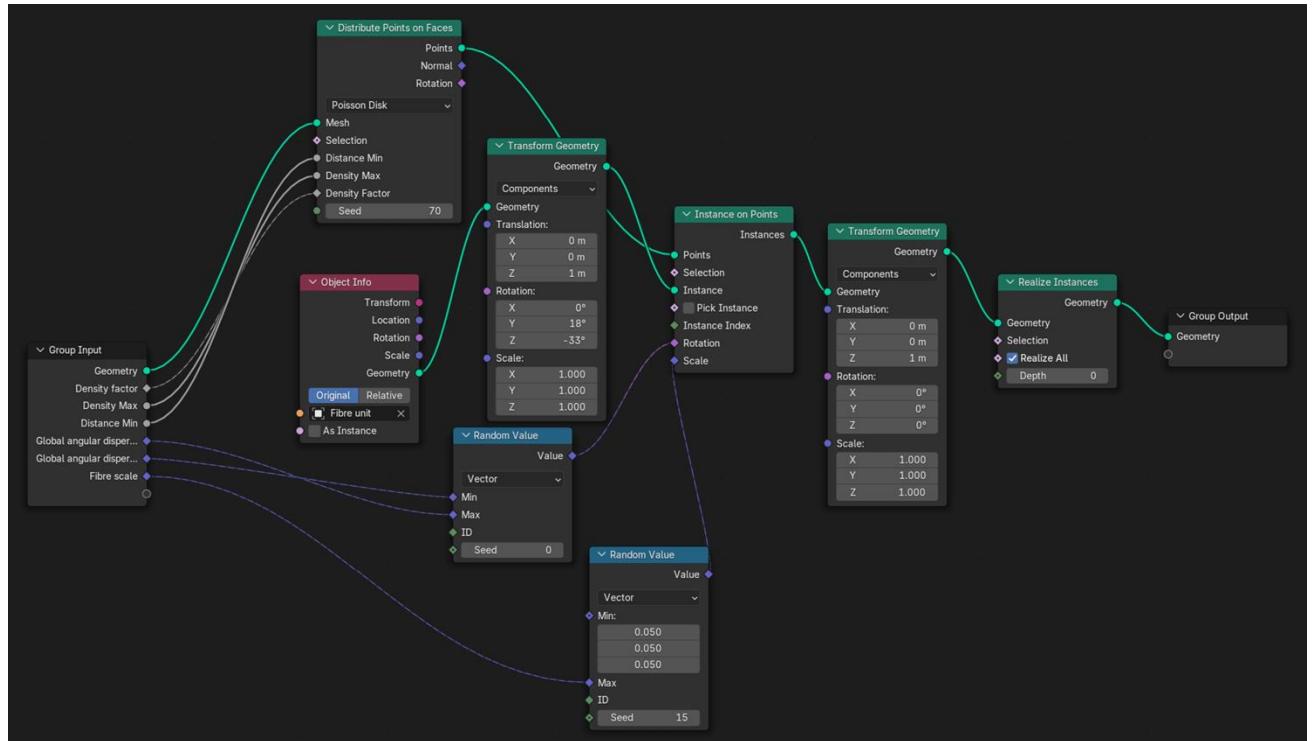
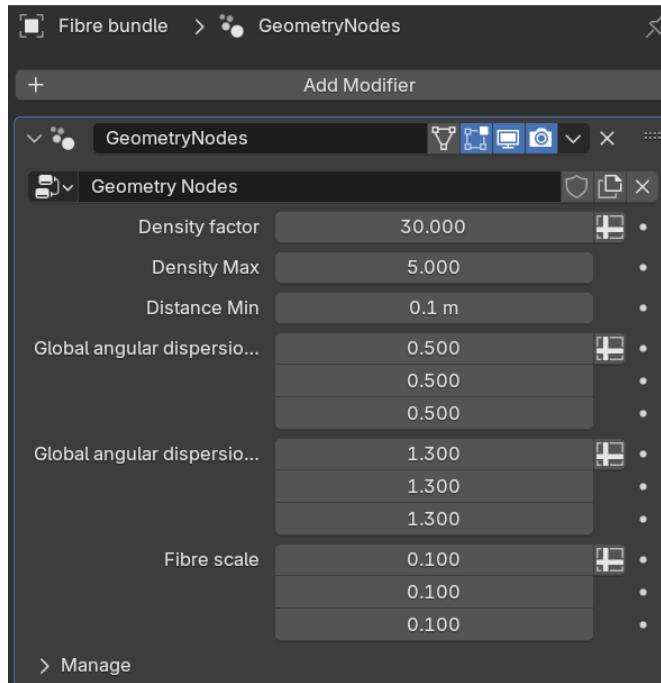


Figure of “Property” panel resulted from the above fibre bundle workflow:



6.5 Full Parameter Table

Blender mesh parameter table

Orange blocks are parameters for 2D fibre configuration.

Mesh Name	Item Name	Parameter	Axis	Range	Default Value	Function
Axon	Fibre unit	Fibre density		1-120	100	Number of fibres generated
		Maximum tortuosity level		0-15	4	Maximum bending frequency
		Switch		On / Off	Off	Selection between Subdivide Mesh mode and Subdivision Surface mode to increase resolution.
		Fibre dispersion	X	0-100	0	Fibre dispersion in 2D
			Y	0-100	0	
			Z	0-100	50	
		Tortuosity level		0-3	0.5	Spatial frequency of bends without changing the number of faces
		Tortuosity detail		0-1	1	Smoothness of bends
		Noise offset	X	0-5	1.5	Bending intensity
			Y	0-5	1.5	
			Z	0-5	1.5	
		Noise similarity		0-50	1	Bending pattern similarity among 2D dispersed fibres
		Noise seed		0-100	0	Noise pattern
		Fibre radius		0.17 - 9	0.53	Basic radius of individual axon fibre (recommended range: 0.17-4.58)
		Polygon resolution		3-32	32	Number of sides of the cross-section of the fibre
		Unit fibre translation	X	-inf - inf	-40	Displacement of unit fibre in 3D space
			Y	-inf - inf	0	
			Z	-inf - inf	0	
		Unit fibre rotation	X	-inf - inf	0	Rotation of unit fibre in 3D space
			Y	-inf - inf	20	
			Z	-inf - inf	0	
		Unit fibre scale	X	-inf - inf	1.5	Dimensions of unit fibre

Mesh Name	Item Name	Parameter	Axis	Range	Default Value	Function
Neuron	<u>Neuron</u>	Dendrite Number		0 - 20	10	Number of dendrites generated
		Dendrite Length		35.7 - 131.35	100	Dendrite length generated
		Dendrite noise level		0-1000	15	Bending frequency of dendrites
		Dendrite root radius		0 - 21.56	15	Radius of cross section of the dendrite root
		Dendrite tip radius		0 - 21.56	5	Radius of cross section of the dendrite tip
		Dendrite start location	X	0 - 21.56	0	Position of dendrite generation point (dendrite root)
			Y	0 - 21.56	0	
			Z	0 - 21.56	0	
		Branching start location		0 - 131.35	50	Branching point of dendrites
		Branching number		0 - 1	0.01	Number of branchings from dendrite generated
		Branching length		0-131.35	60	Length of dendrite branchings
		Branching noise level		0 - 10000	1000	Bending frequency of dendrite branchings
		Soma size	X	17.93 - 21.56	20.65	Dimensions of soma
			Y	17.93 - 21.56	20.65	
			Z	17.93 - 21.56	20.65	
		Soma noise level		0 - 10000	1000	Random deformation of soma surface
		Dendrite noise intensity	X	0 - 10000	15	Maximum intensity of dendrite bending
			Y	0 - 10000	15	
			Z	0 - 10000	15	
		Soma noise intensity	X	0 - 10000	10	Maximum intensity of soma surface deformation

Mesh Name	Item Name	Parameter	Axis	Range	Default Value	Function
Microglial_cell	Microglia	Process Number		0 - 9	2	Number of processes generated
		Process Length		0 - 761.7	20	Process length generated
		Process noise level		-1000 - 1000	2	Bending frequency of process
		Process root radius		0 - 5	1.5	Radius of cross section of the process root
		Process tip radius		0 - 5	1	Radius of cross section of the process tip
		Process start location	X	0 - 5.97	0	Position of process generation point (process root)
			Y	0 - 5.97	0	
			Z	0 - 5.97	0	
		Branching start location		0 - 761.7	10	Branching point of process
		Branching number		0 - 1	0.1	Number of branchings from process generated
		Branching length		0 - 761.7	100	Length of process branchings
		Branching noise level		-10000 - 10000	100	Bending frequency of process branchings
		Soma size	X	0 - 5.97	3	Dimensions of soma
			Y	0 - 5.97	3	
			Z	0 - 5.97	2	
		Soma noise level		0 - 1000	10	Random deformation of soma surface
		Process noise intensity		-10000 - 10000	1	Maximum intensity of process bending
		Soma noise intensity		-10000 - 10000	2	Maximum intensity of soma surface deformation

6.6 Appendix & Test Signals for MCMR Wrapper

Below shows a figure, containing a table with all relevant MCMR parameters found within all steps of the pipeline, as described in section 2.4; and the test signal generated by each an MCMR-defined mesh and a Blender-defined mesh, with the corresponding meshes shown. See section 3.4.

Geo_mode

Parameter	Description
Geo_mode	Integer. Selects geometry creation method: - 1: voxel of spheres - 2: collision-free cylinders - 3: load .ply file

Geo_params (Mode 1)

Parameter	Description
Size	Float64. Voxel size.
td	Float64. Target density (0–1).
mean	Float64. Gamma distribution mean for sphere packing.
var	Float64. Gamma distribution variance.
max_iter	Int64. Max iterations for geometry generation.
min_r	Float64. Minimum sphere radius.
max_r	Float64. Maximum sphere radius.

Geo_params (Mode 2)

Parameter	Description
N	Int64. Number of cylinders.
max_attempts	Int64. Max geometry generation attempts.
z_step	Int64. Steps along z-axis for cylinders.
z_gap	Float64. Distance between z-steps.
tortuosity	Float64. Cylinder 'bendiness'.
spline	Integer. Number of bends.
min_r	Float64. Minimum cylinder radius.
max_r	Float64. Maximum cylinder radius.
xy_bound	Float64. xy-plane boundaries.

Geo_params (Mode 3)

Parameter	Description
geo_params	String. Path to .ply geometry file.

Seq_mode

Parameter	Description
Seq_mode	Integer. Sequence type: - 1: diffusion-weighted - 2: spin echo

Seq_params (Mode 1)

Parameter	Description
b_step	Float64. b-value step size.
k	Integer. Number of b-value steps.
t_echo	Integer. Echo time (ms).
t_rep	Integer. Repetition time (ms).
grad	Vector{Vector{Float64}}. Gradient directions (e.g. [[1.0, 0.0, 0.0]])

Seq_params (Mode 2)

Parameter	Description
b_step	Float64. b-value step size.
k	Integer. Number of b-value steps.
delay	Integer. Delay between spin echo pulses (ms).
t_echo	Integer. Echo time (ms).
t_rep	Integer. Repetition time (ms).

Sim_params

Parameter	Description
T1	Int64. T1 relaxation time (ms).
T2	Int64. T2 relaxation time (ms).
diffusivity	Float64. Water diffusion coefficient.
end_time	Float64. Simulation duration (ms, ≤ TR).
n_spins	Int64. Number of spins (protons).

Out_params

Parameter	Description
skip_TR	Int64. Number of skipped TR cycles.
nTR	Int64. Number of simulated TR cycles.
sim_b	Int64. Index of b-value to simulate.
sim_g	Int64. Index of gradient to simulate.
nTR_i	Int64. Index of TR cycle to extract.
l_t	1 or 2. 1 = transverse, 2 = longitudinal.

Outputs

Parameter	Description
geometry	Defined geometry.
simulation	Simulated object with defined sequence.
signals	Signal vector over time (0.1 ms resolution).
signal_type	'transverse' or 'longitudinal'.
sim_time	Vector of simulation times (ms).
bvals	Vector of b-values used in simulation.
avg_signals	Array of signals across b, grad, time, TR.

Direction Function

Parameter	Description
grad	Vector{Vector{Float64}}. Gradient directions.
Output	3D arrow plot of gradients.

GradGen Function

Parameter	Description
m	Int64. # gradients in yz-plane.
n	Int64. # gradients in xz-plane.
Output	Nested gradient direction array.

HARDI_signals Function

Parameter	Description
b_index	Int64. b-value index.
time_index	Int64. Time index (e.g., t=80ms → 801).
nTR_i	Int64. TR cycle index.
grad	Gradient directions.
avg_signals	3D signal array.
Output	(signals_extract, signals_plot)

HARDI_plot Function

Parameter	Description
signals_plot	Vector of HARDI signal directions.
Output	3D plot of HARDI magnitudes.

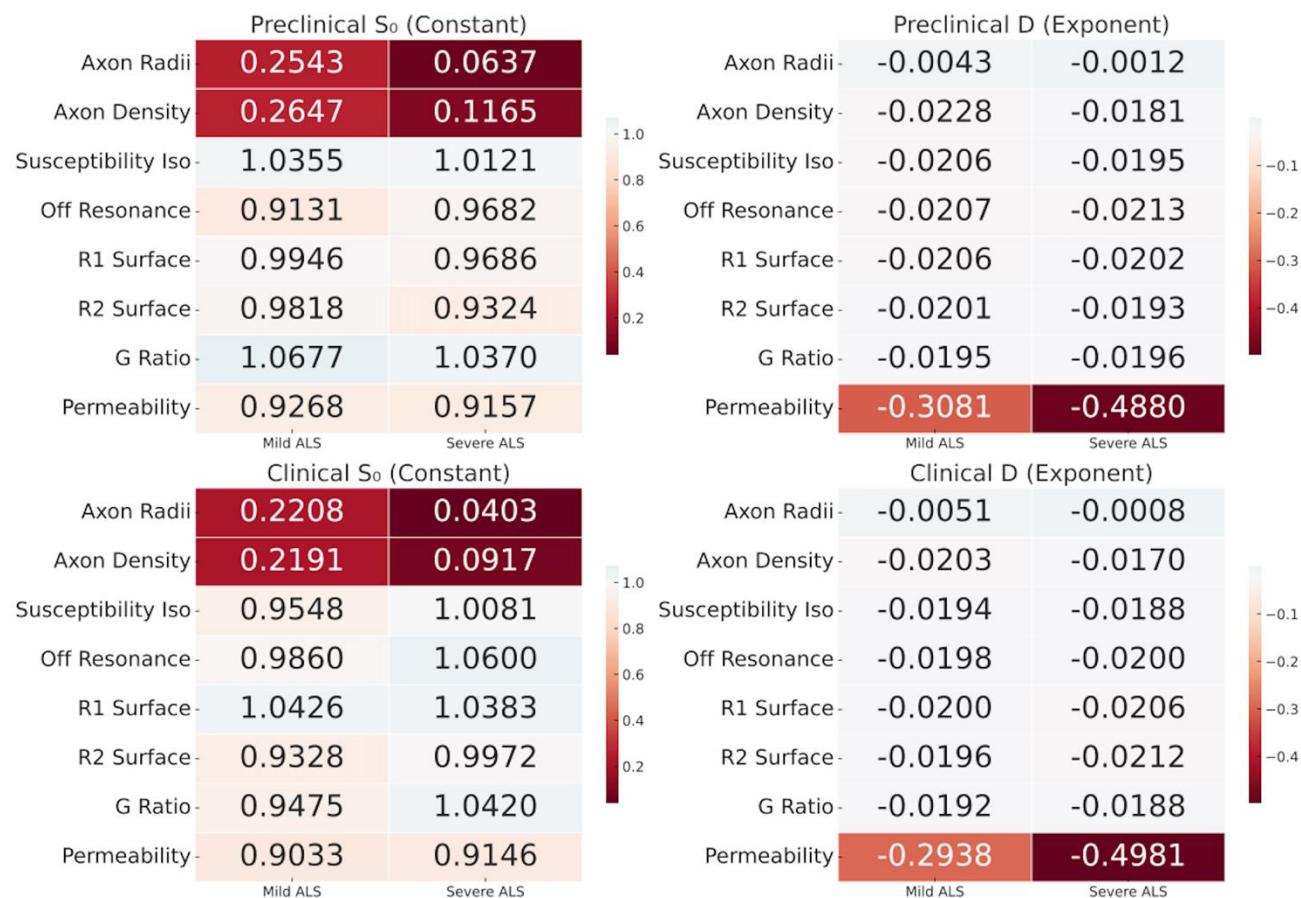
6.7 ALS Exponential Fitting Matrix

Below shows the figure, containing data for both the Normalized Intracellular Signal Strength (S_0) and the negative ADC (-D) for all 48 parameters tested in the preliminary round of testing. This includes 8 tissue parameters, across 3 states (Healthy, Mild-ALS and Severe-ALS). All of the above were tested on parameters which resemble that of a typical Clinical and Preclinical scanner.

Control S_0 : 1.0000

Control D: -0.0200

ALS Parameter Comparison Heatmap



6.8 MCMR Wrapper Usability Questionnaire and Tutorial

Following are the questions asked in the usability survey:

1. Do you have any programming experience?
 - a. Yes
 - b. No
 - c. Somewhat
2. Prior to this survey, do you have any Julia experience?
 - a. Yes
 - b. No
3. Were you successful at getting a simulated signal after the tutorial?
 - a. Yes
 - b. No
4. Rank the part of the tutorial from most helpful to most confusing
 - a. Defining geometry
 - b. Defining sequence and simulation
 - c. Running simulation
 - d. Outputting results
5. Overall, how understandable was the tutorial? *1 for very confusing; 10 for very easy to understand.
6. What other parts about using MCMR tools would you like to see included in the tutorial? *This question prompts for written feedback which is optional.

The tutorial used in the survey consists of essential component of our simulation pipeline accompanied by detailed user guidance. These can be found at our GitHub page: <https://github.com/GradientExplode/MCMR-Code/blob/main/MCMR%20tutorial%20test.ipynb>

6.9 Code Availability

The Blender mesh files, documentation and full parameter table, including default values, adjustable range and function can be found via the GitHub page below:

[GradientExplode/Blender-mesh-toolbox](#)

The Full MCMR Simulation Pipeline and all corresponding Advanced Models, including the ALS neurodegenerative case study, can be found via the GitHub page below:

<https://github.com/Jqxk/MCMR-Wrapper/tree/main>

The source code of Agentic AI-Coder can be found via the GitHub page below:

[GradientExplode/Agentic-AI-Coder](#)