# Continuous modelling by PDEs

## Numerical methods

Computational Biology

Adérito Araújo (alma@mat.uc.pt)
July 25, 2024

## Finite differences method (FDM)

Let us consider the reaction-diffusion equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + f(u), \quad x \in (0, L),\ t > 0,$$

with homogeneous Dirichlet boundary conditions

$$u(0, t) = u(L, t) = 0, \qquad t \geqslant 0$$

and initial condition

$$u(x, 0) = u_0(x), \qquad x \in [0, L].$$

The idea is of the FDM consists on obtaining an approximate solution for the initial boundary value problem replacing the derivatives in the equation by finite differences.

## Finite differences method: algorithm

1. Find a grid on the space domain

$$0 = x_1 < x_2 < \cdots < x_{N-1} < x_N = L$$

   such that $\Delta x = x_{i+1} - x_j$, $i = 1, ..., N-1$.

2. Write the equation on the grid points

$$\frac{\partial u}{\partial t}(x_i, t) = D\frac{\partial^2 u}{\partial x^2}(x_i, t) + f(u(x_i, t)), \quad i = 2, .., N-1,$$
$$u(x_1, t) = u(x_N, t) = 0,$$

## Finite differences method: algorithm

3. Discretize space derivatives using the finite difference formula (obtained by Taylor expansion)

$$\frac{\partial^2 u}{\partial x^2}(x_i, t) = \frac{u(x_{i+1}, t) - 2u(x_i, t) + u(x_{i-1}, t)}{\Delta x^2} - \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\eta_i, t),$$

   where $\eta_i \in (x_{i-1}, x_{i+1})$. Replacing on the equation and eliminating the error we obtain the system of $n$ ODEs

$$\frac{d\hat{u}_i}{dt}(t) = \frac{\hat{u}_{i+1}(t) - 2\hat{u}_i(t) + \hat{u}_{i-1}(t)}{\Delta x^2} + f(\hat{u}_i(t)),$$
$$\hat{u}_1(t) = \hat{u}_N(t) = 0,$$

   for $i = 2, .., N-1$, where $\hat{u}_i(t) \approx u(x_i, t)$

4. The system of ODEs may be solved by an ODE solver (like the Euler or Runge-Kutta solvers).

## Finite differences method: algorithm

4. Find a grid on time

$$0 = t^0 < t^2 < \cdots < t^m < \cdots$$

   such that $\Delta t = t^{m+1} - t^m$, for all $m$.

5. Discretize time using the finite difference formula

$$\frac{d\hat{u}_i}{dt}(t^m) = \frac{\hat{u}_i(t^{m+1}) - \hat{u}_i(t^m)}{\Delta t} - \frac{\Delta t}{2}\frac{d^2\hat{u}_i}{dt^2}(\tau^m)$$

   where $\tau^m \in (t^m, t^{m+1})$. Replacing on the equation and eliminating the error we obtain the algebraic system

$$\begin{aligned} &U_i^{m+1} = U_i^m + r\left(U_{i+1}^m - 2U_i^m + U_{i-1}^m\right) + \Delta t f(U_i^m), \\ &U_1^m = U_N^m = 0, \end{aligned}$$

   for $i = 2, .., N-1$, $m = 0, 1, ...$, where $r = \frac{D\Delta t}{\Delta x^2}$ and $U_i^m \approx \hat{u}_i(t^m) \approx u(x_i, t^m)$.

## Finite differences method: matrix form

The algebraic system may be written in the matrix form as

$$U^{m+1} = U^m + rAU^m + \Delta t F(U_i^m)$$

where $U^m = [U_2^m, U_3^m, \ldots, U_{N-2}^m, U_{N-1}^m]^T$, and

$$A = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}$$

## Finite differences method: algorithm

6. Obtain the approximate solutions $U^{m+1}$, $m = 0, 1, ...$, solving the algebraic system

$$U^{m+1} = U^m + rAU^m + \Delta t F(U_i^m).$$

In alternative, we may consider the semi-implicit method (more stable)

$$U^{m+1} = U^m + rAU^{m+1} + \Delta t F(U_i^m),$$

and obtain the approximate solutions $U^{m+1}$, $m = 0, 1, ...$, solving the linear system of equations

$$BU^{m+1} = U^m + \Delta t F(U_i^m),$$

where $B = I - rA$ and $I$ is the identity matrix of order $N - 2$.

## Computational exercise: diffusion equation

Exercise 3.7: Explore the Matlab codes `Difusion_1D_explicit.m` and `Difusion_2D_explicit.m`.

1. Observe the diffusive behaviour of the solutions.
2. Use the Matlab command `spy` to see the structure of the diffusive matrix.
3. Explore the behaviour of the numerical solution obtained with the explicit methods when you increase $\Delta t$. Try to find a value $r$ such that, for $\Delta t > r\dfrac{\Delta x^2}{D}$ the numerical solution starts to oscillate.

## Computational exercise: reaction-diffusion equations

Exercise 3.8: Starting with the Matlab code `Difusion_1D_explicit.m`, obtain a new file `Fisher_explicit.m` to solve the following problem.

1. Using the FDM, obtain the numerical solution of the Fisher equation in 1D

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + \mu u(1-u)$$

   with $0 \leqslant x \leqslant 2$, $\mu = 0.1$, $D = 0.01$, with the boundary conditions $u(0,t) = u(2,t) = 0$ and initial conditions $u(x,0) = 1$, $x \in [0.8, 1.2]$, and $u(x,0) = 0$ elsewhere.
2. Explore the dynamical behaviour of the system for different values of $\mu$.

## Computational exercise: Fisher equation

Exercise 3.9: Explore the Matlab code `Fisher_explicit.m`.

1. Start with $\mu = 0$ (just diffusion) and see the behaviour of the solution when $\mu > 0$.
2. Explore the behaviour of the numerical solution for different values of $D$ and $\mu$.
3. Try to see numerically a relation between $D$ and $\mu$ such that for $t$ large enough the steady state solution will vanish.

## Homework #12: system of reaction-diffusion equations

Exercise 3.10: Starting with the Matlab code `Fischer_explicit.m`, obtain a new file `Brusselator_explicit.m` to solve the following problem.

1. Using the FDM, obtain the numerical solution of the dumped Brusselator in 1D

$$\begin{aligned}\frac{\partial u}{\partial t} &= D\frac{\partial^2 u}{\partial x^2} + \alpha + u^2 v - (\beta+1)u \\ \frac{\partial v}{\partial t} &= D\frac{\partial^2 v}{\partial x^2} + \beta u - u^2 v,\end{aligned}$$

   with $0 \leqslant x \leqslant 1$, $\alpha = 1$, $\beta = 3$, $D = 1/50$, with the boundary conditions $u(0,t) = u(1,t) = 1$, $v(0,t) = v(1,t) = 3$ and initial conditions $u(x,0) = 1 + \sin(2\pi x)$, $v(x,0) = 3$.
2. Explore the dynamical behaviour of the system.
3. For $D = 0$ (no diffusion), the Brusselator model is a system of ODEs. Try to obtain the plots presented at Exercise 2.7 (Homework #6) for $\alpha = 1$ and $\beta = 1, 2, 3$.

## A final question

Why are there animals with spotted bodies and striped tails, but none with striped bodies and spotted tails?

More in:

https://turing-pattern-project.sites.sheffield.ac.uk/turing-patterns