Sheet 1

MATH19872: Introduction to Mathematica

1.1 Introduction

Mathematica is both a computer algebra package (a piece of software for manipulating mathematical expressions) and a full programming language (known as the *Wolfram* language) which includes a huge array of built-in commands for mathematical operations. This part of the course will introduce only the basics of how to use Mathematica, but it has huge capabilities beyond those shown here.

You can get a copy of Mathematica for use on your personal computer by requesting a license from `https://user.wolfram.com/portal/requestAK/1b39970ad002bd968f0fa5deff50f9ff768195`
(it may take a few days for your license to be approved).

It is expected that you will finish working through each sheet in your own time, in order to be able to complete the coursework.

1.2 Notebook Structure

Mathematica evaluates symbolic expressions within *notebooks*. To start a new notebook go to File › New › Notebook (or press Ctrl + N). One will open automatically when you start Mathematica.

There are two parts to Mathematica, the *Front End* and the *Kernel*. The *Front End* is where you write commands and see output, and is made up of individual cells. These start as being *Input* cells in the notebook, where you can type in expressions and press Shift ⇑ + ↩ (i.e. hold down the shift key and press the Return key) in order to evaluate them. The computer will then give its response in the output cell below the input cell, as shown:

```
In[1]:=  12+39

Out[1]=  51
```

(The blue **`In[]:=`** and **`Out[]=`** tags will be applied automatically when you evaluate the cell, the number tracks how many cells have been evaluated this session. Don't type in this blue text when trying the examples from this worksheet!)

When you evaluate a cell, the contents are sent to the *Kernel*, which is where all the calculations are made, this will be discussed in more detail later on.

You can change a cell to be a title (or text) by selecting Format › Style on the menu bar, or by using the keyboard shortcuts Alt + 1 (or 7). Use these to structure your notebook to make everything is clear.

Mathematica has comprehensive accompanying documentation which can be accessed by pressing the F1 key, and will open at the appropriate page if you have a command highlighted. Within one can find detailed tutorials, starting from the basics. Alternatively, you can type **??** (that is, the question mark) followed by a symbol to see the short version of the corresponding help files.

1.3 Basic Arithmetic

Mathematica can be used as a calculator, using +, −, *, / and ^ for addition, subtraction, multiplication, division and exponentiation respectively. For each line in the input cell, an *output* cell will be produced with the corresponding output. The five lines in the input below are separated by pressing ⏎, and the complete input is evaluated by pressing Shift ⇑ + ⏎.

```
In[2]:=  101+221
         45-32
         17*25
         8892/12
         2^3

Out[2]=  322
         13
         425
         741
         8
```

Unlike a calculator, Mathematica uses exact arithmetic: it will return exact answers when given exact numbers as input. One way to force it to give a numerical value use a decimal point somewhere in the calculation, e.g. 1.0 instead of 1:

```
In[3]:=  1/3
```

Out[3]= $\frac{1}{3}$

```
In[4]:=  1.0/3

Out[4]=  0.333333
```

1.3.1 Exercises

Use Mathematica to evaluate

1. $26 + 73$
2. $647 - 346$
3. 235×572
4. $26/6$
5. 13^7
6. $4 + 5 * 54 - 3$
7. 23^{423}

1.4 Variables

All the arithmetic operations above work with variables as well as numbers. For example,

```
In[5]:= x + 1 + 2 x - y^2 + 5
```

will collect like terms and return

```
Out[5]= 6 + 3 x - y^2
```

Mathematica will treat any set of consecutive characters that starts with a letter as a variable: the two-letter variable **xy** is different to **x*y** or **x y** (with a space in between the letters, where the multiplication sign is implied). Words and symbols that Mathematica already has a definition for will appear in black, whereas those it doesn't appear in blue – this can be very useful for finding typos or errors!

Greek letters may be used, these can be entered via the Basic Input Palette (Palettes › Basic Math Assistant) or by writing e.g. Esc a Esc for α. The input palette can also be used to create nice fractions or superscripts, allowing you to input equations that look the same as by hand.

Many mathematical constants are defined by default, such as **Pi** for $\pi \approx 3.14159$, **E** for the exponential base $e \approx 2.71828$ and **I** for the square root of minus one i. Because letters such as **E** and **I** are already reserved for these values, it is not usually a good idea to use single uppercase letters for your own variables.

1.5 Brackets

Mathematica uses three different types of brackets. These differ from the usual mathematical use of brackets, and are not interchangeable!

- Normal parentheses **()** are used to group terms together and determine operator precedence, in the usual mathematical way: e.g. $5 \times (4 + 3)$.
- Square brackets **[]** are used to provide an argument to a function, for instance **Sin[x]** is the Mathematica code for $\sin x$.
- Curly brackets **{}** are used to collect elements together in lists, such as vectors and matrices. Mathematica uses lists for a wide range other purposes too.

In summary, the Mathematica code **{a(Sin[x]+1),1}** would be used to represent the two-component vector $(a \sin x + a, 1)$.

1.6 Basic Functions

A function in Mathematica is a single word followed by any arguments within square brackets, separated by commas. Function names always start with a capital letter, e.g. **Sin[1]** stands for the $\sin$ of 1 radian, and those with multiple words are written in CamelCase, capitalizing the first letter in each word with no spaces.

As mentioned above, if given an exact number as input, Mathematica will return an exact output:

```
In[6]:= Sin[1]

Out[6]= Sin[1]
```

(the simplest way of expressing the $\sin$ of 1 radian, is just $\sin(1)$!) To force a numerical answer, use the function **N**:

```
In[7]:= N[Sin[1]]

Out[7]= 0.841471
```

This function (and all others that only take a single argument), can be applied by using the syntax **//N**, although you generally only do this for simple commands that you want to apply to the whole expression:

```
In[8]:=  Sin[1] //N

Out[8]=  0.841471
```

Note that Mathematica is case sensitive so **Sin** is different to **sin**. If Mathematica doesn't have any definitions stored for an expression, it will let us know by displaying the expression in blue, and return the input unchanged.

```
In[9]:=  sin[0]

Out[9]=  sin[0]
```

Correcting the error gives a numerical answer:

```
In[10]:=  Sin[0]

Out[10]=  0
```

Some common function names are **Sin**, **Cos**, **Tan**, **Log** (the natural logarithm), **Exp** (the exponential function), **Abs** (absolute value, the modulus function). The value to apply them to should always be in square brackets. For **Log**, a different base can be specified by writing **Log[b,x]** for $\log_b x$. The default base is the natural log (sometimes called $\ln$).

Exercises

Evaluate the following, using **N** to give answers as a decimal:

1. $\sin 1$
2. $\cos \pi$
3. $\tan \frac{\pi}{3}$
4. e^3
5. $\ln 7$
6. $\frac{26}{6}$
7. $|-4|$

1.7 Simplification

Mathematica has numerous functions that can be used to simplify and expand expressions. The first of these is **Simplify**, where Mathematica tries to use a wide range of methods to give the simplest result. For example, it will reduce $\cos^2 x + \sin^2 x$ to 1 using the trig identities. You can again use **//** to apply the function at the end of the expression, so **Simplify[expression]** is the same as **expression//Simplify**.

```
In[11]:=  Cos[x]^2 + Sin[x]^2 // Simplify

Out[11]=  1
```

For polynomials, **Expand** and **Factor** are opposing functions that do what they say, expand and factor the expression as much as possible.

```
In[12]:=  Expand[(1+x)^5]
```

Out[12]= $1 + 5\,x + 10\,x^2 + 10\,x^3 + 5\,x^4 + x^5$

```
In[13]:= Factor[x^4 + 12 x^3 y + 54 x^2 y^2 + 108 x y^3 + 81 y^4]
```

```
Out[13]= (x + 3 y)^4
```

1.8 Solution of Equations

The command **`Solve`** can be used to solve systems of equations. This example solves the equation $3x + 4 = 7$ for x. Note the use of the double equals sign $==$ to represent the equation, and that the value you are solving for turns cyan throughout the expression.

```
In[14]:= Solve[3x + 4 == 7,x]

Out[14]= {{x → 1}}
```

The output that comes from **`Solve`** is a list, which contains a list that contains a single *Replacement Rule*, denoted by the arrow **`->`**. This construct is used to say that the left hand side should be replaced by the right hand side, every time it appears.

`Solve` can also handle systems of simultaneous equations by the use of curly brackets to group the equations and variables into lists,

```
In[15]:= Solve[{x + y == 3,x - y == 1},{x,y}]

Out[15]= {{x → 2, y → 1}}
```

`Solve` can handle nonlinear equations, which may have multiple solutions. In this case, the different possible solutions will be in the nested list,

```
In[16]:= Solve[x^2 == 4, x]

Out[16]= {{x → -2}, {x → 2}}

In[17]:= Solve[{x^2 == 4, y+x ==3}, {x,y}]

Out[17]= {{x → -2, y → 5}, {x → 2, y → 1}}
```

(This is why the solution to the first **`Solve`** command gives a list containing a list with a single output: so that the output is consistent regardless of how many solutions to the equation there are.) If no solution to the equations exist then Mathematica will return an empty list:

```
In[18]:= Solve[{x + 2 y == 1, x + 2 y == 3}, {x, y}]

Out[18]= { }
```

Replacement rules can be applied to an expression by using the syntax **`/.`** (known as **`ReplaceAll`**), as

```
In[19]:= x + y /.{x->3}
         4 x^2 + x + 3 /.{x->2}

Out[19]= 3 + y
         21
```

If the replacement is a list of rules, then the output will also be a list:

```
In[20]:= x + y /. {{x->3}, {x->1}}
         x + y /. {{x->3, y->1}, {x->1, y->0}}

Out[20]= {3 + y, 1 + y}
         {4,1}
```

If you want to take only one part of a list, use **Part**, which has a short form of double square brackets `[[]]`:

```
In[21]:= Part[{4,1,7},1]
         {4,1,7}[[3]]

Out[21]= 4
         7
```

Exercises

Use Mathematica to solve the following equations. You can use the function **NSolve** rather than **Solve** to force numerical answers rather than fractions. Make sure you use a double equals sign!

1. $3x + 5 = 26$
2. $\frac{3}{x-7} - 5 = y$ (solve for x)
3. $x^2 - ax - 28 = 0$ (remember to use `*` or leave a space between **a** and **x**).
4. $x^2 - bx + 2 = 0$
5. $x^3 - 7x^2 - 96x + 432 = 0$
6. $x + 2y = 9, \quad 6x - 7y = -22$
7. $y^2 + 2y - 2x = 3, \quad 2xy + 2x = 0$

1.9 Differentiation

The command `D[expression,x]` produces the derivative of `expression` with respect to x. A second or higher order derivative can be calculated by `D[expression,{x,n}]`, where `n` is the order of the derivative.

```
In[22]:= D[Sin[x], x]
         D[x^2 + x + 1, x]
         D[x^2 + x + 1, {x,2}]

Out[22]= Cos[x]
         1 + 2 x
         2
```

The value of the derivative at a point can be found by applying a replacement rule afterwards:

```
In[23]:= D[Sin[x], x] /.x->Pi
         D[x^2 + x + 1, x] /.x->-3
         D[x^2 + x + 1, {x,2}] /.x->7

Out[23]= -1
         -5
         2
```

For a function of one variable, you can use a prime to indicate the derivative with respect to the argument:

```
In[24]:= Sin'[x]

Out[24]= Cos[x]
```

This also allows for the point to be substituted by writing e.g.

```
In[25]:= Sin'[Pi]
```

```
Out[25]=  -1
```

Exercises

Find the derivatives (with respect to x) of the following.

1. $2x^5 - 3x^3 + 1$
2. $x^6 \sin x$
3. $\frac{\sin x}{x}$
4. $\sin\left(\ln\left(\tan\left(5e^{x^2}\right)\right)\right)$

1.10 Integration

The command **`Integrate[expression,x]`** calculates the indefinite integral of **`expression`** with respect to x. To compute a definite integral with respect to x between limits $x = a$ and $x = b$, use **`Integrate[expression,{x,a,b}]`**.

```
In[26]:=  Integrate[Sin[x], x]
          Integrate[x^2 + x + 1, x]

Out[26]=  -Cos[x]
               x^2   x^3
          x +  --- + ---
                2     3
```

While Mathematica is able to integrate many expressions using a wide variety of techniques, if it can't manage to find an integral, it will return the original expression unevaluated. Other times it will return a special function, which are typically defined to be a particular integral. To evaluate the integral numerically, use **`NIntegrate`** instead (only for definite integrals).

Exercises

Use Mathematica for the following integrals

1. $\int (2x+5)\ dx$
2. $\int_0^1 (2x+5)\ dx$
3. $\int x\sin(x^2)\ dx$
4. $\displaystyle\int \frac{4x+e^{x^2}}{(\sin x)^{5/3}} dx$

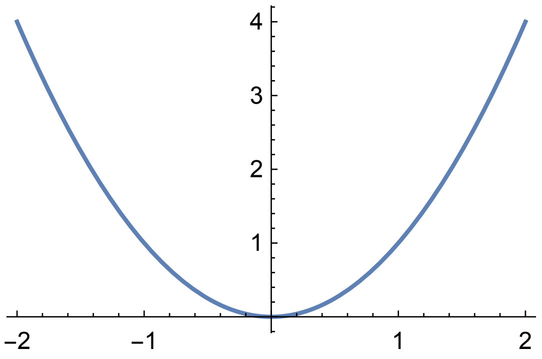
1.11 Graphics

Plots in Mathematica are simple to generate and usually look reasonable without altering the default settings, but they are easy to customise too.

The command **`Plot[y,{x,a,b}]`** gives a graph of y as a function of x between $x = a$ and $x = b$. For example:

```
In[27]:=  Plot[x^2, {x,-2,2}]
```

Out[27]=

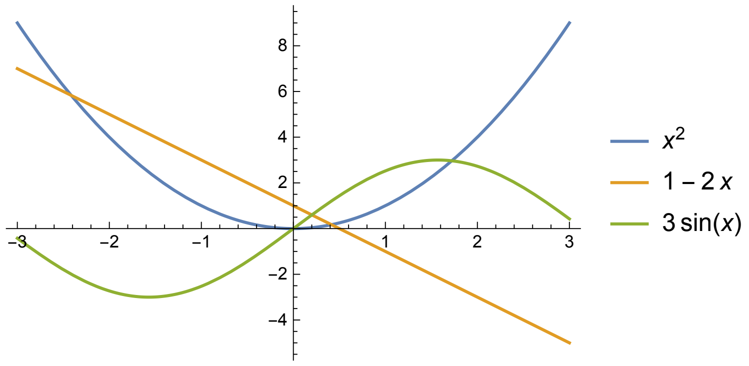


There are many different options that can be given to the **Plot** command in order to change the appearance of the plot. For instance, **Plot[y,{x,0,5},PlotRange->{0,1}]** restricts the y axis to values between 0 and 1.

Multiple functions can be plotted in the same graph via **Plot[{y,z},{x,0,5}]**:

```
In[28]:= Plot[{x^2, 1 - 2x, 3 Sin[x]}, {x, -3, 3},
         PlotLegends -> "Expressions"]
```

Out[28]=



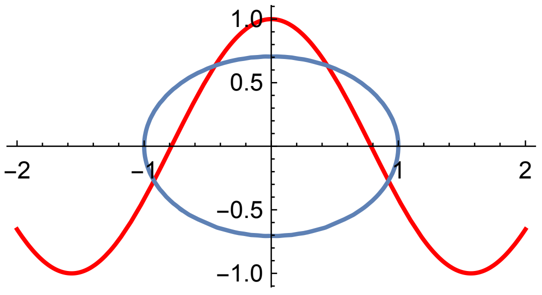
Here **PlotLegends->"Expressions"** adds legends to the plot automatically from what was included in the **Plot** command; to specify manual text legends instead, write

PlotLegends->{"legend1","legend2"}.

Alternatively, multiple **Graphics** objects can be combined using the **Show** command, here **ContourPlot** is used to plot an implicit function of x and y. You may need to include **PlotRange->All** as an option for either **Plot** or **Show** to ensure everything is included. This command is shown here over two lines in order to fit to the page, spaces and line breaks are ignored:

```
In[29]:= Show[Plot[Cos[2x], {x, -2, 2}, PlotStyle -> Red],
         ContourPlot[x^2 + 2 y^2 == 1,{x, -3, 3},{y, -3, 3}],
           AspectRatio->Automatic]
```

Out[29]=



Here **AspectRatio->Automatic** is used to make the x and y axes be on the same scale. See what happens if you leave it out.

Exercises

Plot graphs of the following

1. $y = x^2$ for $-5 \le x \le 5$
2. $y = \frac{x}{x^2+1}$ for $-5 \le x \le 5$
3. $y = \frac{1}{x}$ for $-1 \le x \le 1$
4. $y = \frac{1}{x}$ for $-5 \le x \le 5$, $-2 \le y \le 2$
5. $y_1 = x$ and $y_2 = x - x^5$ for $-1 \le x \le 1$

1.12 Assigning values to variables

Here we will start using Mathematica as a programming language, storing definitions that we will use later. Initially, Mathematica has no definitions for a variable, so will just return it unevaluated,

```
In[30]:=  a

Out[30]=  a
```

Evaluating the following command (with a single equals sign) will tell Mathematica to associate the value of 4 to the variable **a**,

```
In[31]:=  a = 4

Out[31]=  4
```

After evaluating this, the variable **a** in any Input cell will turn black, showing that there is a definition stored for the variable, which will stay in Mathematica's memory until cleared (using the **Clear** function), or replaced by a new definition. It is important to be aware Mathematica does not look at where the code is in the notebook, assigning a value will affect cells above as well as below the cell. It will even stay assigned if the cell is deleted!

```
In[32]:=  a
          a + 2
          a^2 + a y

Out[32]=  4
          6
          16 + 4 y
```

These variables are stored as part of the *Kernel*, which is the part of Mathematica that performs all the calculations. You can entirely clear the Kernel and remove the effect of everything you have evaluated so far by going to Evaluation › Quit Kernel › Local, or by running **Quit[]**. If you get unexpected results this may well fix the problem!

Variables can be defined in terms of other variables, so **b=a+1** will return 5. The right hand side of this assignment will be evaluated when it is called, 'locking in' the value of any variables at that time. A later assignment for **a** will change the value of **a**, but not **b**:

```
In[33]:=  Clear[a,b]
          a = 1; b = a + 1
          a = 2; b

Out[33]=  2
          2
```

The semicolons here suppress the outputs from printing in the setting of **a**, by making it into a single expression.

This behaviour can be changed by using `:=` (a shorthand for the function `SetDelayed`) instead of `=` (`Set`), so that the right hand side is only evaluated when the function is called. This evaluates the right hand side of the assignment every time the left hand side is called, using the value of any variables at that time.

```
In[34]:= Clear[a,b]
         a = 1; b := a + 1; b

Out[34]= 2
```

Now if we change the value of **a**, **b** will update automatically:

```
In[35]:= a = 2; b

Out[35]= 3
```

Any Mathematica expression can be assigned to a variable, so you can do things such as find the solution to an equation and then use that in later code. This often enables you to simplify your code when referring to the same thing multiple times.

Exercises

Use Mathematica to evaluate the following for $a = 5, b = 7, c = 14$.

1. $a + bc$
2. $\frac{a}{b+c}$
3. $(a^b)^c$
4. $\sin(\frac{a-b}{c})$

1.13 Defining Functions

Often we want to define a function that evaluates to a number when given a value, without specifying the value globally, for example if we want to plot $f(x)$ we want to replace x with a wide range of values. The best way to do this is with the following construct:

```
In[36]:= f[x_] := 1 + x^2
```

where the underscore (`Blank`) here means to match anything and use that for the value of `x` on the right hand side. It is essential to include this underscore, otherwise you are storing a definition for only `f[x]`, not for any other value such as `f[1]`. Once defined, `f` will appear in black text (like the built-in functions).

Now we can evaluate f at any value by putting it in square brackets, this doesn't need to be just x due to the underscore, so we can do function composition easily:

```
In[37]:= f
         f[1]
         f[5]
         f[x]
         f[x^2]
         f[u]
         f[Sin[x]]
```

```
Out[37]= f
         2
         26
         1+x^2
         1+x^4
         1+u^2
         1+Sin[x]^2
```

This function **f** can now be used in the same way as the built-in functions like **Sin**,

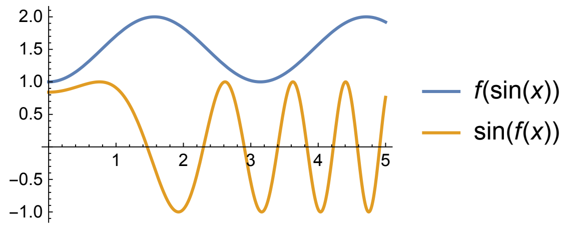
```
In[38]:= D[f[x],x]
         f'[x]
         Integrate[f[x],x]
         Integrate[f[x],{x,0,6}]
         Solve[f[x]==5,x]
```

```
Out[38]= 2x
         2x
              x^3
         x +  ---
               3
         78
         {{x → -2}, {x → 2}}
```

Similarly **f** can be used in **Plot**,

```
In[39]:= Plot[{f[Sin[x]], Sin[f[x]]}, {x, 0, 5},
         PlotLegends -> "Expressions"]
```

Out[39]=



Defining a function like this is very useful, and will be used repeatedly during the rest of the sessions. Note the shorter form **f'[x]** is particularly useful.

Exercises

Consider the function $y = x^4 - x^3 - 18x^2 - 15x + 3.5$. Define a function **y[x]**, and use Mathematica to:

1. find where $y = 0$
2. find the derivative $\frac{dy}{dx}$
3. find where $\frac{dy}{dx} = 0$
4. find the second derivative $\frac{d^2y}{dx^2}$
5. find the value of $\frac{d^2y}{dx^2}$ at points where $\frac{dy}{dx} = 0$
6. find the value of y at points where $\frac{dy}{dx} = 0$
7. use the above information to deduce information about maxima and minima of the curve and points where it crosses the axes.
8. plot a graph of y and verify your conclusions.

End

Make sure that you save your file somewhere that you can retrieve it later. You now have all the material needed to answer Q1 and Q2 on the coursework.