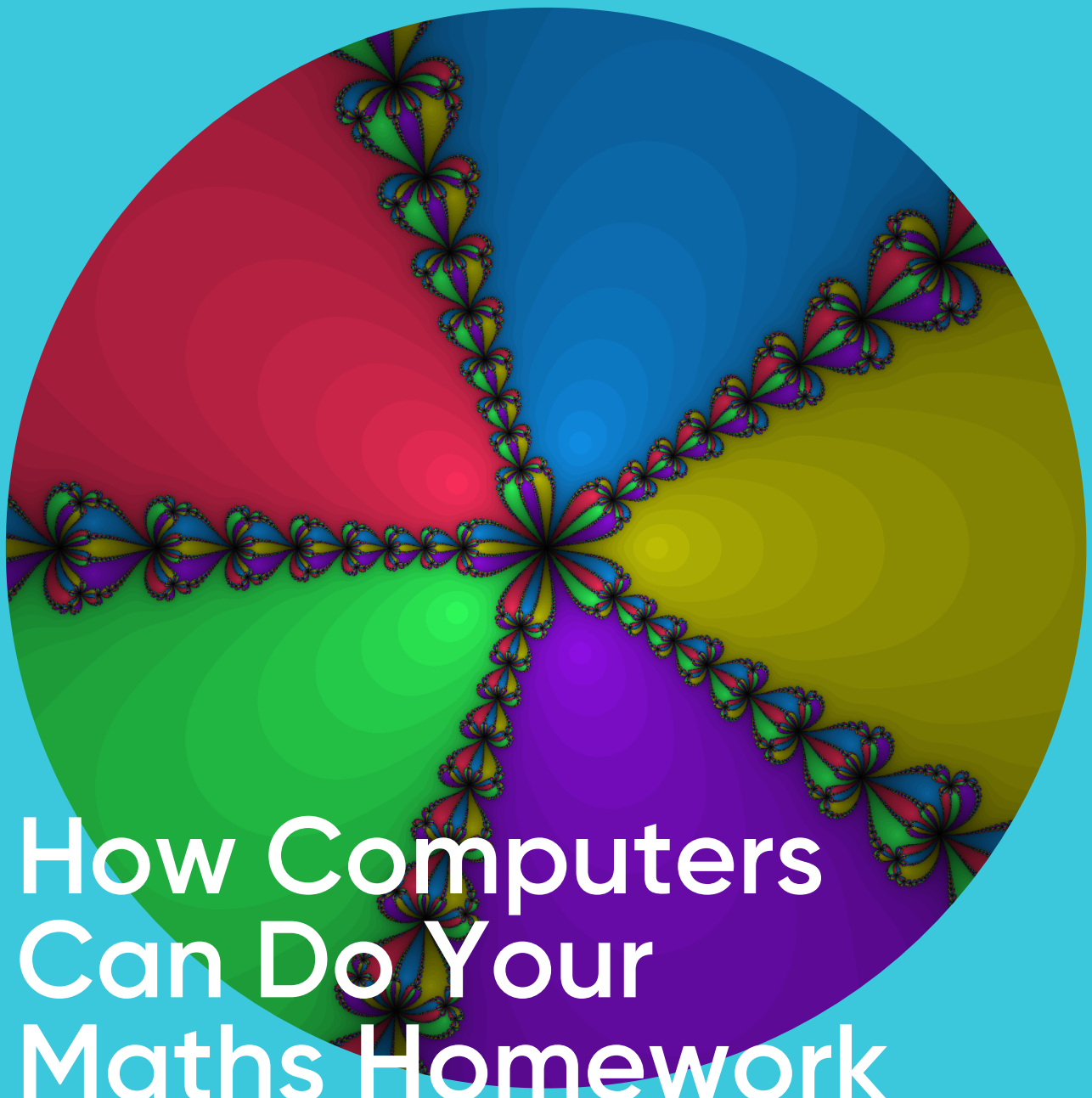


How Computers Can Do Your Maths Homework For You

Pupil Name

Handbook Designed by Dr. Durston

Timetable and Assignment Submission

Timetable – Tutorials

Tutorial	Date	Time	Location
1			
2			
3			
4			
5			
6 (Feedback)			
7 (Feedback)			

Timetable – Homework Assignments

Homework Assignment	Description	Due Date
Tutorial 1		
Tutorial 2		
Tutorial 3		
Tutorial 4		
Tutorial 5		

Assignment Submission – Lateness and Plagiarism

Lateness	
Submission after midnight on 9th August	10 marks deducted
Plagiarism	
Some plagiarism	10 marks deducted
Moderate plagiarism	20 marks deducted
Extreme plagiarism	Automatic fail

Uni Pathways – Pupil Feedback Report

Grade	Marks	What this means
1st	70+	Performing to an excellent standard at A-level
2:1	60-69	Performing to a good standard at A-level
2:2	50-59	Performing to an excellent standard at GCSE
3rd	40-49	Performing to a good standard at GCSE
Working towards a pass	0-39	Performing below a good standard at GCSE
Did not submit	DNS	No assignment received by The Brilliant Club

Lateness	
Any lateness	10 marks deducted
Plagiarism	
Some plagiarism	10 marks deducted
Moderate plagiarism	20 marks deducted
Extreme plagiarism	Automatic fail

Name of RIS teacher	Dr. Durston		
Title of Assignment			
Name of Pupil			
Name of School	Bristol Metropolitan Academy		
ORIGINAL MARK / 100		FINAL MARK / 100	
DEDUCTED MARKS		FINAL GRADE	

If marks have been deducted (e.g. late submission, plagiarism) the PhD tutor should give an explanation in this section:

Knowledge and Understanding
Critical Evaluation/Problem Solving
Structure and Presentation
Resilience Comment

Contents

Course Rationale 5

Mark Scheme 6

Glossary of Keywords 8

Knowledge Organisers 10

Tutorial 1 – Big Problems 11

Scientific Footage 12

Functions in Mathematics 13

Homework: The Baseline Assignment 15

Tutorial 2 - Roots 17

Functional Programming 18

Roots 19

The Intermediate Value Theorem 21

Tutorial 3 – Bracketing Methods 24

The Bisection Method 25

Errors 25

The False Position Method 27

Tutorial 4 – Iterative Methods 30

Gradients 31

The Secant Method 33

The Newton-Raphson Method 35

Tutorial 5 – Convergence 37

Convergence and Divergence 38

Order 39

Homework: The Final Assignment 41

Tutorial 6 - Feedback 42

Tutorial 7 - Feedback 44

Appendices 47

Referencing Correctly 47

Course Rationale

> "Without mathematics, there's nothing you can do. Everything around you is mathematics. Everything around you is numbers."
>
> Shakuntala Devi

Modern science is underpinned by mathematics. Maths provides us with the tools needed to pull apart the clockwork of the universe and figure out how everything works. From predator-prey models in Biology, to resolving forces in Physics and calculating reaction rates in Chemistry, Maths turns up everywhere!

But let's be realistic. Maths can be difficult.

Many of the challenges that crop up in modern research boil down to complicated mathematical problems which can't easily be solved by hand. Trying to solve the equations that come up in weather forecasting, molecular modelling and astrophysical simulations with just pen and paper would be a tremendous challenge! That's where computers come in.

Maths has an interesting relationship with computing. Much of the logic that forms the foundation of computer science comes directly from mathematics. At the same time, the ability to perform billions of calculations quickly and accurately is invaluable to mathematicians. As such, we can say that computer science is built on and builds up mathematics.

This course is about the latter of those two concepts; how we can use computers to help us solve mathematical problems. We will focus on one of the most fundamental and essential mathematical challenges: solving equations.

Mark Scheme

Skills	1st (70-100)	2:1 (60-69)	2:2 (50-59)
Knowledge and Understanding	• All content included is relevant to the general topic and to the specific question/title • Good understanding of all the relevant topics. • Scientific terms are defined and used accurately throughout • Clear justification on how the content included is related to the specific issues that are the focus of the assignment	• Most of the content included is relevant to the general topic and to the specific question/title • Good understanding of most the relevant topics • Scientific terms are used accurately but not always clearly defined. • Adequate justification on how the content included is related to the specific issues that are the focus of the assignment	• Some of the content included is relevant to the general topic and to the specific question/title • Good understanding on some of the relevant topics but occasional confusion on others. • Scientific terms are used mostly accurately with occasional confusion and often not defined. • Some justification on how the content included is related to the specific issues that are the focus of the assignment
Critical Evaluation / Problem solving	• Moved beyond description to an assessment of the value or significance of what is described • Evaluative points are consistently explicit/systematic/reasoned/justified • Effective critiques on the reliability of sources provided • Consistently demonstrate clear, analytical and logical steps to solving problems	• Mostly description but some assessment of the value or significance of what is described • Evaluative points are mostly explicit/systematic/reasoned/justified • Some evidence of critiques on the reliability of sources provided • Some examples of solving problems but not consistently clear, analytical and logical	• Only description with minimal assessment of the value or significance of what is described • Evaluative points are at times explicit/systematic/reasoned/justified • Limited evidence of critiques on the reliability of sources provided • Demonstrate clear steps to solving problems but not consistently analytical or logical

Structure and Presentation	• Ideas are presented in paragraphs and arranged in a logical structure that is appropriate for the assignment • The introduction clearly outlines how the essay/report will deal with the issues • The conclusion summarises all the main points clearly and concisely • All calculations, formulas and methods are clearly structured, clear to follow and correct • Tables and graphs are effectively constructed including appropriate headings, units and scales. • All sources are referenced correctly in an agreed format	• Ideas are presented in paragraphs and arranged in a structure that is mostly appropriate for the assignment • The introduction adequately describes how the essay/report will deal with the issues • The conclusion summarises most of the main points clearly • Calculations, formulas and methods are mostly structured, clear to follow and correct • Most tables and graphs are well constructed • Most sources are referenced correctly in an agreed format	• Ideas are presented in paragraphs and arranged in a structure • The introduction mentions how the essay/report will deal with the issues • The conclusion summarises some of the main points clearly • Calculations, formulas and methods are not always structured, clear to follow and correct. • Some tables and graphs are well constructed but contains some errors • Some sources are referenced correctly in the agreed format with occasional errors

Subject Vocabulary

Word	Definition	In a sentence
simulation		
function		
input		
output		
evaluate		
root		
interval		
gradient		
reciprocal		
bracketing method		
iteration		
maximum error		
estimate		

tangent		
iterative method		
recurrence relationship		
derivative		
convergence		
divergence		
order		
logarithm		

Knowledge Organisers

$$f(x) = \underbrace{x^2 - 1}$$

The name of the function, e.g. "f"

The input

How to calculate the output

The **root** of a **function** is the point, x, where $f(x) = 0$.

Interval bracket → $[0,4]$

Lower bound

Upper bound

An **interval** represents all values between (and including) 2 points.

	Bracketing Methods	Iterative Methods
Error at step n	$\frac{I_n}{2}$	$x_n - x_{n-1}$

where I_n is the size of the interval at step n.

A key results of...

The Intermediate Value Theorem (IVT):

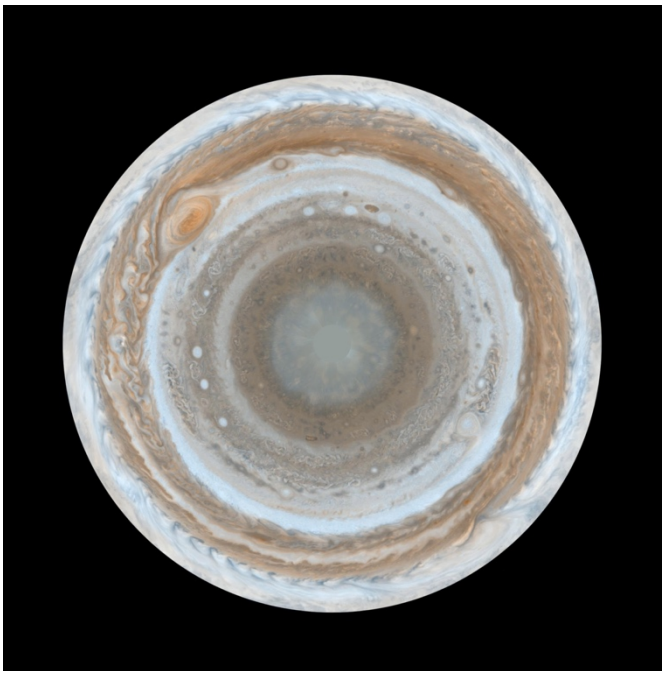
There is a root, $f(x) = 0$, on an interval $[a, b]$ if

$$f(a) \times f(b) \leq 0$$

	Method	Description	Inputs	Key Equations	Always Converges	Never Diverges	Potential Order	f evaluations/step
Bracketing Methods	Bisection	Cuts interval in half, checks which half contains the root.	$[a, b]$	$c = \frac{a+b}{2}$	✓	✓	1	1
	False Position	Uses straight line to cut interval, checks which part contains the root.	$[a, b]$	$c = b - f(b)\frac{(b-a)}{f(b)-f(a)}$		✓	1	1
Iterative Methods	Secant	Estimates tangent (using two points) to guess the root.	x_n x_{n-1}	$x_{n+1} = x_n - f(x_n)\frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$			~1.6	1*
	Newton-Raphson	Uses actual tangent (from derivative function) to guess the root.	x_n	$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$			2	2

* The secant method requires 2 function evaluations on its first step, then 1 for every step afterwards.

Tutorial 1 – Big Problems



What is the Purpose of Tutorial 1?

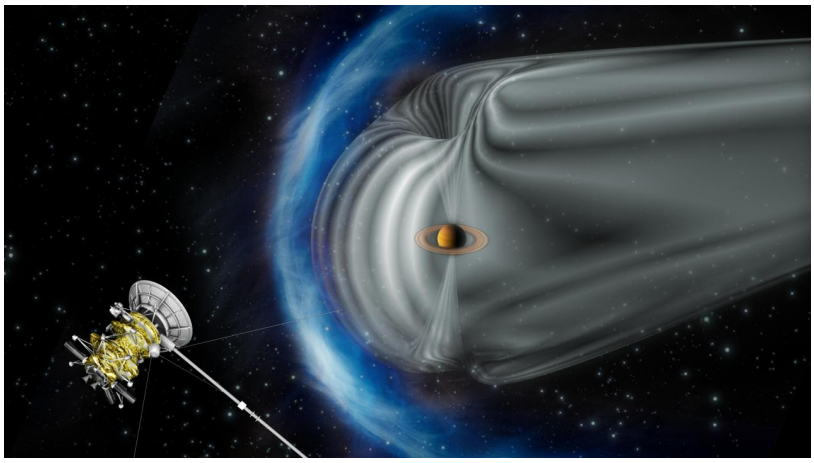
- Think about how scientific research is performed.
- Consider why we use computers to solve problems instead of performing experiments.
- Learn how to calculate the output of a function from an input.

Starter

Think of one surprising fact about yourself. Be prepared to share it with the class!

Scientific Footage

Watch Video 1. What is happening? Write down at least 3 things.



The Cassini-Huygens mission was a collaboration between NASA, ESA and ASI (the Italian Space Agency). It was launched in October 1997 and arrived at Saturn in July 2004, performing flybys of Venus, Earth and Jupiter along the way. After 13 years observing Saturn, the probe ended its mission in September 2017. During its 20 year lifespan, Cassini collected a lot of important data about Saturn and its moons. In late 2000, Cassini collected detailed information and pictures of Jupiter, as seen in tutorial 1's video.

Why is it difficult to learn about Jupiter?

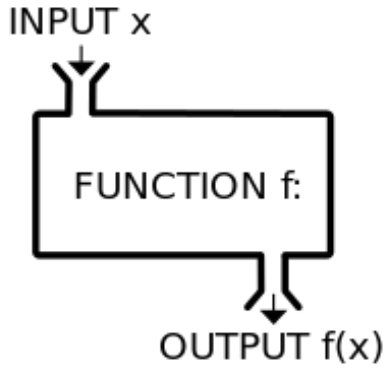
Watch Video 2. What are the differences and similarities between the two videos?

Similarities:	Differences:

Why might we use a computer to solve a problem instead of performing experiments or collecting data? Write down as many ideas as possible!

Why are simulations sometimes difficult to perform?

Functions in Mathematics



A mathematical **function** is a way of relating an **input** to an **output**.

An example of a function is:

$$f(x) = \underbrace{x^2 - 1}$$

The name of the function, e.g. "f"

The input

How to calculate the output

When making a function, you can choose any name the function or the input variable!

Other examples are:

$$g(a) = 3a + 1$$

$$\mathrm{sqrt}(b, c) = \sqrt{b + c}$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

... wait, haven't I seen that last one before?

Yes! That's how we calculate the trigonometric function "sine". Some useful functions are very complicated, which is why we use calculators and computers to calculate them.

We find the output of a function by **evaluating** it at a certain point. For example, if we take $f(x)$ above, we find:

$$f(3) = 3^2 - 1 = 9 - 1 = 8$$

So $f(3) = 8$. Likewise, $f(6) = 35$ and $f(-2) = 3$.

Evaluate the following functions at the given points:

$f(x) = \frac{1}{x^2+1}$ $x = 2$ $x = -1$	$f(x) = 2^x - 1$ $x = 2$ $x = 3$

Make up your own function. Chose two points to evaluate it. Be sure to show your working!

Isn't this just the same as using $y = \ldots$ *?*

Yes! When we write equations like $y = 3x + 2$, we are saying "y is a function of x". We could write:

$$y(x) = 3x + 2$$

If we rearrange the equation to put x on its own, we have $x = \frac{y-2}{3}$, so we can write "x is a function of y":

$$x(y) = \frac{y-2}{3}$$

In the above equations, we have reversed the input and output, but the relationship between them remains the same.

Mathematical functions are important because they are a clear way of writing the relationship between 2 or more numbers.

Homework: The Baseline Assignment

1. Write an essay about how computers can be used to solve scientific problems. [300 words]

Your essay should include one specific example of how computers have contributed to a scientific or mathematical problem. For example, consider:

- Breaking the Enigma Code
- The Human Genome Project
- The Travelling Salesman Problem
- The Four-Colour Theorem
- Weather Forecasting
- Anything else you can find!

2. Investigate the "bisection.py" program.

Download and run bisection.py. What seems to be happening? Write your initial observations below.

Open the program in the editor. Identify the main section of the code.

Try changing the number assigned to `n`. Run the program. Write down the new value of `n` (e.g. `n = 50`) and what happens. Try a few different numbers.

Try changing the value of `a`. What happens? How about `b`?

Identify the section of the code starting with:

```
def f(x):
```

The next line contains the word `return` followed by some symbols. What do you think this represents?

Try changing the code after the word `return` (e.g. to `x - 3`). What happens? Try a few different things, and write down what you changed the code to.

What do you think the code is doing?

Challenge. As much as possible, comment on each line or section of the code (using `#`) and describe what it is doing.

Tutorial 2 – Roots

> "When solving problems, dig at the roots instead of just hacking at the leaves."
>
> Anthony J. D'Angelo

What is the Purpose of Tutorial 2?

- Practice writing functions on a computer.
- Learn about roots of functions.
- Understand the Intermediate Value Theorem.

Starter

Solve the following problems:

$5x + 3 + 2(x + 9) = 0$	$x^2 - 2x - 3 = 0$

Calculate the value of the input, x, which gives the output $f(x) = 0$ for:

$f(x) = 2x - 10$	$f(x) = (x - 1)^2$

Functional Programming

Functions (sometimes called *subroutines* or *modules*) are also important parts of computer programs. Like mathematical functions, program functions accept inputs and outputs (most of the time). Unlike mathematical functions, program functions often accept a wide range of inputs (e.g. letters, symbols, dates, etc) and sometimes involve very complicated outputs. In this course, we will only look at how to write a mathematical function in a computer program.

The way we write a function in a program depends on the program's syntax. Syntax is like grammar but for programming languages instead of spoken languages. In python, there are several different ways of writing a function. In this course, we will only use the following:

```
def f(x):
  return x**2 - 1
```

How would you write the above function in mathematical notation? (Hint: we saw this function in the first tutorial!)

The function acts like a separate, standalone part of the program. To make use of the function we have to call it in the main program. The way we call the program mostly depends on the input it requires and the type of output the function produces. Here are a couple of examples:

```
a = f(-2)
print a                                  # displays "3" in the terminal

x = 6
print f(x) + f(a)                        # displays "38" in the terminal

if f(x) < f(a):
  print 'f(x) is less than f(a)!'        # doesn't display anything
```

Roots

The **roots** of a function are any input points which make the function output equal zero, i.e. $f(x) = 0$. To find the root(s), replace $f(x)$ with 0, and rearrange to find x on its own. For example:

$$f(x) = \frac{x}{3} + 2$$

Find the root of $f(x)$.

First, set the left-hand-side of the function equal to zero:

$$0 = \frac{x}{3} + 2$$

Next, rearrange to find x:

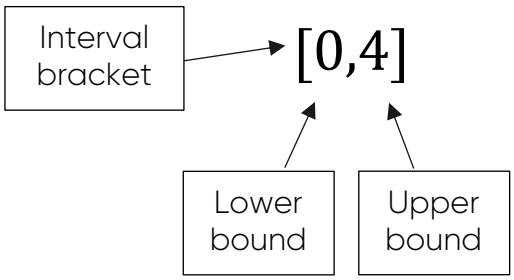
$$-2] \qquad -2 = \frac{x}{3} \qquad [-2$$

$$\times 3] \qquad -6 = x \qquad [\times 3$$

So the root of $f(x)$ is $x = -6$.

Intervals

An **interval** is a range of numbers. We write intervals in this form:



$[0,4]$ means every number between 0 and 4, including 0 and 4 themselves. When we say "every number" in this course, we always mean "every number" – not only whole numbers! This includes any number that can be written as a decimal or fraction. So our interval $[0,4]$ includes the numbers 2.5, $1/3$ and $0.\dot{4}$.

In this course, we will only look at simple roots; a root on an interval with no other roots. It is important to be aware that if there is more than one root on an interval, a computer program may not be able to find both roots, and may not work properly.

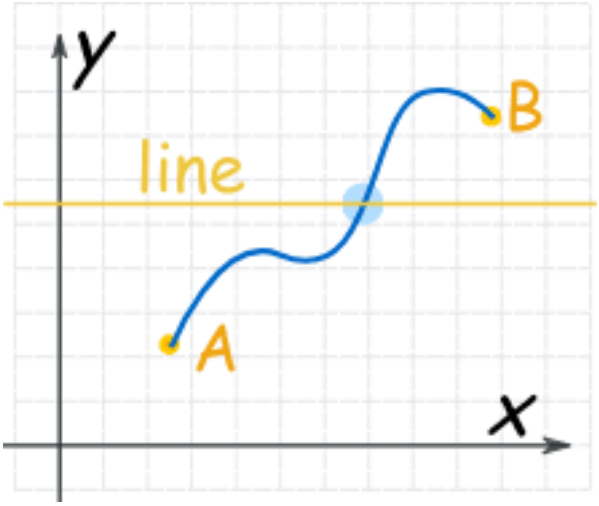
Activity. Take a card. On one side is a function, on the other is an interval. Don't show anyone else the function side! The aim is to find the root of your partner's function, without knowing what the function is.

Tell your partner an input. Your partner must calculate the output using their own function. Write these down in the table below. Try to find the root, i.e. the input which gives an output of "0". Switch roles when you've finished!

Input	Output	Input	Output	Input	Output	Input	Output	Input	Output

The Intermediate Value Theorem

Imagine a function $f(x)$ and an interval $[a, b]$. The output of the function and the beginning and end of the function will be $f(a)$ and $f(b)$.



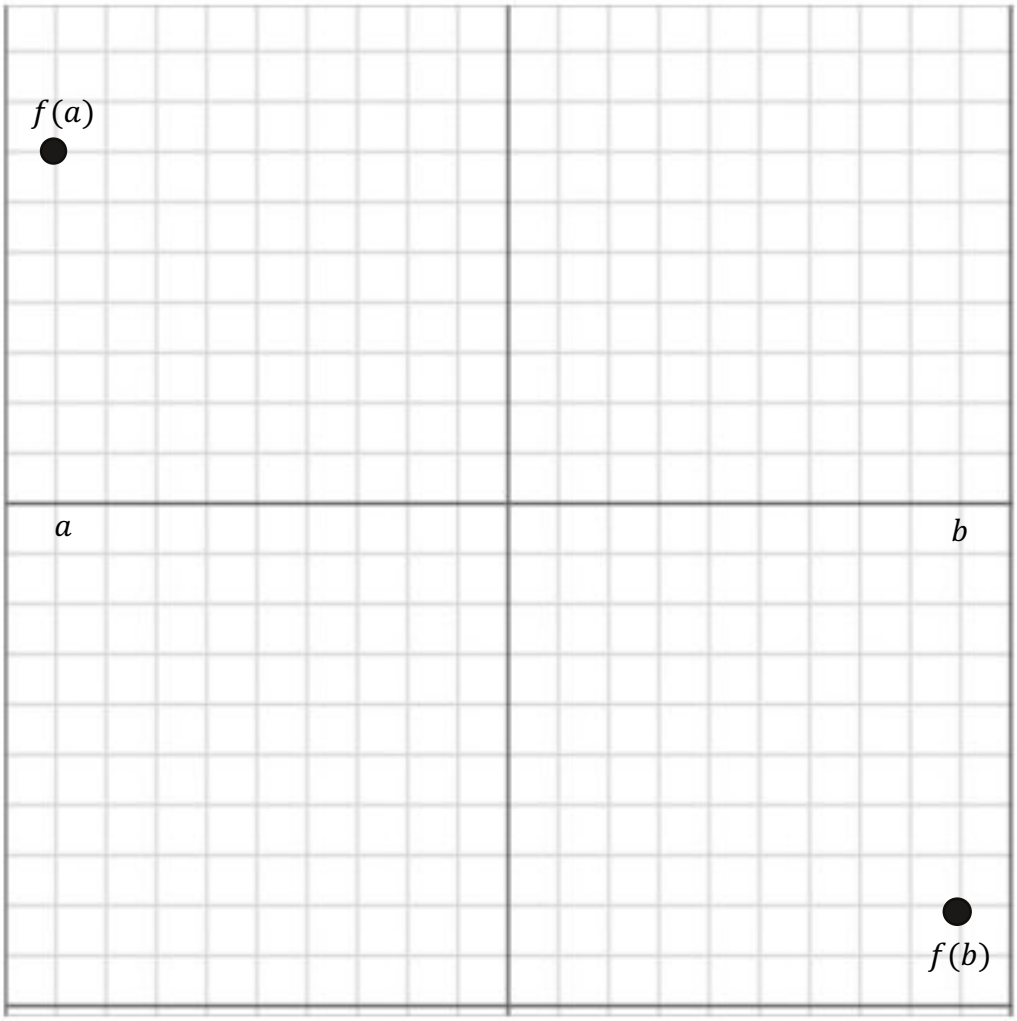
The Intermediate Value Theorem (IVT) states that on the interval $[a, b]$, the output of f will take every value between $f(a)$ and $f(b)$ at some point.

A result of this is that if:

- $f(a)$ is less than zero
- and $f(b)$ is greater than zero
- or vice-versa
- or either $f(a)$ or $f(b)$ is equal to zero

then $f(x)$ **must** pass through zero at some point on the interval $[a, b]$.

Below is a graph with $f(a) > 0$ and $f(b) < 0$. Without taking your pencil off the paper, can you draw a line connecting $f(a)$ and $f(b)$ without passing through the x-axis or leaving the interval?



As a reminder, the requirements for an interval $[a, b]$ to contain a root $f(x) = 0$ are:

$$\begin{matrix} f(a) > 0 \\ f(b) < 0 \end{matrix} \quad \text{or} \quad \begin{matrix} f(a) < 0 \\ f(b) > 0 \end{matrix} \quad \text{or} \quad f(a) = 0 \text{ or } f(b) = 0$$

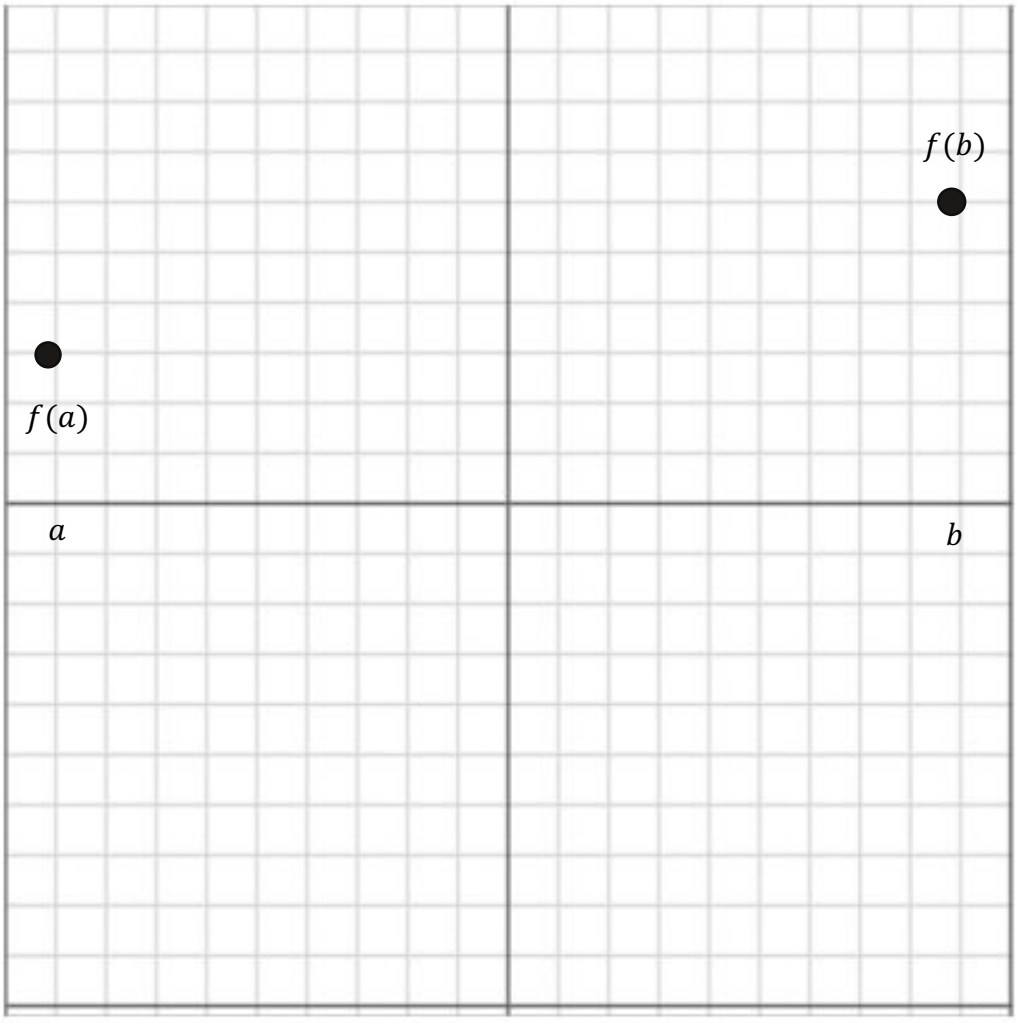
For each of the requirements above, will $f(a) \times f(b)$ be **positive**, **negative** or **zero**?

So, we can check one condition to see if there is a root on an interval:

There is a root $f(x) = 0$ on an interval $[a, b]$ if $f(a) \times f(b)$ $\quad$ 0.

If $f(a) \times f(b) > 0$, does it mean there are no roots on the interval?

Without taking your pencil off the page, try drawing a line from $f(a)$ to $f(b)$ that passes through the x-axis on the graph below. Is it possible?



The IVT **cannot** tell you if there **are no** roots on an interval; only if there **definitely is** a root.

For each of the functions and intervals below, use the IVT to decide if there definitely is a root on the interval or not.

$f(x) = -4x + 2$ on $[0,4]$

$g(x) = 2x + 10$ on $[-5,5]$

$h(x) = x^2 + x - 2$ on $[-3,3]$

How about the same function on $[0,3]$?

Homework

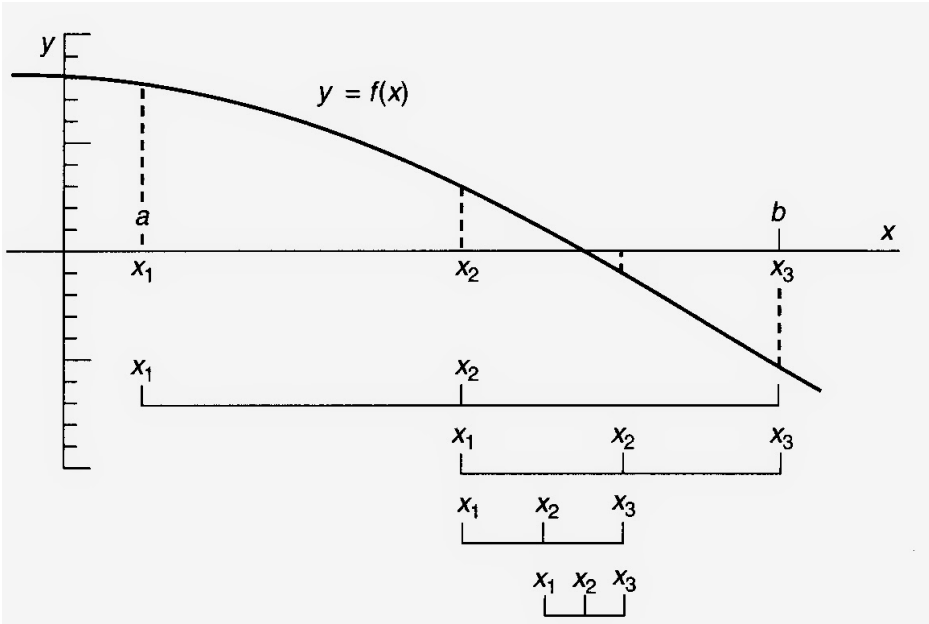
Write a piece of python code which uses the IVT to tell you if a function has a root on a given interval. Test the code using the functions and intervals above. Print or write down the code and bring it to the next tutorial.

(Hint: Your code should include an `if` statement which checks the sign of $f(a) \times f(b)$.)

Challenge. Convert your code into a standalone function, `ivt(f,a,b)` , which takes a function and interval as inputs, and outputs:

- 1 if there is a root.
- 0 if the IVT is inconclusive.

Tutorial 3 – Bracketing Methods



What is the Purpose of Tutorial 3?

- Use the bisection method to find roots.
- Learn about the errors made by computers.
- Compare the bisection and false position methods.

Starter

Use the IVT to determine whether there is a root of $f(x) = x^2 - 1$ on:

$[-2, 0]$	$[1, 5]$

The Bisection Method

The **Bisection Method** is an algorithm that can be used to narrow the interval a root is on. Starting with a function $f(x)$ and interval $[a, b]$ which **must** have a root on it, the algorithm is as such:

1. Find the mid-point of the interval, $c = (a + b)/2$.
2. Check if c is the root.
    - If c is the root, end algorithm.
3. Check if the lower interval, $[a, c]$, contains the root.
    - If $[a, c]$ contains the root, make it the "new interval" ($[a, b] \leftarrow [a, c]$).
    - If $[a, c]$ doesn't contain the root, make $[c, b]$ the "new interval" ($[a, b] \leftarrow [c, b]$).
4. Repeat from step 1.

When we finish the algorithm, we usually use the midpoint of our final interval as our guess for the root.

Errors

Algorithms are designed to be repeated in order to get a good estimation of the root. Each repeat is called an **iteration**. But how many iterations should we do?

The answer to that question depends on how accurate we want our guess to be. Most of the time (but not always) the more iterations of the algorithm we do, the more accurate our result is.

For the bisection method, we can calculate the **maximum error** at each step. This is the distance between the guess (the midpoint of the final interval) and the furthest point the **actual root** could be.

Let's calculate the **maximum error for the bisection method**!

At the start the bisection method, we select an interval, $[a, b]$. Let's call the size of the interval, I_n, where n is the number of iterations so far. After any step, the furthest the actual root could be from the guess would be half the length of the new interval:

$$\text{max error} = \frac{1}{2} \times I_n$$

Before we start using the algorithm, our first interval will be:

$$I_0 = b - a$$

The error (before we even begin) will be:

$$\text{max error} = \frac{1}{2} \times I_0$$

After the 1st step, our new interval is half the size of the original interval:

$$I_1 = \frac{1}{2} \times I_0$$

So the maximum error after the 1st step is:

$$\text{max error} = \frac{1}{2} \times I_1$$

$$= \frac{1}{2} \times \frac{1}{2} \times I_0$$

$$= \frac{1}{4} \times I_0$$

For the 2nd step, write the new interval, I_2, in terms of I_1:

Write I_2 in terms of I_0 (the starting interval):

Using your result above, write the 2nd step max error in terms of I_0:

Compare the 1st step max error and 2nd step max error. In words, can you describe what happens to the max error after each step?

Can you describe this mathematically? Try writing the max error as a function of a and b.

The False Position Method

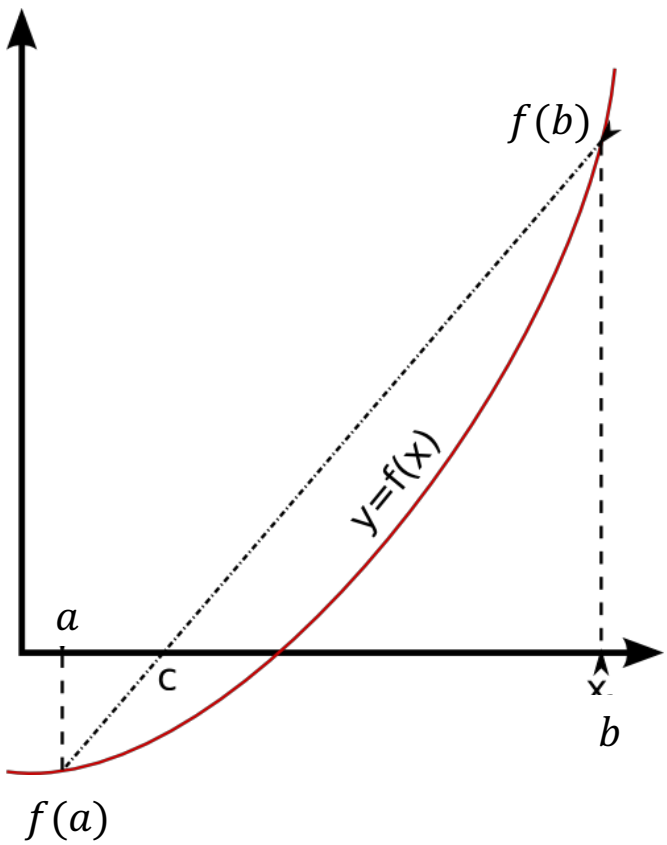
In the bisection method, we:

- took an interval, $[a, b]$
- looked at the midpoint of the interval, c
- checked if the root was below c or above c.

What if, instead of the midpoint of the interval, we choose a different position for c?

In the **False Position Method** (sometimes called *Regula Falsi*) we:

- draw a straight line between $f(a)$ and $f(b)$,
- use the point where that line crosses the x-axis as c,
- do the rest the same as the bisection method (check to see if the root is below c or above c).



Finding c

The **gradient** of a straight line, m, is how steep it is:

$$\text{gradient} = \frac{\text{change in y}}{\text{change in x}}$$

For the straight line in the false position method, $y = f(a)$ when $x = a$, and $y = f(b)$ when $x = b$. We find m by:

$$m = \frac{f(b) - f(a)}{b - a}$$

Now we know how steep the line is. But we also need to know which points it goes through in order to write an equation for it. We use this formula:

$$y - y_* = m(x - x_*)$$

where x_* and y_* are the x and y coordinates of any points the line passes through. We know that $y = f(a)$ when $x = a$, and $y = f(b)$ when $x = b$, so we can use either of these pairs. We'll use the latter:

$$y - f(b) = m(x - b)$$

Although we know what m is, we're not going to put it back in yet in order to save space. However, if we did, we would have an equation which accurately described the false position line!

Now we have to find the point c. We know that c is on the x-axis, which means that $x = c$ when $y = 0$, which we can put into our previous equation:

$$0 - f(b) = m(c - b)$$

$$-f(b) = m(c - b)$$

$$\div m] \qquad -\frac{f(b)}{m} = c - b \qquad [\div m$$

$$+b] \qquad c = b - \frac{f(b)}{m} \qquad [+b$$

Now let's put m back in. We're doing $f(b) \div m$, which is the same as $f(b) \times (1/m)$. We call $1/m$ the **reciprocal** of m, commonly known as "m flipped". This gives us:

$$c = b - f(b)\frac{(b - a)}{f(b) - f(a)}$$

You might be thinking: "That was a lot of work! What was the point?" Don't worry, we'll get to that soon!

Implementing the False Position Method

Other than the choice of c, the false position method works exactly the same as the bisection method! The algorithm is:

1. Calculate c using the equation given on the previous page.
2. Check if c is the root.
    - If c is the root, end algorithm.
3. Check if the lower interval, $[a, c]$, contains the root.
    - If $[a, c]$ contains the root, make it the "new interval" ($[a, b] \leftarrow [a, c]$).
    - If $[a, c]$ doesn't contain the root, make $[c, b]$ the "new interval" ($[a, b] \leftarrow [c, b]$).
4. Repeat from step 1.

You can change your bisection code to make it false position by changing just ONE line in the program. Save the file under a new name, then give it a go!

"What was the point?"

For the bisection method, we saw that the maximum error halved after each iteration of the algorithm. Unfortunately, we can't work out a general rule for how the error decreases in the false position method – it depends on the function and the starting interval.

However, we can investigate using our computer programs!

Activity. In your bisection and false position programs, add a line of code in the main `for` loop which calculates the maximum error at that step:

$$\text{error} = \frac{b - a}{2}$$

Compare the error for the bisection method with the error for the false position method for at least 3 different functions and intervals. Record your observations below.

Challenge. Try to find at least one example of where bisection is better than false position, and one where false position is better than bisection. What kind of function does the false position method find the root of immediately?

Homework

Complete that activity above.

Tutorial 4 – Iterative Methods

> "Few ideas work on the first try. Iteration is key to innovation."
>
> Sebastian Thrun

What is the Purpose of Tutorial 4?

- Understand how gradients can be used to estimate the root.
- Consider the differences between bracketing and iterative methods.
- Compare the secant and Newton-Raphson methods.

Gradients

As we've seen, the false position method can be very powerful when used in certain situations. The strength of the method comes from its use of gradients. Can we improve on this?

So far, we've only looked at gradients of straight lines, with equations in the form $y = mx + c$. We can calculate these gradients accurately and easily.

However, it is much more difficult (and sometimes impossible) to calculate accurate gradients of functions which are not straight lines. It is often these functions that we want to find the roots of.

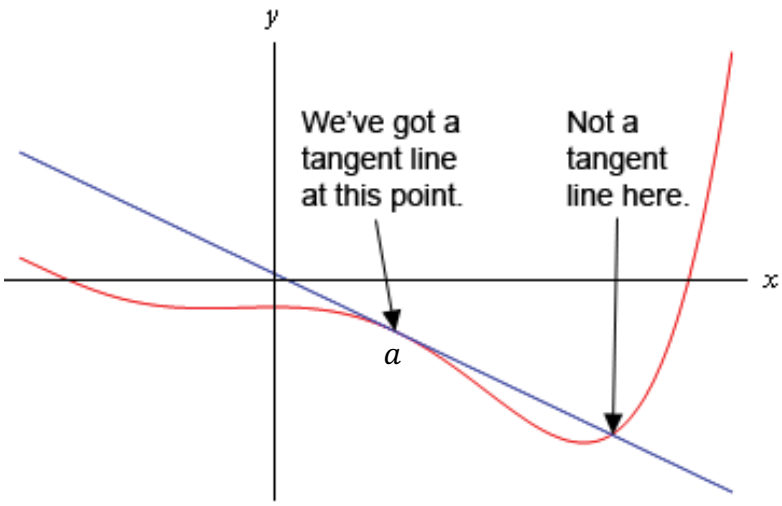
Luckily we can **estimate** the gradient of any function!

Tangents

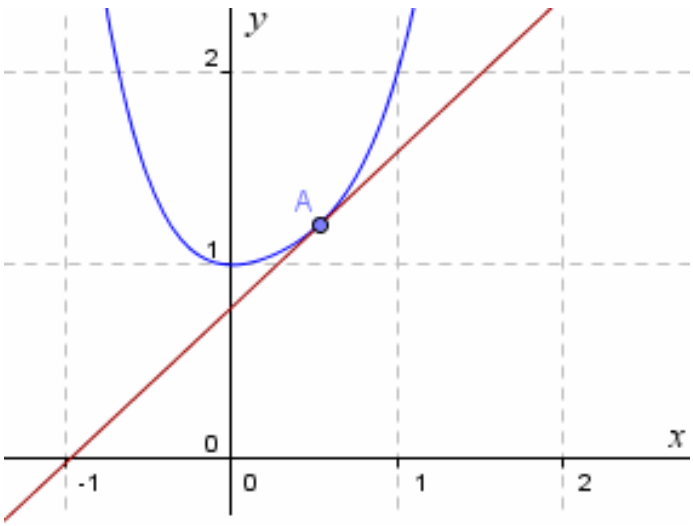
The **tangent** of $f(x)$ at a is the straight line which has the same gradient as $f(x)$ specifically at the point $x = a$.

For the graph on the right, the red line is a curve, $f(x)$, while the blue line is the tangent to $f(x)$ at the point a.

The blue line crosses the curve again, however, it is **not** a tangent at that point, as it does not have the same gradient.



We can estimate a tangent line "by eye" simply by drawing in the line, taking some measurements and writing the equation of the line. Let's do an example together!



In the graph on the left, the blue curve represents $f(x)$ and the red line is the tangent we've already drawn in.

We can calculate the gradient of our tangent by:

$$\text{gradient} = \frac{\text{change in y}}{\text{change in x}}$$

As y goes from 0 to 2, x changes from -1 to 1.5. So

$$m = \frac{2 - 0}{1.5 - -1} = \frac{2}{2.5} = \frac{4}{5}$$

We've calculated the gradient. To construct the line we just need to put one point, (x_*, y_*) into the following equation:

$$y - y_* = m(x - x_*)$$

Our tangent clearly goes through the point $(-1,0)$, so let's use that:

$$y - 0 = m(x - -1)$$

$$y = m(x + 1)$$

So, using the value we found for m:

$$y = \frac{4}{5}x + \frac{4}{5}$$

This equation describes the line that we have estimated is tangent to the curve $f(x)$.

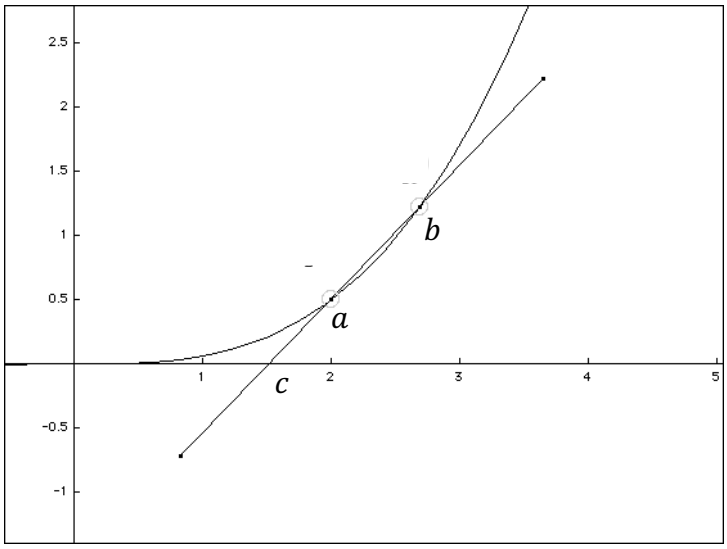
As we have seen, it's quite easy for humans to spot and estimate tangents. However, we want our computer program to be able to do this (so that we can use it to find a root).

On page 28, we saw that knowing two points on a line allows us to calculate the gradient and then the equation of that line. This is how a computer can estimate a tangent.

First we take two points. Generally, if these two points are close together, the line that goes through them will be a good estimate of the tangent.

Let's call these points a and b. We calculated the gradient of the straight line which passes through two points, a and b, on page 28:

$$m = \frac{f(b) - f(a)}{b - a}$$



We then found the equation of the line to be:

$$y - f(b) = m(x - b)$$

Finally, we found the point where this line crossed the x-axis, c, to be:

$$c = b - f(b)\frac{(b - a)}{f(b) - f(a)}$$

So the above equation for c tells us where the estimated tangent to a curve will cross the x-axis. In other words, **where the root for the tangent to $f(x)$** is.

Iterative Methods

So far, our algorithms (bisection and false position) have all used **bracketing** to find the root. At each step, these methods split the current interval into two, and check which one contains the root using the IVT.

For the rest of the course, we will use **iterative methods** to find roots. These algorithms take an initial guess of the root, and use a **recurrence relationship** to find the next guess. This relationship is normally just in the form of an equation, for example:

$$x_{n+1} = f(x_n)$$

The equation above tells us that the next guess of the root will be f of the previous guess. If the previous guess was 1 and the function we were finding the root of was $f(x) = 4x - 2$, then our next guess would be 2. The guess after that would be 6, then 22, etc. (Important note: The above recurrence relationship is clearly a very bad root finding method!)

Complete the following table:

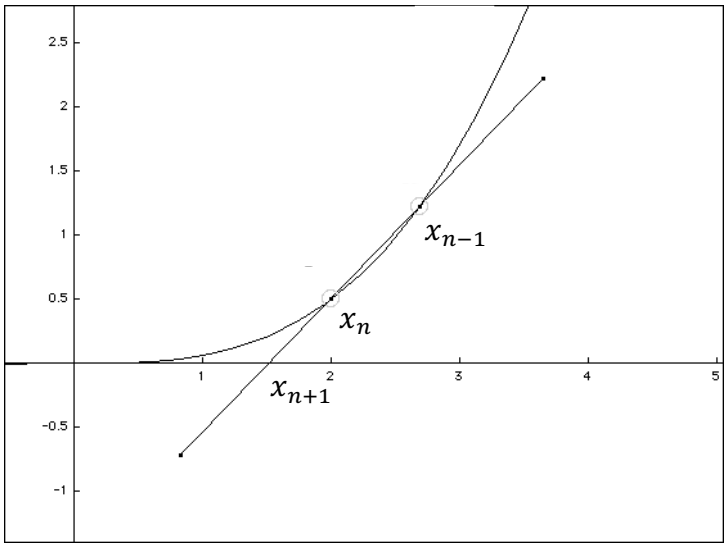
Relationship	Function	x_0	x_1	x_2	x_3	x_4
$x_{n+1} = x_n - 1$	-	2				-2
$x_{n+1} = f(x_{n-1})$	$f(x) = \frac{x}{3}$		18	9		
$x_{n+1} = g(x_n) - x_{n-1}$	$g(x) = 2^x$	0	1			

The Secant Method

The Secant Method is an iterative algorithm that guesses the root of a function, $f(x)$, based on the root of an estimation of the tangent of that function.

The secant method uses two previous guesses, x_{n-1} and x_n, to find the new guess of the root, x_{n+1}. We can replace a, b and c from our equation on page 32 by x_{n-1}, x_n and x_{n+1} to find the recurrence relationship:

$$x_{n+1} = x_n - f(x_n)\frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$



As an example, let's use the secant method to find the root of $f(x) = x^2 - 1$ starting with $x_0 = 4$, $x_1 = 0.25$. Check the answer to x_2, then use your calculator to fill in the missing boxes to 2 decimal places:

x_0	x_1	x_2	x_3	x_4	x_5
4	0.25	0.47			

More Errors

For bracketing algorithms such as the bisection and false position methods, we measured the error at each step as being the distance between the guess (the middle of the interval) and the edge of the interval. This makes sense, as we know the root will be in the interval, we just don't know where. The result will always be half the length of the interval, $I_n/2$.

For iterative methods such as the secant method, we don't use intervals, so we can't measure the error in the same way. Instead, we measure the difference the difference between the current guess and the previous guess. We'll talk more about this next time!

This table summarises the error at step n, for each method:

	Bracketing Methods	Iterative Methods
Error at step n	$\frac{I_n}{2}$	$x_n - x_{n-1}$

The Newton-Raphson Method

The false position and secant methods **estimate** the gradient of the function, $f(x)$, to guess its root. On page 28 we found the following equation, which tells us the point, c, where the **tangent** crosses the x-axis:

$$c = b - \frac{f(b)}{m}$$

In the above equation, b is another point where we think the root might be (i.e. a previous guess), and m is the gradient of the tangent.

For the secant method, we estimated the gradient using:

$$m = \frac{f(b) - f(a)}{b - a}$$

This allowed us to create an iterative method based on two previous guesses to find the root. If we leave the m in the recurrence relationship, we have:

$$x_{n+1} = x_n - \frac{f(x_n)}{m}$$

But what if we already knew the gradient, and didn't need to estimate it?

The **Newton-Raphson Method** (sometimes simply called *Newton's Method*) is an iterative method that uses the **derivative** of $f(x)$ to find its root:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

The derivative, $f'(x)$, is a function that, when given an input, x, tells you the gradient of f at that point.

We can calculate the derivative of some functions using an incredibly powerful and important mathematical tool called *calculus*. If you study Maths at A-level, you will learn a lot more about this! Unfortunately, we won't have time to cover this massive topic in this course, so all derivative functions will be given to you instead.

Use the Newton-Raphson method to find the root of $f(x) = x^2 - 1$, which has the derivative $f'(x) = 2x$, starting with $x_0 = 4$. Compare your answer with that given for x_1, then complete the table (to 2 decimal places):

x_0	x_1	x_2	x_3	x_4	x_5
4	2.13				

Have a look at the table on page 34, which shows the results for the same function using the secant method. How do they compare?

An important difference between the two methods is that the secant method needs two previous guesses to estimate the tangent, but the Newton-Raphson method only needs one previous guess, from which it calculates the **actual** tangent.

While this provides Newton-Raphson with advantages, there are disadvantages such as:

- Needing to know the derivative function
- Requiring more function evaluations per step

Homework

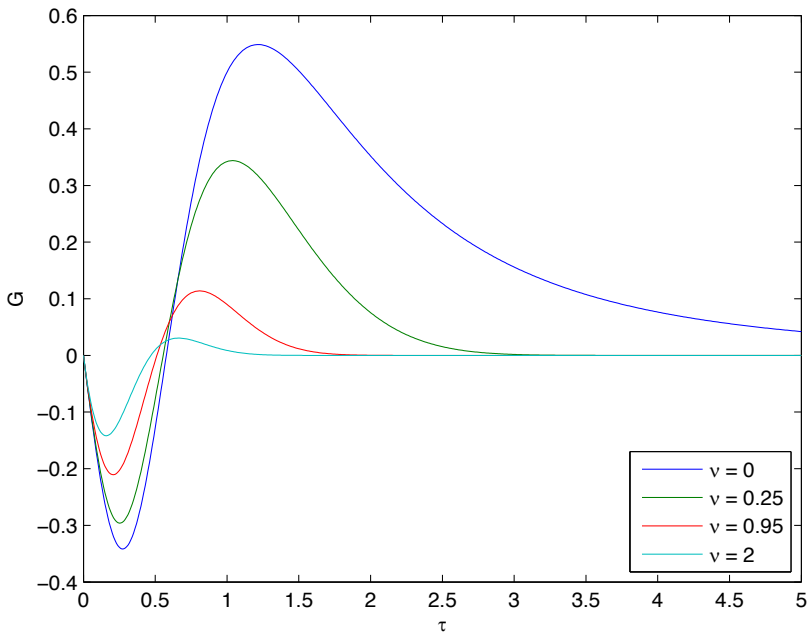
Using the template provided, create a program which performs the secant method **or** the Newton-Raphson method. Like the activity at the end of tutorial 3, compare your chosen method to the bisection and false position methods.

- Test at least 3 different functions.
- Choose your initial intervals (bracketing methods) and initial guesses (iterative methods) appropriately.
    - It's impossible to use the same starting points, as each method requires different inputs ($[a, b]$, x_0 and x_1, or only x_0). Try to choose starting points that are roughly the same distance from the root when testing.
- **Record your results** in a table showing the error for at least 5 steps for each method.

To obtain derivative functions for the Newton-Raphson method, visit wolframalpha.com and type "differentiate" followed by the required function in python script. For example: "differentiate x**2 – 1" gives $f'(x) = 2x$ for $f(x) = x^2 - 1$.

Challenge. Program and compare both the secant and Newton-Raphson methods.

Tutorial 5 – Convergence



What is the Purpose of Tutorial 5?

- Learn about convergence, divergence and orders.
- Apply and test your programming skills.
- Practice presenting scientific results and data.

Convergence and Divergence

Two of the most important words when discussing algorithms are **convergence** and **divergence**.

For a specific function, $f(x)$, and starting guess, an algorithm…

… **converges** if it continues to get closer and closer to a single point. In this course, the single point must be the root. Bracketing methods converge if the interval keeps getting smaller, enclosing the root. Iterative methods converge if each guess is closer to the root than the last.	… **diverges** if it never moves towards a single point. This means the guess gets continually bigger or smaller (towards $\pm\infty$). Bracketing methods never diverge, as the interval can't get bigger than it starts. Iterative methods diverge if each guess is further from the root than the last.

Activity. Test each of the methods we have covered for convergence/divergence. Complete the table (use pencil in case you have to change your decision).

Method	Always Converges	Never Diverges	Tests			
			Function	Initial	Converges	Diverges
Bisection		✓				
False Position		✓				
Secant						
Newton-Raphson						

Convergence and divergence tell us whether an algorithm has been successful with a particular function or not. However, they don't tell us anything about how fast the algorithm finds the root.

Order

Important note. This final section is an advanced topic which we won't cover in detail in this course. As such, some concepts have been simplified.

To investigate how efficient an algorithm is, we look at the **order** of its error. It's easier to describe this mathematically than by words.

Let's call the error at step E_n. Because we are using algorithms which repeat the same operation at each step, we expect the error at any step to be related to the error at the step before it:

$$E_n = m(E_{n-1})^p$$

Here, m is a constant (any number), and p is the "power" of E_{n-1}, called the *order* of the error. Think about what powers do to numbers:

$x^1 = x$ $2^1 =$ $0.1^1 =$	$x^2 = x \times x$ $2^2 =$ $0.1^2 =$	$x^3 = x \times x \times x$ $2^3 =$ $0.1^3 =$

So, big powers make small errors even smaller. Mathematically, if $p > 1$ and $E_{n-1} < 1$, then $E_n < E_{n-1}$. Additionally, the larger p is, the faster the error will shrink. This means, **the higher the order, the more efficient the algorithm**.

Here are some points about the language of describing orders:

- "The algorithm has order 5 convergence" or "the algorithm converges at order 5" means $p = 5$.
- The order of most algorithms changes depending on the function and initial guess. As such, we often talk about the "highest potential order" of an algorithm.
- For orders 1, 2 and 3, we have special words:
  - If $p = 1$, the algorithm converges **linearly**.
  - If $p = 2$, the algorithm converges **quadratically**.
  - If $p = 3$, the algorithm converges **cubically**.
- Additionally, if $1 < p < 2$, the algorithm converges **super-linearly**.

Remember when we calculated the error of the bisection method? (See page 25). We found that the error halved at each step:

$$E_n = \frac{1}{2}E_{n-1}$$

Here, we have $m = 0.5$ and $p = 1$. As such, we can say that **the bisection method converges linearly**.

The false position method also converges linearly. However, the value of m changes depending on the function $f(x)$. Sometimes it will be bigger than 0.5, sometimes it will be smaller. So even though both the false position and bisection methods are linear, they will converge differently.

For more information on the order of the root-finding methods we have seen in this course, see the summary on page 10.

In our programs, we can calculate the order from the current error and previous error using a handy function called a **logarithm**:

$$p = \frac{\log\left(\frac{E_n}{E_{n-1}}\right)}{\log\left(\frac{E_{n-1}}{E_{n-2}}\right)}$$

We won't cover logarithms in this course, but it is important to know that:

- $\log(1) = 0$, so if $E_{n-1} = E_{n-2}$ the order will be impossible to calculate.
- $\log(x)$ is undefined for $x \leq 0$, so once the error is exactly zero, it will be impossible to calculate the order.

To estimate the order in your program, first make sure you have imported the math module (at the very beginning):

```
import math
```

Next, make sure you have saved the errors at 3 consecutive steps. (Note: remember that the estimate is more accurate the closer to the root you are). You might want to use conditional statements (`if`) to get these! Then you can calculate the order by:

```
p = math.log(En/En1)/math.log(En1/En2)
```

Compare the python code above to the equation on the previous page. How would you write the variables in mathematical notation?

En	En1	En2

We expect the order to change depending on the algorithm, function and initial guess. It is important to be aware that, in general, this only applies to guesses close to the actual root. In other words, an algorithm will only show its "true order" when it has already done an appropriate number of steps and is converging. During the first few steps, the calculated order is likely to jump around and change a lot.

Homework: The Final Assignment

Investigate and compare two root-finding methods.

One of your chosen algorithms can be from those we have studied during the tutorials:

- The Bisection Method
- The False Position Method
- The Secant Method
- The Newton-Raphson Method

At least one of your chosen algorithms **must** be from your own independent research. Using books, articles or the internet, look up an algorithm we haven't covered in this course, explain it in your own words, program it and compare the results. Examples include:

- Steffensen's Method
- Halley's Method
- Modifications to the False Position Method:
    - The Illinois Algorithm
    - The Andersen-Björk Algorithm
- The Inverse Quadratic Interpolation Method
- Muller's Method
- Fixed Point Iteration

Your investigation should be written as a scientific report, with written analysis, figures and tables of results. You should include the following sections:

1. General introduction.
    i. Explain root-finding in your own words and introduce the aims of the investigation.
2. Method 1 introduction.
    i. Explain what you know/have researched about this method. This can include information about researchers, history, etc.
    ii. Give a mathematical description of the method.
    iii. Produce and include relevant figures (graphs, diagrams, etc).
3. Method 1 procedure.
    i. Explain how you implemented your chosen method. This should include an annotated copy of your program.
4. Method 2 introduction. (As above).
5. Method 2 procedure. (As above).
6. Results.
    i. Consider errors, convergence, divergence and order.
    ii. Refer to anything you found during your research (e.g. if you have confirmed the order).
    iii. Produce tables of data.
    iv. Don't include every result you obtain, only important results which show unique information.
7. Conclusion.
    i. Compare and contrast the two methods.

The written part of your report (not including annotated programs) should be at least 1000 words long.

Tutorial 6 – Feedback

What is the Purpose of Tutorial 6?

- To receive feedback on final assignments.
- To share examples of best practice with the other pupils in your group.
- To write targets for improvement in school lessons.
- To reflect on the programme including what was enjoyed and what was challenging.

Final assignment feedback

What I did well…	What I could have improved on…
•	•
•	•
•	•

My target for future work is…

Tutorial 7 – Feedback

What is the Purpose of Tutorial 6?

- To receive feedback on final assignments.
- To share examples of best practice with the other pupils in your group.
- To write targets for improvement in school lessons.
- To reflect on the programme including what was enjoyed and what was challenging.

Final assignment feedback

What I did well…	What I could have improved on…
•	•
•	•
•	•

My target for future work is…

Reflecting on Uni Pathways

What did you most enjoy about Uni Pathways?
•   •   •

What did you find challenging about the programme?	How did you overcome these challenges?
• • •	• • •

Appendix 1 – Referencing Correctly

When you get to university, you will need to include references in the assignments that you write, so we would like you to start getting into the habit of referencing in your Brilliant Club assignment. This is really important, because it will help you to avoid plagiarism. Plagiarism is when you take someone else's work or ideas and pass them off as your own. Whether plagiarism is deliberate or accidental, the consequences can be severe. In order to avoid losing marks in your final assignment, or even failing, you must be careful to reference your sources correctly.

What is a reference?

A reference is just a note in your assignment which says if you have referred to or been influenced by another source such as book, website or article. For example, if you use the internet to research a particular subject, and you want to include a specific piece of information from this website, you will need to reference it.

Why should I reference?

Referencing is important in your work for the following reasons:

- It gives credit to the authors of any sources you have referred to or been influenced by.
- It supports the arguments you make in your assignments.
- It demonstrates the variety of sources you have used.
- It helps to prevent you losing marks, or failing, due to plagiarism.

When should I use a reference?

You should use a reference when you:

- Quote directly from another source.
- Summarise or rephrase another piece of work.
- Include a specific statistic or fact from a source.

How do I reference?

There are a number of different ways of referencing, and these often vary depending on what subject you are studying. The most important to thing is to be consistent. This means that you need to stick to the same system throughout your whole assignment. Here is a basic system of referencing that you can use, which consists of the following two parts:

1. A marker in your assignment: After you have used a reference in your assignment (you have read something and included it in your work as a quote, or re-written it your own words) you should mark this is in your text with a number, e.g. [1]. The next time you use a reference you should use the next number, e.g. [2].

2. Bibliography: This is just a list of the references you have used in your assignment. In the bibliography, you list your references by the numbers you have used, and include as much information as you have about the reference. The list below gives what should be included for different sources.

    a. Websites – Author (if possible), title of the web page, website address, [date you accessed it, in square brackets].

    E.g. Dan Snow, 'How did so many soldiers survive the trenches?', http://www.bbc.co.uk/guides/z3kgjxs#zg2dtfr [11 July 2014].

    b. Books – Author, date published, title of book (in italics), pages where the information came from.

    E.g. S. Dubner and S. Levitt, (2006) Freakonomics, 7-9.

    c. Articles – Author, 'title of the article' (with quotation marks), where the article comes from (newspaper, journal etc.), date of the article.

    E.g. Maev Kennedy, 'The lights to go out across the UK to mark First World War's centenary', Guardian, 10 July 2014.

Notes

Notes

Notes

How Computers Can Do Your Maths Homework For You

Key Stage 4 Programme

researchersinschools.org