Mathematical Modelling for Biosciences

Daniel Galvis
Kyle Wedgwood
10th July 2019

Centre for Biomedical Modelling and Analysis
Course content

Motivation/development of mathematic models

Simulation of models

Phase space analysis and predictions

Write/understand code/equations
Today session

Population Models

Chemical Reactions

Neural Models

Motivation for mathematical modelling

The mathematics of modelling

\[ \dot{x} = f(x, t) \]
What is a mathematical model?

Mathematical Model: A mathematical representation of real-world systems. We will focus on "rates of changes" of things such as populations, chemical concentrations, etc.
How are they developed?

Bottom-up models – building up complex systems from simple experimental results (gene pathway models/HH model)

Top-down models - finding the simplest, necessary parts to model a higher level system (predator-prey models)
How are they developed?

Mechanistic – more complex, try to “stay true” the underlying biological processes that lead to the behaviour of interest

Phenomenological - simple models that try to capture a behaviour, generally in the simplest way possible

Hodgkin-Huxley Model

Extracellular space

Intracellular space

Fitzhugh-Nagumo Model

\[ \dot{V} = V - \frac{V^3}{3} - W + I \]

\[ \dot{W} = 0.08(V + 0.7 - 0.8W) \]
Why do we use them?

- Integrate many pieces of information
- Quick and cheap to simulate (compared to lab work)
- Make Predictions
- Bridge Scales
- Simplification (mechanistic -> phenomenological)
Limitations

- Curse of dimensionality
- Parameter uncertainty
- Computational cost
- Noise
- Models can be wrong
The modelling paradigm

“All models are wrong but some are useful”
- George Box
The modelling paradigm

A good model should be

• A faithful representation of the real world system

• Able to be simulated/analysed

• Constructed at an appropriate scale

• Able to provide testable predictions

• Flexible and adaptable

• Simple (but not too simple)
The modelling paradigm

“Everything should be made as simple as possible but no simpler”

- Albert Einstein
The mathematics

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
Xppaut

- Software for simulating and analysing mathematical models
- Written by Bard Ermentrout
  [www.math.pitt.edu/~bard/xpp/xpp.html](http://www.math.pitt.edu/~bard/xpp/xpp.html)
- Essentially a graphical front end for a library of numerical routines to simulate equations
- Reads in text files encoding models
- Easy to use with either mouse or keyboard
- Can be used with python
- Now available on iPad!
Predator prey model

• Simple phenomenological model of one predator species and one prey species

• Overly simplistic, since it ignores all other organisms, but gives general overview of predatory dynamics

• Has also found use in neural modelling, opinion modelling and game theory
Lotka-Volterra model

• Originally a model of chemical reactions, but was shortly afterwards applied to population modelling

• Let \( x \) denote the amount of grass and \( y \) denote the number of rabbits

• Note that these are both continuous (rather than discrete variables). This makes sense if the populations of both are big enough

• In the model grass grows exponentially quickly with rate \( \alpha \) and rabbits die exponentially quickly with rate \( \delta \)

• Grass gets eaten by the rabbits at a rate \( \beta \), and the rabbits breed at a rate, \( \gamma \), dependent on the availability of grass
Lotka-Volterra equations

- Grass grows: \( \dot{x} = \alpha x \)
- Rabbits eat grass: \( \dot{x} = -\beta xy \)
- Rabbits breed after eating: \( \dot{y} = \gamma xy \)
- Rabbits die: \( \dot{y} = -\delta y \)
- Full model

\[
\begin{align*}
\dot{x} &= \alpha x - \beta xy \\
\dot{y} &= \gamma xy - \delta y
\end{align*}
\]

- Behaviour dependent on parameters and initial conditions
Simulating the Lotka-Volterra model
Simulating the Lotka-Volterra model
The Brusselator

Belousov-Zhabotinsky Reaction
The Brusselator

- 6 chemical species

\[ A, B, D, E, X, Y \]

- Well mixed environment

- Constant supply of \( A, B \), immediate clearing of \( D, E \)

\[
\begin{align*}
(1) \quad & A \xrightarrow{k_1} X \\
(2) \quad & 2X + Y \xrightarrow{k_2} 3X \\
(3) \quad & B + X \xrightarrow{k_3} Y + D \\
(4) \quad & X \xrightarrow{k_4} E
\end{align*}
\]
Law of Mass Action

• Simple model of chemical reactions

• Rate of reaction is directly proportional to the product of the masses of the reactants

\[
\begin{align*}
(1) \quad A & \xrightarrow{k_1} X \quad \dot{X} = k_1 A \\
(2) \quad 2X + Y & \xrightarrow{k_2} 3X \quad \dot{X} = k_2 X^2 Y, \quad \dot{Y} = -k_2 X^2 Y \\
(3) \quad B + X & \xrightarrow{k_3} Y + D \quad \dot{X} = -k_3 BX, \quad \dot{Y} = k_3 BX \\
(4) \quad X & \xrightarrow{k_4} E \quad \dot{X} = -k_4 X
\end{align*}
\]
Our first model

\[
\begin{align*}
\dot{X} &= k_1 A \\
\dot{X} &= k_2 X^2 Y \\
\dot{X} &= -k_3 BX \\
\dot{X} &= -k_4 X \\
\end{align*}
\]

\[
\begin{align*}
\dot{Y} &= -k_2 X^2 Y \\
\dot{Y} &= k_3 BX \\
\end{align*}
\]
Neural modelling

- Essentially, almost all biophysical models of neurons are based on the 1952 formalism by Hodgkin and Huxley

- Originally developed to describe the behaviour of the squid giant axon
Hodgkin-Huxley model
Hodgkin-Huxley model

Extracellular space

Intracellular space
Morris-Lecar model

- Variant of the Hodgkin-Huxley model which replaces sodium current with a calcium current
- Hodgkin-Huxley model has 4 state variables, Morris-Lecar has only 2
- Originally a model of:
Morris-Lecar model

Extracellular space

Intracellular space

$I_{Ca}$, $I_{K}$, $I_{L}$

$R_{Ca}$, $R_{K}$, $R_{L}$

$V_{Ca}$, $V_{K}$, $V_{L}$

$C$, $V$
Morris-Lecar model

\[ CV = Q \]

\[ CV = \dot{Q} = I \]
Morris-Lecar model

Extracellular space

C

\[ I = I_{Ca} + I_{K} + I_{L} + I_{app} \]
Calcium current

\[ I_{Ca} = g_{Ca} m(V) (V_{Ca} - V) \]

prop. open channels
conductance Nernst potential
Full model
Extra Slides
Neural networks
Drug delivery