Case Studies in Scientific Computing

M.Sc. in Mathematical Modelling & Scientific Computing

Hilary Term 2025

Numerical Simulation of Electrochemical Experiments

Introduction

The basic idea of an electrochemical experiment is that a potential is applied to an electrode in an electrochemical cell and this causes electron transfer to take place and a current to flow. Based on the current, which can be measured, the properties of the chemical system can be inferred.

Mathematical Model

The concentration of a chemical in the electrochemical cell can be modelled (in dimensionless variables) by the 1D diffusion equation

$$\frac{\partial a}{\partial t} = \frac{\partial^2 a}{\partial x^2}, \quad x, t > 0$$

with appropriate boundary and initial conditions.

The quantity of interest is the current

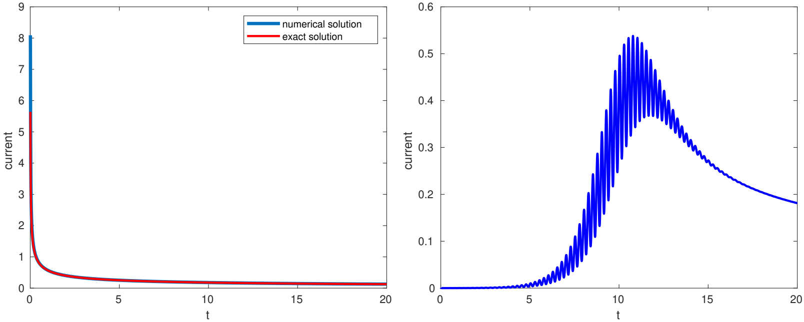
$$I(t) = \left.\frac{\partial a}{\partial x}\right|_{x=0} .$$

Boundary Conditions

The boundary condition at $x = 0$ depends on how the potential is applied:

- constant potential (homogeneous Dirichlet condition);
- linear sweep (mixed boundary condition);
- linear sweep with sine wave superimposed (mixed boundary condition).

Examples of Currents



Techniques

- Solution of 1D PDEs using finite differences;
- Theoretical solution using similarity solutions and Laplace transforms;
- Integral equations;
- Parameter recovery (inverse problem).

Population Growth in a Closed System

The Model

The Volterra model for population growth in a closed system is

$$\frac{\mathrm{d}p}{\mathrm{d}t} = ap - bp^2 - cp \int_0^t p(x)\mathrm{d}x$$

where

- $a > 0$ is the birthrate coefficient;
- $b > 0$ is the crowding coefficient;
- $c > 0$ is the toxicity coefficient.

The term $cp \int_0^t p(x)\mathrm{d}x$ represents the effect of toxin accumulation on the species.

Dimensionless Form of Model

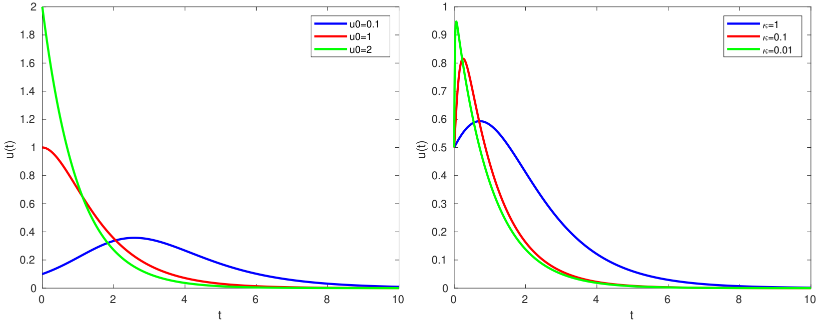
The dimensionless form of the problem is

$$\kappa \frac{\mathrm{d}u}{\mathrm{d}t} = u - u^2 - u \int_0^t u(x)\mathrm{d}x$$

for $t > 0$ with $u(0) = u_0$.

If $\kappa \ll 1$ then we have a stiff problem and we need a small time-step (at least initially).

Examples



Techniques

- ODE solvers;
- Quadrature;
- Adaptive time-stepping;
- Analytical techniques.

Image Colourisation

Problem Statement

Given a greyscale image and some colour information, how can we reconstruct a full colour image?

The idea is to use the fact that pixels which are close together are likely to be similar in colour and those with similar greyscale values are likely to be similar in colour.

Details

We write

$$(\mathrm{red})_i \;=\; \sum_{j=1}^{m} a_j \phi\left(\frac{\|z_i - x_j\|}{\sigma_2}\right)\phi\left(\frac{|g(z_i)-g(x_j)|^p}{\sigma_1}\right),$$

where

- $\phi(r)$ is a radial basis function;
- z_i is a point in the domain, $1 \le i \le n$;
- x_j is a point where colour information is known, $1 \le j \le m \ll n$;
- $g(x_i)$ represents greyscale information at x_i;
- the coefficients a_j are to be found by a minimisation process.

Idea of the Project

Build a GUI (graphical user interface) to solve the problem!

Use this to investigate how different parameters affect the recovery process.

Example

(top left = original, top right = greyscale,
bottom left = greyscale + some colour, bottom right = recovered image)

Numerical Solution of the Cahn-Hilliard Equation

The Cahn-Hilliard Equation

The Cahn-Hilliard equation is:

$$\begin{aligned} \frac{\partial c}{\partial t} - \nabla \cdot (B(c)\nabla w) &= 0 \, , \\ w - \frac{1}{\epsilon}\Phi'(c) + \epsilon \nabla^2 c &= 0 \, , \end{aligned}$$

in $\Omega \times (0, T)$ with boundary conditions

$$\frac{\partial c}{\partial n} = B \frac{\partial w}{\partial n} = 0 \, ,$$

on $\partial\Omega \times (0, T)$ and an initial condition for c.

Typically $\Phi(c) = (1 - c^2)^2/4$ and $B(c) = 1$ or $B(c) = (1 - c^2)_+$.

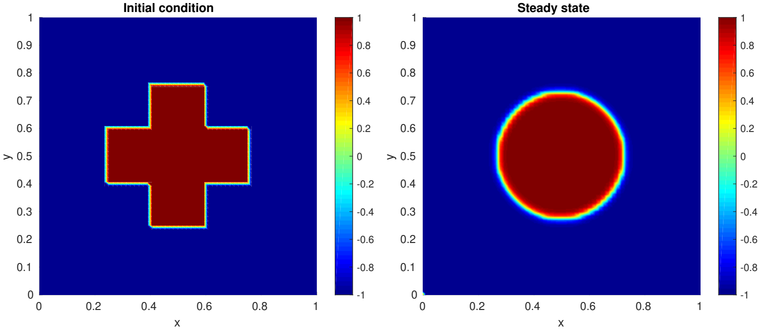
Steady States

The spatially and temporally homogeneous steady states are $c = \pm 1$ and the spatially dependent steady states consist of regions where $c = 1$ and where $c = -1$. The size of the interface between these regions is controlled by ϵ.

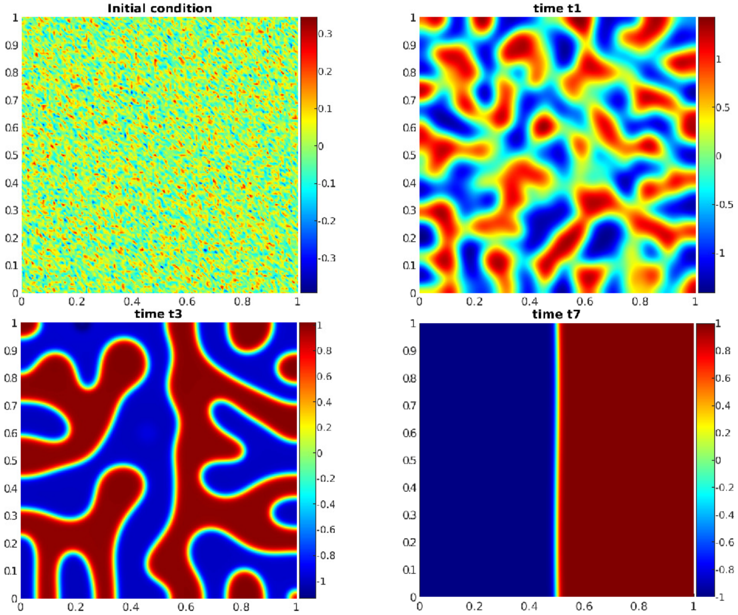
Idea of the project: investigate different time-stepping methods (with finite differences in space) to solve the problem accurately and efficiently.

Use one of these methods to solve the Cahn-Hilliard equation for image recovery.

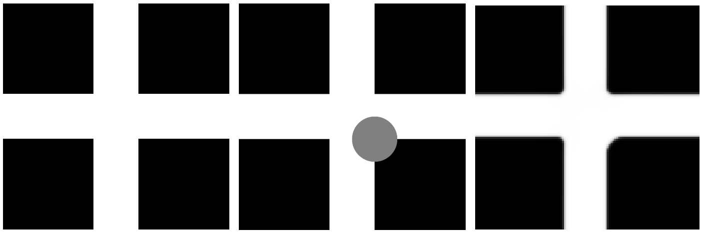
Example 1



Example 2



Example 3



Numerical Solution of Problems in Pattern Formation

The Schnakenberg Model

The general form of a (dimensional) reaction diffusion system is

$$\begin{aligned} \frac{\partial A}{\partial t} &= F(A,B) + D_A \nabla^2 A \,, \\ \frac{\partial B}{\partial t} &= G(A,B) + D_B \nabla^2 B \,, \end{aligned}$$

in $\Omega \times (0, T)$ with boundary conditions

$$\frac{\partial A}{\partial n} = \frac{\partial B}{\partial n} = 0 \,,$$

on $\partial\Omega \times (0, T)$ and with initial conditions for A and B on $\bar{\Omega}$.

In the Schnakenberg model we set

$$\begin{aligned} F(A,B) &= k_1 - k_2 A + k_3 A^2 B \,, \\ G(A,B) &= k_4 - k_3 A^2 B \,. \end{aligned}$$

Analysis

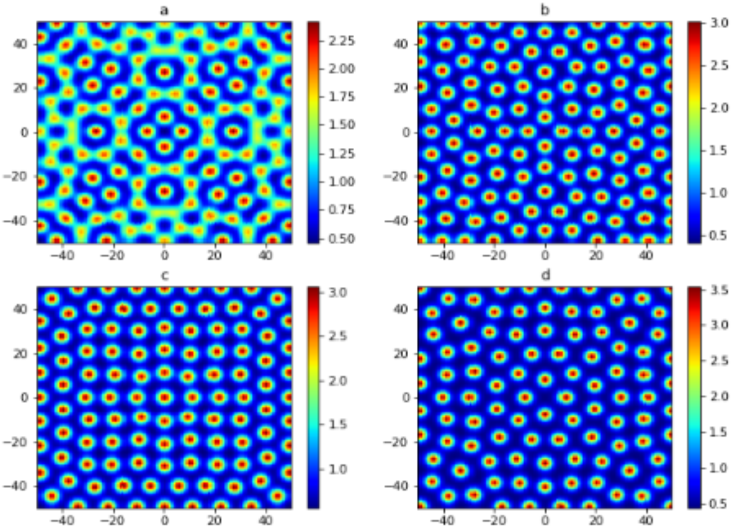
The project will begin with some mathematical analysis to non-dimensionalise the equations and find conditions under which patterns will form. This will follow similar methods to the mathematical biology course.

Numerical Solution

We will then look at numerical solution of the model using finite differences in space and a variety of timestepping schemes.

We will consider solutions in one space dimension to start with and then move on to two space dimensions.

Example

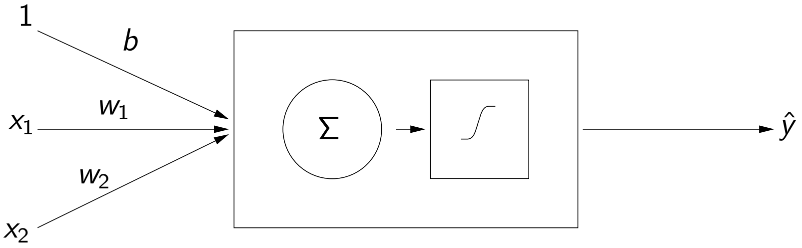


a
b
c
d
40
20
0
−20
−40
−40 −20 0 20 40
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
3.0
2.5
2.0
1.5
1.0
0.5
3.5

Numerical Solution of Differential Equations Using Neural Networks

Units in a Neural Network

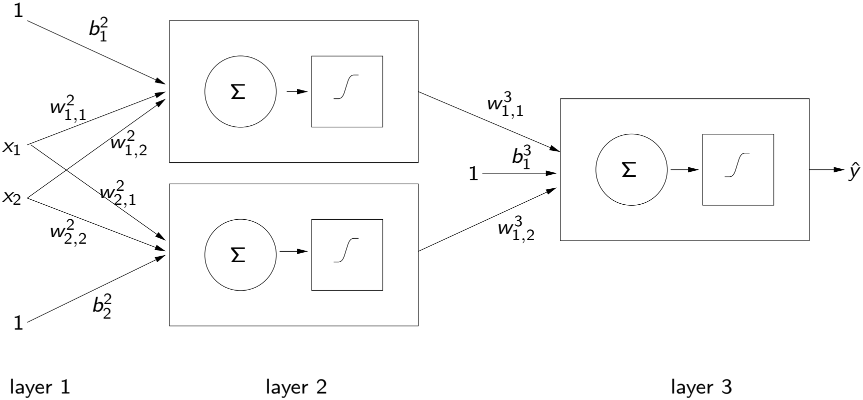
Inputs are 1, x_1, and x_2 and the output is $\hat{y}$. We also have weights w_1 and w_2 and a bias b.



In each unit we compute a weighted sum $z = b + w_1x_1 + w_2x_2$ and then compute a nonlinear function of z (often a sigmoid, e.g. $a = \sigma(z)$).

Neural Network

We can combine these units together to get a feedforward neural network.



Neural Network

We can increase the depth of the network by adding more layers, and the width of the network by adding more units in each layer.

The challenge is to optimise over the weights and biases.

Relation to ODEs

Suppose we want to solve the differential equation

$$\begin{aligned} \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} &= f(x, y)\ , \quad x \in (0, 1) \\ y(0) &= a\ , \\ y(1) &= b\ . \end{aligned}$$

We choose a set of values x_k at which to train the network. Suppose we have a single hidden layer with m units, then for each x_k we compute

$$\begin{aligned} z_i^2 &= b_i^2 + w_i^2 x_k \\ a_i^2 &= \sigma(z_i^2) \end{aligned}$$

for $i = 1, \ldots, m$. Then we compute

$$\hat{y}(x_k) = \sum_{i=1}^{m} w_i^3 a_i^2 + b^3\ .$$

Relation to ODEs

Having computed the $\hat{y}(x_k)$ we can compute a residual type error

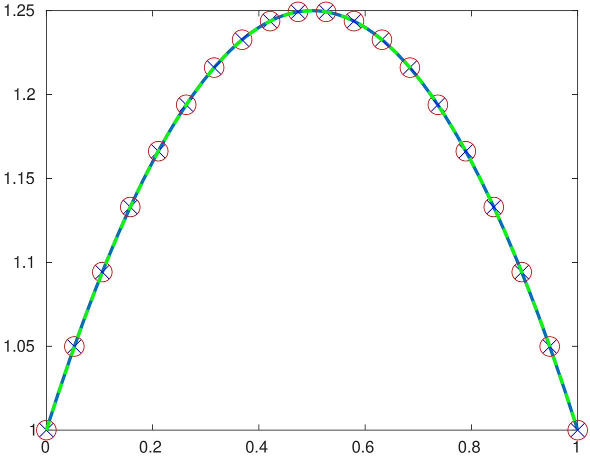
$$L = \sum_k \left(\frac{\mathrm{d}^2 \hat{y}(x_k)}{\mathrm{d}x^2} - f(x_k, \hat{y}(x_k)) \right)^2 + \gamma_1 (\hat{y}(0) - a)^2 + \gamma_2 (\hat{y}(1) - b)^2 .$$

The aim is then to minimise L over the parameters $\boldsymbol{\theta} = (w_1^2, \ldots, w_m^2, b_1^2, \ldots, b_m^2, w_1^3, \ldots, w_m^3, b^3)$.

This can be done using gradient descent or stochastic gradient descent, but in either case we need to know $\partial L / \partial \theta_i$. This is called back-propagation.

Example 1 (ODE)

$L = 1.3093 \times 10^{-6}$



Example 2 (PDE)

$$L = 9.1519 \times 10^{-6}$$