

NMPDE Project 5:

Fisher-Kolmogorov equation for neurodegenerative diseases

GUO HAOYU - 10942356 - haoyu.guo@mail.polimi.it

WANG YANLONG - 11005879 - yanlong.wang@mail.polimi.it

ZHANG YING - 10962084 - ying3.zhang@mail.polimi.it

1. Introduction

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

这是一个非线性反应-扩散方程的初边值问题，是经典的 Fisher-KPP 型方程，被用于建模某些神经退行性疾病中毒性蛋白质在大脑中的扩散与增长行为。

1. 模型分析

$$\frac{\partial c}{\partial t} - \nabla \cdot (D \nabla c) - \alpha c(1 - c) = 0 \quad \text{in } \Omega$$

$c(t, x)$: 表示有毒蛋白质的浓度（如 α -synuclein）。

D : 扩散张量（可为常数或空间变异的正定矩阵），建模蛋白在脑组织中的各向异性扩散。

$\alpha c(1 - c)$: 反应项，具有 logistic 生长特性，建模蛋白质自我增殖。

边界条件： $D\nabla c \cdot n = 0$ ： Neumann（自然）边界条件，表示无通量（蛋白不从边界渗出）。

初始条件： $c(t = 0) = c_0(x)$ ： 设定初始播种区域（比如某个脑区局部蛋白异常积累）。

2. Weak Formulation

Let $v(x) \in V$, where $V = H^1(\Omega) := \{u \in L^2(\Omega) \mid \nabla u \in L^2(\Omega)^d\}$. Multiply the equation by v and integrate over the domain Ω :

$$\int_{\Omega} \left(\frac{\partial c}{\partial t} - \nabla \cdot (D\nabla c) - \alpha c(1 - c) \right) v \, d\mathbf{x} = 0$$

Use the divergence theorem (integration by parts):

$$- \int_{\Omega} \nabla \cdot (D\nabla c) v \, d\mathbf{x} = \int_{\Omega} D\nabla c \cdot \nabla v \, d\mathbf{x} - \int_{\partial\Omega} v (D\nabla c \cdot n) \, ds$$

Since the boundary condition is homogeneous Neumann:

$$D\nabla c \cdot n = 0 \text{ on } \partial\Omega \Rightarrow \int_{\partial\Omega} v (D\nabla c \cdot n) \, ds = 0$$

Substituting everything back:

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

Therefore, the weak formulation reads:

For all $t \in (0, T)$, find $c(t) \in V$ such that $c(0) = c_0$:

$$\int_{\Omega} \frac{\partial c}{\partial t} v \, d\mathbf{x} + \underbrace{\int_{\Omega} D\nabla c \cdot \nabla v \, d\mathbf{x} - \int_{\Omega} \alpha c(1 - c) v \, d\mathbf{x}}_{a(c,v)} = 0,$$

for all $v \in V$.

3. Semi-discrete Formulation

Let us now introduce a mesh \mathcal{T}_h over the domain Ω , and let $V_h = V \cap X_h^r(\Omega)$ be the associated finite element space of degree r . We obtain the **semi-discrete formulation** by looking for $u(t) \in V_h$ and testing against functions $v \in V_h$:

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

$$\int_{\Omega} \frac{\partial c_h}{\partial t} v_h d\mathbf{x} + a(c_h, v_h) = 0 \quad \text{for all } v_h \in V_h.$$

In the above, $c_{0,h} \in V_h$ is an approximation of c_0 .

Introducing the basis functions $\{\varphi_i\}_{i=1}^{N_h}$ of the space V_h , we approximate the solution $c_h(x, t) \in V_h$ as:

$$c_h(x, t) = \sum_{j=1}^{N_h} C_j(t) \varphi_j(x),$$

where $C_j(t)$ are the **unknown time-dependent coefficients**.

The semi-discrete formulation above can be rewritten as:

$$\int_{\Omega} \frac{\partial c_h}{\partial t} v_h dx + \int_{\Omega} D \nabla c_h \cdot \nabla v_h dx - \int_{\Omega} \alpha c_h (1 - c_h) v_h dx = 0$$

And we choose $v_h = \varphi_i$ (test against each basis function). Then the equation becomes:

$$\int_{\Omega} \left(\sum_{j=1}^{N_h} \frac{dC_j}{dt} \varphi_j \right) \varphi_i dx + \int_{\Omega} D \left(\sum_{j=1}^{N_h} C_j \nabla \varphi_j \right) \cdot \nabla \varphi_i dx - \int_{\Omega} \alpha \left(\sum_{j=1}^{N_h} C_j \varphi_j \right) \left(1 - \sum_{k=1}^{N_h} C_k \varphi_k \right) \varphi_i dx = 0$$

Then we simplify it to become a **matrix form**:

$$M \frac{d\mathbf{C}}{dt} + K\mathbf{C} - \mathbf{N}(\mathbf{C}) = 0,$$

$$c(0) = c_0$$

where:

- $\mathbf{C}(t) = [C_1(t), C_2(t), \dots, C_{N_h}(t)]^\top$
- $M_{ij} = \int_{\Omega} \varphi_j \varphi_i dx$ is the **mass matrix**
- $K_{ij} = \int_{\Omega} D \nabla \varphi_j \cdot \nabla \varphi_i dx$ is the **stiffness matrix**
- $N_i(\mathbf{C}) = \int_{\Omega} \alpha c_h (1 - c_h) \varphi_i dx$ is the **nonlinear reaction term**

4. Fully Discrete Formulation

Solution. Let us introduce a partition of the time interval into N_T sub-intervals $(t^n, t^{n+1}]$ of width Δt , with $n = 0, 1, 2, \dots, N_T - 1$ and $t^0 = 0, t^{N_T} = T$. We denote with a superscript n the approximation of the solution at the discrete time t^n , i.e. $c^n \approx c(t^n)$. Let $\theta \in [0, 1]$ be the parameter of the theta method. Then, the fully discrete formulation is the following:

$$M \frac{\mathbf{C}^{n+1} - \mathbf{C}^n}{\Delta t} + K (\theta \mathbf{C}^{n+1} + (1 - \theta) \mathbf{C}^n) - [\theta \mathbf{N}(\mathbf{C}^{n+1}) + (1 - \theta) \mathbf{N}(\mathbf{C}^n)] = 0,$$

$$c(0) = c_0$$

At each time step, solve the nonlinear system for \mathbf{C}^{n+1} :

$$\begin{aligned} & [M + \Delta t \theta K] \mathbf{C}^{n+1} - \Delta t \theta \mathbf{N}(\mathbf{C}^{n+1}) \\ & = [M - \Delta t (1 - \theta) K] \mathbf{C}^n + \Delta t (1 - \theta) \mathbf{N}(\mathbf{C}^n) \end{aligned}$$

5. Newton Iteration Algorithm

In this section, we solve the nonlinear system by the implicit (or backward) Euler method, which sets $\theta = 1$ and newton iteration algorithm at each time step.

$$F(\mathbf{C}^{n+1}) := M \frac{\mathbf{C}^{n+1} - \mathbf{C}^n}{\Delta t} + K \mathbf{C}^{n+1} - \mathbf{N}(\mathbf{C}^{n+1}) = 0$$

非线性项为:

$$\mathbf{N}(\mathbf{C}) = \left[\int_{\Omega} \alpha c_h (1 - c_h) \varphi_i dx \right]_{i=1}^{N_h}$$

由于 $c_h = \sum_j C_j \varphi_j$ ，我们可以计算它的雅可比矩阵的第 i, j 项：

$$J_{ij}(\mathbf{C}) = \frac{\partial}{\partial C_j} \left[\int_{\Omega} \alpha c_h (1 - c_h) \varphi_i dx \right]$$

这可以展开为：

$$J_{ij}(\mathbf{C}) = \int_{\Omega} \alpha [(1 - 2c_h) \varphi_j] \varphi_i dx$$

即：

$$J(\mathbf{C}) = \left[\int_{\Omega} \alpha (1 - 2c_h) \varphi_j \varphi_i dx \right]_{i,j}$$

算法总结：

对每个时间步 $n = 0, 1, \dots$ ：

1. 令初值为上一步解： $\mathbf{C}^{n+1,(0)} = \mathbf{C}^n$

2. 重复直到收敛：

计算残差 $F^{(k)} = F(\mathbf{C}^{n+1,(k)})$

组装 Jacobian 矩阵 $J^{(k)} = J(\mathbf{C}^{n+1,(k)})$

解线性系统：

$$\left[\frac{1}{\Delta t} M + K - J^{(k)} \right] \delta \mathbf{C}^{(k)} = -F^{(k)}$$

更新解：

$$\mathbf{C}^{n+1,(k+1)} = \mathbf{C}^{n+1,(k)} + \delta \mathbf{C}^{(k)}$$

检查 $\|\delta \mathbf{C}^{(k)}\|$ 是否小于容差，若是则停止

3. 最终解为 $\mathbf{C}^{n+1} = \mathbf{C}^{n+1,(k+1)}$

6. Stability and Accuracy Analysis

本方法采用线性有限元进行空间离散, Backward Euler 方法进行时间离散, 并使用 Newton 方法求解每个时间步的非线性系统。其稳定性与精度分析如下:

时间离散的稳定性: Backward Euler 是 A-稳定方法 (见 Dahlquist 定理), 对所有满足 Lipschitz 条件的非线性项在任意 $\Delta t > 0$ 下无条件稳定, 适合处理 Fisher-Kolmogorov 方程中的非线性反应项 $\alpha c(1 - c)$ 。

空间离散的收敛性: 若解析解 $c \in H^2(\Omega)$, 线性有限元逼近满足 Céa 引理, 误差估计为:

$$\|c - c_h\|_{H^1} \leq Ch\|c\|_{H^2}, \quad \|c - c_h\|_{L^2} \leq Ch^2\|c\|_{H^2}.$$

我们使用线性有限元 (P1 元素) 构造试探空间 $V_h \subset H^1(\Omega)$, 该方法在网格均匀、单元形状良好的前提下具备稳定性。根据有限元逼近理论, 当解析解具有足够的正则性时, 其 L^2 误差满足 $\mathcal{O}(h^2)$, H^1 误差满足 $\mathcal{O}(h)$, 其中 h 为网格尺度。

时间精度: Backward Euler 为一阶方法, 截断误差为 $\mathcal{O}(\Delta t)$, 整体时空误差为 $\mathcal{O}(\Delta t + h)$ 。

非线性求解收敛性: Fisher-Kolmogorov 方程中的反应项 $f(c) = \alpha c(1 - c)$ 具有一阶连续可导性且满足 Lipschitz 条件, 根据 Newton-Kantorovich 理论, Newton 方法在初始猜测足够接近解、Jacobian 可逆且源项满足 Lipschitz 条件时, 局部收敛速率为二阶。

7. Computational Aspects

在实际计算中, 为了高效求解 Fisher-Kolmogorov 方程, 采用以下策略:

非线性求解 (Newton 方法): 每个时间步通过 Newton 迭代求解非线性系统, 迭代过程需要多次装配残差与 Jacobian。通常设置最大迭代步数 (如 1000 次) 与残差容差 (如 10^{-3}) 以控制收敛。

线性求解器与 Preconditioner: 每次 Newton 迭代中, 线性系统 $J \cdot \delta = -r$ 使用预条件共轭梯度法 (CG) 解算。Preconditioner 使用 SSOR 预条件器提升迭代效率, 默认松弛因子为 1.0, 适用于对称正定的 Jacobian。

并行计算: 使用分布式内存并行库 MPI 配合 Trilinos 库进行矩阵装配与线性求解, 提升可扩展性。

8. Experiments

待写.....

补充：论文 [13] 部分解释与实际参数设置

2.2. Continuum model of prion-like spreading

朊病毒样疾病（如 AD、PD、ALS）的核心特征是错误折叠蛋白的“增长”与“空间扩散”。该研究使用一个经典的偏微分方程—Fisher-Kolmogorov 方程来描述这些疾病中的生物物理过程。

数学模型公式：

$$\frac{dc}{dt} = \nabla \cdot (\mathbf{D} \nabla c) + \alpha c(1 - c)$$

- $c(\mathbf{x}, t)$: 错误折叠蛋白的相对浓度，取值在 $[0, 1]$
- α : 蛋白“生长”或聚集的速率（反应项系数）
- \mathbf{D} : 扩散张量，控制蛋白的“传播路径与速度”

两种传播机制：

作者将扩散划分为两部分：

$$\mathbf{D} = d^{\text{ext}} \mathbf{I} + d^{\text{axn}} \mathbf{n} \otimes \mathbf{n}$$

- d^{ext} : 胞外扩散速率，蛋白通过脑间质空间弥散传播，各向同性
- d^{axn} : 轴突运输速率，蛋白沿神经纤维传输，各向异性
- \mathbf{n} : 神经纤维方向单位向量（来自 DTI 成像）
- $\mathbf{n} \otimes \mathbf{n}$: 构成了一个张量投影，使扩散沿轴突方向增强

因此，扩散张量 \mathbf{D} 不是常数矩阵，而是空间相关、方向依赖的张量场，这是该模型的关键生物物理创新。

3.3. Prion-like spreading in neurodegenerative disease

这段文字来自论文的结果部分，解释了图 10 至图 12 中关于不同神经退行性疾病中蛋白质聚集的模拟结果，展示了他们的数学模型如何精确再现临床观察到的病理传播路径。

四种疾病的模拟总览

Fig. 10 总结了四种神经退行性疾病中不同蛋白沉积的传播过程，白色箭头标示错误折叠蛋白如何从种子区域向其他脑区扩散：

疾病	蛋白类型	颜色
阿尔茨海默病 (AD)	β -淀粉样蛋白 ($A\beta$)	橙色
阿尔茨海默病 (AD)	tau 蛋白	蓝色
帕金森病 (PD)	α -synuclein	红色
肌萎缩侧索硬化症 (ALS)	TDP-43	绿色

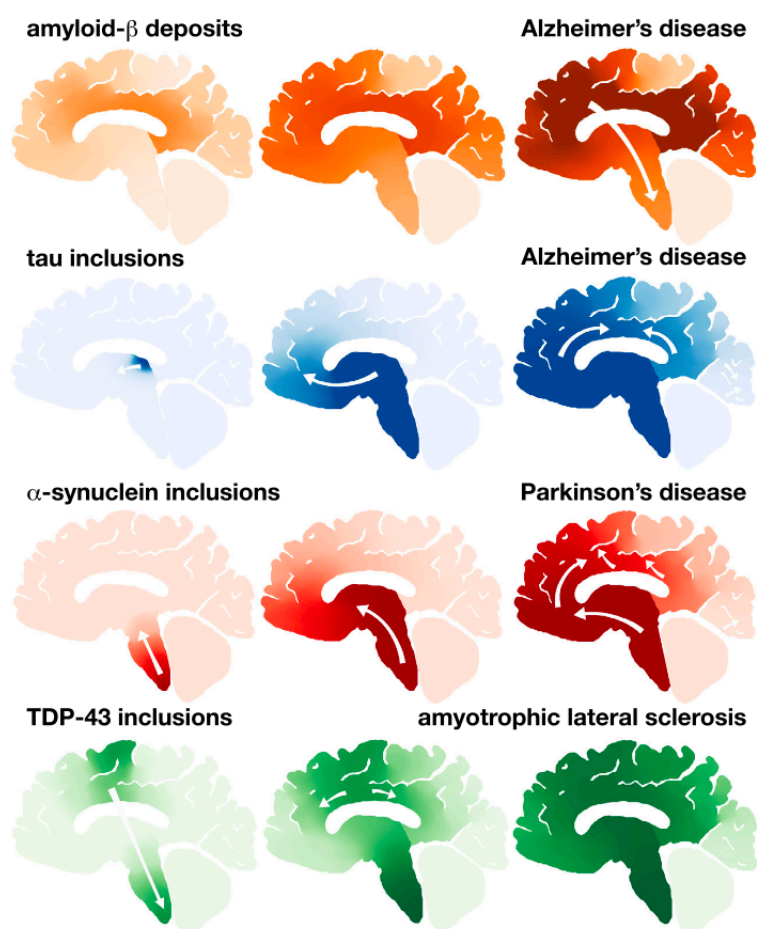


Fig. 10. Prion-like spreading across sagittal section. Amyloid- β deposits in Alzheimer's disease; tau inclusions in Alzheimer's disease; α -synuclein inclusions in Parkinson's disease; and TDP-43 inclusions in amyotrophic lateral sclerosis. Three stages of progressive neurodegeneration, from left to right, with white arrows indicating the putative spread of the lesions. Simulations display excellent agreement with the characteristic progression of specific proteinaceous lesions in neurodegenerative diseases over time inferred from post-mortem analyses of brains (Jucker and Walker, 2013) in Fig. 1.

Fig.5 神经纤维方向 \mathbf{n} : 来自解剖学信息

Fig.7 初始蛋白沉积区域（“种子区域”）：根据不同疾病设定

Fig.8 使用的参数（单位： $\text{cm}^2/\text{年}$ ）如下：

参数	灰质	白质
胞外扩散 d^{ext}	1.5	1.5
轴突扩散 d^{axn}	0.0	3.0
生长率 α	0.3	0.6

时间设置

$T = 40 \text{ years}$, 120 steps, $\Delta t = 1/3 \text{ year}$

传播路径与临床数据的吻合

这部分逐一比较了模拟结果与临床尸检研究观察到的蛋白沉积顺序，具体如下：

1. β -淀粉样蛋白 (AD)

起始于：新皮层，后传播到：旧皮层 (allocortex) \rightarrow 下丘脑等皮层下区域 (subcortex)

2. tau 蛋白 (AD)

起始于：蓝斑核 (locus coeruleus) 和内嗅皮层 (transentorhinal)，后传播至：整个新皮层网络

3. α -synuclein (PD)

起始于：延髓背侧运动核和嗅球，向上扩展到：脑干 \rightarrow 颞叶 \rightarrow 整个皮层

4. TDP-43 (ALS)

起始于：运动皮层、脑干运动核、脊髓前角，扩展至：额叶皮层 \rightarrow 顶叶皮层 \rightarrow 颞叶和海马

图 11 和图 12：三维验证

Fig.11 展示的是三维蛋白扩散过程，与图 10 中二维结果一致

图 12 展示多个视角（矢状、冠状、横断）和整脑浓度分布

补充：起始区域的选择

蛋白类型	起始区域	解剖位置说明	MNI 示例坐标	在右半球模型中可能位置
β -淀粉样蛋白 (AD)	新皮层 (Neocortex)	额、顶、颞、枕叶的灰质层	额叶: ($\pm 40, 30, 30$), 颞叶: ($\pm 60, -20, -10$)	额叶: X=40-80, Y=100-150, Z=50-110, 枕叶: X=20-60, Y=10-50, Z=30-80
	蓝斑核 (Locus Coeruleus)	脑桥背侧, 近第四脑室	($\pm 4, -30, -35$)	Y=50-70, Z=40-60 (脑干上部)
	内嗅皮层 (Transentorhinal Cortex)	颞叶内侧, 海马旁回前部	($\pm 20, -10, -30$)	X=20-40, Y=30-60, Z=30-60
α -Synuclein (PD)	延髓背侧运动核 (DMV)	延髓背侧, 迷走神经起始	($\pm 2, -40, -50$)	Y=50-80, Z=40-70 (脑干尾部)
	嗅球 (Olfactory Bulb)	额叶下方, 大腦前端	($\pm 10, 15, -15$)	X=10-20, Y>140, Z=20-40
	运动皮层 (Motor Cortex)	中央前回 (Brodmann 4 区)	($\pm 30, -20, 60$)	X=30-60, Y=80-120, Z>60
TDP-43 (ALS)	脑干运动核	脑干区域	III 动眼核: ($\pm 4, -20, -10$), VII 面神经核: ($\pm 8, -30, -25$), XII 舌下核: ($\pm 2, -35, -45$)	Y=40-90, Z=30-70
	脊髓前角	灰质前部, 颈段 (如 C5)	(0, 0, C5)	Y=10-40, Z=10-40
	胼胝体	中线附近连接两半球	X 10-20	Y=50-100, Z=50-80
白质结构	内囊	深部白质通路	-	X=30-50, Y=60-90, Z=40-70
	放射冠	从内囊向皮层辐射分布广泛	-	整个半球范围

表 1: 各类神经病理蛋白的起始区域解剖定位与可视化范围参考 (以右侧脑半球为例)

补充：使用 Gmsh 生成 volume cells

写一个 script.geo 脚本处理 STL：

```
Merge "brain-h3.0.stl";

// Define surface loop and volume
Surface Loop(1) = {1}; // or more surfaces if needed
Volume(1) = {1};
// Mesh settings
Mesh.CharacteristicLengthMin = 1;
Mesh.CharacteristicLengthMax = 2;
// 3D mesh
Mesh 3;
// Save to MSH version 2
Save "brain-h3.0.msh";
```

然后执行：

```
gmsh -3 script.geo -format msh2
```

因为 brain-h3.0.stl 文件中只有一个整体封闭表面（如一个封闭的脑结构），并且 Gmsh 成功识别为一个 surface，才可以这样写 Surface Loop(1) = 1.