University of Dundee

MA32011

MA32011

Philip Murray

2026-01-19

Table of contents

Preface

Welcome to the module MA32011 Dynamical systems.

My name is Philip Murray and I am the module lead.

How to contact me?

- email: pmurray@dundee.ac.uk
- office: G11, Fulton Building
- Teams: PM me

Lecture notes

You can find lecture notes for the module on this page. If you would like a pdf this can be easily generated by clicking on the pdf link of the webpage. I will occasionally edit/update the notes as we proceed through lectures. If you spot any errors, typos or omissions please let me know.

Reading

Nonlinear dynamics and chaos, Steven Strogatz Strogatz (2001) Mathematical Biology I, Murray (2002)

Python codes

I have provided Python codes for most of the figures in the notes (you can unfold code section by clicking 'Code'). Note that the Python code does not appear in the pdf.

Many of you have taken the Introduction to Programming module at Level 2 and have therefore some experience using Python. I strongly encourage you to use the provided codes as a tool to

play around with numerical solutions of the various models that we will be working on. The codes should run as standalone Python codes.

> **Note**
>
> To access Python on Uni machines:
>
> 1. Launch Anaconda from AppsAnywhere
> 2. When a folder opens, double click on *Spyder*.
> 3. Paste a code from lecture notes into the editor on the left-hand side.
> 4. Click on the green arrow to run the code.
> 5. The plots should appear in the plots tab on the right-hand side.
> 6. Experiment with the code. When you change a model parameter, does the solution change in an expected way?

I have also provided some examples of how to use Python as a symbolic calculator. This uses a Python library called *sympy* and is quite similar to Maple.

Assessment

- Final exam (80 %)
- 2 class tests (8 % each), Week 7 and 11
- 4 quizes (1 % each), Week 2,4,6 and 9

Plan

Table 1: Projected delivery

Week	Up to Section	Tutorial sheet	Assessment
1		1	
2		1	Quiz 1
3		2	
4		2	Quiz 2
5		3	
6		3	Quiz 3
7		4	Test 1
8		4	
9		5	Quiz 4 4
10		5	

Week	Up to Section	Tutorial sheet	Assessment
11		Test 2	

References

1 Introduction

Dynamical systems

The goal of this module is to provide an introduction to dynamical systems. We will introduce key mathematical concepts and explore examples from physics and biology.

To begin with we introduce some key terminology.

1.1 Discrete v continuous time

1.1.1 Differential equations

Let t be a continuous variable and $x = x(t)$. Consider the ordinary differential equation (ODE)

$$\dot{x} = rx(1 - x). \tag{1.1}$$

$\dot{x}$ is used to denote the time derivative, i.e.

$$\dot{x} := \frac{dx}{dt}.$$

Similarly, the second order derivative is represented by

$$\ddot{x} := \frac{d^2x}{dt^2}.$$

Many of the second order ODE problems that we will examine originate from Newton's Second Law, i.e.

$$m\ddot{x} = F(x)$$

where $x(t)$ represents the position of a particle at time, t, m represents a constant particle mass and F a resultant force.

Consider the case in which F represents a linear restoring force, i.e.

$$F(x) = -kx.$$

The equation of motion can be written as

$$\ddot{x} = -\mu x,$$

where $\mu = -k/m$.

1.1.2 Difference equations

Suppose that n is a discrete variable. Let y_n represent a dependent variable at iteration n. Consider the difference equation

$$y_{n+1} = r y_n (1 - y_n),$$

where $r \in \mathfrak{R}$.

See this link for exploration of model solutions.

1.1.3 Key questions to ask of nonlinear dynamical system

- do solutions exist? If so are they unique?
- Is there an explicit solution?
- Can we qualitatively describe solution behaviour?
- How do the solutions depend on the model parameters?
- Are their critical values of parameters where solution behaviour changes?

1.2 Autonomous v nonautonomous ODEs

For an autonomous system, the update does not explicitly depend on the independent variable. Equation 1.1 is autonomous. But

$$\dot{x} = rx(1 - x + t)$$

is nonautonomous (because of the explicit time dependence on the right-hand side).

1.3 Linear v Nonlinear

Linear systems satisfy a linear supposition principle: a sum of solutions is itself a solution. In general, this property does not hold for nonlinear systems.

In linear dynamical systems, the dynamics are a function of linear sums of the dependent variables. Hence

```
#| '!! shinylive warning !!': |
#|   shinylive does not work in self-contained HTML documents.
#|   Please set `embed-resources: false` in your metadata.
#| standalone: true
#| viewerHeight: 800
```

```
from shiny import App, Inputs, Outputs, Session, render, ui
from shiny import reactive

import numpy as np
from pathlib import Path
import matplotlib.pyplot as plt
from scipy.integrate import odeint

app_ui = ui.page_fluid(

    ui.input_slider(id="u0",label="u_0",min=0.0,max=1.0,value=0.5,step=0.01),
    ui.input_slider(id="T",label="Number of iterations",min=0.0,max=60.0,value=20.0,step=1.0)
    ui.input_slider(id="r_range",label="r zoomed",min=0.0,max=4.0,value=[0.0,4.0],step=0.001)
    ui.input_slider(id="u_range",label="u zoomed",min=0.0,max=1.0,value=[0.0,1.0],step=0.01)
    ui.input_slider(id="r",label="r",min=0.0,max=5.0,value=0.1,step=0.001),
)

def server(input, output, session):
    @output
    @render.plot
    def plot():
        fig, ax = plt.subplots(3,1,figsize=(12, 4))
        #ax.set_ylim([-2, 2])
        # Filter fata


        r=float(input.r())
        u0=float(input.u0())
        T=int(input.T())
        r_min=float(input.r_range()[0])
        r_max=float(input.r_range()[1])
        u_min=float(input.u_range()[0])
        u_max=float(input.u_range()[1])


        # Define rhs of logistic map
        def logistic_map(y,t,r):
          rhs=r*y*(1-y)
          return rhs

        def DiscreteSol(rhs_pop_model,y_0,t,r):
            y=np.zeros_like(t,dtype=float)
            y[0]=y_0
            for i in t:
```

$$\dot{x} = -x$$

is a linear ordinary differential equation (ODE). But

$$\dot{x} = -x^2$$

is nonlinear.

1.4 Quantitative v qualitative solutions

You are likely used to solving problems in which an explicit solution can be found. For example, consider the ODE

$$\dot{x} = -kx, \quad x(0) = x_0$$

where $k, x_0 \in \Re^+$.

We can integrate and express the solution as

$$x(t) = x_0 e^{-kt}$$

Using the explicit solution we can then answer questions about its behaviour. For example, let's say we want to find the time, t^*, at which the solution is half it's maximum. Hence

$$x(t*) = x_0/2 \implies t^* = \frac{\ln 2}{k}.$$

However, in the study of nonlinear systems, most problems will not have an explicit solution. For example, consider the nonlinear ODE

$$\dot{x} = -\frac{k\sin(x) + \sqrt{x}}{1+x}, \quad x(0) = x_0$$

where $0 < x_0 < \pi$.

I cannot integrate this equation in order to find solutions in terms of standard functions. Hence I cannot *quantitatively* describe the solution. However, I can identify that

$$\dot{x} < 0, \ \forall \ 0 < x < \pi.$$

Hence the solution will decrease in value from the given initial condition. This is an example of a *qualitative* analysis.

1.5 Representing solutions

It is useful to introduce some nomenclature to describe the solutions of a dynamical system. Consider an ODE

$$\dot{x} = f(x), \quad x(0) = x_0,$$

where f is a prescribed function and x_0 is an initial condition. At some time t we define

- phase space
- *phase point* - value of the solution at given time point
- *vector field* - the derivative of the solution, i.e. f. The sign represents whether the solution is increasing/decreasing whilst the magnitude represents the rate of change.
- *trajectory* - a line in phase space that traces out a solution as time evolves
- *phase portrait* - collection of trajectories (i.e. solutions with different initial conditions)

1.6 Fixed points and their stability

Many of the dynamical systems that we will study will be nonlinear. Hence it will not be possible to compute exact solutions.

The behaviour of dynamical systems can often be understood by considering the fixed points, i.e. values of the dependent variables at which the dynamics are at steady state.

Stability analyses are used to investigate the dynamics of perturbations about the steady state.

1.7 Uniqueness and existence

We will restrict ourselves to problems in which the vector fields are sufficiently well behaved such that unique solutions exist. However, problems can be identified where solutions do not exists or where multiple solutions exist.

1.8 Nondimensionalisation

In real world problems, variables and parameters typically have units (e.g. time - seconds, Force - Newtons etc.). We can nondimensionalise problems by defining rescaled variables. This process can be used to justify simplifications to models and to reduce the number of parameters.

1.9 Numerical solutions

Numerical solutions are used to numerically compute approximate solutions to problems. The simplest example of a numerical method in a dynamical system is the forward Euler method. Suppose we want to study the ODE

$$\dot{x} = f(x), \quad x(0) = x_0, \quad 0 < t < T.$$

Discretise the independent variable t by defining t=0, t,2 t,...,T.

Approximate the time derivative

$$\dot{x} = \frac{dx}{dt} \sim \frac{x(t + \Delta t) - x(t)}{\Delta t}$$

Hence the solution at time $t + \Delta t$ can be approximated by

$$x(t + \Delta t) = x(t) + \Delta t f(x(t)).$$

Given an initial condition $x(0) = a$ we can compute the approximate solution at time Δt.

Numerical solutions provide a a very useful way to explore solution behaviour. However, they describe the quantitative behaviour of a solution for a particular initial condition and set of parameter values.

Part I

1D flows

2 Flows on the line

First order in time.

Let $x = x(t)$.

$$\dot{x} = f(x). \tag{2.1}$$

f is smooth and real valued. autonomous. Nonlinear.

2.1 Geometric

Example 2.1. Let $x(t)$. Consider the ODE

$$\dot{x} = \sin x.$$

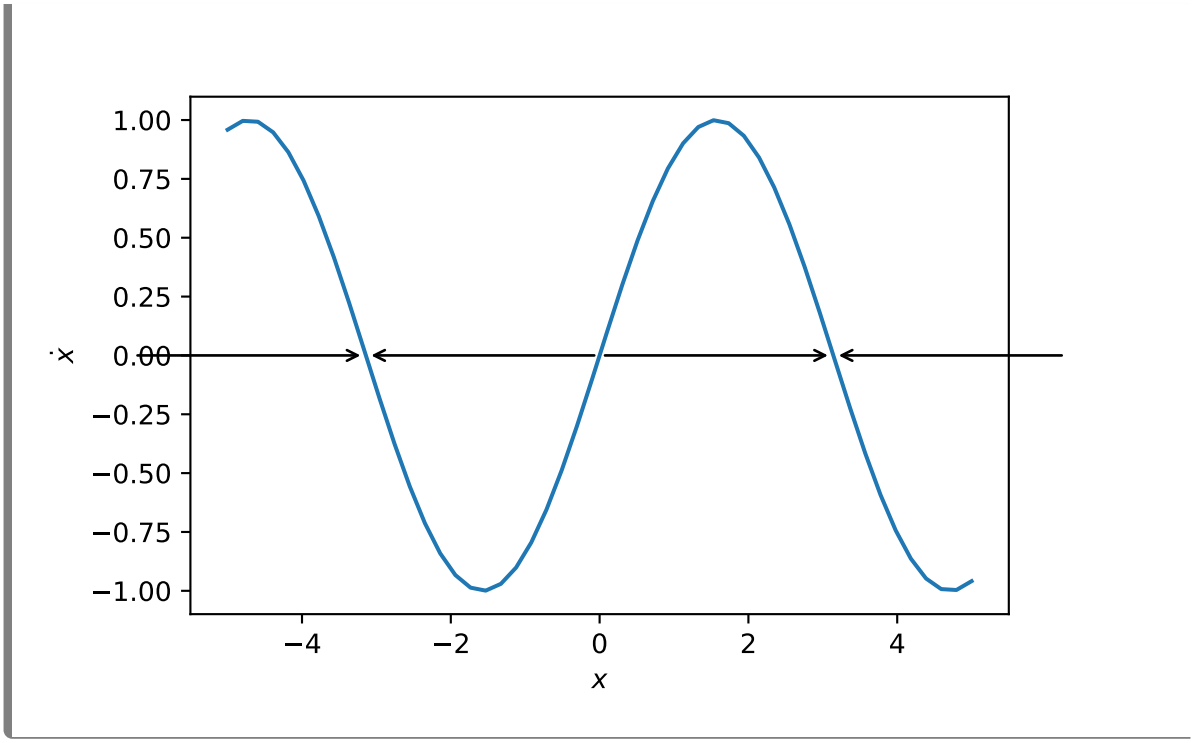
For the initial condition $x(0) = \pi/4$, describe solution behaviour as $t \to \infty$.

Solution

An implicit solution is

$$t = -\ln|\csc x + \cot x| + C,$$

where C is an integration constant.



2.2 Fixed points and stability

2.2.1 Fixed points

Let $x = x^*$ be a fixed point of Equation 2.1. At $x = x^*$

$$\dot{x} = 0 \implies f(x^*) = 0.$$

There are a number of interpretations of x^*:

- roots of f (algebraic)
- stagnation points of the flow (topological)

Corollary 1

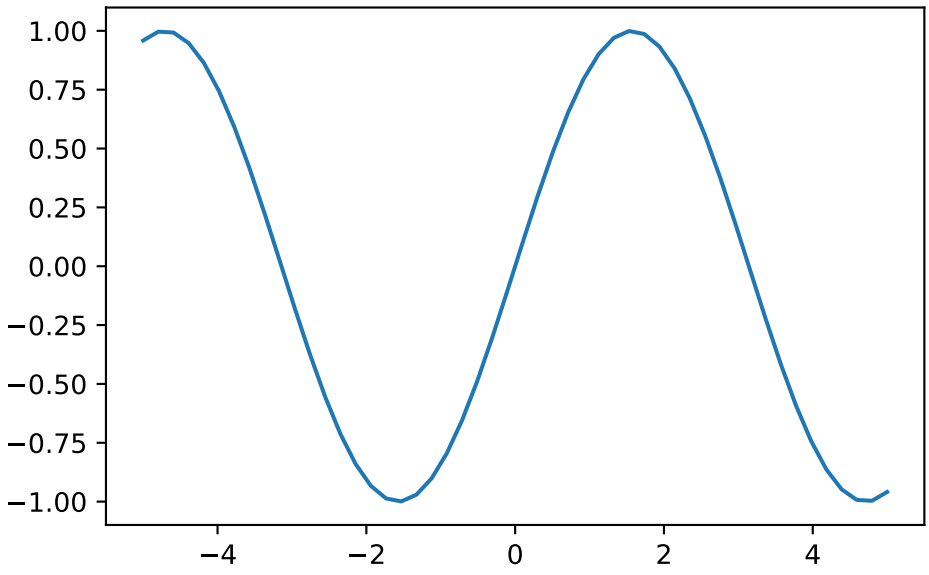
Any trajectory initialised at a fixed point remains there for all t.

Example 2.2. Find all the fixed points of

$$\dot{x} = x^2 - 1,$$

Solution

The fixed points are $x^* = \pm 1$.



2.2.2 Linear stability analysis

2.2.2.1 A change of dependent variable

To perform a linear stability analysis we make the change of variables

$$x(t) = x^* + \hat{x}(t)$$

where the new dependent variable, $\hat{x}(t)$, is a perturbation about the steady state.

The time derivative on the left-hand side of **?@eq-gensinglespecodeagain** transforms to

$$\dot{x} = \frac{d}{dt}(x^*) + \frac{d}{dt}(\hat{x}(t)) = \dot{\hat{x}}.$$

Hence **?@eq-gensinglespecodeagain** transforms to

$$\dot{\hat{x}} = f(x^* + \hat{x}(t)).$$

2.2.2.2 Taylor expansion and a linear system

Employing the Taylor expansion on the right-hand side of **?@eq-gensinglespecodeagain** and making the assumption that perturbations are small

$$\dot{\hat{x}} = f(x^*) + f'(x^*)\hat{x}(t) + f''(x^*)\hat{x}^2(t) + h.o.t.$$

Noting that

$$f(x^*) = 0$$

and retaining linear terms yields

$$\dot{\hat{x}} = f'(x^*)\hat{x}(t)$$

with solution

$$\hat{x}(t) = \eta e^{f'(x^*)t}$$

where η is some initial perturbation about the steady-state.

2.2.2.3 A condition for linear stability

When $f'(x^*) > 0$ the perturbation grows exponentially fast and the steady-state is unstable. When $f'(x^*) < 0$ the perturbation decays exponentially fast and the steady-state is stable.

Example 2.3. Determine the linear stability of the fixed points of

$$\dot{x} = x^2 - 1.$$

Example 2.4. What can be said about the stability of the fixed points of the following ODEs:
1.

$$\dot{x} = -x^3.$$

2.

$$\dot{x} = x^3.$$

2.3 Validity of linear classification

Hyperbolic FP - eigenvalues nonzero Hartman Grobman - if system has hyperbolic FP, classification of nonlinear system determined by linear classification non-hypberbolic FP - need to consider higher order terms.

2.4 Case study: population dynamics

Let $N = N(t)$. The logistic model of population growth, due to Verhulst, takes the form

$$\dot{N} = rN(t)\left(1 - \frac{N(t)}{K}\right), \tag{2.2}$$

where r is the linear growth rate and K is carrying capacity. We consider both $r, K \in \Re^+$.

Questions to ask of such a model are: what type of biologically realistic solutions does it possess? Are there steady-states? If so, are they stable or unstable? Are there bifurcations in solutions?

2.4.0.1 Numerical solutions

In Figure 2.1 we present numerical solutions of equation using different initial conditions. Note the limiting behaviour of solutions as $t \to \infty$. In Figure 2.1 it is clear that even though some solutions are initialised at $N_0 = 0.1$, much closer to $N^* = 0$ than $N^* = K$, they tend to the limit $N = K$. Why do solutions not tend to $N^* = 0$?

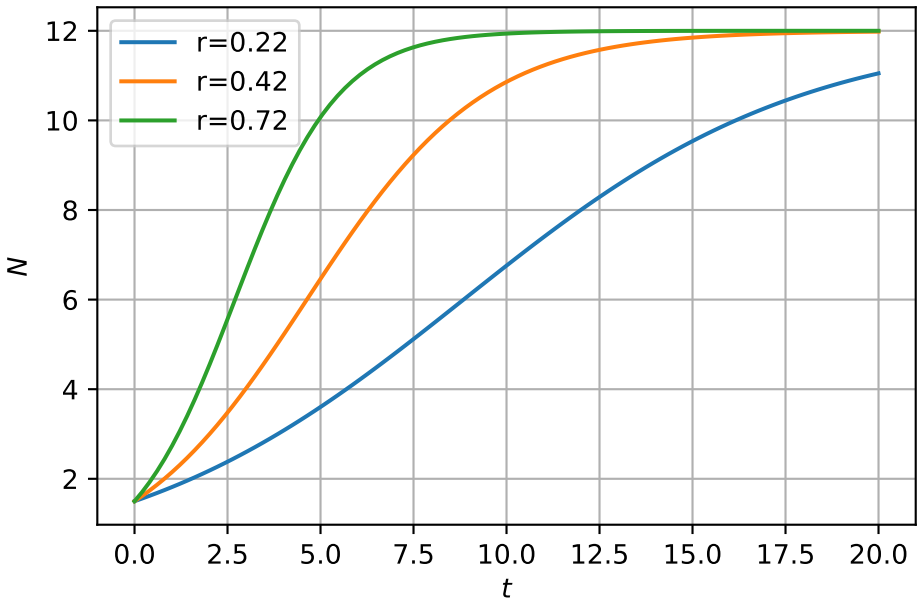


Figure 2.1: Numerical solution of the logistic growth model

2.4.0.2 Dimensional analysis and nondimensionalisation

N represents the population density and has units of one over area (say $1/m^2$) and t has units of time (say, seconds, s). Hence the left-hand side of Equation 2.2 has units of $1/(m^2 s)$. The first term on the right-hand side of Equation 2.2 is rN. N has units $1/m^2$ hence the parameter r must have units of $1/s$ for dimensional consistency. This is consistent as r represents the linear growth rate.

The second term has the form rN^2/K. Given the chosen units for r and N, the parameter K must have dimensions $1/m^2$. Again, this is consistent as K is a carrying capacity (i.e. it has units of population density).

We define the nondimensionalised variables

$$n = \frac{N}{\tilde{N}} \qquad \tau = \frac{t}{\tilde{T}}$$

where $\tilde{N}$ and $\tilde{T}$ are constants that have units of population density and time, respectively. Hence Equation 2.2 transforms, upon change of variables, to

$$\frac{\tilde{N}}{\tilde{T}}\frac{dn}{d\tau} = r\tilde{N}n(1 - \frac{n\tilde{N}}{K}).$$

In the case of the logistic equation there is only one time scale and density scale in the problem, hence we choose

$$\tilde{T} = \frac{1}{r} \quad and \quad \tilde{N} = K$$

and the dimensionless model is

$$\frac{dn}{d\tau} = n(1 - n) \tag{2.3}$$

Note that we can retrieve the original equation by rescaling and calculating $N = \tilde{N}n$ and $t = \tilde{T}\tau$.

2.4.1 Steady states and linear stability

Steady states satisfy

$$n^*(1 - n^*) = 0.$$

Hence

$$n^* = 0, \qquad n^* = 1.$$

To determine linear stability we compute

$$H'(n) = (1 - 2n).$$

When $n = n^* = 0$ we obtain

$$H'(n) = 1.$$

Hence the origin is an unstable steady-state.

At the steady-state $n^* = 1$

$$H'(n^*) = -1$$

hence $n^* = 1$ is linearly stable.

Note that the linear stability analysis can explain the observations regarding the numeric solutions presented in Figure 2.1.

2.4.2 Graphical analysis

In Figure 2.2 we plot the right-hand side of Equation 2.3. We can qualitatively describe model solutions by considering the arrow along the x axis. Suppose we consider an initial condition with $0 < n_0 < 1$. Using the graph of $H(n)$, $dn/d\tau$ is positive, hence n increases as a function of time until $n(\tau) \to 1$.

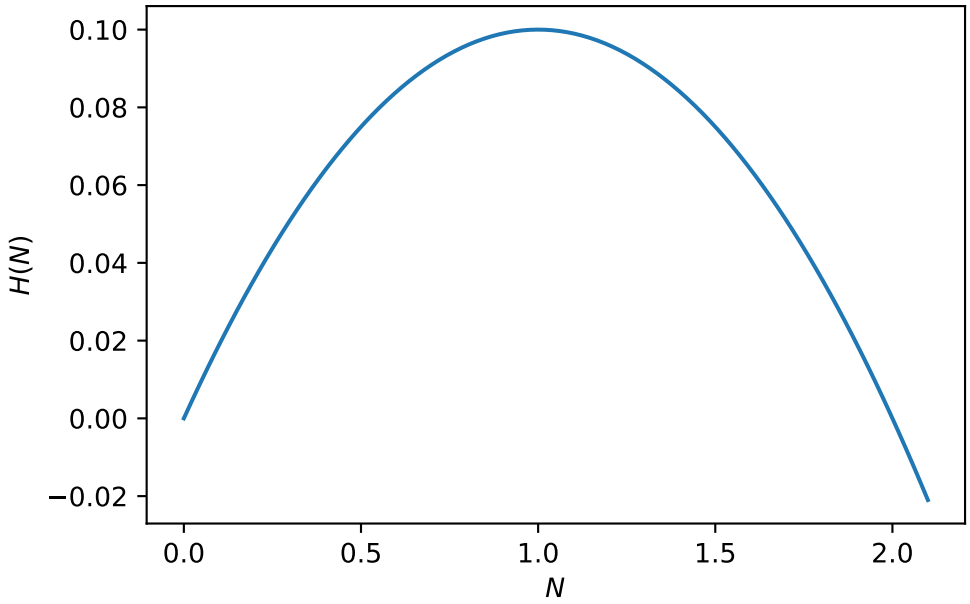


Figure 2.2: Right-hand side of the logistic ODE

2.4.3 An exact solution

Separation of variables yields

$$\int \frac{dN}{N(1-\frac{N}{K})} = r \int dt.$$

Using partial fractions

$$\int \frac{dN}{N} + \frac{1}{K} \int \frac{dN}{1-\frac{N}{K}} = r \int dt.$$

Integration yields

$$\ln N - \ln\left(1 - \frac{N}{K}\right) = \ln \frac{N}{1-\frac{N}{K}} = rt + C.$$

Hence

$$N = \frac{De^{rt}}{1 + \frac{D}{K}e^{rt}}$$

Given an initial condition $N(0) = N_0$, we obtain

$$N(t) = \frac{N_0 K e^{rt}}{K + N_0(e^{rt} - 1)}$$

2.4.3.1 Qualitative analysis of the exact solution

As $t \to \infty$, $N \to K$. At $t = 0$, $N = N_0$ and that for small $N_0 \ll K$ the initial growth phase is exponential, i.e.

$$N(t) \sim N_0 e^{rt} \quad N_0 \ll K, t \ll \frac{1}{r}.$$

Note that in almost all the models that we will consider the above method is not an usually an option as the ODE is not explicitly integrable.

2.5 Existence and uniqueness

Consider the ODE

$$\dot{x} = f(x), \quad x(a) = b.$$

There exists a unique solution on a local interval $x \in [c, d]$ with $c < x_0 < d$ given that f is bounded and Lipschitz continuous on $[c, d]$.

Example 2.5. Show that

$$\dot{x} = x^{\frac{1}{2}}, \quad x(0) = 0$$

has an infinite family of solutions.

2.6 Impossibility of oscillations

2.7 Potential flows

Consider the ODE

$$\dot{x} = f(x).$$

Suppose that

$$f(x) = -\frac{dV(x)}{dx}.$$

Now consider

$$\dot{V}.$$

Applying the chain rule

$$\dot{V} = \frac{dV}{dx}\dot{x} = -(\frac{dV}{dx})^2 \leq 0.$$

Hence for a potential flow V is never increasing.

2.8 Numerical solutions

2.9 References

Part II

Appendices

3 Python

3.1 Symbolic calculations

Symbolic calculations ahve been performed using the Python library Sympy.

This library comes with tutorials.

You are encouraged to familiarise yourself with the syntax by working through some of the tutorial examples provided at the links above.

Many of the calculations that we do throughout the course involve solving systems of algebraic equations

3.2 Numerical solution of difference equations

Difference equations have been solved using a for loop. Routinse have been written to solve either single or coupled system of difference equaitons.

3.3 Numerical integration of ODEs

Throughout the notes systems of ODEs have been integrated using the Scipy function odeint.

3.4 Plotting

Line graphs are plotted using the Python library Matplotlib.

Murray, J. D. 2002. *Mathematical Biology i: An Introduction*. Springer.
Strogatz, Steven H. 2001. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (Studies in Nonlinearity)*. Vol. 1. Westview press.