Do Now

Complete these questions on your mini-whiteboards.

Task 1

Differentiation the following expressions.

1. $3x^3 + 2x$
2. $\frac{1}{4}x^5 - \frac{1}{x}$

Task 2

Integrate the following expressions.

1. $3x^3 + 2x$
2. $\frac{1}{4}x^5 - \frac{1}{x}$

As you can see, I am not Amalia ...

As you can see, I am not Amalia ...

Thomas James MMathPhys
(Preferred: Tom)

As you can see, I am not Amalia ...

Thomas James MMathPhys
(Preferred: Tom)

Experience

As you can see, I am not Amalia ...

Thomas James MMathPhys
(Preferred: Tom)

Experience

- A-Levels in Maths, Chemistry, Physics, and Further Maths at *The Manchester College*.

As you can see, I am not Amalia ...

Thomas James MMathPhys
(Preferred: Tom)

Experience

- A-Levels in Maths, Chemistry, Physics, and Further Maths at *The Manchester College*.
- Master's degree in Mathematics and Physics from *University of Manchester*.
  - Specialised in calculus.

As you can see, I am not Amalia ...

Thomas James MMathPhys
(Preferred: Tom)

Experience

- A-Levels in Maths, Chemistry, Physics, and Further Maths at *The Manchester College*.
- Master's degree in Mathematics and Physics from *University of Manchester*.
  - Specialised in calculus.
- Secondary school maths teacher in *West Yorkshire*.

As you can see, I am not Amalia ...

Thomas James MMathPhys
(Preferred: Tom)

Experience

- A-Levels in Maths, Chemistry, Physics, and Further Maths at *The Manchester College*.
- Master's degree in Mathematics and Physics from *University of Manchester*.
  - Specialised in calculus.
- Secondary school maths teacher in *West Yorkshire*.

tjjames.co.uk/reach-2023

Introduction to Modelling

Mathematics Session 4

T J James

Reach 2023

August 3, 2023

Schedule

1. Numerical Methods
   - Integration - Trapezium Rule
   - Differentiation - Finite Differences

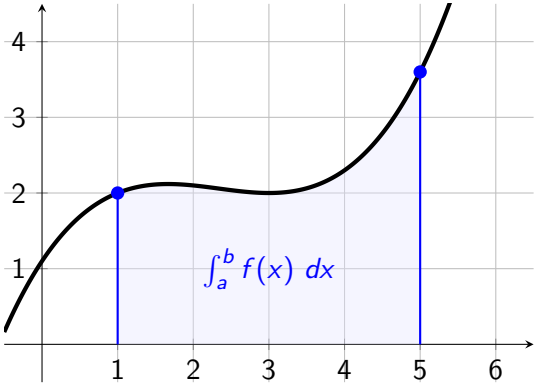
2. Computer Modelling
   - Simulating an ODE
   - ODE Solving Packages

Numerical Methods

Integration - Trapezium Rule

Integration

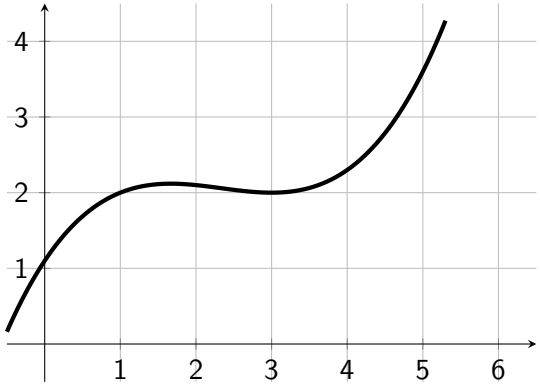
Starting with integration (since it was discovered first) ...
Fundamentally, integration finds the area between a line and the x-axis, between two points.



Approximating Integration

We can make a *first* approximation of the area by using a trapezium, where

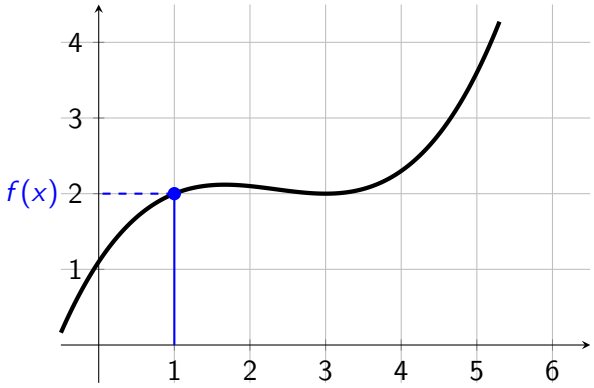
$$Area = \frac{1}{2}(a + b)\, h.$$



Approximating Integration

We can make a *first* approximation of the area by using a trapezium, where

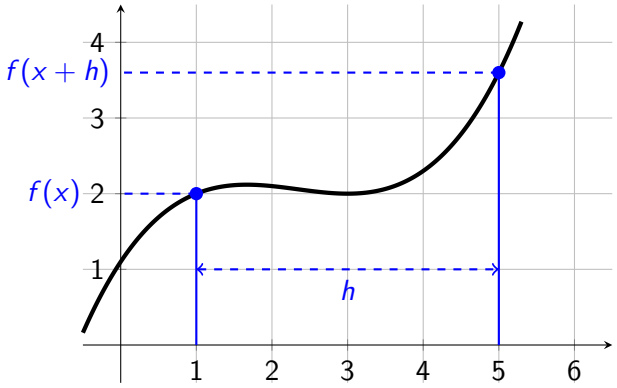
$$Area = \frac{1}{2}(a+b)\,h.$$



Approximating Integration

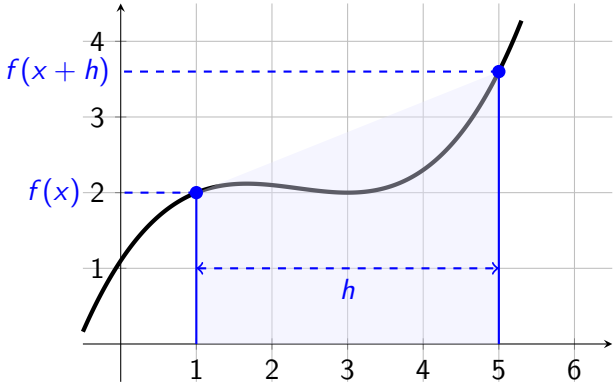
We can make a *first* approximation of the area by using a trapezium, where

$$Area = \frac{1}{2}(a+b)\,h.$$



Approximating Integration

We can make a *first* approximation of the area by using a trapezium

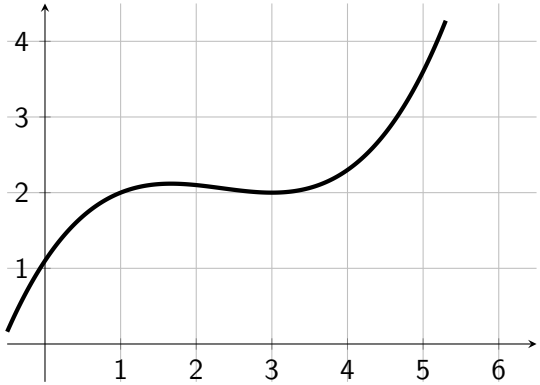


First Approximation

$$\int_a^b f(x)\ dx \approx \frac{1}{2}\left(f(x)+f(x+h)\right)h.$$

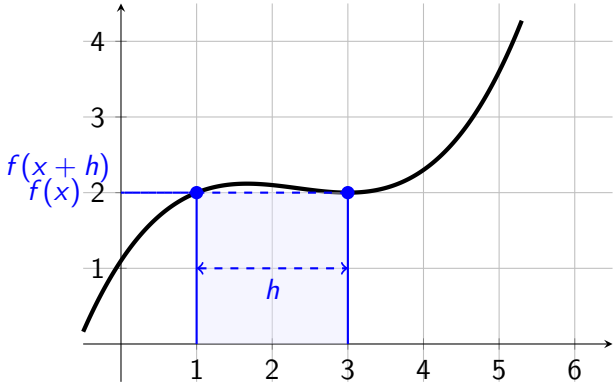
Approximating Integration

We can make a *second* approximation of the area by using two trapezia.



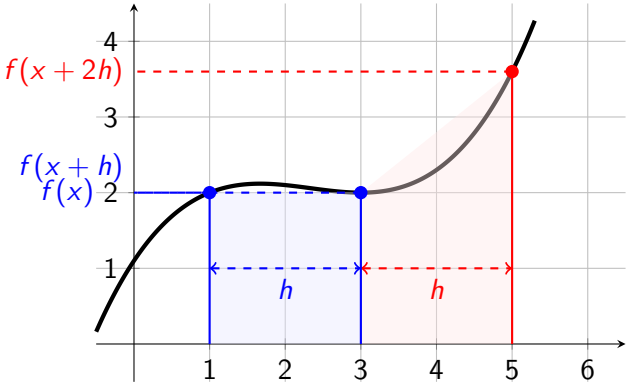
Approximating Integration

We can make a *second* approximation of the area by using two trapezia.



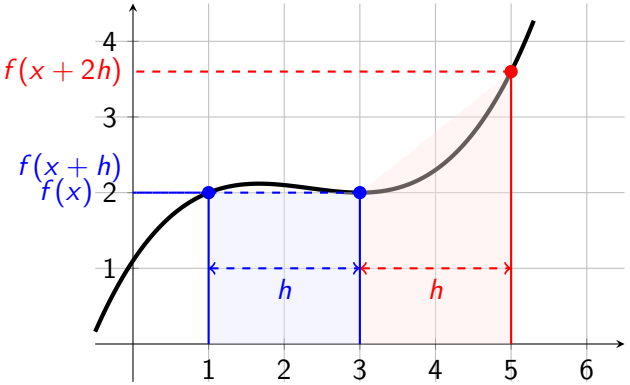
Approximating Integration

We can make a *second* approximation of the area by using two trapezia.



Approximating Integration

We can make a *second* approximation of the area by using two trapezia.

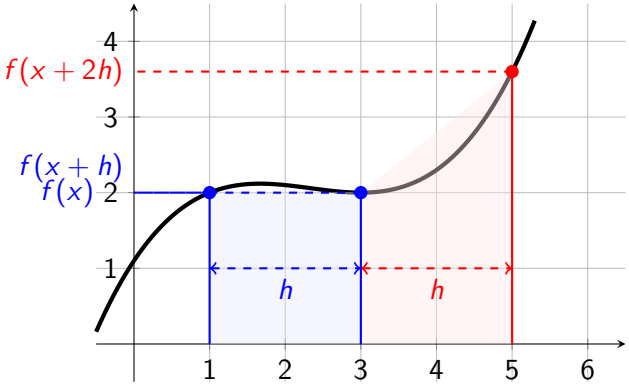


Second Approximation

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\left(f(x)+f(x+h)\right)h + \frac{1}{2}\left(f(x+h)+f(x+2h)\right)h$$

Approximating Integration

We can make a *second* approximation of the area by using two trapezia.

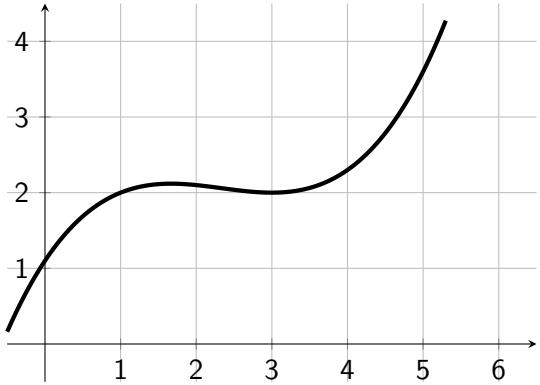


Second Approximation

$$\int_a^b f(x)\,dx \approx \left[\frac{1}{2}\left(f(x)+f(x+2h)\right)+f(x+h)\right]h$$

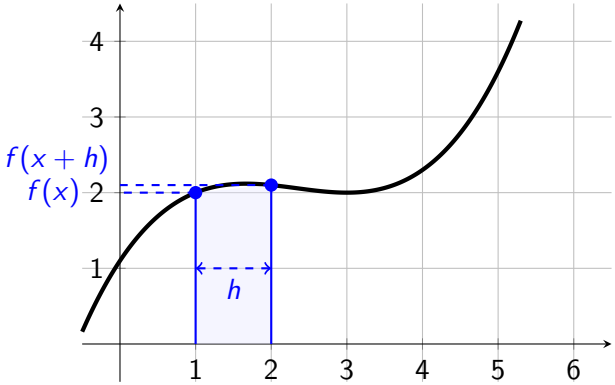
Approximating Integration

We can make *further* approximations of the area by using more trapezia.



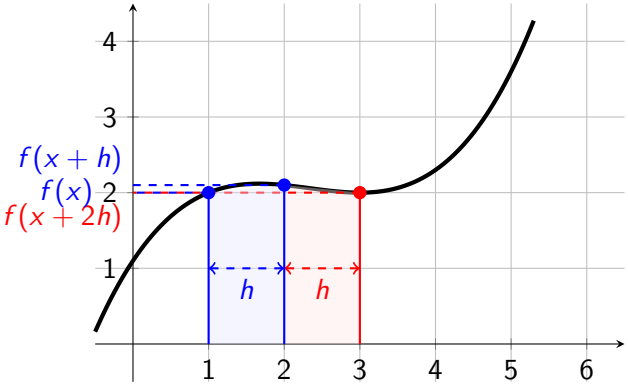
Approximating Integration

We can make *further* approximations of the area by using more trapezia.



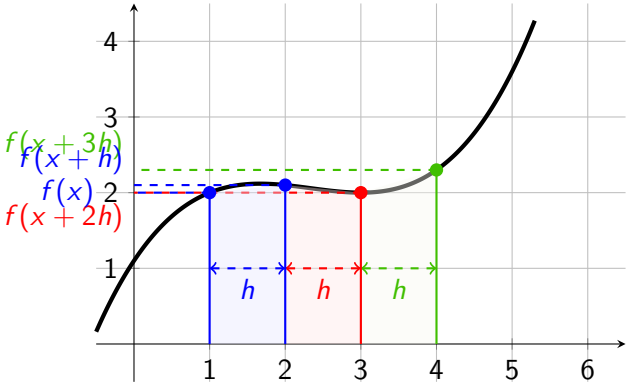
Approximating Integration

We can make *further* approximations of the area by using more trapezia.



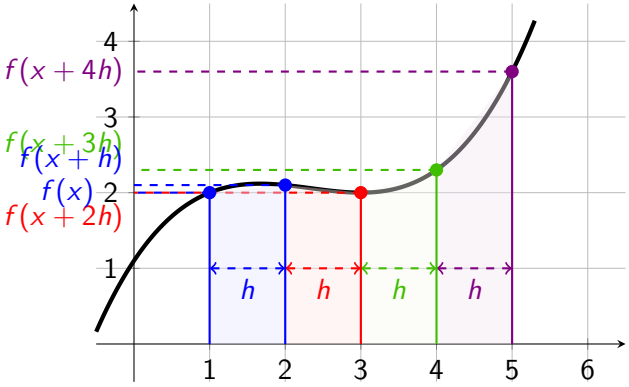
Approximating Integration

We can make *further* approximations of the area by using more trapezia.



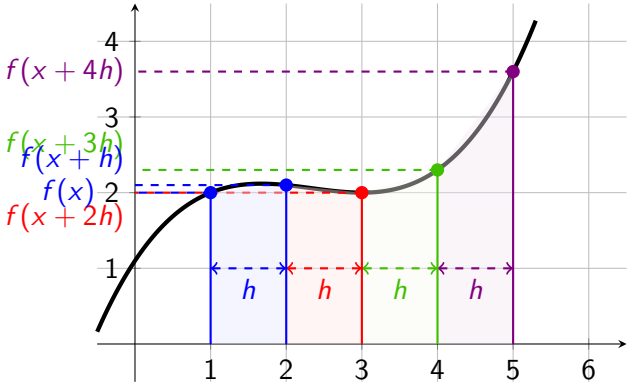
Approximating Integration

We can make *further* approximations of the area by using more trapezia.



Approximating Integration

We can make *further* approximations of the area by using more trapezia.

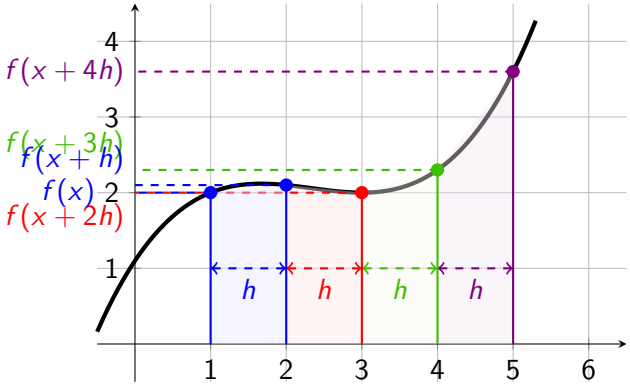


Further Approximation

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\left(f(x)+f(x+h)\right)h + \frac{1}{2}\left(f(x+h)+f(x+2h)\right)h + \ldots$$

Approximating Integration

We can make *further* approximations of the area by using more trapezia.



Further Approximation

$$\int_a^b f(x)\,dx \approx \left[\frac{1}{2}\left(f(x)+f(x+2h)\right)+f(x+h)+f(x+2h)+f(x+3h)\right]h$$

Trapezium Rule

In general, we can split out integral into N trapezia.
So if we call $x = x_0$, $x + h = x_1$, $x + 2h = x_2$, ..., $x + Nh = x_N$, then we can rewrite the trapezium rule to,

$$\int_a^b f(x)\, dx \approx \frac{1}{2} h \left[f(x_0) + 2 \left[f(x_1) + f(x_2) + \cdots + f(x_{N-1}) \right] + f(x_N) \right].$$

Trapezium Rule

In general, we can split out integral into N trapezia.
So if we call $x = x_0$, $x + h = x_1$, $x + 2h = x_2$, ..., $x + Nh = x_N$, then we can rewrite the trapezium rule to,

$$\int_a^b f(x)\ dx \approx \frac{1}{2}h\left[f(x_0) + 2\left[f(x_1) + f(x_2) + \cdots + f(x_{N-1})\right] + f(x_N)\right].$$

Then, if we assume the trapezia are all the same heights (h stays the same),

Theorem

$$\int_a^b f(x)\ dx \approx \frac{(a-b)}{2N}\left[f(x_0) + 2\left[f(x_1) + f(x_2) + \cdots + f(x_{N-1})\right] + f(x_N)\right].$$

Using the Trapezium Rule

We're now going to be using $y = x^2$ as an example.

Using the Trapezium Rule

We're now going to be using $y = x^2$ as an example.

Task 1

Integrate $y = x^2$ and find the area beneath the line between any two points. (Easy: 0 - 4. Challenging: -3 - 5.)

Using the Trapezium Rule

We're now going to be using $y = x^2$ as an example.

Task 1

Integrate $y = x^2$ and find the area beneath the line between any two points. (Easy: 0 - 4. Challenging: -3 - 5.)

Task 2

Work out approximations for this area using the trapezium rule (Easy: N $= 4$. Challenging: N $= 8$)

Break

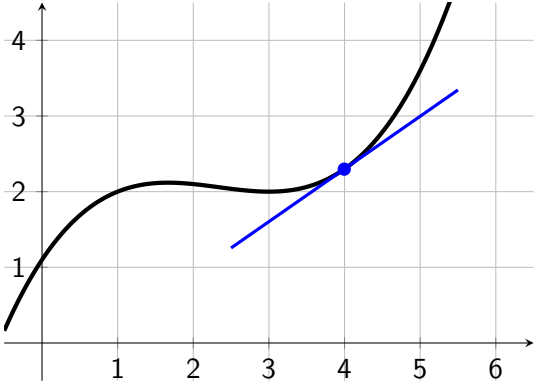
5 minute break

Numerical Methods

Differentiation - Finite Differences

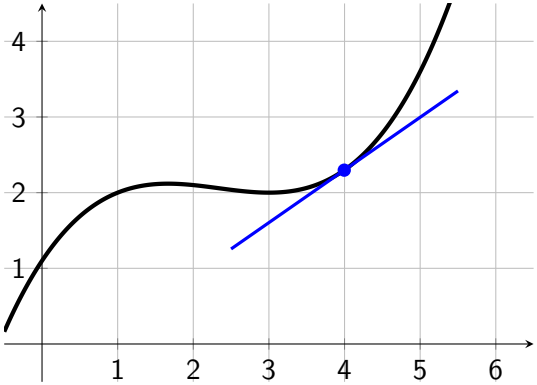
Differentiation

Fundamentally, differentiation finds the gradient of a line at a point.



Differentiation

Fundamentally, differentiation finds the gradient of a line at a point.



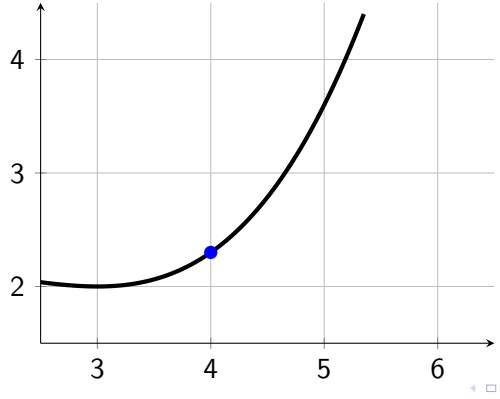
Theorem

$$f'(x) = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$

Approximating Differentiation

Theorem

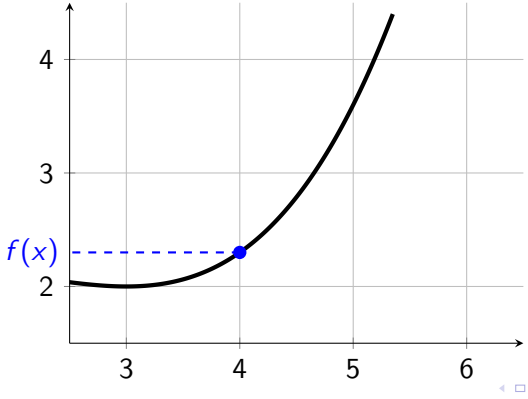
$$f'(x) = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$



Approximating Differentiation

Theorem

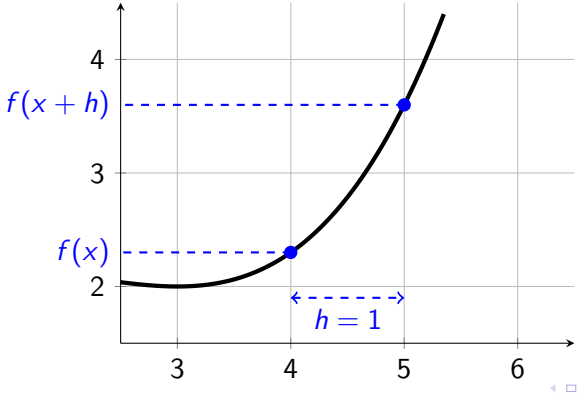
$$f'(x) = \left(\frac{f(x+h) - f(x)}{h}\right)$$



Approximating Differentiation

Theorem

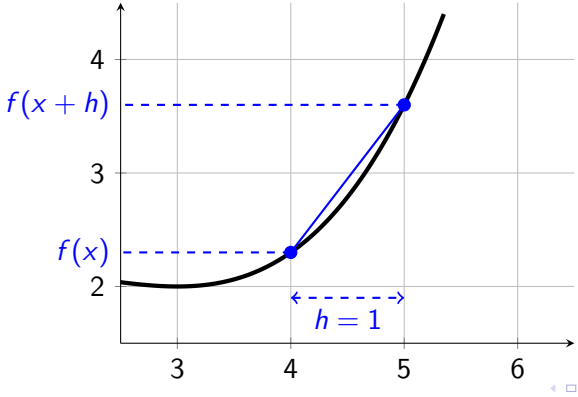
$$f'(x) = \left(\frac{f(x+h) - f(x)}{h}\right)$$



Approximating Differentiation

Theorem

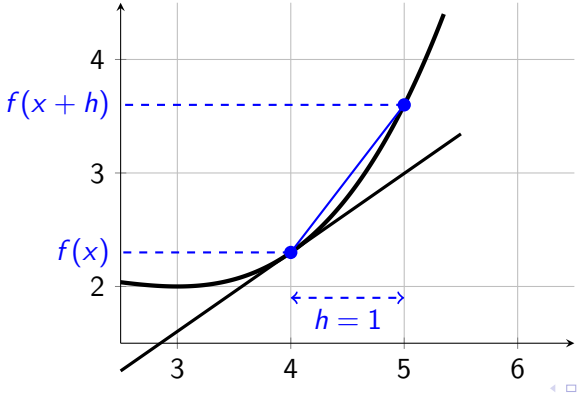
$$f'(x) = \left(\frac{f(x+h) - f(x)}{h}\right)$$



Approximating Differentiation

Theorem

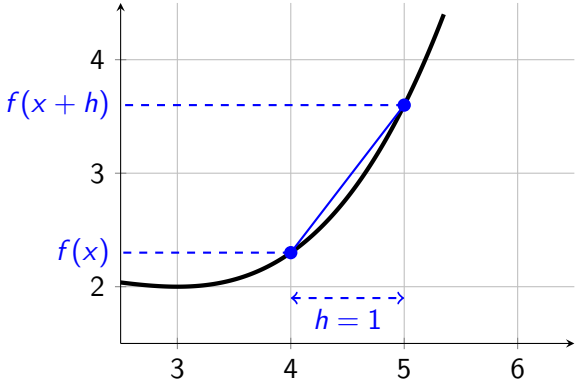
$$f'(x) = \left(\frac{f(x+h) - f(x)}{h}\right)$$



Increasing Accuracy

Theorem

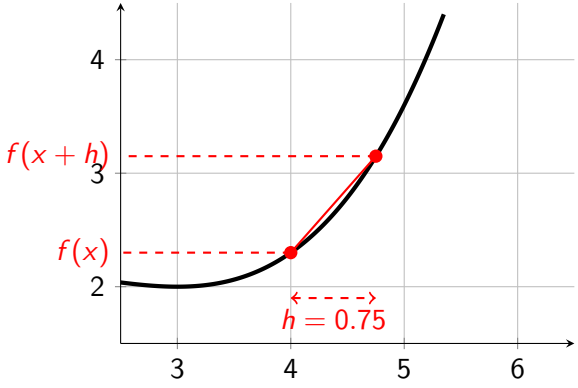
$$f'(x) = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$



Increasing Accuracy

Theorem

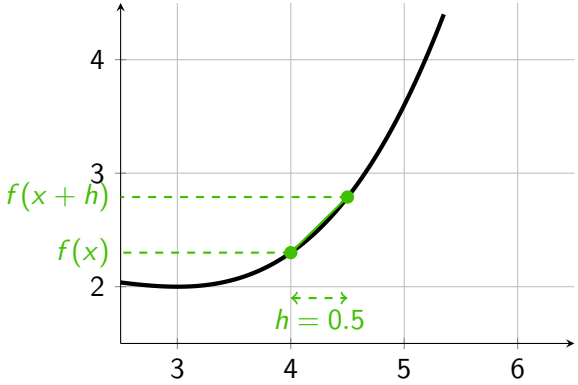
$$f'(x) = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$



Increasing Accuracy

Theorem

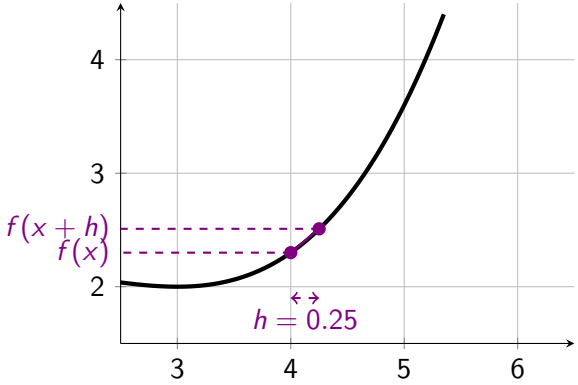
$$f'(x) = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$



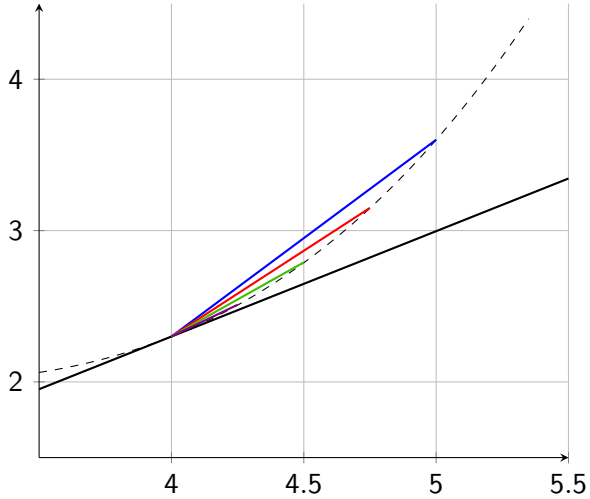
Increasing Accuracy

Theorem

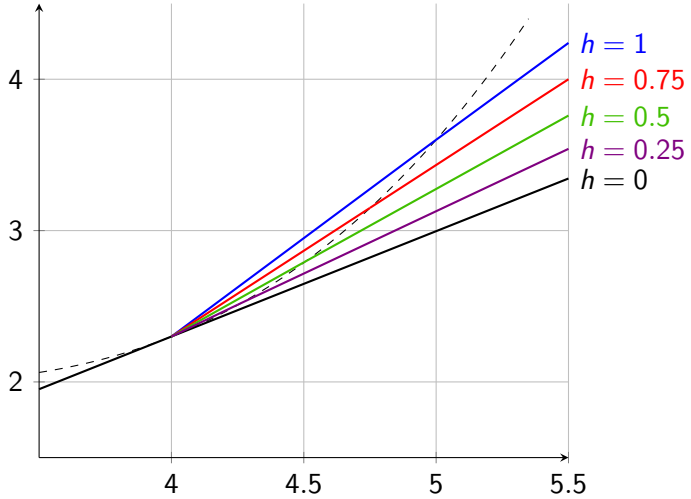
$$f'(x) = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$



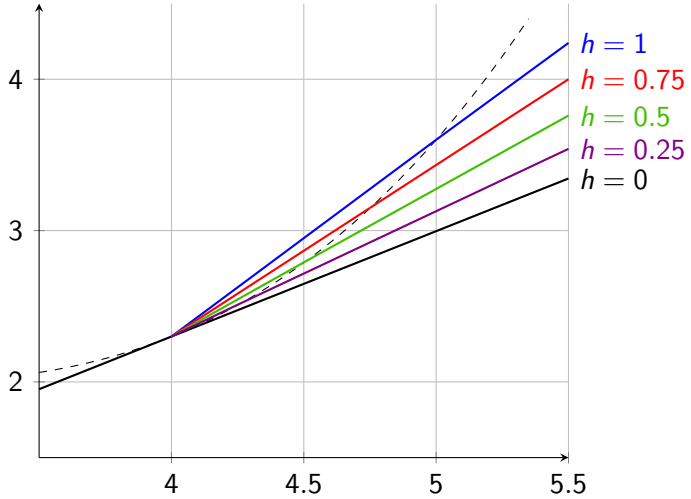
Increasing Accuracy



Increasing Accuracy



Increasing Accuracy



As h gets smaller ($h \to 0$), our approximations gets better.

Using the Method of Finite Differences

We're now going to be using $y = x^2$ as an example again.

Using the Method of Finite Differences

We're now going to be using $y = x^2$ as an example again.

Task 1

Differentiate $y = x^2$ and find the gradient of the line at any point.
(Easy: $x = 1$. Challenging: $x = -2$.)

Using the Method of Finite Differences

We're now going to be using $y = x^2$ as an example again.

Task 1

Differentiate $y = x^2$ and find the gradient of the line at any point.
(Easy: $x = 1$. Challenging: $x = -2$.)

Task 2

Work out the approximations for $\frac{dy}{dx}$ at this point with different values for h.

Break

5 minute break

During the break, get your laptops out and ready.
(We shall need them at the end of the next section)

Solving an ODE

Consider the following ODE,

$$\frac{dy}{dx} = 3y + 2, \quad y = 0, when x = 0.$$

Solving an ODE

Consider the following ODE,

$$\frac{dy}{dx} = 3y + 2, \quad y = 0, when x = 0.$$

Example

Find the solution for this ODE.

Full Solution

Solving a First-Order ODE

Consider the following ODE,

$$\frac{du}{dx} = 3u + 2.$$

Solving a First-Order ODE

Consider the following ODE,

$$u'(x) = 3u(x) + 2.$$

Solving a First-Order ODE

Consider the following ODE,

$$u'(x) = 3u(x) + 2.$$

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Solving a First-Order ODE

Consider the following ODE,

$$u'(x) = 3u(x) + 2.$$

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Substitute the approximation in,

$$\frac{u(x+h) - u(x)}{h} = 3u(x) + 2,$$

Solving a First-Order ODE

Consider the following ODE,

$$u'(x) = 3u(x) + 2.$$

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Substitute the approximation in,

$$\frac{u(x+h) - u(x)}{h} = 3u(x) + 2,$$

which then rearranges to,

$$u(x+h) = u(x) + h\left(3u(x) + 2\right).$$

Solving a First-Order ODE

Consider the following ODE,

$$u'(x) = 3u(x) + 2.$$

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Substitute the approximation in,

$$\frac{u(x+h) - u(x)}{h} = 3u(x) + 2,$$

which then rearranges to,

$$u(x+h) = u(x) + h\left(3u(x) + 2\right).$$

This is the **finite-difference equation**. Solving it will give an approximate solution of the ODE.

Solving an ODE

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Task

Find the finite-difference equations for each of these first order ODEs.

1. $$u'(x) = 4u(x) + 2$$
2. $$7xu'(x) + \frac{1}{x} = 12u(x)$$

Solving an ODE

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Task

Find the finite-difference equations for each of these first order ODEs.

1. $$u'(x) = 4u(x) + 2$$
2. $$7xu'(x) + \frac{1}{x} = 12u(x)$$

Extension

Using the above theorem, find the approximation for $u''(x) = \frac{d^2u}{dx^2}$.

Solving an ODE

Theorem

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Task

Find the finite-difference equations for each of these first order ODEs.

1. $$u'(x) = 4u(x) + 2$$
2. $$7xu'(x) + \frac{1}{x} = 12u(x)$$

Extension

Using the above theorem, find the approximation for $u''(x) = \frac{d^2u}{dx^2}$.

Approximating Second Derivatives

To approximate the second derivative, we can use our original approximation for a first derivative.

$$u'(x) \approx \frac{u(x+h) - u(x)}{h}$$

Approximating Second Derivatives

To approximate the second derivative, we can use our original approximation for a first derivative.

$$u''(x) \approx \frac{u'(x+h) - u'(x)}{h}$$

Approximating Second Derivatives

To approximate the second derivative, we can use our original approximation for a first derivative.

$$u''(x) \approx \frac{u'(x+h) - u'(x)}{h}$$

We can then replace the first derivatives with their approximations

$$u''(x) \approx \frac{\frac{u(x+2h)-u(x+h)}{h} - \frac{u(x+h)-u(x)}{h}}{h}$$

Approximating Second Derivatives

To approximate the second derivative, we can use our original approximation for a first derivative.

$$u''(x) \approx \frac{u'(x+h) - u'(x)}{h}$$

We can then replace the first derivatives with their approximations

$$u''(x) \approx \frac{\frac{u(x+2h)-u(x+h)}{h} - \frac{u(x+h)-u(x)}{h}}{h} = \frac{u(x+2h) - 2u(x+h) + u(x)}{h^2}$$

Approximating Second Derivatives

To approximate the second derivative, we can use our original approximation for a first derivative.

$$u''(x) \approx \frac{u'(x+h) - u'(x)}{h}$$

We can then replace the first derivatives with their approximations

$$u''(x) \approx \frac{\frac{u(x+2h)-u(x+h)}{h} - \frac{u(x+h)-u(x)}{h}}{h} = \frac{u(x+2h) - 2u(x+h) + u(x)}{h^2}$$

This is an approximation for $u''(x)$, but we tend to write based around x rather than $x+h$, so we use,

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

Solving a Second-Order ODE

Consider the following ODE,

$$\frac{d^2y}{dx^2} + 16y = 0.$$

Solving a Second-Order ODE

Consider the following ODE,

$$u''(x) + 16u(x) = 0.$$

Solving a Second-Order ODE

Consider the following ODE,

$$u''(x) + 16u(x) = 0.$$

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

Solving a Second-Order ODE

Consider the following ODE,

$$u''(x) + 16u(x) = 0.$$

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

Substitute the approximation in,

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + 16u(x) = 0,$$

Solving a Second-Order ODE

Consider the following ODE,

$$u''(x) + 16u(x) = 0.$$

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

Substitute the approximation in,

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + 16u(x) = 0,$$

which then rearranges to,

$$u(x+h) = 2u(x) - u(x-h) + 16h^2 u(x),$$

Solving a Second-Order ODE

Consider the following ODE,

$$u''(x) + 16u(x) = 0.$$

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

Substitute the approximation in,

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + 16u(x) = 0,$$

which then rearranges to,

$$\begin{aligned} u(x+h) &= 2u(x) - u(x-h) + 16h^2 u(x), \\ &= \left[2 + 16h^2\right] u(x) - u(x-h). \end{aligned}$$

Solving a Second-Order ODE

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

Task

Find the finite-difference equations for each of these first order ODEs.

1. $$u''(x) = 7u(x) + 2x - 3$$
2. $$u''(x) - 6u'(x) + 24u(x) = \sin x$$

Solving a Second-Order ODE

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

Task

Find the finite-difference equations for each of these first order ODEs.

1. $$u''(x) = 7u(x) + 2x - 3$$
2. $$u''(x) - 6u'(x) + 24u(x) = \sin x$$

Solving a Second-Order ODE

Theorem

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

Task

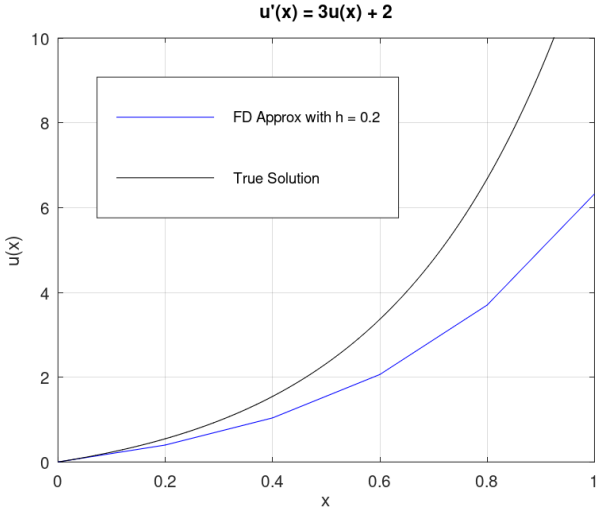
Find the finite-difference equations for each of these first order ODEs.

1. $$u''(x) = 7u(x) + 2x - 3$$
2. $$u''(x) - 6u'(x) + 24u(x) = \sin x$$

Choosing Accuracy

$$u'(x) = 3u(x) + 2, \quad u(0) = 0.$$

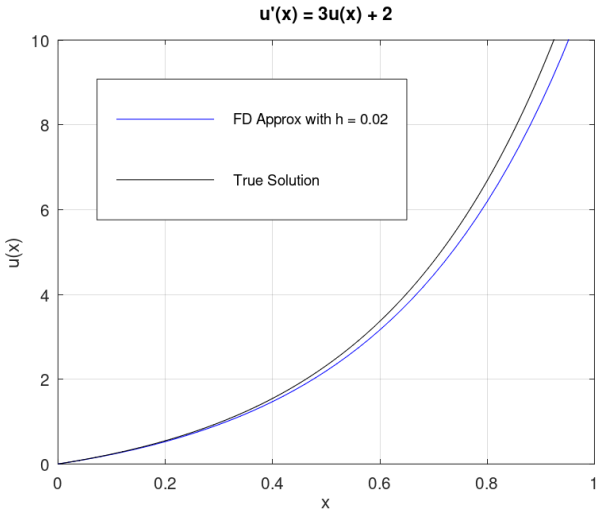
$$u(x + h) = u(x) + h\,(3u(x) + 2))$$



Choosing Accuracy

$$u'(x) = 3u(x) + 2, \quad u(0) = 0.$$

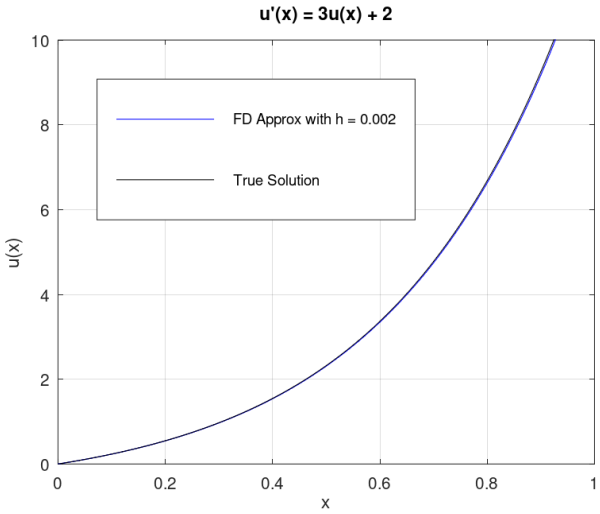
$$u(x + h) = u(x) + h\,(3u(x) + 2))$$



Choosing Accuracy

$$u'(x) = 3u(x) + 2, \quad u(0) = 0.$$

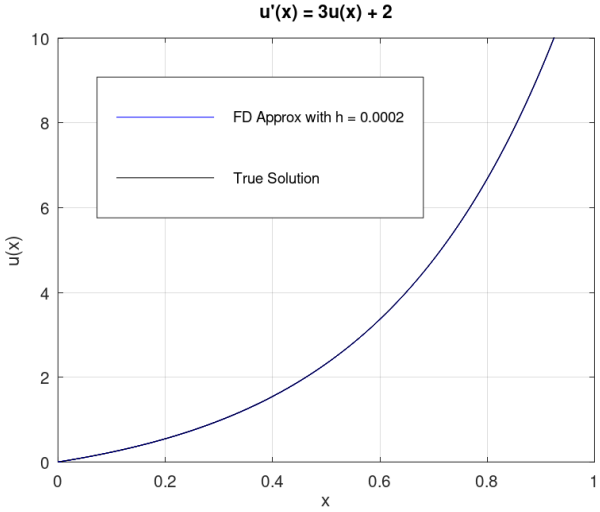
$$u(x + h) = u(x) + h\,(3u(x) + 2))$$



Choosing Accuracy

$$u'(x) = 3u(x) + 2, \quad u(0) = 0.$$

$$u(x + h) = u(x) + h\,(3u(x) + 2))$$



Choosing Accuracy

	Large h	**Small** h
Pros	Quick	High accuracy
Cons	Low accuracy	Slow and resource intensive

When running simulations, it is often a balancing act between accuracy and convenience.

Before the Break

We are going to download Octave, a programming language very similar to MATLAB (but free!)

Navigate to ...

octave.org/download

Before the Break

We are going to download Octave, a programming language very similar to MATLAB (but free!)

Navigate to ...

octave.org/download

Choose the *Installer* option:

Break

15 minute break

Check on Octave's progress occasionally, and start installing it once it has downloaded.

Computer Modelling

Simulating an ODE

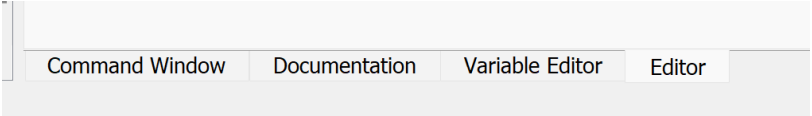
Housekeeping

Octave is a programming language, which is very similar to MATLAB - it is **1-indexed**, not 0-indexed.

Housekeeping

Octave is a programming language, which is very similar to MATLAB - it is **1-indexed**, not 0-indexed.

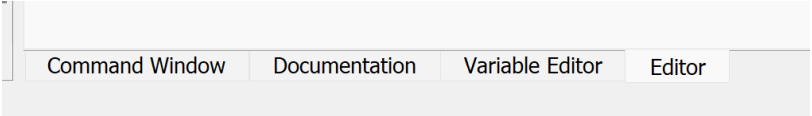
1. We shall be working in the `Editor` window.



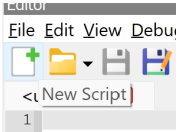
Housekeeping

Octave is a programming language, which is very similar to MATLAB - it is **1-indexed**, not 0-indexed.

1. We shall be working in the `Editor` window.



2. Create a `New Script`, which is where we shall be doing our coding.



   - Save your script now, somewhere you can find it.
   - Your script's name *cannot* have any spaces in it, so must be one word (you can use underscores _).

Building a Script

Together, we're going to build the simulation for the solution of:

$$u'(x) = 3u(x) + 2, \quad u(0) = 0.$$

$$u(x + h) = u(x) + h\,(3u(x) + 2))$$

Building a Script

Together, we're going to build the simulation for the solution of:

$$u'(x) = 3u(x) + 2, \quad u(0) = 0.$$

$$u(x + h) = u(x) + h\,(3u(x) + 2))$$

The first thing to include in any Octave script are these clear lines:

```
clf; clear; clearvars;
```

Defining Parameters

Next, we need to list all our parameters - in this case, our value for h.

- We *could* just put them down as we need them, but listing them at the start makes them easier to find and change.
- IMO - Anything which you set a specific value for should go at the top, and everything else should be worked out from those.

```
h = 0.2;
```

Defining Parameters

Next, we need to list all our parameters - in this case, our value for h.

- We *could* just put them down as we need them, but listing them at the start makes them easier to find and change.
- IMO - Anything which you set a specific value for should go at the top, and everything else should be worked out from those.

```
h = 0.2;
```

And whilst we're defining values, let's define our x values and our inital value for $u(x)$.

```
x_values = 0:h:1;
u_values(1) = 0;
```

Defining Parameters

And whilst we're defining values, let's define our x values and our inital value for $u(x)$.

```
x_values = 0:h:1;
u_values(1) = 0;
```

- For `x_values`, this creates an *array* (a table) of values, between the values 0 and 1, with a gap of `h` between them.
  - So with `h = 0.2`, `x_values` would be `(0, 0.2, ..., 0.8, 1)`.
- When we have a 1D array (which is all we're going to be dealing with), we can find values in that array by putting the name of the array, followed by `(x)` where `x` is the value we're looking for.
  - So `u_values(1) = 0` is setting the first u value to 0.

Calculating Approximation

Now, we're going to use the formula we worked out previously,

$$u(x+h) = u(x) + h\left(3u(x) + 2\right).$$

Calculating Approximation

Now, we're going to use the formula we worked out previously,

$$u(x+h) = u(x) + h\left(3u(x) + 2\right).$$

Using a `for` loop, we'll convert this formula into code:

```
for n = 1:(length(x_values) - 1)
  u_values(n + 1) = u_values(n) + h * (3 * u_values(n) + 2);
end
```

Calculating Approximation

Now, we're going to use the formula we worked out previously,

$$u(x+h) = u(x) + h\left(3u(x) + 2\right).$$

Using a `for` loop, we'll convert this formula into code:

```
for n = 1:(length(x_values) - 1)
  u_values(n + 1) = u_values(n) + h * (3 * u_values(n) + 2);
end
```

- A `for` loop is a piece of code which will repeat itself for a certain number of times.
  - `n` is the counter for the number of times the `for` loop has run. it is 1 for the first run, then 2 for the second, etc.
  - Once again, we're using an array where `n` will count up from the starting number 1 to the end value (which in this case is the number of `x_values` we have, minus one.
- We already have `u_values(1)` defines (we set it 0).
  - So for every pass through the `for` loop, it takes the current value for `u_values` and uses that to work out the next value.

Plotting Approximation

With all the maths done, we can now see what our approximation looks like.

```
plot(x_values, u_values)
```

Plotting Approximation

With all the maths done, we can now see what our approximation looks like.

`plot(x_values, u_values)`

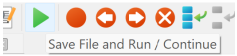
We can then run our script.

Plotting Approximation

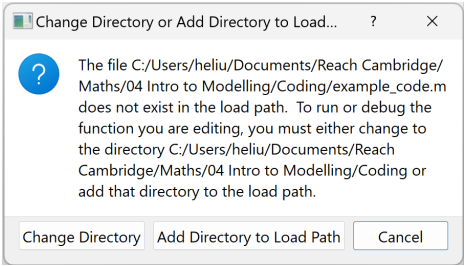
With all the maths done, we can now see what our approximation looks like.

```
plot(x_values, u_values)
```

We can then run our script.



But before our script can run, we need to change our directory so Octave actually knows where our script is.

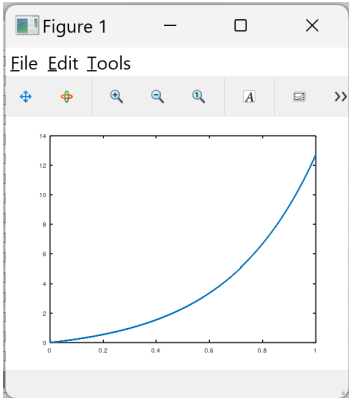


Plotting Approximation

With all the maths done, we can now see what our approximation looks like.

```
plot(x_values, u_values)
```

You should now have a plot which looks a bit like this:



Comparing with True Solution

Since we've been able to calculate the true solution to our original solution, we can plot that as well.

$$y = \frac{2e^{3x}}{3} - \frac{2}{3}.$$

```
true_values = 2 * exp(3 * x_values) / 3 - 2 / 3;
```

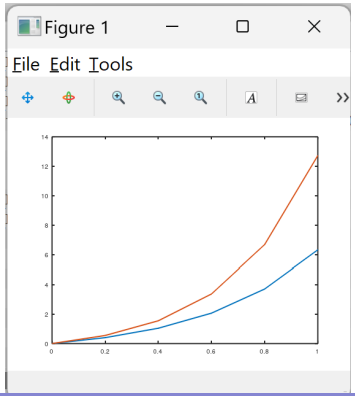
- Because `x_values` is already an array, if we do any maths to it without specifying an entry, each entry will be worked on, giving another array as the output.

Comparing with True Solution

```
true_values = 2 * exp(3 * x_values) / 3 - 2 / 3;
```

Now that we have the true values, we can plot them alongside our approximation.

```
hold on;
plot(x_values, true_values)
```



Final Code

```
clf; clear; clearvars;

h = 0.2;

x_values = 0:h:1;
u_values(1) = 0;

for n = 1:(length(x_values) - 1)
  u_values(n + 1) = u_values(n) + h * (3 * u_values(n) + 2);
end

plot(x_values, u_values)
```

Refining the Approximation

To start looking at how accurate we can make our approximation, all we need to do is vary the `h` value (which is why we put it at the top, so it's easy to find).

Try some different values of `h` ...

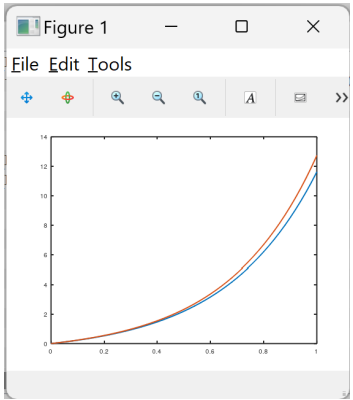
`h = 0.02` `h = 0.002` `h = 0.0002`

Refining the Approximation

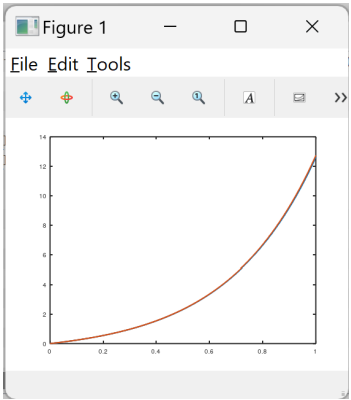
To start looking at how accurate we can make our approximation, all we need to do is vary the `h` value (which is why we put it at the top, so it's easy to find).

Try some different values of `h` ...

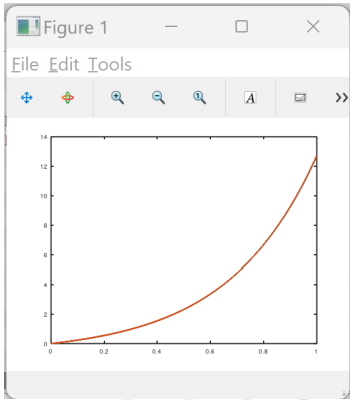
`h = 0.02`



`h = 0.002`



`h = 0.0002`



Make It *Pretty*

Coders spend obscene amounts of money on RGB lighting and pretty mechanical keyboards, hence we like to make out code outputs pretty too.

You could use some of the following options:

```
grid on;
title("u'(x) = 3u(x) + 2");
xlabel("x");
ylabel("u(x)");
axis([0 1 0 10]);

legend(
"Finite-Difference Approximations",
"True Solution",
"location", "northwest"
)
```

Approximating an ODE

Task

Using the previous first-order ODEs which we found approximations for, create a script which will run differing approximations of the ODEs. Choose appropriate initial conditions, and an appropriate range of x-values over which to approximate.

$$u'(x) = 4u(x) + 2$$

$$7xu'(x) + \frac{1}{x} = 12u(x)$$

$$u(x+h) = u(x) + h\left[4u(x) + 2\right]$$

$$u(x+h) = u(x) + \frac{h}{7x}\left[12u(x) - \frac{1}{x}\right]$$

Approximating an ODE

Task

Using the previous first-order ODEs which we found approximations for, create a script which will run differing approximations of the ODEs. Choose appropriate initial conditions, and an appropriate range of x-values over which to approximate.

$$u'(x) = 4u(x) + 2$$

$$7xu'(x) + \frac{1}{x} = 12u(x)$$

$$u(x+h) = u(x) + h\left[4u(x) + 2\right]$$

$$u(x+h) = u(x) + \frac{h}{7x}\left[12u(x) - \frac{1}{x}\right]$$

Extension

1. Find a way to represent each of the different `h` values onto a single graph.
2. Find a way to measure how good the quality of your approximation is
    1. Plot how much your approximation deviates from the true values.

Break

5 minute break

Computer Modelling

ODE Solving Packages

ODE45

We've been solving equations where it isn't too hard to form the finite-difference equations. However, some equations are more complex and less straight-forward to solve.
As a result, there are a few different packages which one can use to find an approximation of an ODE - one in particular is `ode45`.

ODE45

We've been solving equations where it isn't too hard to form the finite-difference equations. However, some equations are more complex and less straight-forward to solve.
As a result, there are a few different packages which one can use to find an approximation of an ODE - one in particular is ode45.

```
: [t, y] = ode45 (fun, trange, init)
: [t, y] = ode45 (fun, trange, init, ode_opt)
: [t, y, te, ye, ie] = ode45 (...)
: solution = ode45 (...)
: ode45 (...)
```

This is an excerpt from the ode45 documentation - it can be daunting to read docs, but they are manageable if you're meticulous and careful.

Understanding ODE45

```
[t, y] = ode45 (fun, trange, init)
```

- ODE45 uses time `t` and `y` in place of x and $u(x)$ which we've been using
  - Because standardised notation is *entirely* overrated.
- ODE45 will produce two arrays, an array of `t` values and an array of `y` values

Understanding ODE45

```
[t, y] = ode45 (fun, trange, init)
```

- ODE45 uses time `t` and `y` in place of x and $u(x)$ which we've been using
  - Because standardised notation is *entirely* overrated.
- ODE45 will produce two arrays, an array of `t` values and an array of `y` values
- ODE45 works by taking a function `fun` which is,

$$\frac{dy}{dt} = f(t, y),$$

and solving it between two times `trange`.

Understanding ODE45

```
[t, y] = ode45 (fun, trange, init)
```

- ODE45 uses time `t` and `y` in place of x and $u(x)$ which we've been using
  - Because standardised notation is *entirely* overrated.
- ODE45 will produce two arrays, an array of `t` values and an array of `y` values
- ODE45 works by taking a function `fun` which is,

$$\frac{dy}{dt} = f(t, y),$$

  and solving it between two times `trange`.
- You specify the initial conditions `init`, which ODE45 uses to find the approximation.

Using ODE45

To get our heads around ODE45, we're going to solve a simple ODE.

$$\frac{dy}{dt} = y' = 2t, \quad y(t = 0) = 0$$

We're going to approximate the solution between $t = 0$ and $t = 5$.

Using ODE45

To get our heads around ODE45, we're going to solve a simple ODE.

$$\frac{dy}{dt} = y' = 2t, \quad y(t = 0) = 0$$

We're going to approximate the solution between $t = 0$ and $t = 5$.

Quick Task

Find the true solution to this ODE.

Using ODE45

To get our heads around ODE45, we're going to solve a simple ODE.

$$\frac{dy}{dt} = y' = 2t, \quad y(t = 0) = 0$$

We're going to approximate the solution between $t = 0$ and $t = 5$.

Quick Task

Find the true solution to this ODE.

Using ODE45

To get our heads around ODE45, we're going to solve a simple ODE.

$$\frac{dy}{dt} = y' = 2t, \quad y(t = 0) = 0$$

We're going to approximate the solution between $t = 0$ and $t = 5$.

Quick Task

Find the true solution to this ODE.

Before doing anything, we should define our variables.

```
time = [0, 5];
y0 = 0;
```

Defining the Function

$$\frac{dy}{dt} = y' = 2t, \quad y(t = 0) = 0$$

We *can* define the function for y' within ODE45 itself, but that's not the best idea - especially if it's a complicated function.

Defining the Function

$$\frac{dy}{dt} = y' = 2t, \quad y(t=0) = 0$$

We *can* define the function for y' within ODE45 itself, but that's not the best idea - especially if it's a complicated function.

We define the function for y' like so,

```
dydt = @(t,y) 2*t;
```

- The function has to be defined in terms of both `t` and `y`, even if you only use one of them.

Defining the Function

$$\frac{dy}{dt} = y' = 2t, \quad y(t=0) = 0$$

We *can* define the function for y' within ODE45 itself, but that's not the best idea - especially if it's a complicated function.

We define the function for y' like so,

```
dydt = @(t,y) 2*t;
```

- The function has to be defined in terms of both `t` and `y`, even if you only use one of them.

Once the function is defined, we can then use it within ODE45.

```
[t, y] = ode45(dydt, time, y0);
```

Plotting the Approximation

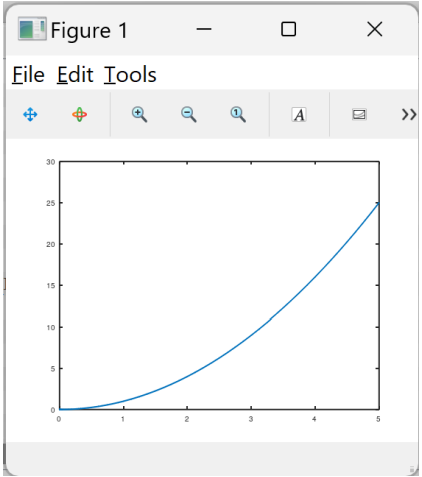
With ODE45 having done its job, it's now simply a matter of plotting its results

```
plot(t, y)
```

Plotting the Approximation

With ODE45 having done its job, it's now simply a matter of plotting its results

```
plot(t, y)
```



Using ODE45

So long as an ODE is first order and can be expressed as $y' = f(t, y)$, you can use ODE45 to solve it.

Task

Using ODE45, plot the approximations for these first-order ODEs. Choose appropriate initial conditions, and an appropriate range of x-values over which to approximate.

$$u'(x) = 4u(x) + 2 \qquad 7xu'(x) + \frac{1}{x} = 12u(x)$$

Extension

Plot your ODE45 results over the approximations you got using the finite-difference equations.

- This is especially applicable for $7xu'(x) = \cdots$ as we don't have a true solution for it. The ODE45 solution is as close to the "true" solution as we're likely to get.