University
of Dundee

MA40001

MA40001

Philip Murray

2024-01-09

Table of contents

Preface 4
- How to contact me? 4
- Software 4
  - Uni machines 4
  - On your own machine 5
- Content 5
  - Lecture notes 5
  - Template codes 5
- Python codes 5
- Assessment 6
- Plan 6

1 Scientific/mathematical writing 7
- 1.1 Common elements across different assessment points 7
  - 1.1.1 Narrative 7
  - 1.1.2 Pitching the content appropriately 8
  - 1.1.3 Equations 8
  - 1.1.4 Figures 10
  - 1.1.5 Schematic diagrams 14
  - 1.1.6 References 14
- 1.2 Written reports 15
- 1.3 Verbal presentation 16

2 File management 17
- 2.2 Github classroom 17
- 2.3 Edit an existing file and push the modified version to github 17
- 2.4 Create a new file on your local machine and push it to github 17
- 2.5 Revert to a previous version of a file using version control 17
- 2.6 Using the command line interface 18

3 Quarto 19
- 3.1 Getting started 19
- 3.2 A first markdown document 19
- 3.3 Creating a pdf 20

3.4 Adding structure to your Quarto documents 21
3.4.1 Section headings 21
3.4.2 Tables 22
3.4.3 Figures 22
3.4.4 Cross referencing 23
3.4.5 Citations and references 25
3.5 Appendices 25
3.5.1 Submitting code 25
3.6 Websites and blogs in Quarto 26

4 Typesetting mathematics **27**
4.1 The math environment in latex 27
4.2 Exercise: an example worksheet 29

5 Presentations **31**

6 Python code development **32**
6.1 Introduction 32
6.2 Some key ideas 32
6.2.1 Calculator 32
6.2.2 Variables and datatypes 32
6.2.3 Lists and dictionaries 33
6.2.4 Logical statements and control loops 33
6.2.5 Writing functions 33
6.2.6 Code debugging 33
6.3 Python libraries 34
6.3.1 Essential 34
6.3.2 Python libraries of interest to particular projects 37
6.3.3 Numerically solving differential equations (sci-py) 37
6.3.4 Optimisation (skikit-learn) 37
6.3.5 Data analysis (pandas) 37
6.3.6 Machine learning (tensorflow) 37
6.3.7 Image analysis (skikit-image) 37
6.4 Writing your own scripts 37

Preface

Welcome to MA40001

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

How to contact me?

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

Software

The taught component on the module will require you to use:

- Quarto
- Visual Studio Code
- Github
- latex
- python

Why bother with all this?

The aim is to provide you with *open-source* tools that enable you to generate modern scientific/mathematical documents.

Uni machines

From Apps Anywhere, open:

- VS Code with Quarto XXX

On your own machine

Install * Quarto * VS Code * Python + libraries (matplotlib.pyplot, numpy,)

Within Quarto: * install the quarto extension for VS Code * install Latex (follow Quarto instruction for generating a pdf). * Configure the Python interpreter (Show and Run Commands->Python Select Interpreter)

Content

Lecture notes

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

Template codes

I have provided template codes via the github page dundeemath/MA40001resources.

We are using github classroom. Hence you you each have a private github repository (which only I can see the content of).

The idea is that you *pull* documents from github, edit them as necessary and *push* the edited documents back to github.

Python codes

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

Many of you have taken the Introduction to Programming module at Level 2 and have therefore some experience using Python. I strongly encourage you to use the provided codes as a tool to play around with numerical solutions of the various models that we will be working on. The codes should run as standalone Python codes (e.g. Paste into Anaconda->Spyder).

Assessment

- Presentation
- Interim report
- Poster
- Final report
- Viva

Plan

Table 1: Module organisation

Week	Material	Assessment
1-3	Quarto	Formative
4-6	Python	Formative
7-10	Presentation practices	Formative
11	Assessed presentation	Summative

1 Scientific/mathematical writing

The assessment of your project will involve:

- verbal presentation (~10 minute slide presentation)
- written reports (interim report and final report)
- poster
- viva (aural exam)

The main aims are to develop:

- mathematical skills,
- independent investigation skills
- scientific/mathematical communication skills.

1.1 Common elements across different assessment points

1.1.1 Narrative

A common theme with the different assessment points is the need for a *narrative* around your project:

- can you describe the project in *one succinct sentence*?
- can you outline why the topic is important?
- what is the background to the project?
- what methods/results/techniques have you developed?
- can you summarise your findings?

Once you have settled on a narrative, the different assessment points can be thought of as variations on the presentation of your project narrative.

1.1.2 Pitching the content appropriately

The first in principle in good communication is to *know your audience*.

If you pitch at too high a level (e.g. assuming that your audience know more than they actually do), the audience will likely be unable to follow your reasoning.

If you pitch at too low a level (e.g. by explaining concepts that your audience is already familiar with) they will likely be bored/feel condescended etc.

At all points of your assessment: assume that you are communicating with your class mates, i.e. Level 4/5 of a undergraduate mathematics degree. This means that you should not assume that the audience has specific knowledge of the details/background of your project.

1.1.3 Equations

Are equations presented accurately? Are mathematical objects accurately defined? Has sufficient background detail been presented so that the arguments can be reasonably followed?

1.1.3.1 Typesetting equations

To typeset formulae is actually quite difficult. Mathematics uses a variety of symbols and several different alphabets: Roman, Greek, Hebrew are the most common. In addition formulae are often more similar to graphics than to text. There are numerators and denominators which in turn can have fractions etc., e.g.:

$$f := \frac{1}{1 + \frac{1}{1+\frac{1}{1+x}}}.$$

To make this formula look good requires either an advanced typesetting program or a lot of effort. Most common typesetting programmes come with some sort of equation editor, but very few can handle such a problem. The most powerful mathematical typesetting program, which is also the format used for almost all mathematical literature is LaTeX. We will learn about LaTeX later.

As with grammar, for a language there exist also certain conventions about how to write formulae. Here is a (far from exhaustive) list of the most import conventions:

- Treat the formula like text. If the formula is at the end of a sentence there has to be a full stop at the end of the formula. If another formula follows use a comma or semicolon. This is how we count
$$1 + 0 = 1, 1 + 1 = 2.$$

- use Roman (typically lower case) letters in *italic* style for all variables: x, y, z, a, b, c, both if we refer to them in the text as well as in formulae. Note difference between x and x.
- use Greek letters for angles, and e.g. differential forms;
- typeset vectors in bold, $\mathbf{a}$, or using an arrow, $\vec{a}$;
- typeset functions in roman, $\sin(x)$ rather than $sin(x)$;
- typeset matrices using capital roman letters, e.g. M;
- represent number systems and also certain vector spaces in a style where certain lines are double: $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{C}$, …;
- use curly brackets for sets, e.g. $\{1, 2, 3\} = \{2, 1, 3\}$, and regular brackets for an ordered list $(1, 2, 3) \neq (2, 1, 3)$;
- denote a range by three dots: $i = 1, 2, ..., n$, (no bracket required);
- use brackets only where necessary, note that multiplication/division takes precedence over addition/subtraction; e.g
  $$a + (b \cdot c)$$
  does not need the brackets, but
  $$(a + b) \cdot c$$
  does. Any fraction replaces a bracket, so $\frac{5}{a+b}$ does not need the brackets.
- use a separate line for any formula that uses more than one line or complicated formula. Don't write $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$ in line. Instead write
  $$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Common mistakes

Common mistakes are (in addition to violating the above conventions):

- there are typos/inconsistencies in the equations
- the presented system is not mathematically self-consistent (e.g. a system of ODEs is missing initial conditions)
- variables/parameters are not defined
- using different font styles and sizes to represent the same quantity;
- not using the same symbol (or font) in the text and separate formulae;
- not using spaces between symbols, and/or wrong symbols, e.g axb instead of $a \times b$;
- forgetting brackets or placing too many brackets.

1.1.4 Figures

A formula is often the best and most concise way to communicate a relation or function to a mathematically trained audience. However, there are many cases where even a mathematician might have difficulty understanding (e.g. in the short time available in a presentation) what a function represents. For functions of one and two variables there is always the option to show a plot of the function. Here is an example of a function in two variables:

$$f(x, y) = \cos(x)\sin(xy). \tag{1.1}$$

Although this is a comparatively simple function it takes some time to figure out what properties the function has, that is for instance:

- Is the function periodic?
- How many maxima/minima does it have?
- How does it behave for $x, y \to \infty$?

To explain the behaviour of such a function, a plot will help! But which plot? For example, we could use a surface or contour plot (see Figure 1.1). The advantage of a contour plot is that it is often easier to see the locations of maxima and minima but it is not so easy to see how high the extrema are.

Connecting figures to text

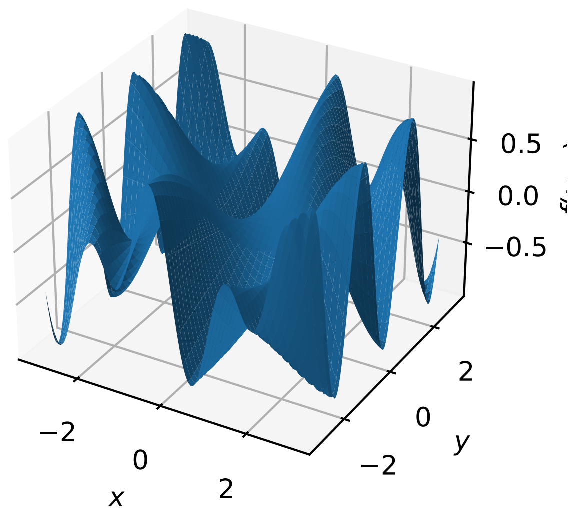
Note that all plots should be cross referenced in the text, i.e. we should never have figures that are not referenceed somewhere in the text.
A good rule of thumb is: if figures and tables are removed from the text, does the text still read coherently? i.e. the figure is helping the reader to understant a point that is made in the text. It is not replacing the need for text.

Other types of plots are useful for different purposes:

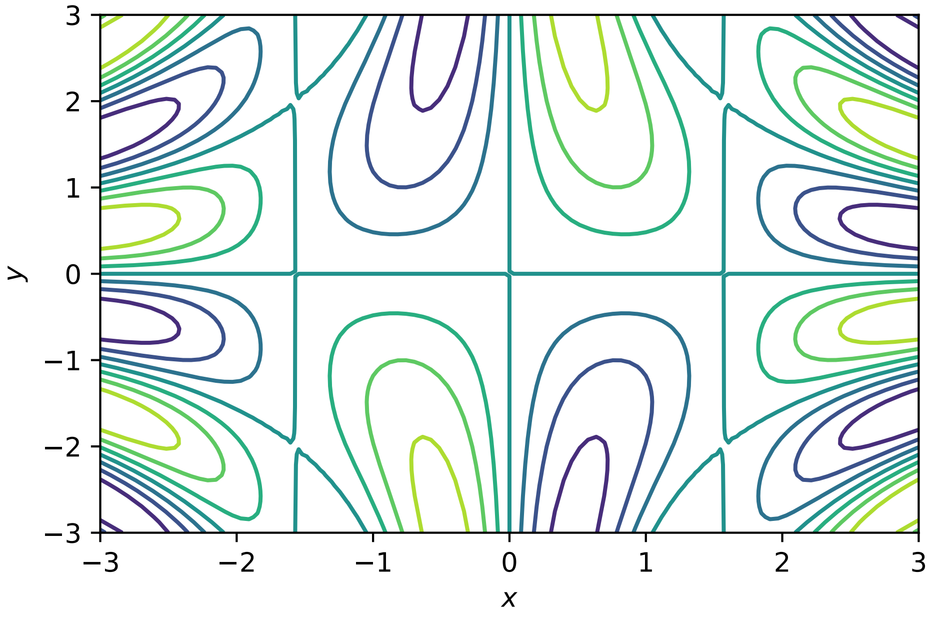
- bar chart (Figure 1.2)
- pie chart (Figure 1.3)
- scatter plot (Figure 1.4)
- line chart (Figure 1.5).

common mistakes

- hanging figures (i.e. figures that not placed in context (e.g. connected to the text))
- figures without axis labels
- coloured graphs without colour scale/legend
- labels/ tick marks are too small
- type of graph unsuitable for the data shown



(a) 3D plot of Equation 1.1



(b)

Figure 1.1

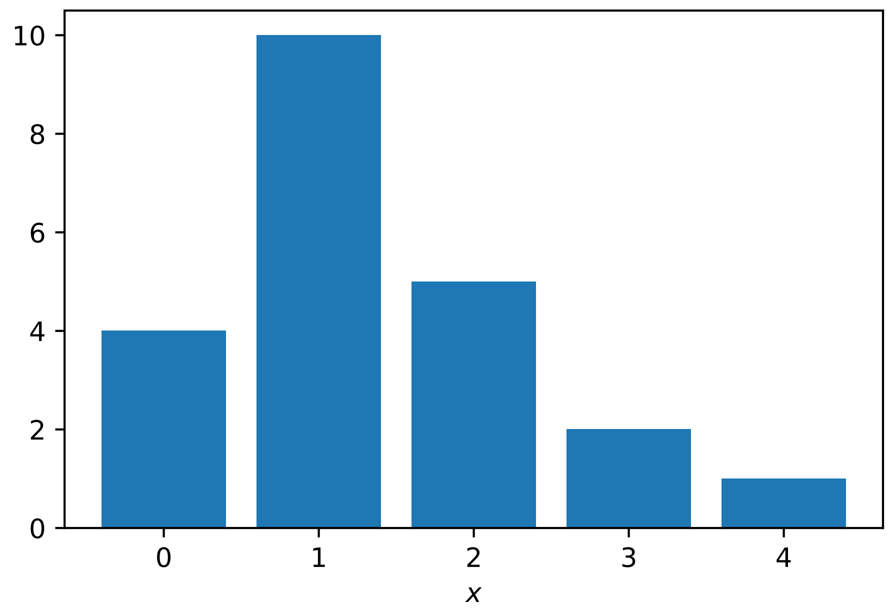


Figure 1.2: A bar chart.

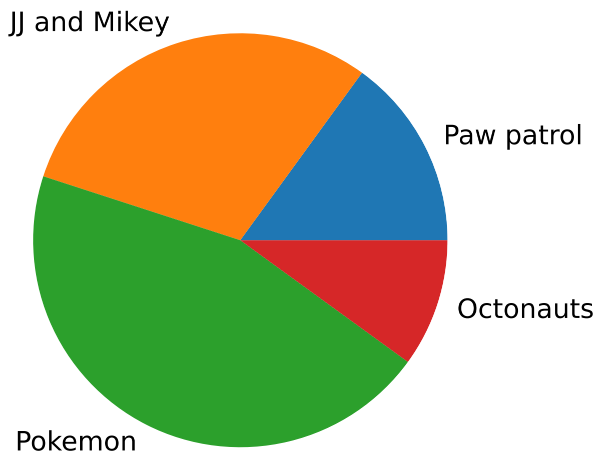


Figure 1.3: A pie chart.

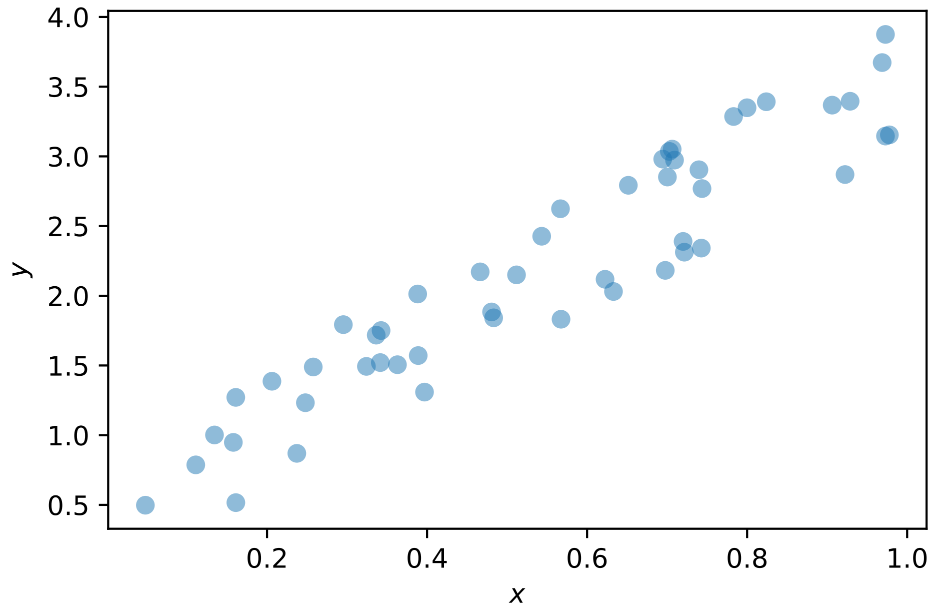


Figure 1.4: A scatter plot.

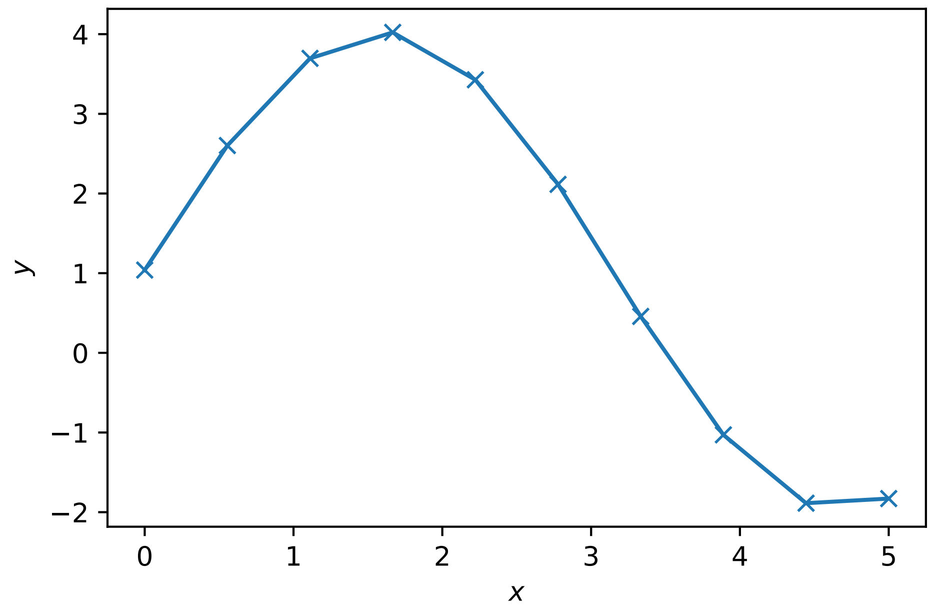


Figure 1.5: y is plotted against x.

- the figure was not adequately connected to the text (it is not clear, for example, what equations were solved, what method was used.)

Some rules -of-thumb:

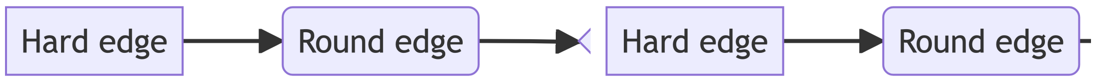
- check that any text in axis labels/legends is approximately the same size as font in the main text

1.1.5 Schematic diagrams

Schematic diagrams are a useful way to help to try and introduce a new concept/summarise a finding.

There are many ways to generate schematic diagrams (e.g. generating in software such as Photoshop or Illustrator and saving as an image).

Alternatively Quarto provides an itnerface to a number of graph tools that generate schematic diagrams, e.g.



1.1.6 References

All text in your report must be either original or attributed to the originating author(s). A word-for-word quotation should be indicated with quotation symbols. Such as: The whatever effect causes “extremely aggressive phenotypes” to dominate (Smith et al. 2010). Extensive quotes of more than one sentence should generally be set apart as a separate paragraph: According to Tolkien (1954):

“ One Ring to rule them all, One Ring to find them, One Ring to bring them all and in the darkness bind them. ”

It is generally best to limit quotations to no more than 1-3 sentences. Instead, one should paraphrase and/or summarise cited work. Moreover, the majority of the content of a report should be in the authors original words. It should be the authors original logical argument, with citations used to document and justify key assertions and facts. Hence, quotations should be kept to a minimum.

Students must take great care when quoting or paraphrasing text to cite the original source. To do otherwise is *plagiarism*, which undermines the integrity of science, and has serious consequences for the authors as well. (e.g., failing grades, expulsion from a degree program, loss of an academic position, being shunned by the scientific community, etc.)

University of Dundee definition of plagiarism

Plagiarism is the unacknowledged use of another's work as if it were one's own. Examples are:

- the inclusion of more than a single phrase from another's work without the use of quotation marks and acknowledgement of the source;
- summarising another's work by changing a few words or altering the order of presentation without acknowledgement;
- copying another's work;
- the use of another's ideas without acknowledgement.

NB: if you wish to reference your own work, it is important to acknowledge yourself as the source and provide the appropriate reference.

Wikipedia is an excellent resource for learning new topics. However, it is not an acceptable source for citation in a report or publication for various reasons: * the content of wikipedia changes continuously and what we refer to today might not be there tomorrow. , * it is generally not written by experts and often contains errors.

Instead you should try and try and cite primary sources (e.g. research papers or textbooks) as these have permanent bibliographic identifiers (dois).

1.2 Written reports

The overall structure of a report is (generally) as follows:

1. Title page: This should give the title of the report, the author list, and the date.
2. Abstract: provide a short synopsis (~10 sentence) of the project.
3. Introduction: This introduces the overall context and importance of the problem you are addressing. It should give:
    - A basic paragraph or two on background of the problem, its significance, and motivation for the paper. It should make us want to continue reading.
    - In a research-grade paper, you would include information as to why preceding work by others has been insufficient. What work, findings, or improved methodology are required?

    - A summary of any hypotheses that you will develop and justify throughout the paper
    - A basic outline of the remainder of the paper, including a note on the methodology (in mathematics, these would be your modelling, analytical, and numerical techniques).

4. Actual content: There should be one or more sections that logically progress your argument and analysis. For example:

    - Formulation of the problem
    - Methods used to solve the problem
    - Results

5. Discussion and Conclusions: Wrap up with a summary of your major results, the significance of your conclusions (including any additional analysis of the results to lay out this significance), and an outlook what else could be done/ or where improvements are required.

> Some common errors in the stucture
>
> - Insufficient development of background (the author assumes that the reader knows more than they do)
> - The aims of the project are not clearly stated
> - The importance of the topic is insufficiently described (e.g. why study this problem?)
> - the discussion is too short

7. References: This section contains the full list of publications that you cited in your report.

8. Appendices (optional): Lengthy and/or tedious calculations or details that are too distracting to the overall flow of your paper, and yet are necessary to fully document your work, should be placed in appendices. You must also provide evidence for any codes that you have developed.

A template project report is available via the MA40001Resources github repository.

1.3 Verbal presentation

A template presentation is available via the MA40001Resources github repository.

2 File management

2.1

Github is a developer platform that allows developers to create, store, manage and share their code. It is version-controlled, meaning that once you *push* files to the github cloud, you will have access to it and all previously *pushed* version of that file.

- Github is used across academia and industry.
- It provides you with a back-up for your project.

2.2 Github classroom

- Find the Github Desktop app on Apps Anywhere.
- You will need to create a github account if you do not currently have one.
- Download the MA4000Resources repository using the provided link.

2.3 Edit an existing file and push the modified version to github

- Open the file *.text
- Edit the text
- Commit the file and push it
- Check that the version of the file in the cloud has changed to reflect your edit.

2.4 Create a new file on your local machine and push it to github

2.5 Revert to a previous version of a file using version control

Suppose that you have made a mistake and want to revert to a previous version of a file. - Find the file in the github repository - Look back at previous versions of the the file - Download the copy that you want to revert to

2.6 Using the command line interface

You can also do all of the above from the terminal.

3 Quarto

Quarto is an open source scientific and technical publishing system. It can be used to make a range of publishable outputs (reports, posters, slides, blogs, webpages dashboards etc.). In this section you will learn to use Quarto to write reports and make slide decks. However, one of the advantages of learning Quarto is that it is straightforward to write and publish websites, blogs, dashboards and books.

It is assumed that you are using VSCode with Quarto from Apps Anywhere on the Uni machines. However, you are also encouraged to install Quarto/VSCode on your personal computer.

Quarto documents are written using Markdown (you may have previously used RMArkdown). In the background, Quarto uses an open source document convertor called Pandoc.

3.1 Getting started

Apps Anywhere Visual Studio Code X with Quarto extension-> Launch File->New File->Quarto document

To compile the document: click preview button in top tight corner

3.2 A first markdown document

To begin with, let's consider a simple markdown document.

```
<!-- Configuration information -->
---
title: "Hello, Quarto"
format: html
---

<!-- Insert content below -->
This is a Quarto document.
```

At the top of a file we include a YAML block (enclosed by `---`). This provides configuration information for the document.

To run this code you should:

1. Open VisualStudio Code.
2. Create a new `.qmd` file and save in a sensible directory.
3. Copy and paste the above code into the .qmd file.
4. - Compile the code by clicking the preview on the top right of your VSCode editor.

   OR

   - Open a terminal, navigate to the directory and type

   ```
   quarto render
   quarto preview
   ```

5. Have a look at the contents on the directory. You shoule be able to see that a .html fil has been generated.

Quarto on your personal computer

On your personal computer you may need to install Quarto.
You can find some guidance on how to do this here.

3.3 Creating a pdf

Over the course of your project you will need to submit documents in pdf format. To get Quarto to generate PDF you need to edit the YAML information as follows.

```
---
title: "Hello, Quarto"
format: pdf
---

This is a Quarto document
```

Repeat the above compiling steps. You should now be able to see that a pdf file has been created.

Note

It is crucial that you can generate a pdf from your Quarto document as your final report will be submitted in pdf format.
It is advised to regularly check that a pdf is generated when you render your Quarto document. I have noticed that with some bugs in latex, Quarto can render a document in .html format (it looks ok in preview) but be unable to generate a pdf.

LAtex on your personal computer

To generate a pdf, your computer must have a version of latex installed. In the Uni machines this has been set up for you. On your personal machines you may need to install latex and configure Quarto.
You can find guidance on how to do this here.

3.4 Adding structure to your Quarto documents

Information on document structure can be found on the Quarto help pages. Below I will highlight some of the key document structures that you will likely need in your project.

3.4.1 Section headings

Below is some Quarto code that will generate a document with Section headings.

- Create a new .qmd file in VS Code
- Copy and paste the code below
- Render the code and check that a .html file has been generated in
- Open the .html in a browser.

```
---
title: " Quarto sections"
format: html
code-fold: True
---

# Main section
This is a Quarto document

## Sub section title
This is a subsection
```

```
###
This is a further subsection

#
This is a new section

##
```

3.4.2 Tables

Tables can provide a concise way to present information. Typical uses in a report might be to gather information about model parameters or to display data.

Here is an example of Quarto code to make a table:

```
| Parameter | Value | Unit  |
|---------|:-----|------:|
| $a$      | 1  |    ms$^{-1}$ |
| $b$      | 2  |   s$^{-1}$  |
| $c$        | 3    |     Nondim  |

: Demonstration of  table
```

The rendered version appears as

Table 3.1: Demonstration of table

Parameter	Value	Unit
$a$	1	ms$^{-1}$
$b$	2	s$^{-1}$
$c$	3	Nondim

You can find more info. on Quarto tables here.

3.4.3 Figures

To include an existing figure in a Quarto document then you must firstly store the figure file in a sensible directory. Then you can include the figure (‘DemoFigureMA40001.png’) using the syntax:

```
![This is the figure caption.](DemoFigureMA40001.png)
```

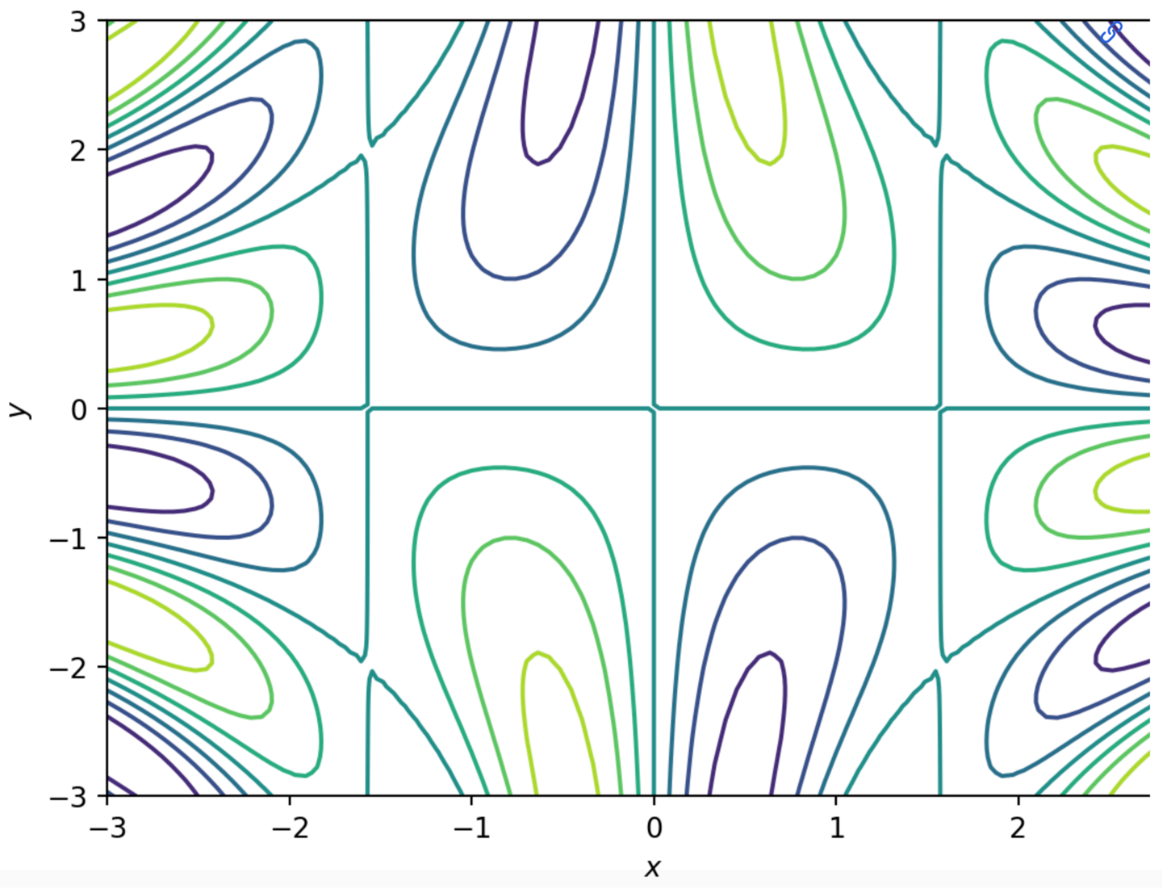


Figure 3.1: This is the figure caption.

You can find more info on Quarto figures here.

3.4.4 Cross referencing

Throughout your document you will need to refer to different objects that you have created (e.g. Tables, Figures, equations, theorems).

In Quarto objects are tagged using a specific syntax (prefix for that specific obect + unique tag). See Table 3.2 for some examples. The object is cross referenced using an '@' tag.

Table 3.2: A table with syntax for referencing some Quarto objects.

Object	Tag	Reference
Table	`{#tbl-mytable}`	`@tbl-mytable`
Figure	`{#fig-myfigure}`	`@fig-myfigure`
Equation	`{#eq-myequation}`	`@eq-myequation`

The table could be cross referenced using a tag `{#tbl-parameters}` as follows

```
| Parameter | Value | Unit  |
|---------|:-----|------:|
| $a$      | 1  |    ms$^{-1}$ |
| $b$      | 2  |   s$^{-1}$  |
| $c$        | 3    |     Nondim  |

: Demonstration of  table {#tbl-parameters}
```

The tag can now be cross referenced using the tag (see Table 3.3).

Table 3.3: Demonstration of table

Parameter	Value	Unit
a	1	ms^{-1}
b	2	s^{-1}
c	3	Nondim

In a similar manner we can cross reference an equation (see tag `#eq-emc2`). This is cross-referenced using the handle `@eq-emc2`.

```
$$
E=mc^2
$$ {#eq-emc2}
```

So if I defined an equation

$$E = mc^2, \tag{3.1}$$

I can refer to it in the text via Equation 3.1.

Quarto allows for definition and cross referencing of a range of mathematical objects (e.g. theorems, corrolorarries)

You can find out more about cross referencing here.

3.4.5 Citations and references

In your final report will need to provide a list of references that are cited at relevant points in your document.

Thiscan be achieved relatively straightforwardly in Quarto.

You need to create a `.bib` file (e.g. `mybibliography.bib`) and save it in a sensible directory (e.g. alongside your .qmd files).

You need to populate the .bib file with bibliographic entries, each of which will have a unique tag (e.g. `my_bib_tag`)

Then in your `.qmd` file you can cite a reference using the `@' handle (e.g.`@my_bib_tag').

You can find out more about citations in Quarto here.

i Creating a .bib file

You could use a reference manager such as Mendeley or Jabref.
Alternatively, you

- go to google scholar.
- go to settings tab
- In the Bibliography manager, select -> *Show links to import citations into* Bibtex
- Now when you search for a paper/textbook in Google scholar, there should be an additional link: ‘import into Bibtex’
- Copy and paste the contents in the link into your .bib file

3.5 Appendices

3.5.1 Submitting code

You should include codes that you have developed in the appendix of your report. You can do this using a *code block*.

~~~
``` markdown
Copy and paste code here
```
~~~

3.6 Websites and blogs in Quarto

Given what you have achieved thus far, it is not a very big step to generate and publish websites, blogs and dashboards.

There are some tutorials avilable on how to do this on the Quarto pages.

For guidance on publishing material see here.

4 Typesetting mathematics

4.1 The math environment in latex

Latex is a programming language used for mathematical typesetting. In its original form a latex file is compiled to generate a .pdf file. Mathematical notation is written in the 'math environment'. You can find a detailed introduction to latex here.

Within Quarto we can access the latex math environment by enclosing text within dollar symbols.

To typeset mathematics inline (e.g. $x + y = 2$) we write …

```
To typeset mathematics in line (e.g. $x+y=2$) we write ...
```

To typeset mathematics in a new line we use double dollar symbols. To obtain the expression

$$\frac{x+y}{2} = 4,$$

we write

```
$$
\frac{x+y}{2}=4.
$$
```

It is worth spending some time familiarising yourself with basic latex commands.

Here is the same equation with a cross reference tag, i.e.

$$\frac{x+y}{2} = 4. \tag{4.1}$$

Now I can cross reference Equation 4.1.

This has been achieved using

```
$$
\frac{x+y}{2}=4.
$${#eq-myequation}
```

It is worth knowing how to:

- write a system of aligned equations

$$
\begin{aligned}
\sum_1^3 n = & \quad 1+2+3 \\
= & \quad 6 \\
\sum_1^4 n = & \quad 1+2+3+4 \\
= & \quad 10
\end{aligned}
$$

  Note that the equations are aligned such that the equal signs within the ampersands are at the same place;

- Use limits and sums, i.e.

$$\lim_{n\to\infty}\sum_{k=1}^{n}\frac{1}{k^2}=\frac{\pi^2}{6};$$

- define sets of numbers, i.e.

$$x^2 \geq 0 \qquad \text{for all} x \in \mathbb{R}$$

- have several expressions separated by some space

$$\sqrt{x^2+\sqrt{y}} \quad \overline{m+n} \quad \underbrace{a+b+\cdots+z}_{26};$$

- write a matrix

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix};$$

- use conditional statements

$$y = \begin{cases} a, & \text{if } d > c, \\ b+x, & \text{in the morning}, \\ l, & \text{all day long}; \end{cases}$$

- adjust the size of brackets

$$(\frac{4}{3}) \leftarrow \text{this is bad}. \qquad (4)$$
$$\left(\frac{4}{3}\right) \leftarrow \text{much better}$$

4.2 Exercise: an example worksheet

Try to typeset the questions below in latex:

1. Denote the roots of the equation $x^2 + 5x + 1 = 0$ by $x_1 = \alpha$ and $x_2 = \beta$.
2. $$\sin\left(3\theta + \frac{\pi}{2}\right) = \frac{1}{2} \qquad 0 \le \theta \le \pi.$$
3. Express $\dfrac{2x-26}{x^2-2x-8}$ in partial fractions.
4. $$\text{(a)} \sum_{k=1}^{500}(2k-21), \qquad \text{(b)} \sum_{k=1}^{20}\frac{(-2)^k}{5}, \qquad \text{(c)} \sum_{k=1}^{\infty}5\left(\frac{1}{3}\right)^k.$$
5. $$\int_0^{1/2} x\sqrt{1-2x}dx.$$
6. $$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} + v^2(1-v) \qquad x \in \mathbb{R}, \;\; t > 0.$$
7. $$\begin{aligned} \frac{\partial u}{\partial t} &= (a - u + u^2 v) + \frac{\partial^2 u}{\partial x^2}, \\ \frac{\partial v}{\partial t} &= (b - u^2 v) + d\frac{\partial^2 v}{\partial x^2}, \end{aligned}$$
8. $$\begin{aligned} n_t &= -(nu_t)_x + rn(1-n), \\ Nu_{xx} + (\tau n\rho)_x &= s\rho u, \\ \rho_t + (\rho u_t)_x &= 0, \end{aligned}$$

9.

$$I(t) = \begin{cases} I_0(t), & 0 \leq t < \tau, \\ I_0(t) + S(0) - S(t-\tau), & \tau \leq t < \tau + \sigma, \\ S(t-\tau-\sigma) - S(t-\tau), & \tau + \sigma \leq t, \end{cases}$$

10.

$$A = \begin{bmatrix} 0 & 5 & -2 \\ 5 & -7 & 5 \\ -2 & 5 & 0 \end{bmatrix}.$$

11.

$$f(x) = \begin{cases} x^2 & x < 0 \\ \sqrt{x+1} & x \geq 0 \end{cases}$$

12.

$$x^2 + 4x + 4 = 0. \tag{4.2}$$

Equation Equation 4.2 is a quadratic.

5 Presentations

As part of your project you will give an approx. 10 minute presentation. You will have to develop some slides to present.

Quarto has a number of different options for producing slide decks. Below we will use Revealjs.

6 Python code development

6.1 Introduction

It is strongly encouraged that you use and develop programs over the course of your project. You should discuss programming with your project supervisor. In this section we will develop Python codes to solve a number of example problems.

The focus of the examples: - developing programming skills for problem solving - writing your own programmes - using existing packages

It is usually a very useful exercise to try to implement your own version of an algorithm. However, in many cases there already exist well developed codes that we of course should use!

You can use Python via Apps Anywhere (Open Anaconda and then use the SPyder IDE).

Alternatively, you can run Python codes via your Quarto document. On the Uni machines Quarto has been connected to an installation of Python.

To run python via Quarto create a python environment

```
Insert syntax for running python.
```

6.2 Some key ideas

6.2.1 Calculator

6.2.2 Variables and datatypes

Strings Lists Tuples Sets Dictionaries

```
<class 'int'>
<class 'float'>
<class 'str'>
```

6.2.3 Lists and dictionaries

```
[1, 2, 3, 4, 5]
The second entry in the list is
2
```

6.2.4 Logical statements and control loops

6.2.5 Writing functions

A good rule of thumb is that if you find yourself using the same piece of code three or more times you should write a function. This avoids duplication of code.

Suppose we find ourselves manually computing the sum of positive integers many times, i.e.

$$s_2 = 0 + 1 + 2 = 3$$

and

$$s_3 = 0 + 1 + 2 + 3 = 6.$$

It makes sense to write a function that computes the sum for arbitrary n. Then we call that function when needed.

```
3
7140
```

6.2.6 Code debugging

- syntax errors
- runtime errors
- logic errors

```
sum=0.0
for i in range(5)
  sum=sum+i
```

```
for i in range(5):
  sum=sum+i
```

6.2.6.1

Plan your code Keep code clean (e.g. use variables, write expressiosn that are easy to interpret) Test code often Comment code

6.2.6.2 interpreting error messages

6.2.6.3 print to screen

6.2.6.4 check datatypes

6.2.6.5 check syntax of function call

6.2.6.6 Debugger

6.3 Python libraries

6.3.1 Essential

6.3.1.1 Matrix computation (numpy)

Numpy is a widely used Python libary. It is a standard way to use arrays in Python. Numpy also contains lots of algorithm (e.g. linear algebra, calculus, mathematical functions, integration, random number generation etc.). You can find a beginner's guide here.

Numpy provides tools for calculating many mathematical operations.

```
sin (3.14) is:
0.0015926529164868282
1.2246467991473532e-16
```

We can also use numpy to define and manipulate arrays. In the example below we use python lists to define two 1D arrays.

```
[1 2 3 4 5 6]
The sum of a and b is:
[ 8 10 12 14 16 18]
The first entry in a is

1
```

We can also use numpy for higher dimensional arrays

Calculate the determinant of the 2x2 matrix

$$A = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}.$$

```
The matrix is:
[[4. 3.]
 [2. 1.]]
The determinant is:
-2.0
```

1. Compute the trace of the matrix A.
2. Compute the determinant of the 3×3 matrix

$$B = \begin{pmatrix} 4 & 3 & 2 \\ 2 & 1 & 4 \\ 3 & 2 & 1 \end{pmatrix}.$$

3. Compute the eigenvalues of B.

6.3.1.2 Plotting (matplotlib)

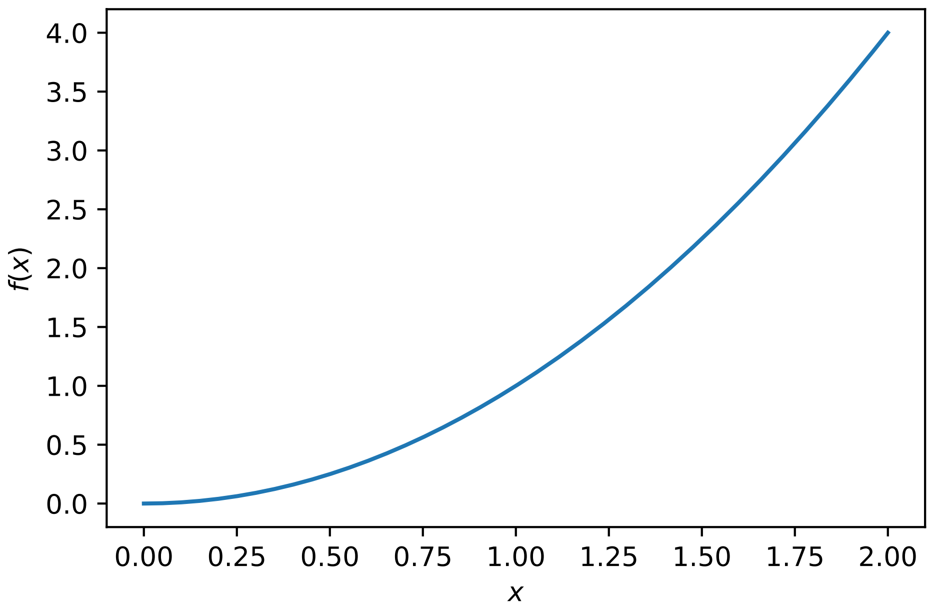
Matplotlib is a pyton library for plotting.

Let's plot some well-known functions.

Suppose we wish to plot the function

$$f(x) = x^2, \quad x \in [0, 2].$$

```
Text(0, 0.5, '$f(x)$')
```



Exercises:

1. Change the domain to
$$x \in [0, 4]$$
.
2. Plot the function
$$f(x) = \sin(x)$$
.
3. Create two subplots.
4. Make the font on the labels larger
5. Add a legend to the figures.

6.3.1.3 Symbolic computation in Python (sympy)

6.3.2 Python libraries of interest to particular projects

6.3.3 Numerically solving differential equations (sci-py)

6.3.3.1 ODEs

6.3.3.2 PDEs

6.3.4 Optimisation (skikit-learn)

6.3.5 Data analysis (pandas)

6.3.6 Machine learning (tensorflow)

6.3.7 Image analysis (skikit-image)

6.4 Writing your own scripts

At some point (either over the course of your project or later) you will likely encounter a problem that cannot be solved using existing code libraries, i.e. you will need to write your own progammes. To prepare for this day, it is a good idea to practice your code development skills on problems where the solution are already known. Over the course of project assessment you will be asked about the methods that you have used in your project. It is much easier to defend the use of a method if you have a clear idea how to programme it; then the limitations of a method become much clearer. You can also include this code in appendix in your written report.