MA30287: Mathematics of Planet Earth

Philippe H. Trinh

2024-04-16

Table of contents

# Preface

Welcome to to the 2023-24 delivery of [MA30287 Maths of Planet Earth](https://www.bath.ac.uk/catalogues/2023-2024/ma/MA30287.html) at the University of Bath.

Here is a picture that represents the course.

|  |
| --- |
| Figure 1: This picture was produced by [Abstruse Goose](https://abstrusegoose.com/275) and has the title “This is how scientists see the world”. |

## Lectures and office hours

**Lectures** take place at the following times and locations:

* Tuesdays 9:15 in 6W 1.2
* Wednesdays 11:15 in 6W 1.2
* Thursdays 15:15 in 8W 3.22

**Office hours:** You will be able to find me for an office hour in 4W 2.18 on Thursdays (following the lecture). Typically it is best to set this up, beforehand, by email appointment.

## Coursework and examinations

Your final mark will be 25% coursework and 75% final exam.

Details of the coursework will be released in Week 7 and it will be due in Week 10[[1]](#footnote-27).

## Resources

In general, you will have access to a few kinds of resources:

1. The [Moodle portal](https://moodle.bath.ac.uk/course/view.php?id=60108) will be the main organisation portal.
2. Lecture notes, coursework, and other resources will be found in an online format and will be linked on Moodle.
3. Coding will be done via Noteable, accessed via the Moodle website.

Naturally, because this is a relatively new module at Bath, there will be a fair amount of activity as we settle the material over the semester. Whenever we complete a lecture note (i.e. a ‘chapter’), we will use a box like this to indicate when the material was covered and in which lecture:

|  |
| --- |
| **2023-24 note**  The material in this note was covered in Lecture XX. |

Hopefully by the time the module ends, every relevant chapter will have such a note. This allows you to judge what material has been ‘finalised’.

## Prerequisites

This course officially requires MA20221 (modeling and dynamical systems) or XX20231 (mathematical and statistical methods for the life sciences).

It is designed to be somewhat stand-alone in the sense that applied mathematical techniques learned in other courses will be introduced in some capacity. Such techniques will involve:

* Solutions of ordinary differential equations (MA10230 and MA20220).
* Multivariable calculus, partial differentiation, and multiple integrals (MA10230 and MA10236); some review/introduction of concepts from MA20223.
* Dynamical systems, stability, phase planes (MA20221, MA30060).
* Numerical methods in Python (MA10276).

Whenever possible, I have isolated such reviews/introductions and these can be found in the *Mathematical methods* section of these notes.

## Resources

This course is designed around the following sources:

* Sustainable energy – without the hot air by (MacKay 2009)
* Mathematics & Climate by (Kaper and Engler 2013)
* Mathematical Geoscience by (Fowler 2011)
* [A gentle introduction to numerical simulations with Python](https://hplgit.github.io/prog4comp/)

# Lecture plan

A lecture plan can be found at [this link](https://docs.google.com/spreadsheets/d/1vbFh8m_SmNg15Bwn8YkjktpgffGf19gW4p4jHZl4GbY/edit?usp=shari).

# 1. Conservation laws and constitutive laws

In the first chapter of (Fowler 2011), there is a concise introduction to the different categories of techniques and approaches that you might use when doing mathematical modelling in the real world. Some of these ideas will be introduced to you in this course.

Here, we provide a brief intro to the highlights, involving the use of conservation laws (and PDEs) and also the concept of non-dimensionalisation (which you would have encountered previously), studied in [Chapter 2](#sec-intro-nondim).

|  |
| --- |
| **Vectors and PDEs**  Vectors and PDEs is not a prerequisite for this course, but naturally in studying anything related to the physical real world, we must discuss partial differential equations. The hope is that the necessary theory for PDEs will be presented to you as this course evolves, so that it can appreciated by both newcomers and experienced readers. |

Conservation laws can be expressed as mathematical equations that represent the idea that some quantity is conserved. In processes governing the planet, these might correspond to conservation of heat, of water, of air, of momentum, etc.

In [Chapter 3](#sec-basic-energy), we will develop the simplest possible model governing the temperature on the surface of the Earth. It is a conservation equation for energy and is zero-dimensional (does not involve time and does not involve spatial variation).

## 1.1 Derivation of the 1D heat equation

In order to demonstrate some of the basic principles of this course, let us demonstrate the derivation of the heat equation. We are interested in modelling the heat in a volume, , which, for the sake of concreteness is given by a long cylinder with its axis along . We assume that the side walls of the cylinder are insulated and the temperature only varies along the direction.

At any point along this rod, the internal heat is given by , where is the density of the material (kg/m3), is the specific heat capacity (J/(kg K)), and is the temperature (K). Therefore, the heat energy along any segment in the rod is calculated from

|  |
| --- |
| **Internal heat energy** |

If the heat changes, then the rate of change of heat energy is given the time derivative of the above quantity. By conservation of energy, any change of the internal energy must be equal to the inflow or outflow of heat at the ends, or . We therefore write for the flux (or flow) of heat.

We need a **constitutive law** that dictates how energy is exchanged at the boundaries. Based on intuition, it is sensible to assume that the flow of heat proceeds from hot to cold. For example, hot air rises towards cool air; or heat from a hot mug of tea flows and diffuses outwards into a cold room. Therefore, we write this as

|  |
| --- |
| **Fourier’s law**  Fourier’s law in 1D specifies that the heat flux is given by |

This is known as Fourier’s law. The quantity is the thermal conductivity, and its units are W/(m K). Because a Watt is a Joule/s, you can also see that the units of are J/(m K s). The quantity is the flux, and you can verify that it is given in units of J/(m2 s).

Therefore by energy conservation, we have

$$
\frac{\mathrm{d}}{\mathrm{d}t} \int\_a^b \rho c T \, \mathrm{d}x = {\color{blue}q(x = a, t) - q(x = b, t)},
$$

i.e. the change in internal heat is equal to the flow through the ends. Note that we have added-in the heat flux due to assuming positive refers to heat moving from left-to-right. Conversely, we subtract it away from . We can alternatively write this as

$$
\frac{\mathrm{d}}{\mathrm{d}t} \int\_a^b \rho c T \, \mathrm{d}x = {\color{blue} -\int\_a^b \frac{\partial q}{\partial x} \, \mathrm{d}x}.
$$

Substitution Fourier’s law, we can then write

Because the above integral identity needs to be true for all possible values of and , then it must be true everywhere (this is sometimes referred to as the *du Bois-Reymond lemma* or the *bump lemma*. Therefore we are left with the classic heat equation.

|  |
| --- |
| **Heat equation** |

In order to produce a sensible physical solution, partial differential equations are typically supplemented by initial conditions and boundary conditions. The **initial condition** prescribes the state of the function at some initial time, typically . **Boundary conditions** prescribe how the function behaves on the boundary of its domain, which in this case is and . An example might be

|  |
| --- |
| **Initial conditions (IC) and boundary conditions (BC)** |

which expresses, respectively, that the temperature starts from a constant temperature, , and where the ends of the rod are kept at temperature and .

### 1.1.1 Steady states and long-time behaviours

|  |
| --- |
| **2024 note**  This was a new addition in 2024. |

When we refer to a **steady-state** solution, we are typically referring to a time-independent solution.

**Definition 1.1 (Steady-state solutions)** Given an evolving system described by a functio, say , defined on some spatial domain and with , the steady-state solution refers to time-independent solutions with

One can envisage that as the system evolves with , it reaches a state that is independent of time. However, not all systems will approach a steady state. Moreover, not all steady states are stable or attractive (and might never be reacheable in a real-life experiment).

For the case of heat flow, such a steady-state solution would be . In this case,

Therefore, for the heatflow along a segment of length with left boundary held at and right boundary held at we have

## 1.2 Deriving the 1D transport (continuity) equation

|  |
| --- |
| **2024 note**  This was a new addition in 2024. |

Consider the mass transport of some substance with density , immersed in a fluid, along a one-dimensional line in . If desired, you may consider the substance as existing in three-dimensional space, and propagating along the -direction, with its behaviour independent of and . Initially, when considered at time , the mass of the substance between two points, , and , is:

As time increases the particles of the substance will move due to the fluid moving; at the same time, the fluid volume which is initially contained in will also move. We want to find how the mass of the blob changes in time, and hence consider the quantity

The above considers the intrinsic rate of change of the function within the integrand, but then adds the extra mass due to the right boundary shifting rightwards () and subtracts the mass due to the left boundary shifting rightwards (). It is known as the [Leibniz integral rule](https://en.wikipedia.org/wiki/Leibniz_integral_rule). We can thus write this within the integral as

However, the quantity

represents the velocity of the fluid (which for the moment we assume to be a known and provided quantity). Therefore we can write the mass change as

This result, which explains how to pass a derivative through an integral express mass of a substance within a flow is known as the *Reynolds Transport Theorem*. We have just derived it in 1D.

|  |
| --- |
| **Reynolds Transport Theorem**  Let be some quantity (such as density) that is advected along a one-dimensional line in due to a fluid with velocity . Then |

We may now consider the substance being transported along the -direction. If there is no interior creation or destruction of the source, then by conservation of mass, it must be the case that

Thus, again since the above integral identity applies to all possible values of and , it must be the case that the integrand is zero. Thus we conclude with the so-called tranport equation.

|  |
| --- |
| **One-dimensional transport equation**  The transport of a substance described by advected along a one-dimensional line in due to a fluid with velocity is given by |

Again, we must consider the above problem in combination with potential initial conditions and boundary conditions. For instance, we might specify that the substance begins from some initial state, say

The boundary conditions are more subtle. It is not always obvious what the boundary conditions should be on a problem.

Here is an example of a solution of such a problem.

|  |
| --- |
| **Linear advection equation with constant speed**  Solve the problem given by  where the velocity is constant. You may assume the initial condition is given by |

|  |
| --- |
| **Solution**  The general solution is given by  where is an arbitrary (differentiable) function. You can verify this via the chain rule, noting that  Applying the initial condition at gives and thus  You may plot this to discover that it is, as perhaps expected, a Gaussian profile moving to the right at constant speed (for ). |

# 2. Dimensional scaling analysis

Putting a mathematical model into non-dimensional form is fundamental (Fowler 2011). When we refer to a “back-of-the-envelope calculation”, we often mean simple algebraic calculations that still nevertheless provide enormous insight on problems. Such analyses are often based on dimensional analysis, which identifies the relationships between the different quantities involved in the problem.

## 2.1 Dimensional quantities

Every physical quantity, say Q, can be expressed as a product of a dimensional unit, denoted [Q], and a magnitude, say Q’. Thus we write

For example, if corresponds to the physical length in a problem, we might select or or . It is important to choose the dimensionalisation to suit the problem under consideration.

### 2.1.1 SI units

The International System (SI) of Base Units sets out a distinct selection of choices for dimensions in certain physical quantities. The seven fundamental dimensional units are

* [Length] = metre
* [Time] = seconds
* [Mass] = kilogram
* [Temperature](#temperature) = Kelvin
* [Electric current] = ampere
* [Light intensity] = candela
* [Material quantity] = mole

Dimensional units that can be expressed in terms of other fundamental units are known as *derived units*. For example:

* [Speed] = metre/second
* [Acceleration] = metre/second2
* [Force] = kilogram . metre/second2

## 2.2 Dimensional homogeneity and non-dimensionalisation

All terms in any equation must have the same dimensions. This is the principle of dimensional homogenuity. For example, Newton’s second law expresses the fact that

We can check, then, that the units do indeed match up on either side. Here, the RHS has units of [m] [x]/[t]2 or in SI units, kg . metres / seconds2. This indeed matches our previous given SI unit decomposition for force.

Notice in addition that the input to functions like and must be non-dimensional (or dimensionless).

The process of nondimensionalisation is then as follows. Given an equation, we know that each term must have the same dimension. Therefore, we can scale all the dependent and independent variables by dimensional constants in order to yield a non-dimensional equation.

Why this is an important tool is demonstrated by the below.

## 2.3 Returning to the heat equation

|  |
| --- |
| **2023-24 note**  The below has been cleaned up to match the notation on the 13 Feb 2023 delivery. |

Exact units are not relevant for dynamics, and it is instead the ratio of units that we care about. To apply this principle, let us non-dimensionalise the equation [Equation 1.1](#eq-heatintro). Let us consider the heat equation as given by the system

Note that the constant is known as the thermal diffusivity, and is given by

which were the units previously introduced.

We introduce typical scales for each of the variables. For example, we non-dimensionalise the temperature, distance, and time by setting

Remember that via the chain rule, we have that

Substitution into the equation now yields

Be sure to work out the above for yourself. For example, notice that the previous boundary at is now sent to . Now we see that the final equation is beautifully simple: At this point, we can identify a crucial non-dimensional parameter given by

Although this looks quite complicated, you can test that this parameter is indeed non-dimensional by looking up the units for thermal conductivity, , density , and specific heat , and verifying that indeed is non-dimensional.

## 2.4 Choice of units

It is important to remember that there are technically no wrong choices for the scales [x], [T], and [t], as long as they yield consistent balances. However, some choices are better than others for the context of the problem. Let us select:

This means our unit of length is “one pipe” (if this is indeed a pipe), and our unit of temperature is whatever the initial temperature was set to be. The choice of time is free at the moment, and therefore we can select time as

Now we see that the final equation is beautifully simple:

### 2.4.1 Interpretation

What is the whole point of the above exercise?

In the original problem, there were five parameters:

Imagine a situation where you are performing an analysis of heat spread in different pipes of different materials (characterised by ) and lengths (), with different initial and boundary temperatures. That is potentially a highly complicated parameter space to search (five-dimensional).

Rather than solving the problem again and again for each change in parameters, what the analysis yielding ([Equation 2.1](#eq-heat-nondim)) reveals is that the parameter space only needs to be two-dimensional, for parameters and . Therefore, by solving the problem at many different values of and , the solution space can be completely mapped.

Here is an example. Consider heat diffusion through common [brick](https://www.engineersedge.com/heat_transfer/thermal_diffusivity_table_13953.htm), which possesses a thermal diffusivity of

Therefore a brick that is about L = 10cm = 0.1m in size has a typical timescale of

The above choice of [t] is representative of the typical timescale you can expect for heat to diffuse through the brick. As you can see it’s very long.

In summary, we have gone from a system where we needed to consider five parameters to a system where we only need to specify two (essentially, the ratio of the initial heat to the boundary values). This is an enormous simplification.

Moreover, your analysis has further identified the key non-dimensional parameters that appear, including:

In this particular problem, you had the freedom to choose to remove one parameter (the ratio of lengths), and time could be chosen to set .

# 3. Basic energy models

In this chapter, we introduce the simplest climate model to describe the temperature of the Earth. Despite its apparent simplicity, this basic climate model is the foundation of all further discussion. We will find that, according to the most basic model, the surface temperature of the Earth is expected to be much colder than it is actually; we conclude that the likely culprit is greenhouse gases.

Consider the Earth as a single averaged body. Our goal is to obtain an equation for the surface temperature, , by considering the energy recieved from the sun versus the energy the Earth must emit into space. This kind of balance-equation procedure is used everywhere in mathematical modelling. In essence, we are looking to write down a precise statement of the following:

|  |
| --- |
| **Basic energy balance** |

where is the incoming energy from the sun and is the energy leaving the Earth. Note that the left hand-side (the internal energy) will depend on the temperature, .

## 3.1 The basic energy model

The basic model is derived as follows.

1. By considering the incoming radiation from the sun, obtain an estimate for the incoming energy, .
2. Use a constitutive law that indicates how much outgoing energy is produced by an object heated to some temperature.
3. Equate the change in internal energy to be equal to the difference between the above two items.

Note that the primary components of the global energy balance are radiative fluxes: we receive short-wave radiation (UV and visible light) from the Sun, and emit longwave radiation (infra-red) to space.

|  |
| --- |
| Figure 3.1: Radiation in the atmosphere |

First we consider incoming energy.

|  |
| --- |
| **Energy from the Sun**  First, note that the shortwave radiation (UV and radiation) recieved from the sun is (which we consider measured at a point near the planet).  If we consider only that radiation that is absorbed into the Earth, we have    where is the Earth’s radius.  In the above formula, we have multiplied the flux, , with the visible surface area, . There is an additional multiplication by where is the *planetary albedo*, which characterises amount of energy reflected due to the surface properties. Light surfaces like snow will have high albedo, , while darker surfaces like the ocean have smaller albedo, . The global average albedo is . |

Next we consider outgoing energy.

|  |
| --- |
| **Energy from the Earth**  We now wish to characterise the energy, , and in the case of Earth, this will correspond to longwave radiation (infra-red) emitted into space. All bodies characterised by a temperature, say , will emit radiation, . As a model, we can consider to be given by the [Stefan-Boltzmann law](https://en.wikipedia.org/wiki/Stefan-Boltzmann_law), which states that    where is the Stefan-Boltzmann constant.  Now although the Earth’s surface may emit radiation according to ([Equation 3.3](#eq-SBlaw)), some of this radiation will be absorbed by the atmosphere and reflected back. This is the **greenhouse effect**. As a consequence of the greenhouse gas, the surface temperature of the Earth, , will be larger than the effective emitting temperature, . For the moment, we model this as    where is a greenhouse gas factor, which depends on the properties of the atmosphere. |

Now combining the above equations, we have

which gives the incoming energy per unit time.

|  |
| --- |
| **Internal heat energy**  Due to this incoming energy, the Earth will cool or heat in response. We need to know how the internal temperature of an object responds to an input in energy. The general formula is  The key quantity is the experimentally determined, , which corresponds to the [specific heat capacity](https://en.wikipedia.org/wiki/Specific_heat_capacity%20of%20air). It is given in the SI units of , i.e. energy per unit mass per unit temperature. Note that this applied to a shell around the planet of thickness in the atmosphere, and so the mass is given by    where is the average density of the atmosphere. Let us imagine the increase in temperature, ,during an interval of time, . You can now verify that  $$ \Bigl[ (4 \pi R^2) d \rho \Bigr] c\_p (\Delta T) $$  returns the units of Joules—i.e. this is the internal energy produced during the time . We then have  $$ \Bigl[ (4 \pi R^2) d \rho \Bigr] c\_p \Delta T = \Delta t (E\_{in} - E\_{out}). $$ |

Putting in ([Equation 3.5](#eq-EinEout)) and taking the limit of , we finally have a heat equation for the Earth’s temperature.

|  |
| --- |
| **Zero-dimensional model for the surface temperature of the Earth**  The Earth’s temperature, as measured on a layer of thickness at the troposphere is given by the following ordinary differential equation (ODE) for :    where we have defined  as the heat capacity of the atmosphere. Above, the solar flux, Q is often taken to be W/m2, W/(m2K4), and is the greenhouse gas factor. |

The above equation [Equation 3.7](#eq-heateq-earth) is time-dependent, but we may consider that the surface temperature, either over long-time or in an averaged sense, is dictated by the steady-state (time-independent) solution. Setting to zero, we see that there is a unique steady-state given by

If we take , , , we then get

under the assumption that . That’s pretty cold! The actual average temperature is around .

The above back-of-the-envelope calculation seems to suggest that the parameter plays an important role in keeping the Earth warm enough for us to live on, and indeed the value of inferred by the above is roughly . Later on in the course, we will develop a more rigorous model to predict such a by studying the properties of the atmosphere.

## 3.2 The history of global warming

The history of global warming (and hence the estimation of ) is convoluted, but the origins can be considered as far back as the work of (Fourier 1827) and (Pouillet 1838), and is discussed in the work by (Van der Veen 2000).

|  |
| --- |
| Figure 3.2: An illustration of Fourier’s glass box |

## 3.3 Next in the study of energy balance models?

At this stage, the basic energy balance model studied in this chapter can be made more complex and realistic through many different extensions. For instance:

* Consider the effects of a non-constant and nonlinear albedo. Since the albedo depends on the material property (water vs. ice vs. land), we can incorporate a toy model for spatial variability by allowing the albedo to depend on temperature, .
* Consider the time-dependence. What happens if the system starts out of the steady state? Does it tend to relax to its equilibrium profile? Can we analyse the temporal properties of the solutions?
* Can we incorporate latitude (or longitudinal) dependence into the model. For instance, should the albedo and solar radiation be considered as functions of the latitude?
* Should we incorporate the effects of diffusion (heat spreading out) or convection (heat transported with flows)?
* To what extent should we consider the effects of the atmosphere on the absorption and reflection properties of the incoming (and outgoing) energy?

Many of these questions are much more involve, as they require differential or partial differential equations. Therefore in the next few chapters, we will discuss, in a very general way, the collection of applied and numerical mathematics techniques that can be used when studying problems in the physical sciences.

# 4. Problem class 1

|  |
| --- |
| **Abstraction**  It may seem strange to study examples from elementary physics in a course that is supposed to be about Planet Earth. But simple examples are the best ways to learn these important techniques. The full climate equations are often very involved. These toy models still nevertheless capture the spirit of what you must do when attacking any scientific problem. |

In this problem class, we will practice some concepts about non-dimensionalising in preparation for the Problem Set 1 in [Chapter 26](#sec-PS1). We will cover these two strategies in choosing scalings.

|  |
| --- |
| **Scaling principle 1**  Select the characteristic scales so that as many of the possible non-dimensional numbers, , are normalised. |

|  |
| --- |
| **Scaling principle 2**  Select characteristic scales so that no terms in the model diverge in the physical limit of interest. |

Here are the problems we shall do in the problem class.

## 4.1 Projectile motion

A projectile of mass (in kg) is launched vertically with initial velocity (in m/s) from a position (in m) above the surface. Thus the mass’s position, is governed by Newton’s second law (applied to the mass and the mass of the Earth) and the set of equations

where and is the radius of the Earth.

1. Non-dimensionalise the equation using arbitary length and time scales.
2. Identify the non-dimensional constants, .
3. Choose a length scale of and time scale of . Discuss the resultant equation and the interpretation of choosing these scales.
4. Does your above choice allow you to easily study the limit of ? If the limit can be taken, reduce the governing system to a simpler equation.
5. Does your choice in 3. allow you to easily study the limit of ? If not, choose an alternative choice of length and time scales and in that case, reduce the set of equations.

## 4.2 Terminal velocity

A ball of radius (in m) and uniform density (in kg/m3) falls in a viscous fluid. The fluid has density (in kg/m3) and viscosity (a measure of friction or resistance) (in kg/(m s)). The equation that governs the velocity is

1. Choose appropriate velocity and time scales to non-dimensionalise the equation so as to leave only a single non-dimensional number on the drag term (the last term on the right hand-side).
2. Define the non-dimensional parameter expressing a ratio between drag force and gravity force by the Stokes number (St) and confirm that it is
3. Comment on the two limits of and . Can the problem be reduced in these two limits? If so, reduce and solve.

# 5. Asymptotic approximations I

One powerful set of techniques for approximating solutions to equations is called *asymptotic analysis* or *perturbation theory*. To begin with, in this chapter, we introduce you to these techniques as a means to approximating the solutions to equations like:

when is a small parameter.

Soon, we apply the techniques to approximating solutions of differential equations.

## 5.1 A simple quadratic

|  |
| --- |
| **A singular quadratic**  Consider the solution of    where is a fixed and very small positive number, say . Forget that we know how to solve a quadratic equation: is it possible to develop a systematic approximation method? |

If , then . Moreover, if we substitute into the equation, then we see that the error is small and proportional to . It is natural to seek an approximation in powers of . We call this an . We write

Substitution into the equation yields

$$
\epsilon \Bigl(x\_0 + \epsilon x\_1 + \epsilon^2 x\_2 + \ldots\Bigr)^2 + \Bigl( x\_0 + \epsilon x\_1 + \epsilon^2 x\_2 + \ldots \Bigr) - 1 = 0.
$$

Expand and collect terms in powers of :

$$
\Bigl( x\_0 - 1 \Bigr) + \epsilon\Bigl(x\_1 + x\_0^2\Bigr) + \epsilon^2\Bigl(x\_2 + 2 x\_0 x\_1\Bigr) + \ldots = 0.
$$

Now we equate coefficients at each order in . This gives

We therefore have obtained the three-term approximation,

Clearly we could continue this process *ad infinitum* obtaining increasingly accurate approximations to one of the roots.

### 5.1.1 The singular root

But where has the other quadratic root gone?

The problem is that in considering to be small, we began by ignoring the leading term, . We effectively assumed that the equation was primarily *balanced* by setting the term with the term, and the sum of the two terms approximately equalling zero.

But if is large, then clearly our assumption that being small may not be necessarily true for it depends on how large is compared to . Note that if is large, then necessarily the last term, , is negligible in comparison. Therefore, in order for to balance , we see that must be of size .

Therefore this suggests that we should re-scale our solution as follows

Substitution into the original quadratic now yields

Now notice that expresses the correct balance in order to detect that missing root. Again we write

and attempt to solve order by order. Substitution into the equation yields

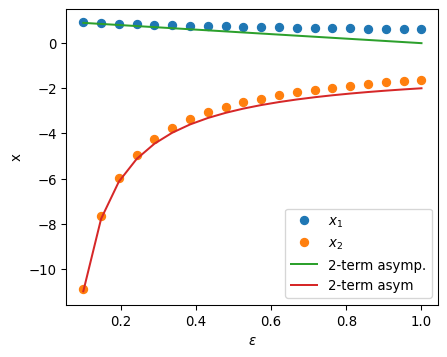
$$
\Bigl(X\_0 + \epsilon X\_1 + \epsilon^2 X\_2 + \ldots\Bigr)^2 + \Bigl(X\_0 + \epsilon X\_1 + \epsilon^2 X\_2 + \ldots\Bigr) - \epsilon = 0.
$$

Expand and collect orders of :

and thus to two orders, we have

Of course, we have used a very simple example (a solvable quadratic) to illustrate the idea of asymptotic approximations, but you should hopefully see that this method is extensible to much more complicated equations.

import numpy as np  
import matplotlib.pyplot as plt  
  
ep = np.linspace(1,0.1, 20)  
root1 = (-1 + np.sqrt(1-4\*ep\*(-1)))/(2\*ep)  
root2 = (-1 - np.sqrt(1-4\*ep\*(-1)))/(2\*ep)  
asym1 = 1 - ep  
asym2 = -1/ep - 1  
plt.plot(ep, root1, 'o')  
plt.plot(ep, root2, 'o')  
plt.plot(ep, asym1, '-')  
plt.plot(ep, asym2, '-')  
plt.legend(['$x\_1$', '$x\_2$', '2-term asymp.', '2-term asym'])  
plt.xlabel('$\epsilon$')  
plt.ylabel('x');



## 5.2 Order notation and the tilde sign for asymptotic

We define precisely what we mean when we say that two functions, say and , exhibit the same behaviour in some limit, say or or and so forth. For instance, we claim that the graphs of and look very similar as . Thus we might write

This notation of similarity allows us to specify functional behaviours at a deeper level than just limits. As you can see, it is not as useful to specify that

In contrast, the asymptotic relation is much more prescriptive about the way that the functions are approaching the limit.

|  |
| --- |
| **Definition of , , and**  First, the notation  is read as “ is much smaller than as ” and means  We may analogously use for “much greater than”.  Second, the notation  is read as “ is asymptotic to as ”, and means that the error between and tends to zero as , or  We will often say “ is like ” or “ behaves like ”, |

Here are some examples.

|  |
| --- |
| **Examples**   * as * as * as |

In the examination of limiting processes, often the main issue of consideration is the relative sizes of quantities defined according to their powers. For example, if is a very small number, with , then is much smaller than (in terms of our notation, as ). On the other hand, we might not care so much about the difference between

The point is that the of and is the same as . The “Big-Oh” notation formalises this distinction.

|  |
| --- |
| **Definition of Big-Oh**  We write as to mean that there exists constants and such that |

In practice, the use of the order symbol is very natural and you will not need to work with the technical definition. For example, when you derive the terms of the Maclaurin/Taylor series, you are naturally clustering all the terms of the same order (power) together. For us, the symbol provides a very convenient way of separating terms of different sizes.

|  |
| --- |
| **Examples**   * as * as |

Let us return to the case of the quadratic example ([Equation 5.1](#eq-quad)). Using the O notation, we can write

for the two roots. Alternatively, we can truncate the expansions and simply using the symbol:

# 6. Asymptotic approximations II

In the previous chapter, we learned about the technique of asymptotic expansions, whereby the solution of an equation is expressed in terms of a series in powers of a small parameter:

The precise choice of power progression (here integer powers of will depend on the particular problem. The same idea can be extended to approximating solutions of differential equations. The upshot of this procedure is that at each order of the scheme, a simpler problem can be studied.

Again it is best to demonstrate through examples.

## 6.1 Returning to the projectile problem

In [Chapter 4](#sec-pclass1) and [Chapter 26](#sec-PS1) you studied the non-dimensionalisation of the projectile problem. Once re-scaled, it takes the following form:

This is a difficult problem without, in fact, any explicit solutions. However, we can estimate the solution in the limit . We expand the solution as

In order to expand the denominator, you can use Taylor’s theorem to expand the function

around .

The differential equation now yields

so grouping terms together order-by-order yields

$$
\Bigl[ y\_0'' + 1 \Bigr] + \epsilon \Bigl[ y\_1'' - 2y\_0\Bigr] + \ldots = 0.
$$

We can similarly substitute the expansion into the initial conditions. Altogether, at leading order, we obtain the following system to solve:

Integrating twice and applying the boundary conditions gives us

In fact, this is simply the parabolic motion you would expect from school Physics. The solution corresponds to assuming that the mass at the centre of the planet is dominant and then acceleration is constant.

However, we can now proceed to higher order and examine the nonlinear effects. Proceeding to , we have the following system to solve:

Notice the boundary conditions come from the fact there are no corrections in the original boundary conditions, so for all . Again this system is simple to integrate. Integrating the solution for twice and substitution of the initial conditions yields

We have thus solved for the asymptotic approximation to two orders. We have

$$
y(t) \sim \Bigl[ -\frac{1}{2}t^2 + t\Bigr] + \epsilon \Bigl[ -\frac{1}{12}t^4 + \frac{1}{3}t^3\Bigr].
$$

This was quite an accomplishment! We have taken a problem that was not easily solvable in explicit form and through fairly simple integrations, obtained an approximation to two orders in . How good is it? Let us solve the problem numerically and compare with the asymptotic approximation.

## 6.2 Numerical solutions of IVPs

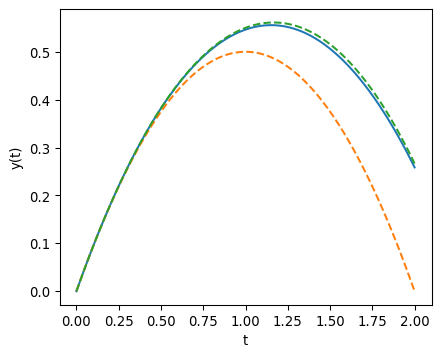
We first demonstrate how to solve ODEs (initial-value-problems, IVPs) using black-box functions in Python. For starters, most numerical formulations for ODEs will require that the problem be posed in terms of a first-order system of equations. To convert ([Equation 6.1](#eq-asymball)) into such a form, create a set of unknowns for the derivatives. Set

Then we have the following first-order system:

You can find a little guide on using solve\_ivp in Python [here](https://ipython-books.github.io/123-simulating-an-ordinary-differential-equation-with-scipy/). Here is the Python code to solve the differential equation.

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.integrate import solve\_ivp  
  
ep = 0.2 # epsilon value  
tmax = 2 # max time  
t = np.linspace(0, tmax, 100) # mesh used for plotting  
  
# Define function for the ODE  
def f(t, Y):  
 ep = 0.2  
 y, yp = Y  
 ypp = -1/(1 + ep\*y)\*\*2  
 return [yp, ypp]  
  
# define the initial condition  
Y0 = [0, 1]  
  
sol = solve\_ivp(f, [0, tmax], Y0, dense\_output=True)  
  
# Prior to plotting, re-interpolate solution on a fine grid  
yy = sol.sol(t)  
# Asymptotic solutions  
y0 = -1/2\*t\*\*2 + t  
y1 = -1/12\*t\*\*4 + 1/3\*t\*\*3  
  
# Plot it all  
plt.plot(t, yy[0,])  
plt.plot(t, y0, '--')  
plt.plot(t, y0 + ep\*y1, '--')  
plt.xlabel('t')  
plt.ylabel('y(t)')

Text(0, 0.5, 'y(t)')



The two-term approximation does beautifully well, even at this moderate value of .

# 7. Numerical solutions of IVPs

In the previous section, we used built-in ODE solvers to develop numerical solutions. It is important to gain an understanding how a simple ODE solver works. The simplest scheme is called [Euler’s method](https://en.wikipedia.org/wiki/Euler_method), and this we now explain.

Begin from the system ([Equation 6.2](#eq-asymsys)). We assume that the solution is represented by a discrete set of points, at the times , , , and so on. The time derivative is written as a discrete derivative while we approximate the right hand side by its value at the nth time step:

Rearranging yields a very simple algorithm for solving the ODE:

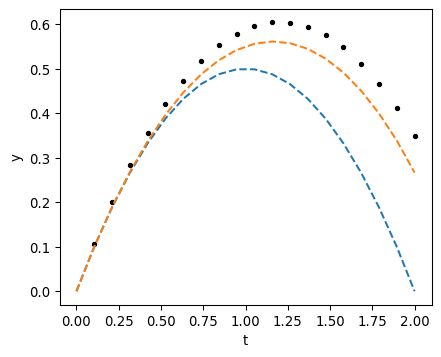
for

This would be implemented via the following pseudocode:

|  |
| --- |
| **Euler’s method**  1. Input: function f(t, Y)   time step, dt  initial condition, Y0  2. Set initial condition Y = Y0  2. Take one Euler step and overwrite previous value   Y = Y + f(t, Y)  3. Increment t by dt and goto 2 |

Euler’s method is conceptually simple but quite inaccurate. But in this case, we see that it works fairly well in comparison to the built-in solvers.

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.integrate import solve\_ivp  
  
ep = 0.2 # epsilon value  
tmax = 2 # max time  
N = 20 # number of steps  
t = np.linspace(0, tmax, N) # mesh used for plotting  
dt = t[1] - t[0]  
  
# Define function for the ODE  
def f(t, Y, ep):  
 y, yp = Y  
 ypp = -1/(1 + ep\*y)\*\*2  
 return np.array([yp, ypp])  
  
# define the initial condition  
Y = [0.0, 1.0]  
ti = 0  
  
# define the solution vector  
for i in range(1, N):  
 ti = ti + dt # Increment time  
 Y = Y + f(ti, Y, ep)\*dt # Euler step  
 plt.plot(ti, Y[0], 'k.')  
  
# Asymptotic solutions  
y0 = -1/2\*t\*\*2 + t  
y1 = -1/12\*t\*\*4 + 1/3\*t\*\*3  
plt.plot(t, y0, '--')  
plt.plot(t, y0 + ep\*y1, '--')  
plt.xlabel('t');  
plt.ylabel('y');



# 8. Asymptotic approximations III

## 8.1 Regular vs. singular problems

In [Chapter 6](#sec-asymptotics02), we studied how asymptotic expansions can be used to approximate equations like

by expanding the solution as . These are known as *regular* problems because a small perturbation, , does not seem to fundamentally change the solution beyond a small perturbation. This is not always the case. In *singular* problems, the situation of is fundamentally different than the situation from . You have already seen such an example in [Chapter 5](#sec-asymptotics01). The equation

has one root for and two roots for non-zero small —even infinitesimally small values! This is quite interesting. From a wider scientific perspective, you may wonder what other problems in nature possess such singular effects.

The point of this lecture is study a technique known as *matched asymptotics*. These matched asymptotics are often necessary for singularly perturbed differential equations.

## 8.2 A singular first-order ODE problem

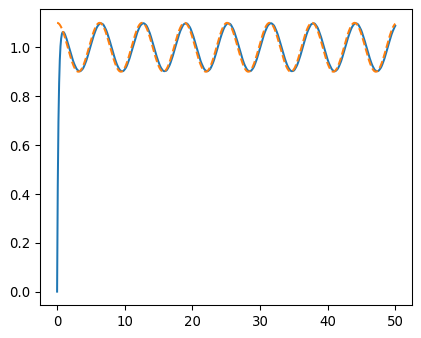
Previously in [Chapter 3](#sec-basic-energy), we derived a basic equation that governs the temperature on the surface of the planet. This equation had the following form:

For the purpose of this section, let us make up a toy model. We suggest, in non-dimensional form,

where we consider and . You can think of the above model as modelling the temperature on a substance that radiates heat in a fashion proportional to itself () and is being subjected to an (incoming) heat source, . Let us take as an example,

Our choice for is not so important. This equation is, in fact, solvable in closed form (how?) but let us get additional practice solving numerically.

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.integrate import solve\_ivp  
  
Ts = 0  
ep = 0.2  
A = 0.1  
f = lambda t, T: 1/ep\*(1 + A\*np.cos(t) - T)  
  
tmax = 50  
sol = solve\_ivp(f, [0, tmax], [Ts], dense\_output=True)  
  
t = np.linspace(0, tmax, 1000)  
y = sol.sol(t)  
plt.plot(t, y[0])  
plt.plot(t, 1 + A\*np.cos(t), '--')



What do we observe? If , then we expect the solution . This is shown with the dashed line. However, this solution does not satisfy the necessary initial condition. We observe that near , the exact solution seems to very rapidly diverge from the approximation in order to satisfy the proper boundary condition. The region in where this rapid change occurs is called a *boundary layer*.

If we repeat the experiment with an even smaller value of , we would observe that the size of this boundary layer seems to tends to zero as . This numerical experiment thus inspires the following method.

## 8.3 Boundary layer theory

We seek a method that will allow us to develop a uniformly valid approximation, i.e. an approximation that is good everywhere in the relevant domain, . Begin by performing the usual asymptotic approximation:

Substitution into the ODE ([Equation 8.1](#eq-matched-T)) yields at leading order,

As we have noted, this approximation fails to satisfy the initial condition, in general. It is possible to go to higher order but this is not so important at the moment. So for now, we have obtained:

$$
T\_{\text{outer}} \sim \Bigl[ 1 + A\cos t\Bigr].
$$

We have chosen to refer to this as the outer solution for reasons that will be abundantly clear. But rather than satisfying , this approximation has the limiting behaviour of

$$
\lim\_{t \to 0} T\_{\text{outer}} \sim \Bigl[ 1 + A\Bigr]
$$

Above, we have only included the leading term in the limit expression.

### 8.3.1 The inner scaling

Our intuition follows a very similar logic to the examination of the singular root in [Section 5.1.1](#sec-asym1-singular). Above, our naive assumption was that could be ignored since is a small number. However, this may not be the case if the gradient is very large.

Our intuition further suggests that the boundary layer occurs near and that it scales in size with . Therefore, let us set

as a change of coordinates. We expect (otherwise is not small), and within this region, we expect the new coordinate, , to be (of moderate size). We then transform the unknown function:

and seek a new differential equation for . By the chain rule,

Before substituting into the equation, we are prudent to examine the behaviour of near . We know by Taylor’s theorem that

Therefore, under the substitution, we may approximate by its leading terms:

For now, we will not need more terms than this. Substituting into the ODE now gives

$$
\epsilon^{1-\alpha} \frac{\mathrm{d}U}{\mathrm{d}s} \sim \Bigl[1 + A\Bigr] - U.
$$

Now in order to involve the first term, it is sensible to select

### 8.3.2 The inner equation

Therefore, the correct coordinate re-scaling was the ‘obvious’ one:

Substituting this again in ([Equation 8.1](#eq-matched-T)):

The procedure is now exactly the same. We expand

At leading order, we get

The above ODE can be solved by integrating factors. Multiplying both sides be , we have

Integrate and use the initial condition:

This is exactly what we expect. Notice that

therefore the outer limit of our inner solution matches the inner limit of our outer solution. In terms of outer coordinates, our inner solution is approximated as follows:

Let’s finally plot this with our previous curves:

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.integrate import solve\_ivp  
  
Ts = 0  
ep = 0.2  
A = 0.1  
f = lambda t, T: 1/ep\*(1 + A\*np.cos(t) - T)  
  
tmax = 20  
sol = solve\_ivp(f, [0, tmax], [Ts], dense\_output=True)  
  
t = np.linspace(0, tmax, 1000)  
y = sol.sol(t)  
plt.plot(t, y[0])  
plt.plot(t, 1 + A\*np.cos(t), '--')  
plt.plot(t, (1 + A) + (Ts - (1 + A))\*np.exp(-t/ep), '--')



It works beautifully!

### 8.3.3 Summary

Let us summarise the procedure of matched asymptotics.

1. Expand the solution of the differential equation naively in the typical asymptotic expansion (e.g. in powers of ).
2. Notice that the approximation does not satisfy certain boundary conditions.
3. Re-scale the coordinates in the ‘inner’ regions.
4. Develop an inner solution that satisfies the boundary condition. Ensure it matches the outer solution.

You will get more practice of this procedure in the problem sets.

# 9. Problem class 2

This problem class will involve the completition of the ODE studied in asymptotic approximations III. We will then study the boundary-value problem which forms part of the third problem set, namely:

The associated Python note is in lectures/lecture12-SolvingBVPs. You will gain an appreciation for some of the unique elements that are introduced when looking at in boundary-value problems.

# 10. Nonlinear root finding

Before we move on back to the subject of applications, it is worth providing an introduction to methods for solving nonlinear equations. Many problems you will encounter in applications, including for ordinary or partial differential equations, can be re-formulated as the solution of a nonlinear system of equations.

Newton’s method is the most well-known scheme for solving nonlinear equations. Suppose we wish to solve the scalar equation,

given some initial guess, , of the root. Suppose the root lies at . Then by Taylor’s theorem,

If we assume the quadratic terms are negligible then solve for this gives

There is a geometrical interpretation of the above. Essentially, in order to estimate the root of , we have used the tangent line at the point , and used the intersection of this tangent line with the axis as the approximation. This procedure can then be iterated.

Thus, provided that the desired root, , is such that , and is sufficiently close to , then the following iterates converge to the root:

## 10.1 Demo of Newton’s method for scalar equations

Here is a simple demonstration of Newton’s method in order to solve for one of the roots of the following:

We will start with the initial guess of . We can do this by hand in lectures using a pocket calculator. The solution is

| i | xi | f(xi) | f’(xi) | -f(xi)/f’(xi) | error |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |

import numpy as np  
  
x0 = 0  
N = 10  
  
def Newton(f, df, x, maxiter=10):  
 i = 0  
 while (i < maxiter):  
 err = f(x)  
 x = x - err / df(x)  
 print("f(x) = ", np.abs(err), ", x = ", x)  
 i = i + 1  
 return x, err  
  
f = lambda x: x\*\*3 + x - 1  
df = lambda x: 3\*x\*\*2 + 1  
  
x, err = Newton(f, df, x0, N)  
print("Final approximation = ", x)

f(x) = 1 , x = 1.0  
f(x) = 1.0 , x = 0.75  
f(x) = 0.171875 , x = 0.686046511627907  
f(x) = 0.008941036638283384 , x = 0.6823395825973142  
f(x) = 2.823062168566537e-05 , x = 0.6823278039465127  
f(x) = 2.839946056099052e-10 , x = 0.6823278038280194  
f(x) = 2.220446049250313e-16 , x = 0.6823278038280193  
f(x) = 1.1102230246251565e-16 , x = 0.6823278038280193  
f(x) = 1.1102230246251565e-16 , x = 0.6823278038280193  
f(x) = 1.1102230246251565e-16 , x = 0.6823278038280193  
Final approximation = 0.6823278038280193

It is a good idea to also learn how to do this using built-in packages. The ‘fsolve’ function provides a Newton-like nonlinear solver. In fact, it can estimate the Jacobian (derivative), so only the function values need to be provided.

import numpy as np  
from scipy.optimize import fsolve  
  
f = lambda x: x\*\*3 + x - 1  
  
x0 = 1  
x, info, ier, msg = fsolve(f, x0, full\_output=True)  
  
print(msg)  
print(x)

The solution converged.  
[0.6823278]

## 10.2 Newton’s method for systems of nonlinear equations

Newton’s method generalises naturally to the case of a system of equations. Suppose we wish to solve for the unknowns via

We have, via Taylor’s formula,

where is the Jacobian matrix

Therefore, Newton’s method forms the iterates of

$$
\mathbf{x}\_{i+1} = \mathbf{x}\_i {\color{blue}-} J^{-1}(\mathbf{x}\_i) \mathbf{F}(\mathbf{x}\_i),
$$

which takes a very similar form to the scalar case.

However, solution of the inverse of is typically inefficient, and it is better to instead solve for via

and then calculate . There are many ways of solving the above matrix problem efficiently using built-in routines that perform, e.g. Gaussian elimination.

## 10.3 Secant method

In many situations, evaluation of the Jacobian (or derivative) is the most time-consuming or difficult part of a nonlinear solver. Built-in solvers like ‘fsolve’, in fact, have the ability to approximate the derivative numerically.

The Secant Method is similar to Newton’s Method but replaces the derivative by a finite difference. Geometrically, the tangent line is replaced with a line through the two last known guesses. The algorithm goes as follows.

|  |
| --- |
| **Secant method**   1. Develop two initial guesses to the solution, x0 and x1 2. Compute |

# 11. Numerical solutions of PDEs

Like the subject of numerical solutions of ODEs, you will only be given a very brief introduction to how PDEs are solved based on the simplest (Euler) scheme.

Let us consider the solution of the heat equation on a finite interval:

## 11.1 Finite difference problem

We consider a discretisation of the spatial domain via the points

Typicall is small. We also have time steps of size , and thus going from , , and so forth. We represent the solution by the indexing scheme:

Like for Euler’s method, we consider the approximation of the time derivative as follows:

In order to approximate the second-order derivative, Hwe will use

Note that the finite difference cannot be applied directly to the first and last points ( and ). Substituting the above finite differences into the heat equation, we see that we now have the prescription:

which applies for .

## 11.2 Boundary and initial conditions

The initial conditions are implemented by setting

The boundary conditions require

During the lectures, we will examine script `

# 12. Problem class 3

This problem class, we will study an example of a PDE model for heat transport, and then use it to inform ourselves on a typical physical problem.

We are interested in finding the optimal depth for a storage cellar that might be used to house wine. Based on our previous discussions, if we assume that the temperature beneath the ground is given by , with below the surface, then the temperature is governed by

for . This is a highly averaged model, where we assume there is no transport of heat into the horizontal plane.

We shall also assume that the surface temperature is modelled by an oscillatory function. On the assumption that the temperature has been normalised so that corresponds to the mean temperature of the surface, then we have

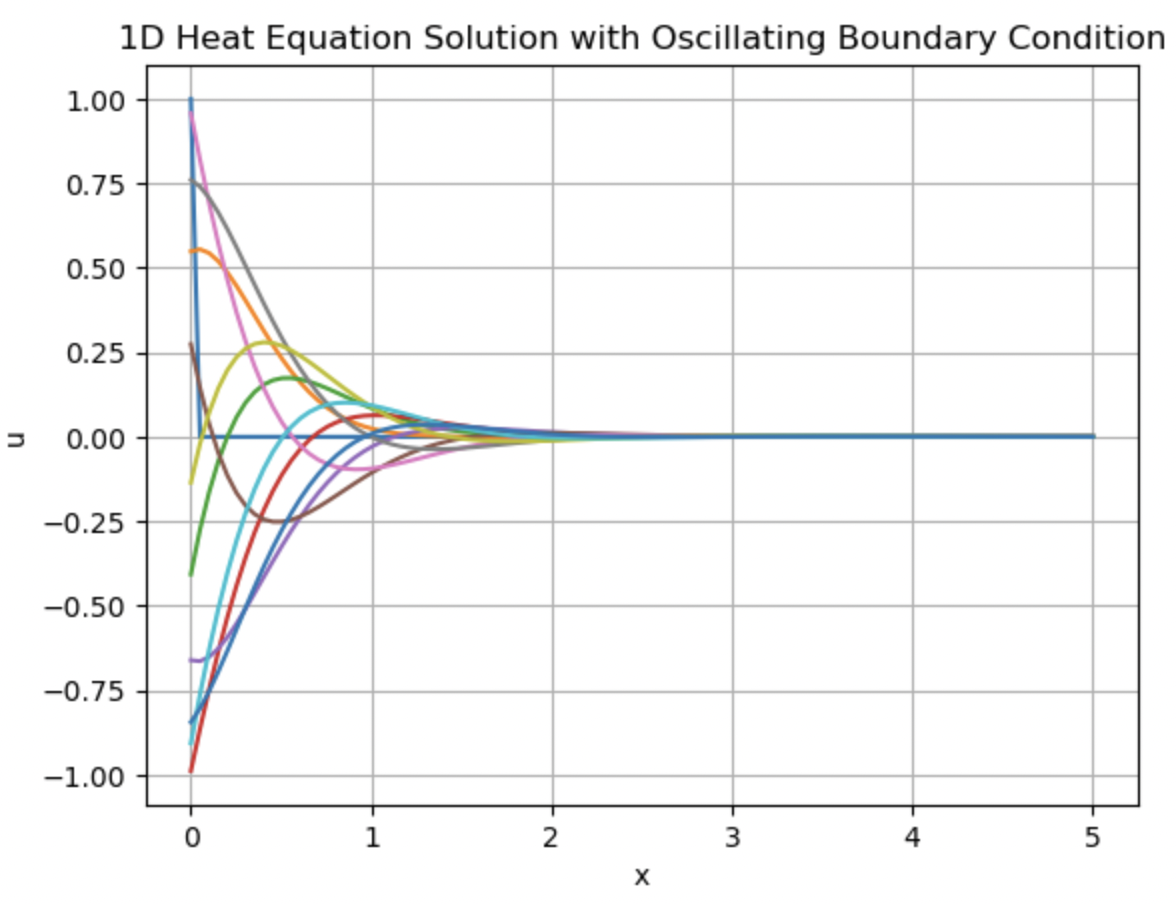
The wavelength is then which we can set to one year. So .

In addition, we expect that the temperature should not be singular far beneath (in particular we expect it to get colder as we go further underground), at least on a scale where we are not approaching the centre of the Earth! Thus it is sensible that

You will not require an initial condition of the discussions below.

## 12.1 Numerical solutions

An example code that uses first-order explicit Euler timestepping is given in lecture15-winecellar on the lectures coding folder. A typical numerical output is given here:



Numerical solution of the wine cellar problem

## 12.2 Analytical solution

Before we begin, it is useful for the purposes of manipulation to write the boundary condition as

This has the advantage that we can manipulate exponentials much more easily.

To solve the problem analytically, we attempt a solution method called separation of variables. Let

Substitution into the PDE gives

and we attempt to separate the functions of to one side and the functions of to the other.

Now in order for two completely different functions dependent on two completely different independent variables to be equal in this way, then this implies that both sides must be equal to a constant. So we have

Therefore . Because of the boundary conditions that would require , you can check that and real are both impossible. The only way to satisfy the boundary conditions is if . So we conclude that

To solve

you can attempt the substitution and solve the resultant equation for the characteristic polynomial in . You will do this in the problem set, and obtain

Therefore

The first solution should be discarded since it blows up as .

Finally, we conclude that the solution is given by

Move the imaginary arguments together:

Now we take the real part, giving

and we see that having set the constant to the solution indeed satisfies .

## 12.3 Selection of optimal depth

The optimal depth of the wine cellar is selected by taking the depth to be such that the cosine is completely out of phase with the surface solution. This means that

You can substitute other multiples of . Thus we have

We can estimate the numbers as follows. Converting from 1 year to seconds yields

Also, the for dry soil is around .

Putting these numbers in yields

# 13. EBM with nonlinear albedo

Recall that we previously introduced the basic energy balance model in [Chapter 3](#sec-basic-energy). There, we derived the basic heat equation model ([Equation 3.7](#eq-heateq-earth)) for the Earth’s temperature given by the following ordinary differential equation (ODE) for ,

where we have defined as the heat capacity of the atmosphere. Above, the solar flux, Q is often taken to be W/m2, W/(m2K4), and is the greenhouse gas factor.

In this chapter, we discuss the effects of considering a nonlinear albedo, , as well as some of the numerical and analytical tools at our disposal for studying the above equation.

## 13.1 Steady-state analysis

Below, we shall let be the steady-state solution that is independent of time. Whenever convenient, we will drop the subscript notation and simply refer to the steady state as .

Previously, we have assumed that the planetary albedo, , is constant and independent of temperature. In actuality, water can turn to snow and ice and vice versa; since snow and ice have much higher albedo than open water, then we should consider .

Let us assume that there are two relevant ranges to consider: T < 150K (cold) and T > 280K (hot). Let us assume that the albedo is, in these two regions:

The above guarantees that more energy is reflected if temperatures are low. To model this process, we can use a ramp function to specify the albedo over all temperatures:

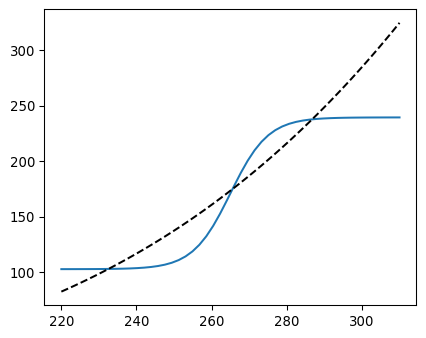
where , $B = {\color{blue}0.2}$, , and . Recall that the tanh function is given by

Let us further assume that the system is in steady state, so that the temperature is determined by solving the equation

In the following code, we plot the two terms that make up , and their intersections indicate roots of . We then use the Python ‘fsolve’ function to approximate the roots given initial guesses.

import numpy as np  
import matplotlib.pyplot as plt  
import scipy.optimize as sciopt  
  
Q = 342  
sigma = 5.67e-8   
gamma = 0.62  
  
TT = np.linspace(220,310,50)  
  
def fun(T):  
 a = 0.5 - 0.2\*np.tanh((T - 265)/10)  
 x = (1-a)\*Q  
 return x  
LHS = fun(TT)  
  
plt.plot(TT, LHS)  
plt.plot(TT, gamma\*sigma\*TT\*\*4, 'k--')  
  
def eq(T):  
 x = fun(T) - gamma\*sigma\*T\*\*4  
 return x  
T1 = sciopt.fsolve(eq, 230)  
T2 = sciopt.fsolve(eq, 265)  
T3 = sciopt.fsolve(eq, 290)  
print("T1 = {:.2f}".format(T1[0]))  
print("T2 = {:.2f}".format(T2[0]))  
print("T3 = {:.2f}".format(T3[0]))

T1 = 232.55  
T2 = 265.56  
T3 = 286.74



Therefore multiple equilibria are observed.

## 13.2 Bifurcation diagram

During the lectures, the above script plotting the steady-state solutions will be examined in order to understand the effects of changing .

It will be observed that dependent on the solar radiation, there may be one or three steady-state solutions. This leads to the following key bifurcation diagram.

|  |
| --- |
| Figure 13.1: Bifurcation diagram of vs |

We noted the following:

* The system has three steady states, given by the green, red, and blue curves.
* The middle state is unstable (shown dashed).
* The system exhibits hysteresis. Note that if we decrease past the tipping point, marked in the image, then we would evolve to the lower stable steady state (which is the ice state). However, while we are in the ice stated, if we were to attempt to increase the solar radiation to return to the green branch, we would need to arrive at to do so; this irreversibility is known as hysteresis. ## Dynamics and phase line solutions

The full time-dependent model is given by

so we may use the positivity or negativity of in order to sketch the time-dependent behaviour of the system.

To see this, we can perform an asymptotic analysis near the fixed points. Let the initial condition be considered near the fixed point:

where . Then we expand the solution into an asymptotic expansion,

Substitution into the above ODE gives, at ,

and hence, with , we have

Therefore, depending on the positivity or negativity of the gradient function, the perturbation will either decay or grow as .

It can then be verified that the centre equilbria is unstable while the other two are stable. The higher temperature corresponds to the one that the Earth is currently in, but according to this model, there seems to be the possibility of a colder climate (50 degrees colder) where the Earth is entirely covered with snow and ice.

Interestingly, there is some evidence that the Earth’s climate may have been in this so-called state up to four times between 750 million and 580 million years ago (Neoproterozoic age). Observations of geological deposits suggest that the Earth has undergone periods of complete global glaciation where there have been very minimal biological activity. During this period, there is a massive build-up of CO2 in the atmopshere, leading to huge greenhouse effect. As decreases in our model, the equilibrium can then shift, suddenly transitioning the Planet into the warm state.

## 13.3 Re-scaling and Budyko’s model

To model the outgoing radiation, we use the quartic Stefan-Boltzmann law. However, over the range of temperatures we are interested-in, it seems that a simpler approximation is sufficient. In your homework, you will investigate the re-scaling and shifting of temperature, such that

where . Then under the assumption that temperatures are not-so-far from , we expand

This simplifies the model considerably.

This is, in fact, a theoretical justification of **Budyko’s model**, first suggested by Budyko (Budyko 1969), where instead of the Stefan-Boltmann law that provides the cumbersome quartic power of , we use instead

where and will vary with location and climate. For instance, for the Northern Hemisphere, (Kaper and Engler 2013) gives the values of and and where temperature is measured in degrees Celcius. As shown above, one can interpret this as a linear expansion of the Stefan-Boltzmann law (after which similar values of and are derived).

The above Budyko model will be a staple of the latitude-dependent model introduced in the next chapter.

# 14. EBM with latitude I

We now introduce a model that allows for two additional effects: an albedo that models ice dynamics and latitude variation; and a transport term, which allows for heat movement (primarily convection).

To begin, let us use a spherical coordinate system where is the latitude angle, with at the equator and at the North Pole. Note that this differs from the [traditional spherical coordinate system](https://en.wikipedia.org/wiki/Spherical_coordinate_system) where is the polar angle. It is moreover convenient to use

Our conservation equation takes the general form for the temperature, :

where we shall use the following:

Each of these terms we explain below in turn.

The above equation will be supplemented with initial conditions:

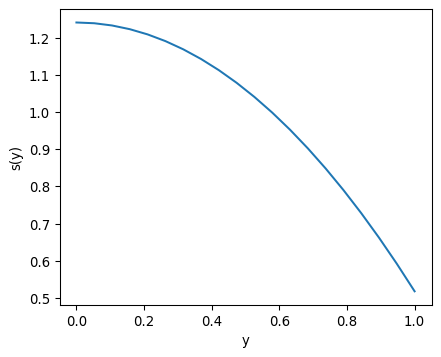
## 14.1 Incoming energy

We make the following remarks about .

* Notice that now, compared to the previous model, we have weighted the incoming solar constant with a term and also our albedo no longer directly depends on temperature but instead depends on the latitude.
* Points on Earth that are closer to the equator receive more direct sunlight and experience more hours of daylight on average. We thus define a function to account for this and consider the incoming radiation as weighted by .
* Note that the total solar input is calculated by a surface integral:
* if the function has been normalised appropriately so that its integral is equal to one, and furthermore the radiation is assumed symmetric about the equator.
* The above is equal to the solar flux intercepted by an area of the circular disk of the earth seen by the sun, which is for . Therefore .

Finally the function is fitted to data. We can assume it to be given by

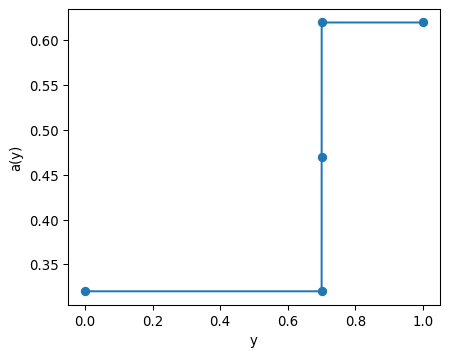
import numpy as np  
import matplotlib.pyplot as plt  
  
y = np.linspace(0, 1, 20)  
plt.plot(y, 1 - 0.482\*(3\*y\*\*2 - 1)/2)  
plt.xlabel('y'); plt.ylabel('s(y)');



We assume that an ice sheet forms if the temperature is sufficiently low, and . The ice forms at an ice line, where . We then have the following form form the albedo:

Therefore we are using a discontinuous model for the albedo. Note that the ice boundary is an unknown to be determined by the model.

import numpy as np  
import matplotlib.pyplot as plt  
  
ys = 0.7  
y = np.array([0, ys-0.0001, ys, ys+0.0001, 1])  
ai = 0.62; aw = 0.32;   
a = np.piecewise(y, [y < ys, y == ys, y > ys], [aw, 0.5\*(aw+ai), ai])  
plt.plot(y, a, '-o')   
plt.xlabel('y'); plt.ylabel('a(y)');



The above forms what is referred to as a free-boundary problem, as the boundary is solved as part of the model. Because is essentially determined by where the temperature reaches the critical value, i.e. , then the above albedo not only depends on latitude, but also on the current temperature distribution as well.

## 14.2 Outgoing energy

The outgoing energy is approximated via the linear Budyko approximation discussed in [Section 13.3](#sec-budyko).

where and . These figures are taken from a 1993 paper by [Graves, Lee, and North](https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/92JD02666).

## 14.3 Transport energy

The next term we should consider is the transport. In the case of this simple model, transport is taken to be modelled via a “Newton law of cooling” type expression. We assume that

where it can be verified (exercise) that the global mean temperature is the same as the hemispherically averaged temperature, i.e.

Above, we shall use a value of , again estimated based on data and studies.

This completes our review of all the components ([Equation 14.2](#eq-EBMlatcomp)) that make up the EBM model ([Equation 14.1](#eq-EBMlat)).

## 14.4 Steady-state temperature

Let be the steady-state temperature and be the corresponding global mean temperature. Then from setting the LHS of ([Equation 14.1](#eq-EBMlat)) to zero, we have the fact that is given by solving the implicit equation:

$$
T\_\infty(y) = \Phi(T\_\infty) = \frac{{\color{blue}k}\bar{T}\_\infty + Qs(y)(1 {\color{blue}-} \alpha(y)) - A}{B + C}.
$$

It is important to remember that the albedo, , implicitly depends on because at each location , we must decide if the temperature is less or greater than the threshold , which is the temperature of the ice line.

The above equation is somewhat tricky because of its implicit nature; it is made even trickier by virtue that multiple equilibria exist at the same parameter values. We will begin studying this problem, but before doing so, let us make a few remarks:

* It is possible to solve for the steady-state global mean, , as a function of the ice line location, .
* It is possible to solve for the ice line location, , although not entirely explicitly. Up to four solutions are possible.
* Once the ice line is known it is possible to solve for the steady-state.

The next few chapters will investigate the above steady-state properties. However note that a numerical solution can be developed without the need to go through the above procedures, as well, albeit in a complicated fashion (In 2024, this was addressed during the Lecture 25; see the visualiser notes).

# 15. Problem class 4

This problem class will be devoted to getting started on problems from Problem Set 4 [Chapter 29](#sec-ps4).

# 16. EBM with latitude II

This lecture, we continue our investigation of the latitude-dependent EBM.

|  |
| --- |
| **Latitude-dependent EBM**  Our conservation equation takes the general form for the temperature, :    where we shall use the following: |

Our goal is to study the steady-state solutions, , given by solving the implicit equation:

We want to know whether other solutions exist with ice lines at .

## 16.1 Development of an equation for the ice line

It turns out to be possible to develop an equation for the ice line. First, note from your problem set 5 in [Chapter 30](#sec-ps5), you find that it is possible to solve for the steady-state mean temperature, which we write as . In your problem set, you will develop the following equation by integrating over ([Equation 16.1](#eq-EBMlat2)) from to .

Taking the steady-state limit of ([Equation 30.1](#eq-meanT-evo-ps)) this yields

The quantity is given by integrating the albedo as follows:

$$
\bar{a} = \int\_0^1 s(y) a(y) \, \mathrm{d}y = a\_i + (a\_w {\color{blue}-} a\_i)y\_s[1 - 0.241 (y\_s^2 - 1)],
\qquad(16.5)$$

which is given by ([Equation 30.2](#eq-ps4-abar)). You may check that the above formula has the right signs by considering either the complete ice case (ice line at the equator ) or the complete water/land case (ice line at ).

Now we return to the implicit equation for the temperature, which is given by ([Equation 16.3](#eq-latitude2-Phieq)) and repeated here:

Substitute the mean temperature in, and this now yields

This ice line is then found by setting at in the above formula. Notice, though, that since is a cubic function, then we would need to solve a cubic equation in general.

## 16.2 A word about the parameter space

Solutions to our latitude-dependent EBM can be symbolically written as follows:

As you can see, even though it is a relatively simple equation in the sense it is only an ODE in time (and does not involve any spatial derivatives in ), is still complicated because the behaviour of the system can depend in a non-trivial way on all the parameters.

You can schematically think of the solution space as a being plotted in 8-dimensional space (or even higher, since the albedo can be specified more generally). So for instance, bifurcation diagrams can then be plotted for some norm of the solution, say versus the seven other parameters.

There are other representations of the bifurcation diagram(s). For example, you might plot vs or vs , and so forth.

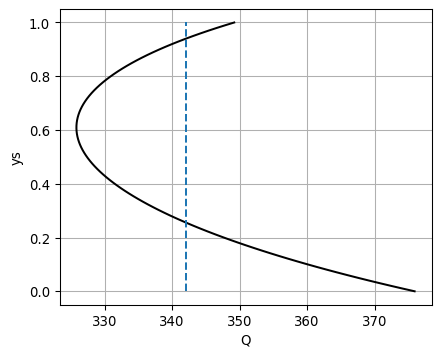
## 16.3 A bifurcation diagram for vs

Returning to the ice line, we are interested in keeping all other parameters fixed, and then attempting to understand how the ice line evolves as the solar constant is changed. For example, we might believe that as increases (and hence temperatures rise), the ice line will move towards the North Pole. And as decreases, the ice line moves towards the equator.

We return to ([Equation 16.6](#eq-Tstar)) and consider inverting the formulation. For each given ice line location, , we solve for the value. This gives

Above, we have used the fact that . This can now be plotted in Python, either in the notebook written during lectures, or via the code below.

import numpy as np  
import matplotlib.pyplot as plt  
A = 202 # outgoing radiation  
B = 1.9 # outgoing radiation  
k = 1.6\*B # transport parameter  
s = lambda y: 1 - 0.482\*(3\*y\*\*2 - 1)/2 # solar weighting  
aw = 0.32 # water albedo  
ai = 0.62 # ice albedo  
Tc = -10.0 # critical temperature for ice formation  
Q0 = 342.0 # solar constant (1370 W/m^2 divided by 4)  
  
abar = lambda ys: ai + (aw - ai)\*ys\*(1 - 0.241\*(ys\*\*2 - 1))  
Qfunc = lambda ys: (Tc + A/B)\*(B+k)/(s(ys)\*(1 - (ai+aw)/2) + k/B\*(1 - abar(ys)))  
  
# Solve for the ice line  
ys = np.linspace(0, 1, 100);  
Qs = Qfunc(ys);  
plt.plot(Qs, ys, 'k')  
plt.plot([Q0, Q0], [0, 1], '--')  
plt.xlabel('Q');  
plt.ylabel('ys');  
plt.grid(1)



On the above graph, we have plotted the reference line, which is at W-m-sq. So amazingly, there are two intersections with the black curve, which in fact indicates that two ice lines seem to be possible. One ice line as near and the other has much lower down and near the equator.

Moreover, the above graph is not complete! When we computed it, we specified that but this would not be true if there were no ice line, or equivalently if the ice line is located directly at or . These yield the so-called ice-free states and ice-covered states, respectively.

## 16.4 Ice-free state

The ice-free state is the state for which for all . In this case, the solution can be directly calculated from [Equation 16.6](#eq-Tstar). It would then be given by

This solution has a requirement, which is that when considering the solar constant , the solar constant cannot be so weak so that an ice line appears for . The minimum value of is therefore determined by pinning the ice line right at the North Pole. Thus

It is of interest to verify that the global mean temperature for a system in this state is a warm degrees Celsius. According to this model, then, if the mean temperature is above this value, it is possible to have a state where there is no ice anywhere!

## 16.5 Ice-covered state

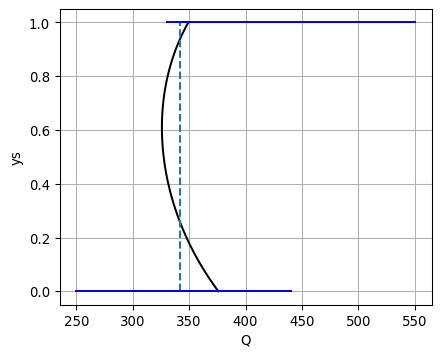
A similar argument applies setting for all . In this case,

This time, there is a maximal condition on such that any higher value would require an ice line somewhere within the domain, i.e. . Thus

Again we may verify that this corresponds to a mean temperature of degrees Celsius.

import numpy as np  
import matplotlib.pyplot as plt  
A = 202 # outgoing radiation  
B = 1.9 # outgoing radiation  
k = 1.6\*B # transport parameter  
s = lambda y: 1 - 0.482\*(3\*y\*\*2 - 1)/2 # solar weighting  
aw = 0.32 # water albedo  
ai = 0.62 # ice albedo  
Tc = -10.0 # critical temperature for ice formation  
Q0 = 342.0 # solar constant (1370 W/m^2 divided by 4)  
  
Qmin = ((B+k)\*(Tc + A/B))/((1-aw)\*(s(1)+k/B))   
Qmax = ((B+k)\*(Tc + A/B))/((1-ai)\*(s(0)+k/B))  
print("Minimal Q for ice-free state = ", Qmin)  
print("Max Q for ice-covered state = ", Qmax)  
  
# Note version in lectures was a 1/2 factor off on the second grouping of terms;   
# now corrected to match Q6 of PS4  
abar = lambda ys: ai + (aw - ai)\*ys\*(1 - 0.241\*(ys\*\*2 - 1))  
Qfunc = lambda ys: (Tc + A/B)\*(B+k)/(s(ys)\*(1 - (ai+aw)/2) + k/B\*(1 - abar(ys)))  
  
# Solve for the ice line  
ys = np.linspace(0, 1, 100);  
Qs = Qfunc(ys);  
plt.plot(Qs, ys, 'k')  
plt.plot([Q0, Q0], [0, 1], '--')  
plt.plot([250, Qmax], [0, 0], 'b')  
plt.plot([Qmin, 550], [1, 1], 'b')  
plt.xlabel('Q');  
plt.ylabel('ys');  
plt.grid(1)

Minimal Q for ice-free state = 330.3616063989335  
Max Q for ice-covered state = 440.72694936919913



## 16.6 Partially ice-covered states

In lectures, we demonstrated, that there are four possible solutions of the latitude EBM. Two of the solutions, developed above, correspond to completely covered ice-state and ice-free states. The other two solutions are partially-iced states.

|  |
| --- |
| Figure 16.1: The four possible solutions; orange and blue show the completely-ice and ice-free states; red and green show the partial ice-covered state. These are shown for , , , and = 342$ |

# 17. EBM with latitude III

We continue our investigation of the latitude-dependent model of [Chapter 16](#sec-latitude2).

## 17.1 Studying the mean temperature

Following the previous chapter, we now have a better understanding of the relationship between the ice line, , and the solar forcing, . We demonstrated that for the baseline case of , (at least) four solutions are possible.

|  |
| --- |
| Figure 17.1: The iceline |

Examination of the root shows that the two non-trivial ice lines are approximately at and for . Using the script developed in lectures, we can study a similar graph of the mean temperature versus the solar forcing.

This yields the following picture:

|  |
| --- |
| Figure 17.2: Mean temperature |

During the lecture, we discussed the so-called *Snowball Earth* scenario where decreasing the solar constant may cause the stable state of the planet to jump down to the green branch shown above (completely frozen Earth). The solution exhibits *hysteresis* which describes the non-reversibility shown in the bifurcation diagram. This will be further expounded in the following discussion on stability.

## 17.2 Stability

So far, we have only discussed the steady-state solutions of the latitude-dependent EBM, repeated below:

In this section, we investigate the stability analysis for the mean temperature, . This was given in [Equation 30.1](#eq-meanT-evo-ps) and repeated below:

The following linearisation argument was presented by [Cahalan and North (1979)](https://moodle.bath.ac.uk/pluginfile.php/2213909/mod_resource/content/0/Cahalan_North_1979.pdf), with some additional details to avoid confusion. Let us linearise this about a steady-state solution, writing

where . By Taylor series,

so substitution into the ODE yields the linear equation

We need to calculate the derivative term on the RHS. First, using the equation for in [Equation 17.1](#eq-meanT-evo), we have

The above calculation assumes that is a fixed number, as it should, since is regarded as a parameter (we choose a value of , then this allows us to evolve the ODE).

|  |  |
| --- | --- |
| **Calahan & North’s “trick”**  The following presentation took about 3 lines of text in the presentation of Cahalan & North (1979), between eqns (1.2) and (1.3) in the image below. Students have inquired about the manipulations.   |  | | --- | | Figure 17.3: From Cahalan & North, p.1179 | |

The problem is what to do with the derivative of that appears in ([Equation 17.3](#eq-Gprime)). This is a strange quantity, since does not explicitly depend on , and is given in ([Equation 16.5](#eq-abar-again)). The issue is that there is a hidden dependence: depends on the ice line, , which depends on not only , but is multivalued as well!

Cahalan & North, whether knowingly or unknowingly, applied a trick, which is to consider that is not a parameter, but that it can be viewed as depending on , the steady state (this dependence is shown in [Figure 17.2](#fig-Tbar)). Therefore, Cahalan & North returned to the equation ([Equation 16.4](#eq-meanT-again)) that gives the steady-state:

We differentiate this equation with respect to , but now assume both and depend on . This yields

We substitute the above value of into ([Equation 17.3](#eq-Gprime)), and then finally into the equation for the linearisation ([Equation 17.2](#eq-linearu)). Then we have

where the constant has been defined by the above. Thus, the solution is given by

Note then that the crucial quantity is

If is positive, the equilibrium point is stable, while if is negative, the equilibrium is unstable. Because , then this yields the so-called slope-stability theorem coined by [Cahalan and North (1979)](https://doi.org/10.1175/1520-0469(1979)036%3C1178:ASTFEB%3E2.0.CO;2):

We may now plot the bifurcation diagram, shown in -space. First, return to [Figure 17.2](#fig-Tbar). In lecture 21, we drew the following picture:

|  |
| --- |
| Figure 17.4: Bifurcation diagram for |

Note the following characteristics.

1. There are up to four steady states (green, orange, two blacks).
2. Those branches with positive gradient are stable; there is only one branch with a negative gradient and it is thus unstable. In the image, these are shown solid for stable and dashed for unstable.
3. The diagram indicates tipping points and hysteresis. As explained in the lectures, one can encounter a situation where, beginning from the upper orange branches, the solar radiation is decreased past the tipping point. The solution must then evolve to the lowermost stable state (green). However, increasing does not return us to the original non-frozen state—unless is increased by a massive amount. This ‘hysteresis’ loop is shown with the maroon colours in the image above.

# 18. EBM with latitude IV

//Temporary spare lecture//

# 19. Ice ages I: an introduction

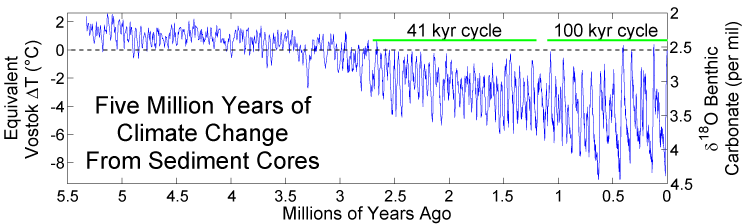
Our next investigation will take us to the topic of fast-slow systems in order to posit toy mathematical models for ice ages.

## 19.1 Background

We have mainly been concerned about time periods in the decades, but it is now time to discuss much longer time periods. In particular, we are interested in the *Pleistocene era*, which lasts from around 2.5 million years go to 11,700 years ago; this period marked the evolution of *Homo sapiens*. Begin by examining this figure:

|  |
| --- |
| Figure 19.1: Figure from [here](https://en.m.wikipedia.org/wiki/File:Ice_Age_Temperature.png) |

This figure indicates temperature changes tracked via various measures at different locations. The top two graphs show the temperatures derived from two different ice core measurements. The lower curve is the total ice cover. In this figure, you see evidence for the four complete glacial cycles with the Earth temperature dropping into the ice ages. A key observation concerns the regular periodic behaviour, with regular cycles appearing every 100k years.

Turn now to this graph: 

This graph goes back even further. The key observation is that between 1 million and 2.5 million years, the periods seem to be predominantly 41k years. Thus sudden transition is referred to as the [Mid-Pleistocene Transition (MPT)](https://en.wikipedia.org/wiki/Mid-Pleistocene_Transition). The question of how or why the periods can suddenly change in such a manner is referred to as the [100,000-year problem](https://en.wikipedia.org/wiki/100,000-year_problem).

## 19.2 Milankovitch cycles

The most natural theory to explain the ice age periods is to consider that the Earth’s orbit is affected by the gravitational attractions of the Sun, Moon, and the other planets. The complex interaction results in a slow, cyclic change in three important parameters of the orbit (Taylor 2005):

1. The *eccentricity* of the ellipse that the Earth describes in its orbit around the Sun each year (100kyr);
2. The angle between the equatorial plane and the orbital plane, known as the Earth’s *obliquity* (41kyr);
3. The *precession* of the spin axis around the normal to the orbital plane (20kyr).

|  |
| --- |
| Figure 19.2: Figure from [here](https://skepticalscience.com/graphics.php?g=342) |

The name given to these cycles is *Milkankovich cycles*.

However, the key problem is that the Milankovich cycles do not seem to explain the MPT problem. The Milankovich cycles exhibit the necessary periods on both the 100kyr scale and 41kyr scale, but it cannot explain why there was a sudden transition. Moreover, the response of EBMs to the 100kyr forcing (due to the eccentricity) is not enough to explain why this forcing period takes precedence since it is the 41kyr cycle due to the obliquity that produces the largest effect on the solar radiation.

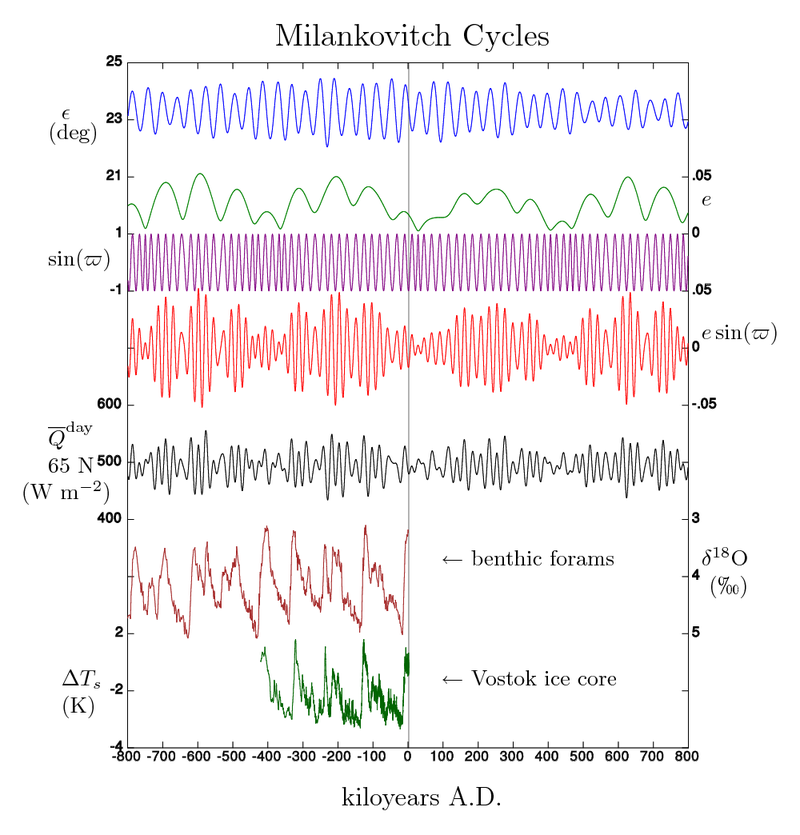


Figure from [here](https://en.wikipedia.org/wiki/Milankovitch_cycles#/media/File:MilankovitchCyclesOrbitandCores.png)

## 19.3 The mathematician’s perspective on models

We start with some discussion of the perspective of a mathematician. The last few weeks, we have been studying models of Planet Earth that take the form of first-order autonomous systems:

or perhaps in the more recognisable ‘dynamical-systems’ form of

for some function . Because this is such a simple model, the analysis of the time-dependent evolution can be done on a **phase line**. Solutions can only exhibit two behaviours as : they can tend towards a fixed point or they can diverge (to ).

There are numerous ways to extend the basic EBM. For instance:

1. (**Latitude variation**) Add latitude variation, with . Indeed this is what we had done in the last few chapters, and using a toy model (Newton’s law of cooling) that imposes a driver into the ODE.
2. (**PDEs**) A more accurate model is to add a diffusion term, e.g. . The Laplacian operator is then turned into a surface derivative in spherical coordinates along the planet. This turns the equation into a PDE, so is more involved to solve.
3. (**Non-autonomous forcing**) We can consider the addition of temporal forcing, for instance,

* This would model, for instance, some oscillatory source input. For instance, this might model extra oscillatory effects due to the influence of the other planets.

1. (**Systems of DEs** ) We might be interested in extending the model so that, in addition to modelling the evolution of the temperature, we also model the evolution of CO2 in the atmosphere. Or we might want a model where the albedo, itself, can evolve in time. An example of such a model is

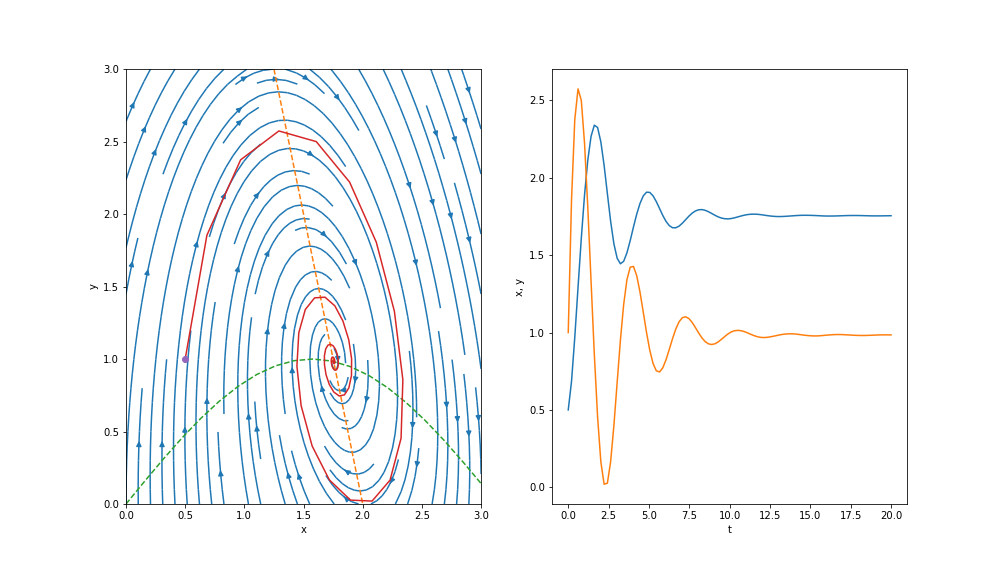
The above provides a model for ice sheet dynamics.

## 19.4 Phase plane analysis for two-dimensional systems

Coming back now to the standard non-autonomous system of two ODEs, we have

What is the difference between these kinds of systems and the simpler 1D equations? The main difference is that the analysis is done in the **phase plane** given by . Fixed points lie in the phase plane, but now there is the additional possibility of oscillations or circular orbits.

An example of this is the system

For instance, choosing and yields the following phase plane diagram: 

We see there exists a fixed point at where the two nullclines (dashed) intersect. Analysis near this fixed point would reveal that solutions encircle in a clockwise direction and tend towards the fixed point. This is shown by the solution plot on the right.

Therefore, one way to extend our EBMs is to consider coupling to another component. Keep this in mind as this is where we will be going in the next lecture.

# 20. Ice ages II: a model for fast-slow dynamics

In order to capture the kind of dynamics observed in ice ages, we are interested in studying mathematical models that produce fast-slow dynamics – that is, dynamics that might consist of a slow evolution in one state, before rapidly jumping to another state; such dynamics might exhibit the kind of periodicity we observed in the profiles of temperatures through the ice ages. We might also be interested in understanding how the periodicity might suddenly change (the Mid-Pleistocene Transition problem) but this will be more challenging.

It turns out that such fast-slow systems are incredibly common. If we return to the canonical example presented in the previous chapter, we can modify it sightly so that it takes the form where is a small parameter.

Studying the above, notice that if is small, then we expect for , and therefore the dynamics should be constrained to lie along this given curve. However, this seems incompatible with the evolution that is specified by the second equation. The result is that this system will exhibit a sequence of slow dynamics (where it is near ) followed by fast dynamics where it hops between the branches of the -nullcline. This will be clear with an example.

## 20.1 The van der Pol oscillator

In an article by [Ditlevsen and Ashwin (2018)](https://www.frontiersin.org/articles/10.3389/fphy.2018.00062/full) the authors argued that the following van der Pol oscillator presents a good toy model for understanding some of the dynamics that occur in ice ages:

where , and represents a model for astronomical forcing. We will not discuss their ‘derivation’ of the above model (presented in their sec. 5), though it seems best to consider it solely as a toy model since otherwise their derivation of its validity based on EBM arguments is somewhat sketchy.

In any case, we are firstly interested in the case of zero forcing, . It can be verified that through a transformation of time, , the above equation becomes

where we have chosen, without loss of generality, for . Again we are considering . This is a very well-known problem in nonlinear oscillations and it is the canonical example of fast-slow dynamics.

### 20.1.1 The Lienard transformation

We first explain how to transform the above second-order equation so that it resembles the system of two linear differential equations shown above. Notice that the equation can be written in the form

We then define

which can be re-arranged to yield

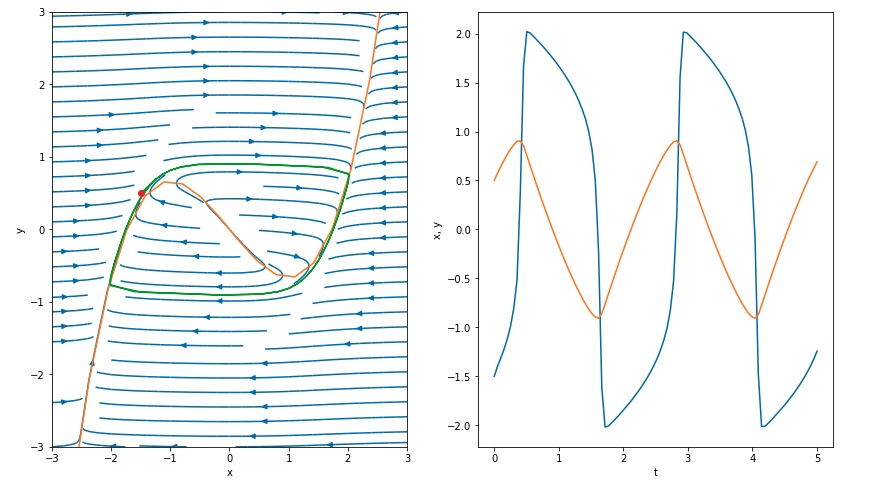
where we have used dots to indicate derivatives in time. The second equation is found by [Equation 20.1](#eq-lienard) itself, since this yields

Therefore together, we have our system:

### 20.1.2 An investigation in Python

The lecture will now investigate the scripts to see the qualitative behaviour of the ODE. We will design a phase plane plotter in Python and see why this is called a fast-slow system. The script can be found here: [lecture23a-typicalphaseplot](https://maths.jupyterhub.bath.ac.uk/hub/user-redirect/lab/tree/courses/MA30287/notebooks/lecture23a-typicalphaseplot.ipynb) (where we show how to do phase plotting in Python) and then here: [lecture23b-fastslow](https://maths.jupyterhub.bath.ac.uk/hub/user-redirect/lab/tree/courses/MA30287/notebooks/lecture23b-fastslow.ipynb).

During the lecture, we discussed the following image:



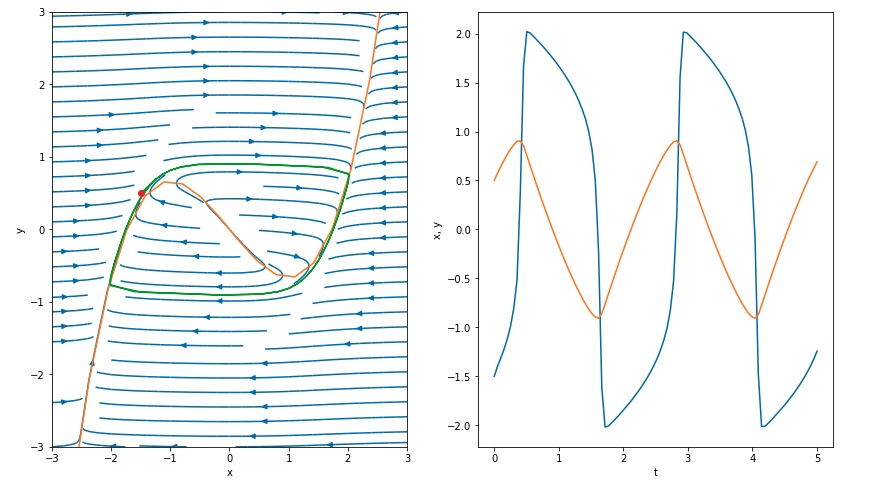
Phase plot of the above system

We interpreted the dynamics that follow either the slow manifold, , and also the fast transitions that occur between the branches of the cubic. Also shown on the right side of the above plots are the profiles, and versus . In the limit , the transitions begin to resemble shocks.

# 21. Ice ages III: analysis of the van der Pol equation

In the previous chapter we introduced a canonical model for fast-slow dynamics given by the following system for and :

where and . This led to our study of then numerical computation of solutions and example graphics like this:



Phase plot of the above system

We now show how to perform the asymptotic analysis of the above system. The techniques used relate to the analysis of boundary layers studied in [Chapter 8](#sec-asymptotics03).

## 21.1 Analysis of the slow manifold

Let

At leading order, we get the following system of equations:

First, examine the differential equation on the second line. Since the rate-of-change of is equal to , we can analyse the motion in the phase plane by the following cases:

Therefore, we conclude that the axis is nullcline in this leading-order approximation. The remaining equation restricts the solution to lie along the **slow manifold**,

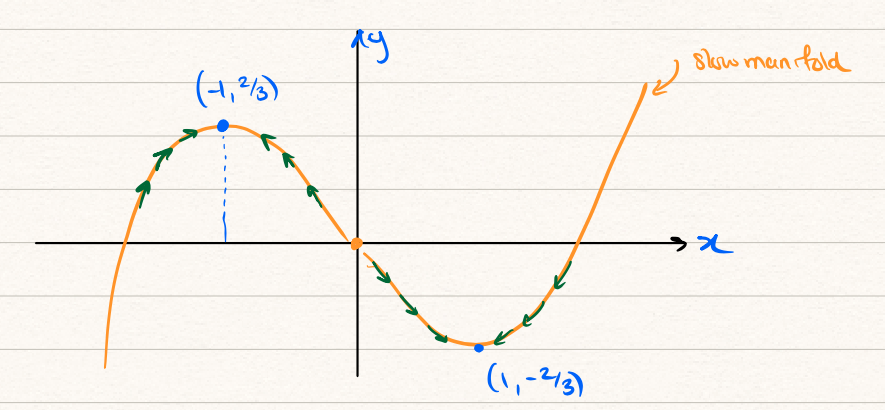


Illustration of dynamics along the slow manifold

According to our analysis, we should have both as and also increasing on while decreasing on . But this yields an apparent contradiction as and .

The issue is that, at the critical point(s) where , say , the solution encounters a shock or boundary layer where the gradient is no longer small. Thus our prior assumption where we neglected is invalid.

## 21.2 Fast analysis

Near a shock, we re-scale

and set and under this new coordinate change. Essentially, this re-scales the coordinate axes, originally in , near the unknown location of the shock, . We can verify that the correct balance for re-scaling is chosen with . From the chain rule, we convert derivatives as follows:

Then the system becomes

We expand now in the typical fashion, setting

At leading order, this now yields

Therefore, from the second equation, we obtain

which is to be expected from the phase-plane diagrams we have shown. Since we have established that the max/min of the slow manifold, , is given by , then we know that the above constant is this value *if the solution had begun from the slow manifold* (see comment later about the initial condition).

Next, the equation for has the form of a first-order nonlinear differential equation that can be studied by phase-line analysis. If the constant is indeed , then the differential equation for the other component takes the form

This leads to the following picture.

|  |
| --- |
| Figure 21.1: Fast dynamics |

To begin, we consider the evolution from the points marked ‘A’ and ‘B’. This occurs along the slow manifold. At the point ‘B’, time reaches the critical point, , and the solutions enter the fast dynamics, following along points ‘B’ to ‘C’. Notice that along this segment in the phase plane,

therefore must increase during this time. However, upon approaching the point ‘C’, the above LHS tends to zero, and hence slows its increase. This exists the fast dynamics layer and again the solution continues along the slow manifold from ‘C’ to ‘D’. Upon reaching ‘D’, it again enters a stage of fast dynamics (the point here is a new one).

## 21.3 Initial conditions

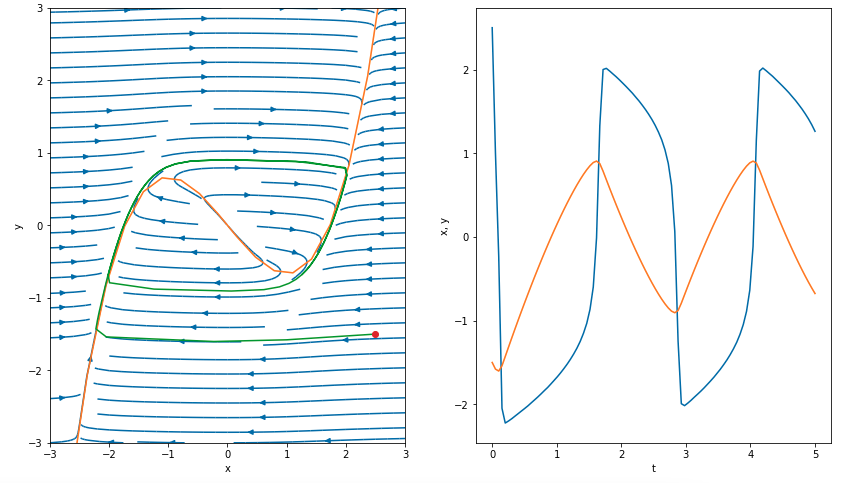
You will notice that above, we have not really discussed the effect of the initial conditions, say,

Perhaps more specifically, we have performed our asymptotic analysis with the assumption that the solution starts directly on the slow manifold, i.e. . Returning to the slow analysis leading to [Equation 21.1](#eq-sloweqns) notice that the above conditions will not in general be compatible with the equations (since ‘A’ and ‘B’ might be chosen completely different from those values where ). The conclusion is then that the slow analysis applied cannot be valid near the initial condition.

This turns out to be very similar to the example studied in [Chapter 8](#sec-asymptotics03), which also had a time-dependent problem where the initial condition was incompatible.

Therefore near , we must introduce an analogous boundary layer. This analysis will be identical to the above fast-scale dynamics except that . The conclusion is that if we begin off the manifold , the solution will rapidly tend left or right in the phase plane (at near-constant values of ). Once it intersects with the manifold, the trajectories then resume the slow and fast dynamics observed above.

This is now shown below for the initial condition of , and .



Notice the red circle in the left phase plot showing the initial condition. We see that the solutions then exhibit the fast dynamics, with decreasing, until encountering the slow manifold. Thereafter it follows the standard slow-fast dynamics.

On the right image, notice the significant shock or boundary layer near . If desired, we could confirm the size of this boundary-layer regions scales with .

# 22. Fast-slow dynamics for higher-order systems

In the last two lectures, we discussed the use of toy models that exhibit fast-slow dynamics. These have been used to posit possible changes in behaviour in ice ages, where historical records confirm that multiple periodic states are possible.

It is important to note that the van der Pol oscillator we have studied, which is equivalent to two first-order differential equations, is really not that unusual, and systems of (three or more) differential equations will often exhibit the kind of fast-slow behaviours observed. This occurs commonly in applications much more general than climate or geoscience—of course, if your view is to concentrate to models of Planet Earth, you can expect this kind of behaviour whenever you have moderately complicated systems of coupled oscillators describing the planet!

As as means of providing you with additional toy examples, let us examine a model used in studying chemical reactions. Consider an enzyme-mediated chemical reaction given by

where is the “substrate” reactant and is the concentration of the product. The enzyme is a compound that allows for an immediate reaction step that leads to the overall reaction, which in this case is . The intermediate substance is , the complex. The point here is that dependent on the rate constants, , , , and the amount of each substance, the overall chemical reaction may be coarse-grained so that you only observe (this is the slow dynamics). We can consider the dynamics of the three substances and derive the following system of three equations

where the three substances have been suitably non-dimensionalised. The key is to assume that the parameters , while

Here is a typical numerical calculation found in the script [here](https://maths.jupyterhub.bath.ac.uk/hub/user-redirect/lab/tree/courses/MA30287/notebooks/lecture25-ThreeTermChemicalReaction.ipynb).

|  |
| --- |
| Figure 22.1: Numerical calculation of the system of three equations with , , and . |

Here is the same solution but now plotted as a function of time.

|  |
| --- |
| Figure 22.2: Plot of the solution evolution with time |

The above graph suggests that there is a boundary layer for the complex near .

## 22.1 Slow analysis

We thus expand the solutions in the typical way, setting

This yields at leading order,

The second equation yields the slow manifold, for which we expect

|  |
| --- |
| Figure 22.3: Numerical solution as before but now plotted in the phase plane. The dashed line is the approximation derived above. |

## 22.2 Fast dynamics near the initial condition

As with our van der Pol example, the initial condition, where and , does not lie on the slow manifold above. Therefore we expect that there is a boundary layer where the system rapidly begins from the initial condition and tends to the slow manifold. The distinguished scaling on time can be confirmed to be

where . We then set

for our new unknowns within this later. Under the scaling, the first two ODEs in ([Equation 22.1](#eq-3chem)) become

Therefore, you see that the leading-order value of is expected to be constant, and it remains at its initial value,

Although we could solve for the concentration of or , this turns out not to be necessary. It should first be remarked that as the outer solution, , tends to the initial condition, , we should have

Compare this with the condition ([Equation 8.2](#eq-matched-matchingcond)) when we initially studied the method of matched asymptotics. Therefore, even though there is a boundary layer necessary for the concentration near , we do not need to resolve it.

## 22.3 An equation for the substrate and product

Take now ([Equation 22.3](#eq-s0c0)), which yields

and substitute this into the first and third equations ([Equation 22.2](#eq-3chem-lead)). This gives

Notice that actually, if we add these two equations, we come out with

which is a conservation of mass/concentration statement that indicates that the total substrate and product is conserved. Therefore, if we only solve the first equation for :

we can obtain the product concentration by .

The above equation ([Equation 22.4](#eq-mmlaw)) is in fact a famous result in biochemistry. It indicates that the substrate concentration follows a nonlinear rate and is called *the Michaelis-Menten law*. We have significantly simplified the necessary mathematical analysis down from the analysis of coupled system of three equations to a single equation for a single unknown ()!

|  |
| --- |
| Figure 22.4: Numerical solution of the single differential equation |

# 23. Basic models of the ocean

## 23.1 Terminology and context

The ocean plays a significant role in regulating the Earth’s climate, as it acts as a massive heat sink and helps to distribute heat and moisture around the planet. In addition, carbon dioxide is water soluble, and through precipitation and wave motion, is transferred into the oceans. Thus the ocean acts as a sink, absorbing large amounts of this greenhouse gas from the atmosphere.

The thermohaline circulation (THC), also known as the global ocean conveyor belt, is a complex ocean circulation pattern that is driven by differences in water temperature and salinity. It is an important component of the Earth’s climate system, as it helps to distribute heat and other properties throughout the planet’s oceans.

The thermohaline circulation is driven by the sinking of cold, dense water in the polar regions, which then spreads out and flows towards the equator. As the water warms and becomes less dense, it rises to the surface and returns to the poles, creating a continuous loop of ocean currents. The role of salinity in driving the thermohaline circulation is due to the fact that the dissolved salts in seawater increase its density.

This circulation pattern has a significant impact on global climate, as it helps to regulate the exchange of heat and other properties between the oceans and the atmosphere. Changes in the thermohaline circulation, such as those caused by global warming, can have far-reaching effects on the planet’s climate and weather patterns.

|  |
| --- |
| Figure 23.1: The thermohaline circulation. In the Atlantic, the circulation carries warm water (red) north near the surface and cold deep water (blue) south. Image from NASA/JPL. |

## 23.2 Temperature

|  |
| --- |
| **Extra content**  The material in this subsection is not examinable in 2022-23 and is given here for further context. |

In regards to the temperature, the ocean can be divided into three layers.

1. The top layer is thin (on the order of metres) and is heated from the Sun. Mixing is a dominant effect due to wind and waves, and so the temperature in this region is mostly constant.
2. The *thermocline region* is the intermediate layer. Here, the temperature decreaes approximately linearly.
3. The deep *abyssal zone* comprises 98% of the total volume of the oceans. The temperature in this region is mostly constant, and a few degrees above freezing.

Within the intermediate region, the temperature can be modelled by an **advection diffusion** equation,

where is the upswelling velocity and is the diffusion coefficient of the fluid.

Let us assume that the temperature in this region is near steady state and that the upwards velocity is constant. Then we integrate the ODE to find

where and are constants. From (Kaper and Engler 2013), the typical orders for the parameters are and , so , which is quite slow.

|  |
| --- |
| Figure 23.2: Cross section of the Atlantic Ocean, showing the temperature and salinity profiles on the right. Image from (Kaper and Engler 2013). |

## 23.3 Salinity and density

Salinity is a key component in the oceans since the salts have a large effect on the water density (which consequently drives motion). Salinity is measured in psu or *practical salinity units*, which is a non-dimensional ratio of conductivities. In the mixed layer, the sainity ranges from 31-39 psu, and is about 35 in the abyssal zone. You can inspect the profile in [Figure 23.2](#fig-oceanslice).

Below in the study of the ocean, we will need a so-called *equation of state*, which connects the density, , with the salinity, say , and temperature, say . Here is a typical graph of the relationship.

|  |
| --- |
| Figure 23.3: Image from (Kaper and Engler 2013) |

## 23.4 Two-box model of the ocean

Modelling the THC is a challenging task! In principal, this might involve the solution of multiple coupled PDEs for the flows and temperatures, which would then need to be solved on a very complicated domain. In addition, such models require a number of empirical equations of state (connecting density to temperature and salinity).

Toy models can be developed much more easily at the ‘systems level’ via box models.

## 23.5 A one-dimensional model (constant temperature)

Construction and assumptions of the box model.

1. We consider two boxes, labeled ‘1’ and ‘2’, respectively via subscripts. Box 1 corresponds to high latitudes (near poles) and Box 2 to low latitudes (near equator). Each box has a corresponding temperature, , and salinity, . In each box, the temperature and salinities are well mixed, so they take different values in either box, in general.

* Differences between the two boxes will drive a flow through the capillary pipe that connects the boxes at the bottom. A compensating flow at the surface ensures that the volume of water in each box does not change. Together, these two exchanges (top and bottom) represent the overturning circulation of the THC.

1. We assume (see above) that the strength of the exchange flow between the boxes is linearly proportional to their differences of temperature and salinity. In particular, let and be the two densities in the corresponding boxes. The flow in the capillary pipe is driven by the difference

* where is the hydraulic constant and takes the typical value of . The reference density is defined in conjunction with the equation of state.
* In the normal state of the planet, when the flow through the capillary goes in the direction of the equator as a result of higher densities at high latitudes.

1. We need an equation of state that connects the densities, , with the temperature and salinities. This concerns the figure [Figure 23.3](#fig-salinity). We assume that

* where and are average values of temperature and salinity; is the density if and ; and are constants with typical values and . Combining ([Equation 23.2](#eq-ocean-q)) with ([Equation 23.3](#eq-ocean-state)) gives

1. External wind forces and Coriolis effects are ignored.
2. We assume that in each box, there is an exchange of heat and salinity to the surrounding environment. For instance, salinity will exchange due to evaporation, precipitation, and runoff. These will have constant of proportionalities of and in the equations below.
3. Precipitation, evaporation, and runoff from the continents and atmosphere will also cause the salinity in either box to change. We therefore assume that this is modeled by a flux .

* Remember that in general, (the temperatures at the equator are higher). Consequently, evaporation dominates Box 2; there is a net loss of salt-free moisture in the atmosphere and this causes a compensating virtual flux of salt, , into Box 2. At the same time, there is higher precipitation and runoff into Box 1 (due to lower temperatures), causing a net gain of salt-free moisture and therefore a loss of virtual salt, say . We assume these two virtual salt fluxes sum to zero.
* Note that in the normal state of affairs, , and the lower latitude water is saltier than that of high latitudes.

An image of the box model is shown below.

|  |
| --- |
| Figure 23.4: Two-box model of the North Atlantic with evaporation and precipitation. Image from (Kaper and Engler 2013) |

The equations are given as follows.

Note that in the exchange terms corresponding to the flow in the capillary pipes, we use an absolute value since it does not matter (flows are exchanged in both lower and upper pipes).

Note also that as given in ([Equation 23.4](#eq-ocean-qfinal)).

## 23.6 Reducing the equations via symmetry

By adding the equations ([Equation 23.5](#eq-2box)), it can be verified that in the limit , the average temperature in the system, tends to the average temperature in the surrounding basins . The same conclusion can be drawn for the corresponding salinities.

It is then sensible to write all temperature and salinity in terms of this baseline scenario. So let us write

where . Then the first equation becomes

where .

The analogous manipulations are done to the quantities for the salinity. In the end, if we (confusingly) re-write for , then we obtain

Comparing the above to [Equation 23.5](#eq-2box), the main difference is that, in expressing the temperature and salinities with respect to the average values in the basin, we have eliminated two of the constants from the set now only into two constants .

In the situation of zero salt flux. , the above model reduces to Stommel’s box model studied later in this course.

## 23.7 Analysis of a reduced 1D model for the salinity

It is possible to do some analysis on the system ([Equation 23.6](#eq-2box-again)), though we may prefer to simply move on to study the more historically well-known reduction called *Stommel’s box model*.

We make the following assumptions:

* We assume that on the timescale of interest in the THC, the temperature of each box equilibrates quickly with the surrounding basin. Therefore the system can be regarded as being in steady state.
* The difference in temperatures between the two boxes is small; together with the top assumption, this implies that and .
* Salinity exchanges by negligible amounts with its surrounding basin, i.e.

This leaves us with where .

Now, the formulation for the salinities can be placed into a single equation for , which satisfies

This is now a much simpler and elegant formulation for study. Under a suitable non-dimensionalisation and scaling, we can even place the above equation in the final simplified form of

where (for temperatures in the usual configuration, with , and a suitably defined function .

# 24. Stommel’s box model

In addition to the model studied in [Section 23.7](#sec-ocean-toysalinity), there is another kind of reduced model that is examined in this chapter.

In his work (Stommel 1961) did not account for evaporation or precipitation, so there is no virtual salt flux, , but each box does exchange salinity with its surrounding environment, so . The set of four differential equations presented in [Chapter 23](#sec-oceanbox) ([Equation 23.6](#eq-2box-again)) can be reduced two coupled differential equations for the mean temperature and salinity. We notice that if we subtract the pairs of equations, then we can write

where , , and we have defined the two new unknowns,

and now, the flow is given by

## 24.1 Non-dimensionalisation

We can then nondimensionalise the system by setting

where is nondimensional time. Dropping the primes henceforth, we have the following set of non-dimensional equations to study for the unknowns, , and :

where we have introduced the function,

where there are now three non-dimensional parameters given by

Together [Equation 24.2](#eq-stommelsys) and [Equation 24.3](#eq-stommelf) form a system of equations for , with parameters , , and .

Although there are many parameters, it is important to note that the quantity is crucial since it corresponds to the direction of flow in the bottom pipe of the Ocean model. By convention, the system was set up so that corresponds to flow through the bottom pipe goes in the direction of the equator (Box 1) as a result of higher densities at the high latitudes. We are subsequently interested in whether it is possible for to switch sign, which would correspond to a bottom flow going from equator to poles, and the entire system switching direction.

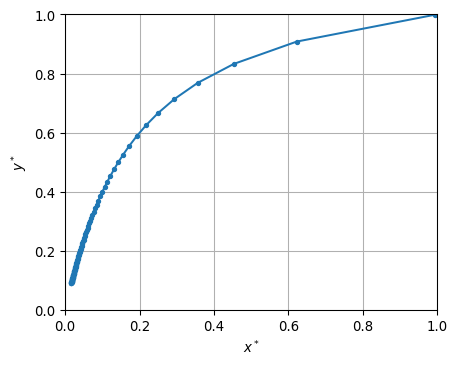
## 24.2 Equilibrium states

Let the equilibrium states be given by with . Then setting the right hand sides of ([Equation 24.2](#eq-stommelsys)) to zero and solving, we find

We can solve for in both equations and equate the result to one another. This gives

Therefore the equilibrium points lie along the above curve. Let us generate different values of and plot the combination.

import numpy as np  
import matplotlib.pyplot as plt  
  
delt = 1/6  
fmat = np.linspace(1e-3, 10, 101)  
xs = delt/(delt + np.abs(fmat))  
ys = 1/(1 + np.abs(fmat))  
plt.plot(xs, ys, '.-')  
plt.xlim((0,1))  
plt.ylim((0,1))  
plt.xlabel("$x^\*$"); plt.ylabel("$y^\*$");  
plt.grid(1)



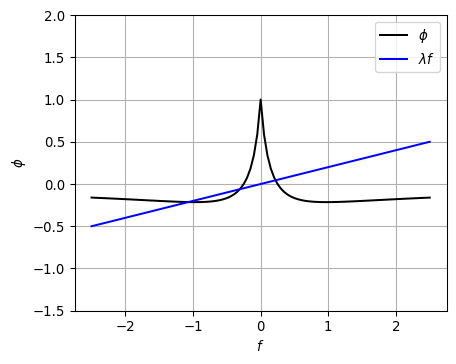
The above shows the steady-state values at . Each point refers to a different value of . Note that as , then while as , then .

Now the above does not tell us what equilibrium states will exist at a specific value of since it requires information about . We take ([Equation 24.3](#eq-stommelf)) and combine with ([Equation 24.4](#eq-stommel-xsys)). We conclude that the equilibrium points must satisfy

where we have defined the function according to

Therefore, for each combination of the parameters , we must solve to determine . Once the values of are known, then the steady-states are also known. Here is a typical graph showing the potential intersections at the values of , and .

import numpy as np  
import matplotlib.pyplot as plt  
import scipy.optimize as sciopt  
  
lam = 1/5;  
delt = 1/6;  
R = 2;  
  
f = np.linspace(-2.5, 2.5, 101)  
phi = delt\*R/(delt + np.abs(f)) - 1/(1 + np.abs(f))  
  
plt.plot(f, phi, 'k', label='$\phi$')  
plt.plot(f, lam\*f, 'b', label='$\lambda f$' )  
plt.ylim((-1.5,2))  
plt.xlabel("$f$")  
plt.ylabel("$\phi$")  
plt.legend()  
plt.grid(1)



Although there are many parameters that can alter the shape of the overall graphs, note that if is increased from its current value of , then two of the left roots will disappear, leaving only a single root.

## 24.3 Stability

We let and and linearise the system about the fixed points. This gives

where the matrix is given by

if .

There are two ways or proceeding. We can numerically substitute the fixed points into the above matrix for and then calculate the eigenvalues (numerically or otherwise). Or we can directly proceed analytically.

If proceeding analytically, we can then calculate the trace and determinant, giving

Using the above, we can analytically calculate the key discriminant expression of (or numerically) in order to verify stability. The details of the classification scheme is given in the appendix [Chapter 39](#sec-appendix-dyn).

For our purposes, we will mainly proceed using numerics, and bypass the need to study the above results analytically.

[In the accompanying lecture script](https://maths.jupyterhub.bath.ac.uk/hub/user-redirect/lab/tree/courses/MA30287/notebooks/lecture29-StommelPhasePlane.ipynb), we will obtain the eigenvalues numerically, and generate the phase plane. At the test point , , and , we generate the following phase plane picture.

|  |
| --- |
| Figure 24.1: Phase plane for , and . The blue, red, and green fixed points correspond to , , and in order. The blue point is a stable node, the red point is an unstable saddle, and the green point is a stable spiral. |

## 24.4 Bifurcation diagrams

Once the above stability analysis, it is possible to sketch the final bifurcation diagram.

First, we can note that as increases, the graph in [Section 24.2](#sec-ocean2-equil) indicates that there may either be three intersections (as shown) or if increases then two of the intersections may disappear. We can note that:

|  |  |  |
| --- | --- | --- |
|  |  | none |
|  | fixed point < 0 | none |
|  | fixed point > 0 | 0 |

We can also ascertain that there should be a point, say where the two roots coalesce, , and thereafter disappear. This allows us to now plot the bifurcation -plane:

|  |
| --- |
| Figure 24.2 |

## 24.5 Stommel’s conclusion

What is the final takeaway message(s)?

1. It is possible to pose a conceptual model of the ocean and the thermohaline circulation, which governs the exchange of hot-and-cold waters and salty waters from equatorial-to-pole zones.
2. This conceptual model suggests that tipping points and hysteresis are possible in such systems. In particular, it is possible, according to these models, for the flow directions to entirely change!

|  |
| --- |
| **As noted by (Kaper and Engler 2013):**  “Clearly, two-box models are only a caricature of the THC. At best, they account for the pole-to-equator circulation in single ocean basin (the North Atlantic). They certainly do not account for the fact that all the Earth’s oceans are connected, nor for the fact that the oceans are coupled to the atmosphere and other components of the climate system. Nevertheless, the finding that such simple models predict the possibility of two distinct stable modes of circulation, and that transitions from one more to another can be induced by changing the forcing parameters [–] has had a significant impact in oceanography and climate science.” |

# 25. Box models for flood estimation

# 26. Problem set 1

|  |
| --- |
| For feedback, hand in by Friday Week 3. |

The intention of this problem set is to practice concepts from material related to conservation laws and non-dimensionalisation. Although these concepts seem quite separate from “Maths of Planet Earth”, actually, they form important pillars of mathematical modelling.

## Q1. Bump lemma

Prove the following one-dimensional lemma, which was used in the derivation of the heat equation.

If for all and , then for all .

*Hint:* think of a proof by contradiction.

## Q2. A source in the heat equation

Consider the same heat experiment discused in [Chapter 1](#sec-intro-PDEs) but now consider a bar that has an internal source or sink generating or removing heat, such as the case of a boiler with an internal heating element. By adapting a similar derivation to the one presented, explain why the modified conservation of heat equation is

$$
\frac{\mathrm{\partial}}{\mathrm{\partial}t} \int\_a^b \rho c T \, \mathrm{d}x = {\color{blue}q(x = a, t) - q(x = b, t)} + \int\_a^b R(x, t) \, \mathrm{d}x.
$$

In addition:

1. By studying the dimensions of the other terms in the above equation, find what the dimensions of are. What does mean and ?
2. Hence derive the partial differential equation that governs the temperature .
3. By introducing the appropriate scalings on each of the variables, , , and , non-dimensionalise the PDE and discuss the non-dimensional parameters (there will be two).

## Q3. Choice of scalings

Consider the dimensional problem for the motion of a projectile launched from close to the surface of the Earth. The dimensional height of the projectile, , is given by Assume that the Earth is spherical and of uniform density, with its mass given by . Non-dimensionalise the height using and time using . Consider the following cases:

1. fixed, , fixed;
2. fixed, fixed, ;
3. fixed, fixed, ;
4. , fixed, fixed.

For each case:

1. Explain the physical interpretation of the limits.
2. Choose the scalings and to normalise as many terms as possible.
3. Choose the scalins so that the time it takes for the projectile to fall should be finite for the given limit, and for the speed, acceleration, and initial height to be well behaved (finite).
4. Write out the scaled problem and identify all remaining nondimensional parameters.
5. Identify the limiting small parameter for each case. Write out the problem (leading-order problem) when the parameter is set to zero.

## Q4. The unique timescale in the heat equation

During our investigation of the heat equation, we found that it was possible to scale time so as to scale out the only non-dimensional parameter that appears in the PDE (). This produced ([Equation 2.1](#eq-heat-nondim)). The disappearance of all non-dimensional parameters is due to the fact that only a single sensible timescale exists.

By adjusting the boundary conditions, we may create a new problem involving heat flow where a unique ‘special’ timescale can no longer be chosen.

Consider a system where one side of the rod is heated in some periodic fashion, e.g. set the initial and boundary conditions to be

1. What must the units of be?
2. Non-dimensionalise as usual and, without selecting the timescale, , identify the key non-dimensional parameters that remain. Write a brief sentence to describe their physical interpretation.
3. There are two sensible choices for setting the timescale, . Identify the two choices and present the reduced set of equations in each case.

## Q5. Timescale in the surface energy

Take the basic zero-dimensional energy model studied in ([Equation 3.7](#eq-heateq-earth)) for the temperature of the troposphere:

1. Non-dimensionalise the model by choosing , , and . Show that it is possible to select the scalings on the temperature and time so as to completely remove all constants from the problem when is assumed to be constant.
2. Thus, show that the analysis of the above equation is equivalent to studying

* where we have dropped the primes and assumed that the albedo is such that .

1. From your choice of , estimate the typical dimensional value using , , .  
     
   Use a pocket calculator to verify your calculations and conclude that this time-scale is on the order of a month. What is the relevance of this approximation as it concerns the steady-state solution?

# 27. Problem set 2

|  |
| --- |
| For feedback, hand in by Friday Week 4. |

The intention of this problem set is to practice concepts from [Chapter 5](#sec-asymptotics01) (asymptotic approximations of algebraic equations) to [Chapter 7](#sec-euler) (Euler’s method and numerical solutions of differential equations). These techniques form some of the most powerful techniques at your disposal in applied maths.

## 27.1 Getting started with Noteable

Use your [Moodle course page](https://moodle.bath.ac.uk/course/view.php?id=60108) to access the Noteable Python interface by clicking on the appropriate link in the course materials. Follow the instructions on the Moodle page to add the Git repository to your directory.

1. Navigate to the welcome screen for MA30287 and follow the instructions in Welcome.ipynb and FirstTimeSetup.ipynb in order to create your own local directory at /MA30287\_workspace/
2. Navigate to the workspace folder. Click New -> Notebook. If asked, select the kernel Python 3 (ipykernel).
3. In the first line of input, select, in toolbar Code -> Markdown. This allows you to annotate your notebook with Markdown-style text input.

* In the first line of input, type # Problem set 2. Then either type Shift + Enter or press the play button which will execute the line(s) of input. Your markdown text should render as a nicely formatted entry.

1. Rename the file to something appropriate, like ps02\_scripts. You can do this by right-clicking the filename in the file manager and selecting Rename.

* Proceed to the next question.

## 27.2 Testing the solutions of a cubic

In this question, you will develop the numerical solutions for the roots of the cubic equation:

Type the following code into your Jupyter notebook. This code uses a command, which you will learn in a later week, called fsolve, in order to solve nonlinear equations. When writing the below code, you may want to separate the import commands into their own input field in the notebook.

import numpy as np  
from scipy.optimize import fsolve  
  
ep = 0.1 # epsilon value  
xguess = 1.1 # Initial guess of root  
  
f = lambda x: ep\*x\*\*3 - x + 1  
xsol = fsolve(f, xguess)  
print("Solved root at x = ", xsol)

Solved root at x = [1.15346731]

## 27.3 Analysis of singular cubic equation

Consider the cubic equation

with and .

1. Develop the first three terms of an asymptotic expansion about the root by setting
2. Fill out the following table.

|  |  |  |
| --- | --- | --- |
| 0.1 | – | – |
| 0.08 | – | – |
| 0.06 | – | – |
| 0.04 | – | – |
| 0.02 | – | – |

* Use your code in [Section 27.2](#sec-ps2-cubic) to input numerical approximations to the above entries.
* Create a graph **by hand** of the data (it does not have to be extremely accurate), as plotted in the -plane. Fit a line to this graph and estimate the gradient. Is this consistent with what you derived above?

1. By rescaling appropriately in terms of , derive the first three terms of the asymptotic approximations of the remaining roots.

## 27.4 A damped projectile problem

In [Chapter 6](#sec-asymptotics02) you performed the asymptotic analysis for a projectile. The small parameter was and represented (a parameter that includes the initial velocity, , gravity , and the radius of the Earth, .

If air resistance is included, then the non-dimensional toy model is instead where is the parameter that controls air resistance.

1. Begin by assuming that is a fixed number and consider the limit where . Find a **one-term** asymptotic expansion of the solution for small .
2. (**Challenging**) Is the effect of the air resistance to increase or decrease the flight time? Justify based on your analytical solution.

## 27.5 ODE solvers and Euler’s method

Return to the setup of the above question.

1. Modify the script shown in [Section 6.2](#sec-asym2-num) in order to solve the initial-value problem from the previous question at a prescribed value of and .
2. Using **a pocket calculator** (or your phone calculator) apply Euler’s method with , , and to determine the position of the projectile at .
3. Compare your hand calculation with the result from the Python output, as well as with your asymptotic approximations.

# 28. Problem set 3

|  |
| --- |
| For feedback, hand in by Friday Week 5. |

In [Chapter 8](#sec-asymptotics03) we discussed how the zero-dimensional energy equation occurs in the form of

This is oversimplistic since in actuality, your temperature function, , should also depend on space. For instance, if you are interested in modelling with a depth coordinate, then it would be . In this case, we know that temperature would be governed by a second-order differential equation (in space) such as the kind that appears in ([Equation 1.1](#eq-heatintro)).

In this problem set, we will make up a toy example of a differential equation that might be encountered in heat flow. Consider now (in non-dimensional coordinates):

which is determined for some function . Here, the small parameter , , is linked to the heat diffusion.

We will study the steady-state version of the problem. So consider only . We will make up some boundary conditions as well.

The challenge is to study the above problem for small values of .

## Q1. Conversion to first-order system

By using the procedure reviewed in [Section 6.2](#sec-asym2-num), convert ([Equation 28.1](#eq-ps3-sys)) to a first-order system of equations.

## Q2. Numerical solutions

By adapting the code studied in lectures (script lecture12-SolvingBVPs), write a numerical code to solve ([Equation 28.1](#eq-ps3-sys)) using Python’s built-in functions. Use your code to investigate the solution profiles for different values of .

## Q3. Investigation of the boundary layer

Using your code, use the following command in order to investigate the maxima as varies:

ind = np.argmax(Y[0])  
print(ep, z[ind], Y[0, ind])

You may want to consider, as an example, the values and fill the following table.

|  |  |  |
| --- | --- | --- |
| 0.05 |  |  |
| 0.1 |  |  |
| 0.15 |  |  |
| 0.2 |  |  |

Create a plot of and discuss the observed trend and its implications.

## Q4. Outer asymptotic solutions

Begin by setting

Substitute the above expansion into the system and solve for the first two orders.

You may verify that the solution is given by

## Q5. Inner asymptotic solutions

There will be a boundary layer near . Set and . Follow the same procedure, as in [Chapter 8](#sec-asymptotics03) in order to determine the correct choice of for the inner region. This choice should balance the two terms and .

## Q6. Matching and comparison

Expanding the inner solution as , write down the equation and boundary conditions that must satisfy. You will notice that is governed by a second-order differential equation and therefore needs two boundary conditions. One boundary condition comes from , i.e.

The other boundary condition is a matching condition:

which imposes that the inner solution, as it leaves the boundary layer, matches the outer solution, it tends into the inner region.

Solve for .

# 29. Problem set 4

|  |
| --- |
| For feedback, hand in by Friday Week 6. |

## Q1. The wine cellar problem I

Within [Chapter 12](#sec-pclass3), the heat system was studied:

for . The initial condition can be ignored.

The solution was sought in the form of using a separation of variables procedure leading to

Go through an argument that the cases of and real lead to incompatible solutions. Review the calculation that leads to the desired solution given in ([Equation 12.1](#eq-wine-sol)).

## Q2. The wine cellar problem II

Review the code written in lecture15-winecellar.ipynb.

Verify how well the numerical solution matches the exact analytical solution by overlaying the numerical solution with the analytical solution.

Investigate the solutions under different choices of parameters and time-stepping tolerances until you are confident you know how the numerical algorithm works.

## Q3. EBMs with variable sun

Satellite data indicates that , varies roughly between 341.37 W/m2 and 341.75 W/m2, with a period of about 11 years.

1. Use the simple EBM given in ([Equation 13.3](#eq-energy-ss)) and repeated below:

* with a constant albedo, and greenhouse gas factor to estimate the resultant variation (max and min) in the Earth’s mean surface temperature .

1. Similar to (a) but this time, use the Budyko balance equation,

* with and to estimate the resultant variation in the surface temperature. Use .

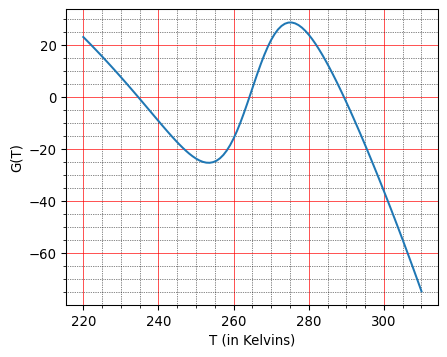
1. The actual variation in surface temperature is in fact less than what you computed above. Why might this be?

## Q4. EBMs and phase line analysis

Consider the energy balance equation

with given by ([Equation 13.2](#eq-albedokaper)). Because the differential equation is autonomous, we can apply phase-line analysis in order to qualitatively understand the evolution. Below is a plot of the function :

import numpy as np  
import matplotlib.pyplot as plt  
  
Q = 342; sigma = 5.67e-8; gam = 0.6;  
a = lambda T: 0.5 - 0.2\*np.tanh(0.1\*(T-265));  
T = np.linspace(220, 310, 100);  
G = Q\*(1-a(T)) - sigma\*gam\*T\*\*4  
fig, ax = plt.subplots()  
ax.plot(T, G)  
ax.grid(); ax.minorticks\_on();  
# Customize the major grid  
ax.grid(which='major', linestyle='-', linewidth='0.5', color='red')  
# Customize the minor grid  
ax.grid(which='minor', linestyle=':', linewidth='0.5', color='black')  
plt.xlabel('T (in Kelvins)'); plt.ylabel('G(T)');



1. Sketch the solution of this equation for if and .
2. Sketch the solution of this equation for if . Then sketch the solution of this equation with the same initial data in the same coordinate system if is twice as large. Explain your answer.
3. If is decreased due to the increased greenhouse effect, the entire curve is shifted upwards. Sketch the solution if . Sketch the solution with the same initial data if is decreased. Explain your answer.

# 30. Problem set 5

|  |
| --- |
| For feedback, hand in by Friday Week 7. |

## Q1. Evolution

Consider again the basic time-dependent EBM given in ([Equation 29.1](#eq-ps4-main)). Let be a steady-state solution and set where is a small perturbation from the steady state.

1. Show that the perturbation satisfies

* and hence solve for the general solution of the leading-order perturbation (ignoring quadratic terms). What are the conditions on so that the steady state is linearly stable?

1. Assuming is linearly stable, find the typical response time to a perturbation. For instance, what is the time it takes for the perturbation to reach the value if ? How does this response time change with ? What is the physical interpretation of this regarding the climate?

## Q2. Integral of energy over the planet

Ignoring the effects of albedo, the total radiation absorbed over the surface of the planet (per unit time) is given by

This is what is known as a surface integral ([Section 40.1](#sec-appendix-surfaceintegral)). In the case of the spherical coordinate system, this is calculated by

Use the properties of in ([Equation 14.3](#eq-EBM-s)) to conclude that the total radiation absorbed is .

## Q3. Mean temperature in the latitude-dependent EBM

Consider now the latitude-dependent EBM

Recall the albedo is given by for and for .

1. By integrating the above equation over , show that the mean temperature is given by

* where

1. In the case that is given by ([Equation 14.3](#eq-EBM-s)), show that

* What is in the two situations of a completely ice-covered world and an ice-free world?

## Q4. Sensitivity of the climate

Consider the equation for the global average temperature given in ([Equation 30.1](#eq-meanT-evo-ps)):

We would like to understand how the climate behaves under a small perturbation in the solar forcing. In general, though, the ‘constants’, , , and will depend on . For example, their values may have been calibrated under a static situation. Therefore, let , , and .

Below, we drop all bars for convenience.

1. Consider a perturbation of the solar radiation, say where is small in comparison to . Expand now the temperature into a series:

* Show that at , the perturbation is governed by

1. Consequently, show that the temperature perturbation can be written as

* where

The parameter measures the time scale of the climate system’s thermal inertia. It involves the thermal inertia of the atmosphere and also the much larger inertia of the oceans. The factor is the climate gain; it amplifies any response to the radiative perturbation by a factor of . Also incorporates the effect of water-vapor feedback and that of ice and snow albedo feedback.

1. Consider ([Equation 30.3](#eq-T1eqpertb)) at steady state, so therefore the perturbed equilibrium temperature is equal to

* If the CO2 level in the atmosphere doubles, then the radiative forcing might be adjusted as:
* Assuming that the climate gain is and , what is the expected increase in temperature?

# 31. Problem set 6

|  |
| --- |
| Problem sets 6-7 will not be released until after the Easter break. |

|  |
| --- |
| To release for 2023-24. |

# 32. Problem set 7

|  |
| --- |
| Problem sets 6-7 will not be released until after the Easter break. |

|  |
| --- |
| To release for 2023-24. |

# 33. Problem set 1 solutions

## Q1 The bump lemma

Prove the following one-dimensional lemma, which was used in the derivation of the heat equation.

If for all and , then for all .

*Hint:* think of a proof by contradiction.

|  |
| --- |
| **Solution**  Assume by contradiction that between two arbitrary points, say and , both in the interval . Then . This assumes some degree of smoothness and integrability (as long as is not too pathological). |

## Q2. A source in the heat equation

Consider the same heat experiment discussed in [Chapter 1](#sec-intro-PDEs) but now consider a bar that has an internal source or sink generating or removing heat, such as the case of a boiler with an internal heating element. By adapting a similar derivation to the one presented, explain why the modified conservation of heat equation is

In addition:

1. By studying the dimensions of the other terms in the above equation, find what the dimensions of are. What does mean and ?

|  |
| --- |
| **Solution**  On the left, density has units kg/m^3, specific heat J/(kg K), and temperature K. But then dx has units of m. And d/dt has units of 1/s. So altogether, we have the following SI units on the left:  So its energy per unit area per unit time. The reason why it’s per unit area is because you have not integrated over the sides of the cylinder/bar.  So therefore this needs to match the units on the right, which are [R] [dx]. So you conclude that  i.e. energy per unit time per unit volume. This is energy density.  R > 0 means heat is added into the system (source). R < 0 means heat is removed from the system (sink). |

1. Hence derive the partial differential equation that governs the temperature .

|  |
| --- |
| **Solution**  The derivation follows exactly as in the notes, except you have an additional term on the RHS. Use Fourier’s law to convert . Write everything under the integral and this yields: |

1. By introducing the appropriate scalings on each of the variables, , , and , non-dimensionalise the PDE and discuss the non-dimensional parameters (there will be two).

|  |
| --- |
| **Solution**  Same derivation as in the notes. Let  This time you will remember to scale out the typical source strength, so  This leads to  So this gives the two sets of non-dimensional numbers  If you want, you can write it in the alternative form by taking the time scaling to be , where is some unspecified velocity scale.  For the interpretation, you can simply take to be a measure of diffusive effects (heat spreading out) and to be a measure of source strength. Note that this problem can be simplified even further by choosing the time scale so as to set . If you do this, you will be left with a single non-dimensional parameter that balances diffusion with the strength of the source. |

## Q3. Choice of scalings

Consider the dimensional problem for the motion of a projectile launched from close to the surface of the Earth. The dimensional height of the projectile, , is given by Assume that the Earth is spherical and of uniform density, with its mass given by . Non-dimensionalise the height using and time using . Consider the following cases:

1. fixed, , fixed;
2. fixed, fixed, ;
3. fixed, fixed, ;
4. , fixed, fixed.

|  |
| --- |
| **NB!**  This problem ended up a lot longer than expected. Nevertheless do at least of the few cases to get a feel. |

For each case:

1. Explain the physical interpretation of the limits.

|  |
| --- |
| **Solution**   1. Very fast initial velocity 2. Very heavy object 3. Very light object 4. Very small Earth radius |

1. Choose the scalings and to normalise as many terms as possible.
2. Choose the scalings so that the time it takes for the projectile to fall should be finite for the given limit, and for the speed, acceleration, and initial height to be well behaved (finite).
3. Write out the scaled problem and identify all remaining nondimensional parameters.
4. Identify the limiting small parameter for each case. Write out the problem (leading-order problem) when the parameter is set to zero.

|  |
| --- |
| **Solution**  The solution for parts (b) to (d) is given below. The solution for part (e) is simply setting in each case. |

## Q4. The unique timescale in the heat equation

During our investigation of the heat equation, we found that it was possible to scale time so as to scale out the only non-dimensional parameter that appears in the PDE. As explained in lectures, the disappearance of all non-dimensional parameters is due to the fact that only a single sensible timescale exists.

By adjusting the boundary conditions, we may create a new problem involving heat flow where a unique ‘special’ timescale can no longer be chosen.

Consider a system where one side of the rod is heated in some periodic fashion, e.g. set the initial and boundary conditions to be

1. What must the units of be?

|  |
| --- |
| **Solution**  Note must be unitless. Therefore . |

1. Non-dimensionalise as usual and, without selecting the timescale, , identify the key non-dimensional parameters that remain. Write a brief sentence to describe their physical interpretation.

|  |
| --- |
| **Solution**  This is exactly the same as in the above heat question. You will still obtain  where the non-dimensional parameters is the Peclet number introduced in lectures:  The main difference is that now, your boundary conditions are  where  The scaling on the temperature was taken to be such that the final initial condition is set to unity: |

1. There are two sensible choices for setting the timescale, . Identify the two choices and present the reduced set of equations in each case.

|  |
| --- |
| **Solution**  So there are now two ways of choosing the time scale. Either choose it to set or choose it to set . In the former choice, you are choosing to set time according to the diffusive rate. In the latter, you are setting time according to how the boundary is heated. |

## Q5. Timescale in the surface energy

Take the basic zero-dimensional energy model studied in ([Equation 3.7](#eq-heateq-earth)) for the temperature of the troposphere:

1. Non-dimensionalise the model by choosing , , and . Show that it is possible to select the scalings on the temperature and time so as to completely remove all constants from the problem when is assumed to be constant.

|  |
| --- |
| **Solution**  The question’s suggestion of setting was excessive since is just a constant in this case. So proceeding as usual:  Set in the scalings. This time it is easier to balance the right had-side first.  So select the temperature scaling so that both coefficients on the RHS are equal.  Now choose the time scaling also to match the LHS coefficient to the RHS coefficient.  Once you have done this, the equation transforms to the one given in part b. |

1. Thus, show that the analysis of the above equation is equivalent to studying

* where we have dropped the primes and assumed that the albedo is such that .

|  |
| --- |
| **Solution**  Done above. |

1. From your choice of , estimate the typical dimensional value using , , .  
     
   Use a pocket calculator to verify your calculations and conclude that this time-scale is on the order of a month. What is the relevance of this approximation as it concerns the steady-state solution?

|  |
| --- |
| **Solution**  Below ([Equation 3.7](#eq-heateq-earth)), we defined . So the scaling on time is (put in the scaling for [T]):  So we know that J/(m^2 s K^4), J/(m^2 s). We can also calculate  We can use for an estimate. The right-most factor is  If we use a = 0.3 then this gives a total of  To convert to days, note 1 day = 24 hours = 24*60 minutes = 24*60\*60 seconds. This is then  So on the order of 6 months. Any answer you get that’s roughly on the order of months sounds about right. This is the rough approximation of how long time-scale effects take to appear. |

# 34. Problem set 2 solutions

## 34.1 Getting started with Jupyter

|  |
| --- |
| N/A |

## 34.2 Testing the solutions of a cubic

|  |
| --- |
| See notebook/ps2\_sol.ipynb notebook for the code. |

## 34.3 Analysis of singular cubic equation

Consider the cubic equation

with and .

1. Develop the first three terms of an asymptotic expansion about the root by setting

|  |
| --- |
| **Solution**  Substitution into the cubic gives  Equating at each order and solving gives So the three-term approximation is |

1. Fill out the following table.

|  |  |  |
| --- | --- | --- |
| **Solution**  | $\epsilon$ | $x\_{\text{exact}}$ | $x\_{\text{exact}} - x\_0$ | | -----------| -------------------|--------------------------| | 0.1 | 1.1535 | 0.1535 | | 0.08 | 1.1092 | 0.1092 | | 0.06 | 1.0744 | 0.0744 | | 0.04 | 1.0457 | 0.0457| | 0.02 |1.0213 | 0.0213 |  This is a really excellent demonstration of an ‘organic’ discovery process. Below we plot the errors we found in the above table. I rounded the values to only the first two decimals. It is not so important to be extremely accurate (in fact, this is the point of requiring you to do so by hand!)   |  | | --- | | Figure 34.1: Hand drawn image |   When doing this, I tried to fit to a line because I knew the error was supposed to be  so was expecting a line of unit gradient. But the gradient estimated above was a bit higher, at 1.6. Returning to Python, we see the problem.   |  | | --- | | Figure 34.2: Python image |   So over the range investigated of , the behaviour is quadratic.This is a nice lesson on how to discover new results based on computation and analysis. |

1. By rescaling appropriately in terms of , derive the first three terms of the asymptotic approximations of the remaining roots.

|  |
| --- |
| **Solution**  We expect the critical scaling to balance with since these two will be dominant when is large. So we set  Now we have  You will notice that the progression here should go in powers of . Alternatively set and now the problem is standard in :  And we expand  We get So it seems that the three-term approximation for the other two roots is |

## 34.4 A damped projectile problem

If air resistance is included, then the non-dimensional toy model is instead where is the parameter that controls air resistance.

1. Begin by assuming that is a fixed number and consider the limit where . Find a one-term asymptotic expansion of the solution for small .

|  |
| --- |
| **Solution**  First, note that  You will not need more terms than that. Substitute into the equation. This gives, for the leading terms To solve this, you will need to refresh your memory on second-order differential equations learned in 2nd year. You can see [Chapter 38](#sec-odes) for references. To solve this set and solve for for the homogeneous equation. You will get  From here, we need a particular solution. This can be obtained by guessing (also see [this reference](https://tutorial.math.lamar.edu/Classes/DE/UndeterminedCoefficients.aspx)). If we guess  we see that  implies . Therefore the general solution is  We now select the constants to satisfy the initial conditions. This yields  from which the solution can be easily derived. It is  The solution of proceeds in the same way, but since we did not anticipate that this question would be so algebraically involved, we will not require solving for . This can be left as an exercise if you so wish. |

1. (**Challenging**) Is the effect of the air resistance to increase or decrease the flight time? Justify based on your analytical solution.

|  |  |
| --- | --- |
| **Solution**  The above solution seems quite strange. After all, when , we know that parabolic motion is expected. We found previously that  which matches our intuition about the expected flight being parabolic. However, notice that if is small, we can expand the exponential, . Substitution into the derived formula for gives  $$ \color{blue}y(t) \sim \left(\frac{1}{\alpha^2} + \frac{1}{\alpha}\right) - \left(\frac{1}{\alpha^2} + \frac{1}{\alpha}\right)\left[1 - \alpha t + \ldots\right] - \frac{t}{\alpha}. $$  We can now simplify the expression and we see that the limit is not problematic, because it now becomes:  and the previous terms that we believed might tend to infinity as cancel. Inspecting the above solution, notice that the leading-order result agrees with the previously derived result for . The next term shows the small effect of drag.  You can simply plot the curves and show the difference.   |  | | --- | | Figure 34.3: With (solid) and without damping (dashed) | |

Notice that the dashed line brings down the parabolic path initially.

## 34.5 ODE solvers and Euler’s method

Return to the setup of the above question.

1. Modify the script shown in [Section 6.2](#sec-asym2-num) in order to solve the equation from the previous question at a prescribed value of and .

|  |
| --- |
| **Solution**  This is found in problemsets/sol02\_solutions.ipynb. |

1. Using **a pocket calculator** (or your phone calculator) apply Euler’s method with , , and to determine the position of the projectile at .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Solution**  Euler’s method is simply  Here  We split the time steps into {0, 0.2, 0.4, 0.6}. We also round to two figures.   | t | Y |  | F |  | | --- | --- | --- | --- | --- | | 0 | (0, 1) | 1 | (1, -2) | (0.2, -0.4) | | 0.2 | (0.2, 0.6) | 1.04 | (0.6, -1.5) | (0.12, -0.3) | | 0.4 | (0.32, 0.3) | 1.06 | (0.3, -1.16) | (0.06, -0.23) | | 0.6 | (0.38, 0.07) |  |  |  |   After three iterations, we get a value of . |

1. Compare your hand calculation with the result from the Python output, as well as with your asymptotic approximations.

|  |
| --- |
| **Solution**  We have (rounded to two digits) |

# 35. Problem set 3 solutions

The challenge is to study the above problem for small values of .

## 35.1 Conversion to first-order system

By using the procedure reviewed in [Section 6.2](#sec-asym2-num), convert ([Equation 28.1](#eq-ps3-sys)) to a first-order system of equations.

|  |
| --- |
| **Solution**  Introduce and . Then |

## 35.2 Numerical solutions

|  |
| --- |
| **Solution**  A script can be found in problemsets/ps03\_solutions.ipynb |

## 35.3 Investigation of the boundary layer

Using your code, use the following command in order to investigate the maxima as varies:

ind = np.argmax(Y[0])  
print(ep, z[ind], Y[0, ind])

You may want to consider, as an example, the values and fill the following table.

|  |  |  |
| --- | --- | --- |
| 0.05 | 0.1122 | 1.5479 |
| 0.1 | 0.1924 | 1.4742 |
| 0.15 | 0.2605 | 1.4094 |
| 0.2 | 0.3226 | 1.3501 |

Create a plot of and discuss the observed trend and its implications.

|  |  |
| --- | --- |
| **Solution**   |  | | --- | | Figure 35.1: A plot of the above data. This looks linear, and indeed this provides numerical confirmation that the boundary layer should be sized with | |

## 35.4 Outer asymptotic solutions

Begin by setting

Substitute the above expansion into the system and solve for the first two orders.

You may verify that the solution is given by

|  |
| --- |
| **Solution**  **Remember**: as you solve the outer solution, you impose the boundary condition of .  **Leading-order**: at leading order, we get the problem subject to . This is solved either by separation of variables or by integrating factor. Using the technique of solving first-order linear equations, (see [Chapter 38](#sec-odes)) we divide,  We then multiply both sides of the equation by .  $$ \frac{\mathrm{d}}{\mathrm{d}z}\Bigl[T\_0 e^{z/2}\Bigr] = 0. $$  Integrate, and solve, giving . We now impose which yields .  **First-order**: at next order, we get  along with the boundary conditions of .  Again place this into the appropriate form. Again, this is a linear first-order equation. We multiply by the integrating factor , placing the equation into the form  $$ \frac{\mathrm{d}}{\mathrm{d}z} \Bigl[ T\_1 e^{z/2} \Bigr] = -\frac{1}{2 }T\_0'' e^{z/2} = -\frac{1}{8} e^{1/2}. $$  We integrate once and simplify, yielding  and then impose the condition that . This yields hence |

## 35.5 Inner asymptotic solutions

There will be a boundary layer near . Set and . Follow the same procedure, as in [Chapter 8](#sec-asymptotics03) in order to determine the correct choice of for the inner region. This choice should balance the two terms and .

|  |
| --- |
| **Solution**  We re-scale and let . Then the ODE becomes  Balancing the first two terms thus requires that  Thus the differential equation now becomes |

## 35.6 Matching and comparison

Expanding the inner solution as , write down the equation and boundary conditions that must satisfy. You will notice that is governed by a second-order differential equation and therefore needs two boundary conditions. One boundary condition comes from , i.e.

The other boundary condition is a matching condition:

which imposes that the inner solution, as it leaves the boundary layer, matches the outer solution, it tends into the inner region.

|  |
| --- |
| **Solution**  Solve for .  We thus expand . To leading order, we have  Integrating, we find that the general solution is given by  We require two boundary conditions. The first condition is given by and hence . The second condition is given by requiring that the inner solution, as , matches the outer solution, as . Using our solution above, we find that  Thus the second boundary condition is  Together, both boundary conditions are used to conclude that  or in terms of , the inner solution is |

# 36. Problem set 4 solutions

## Q1. The wine cellar problem I

|  |
| --- |
| **Solution**  Separation of variables leads to the equation  Crucially in considering possible solutions, we want solutions to obey the necessary boundary condition of . This is highly restrictive.  We first consider . Then we have that . This is then no way to satisfy the time-varying boundary condition.  We next consider . Then . So again the behaviour of the time-dependent part is incompatible with the oscillating boundary conditions.  Therefore the only remaining possibility is where is real. Then we have that  Written in terms of real-valued functions, this gives essentially a linear combination of and . In order to be compatible with the boundary conditions, we would thus need  i.e. taking the imaginary part (if we wish to continue manipulating exponentials).  Now solving for , we set as a solution ansatz, giving the characteristic equation  Therefore the general solution for yields  In order to parse the above form, it is best to re-write the real and complex parts of the exponential argument separately. We note that by Euler’s theorem,  So now we can re-write  By the boundedness assumption, we require that does not blow up as . So therefore . We can finally put the results together, yielding:  Note that the reason you have re-written it with the exponentials arranged in this fashion is because the portion of the exponential with the imaginary argument becomes oscillatory. Taking the real part yields  You can double check that the above solution satisfies all the required conditions, including the condition at (forcing us to choose the prefactor ). |

## Q2. The wine cellar problem II

|  |
| --- |
| **Solution**  Solution script can be found on the solutions/ folder of the Noteable Python website.  The numerical solution is tricky for two reasons. First, you must consider the right range of parameters. Second, the numerical algorithm suffers from instability, so you must consider the right spatial and time discretisation.  To begin, from the notes [Chapter 12](#sec-pclass3), you are given that typical parameters are  This has already been converted to seconds, so instead use  Indeed this value of was chosen so that has a wavelength of 1 years.  Next, it is given that  We use the fact that  So  When distributing the typical mesh in the direction, we want to make sure we go deep enough. We had found that putting the cellar in about 4m deep was enough, so as long as your code solve for a range of values, say, between and , this should be sufficient.  The script is given in the solutions folder. |

## Q3. EBMs and variable sun output

1. Regarding the resultant variation on the Earth’s mean surface.

|  |
| --- |
| **Solution**  The variation seems to be very small. Ignoring units, we have  Substituting the values of given this gives a temperature that ranges from to . Converting this to Celsius gives to i.e. a fraction of a degree. |

1. Regarding the variation using the Budyko model.

|  |
| --- |
| **Solution**  We obtain the range of approximately to . |

1. Regarding why the disparity between actual surface measurements.

|  |
| --- |
| **Solution**  There are many ways to respond, but one major factor is the oceans, which are ignored in our model. The oceans provide a massive energy sink for the planet, and so we would expect that the minor variations of due to the 11-year cycle would be smoothed out by the effects of the oceans. |

## Q4. Phase line analysis

1. Sketch the solution of this equation for if and .

|  |  |
| --- | --- |
| **Solution**  You should be able to do this question by hand, but the following graph is generated via the accompanying Jupyter script in the solutions folder. The point is that once the steady-state solutions are known (dashed) then each solution given by the different initial condition can be approximated by whether it tends towards or away from the nearest fixed point.   |  | | --- | | Figure 36.1: Graphical solution for Q3a | |

1. Sketch the solution of this equation for if . Then sketch the solution of this equation with the same initial data in the same coordinate system if is twice as large. Explain your answer.

|  |
| --- |
| **Solution**  The below numerical solution shows blue for and orange for . Notice that increasing seems to decrease the rate of change of the evolution. Indeed, since  then multiplying by factor is equivalent to slowing down the evolution by the inverse of that factor. A factor of that is twice as large would slow down the evolution by half. Note that the oscillations shown in the graph below are numerical artifacts due to the tolerances on the ODE solver (how do you know this?)  Graphical solution for Q3b  Graphical solution for Q3b |

1. If is decreased due to the increased greenhouse effect, the entire curve is shifted upwards. Sketch the solution if . Sketch the solution with the same initial data if is decreased. Explain your answer.

|  |
| --- |
| **Solution**  Decreasing has the effect of shifting the curve upwards. The hottest steady state is consequently increased. There are two main effects. First, the steady-state is much hotter, so the system will tend towards a much hotter state. Second, because decreasing $ $ will increase the rate of change of (since we are subtracting less via the factor ), then the evolution towards the hotter state is initially much faster.  You see that in the following diagram.  Graphical solution for Q3c  Graphical solution for Q3c  The above makes sense on a physical level. Decreasing is equivalent to increasing greenhouse gases. We do expect the system to tend to a hotter state, then. |

# 37. Problem set 5 solutions

## Q1. Evolution

Consider again ([Equation 29.1](#eq-ps4-main)). Let be a steady-state solution and set where is a small perturbation from the steady state.

1. Show that the perturbation satisfies

* and hence solve for the general solution of the leading-order perturbation (ignoring quadratic terms). What are the conditions on so that the steady state is linearly stable?

|  |
| --- |
| **Solution Q1a**  We linearise via where . Then we have  The first term on the RHS is zero since is a steady state. Ignoring quadratic terms, we see then that  where . The solution is thus  where is a constant. In order to be linearly stable, we then need the above to tend to zero as , which is guaranteed if . Since then we only need |

1. Assuming is linearly stable, find the typical response time to a perturbation. For instance, what is the time it takes for the perturbation to reach the value if ? How does this response time change with ? What is the physical interpretation of this regarding the climate?

|  |
| --- |
| **Solution Q1b**  The question is somewhat ill-posed. By the behaviour of the system, we know that it would take infinite time for the perturbation, , to reach . However, we can still provide a sensible answer.  If the system starts from , then to reach , it would need to take the time  For instance, to reach (half its original perturbation) would take times (C/D).  This response is proportional to so for instance, doubling takes twice as long as before to reach the same value of . One can think of this as equivalent to decreasing the responsiveness of the planet to potential changes of stability. |

## Q2. Integral of energy over the planet

|  |
| --- |
| **Solution Q2**  This is a straightforward calculation without tricks. Since is constant, we have  Converting now to integration in , we have  Examining the function in [Equation 14.3](#eq-EBM-s) we note that it is symmetric in and has been defined so its integral is one. Thus  This is the solar constant multiplied by the surface area of the planet (a sphere). |

## Q3. Mean temperature in the latitude-dependent EBM

|  |
| --- |
| **Solution Q3a**  We have from integration,  Simplifying and using the property of gives  where we have defined |

|  |
| --- |
| **Solution Q3b**  Again this question is free of tricks and only involves the integration using  where and . It is useful to compute the two quantities  Multiply the first quantity by and add to the second multiplied by and simply to obtain |

## Q4. Sensitivity of the climate

1. Consider a perturbation of the solar radiation, say where is small in comparison to . Expand now the temperature into a series:

* Show that at , the perturbation is governed by

|  |
| --- |
| **Solution Q4a**  Each of the functions should be expanded in the usual way. For example, from Taylor series we have  Then using , we have  Do the same for both and . Then substituting into the equation gives (remember to drop bars):  while for the right hand-side  Now the first, , grouping of terms above is zero by assumption that is steady-state. So considering only those terms that are multiplied by , we have |

1. Consequently, show that the temperature perturbation can be written as

|  |
| --- |
| **Solution Q4b**  There is nothing tricky about this, but it just requires keeping track of bookwork. On the right hand-side, separate those terms proportional to and write it in the desired form. Refer to the lecture videos where we completed this (modulo a possible sign!) |

1. Consider ([Equation 30.3](#eq-T1eqpertb)) at steady state, so therefore the perturbed equilibrium temperature is equal to

* If the CO2 level in the atmosphere doubles, then the radiative forcing might be adjusted as:
* Assuming that the climate gain is and , what is the expected increase in temperature?

|  |
| --- |
| **Solution Q4c**  So we know that . So according to the equation, we have the fact that  So this is the expected increase in temperature. Note that if a climate gain factor of is used then this gives an expected increase of 1.9 degrees instead.. The range of current model predictions, in fact, is about 1.5 to 4.5 degrees so this is not bad. |

# 38. Differential equations

## 38.1 First-order linear differential equations

First-order linear differential equations are of the form

These are solved by integrating factors. One trick often employed us to think of placing the left hand-side of the equation into the form:

where the derivative of ?? yields .

Officially, to do this, we multiply both sides of the original equation by the integrating factor

so that the equation can be placed in the form

Then integrate and solve for . This gives

It is usually much easier to work with specific problems, as many of the integrals are simpler than the general form given above. You will need to impose any required boundary conditions in order to determine the constant of integration above.

The solution of first-order linear equations is a standard problem in most first differential equations courses. You can find many references online and here is a good one providing plenty practice: [Paul’s online notes](https://tutorial.math.lamar.edu/classes/de/linear.aspx)

## 38.2 Second-order constant coefficient ODEs

Second-order linear constant-coefficient ODEs are of the form

With both forced () and unforced () varieties studied. For the unforced variant (homogeneous), you will remember that the equation is solved by attempting the ansatz and solving for .

There are many references on this topic since it is the standard second-order theory learned in most initial differential equations courses.

References:

* [Paul’s online notes](https://tutorial.math.lamar.edu/Classes/DE/IntroSecondOrder.aspx)

# 39. Dynamical systems

|  |
| --- |
| **Appendices**  These appendices may contain material that is added during the term, dependent on student enquiries and need. |

We will need some of the theory from your prior ODEs and Modelling and Dynamics courses as related to the solution of 2x2 linear systems of the form

where

If you need a refresher, refer to your old notes; there is also a good review on [these notes](https://www.ndsu.edu/pubweb/~novozhil/Teaching/266%20Data/lecture_23.pdf) and also [these notes](https://tutorial.math.lamar.edu/Classes/DE/SystemsIntro.aspx).

## 39.1 Stability analysis for 2x2 systems

In the case of 2x2 systems, there is a useful classification and shortcut. The eigenvalues, and are given by

where and . Once and are known, then the fixed points can be classified based on the following diagram.

|  |
| --- |
| Figure 39.1: Classification diagram from (Kaper and Engler 2013) |

Above, the critical parabola is where .

# 40. Vector calculus

|  |
| --- |
| **Appendices**  These appendices may contain material that is added during the term, dependent on student enquiries and need. |

## 40.1 Surface integrals

Surface integrals are generalisations of the concept of a multiple integrals to integrals over surfaces. For example, consider a flat plate of area heated to a certain temperature, and thus emits a certain amount of energy, , per unit area (per unit time). The total energy emitted per unit time would then be .

If we want to consider the same problem, but now posed on a general surface, say the planet , and where varies along the surface, then we must consider adding together . We thus chop our surface into smaller pieces, each with area , multiply each piece with its corresponding energy, and then sum the result.

The result is the surface integral

A tutorial for computing surface integrals can be found [here](https://tutorial.math.lamar.edu/classes/calciii/surfaceintegralsintro.aspx).

You will not need to know how to calculate surface integrals in general in the course.

# 41. Modelling

|  |
| --- |
| **Appendices**  These appendices may contain material that is added during the term, dependent on student enquiries and need. |

# 42. Finite difference approximations

|  |
| --- |
| **Appendices**  These appendices may contain material that is added during the term, dependent on student enquiries and need. |

Many of the numerical methods we need will require using numerical approximations for the derivatives of functions. For instance, using Newton’s method to solve for the zeros of a system of equations requires calculating the Jacobian matrix.

## 42.1 Forwards, backwards, and centred differences

We want to learn how to use finite differences in order to approximate derivatives numerically. We know that

provided the limit exists. Therefore, a simple idea to approximating the value of is to use a small numerical value for , and calculate (the gradient of the secant line),

where is a specified small number. The above is known as the **two-point forward-difference formula**. In fact, we can determine exactly the error of such an approximation via Taylor’s theorem. If is twice continuously differentiable, then

for some point . Therefore by re-arrangement, we see the following.

|  |
| --- |
| **Two-point forward-difference formula**  where . |

Notice that the error in using the two-point forward difference approximation is then , and this error tends to zero as (as long as is continuous). We thus call the above formula a first-order finite-difference approximation. If the error is , we call the corresponding formula an th-order approximation.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Example**  Use the two-point forward difference formula with different values of in order to approximate the derivative of at .   |  |  |  |  | Error | | --- | --- | --- | --- | --- | | … | … | … | … | … | | … | … | … | … | … | | … | … | … | … | … | |

Similar formulae can be developed for the backwards difference (send ).

A more accurate formula can be developed via subtracting the Taylor series for from that for . This results in:

|  |
| --- |
| **Three-point centered-difference formula**  where . |

Thus we see that the centered difference formula is accurate to .

## 42.2 Jacobian matrices

|  |
| --- |
| **Reference**  You will have encountered the Jacobian, firstly in your first-year Methods courses when performing change-of-coordinates in integration formulae, and secondly in your second-year Modelling and Dynamics courses when studying differential equations. It also comes up in the second-year Vectors and PDEs course. |

The Jacobian matrix, , of a vector function, ,

is the matrix of all its first-order partial derivatives,

We can essentially think of the Jacobian as the multi-dimensional extension of the basic derivative. It encodes all of the (first-order) information about the rate-of-change of the function.

It is interesting and important to consider the numerical evaluation of the Jacobian, in the event that the function cannot be easily differentiated exactly. The simplest algorithm is based on approximating each entry of the matrix by a finite difference.

For example, let us consider approximating

We define the th unit vector by , which the th entry being one and all entries being zero. Then we can approximate the derivative by a central difference,

where is a small step size.

Let us test this by calculating the Jacobian for

The pseudocode for this looks like this

|  |
| --- |
| **Pseudocode for numerical Jacobian**  Input: function F (size m), point x (size n), step size h  1. Create an (m x n) matrix, J 2. a. Loop through all the rows indexed by i  b. Loop through all the columns indexed by j  Create the ej unit vector  Calculate the difference of F\_i(x + h e\_j) - F\_i(x - h e\_j)  Divide by 2h  Assign this value to the (i,j)th value of the Jacobian  Output: the (mxn) Jacobian matrix J |

Here is a code that puts it into practice.

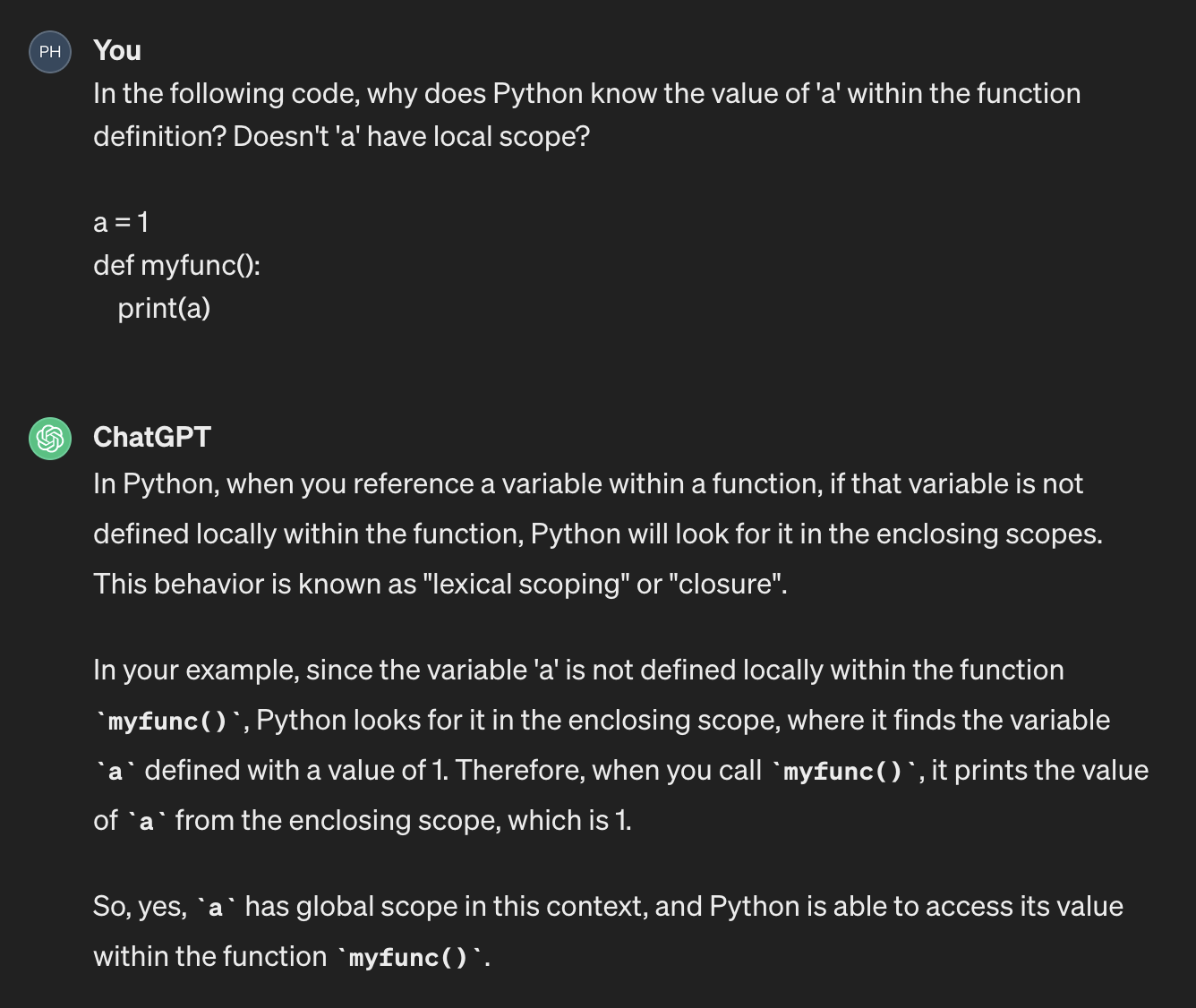
import numpy as np  
  
def jacobian(func,initial,delta=1e-3):  
 f = func  
 nrow = len(f(initial))  
 ncol = len(initial)  
 output = np.zeros(nrow\*ncol)  
 output = output.reshape(nrow,ncol)  
 for i in range(nrow):  
 for j in range(ncol):  
 ej = np.zeros(ncol)  
 ej[j] = 1  
 dij = (f(initial+ delta \* ej)[i] - f(initial- delta \* ej)[i])/(2\*delta)  
 output[i,j] = dij  
 return output  
   
def myf(x):  
 x1 = x[0]  
 x2 = x[1]  
 output = np.zeros(3)  
 output[0] = x[0]\*\*2  
 output[1] = x[1]\*\*2  
 output[2] = x[0]\*x[1]  
 return output  
   
jacobian(myf,[1,2])

array([[2., 0.],  
 [0., 4.],  
 [2., 1.]])

# 43. Coding

## 43.1 Variable scope

During the 27 Feb 2024 lecture, the question of variable scope came up. Firstly, here is what ChatGPT says:



What ChatGPT says on scope

So indeed it looks like Python uses a scheme where if a variable is not locally declared, it will use values of the variable outside of the function. This can allow us to produce this kind of thing:

def myfunc():  
 print(a)  
  
a = 1  
myfunc()  
a = 2  
myfunc()  
a = 3  
myfunc()

1  
2  
3

So I was wrong! Python variables are global when defined outside the function! [Here is a reference](https://www.w3schools.com/python/python_variables_global.asp).

This is certainly easier to use! However it can lead to confusion because for very complex codes, you may not be sure what value of a variable is being used within a function. Because most of the programming you will do in this course is very simple, this is unlikely to lead to trouble.

It’s generally good to avoid this whenever it is possible, though. For instance in this particular case, it makes much more sense to pass in the variable.

def myfunc(a):  
 print(a)  
  
a = 99  
myfunc(1)  
myfunc(2)  
myfunc(3)

1  
2  
3

In the above case, the a that appears in the function is different from the a that is declared outside the function. This makes it very easy to understand what your function is doing without knowing what is happening outside of it.

For example, there is no danger of values being reset:

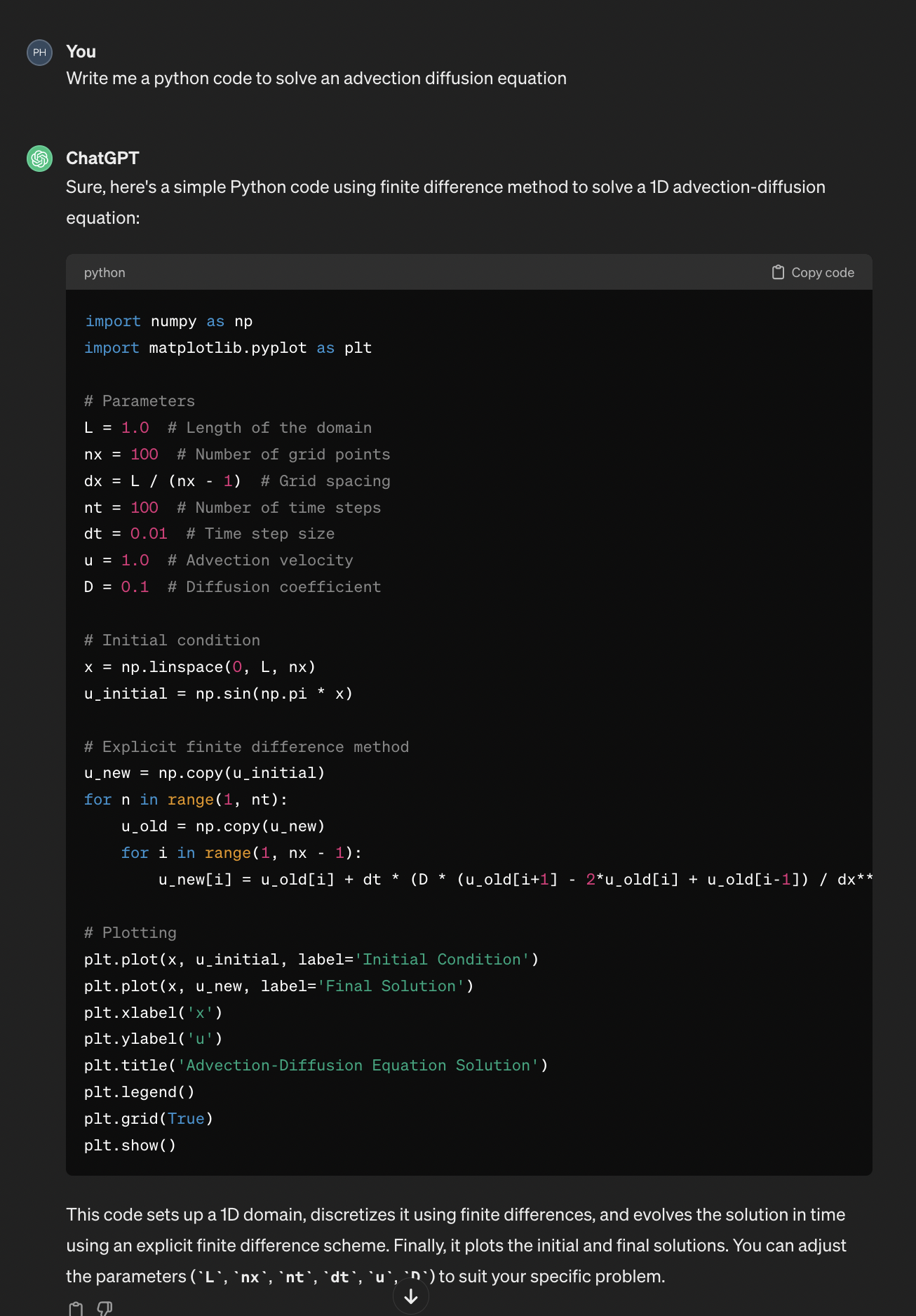
def myfunc(a):  
 a = a + 1000  
 print(a)  
  
a = 99  
myfunc(1)  
myfunc(2)  
myfunc(3)  
print(a)

1001  
1002  
1003  
99

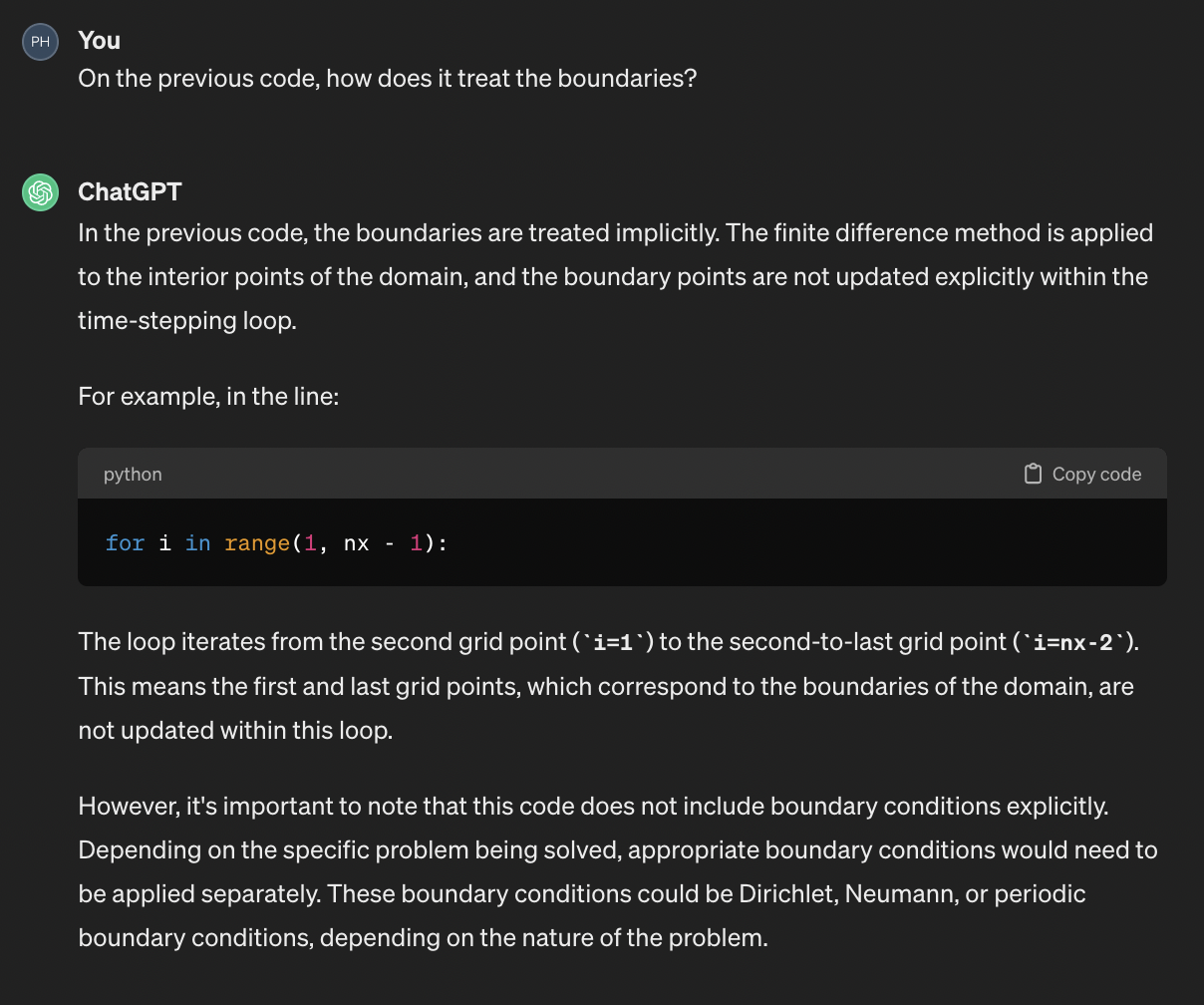
## 43.2 Conversations with ChatGPT

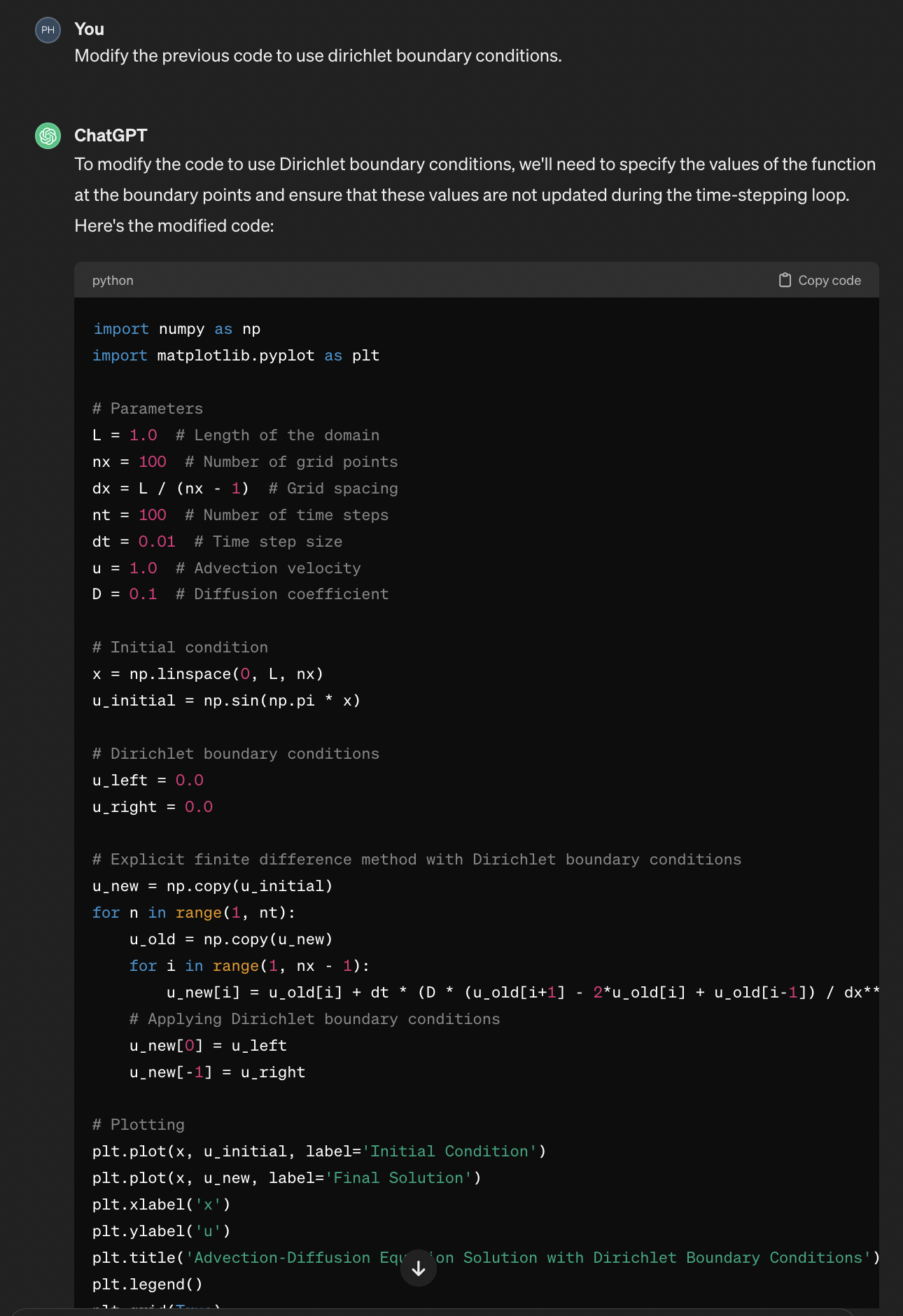
We should have an informal discussion of how the use of GenAI in education. Does it help with coding? When should it shoudn’t it be used?

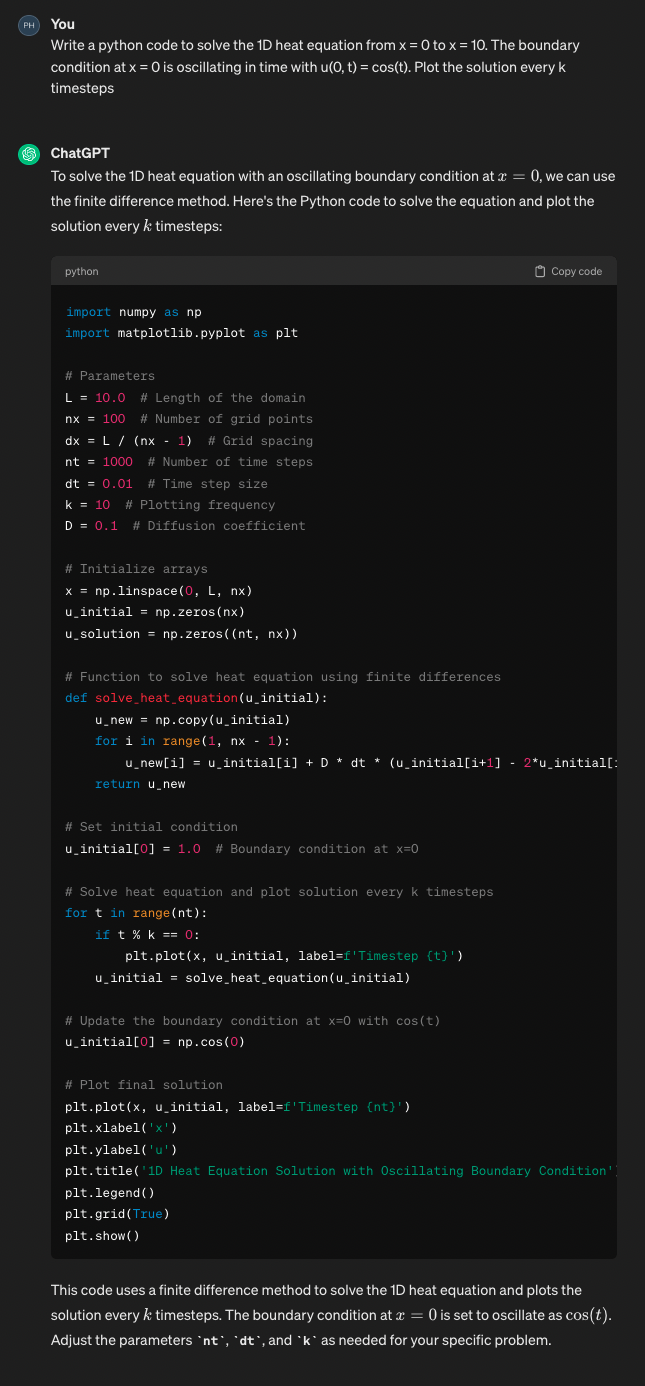
Asking something simple.

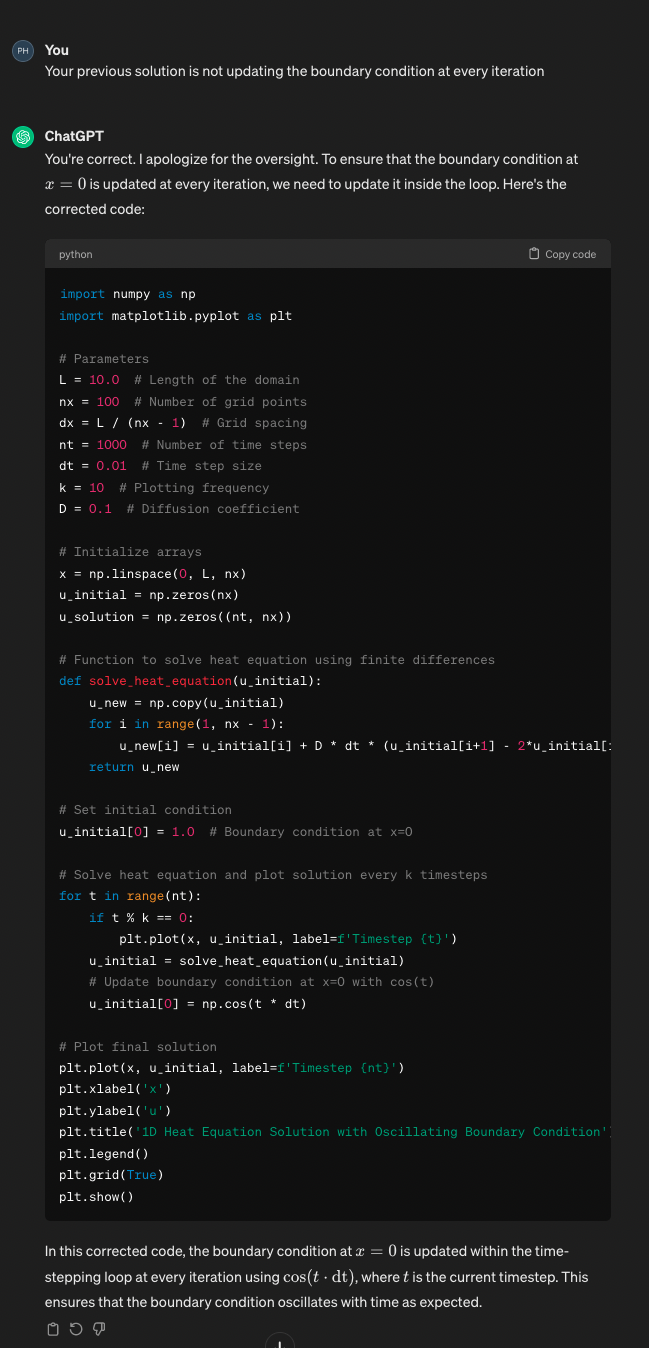


A nice chat with ChatGPT

Wait a minute… 

Modifying.. 

Modifying again to oscillate in time… 

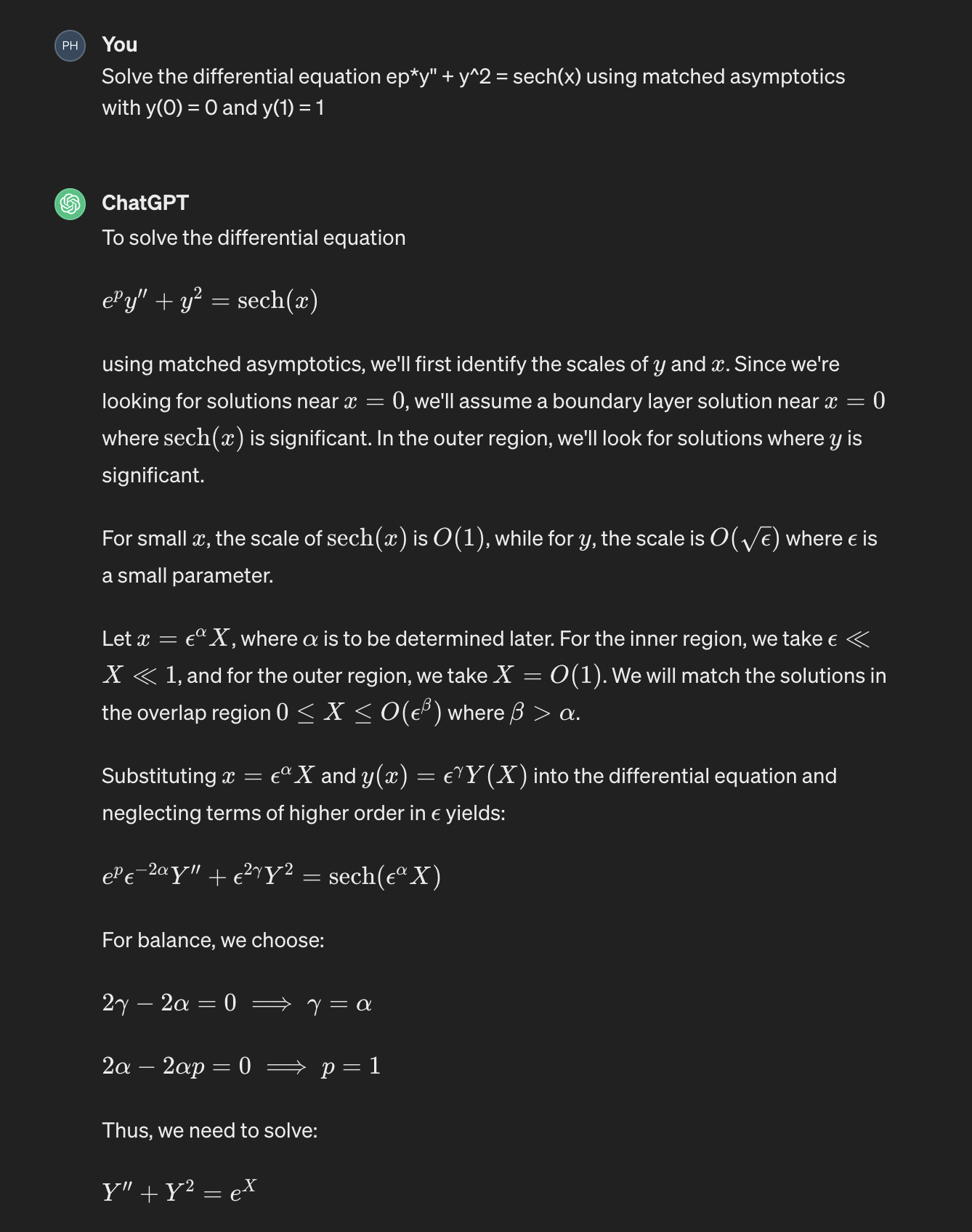
Oops! 

# 44. An example of ChatGPT going wrong

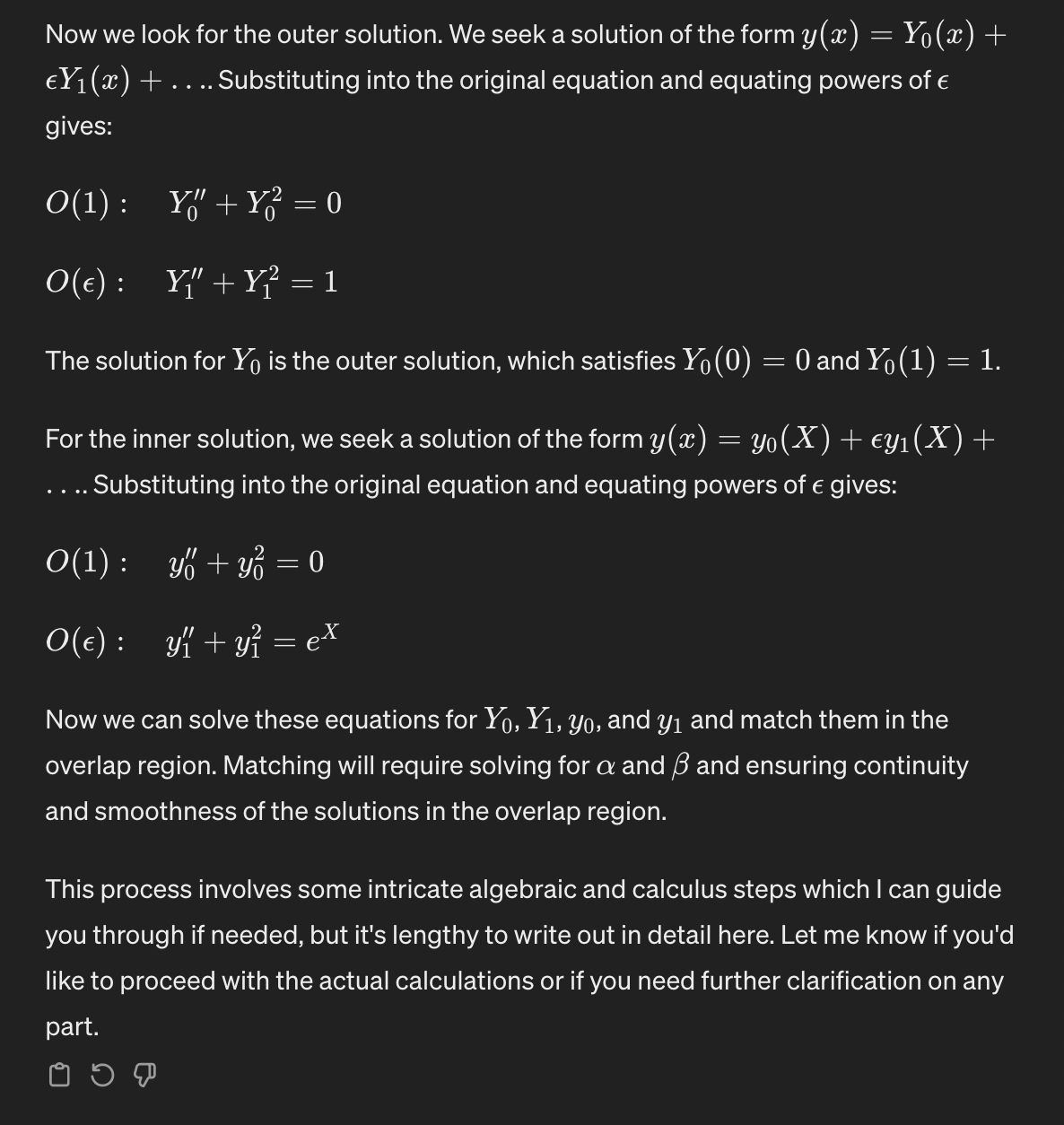
Please refer to [the Courseworks and genAI document for guidance](https://bath-prod-sss1.s3.eu-west-1.amazonaws.com/35/77/35774014a834e23684e6c48abb5314129715891d?response-content-disposition=inline%3B%20filename%3D%22gen-ai-guidelines.pdf%22&response-content-type=application%2Fpdf&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJBFBMNJTZPM2NVZA%2F20240307%2Feu-west-1%2Fs3%2Faws4_request&X-Amz-Date=20240307T141016Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21584&X-Amz-Signature=25f0f09f8861ee9f73f8346a64b643f6bbc9bc6f3bfe5a7253a94ffb5ff2b271).

Remember that standard generative AI tools do not have mathematical engines. They have mathematical information, but no knowledge of mathematics. They can produce content that looks convincing but is incorrect. Worse still, in order to diagnose the issues, you may need to already be an expert!

Here is an example based on material from this course.



Example ChatGPT output



Example ChatGPT output

It is important to remember that “correcting” the output of ChatGPT is sort of a misnomer. ChatGPT or similar genAI or Large Language Models and algorithms do not include common sense, or any algorithmic understanding of syntactic rules. With ``correcting” misunderstandings, you generally need to understand why the error was made in the first place. With LLMs, the “why” is replaced by the fact that, within its training set, words are simply close together in the high-dimensional vector space.

This brings to mind the well-known phrase *“Not even wrong”*



Not even wrong

## 44.1 Breakdown of issues

To solve the differential equation

|  |
| --- |
| **Solution**  Why did it change “ep” to ? |

using matched asymptotics, we’ll first identify the scales of and . Since we’re looking for solutions near , we’ll assume a boundary layer solution near where is significant.

|  |
| --- |
| **Solution**  Why did it assume the boundary layer is near ? Also, is not necessarily ``significant” near . |

In the outer region, we’ll look for solutions where is significant.

|  |
| --- |
| **Solution**  Doesn’t really make sense. |

For small , the scale of is , while for , the scale is where is a small parameter.

|  |
| --- |
| **Solution**  That’s not right… is not near . It’s not obvious at this point what the scaling for is. |

Let , where is to be determined later. For the inner region, we take , and for the outer region, we take . We will match the solutions in the overlap region where .

|  |
| --- |
| **Solution**  This isn’t right. In the inner region and the outer region . Almost all the statements are wrong. |

Substituting and into the differential equation and neglecting terms of higher order in yields:

For balance, we choose:

Thus, we need to solve:

|  |
| --- |
| **Solution**  The above lines don’t make any sense. For one thing, near , you should have . Secondly if you examine what its working gives, no value of is obtained. The correct inner equation is found by balancing the first and last terms, giving and |

Everything else is wrong.

# References

Budyko, Mikhail I. 1969. “The Effect of Solar Radiation Variations on the Climate of the Earth.” *Tellus* 21 (5): 611–19.

Fourier, Joseph. 1827. “Mémoire Sur Les Températures Du Globe Terrestre Et Des Espaces Planétaires.” *Mémoires de l’Académie Royale Des Sciences de l’Institut de France* 7: 570–604.

Fowler, Andrew. 2011. *Mathematical Geoscience*. Vol. 36. Springer.

Kaper, Hans, and Hans Engler. 2013. *Mathematics and Climate*. SIAM.

MacKay, D. J. C. 2009. *Sustainable Energy – Without the Hot Air*. UIT Cambridge Ltd.

Pouillet, Claude Servais Mathias. 1838. “Memoire Sur Le Chaleur Solaire.” *Paris*.

Stommel, Henry. 1961. “Thermohaline Convection with Two Stable Regimes of Flow.” *Tellus* 13 (2): 224–30.

Van der Veen, CJ. 2000. “Fourier and the ‘Greenhouse Effect’.” *Polar Geography* 24 (2): 132–52.

1. Subject to confirmation. [↑](#footnote-ref-27)