University of Dundee

MA40001

MA40001

Philip Murray

2024-01-09

Table of contents

Preface

Welcome to the module MA40001.

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

The main aims of the module are to develop:

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

How to contact me?

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

Software

The taught component of the module will require you to use the following software:

- Quarto
- Visual Studio Code
- Github
- Latex
- Python

More details available here (see Chapter 1.1).

> **i** Why bother with all this?
>
> The aim is to provide you with *open-source* tools that enable you to generate modern scientific/mathematical documents.

Teaching materials

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 MA40001REsources.

In **?@sec-github** you will learn how to *fork* this github repository so that you have template project/presentations.

Assessment

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

Further details are available on MyDundee.

Teaching plan

Table 1: Semester 1

Week	Material	Assessment
1	Intro+Quarto	Formative
2-3	Quarto	Formative
4-6	Python	Formative
7-10	Presentation practices	Formative
11	Assessed presentation	Summative
13	Interim report	Summative

Table 2: Semester 2

Week	Material	Assessment
11	Assessed poster presentation	Summative
13	Final report submission	Summative
Exam period	Viva (aural exam)	Summative

1 Software

1.1

A version of Visual Studio Code with Quarto + other software is available via Apps Anywhere via the app:

- Visual Studio Code 1.92.2 with Quarto 1.5.56

VS Code is an integrated development environment (IDE) - a platform from within which to develop and run codes.

The Apps Anywhere app has necessary codes pre-installed for you to immediately begin working on the module.

> **Note**
>
> The AppsAnywhere app is still being finalised by UoDIT. There is currently a bug that means: - you cannot use the command————- - simultaneuosly generate multiple output formats - compile multi-file projects

1.2 On your personal computer

It is advised that you get VS Code + ncessary software running on your personal computer.

install the following free software:

- Quarto
- VS Code
- Anaconda - to get Python and then libraries (matplotlib.pyplot, numpy, pandas, etc.)
- Github Desktop

Within Quarto:

- Extensions - install the quarto extension for VS Code
- install Latex (follow these guidelines).

- Configure the Python interpreter (Show and Run Commands->Python Select Interpreter)

After doing this you should be able to follow the exercises in lectures notes on your personal computer.

2 Scientific/mathematical communication

The assessment of your project will requires you develop:

- presentation slide
- poster
- written reports (interim report and final report)

2.1 Common elements across different assessment points

2.1.1 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.

> ❗ Exercise - project narrative
>
> Define a narrative for your project:
>
> - One sentence description
> - Why is it important to study this topic?
> - Describe any relevant background
> - What are the project Aims?
> - What work have you done to address the Aims?
> - Outlook

2.1.2 Pitching the content appropriately

A key principle of 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 confused and 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 peers, i.e. Level 4/5 of a undergraduate mathematics degree. This means that you should *not* assume that the audience have specific knowledge of the details/background of your project.

> ! Exercise - Pitching your project
>
> Practice pitching your project at different target audiences:
>
> - A high school student with Higher mathematics (*how much mathematical detail should you give?*)
> - A researcher in the field that you are working (*how much intro will they need?*)
> - A family member/flatmate.
> - At a job interview (*focus on parts of the project that are relevant*)

2.1.3 Equations

You will almost certainly use mathematical notation in your assessment.

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

2.1.3.1 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.

2.1.4 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:

$$f(x, y) = \cos(x) \sin(xy). \tag{2.1}$$

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

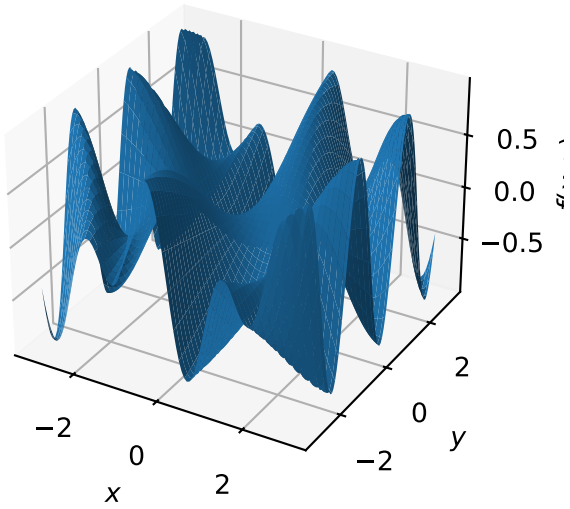
- Is the function periodic?
- How many maxima/minima does it have?
- How does it behave for $x, y \to \infty$?

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 2.1). 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.

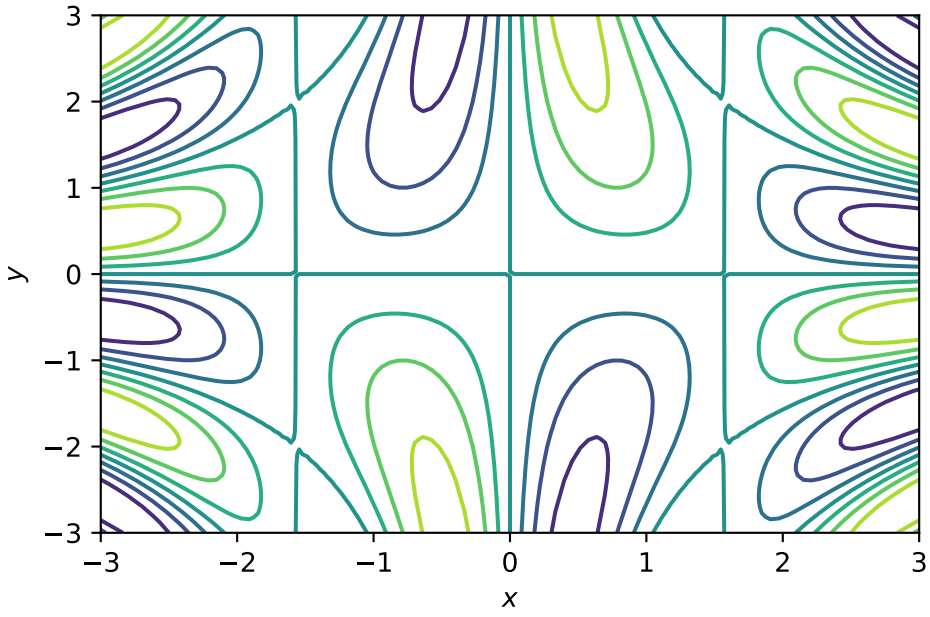
A good rule of thumb is: if figures and tables are removed from the text, does the text still read coherently? i.e. the figure is helping the reader to understant a point that is made within the text. It is *not* replacing the need for text.

Other types of plots are useful for different purposes:

- bar chart (Figure 2.2)
- pie chart (Figure 2.3)
- scatter plot (Figure 2.4)



(a) A 3D plot. $f(x, y)$ is plotted against x and y



(b) A contour plot. Contours of $f(x, y)$ are plotted against x and y.

Figure 2.1: 3D plot of Equation 2.1

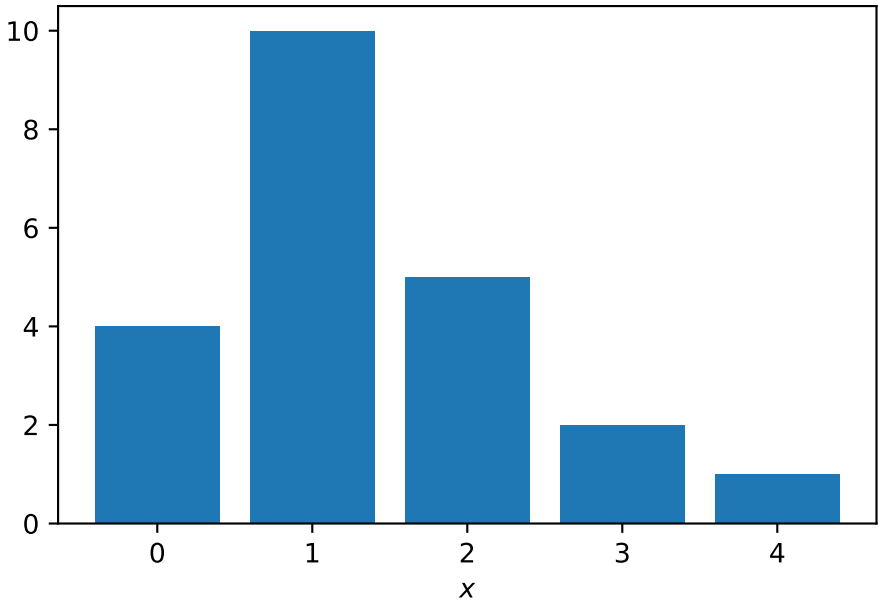


Figure 2.2: A bar chart.

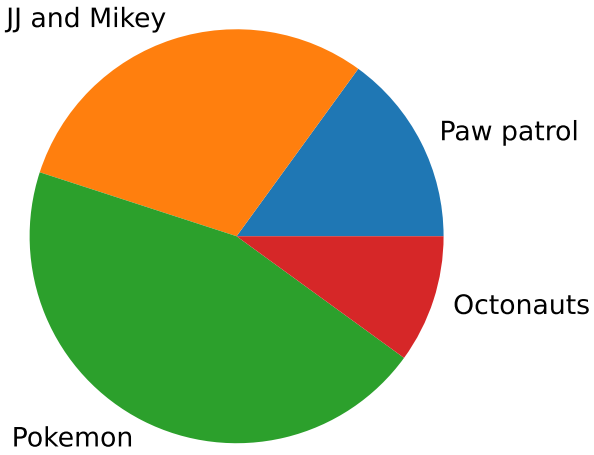


Figure 2.3: A pie chart.

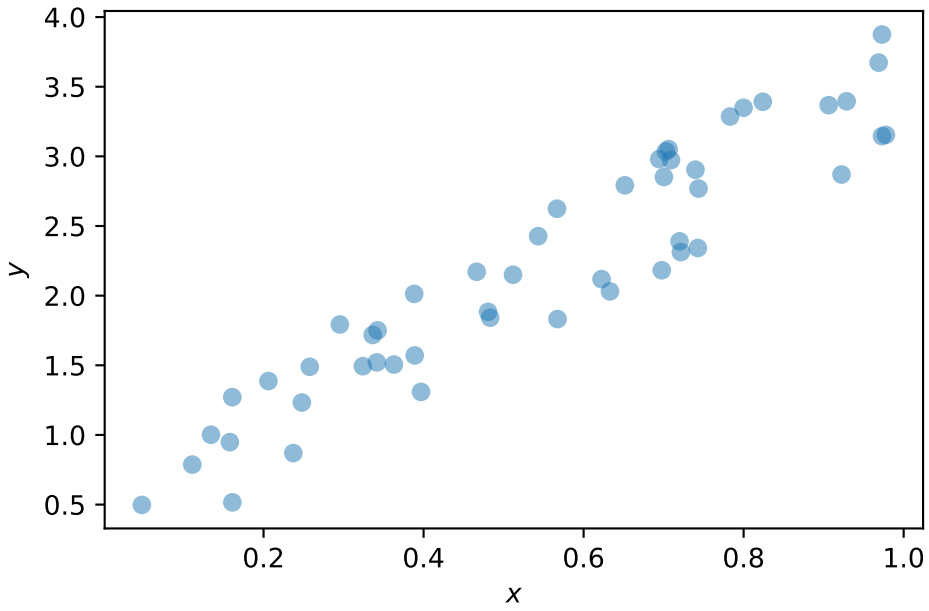


Figure 2.4: A scatter plot.

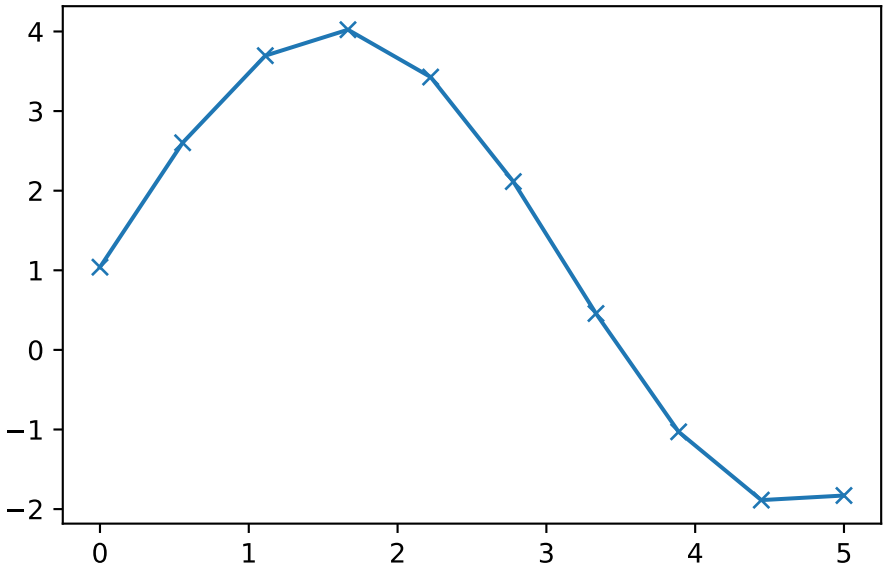


Figure 2.5: y is plotted against x.

common mistakes

- hanging figures (i.e. figures that are not referenced/placed in context in the text (e.g. context for the information in Figure 2.3 is not described in the text. Figure 2.5 is not dven referenced)
- figures without axis labels (Figure 2.5 has no axis labels)
- coloured graphs without colour scale/legend
- labels/ tick marks are too small
- type of graph unsuitable for the data shown
- 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:

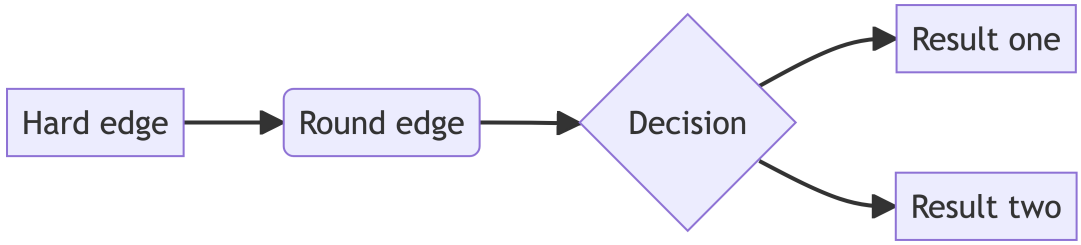
- check that any text in axis labels/legends is approximately the same size as font in the main text

2.1.5 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 (Chapter 4)) provides an interface to a number of graph-based tools that generate schematic diagrams, e.g.



2.1.6 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.

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 author's original words. It should be the author's 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.)

i 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).

2.2 Use of AI

Large language models (LLMs), such as ChatGPT, Dal-E, are being used to generate increasingly complex outputs (e.g. images, text). AI is also increasingly used in the development of codes (e.g. CodePilot).

LLMs use statistical patterns in datasets upon which they have been trained to generate responses to natural language queries. To the untrained eye, the output can seem very con-

vincing and, depending on the query, it can be accurate. But the output can also be completely incorrect.

Using AI-generated materials without attributing the source is a form of plagiarism. Hence if you use generative AI in your project it must be clearly declared in your *Use of AI* statement (see template project).

2.3 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

- Use of AI Statement
- (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 present any codes that you have developed.

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

2.4 Verbal presentation

At the end of Semester 1 you will be asked to give an approx. 10 minute verbal presentation of your project. More information will be provided in Weeks 7-10.

A template presentation is available via the MA40001Resources github repository.

3 Quarto

4 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.

4.1 Getting started

Open VS code - Apps Anywhere Visual Studio Code 1.92.2 with Quarto 1.5.56 extension

4.2 A first markdown document

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

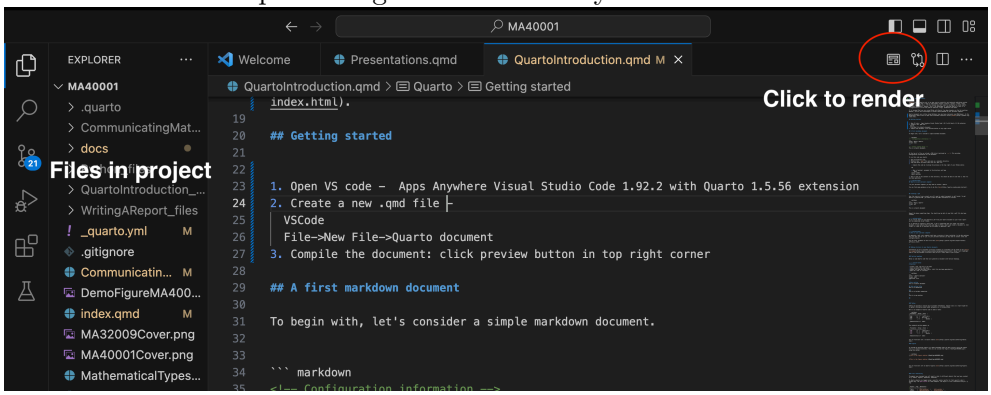
```
<!-- Configuration information -->
---
title: "Hello, Quarto"
format: html
---

<!-- 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.

❗ Exercise

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 file has been generated.

4.3 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
```

> **!** Exercise
>
> Repeat the above compiling steps. You should now be able to see that a pdf file has been created in the directory alongside your .qmd file.
> 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.

4.4 Adding structure to your Quarto documents

Information on how to document structural elements to a document 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.

4.4.1 Section headings

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

```
---
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
```

```
##
```

> **!** Exercise
>
> - Create a new .qmd file in VS Code
> - Copy and paste the above code.
> - Render the code and check that a .html file has been generated.
> - Add new section headings.

4.4.2 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 4.1: 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.

> **!** Exercise
>
> - Copy and paste the above table into your .qmd file.
> - Check that it renders.
> - Add another column to the table and render.

- Add another row to the table and render.

4.4.3 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)
```

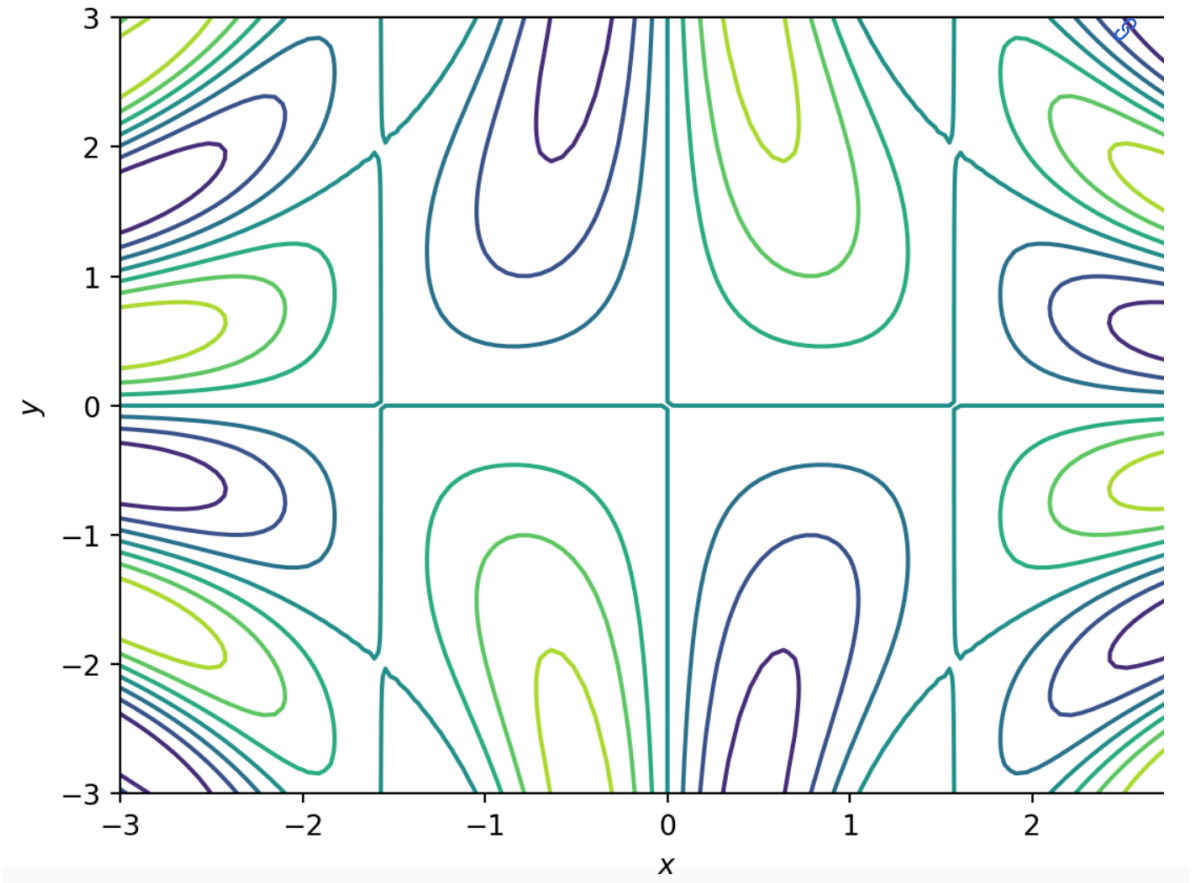


Figure 4.1: This is the figure caption.

You can find more info on Quarto figures here.

! Exercise

- Choose a .png file to include in your document.
- Save the .png file in the same directory as your .qmd file.
- Include the image and render your Quarto document.

4.4.4 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 4.2 for some examples. The object is cross referenced using an '@' tag.

Table 4.2: 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 4.3).

Table 4.3: 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

$$E = mc^2, \tag{4.1}$$

I can refer to it in the text via Equation 4.1.

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

You can find out more about cross referencing here.

Exercise

- Tag the table and figure that you made above.
- Cross-reference them in your document.

4.4.5 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.

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
- Cite the source in your Quarto document.

4.4.6 Schematic diagrams

You can learn how to make diagrams here. Here is an example:

````
```{mermaid}
flowchart LR
  A[Hard edge] --> B(Round edge)
  B --> C{Decision}
  C --> D[Result one]
  C --> E[Result two]
```
````

4.5 Appendices

4.5.1 Submitting code

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

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

4.6 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.

5 Typesetting mathematics

5.1 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.

$$\frac{x+y}{2}=4. \tag{5.1}$$

Now I can cross reference Equation 5.1.

This has been achieved using

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

It is worth knowing how to:

- write a system of aligned equations

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

  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

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

5.2 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. $$\sin\left(3\theta + \frac{\pi}{2}\right) = \frac{1}{2} \qquad 0 \leq \theta \leq \pi.$$
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.

$$x^2 + 4x + 4 = 0. \tag{5.2}$$

Equation Equation 5.2 is a quadratic.

6 File management

6.1 Introduction

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.

6.1.1 Github exercises

1. Launch the Github Desktop app on Apps Anywhere. To deal with a bug in appsAnywhere:
    - Open CloudpagingPlayer (little blue icon on bottom right of your screen): * Stop Github Desktop.
    - Remove Github Desktop.
    - Relaunch Github Desktop in AppsAnywhere.
2. You will need to create a github account if you do not currently have one.
3. Go to the MA40001Resources github page.
    - Follow the instructions here for *forking* the repository. This creates a fork of the repository in your github account.
    - Follow the instructions for cloning the repository. This creates a copy of the repository on your local machine.
    - You should now have a local copy of the resources on your computer and own a new (forked) repository.
4. Launch VS Code on AppsAnywhere and open the local copy of your MA40001Resources project.

6.1.2 Updating your repository from github

You can

Fetch origin Pull origin (update your local version of files)

View the files of your repository in Explorer

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

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

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

- Create a new file.
- Commit the file and push it to github.
- Check that the version of the file in the cloud has changed to reflect your edit.

6.1.5 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 the file *dummy_file.txt* - 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.

6.2 Using the command line interface

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

7 Presentations

As part of your project you will give an approx. 10 minute presentation. To do this, 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.