Fisher-Kolmogorov equations for neurodegenerative diseases

Andrea Boin, Giacomo Pauletti, Lorenzo Pettenuzzo

Contents

1 Introduction

The objective of this project is to apply various numerical methods to solve the Fisher-Kolmogorov equations to reproduce the results of the paper [1]. The Fisher-Kolmogorov equations can be used to effectively model the spread of misfolded proteins in the brain, a process associated with numerous neurodegenerative diseases.

1.1 Fisher-Kolmogorov equation

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

- c: concentration of the misfolded protein in a region of the brain ($0 \leq c \leq 1$)
- α: constant of concentration growth
- D: diffusion coefficient of the misfolded protein.
  It can be isotropic (a scalar) or anisotropic (a square matrix).
  In case of anisotropic coefficient the term can be computed as:

  $$\underline{\mathrm{D}} = d^{\mathrm{ext}}\underline{\mathrm{I}} + d^{\mathrm{axn}}(\mathbf{n} \otimes \mathbf{n})$$

  where d^{ext} is the extracellular diffusion term, d^{axn} is the axonal diffusion term and $\mathbf{n}$ the direction of axonal diffusion.

  Usually extracellular diffusion is slower than axonal diffusion: $d^{\mathrm{ext}} < d^{\mathrm{axn}}$.

  The Fisher-Kolmogorov equation is a **diffusion-reaction** equation with a nonlinear forcing term that can be used to model population growth. In this case it is used to model the spreading of proteins in the brain.

  The interested problem is a **nonlinear parabolic PDE** with **Neumann boundary conditions**.

1.2 Mesh

The mesh we used for the simulation is a 3D representation of a hemisphere of the human brain with 21211 points and 42450 cells.
To process the mesh with our software, we did convert the format from *.stl* to *.msh* using **GMSH** with the following procedure:

1. Import the mesh (*.stl*) in GMSH
2. From the left menu, select "geometry $\rightarrow$ add $\rightarrow$ volume"
3. Save the new generated *.geo* file
4. Define the 3D mesh: "mesh $\rightarrow$ define $\rightarrow$ 3D"
5. Export the file as *.msh*: "file $\rightarrow$ export $\rightarrow$ msh"

1.3 Weak formulation and semi-discretized formulation

By choosing $V = H^1 = \{v \in L^2 | \nabla v \in L^2\}$ and considering a time domain $(0, T)$, the weak formulation of the problem is:

Find $c(t) \in V$ such that $\forall v \in V$ and $\forall t \in (0, T)$:

$$\begin{cases} \int_\Omega \frac{\delta c}{\delta t} v d\Omega + \int_\Omega D \nabla c \nabla v d\Omega - \int_\Omega \alpha c(1-c) v d\Omega = 0 \\ c(t=0) = c_0 \end{cases}$$

By renaming:

- $a(c, v) = \int_\Omega D \nabla c \nabla v d\Omega$
- $n(c, v) = -\int_\Omega \alpha c(1-c) v d\Omega$

By introducing a triangulation $T_h = \{K | \Omega = \bigcup K\}$ of the domain Ω and defining with it a polynomial space

$$X_h = \{v_h \in C^0(\bar{\Omega}) | v_{h|k} \in \mathbb{P}^r(K), \forall K \in T_h\}$$

we can obtain the discrete space $V_h = V \cap X_h$ for our discrete formulation.
The semi-discrete formulation can then be written as:

Find $c_h \in V_h$ such that, $\forall v_h \in V_h$ and $\forall t \in (0,T)$:

$$\int_\Omega \frac{\delta c_h}{\delta t} v_h d\Omega + a(c_h, v_h) + n(c_h, v_h) = 0$$

$$c_h(t=0) = c_{h,0}$$

2 Methods

We studied the problem with 3 methods and implemented 2 of them algorithmically:

- An **explicit** scheme in which all terms have been treated explicitly to handle the nonlinear part of the model.
- A **mixed explicit/implicit** scheme in which the linear terms have been treated implicitly while the nonlinear terms explicitly to get rid of nonlinarities.
- An **implicit** scheme in which all terms in the equation have been treated implicitly, and then the nonlinear parts have been solved with the Newton method.

To obtain a full discretization of the problem we need to partition the time domain in N partitions of size Δt, obtaining $(0,T) = (0, N\Delta t) = \bigcup_{n=1}^{N}(t^n, t^{n+1}]$ where $t^{n+1} - t^n = \Delta t$, $t^0 = 0$ and $t^N = T$. We can then use an upper-index notation to identify time dependent elements: $c^n = c(t^n)$.

2.1 Explicit scheme

The fully discrete formulation for the **explicit** scheme becomes:

Find $c_h(t) \in V_h$ such that, $\forall v_h \in V_h$, $c_h^0 = c_{h,0}$ and $\forall n \in \{0, N\}$:

$$\int_\Omega \frac{c_h^{n+1} - c_h^n}{\Delta t} v_h d\Omega + a(c_h^n, v_h) + n(c_h^n, v_h) = 0$$

By introducing a basis $\{\phi_i\}$ for the space V_h the problem can be written as:

Find $c_h(t) \in V_h$ such that $c_h^0 = c_{h,0}$ and $\forall n \in \{0, N\}$:

$$Mc^{n+1} = F^n$$

where the **mass matrix** can be computed as:

$$M_{ij} = \frac{1}{\Delta t}\langle \phi_j, \phi_i \rangle$$

and the **forcing term** is:

$$F_i^n = \frac{1}{\Delta t}\langle c_{h,j}^n, \phi_i \rangle - a(c_{h,j}^n, \phi_i) - n(c_{h,j}^n, \phi_i)$$

2.1.1 Stability and Accuracy

The accuracy is $O(\Delta t)$ for time and $O(h^2)$ for space.
The stability condition of the explicit scheme is: $\Delta t \leq \min(\frac{h^2}{2D}, \frac{2}{\alpha})$. The first term acts as a bottleneck for the method. With our values for example $(h = 1[cm], D = 1.5[cm/year], \alpha = 0.5[1/year])$, $\Delta t \leq \min(\frac{1}{3}, 4) = \frac{1}{3}$. The following methods allow for a larger choice of Δt and are generally faster so we decided to implement them.

2.2 Mixed explicit/implicit scheme

The full discretization for the **mixed explicit/implicit** scheme is:

Find $c_h \in V_h$ such that, $\forall v_h \in V_h$ and $c_h(t = 0) = c_{h,0}$:

$$\int_\Omega \frac{c_h^{n+1} - c_h^n}{\Delta t} v_h d\Omega + a(c_h^{n+1}, v_h) + n(c_h^n, v_h) = f(v_h)$$

The problem can be rewritten, by introducing a basis $\{\phi_i\}$ for V_h as:

Find $c_h(t) \in V_h$ such that $c_h^0 = c_{h,0}$ and $\forall n \in \{0, N\}$:

$$Mc^{n+1} = F^n$$

where the **mass matrix** can be computed as:

$$M_{ij} = \frac{1}{\Delta t}\langle \phi_j, \phi_i \rangle + a(\phi_j, \phi_i)$$

and the **forcing term** is:

$$F_i^n = \frac{1}{\Delta t}\langle c_{h,j}^n, \phi_i \rangle - n(c_{h,j}^n, \phi_i)$$

2.2.1 Algorithm

2.2.2 Stability and Accuracy

The accuracy for this method is the same as the explicit one: $O(\Delta t)$ for time and $O(h^2)$ for space.
The stability condition though is better: $\Delta t \leq \frac{2}{\alpha} = 4$ allowing for a larger choice of Δt and a quicker convergence.

2.3 Implicit scheme

2.3.1 Stability and Accuracy

Stability and accuracy analysis of the implemented methods.

3 Results and algorithmic comparation

Results of the project for all the implemented algorithms and graphs.

References

[1] J. Weickenmeier, M. Jucker, A. Goriely, and E. 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.