



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE



Fisher-Kolmogorov equation for neurodegenerative diseases

NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

Marco Allanda, 10728587

Edoardo Carrà, 11015152

Daniele Ferrario, 10709932

Lorenzo Gentile, 11001962

Contents

1	Introduction	3
2	The mathematical model	3
2.1	The Fisher-Kolmogorov model	3
2.2	Weak formulation	4
3	Numerical Methods	5
3.1	Semi-discrete formulation	5
3.2	Fully-discrete problem	5
3.2.1	Adaptive Forward Euler	5
3.3	Newton's method	6
3.4	Solving the linear system	7
4	Application to Brain Modelling	8
4.1	Mesh Generation	8
4.1.1	3D Volumetric Mesh	8
4.1.2	2D Sagittal Mesh	8
4.2	Vector Field Implementation	9
4.3	Initial Conditions	10
4.4	White/Gray Matter Partitioning	10
5	Results	11
5.1	Some numerical observations	11
5.1.1	Convergence of the Newton's method	11
5.1.2	Stability observations	11
5.2	Sensitivity Analysis	12
5.2.1	Sensitivity analysis – coefficients	12
5.2.2	Sensitivity analysis – fiber orientation and seeding regions	13

1. Introduction

Neurodegenerative diseases represent a significant challenge in current medical research. Due to the aging of the global population, the number of people affected by these pathologies is constantly increasing. In some cases, the cause of the disease is still unknown, and the available treatments are often limited to alleviating symptoms rather than curing the disease itself, an approach known as symptomatic therapy. In many clinical cases, the disease is characterized by the agglomeration of prions, which are misfolded proteins that can induce other proteins to misfold as well, leading to a chain reaction of misfolding, growth and spreading. This process appears to be similar in other age-related neurodegenerative diseases, such as Alzheimer's disease, Parkinson's disease, and amyotrophic lateral sclerosis.

Finding a complete model describing neurodegenerative diseases is an open problem, but some key features can be extracted from the stereotypic behavior of prions: once they are seeded, misfolded proteins can spread across the brain by two distinct mechanisms, extracellular diffusion and axonal transport. The latter is a process in which the misfolded protein is transported along the axons of neurons, which are the long, slender projections of nerve cells that transmit signals throughout the nervous system. This type of transport is determined by the local axonal direction, which can vary across the brain. Three main models are described in the literature:: the monomeric seeding model, in which a single misfolded protein can induce the misfolding of a healthy one, the co-operative autocatalysis model, in which an aggregate of an healthy and a misfolded protein can induce the misfolding of a healthy one, and finally the polymeric seeding model, in which a polymeric misfolded protein can induce the misfolding of a monomeric healthy protein.

In this work, we introduce the Fisher-Kolmogorov equation and study its qualitative behavior under different initial conditions and parameters. Furthermore, we define the weak formulation of the Fisher-Kolmogorov equation and then we obtain the semi-discrete formulation employing the Galerkin method for space discretization. Next, we introduce the Theta-method to obtain the fully-discrete formulation of the problem, and we implement a stepsize adaptivity procedure for the Forward Euler scheme, using a suitable a posteriori error estimator. We then apply the Newton's method to solve the nonlinear problem, obtaining a linear system at each step.

We propose a solver for the problem using the dealII [1] library. We calibrate the model parameters using the one proposed in [7], for modelling Alzheimer's disease, Parkinson's disease and amyotrophic lateral sclerosis. We implement and test different axonal fields, to model the anisotropic diffusion of the misfolded protein, as well as different initial conditions, to model the seeding. In addition to that, we implement a white/gray matter partitioning of the brain, to account for the different diffusion properties of the two types of tissues.

2. The mathematical model

In this section, we introduce the chosen mathematical model as well as its characteristics. Then, we will introduce the weak formulation of the problem, which will be the starting point for the numerical discretization.

2.1. The Fisher-Kolmogorov model

Several mathematical models for prion dynamics can be found in the literature. One of the most complete models is the Smoluchowski model [5]. Even though this results to be computationally expensive, it is able to describe time evolution during aggregation and diffusion of coagulating particles. Other examples can be the Heterodimer model [2], which treats separately the healthy and the misfolded proteins.

In this work, we consider the Fisher-Kolmogorov (FK) equation, which describes a non-linear diffusion reaction problem. This is a more simplified model, compared to Smoluchowski and Heterodimer models, which accounts for the common features of various types of neurodegenerative disorders. Picking a final time $T > 0$, the solution to our problem $c = c(\mathbf{x}, t)$, depends on the position $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$) and the time $t \in (0, T]$.

$$\begin{cases} \frac{\partial c}{\partial t} - \nabla \cdot (\mathbf{D} \nabla c) - \alpha c(1 - c) = 0 & \text{in } \Omega \times (0, T] \\ (\mathbf{D} \nabla c) \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \times (0, T] \\ c(t = 0) = c_0 & \text{in } \Omega \end{cases} \quad (1)$$

System of equations (1) describes a nonlinear parabolic problem equipped with homogeneous Neumann boundary conditions and non-homogeneous initial condition, $c_0 \geq 0$.

In the case of prion-like diseases, c denotes the concentration level of misfolded protein, $\alpha > 0$ signifies the rate of concentration increase, and $\mathbf{D} = \mathbf{D}(\mathbf{x})$, usually denoted as the diffusion tensor, describes the spreading of misfolded proteins inside the brain. The latter can be expressed as the contribution of two terms: extracellular diffusion d^{ext} and axonal transport d^{axn} , with $d^{ext}, d^{axn} \geq 0$. In fact, once seeded, misfolded protein can spread

across the brain uniquely by these two distinct mechanisms. Thus, we can express

$$\mathbf{D} = d^{ext} \mathbf{I} + d^{axn} \mathbf{m} \otimes \mathbf{m} \quad (2)$$

where $\mathbf{m} = \mathbf{m}(\mathbf{x})$ represents the local axonal direction at the point $\mathbf{x} \in \Omega$. It is generally assumed that axonal transport is faster than extracellular diffusion $d^{axn} \geq d^{ext}$.

Throughout this report we will use a standard notation of Sobolev spaces and the scalar product in $L^2(\Omega)$, denoted by $(\cdot, \cdot)_\Omega$. The induced norm is denoted by $\|\cdot\|_\Omega$. For vector-valued and tensor-valued functions, the definition extends componentwise.

Under the following regularity assumptions provided in [3]:

- $\mathbf{D} \in L^\infty(\Omega, \mathbb{R}^{d \times d})$ and $\exists g > 0, \forall \gamma \in \mathbb{R}^d : g|\gamma|^2 \leq \gamma^T \mathbf{D} \gamma \quad \forall \gamma \in \mathbb{R}^d$
- $c_0 \in L^2(\Omega)$

and restricting to the case $c_0(\mathbf{x}) \in [0, 1]$, we have $c(\mathbf{x}, t) \in [0, 1] \forall \mathbf{x} \in \Omega$ and $t > 0$. In addition to that, we have two obvious solutions: an unstable equilibrium in $c(\mathbf{x}, t) = 0$ and a stable one $c(\mathbf{x}, t) = 1$. This is due to the reaction term $c - c^2$: the solution is repelled from $c = 0$ and attracted to $c = 1$. This means, that any non-zero value of the concentration, will generate an irreversible chain reaction:

$$\lim_{t \rightarrow \infty} c(\mathbf{x}, t) = 1 \quad \text{if } \exists \mathbf{x} \in \Omega : c_0(\mathbf{x}) > 0.$$

2.2. Weak formulation

Selecting the Sobolev functional space $V := H^1$ for choosing both the test function and the weak solution c , the weak formulation reads:

Find for all $t \in (0, T)$, $c(t) \in V$ such that $c(t = 0) = c_0$ and

$$\int_\Omega \frac{\partial c}{\partial t} v \, d\mathbf{x} + \int_\Omega \mathbf{D} \nabla c \cdot \nabla v \, d\mathbf{x} - \int_\Omega \alpha c(1 - c)v \, d\mathbf{x} = 0 \quad \forall v \in V \quad (3)$$

We define the weak nonlinear form $b : V \times V \rightarrow \mathbb{R}$, corresponding to the spatial part of the strong problem, as follows:

$$b(w)(v) = \int_\Omega \mathbf{D} \nabla w \cdot \nabla v \, d\mathbf{x} - \int_\Omega \alpha w(1 - w)v \, d\mathbf{x} \quad \forall w, v \in V \quad (4)$$

This form is nonlinear in the first variable but linear in the second.

The weak formulation (3) now reads:

Find for all $t \in (0, T)$, $c(t) \in V$ such that $c(t = 0) = c_0$ and

$$\int_\Omega \frac{\partial c}{\partial t} v \, d\mathbf{x} + b(c)(v) = 0 \quad \forall v \in V. \quad (5)$$

The correct way to theoretically treat this nonlinear problems consists in applying the Newton-Kantorovich Theorem, which generalizes the Newton's method to infinite-dimensional Banach spaces. Once the problem is reduced to a sequence of infinite dimensional linear problems, spatial discretization can be applied to obtain a sequence of finite dimensional linear problems. This is the so-called *differentiate-then-discretize* approach, and is explained in detail for nonlinear elliptic problems in [4]. Moreover, the time discretization represents a further challenge to fitting the problem into the framework of the Newton-Kantorovich Theorem. For simplicity, we will not treat the problem in this way, but we will instead apply Newton's method directly to the fully discrete problem and we assume well-posedness at every step. This approach is used in practice and, even though we do not have a theoretical guarantee of convergence, the resulting algebraic system is identical to the one obtained through the Newton-Kantorovich Theorem.

Nonetheless, we still derive the Fréchet derivative $a(w)(\delta, v)$ of the weak form $b(w)(v)$, since its restriction to finite dimensional subspaces of V corresponds to the Jacobian that is used in the Newton's method.

The Fréchet derivative of a functional $b : V \rightarrow \mathbb{R}$ at a point $w \in V$ in the direction $\delta \in V$ is defined as:

$$a(w)(\delta, v) = \lim_{\epsilon \rightarrow 0} \frac{b(w + \epsilon \delta)(v) - b(w)(v)}{\epsilon} \quad (6)$$

Applying this definition, we arrive at the following expression:

$$a(w)(\delta, v) = \int_\Omega \mathbf{D} \nabla \delta \cdot \nabla v \, d\mathbf{x} - \int_\Omega \alpha(1 - 2w)\delta v \, d\mathbf{x} \quad (7)$$

3. Numerical Methods

3.1. Semi-discrete formulation

To obtain the Galerkin-Finite Element approximation, we introduce a triangulation T_h over the domain Ω . Let us define $\mathbb{P}^r(K)$ as the space of polynomials of degree at most r on the mesh element K . Then, we can introduce the following finite element space:

$$\chi_h^r := \{w_h \in C^0(\Omega) : w_h|_K \in \mathbb{P}^r(K) \forall K \in T_h\} \quad (8)$$

and we set $V_h(\Omega) = X_h^r \cap H^1(\Omega)$, which is the space of continuous piecewise polynomials of degree at most r on the mesh T_h . Clearly, $V_h(\Omega) \subset V$ and $V_h(\Omega)$ is finite dimensional. The semi-discrete formulation reads:

Find for all $t \in (0, T)$, $c_h(t) \in V_h$ such that $c_h(t=0) = c_{0,h}$ and

$$\int_{\Omega} \frac{\partial c_h}{\partial t} v_h \, d\mathbf{x} + b(u_h)(v_h) = 0 \quad \forall v_h \in V_h. \quad (9)$$

where $c_{0,h}$ is a suitable approximation of c_0 in V_h .

3.2. Fully-discrete problem

We introduce a time discretization of the interval $[0, T]$ by defining a partition of N intervals $0 = t_0 < t_1 < \dots < t_N = T$. We denote by a superscript n the approximate solution at time t_n , i.e. $c_h^n \approx c_h(t_n)$. The fully discrete formulation of the problem by means of the θ -method, can be expressed as:

Find for all $n = 0, 1, \dots, N-1$, $c_h^{n+1} \in V_h$ such that $c_h^0 = c_{0,h}$ and

$$\int_{\Omega} \frac{c_h^{n+1} - c_h^n}{\Delta t} v_h \, d\mathbf{x} + b((1-\theta)c_h^n + \theta c_h^{n+1})(v_h) = 0 \quad \forall v_h \in V_h. \quad (10)$$

where $\Delta t = t_{n+1} - t_n$ is the time step and $\theta \in [0, 1]$ is a parameter that allows to choose the method. In particular, $\theta = 0$ corresponds to the Forward Euler method, $\theta = 1/2$ corresponds to the Crank-Nicolson method and $\theta = 1$ corresponds to the Backward Euler method.

We then define the residual $\forall n = 0, 1, \dots, N-1$, $R : V_h \times V_h \rightarrow \mathbb{R}$ as:

$$R(c_h^{n+1})(v_h) = \int_{\Omega} \frac{c_h^{n+1} - c_h^n}{\Delta t} v_h \, d\mathbf{x} + b((1-\theta)c_h^n + \theta c_h^{n+1})(v_h). \quad (11)$$

This allows us to rewrite the fully discrete problem in terms of the nonlinear residual:

Find for all $n = 0, 1, \dots, N-1$, $c_h^{n+1} \in V_h$ such that $c_h^0 = c_{0,h}$ and

$$R(c_h^{n+1})(v_h) = 0 \quad \forall v_h \in V_h. \quad (12)$$

3.2.1. Adaptive Forward Euler

Even if Forward Euler is an explicit method, and thus easy to implement and computationally efficient, it is not unconditionally stable. This means that the time step Δt must be chosen small enough to ensure stability. To this end, we can use a posteriori error estimator to adaptively control the time step size. This procedure is called *stepsize adaptivity*, where the time step size is adjusted according to the estimated error in the solution. The following stepsize adaptivity procedure is based on the a posteriori error estimator proposed in [6]. The idea is to compute at each time step two solutions: one using Δt and the other using $\Delta t/2$. Then, we feed the estimator with the difference between the two solutions, and if the estimated error is larger than a given tolerance ϵ , we reduce the time step size and recompute the solution. To avoid excessively small time steps, in addition to ϵ , we set a percentage η to define a window of acceptance for the error (figure 1). In this way, we can ensure that the solution is computed with a time step size that is small enough to ensure stability, but not too small to waste computational resources.

The a posteriori error estimator is defined as follows:

$$\|c(t_n) - c_h^{n+1/2}\|_{L(\Omega)} \approx E = \frac{1}{3} \|c_h^{n+1/2} - c_h^{n+1}\|_{L^2(\Omega)} \quad (13)$$

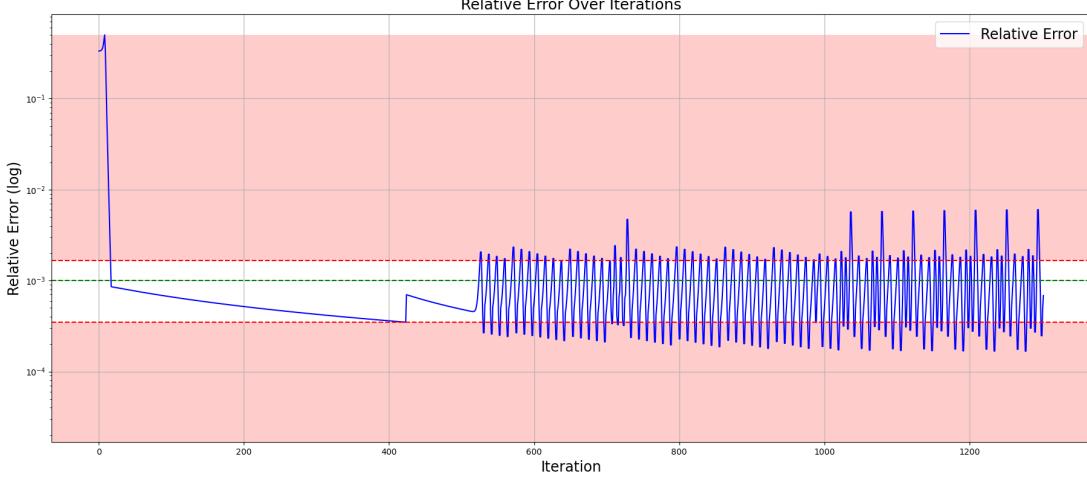


Figure 1: Relative error E_r of the solution at each time step, computed with the stepsize adaptivity procedure. $\alpha = 0.45$, $r = 1$, $\epsilon = 10^{-3}$, $\eta = 0.6$, initial $\Delta t = 1$, $d^{ext} = 1.5$, $d^{axn} = 3$, Radial Axonal Field, $c_0(x_1, x_2) = 0.4e^{-(x_1^2+x_2^2)/(0.01)}$

or alternatively, we can use its relative version:

$$E_r = \frac{\|c_h^{n+1/2} - c_h^{n+1}\|_{L^2(\Omega)}}{3\|c_h^{n+1/2}\|_{L(\Omega)}} \quad (14)$$

Where $c(t_n)$ is the exact solution at time t_n , $c_h^{n+1/2}$ is the solution computed with time step $\Delta t/2$ and c_h^{n+1} is the solution computed with time step Δt .

The stepsize adaptivity procedure can be summarized as follows:

1. Set the initial time step Δt and the tolerance ϵ .
2. Compute the solution c_h^{n+1} using the Forward Euler method with time step Δt .
3. Compute the solution $c_h^{n+1/2}$ using the Forward Euler method with time step $\Delta t/2$.
4. Estimate the error as E using (13) or E_r (14).
5. If $E > \epsilon(1 + \eta)$, reduce the time step size by half and go to step 2.
6. If $E < \epsilon(1 - \eta)$, double the time step and go to step 2.
7. If $\epsilon(1 - \eta) \leq E \leq \epsilon(1 + \eta)$, accept the solution $c_h^{n+1/2}$ and proceed to the next time step.

3.3. Newton's method

To solve the fully discretized nonlinear problem (12) we employ Newton's method at each time step. As mentioned before, the Jacobian of the discrete residual is given by the restriction of the Fréchet derivative of the weak form b to V_h . This leads to the following iterative scheme, where the superscript (k) denotes the k -th iteration of the Newton's method:

1. For $n = 0, 1, 2, \dots, N_T - 1$ do:
 - (a) Take as initial guess $c_h^{n,(0)} = c_h^n$.
 - (b) Compute $\delta^{n,(k)}$ by solving the tangent problem corresponding to the weak residual:

$$\int \frac{\delta_h^{(k)}}{\Delta t} d\mathbf{x} + \theta a((1 - \theta)c_h^{n,(k)} + \theta c_h^{n+1,(k)})(\delta_h^{(k)}, v_h) = -R(c_h^{n+1,(k)})(v_h) \quad \forall v_h \in V_h. \quad (15)$$

- (c) Update the solution $c_h^{n+1,(k+1)} = c_h^{n+1,(k)} + \delta_h^{(k)}$.
- (d) Check the convergence criterion: $\|R(c_h^{n+1,(k+1)})\| \leq \epsilon$.

Now we take a suitable basis of V_h , denoted by $\{\varphi_i\}_{i=1}^{N_h}$, where $\dim V_h = N_h$. The tangent problem 15 is linear, thus we can express $\delta_h^{(k)}$ and $c_h^{n,(k)}$ as a linear combination of the basis functions:

$$\delta_h^{(k)}(\mathbf{x}) = \sum_{j=1}^{N_h} B_j^{(k)} \varphi_j(\mathbf{x}) \quad (16)$$

$$c_h^{n,(k)}(\mathbf{x}) = \sum_{j=1}^{N_h} C_j^{n,(k)} \varphi_j(\mathbf{x}) \quad (17)$$

where $B^{(k)} = (B_1^{(k)}, B_2^{(k)}, \dots, B_{N_h}^{(k)})^T$ and $C^{n,(k)} = (C_1^{n,(k)}, C_2^{n,(k)}, \dots, C_{N_h}^{n,(k)})^T$ are the coefficients of the linear combination.

Now, we define the mass and the stiffness matrix, $\mathbf{A}^{n+1,(k)} \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$, respectively, as follows:

$$[\mathbf{A}^{n+1,(k)}]_{ij} = \theta a((1 - \theta)c_h^{n,(k)} + \theta c_h^{n+1,(k)})(\varphi_j, \varphi_i) \quad \forall i, j = 1 \dots N_h \quad (18)$$

$$[\mathbf{M}]_{ij} = \int_{\Omega} \varphi_j \varphi_i d\mathbf{x} \quad \forall i, j = 1 \dots N_h \quad (19)$$

and the righthand side vector $\mathbf{R}^{n+1,(k)} \in \mathbb{R}^{N_h}$, which is defined as:

$$[\mathbf{R}^{n+1,(k)}]_i = R(c_h^{n+1,(k)})(\varphi_i) \quad \forall i = 1 \dots N_h \quad (20)$$

The tangent linear system can then be expressed in algebraic form:

$$\frac{1}{\Delta t} \mathbf{M} \mathbf{B}^{(k)} + \mathbf{A}^{n+1,(k)} \mathbf{B}^{(k)} = -\mathbf{R}^{n+1,(k)} \quad (21)$$

At each k-th step of the Newton method, we need to assemble $\mathbf{A}^{n+1,(k)}$ and $\mathbf{R}^{n+1,(k)}$ with the updated solution $C^{n+1,(k+1)}$, and then solve the linear system 21 in $\mathbf{B}^{(k)}$.

3.4. Solving the linear system

The linear system 21 is symmetric but not always positive definite, due to the presence of the reaction term. In fact, for certain values of the parameters and of the solution $c^{n+1,(k)}$, one can verify that the linear system is not positive definite. For this reason we employ the MINRES method, which is suitable for solving symmetric but indefinite linear systems. As the linear system may be ill-conditioned, we apply a preconditioner to improve the convergence of the MINRES method. We tried several preconditioners provided by the Trilinos library, such as the incomplete LU (ILU) preconditioner, the symmetric successive over-relaxation (SSOR) preconditioner, and the algebraic multigrid (AMG) preconditioner. The preconditioner is recomputed at each time step, and at each Newton's iteration, as the solution $c^{n+1,(k)}$ changes and thus the $\mathbf{A}^{n+1,(k)}$ matrix changes as well. We observe that the SSOR preconditioner is the most effective in terms of speed and convergence, while the ILU is the most robust in terms of convergence. Further investigation should be carried out in terms of memory usage and speed, as it may impact the performance of the solver in large-scale problems.

4. Application to Brain Modelling

To apply the Fisher-Kolmogorov model to simulate neurodegenerative disease progression, several key components must be established: a computational domain representing the brain, appropriate definitions for the anisotropic diffusion tensor \mathbf{D} reflecting neural pathways, realistic initial conditions for misfolded protein concentration c_0 , and a method to account for differing properties of brain tissues. This section details our approach to generating suitable computational meshes for these simulations.

4.1. Mesh Generation

Our study utilized both three-dimensional (3D) and two-dimensional (2D) computational meshes to represent brain geometry. The 3D model allows for a comprehensive representation, while the 2D sagittal model, similar to that used for parametric studies in [7], offers reduced computational cost, facilitating initial model testing and parameter exploration.

4.1.1. 3D Volumetric Mesh

The provided STL model of the brain hemisphere was imported into Gmsh. A Gmsh scripting file (`brain_3d.geo`) was developed to control the meshing process. This script instructed Gmsh to generate a volumetric mesh composed of tetrahedral elements, suitable for 3D finite element analysis. The resulting 3D mesh consists of 634,472 tetrahedral cells and 111,951 vertices.

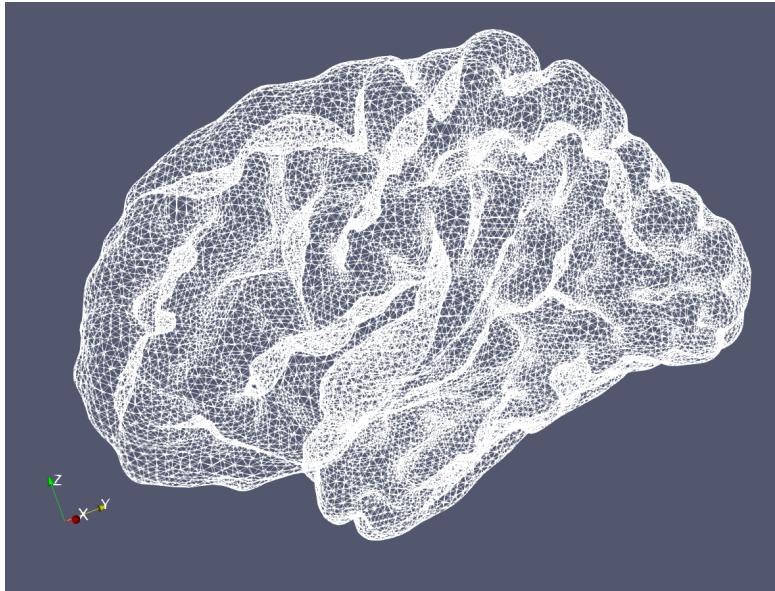


Figure 2: Volumetric tetrahedral mesh of the brain hemisphere generated with Gmsh from the initial STL file. The mesh comprises 634,472 cells and 111,951 vertices. Only the meshed surface is shown to avoid cluttering.

4.1.2. 2D Sagittal Mesh

To create a 2D sagittal mesh, the 3D STL model was first imported into Paraview. Paraview's "Slice" filter was employed to extract a 2D cross-section corresponding to a sagittal view of the brain. Critically, this slice was chosen to include internal anatomical features, such as the ventricle, which appears as an internal void or "hole" within the 2D domain. This characteristic is consistent with the sagittal model morphology depicted in Figure 4 of [7].

The boundary of this 2D slice was exported from Paraview as an .obj (Wavefront OBJ) file. This file format describes the geometry as a collection of vertices and line segments defining the external and internal (ventricular) boundaries of the slice. To facilitate 2D meshing with Gmsh, a custom Python script (`obj_slice_to_msh.py`) was developed. This script parsed the .obj file, extracting the vertex coordinates and line connectivity, and then programmatically generated a Gmsh .geo script (`slicegeo`). This .geo script defined the points, lines (forming the boundaries), and planar surface for Gmsh.

Finally, this .geo script was processed by Gmsh to generate a 2D mesh composed of triangular elements. The resulting 2D sagittal mesh contains 120,590 triangular elements and 60,828 vertices.

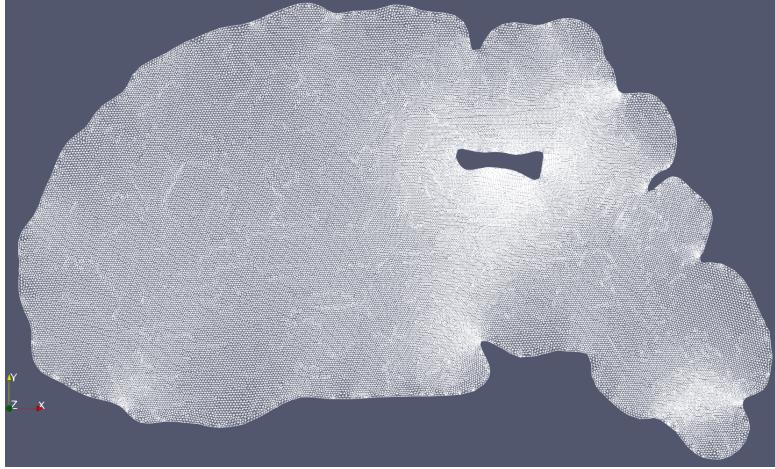


Figure 3: 2D triangular mesh of a sagittal brain slice, generated using Gmsh. The geometry was derived from a Paraview slice of the 3D STL model and includes an internal void representing the ventricle. The mesh consists of 120,590 triangles and 60,828 vertices.

4.2. Vector Field Implementation

The anisotropic diffusion term $d^{axn} \mathbf{m} \otimes \mathbf{m}$ in the diffusion tensor (Equation 2) requires the definition of a local direction vector field $\mathbf{m}(\mathbf{x})$. We implemented three fundamental vector field types to model different directional transport patterns:

- **Radial Field:** Vectors emanate outwards from a central point \mathbf{x}_c , with $\mathbf{m}(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{x}_c}{|\mathbf{x} - \mathbf{x}_c| + \epsilon}$ where $\epsilon = 10^{-5}$ prevents division by zero at the center.
- **Circumferential Field:** Vectors run tangentially around a central point or axis. In 2D, the field is orthogonal to the radial direction within the simulation plane. For 3D implementations, we project onto the YZ-plane to maintain computational efficiency while preserving the essential tangential behavior.
- **Axonal Field:** A spatially-varying combination of radial and circumferential patterns, defined by an ellipsoidal boundary. Points inside the ellipsoid follow circumferential patterns, while exterior points follow radial patterns.

The implementation utilizes a template-based architecture supporting both 2D and 3D domains. For the composite field, we define an ellipsoidal boundary using semi-axes parameters. The field selection criterion $\sum_{i=1}^{DIM} \left(\frac{x_i - x_{c,i}}{a_i} \right)^2 < 1$ determines whether a point lies within the ellipsoid, where a_i represents the semi-axis length in the i -th direction.

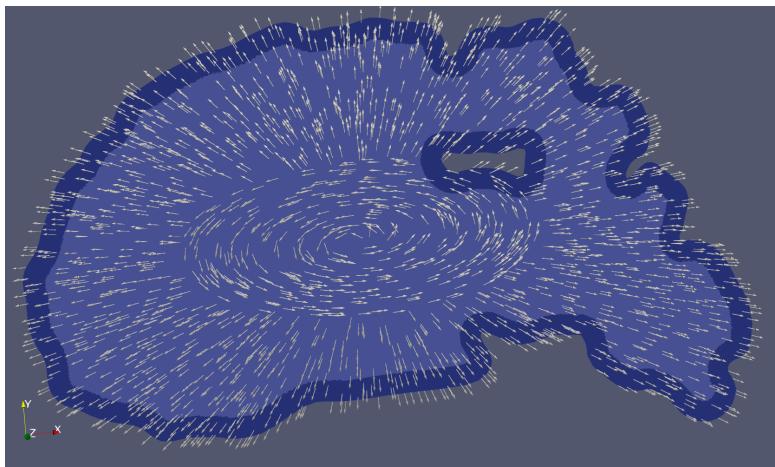


Figure 4: Axonal fiber field visualization over the 2D sagittal mesh.

4.3. Initial Conditions

The initial condition for the concentration field, $c_0(\mathbf{x})$, is fundamental to the subsequent evolution of the PDE system. We investigated several functional forms for $c_0(\mathbf{x})$ to define a localized initial concentration and analyze its impact on the stability and dynamics of the numerical solution. The implemented profiles include:

- **Piecewise Constant:** A constant value, $c_0 = \text{const}$, inside a spherical region around a seeding point and zero elsewhere. This introduces a sharp discontinuity at the boundary of the seeding region.
- **Gaussian Profile:** A smooth, localized peak centered at \mathbf{x}_s , defined by $c_0(\mathbf{x}) = A \exp(-k\|\mathbf{x} - \mathbf{x}_s\|^2)$. This provides a C^∞ initial condition.
- **Smooth Bump Function:** A compactly supported C^∞ function, constructed to be non-zero only within a specific radius of a central point, providing a smooth and strictly localized initial state.

To reproduce the initial conditions used in [7], we implemented a `SeedingRegions` class that allows for the definition of rectangular (in 2D) or cuboidal (in 3D) regions of the meshes where the initial concentration is set to a specified value (e.g. 0.2). We provide predefined regions such as Tau inclusions and Alpha-synuclein inclusions, which can be easily selected and applied to the mesh. The initial condition is then set to zero outside these regions, ensuring a localized seeding of misfolded proteins.

4.4. White/Gray Matter Partitioning

To model physical systems with spatially varying properties, the governing PDE coefficients must be defined as piecewise functions over the domain. In our application, this corresponds to distinguishing between white matter (WM) and gray matter (GM), which are assumed to have different properties. This heterogeneity is incorporated into the finite element model by assigning different values for the diffusion coefficients (d^{ext} , d^{int}) and the reaction coefficient (α) based on a cell's material type.

We implemented a geometric partitioning scheme to assign a `material_id` to each cell in the mesh. The gray matter is approximated as a layer of constant thickness adjacent to the domain boundary, $\partial\Omega$. The partitioning is performed using a two-pass algorithm after the mesh is loaded, leveraging the facilities of the `deal.II` library:

1. **Boundary Cell Identification:** In the first pass, the algorithm iterates over all cells of the global (serial) triangulation. The centroids of cells that lie on the domain boundary are collected and stored in a vector that is shared across all parallel processes.
2. **Distance-Based Classification:** In the second pass, each process iterates over its locally-owned cells in the distributed triangulation. For each cell, it computes the Euclidean distance from its centroid to every boundary centroid identified in the first pass. If the minimum computed distance is less than a prescribed `distance_threshold`, the cell is classified as gray matter and its `material_id` is set to 1. Otherwise, it is classified as white matter and its `material_id` is set to 0.

This pre-processing step effectively partitions the mesh into two subdomains. The assigned `material_id` is then queried during the finite element assembly loop to supply the correct, spatially-dependent coefficient values when constructing the local stiffness and mass matrices for each cell. This allows for a straightforward and efficient implementation of a spatially heterogeneous model.

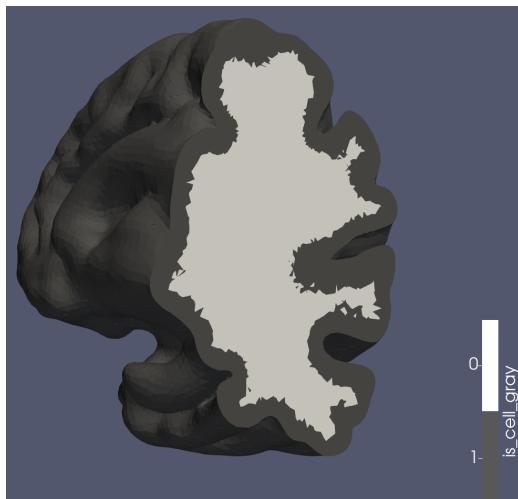


Figure 5: Illustration of the mesh partitioning into gray matter (exterior layer) and white matter (interior), distinguished by different `material_ids`. This partitioning enables the use of region-specific model parameters during the finite element matrix assembly.

5. Results

5.1. Some numerical observations

In this section we present some numerical observations that we made during the implementation of the Fisher-Kolmogorov model. These observations are not directly related to the main results of the paper, but they are important to understand the numerical behaviour of the model and the implementation choices that we made.

5.1.1. Convergence of the Newton's method

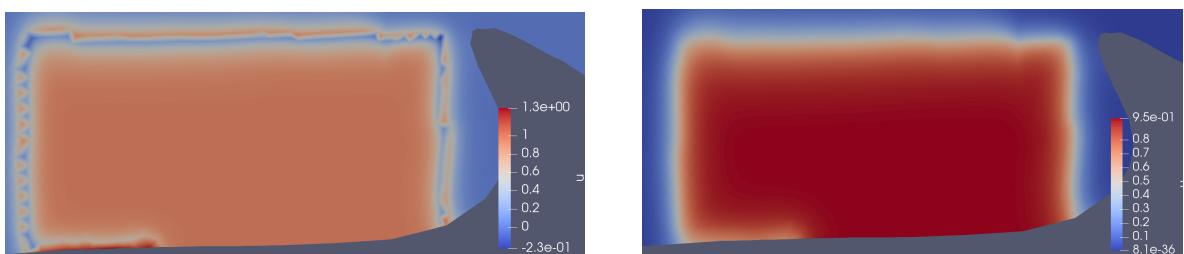
It is known from theory that θ methods are unconditionally stable for $\theta \in [0.5, 1]$, while for $\theta < 0.5$ the method is conditionally stable and the time step size must be chosen small enough to ensure stability. But in our case we are combining the θ method with the Newton's method to solve the nonlinear problem. In particular the discrete residual we are linearizing contains in its definitions terms involving theta, so the two methods are tightly interconnected. Since we are using as initial guess for the Newton's method the solution computed at the previous time step, we cannot choose the time step size arbitrarily large, otherwise the true solution at the next time step will be too far from the initial guess and the Newton's method will not converge. We experimented with large time steps and we observed divergent behaviour of the Newton's method, *independently of the value of θ* . This is not in contrast with the unconditionally stable behaviour of the θ method, since Newton's method should also be taken into account in the analysis.

5.1.2. Stability observations

Depending on the choice of parameters, we observed that the numerical solution of the Fisher-Kolmogorov equation can exhibit instability, in particular it can become negative, a behavior that is theoretically not possible for the continuous equation, and then explode to negative infinity. If certain parameters are chosen, we couldn't get rid of this instability in any way (time steps refinement, different θ values, smoother initial conditions, etc.). The Fisher-Kolmogorov equation has a diffusion term (similar to the heat equation) and a (nonlinear) reaction term. The diffusion term is responsible for the spreading of the solution, while the reaction term is responsible for the growth of the solution. On one hand it is known from theory that the heat equation has a regularizing effect, which means that even if the initial condition is discontinuous, the solution is smooth for any strictly positive time. On the other hand, we observed that the aforementioned instabilities occur whenever the reaction term is too strong, i.e. when the growth rate α is too large compared to the diffusion coefficients d^{ext} and d^{axn} . If the diffusion coefficients are strong enough, the solution remains positive and correctly tends to the steady state.

We also observed that the form of the initial condition plays a crucial role in the stability of the numerical solution. In particular, we found that using a piecewise constant initial condition leads an oscillatory behaviour during the first time steps along the discontinuity of the initial condition, which smooths out as time goes on. This behaviour was observed for any θ value except for $\theta = 1$ (Backward Euler), which could be linked to stronger stability properties of the Backward Euler method, like the concept of *L-stability*.

On the other hand, if a smooth compactly supported initial condition is used, the solution does not exhibit this oscillatory behaviour and remains stable for many values of θ and for larger time steps.



(a) Crank-Nicolson. The oscillatory behaviour is visible and the solution is below zero.

(b) Backward Euler. The solution is smooth and positive, no oscillatory behaviour is present.

Figure 6: Comparison of numerical solutions evolved from piecewise constant initial conditions after one timestep.

5.2. Sensitivity Analysis

This section quantifies how the misfolded-protein front responds to (*i*) variations of the kinetic and transport **coefficients** and (*ii*) changes in the **seeding geometry and fibre orientation**. All simulations use the baseline parameters reported by Weickenmeier *et al.* [7]: isotropic (extra-cellular) diffusion $d^{\text{ext}} = 1.5 \text{ cm}^2 \text{ yr}^{-1}$, axonal transport $d^{\text{axn}} = 3.0 \text{ cm}^2 \text{ yr}^{-1}$ in white matter ($d^{\text{axn}} = 0$ in grey matter) and growth rates $\alpha_{\text{WM}} = 0.6 \text{ yr}^{-1}$, $\alpha_{\text{GM}} = 0.3 \text{ yr}^{-1}$. In the following, we will refer to the activation time of a point in the mesh at the instant when the concentration there reaches a numerical threshold of 0.95.

Note that one numerical time step corresponds to $\Delta t = \frac{1}{3} \text{ yr}$, so that the activation-time legend [56, 172] time steps in our plots represents $\approx [18.7, 57.3] \text{ yr}$.

5.2.1. Sensitivity analysis – coefficients

In this section, we will perform a sensitivity analysis on the coefficients of the PDE system. We will vary the extra-cellular diffusion coefficient d^{ext} , the axonal transport coefficient d^{axn} and the reaction coefficient α one at a time and observe the effect on the activation time.

Figure 7 compares the baseline to three targeted perturbations:

- (a) 4 × increase in d^{ext} (top right),
- (b) 8 × increase in d^{axn} (bottom left),
- (c) 2 × increase in α (bottom right).

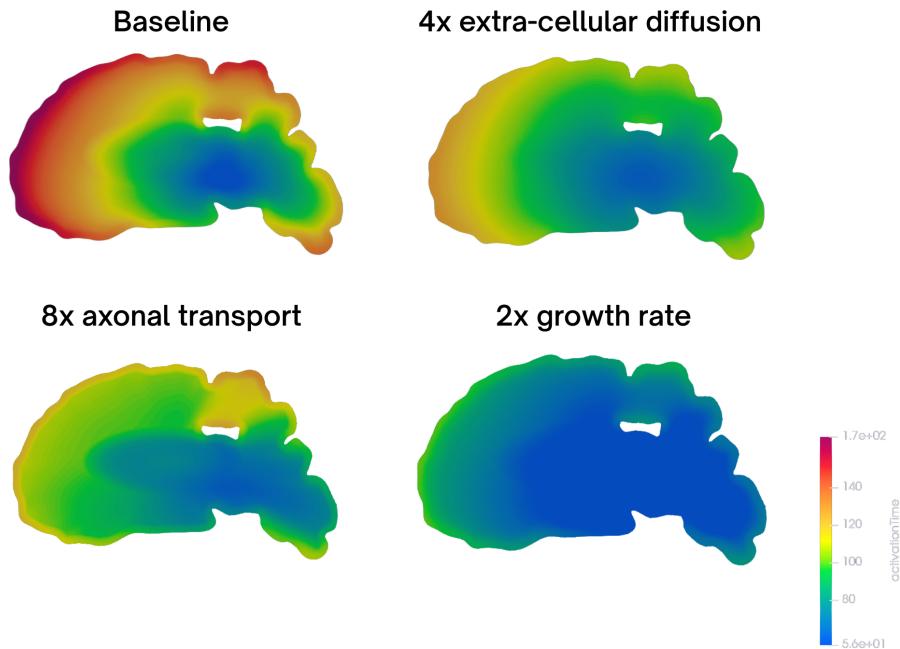


Figure 7: Activation time maps (blue = earliest, red = latest). The colour scale is inverted with respect to the convention in [7]. The legend shows as unit of measure the number of time steps, while the activation times are in years. As a consequence, the activation time ranges between approximately 18.7 years and 57.3 years.

Key observations.

- **Baseline.** The wave starts in the locus coeruleus/trans-entorhinal region and follows the stereotypical Braak trajectory, matching Fig. 8 of Weickenmeier *et al.* [7].
- **Higher d^{ext} .** Faster / smoother propagation, but the earliest activation shifts to the brain stem, indicating that excessive isotropy suppresses the anatomically guided spread.
- **Higher d^{axn} .** Propagation is channelled even more strongly along the ventricular fibre loop; the front rapidly circumvents the ventricle and reaches frontal regions sooner, reproducing the trend noted by the authors when the $d^{\text{axn}}:d^{\text{ext}}$ ratio exceeds 2:1 [7].

- **Higher α .** Growth accelerates locally, shortening the overall time-to-saturation, but the spatial pattern remains almost unchanged – corroborating the original conclusion that *transport*, rather than *kinetics*, dictates the shape of the wavefront.

5.2.2. Sensitivity analysis – fiber orientation and seeding regions

Figure 8 shows concentration maps taken when the global misfolded-protein load reaches 50 %. Four seeding scenarios (amyloid- β , tau, α -synuclein, and TDP-43) are crossed with four transport set-ups: isotropic, circumferential, radial, and axon-based anisotropy.

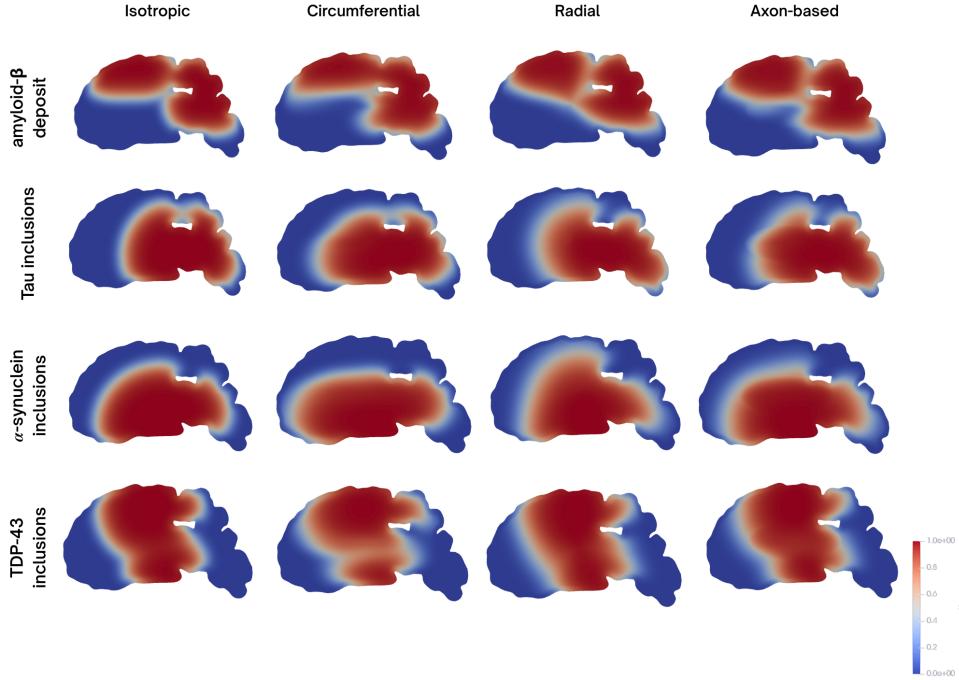


Figure 8: Sensitivity analysis with respect to the fiber orientation and initial seeding region. All the simulations are performed with the baseline parameters, as shown in [7] this figure shows concentration maps when the global misfolded-protein load reaches 50 %

Key observations.

1. **Isotropic diffusion** produces almost concentric activation shells independent of the seed, failing to reproduce clinical patterns.
2. **Circumferential orientation** traps the wave around the ventricle; outward invasion is delayed, echoing the elliptical activation reported in Fig. 9a of [7].
3. **Radial orientation** generates spoke-like fronts that over-emphasise cortical invasion and under-represent subcortical progression.
4. **Axon-based field.** Combining circumferential fibres in periventricular white matter with radial projections elsewhere yields four distinct, pathology-specific trajectories that are qualitatively consistent with the autopsy-derived staging schemes reproduced in Fig. 1 of [7]. This confirms that capturing the true axonal geometry is essential for realistic whole-brain predictions.

References

- [1] Daniel Arndt, Wolfgang Bangerth, Maximilian Bergbauer, Marco Feder, Marc Fehling, Johannes Heinz, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, Peter Munch, Jean-Paul Pelteret, Bruno Turcksin, David Wells, and Stefano Zampini. The deal.II library, version 9.5. *Journal of Numerical Mathematics*, 31(3):231–246, 2023.
- [2] Mattia Corti. Exploring tau protein and amyloid-beta propagation: a sensitivity analysis of mathematical models based on biological data. 04 2024.
- [3] Mattia Corti, Francesca Bonizzoni, Luca Dede’, Alfio M. Quarteroni, and Paola F. Antonietti. Discontinuous galerkin methods for fisher-kolmogorov equation with application to -synuclein spreading in parkinson’s disease. *Computer Methods in Applied Mechanics and Engineering*, 417:116450, 2023.
- [4] Patrick E. Farrell. Finite element methods for pdes, 2021. Lecture notes for course C6.4, Hilary Term 2021, University of Oxford.
- [5] B. Franchi and S. Lorenzani. From a microscopic to a macroscopic model for alzheimer disease: Two-scale homogenization of the smoluchowski equation in perforated domains. *Journal of Nonlinear Science*, 26:717–753, 2016.
- [6] Alfio Quarteroni, Fausto Saleri, and Paola Gervasio. *Scientific Computing with MATLAB and Octave*, 3rd ed. 06 2010.
- [7] Johannes Weickenmeier, Mathias Jucker, Alain Goriely, and Ellen Kuhl. A physics-based model explains the prion-like features of neurodegeneration in alzheimer’s disease, parkinson’s disease, and amyotrophic lateral sclerosis. *Journal of the Mechanics and Physics of Solids*, 124:264–281, 2019.