The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

Computer Programming with MATLAB

MM1CPM - Lecture 10

Final important things and the last exam paper

Dr. Roderick MacKenzie
roderick.mackenzie@nottingham.ac.uk
Autumn 2015

www.mm1cpm.com

Announcements

- Coursework deadline Wednesday, 10 December 2015, 3:00 PM, via Moodle.

- We will no longer be using A19 for the examples class.

- Next lecture is on the last day of term.... I will run a help class and go through some problems.

- I will only run the help class if enough people turn up >2% (6 people).

Outline of today's lecture

- Final important things that you should know
  - **Newton's method**
  - Breaking loops early
  - Combining if statements, and switch statement
- Free software
- Last years exam

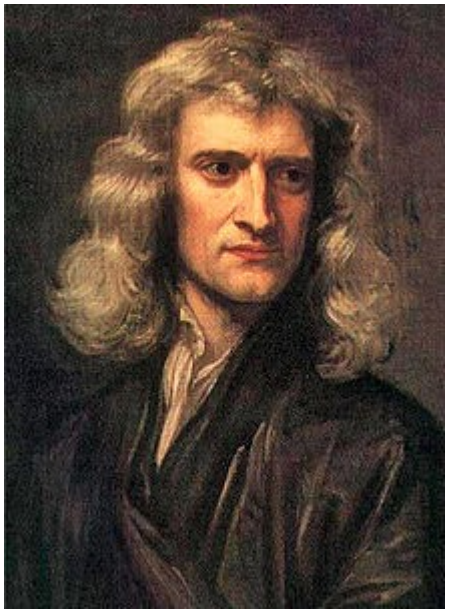
Newton's method

•Newton's method is (in my opinion) the **most powerful** and **robust method available** for solving any equation thrown at you.

•If you can't solve an equation by hand, Netwon's equation will almost always give you an answer.

•Published by Newton in ***1671***, in his book ***Method of Fluxions***.

•I regally use it in my research to solve sets of **$1x10^6$** equations within a few seconds.



Newton's method: let's start with something easy

- For what values of x does this equation equal zero?

$$0=(x-3)(x-40)$$

- i.e. what are the roots?

Any ideas?

Hint:this is not a trick question

Newton's method

- For what values of x does this equation equal zero?

$$0=(x-3)(x-40)$$

$$x=3,\ x=40$$

Newton's method

- Another way to write this equation would be:

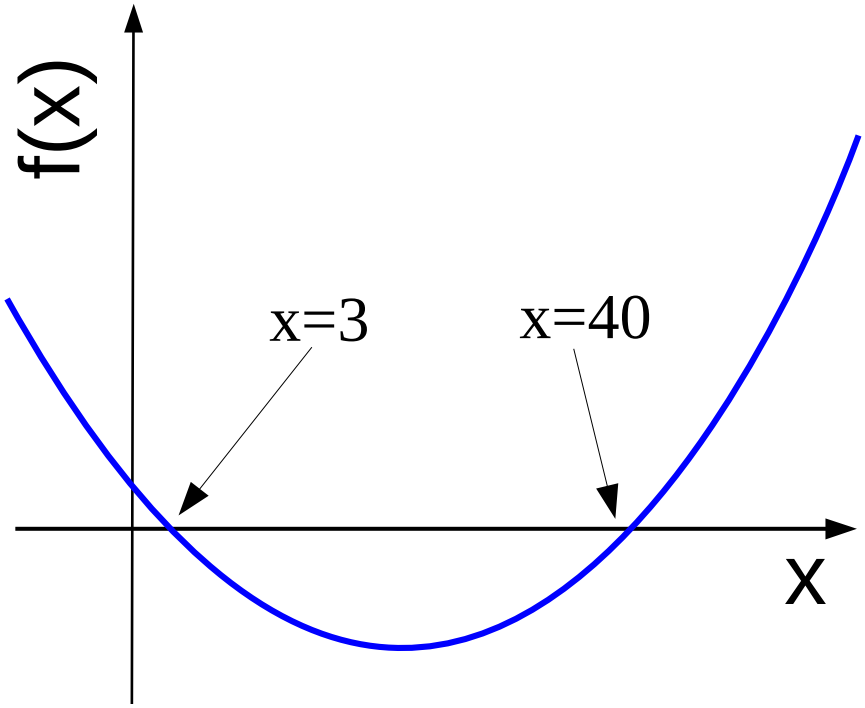
$$f(x)=(x-3)(x-40)$$

Newton's method

- We could find the roots graphically

$$f(x)=(x-3)(x-40)$$

- By plotting f(x)
- When f(x)=0 we have found a root.
- i.e. 3 and 40



Newton's method

- When does this equation equal zero?
- Any ideas?

$$f(x)=(x^2+10x-3)\cos^2(4x)/\tan(x)$$

Newton's method

- When does this equation equal zero?
- Any ideas?

$$f(x)=(x^2+10x-3)\cos^2(4x)/\tan(x)$$

- No, me neither.
- In the next few slides I am going to show you how to solve ANY equation like this using **MATLAB** and **Newton's** method.....

Step 1 Newton's method:
Differentiate the equation.

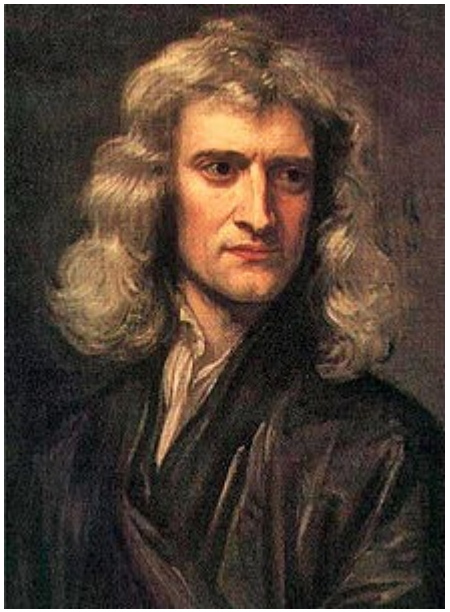
$$f(x)=x^2+10x-3$$

$$\downarrow$$

$$\frac{\partial}{\partial x}$$

$$\downarrow$$

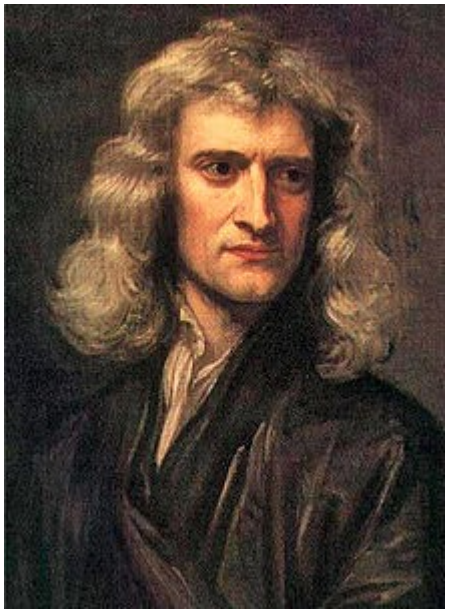
$$f'(x)=2x+10$$



Step 2 Newton's method: Guess the root

What value do you think the root to this equation will be? – give me a guess

$$f(x)=x^2+10x-3$$

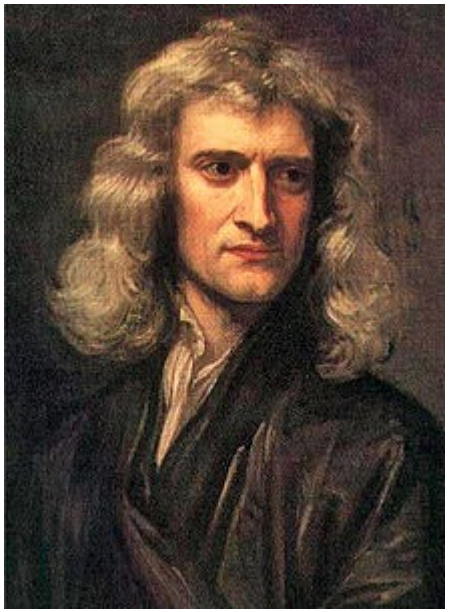


Step 2 Newton's method: Guess the root

What value do you think the root to this equation will be?

$$f(x)=x^2+10x-3$$

My guess is **10** – it's a really bad guess but it does not matter.

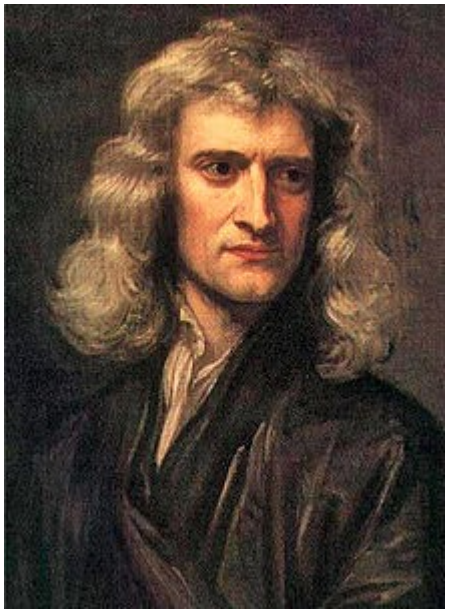


Step 3 Newton's method: Evaluate f(x) and f'(x) at your guess i.e 10.

Now we use our guess to evaluate the equation and the derivative of the equation.

$f(x)=x^2+10x-3$ → $f(10)=197$

$f'(x)=2x+10$ → $f'(10)=30$

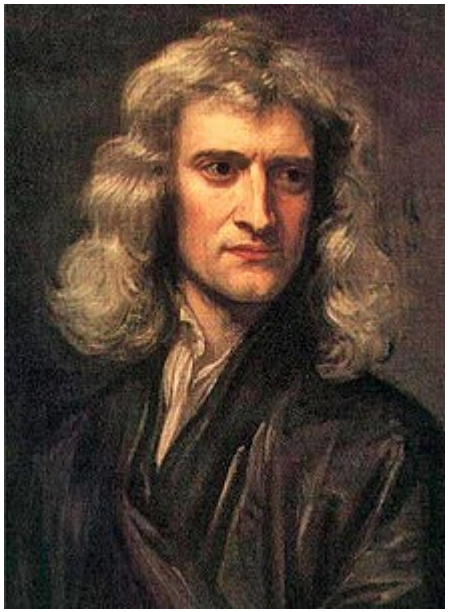


Step 4 Newton's method: Calculating a better guess:

Newton said use this equation to calculate a better guess of x i.e. x_{+1}:

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

Better guess of x



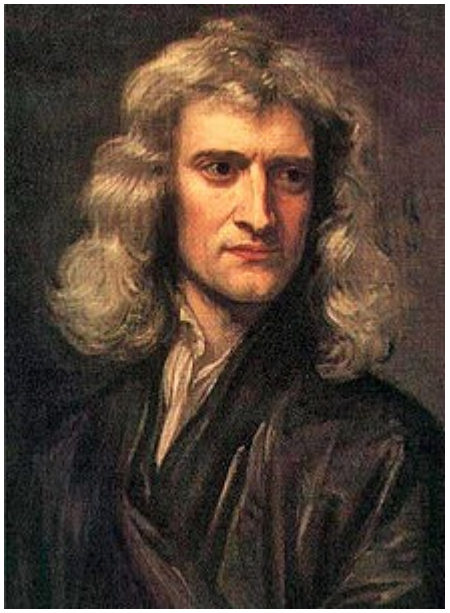
Step 4 Newton's method: Calculating a better guess

We know x

x=10

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

Better guess of x



Step 4 Newton's method: Calculating a better guess

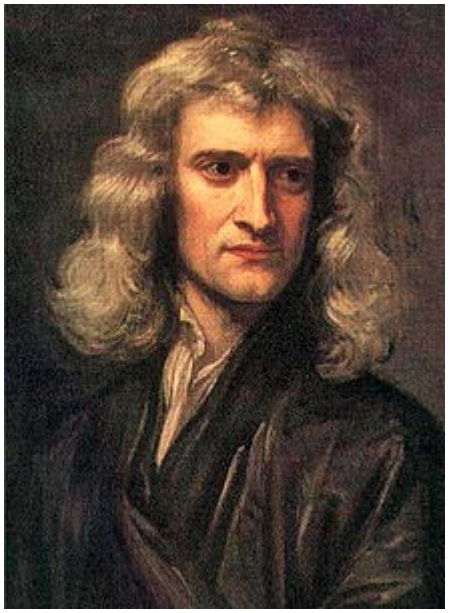
We know f(10)

f(10)=197

x=10

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

Better guess of x



Step 4 Newton's method: Calculating a better guess

We know f'(10)

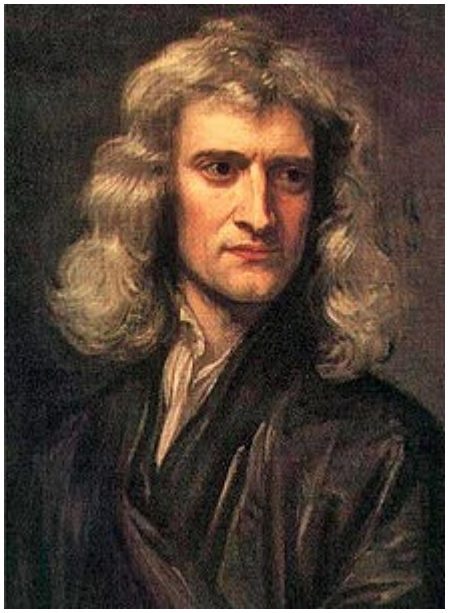
f(10)=197

x=10

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

Better guess of x

f'(10)=30



Step 4 Newton's method: Calculating a better guess

So our better guess of x is

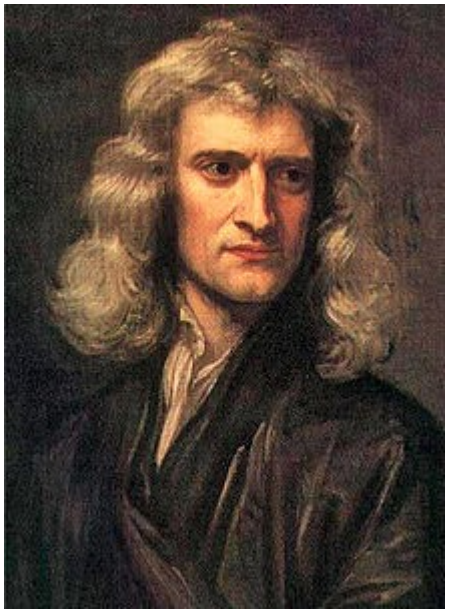
f(10)=197

x=10

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

3.433

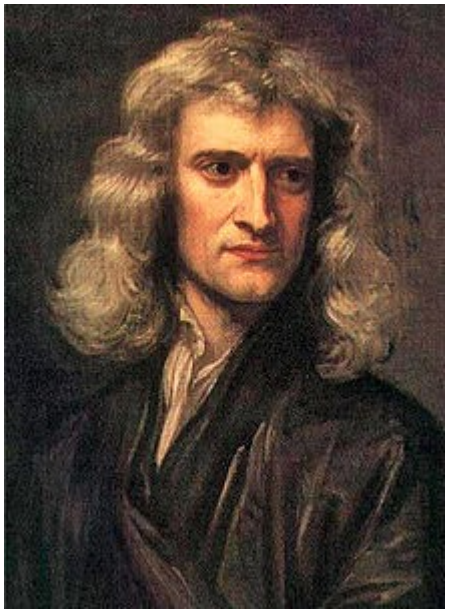
f'(10)=30



Step 4 Newton's method: Now use 3.433 to evaluate the equation again

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

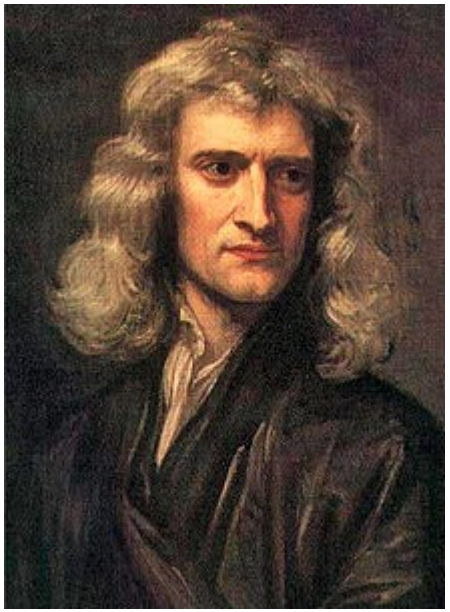
3.433



Step 4 Newton's method: Calculating a better guess

x=3.433

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

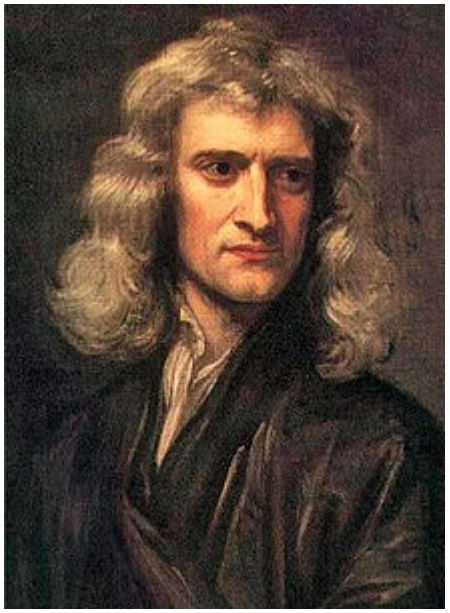


Step 4 Newton's method: Calculating a better guess

f(3.433)=43.115

x=3.433

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



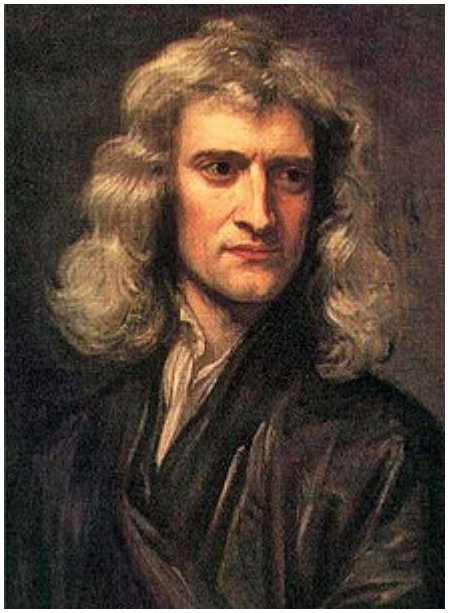
Step 4 Newton's method: Calculating a better guess

f(3.433)=43.115

x=3.433

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

f'(3.433)=16.866



Step 4 Newton's method: Calculating a better guess

And now we can calculate an even better guess of x

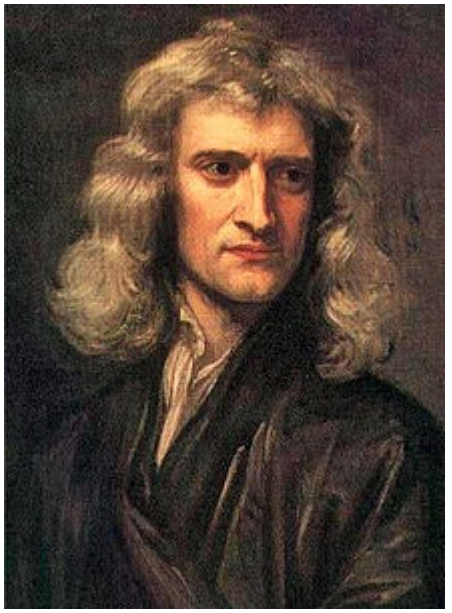
f(3.433)=43.115

x=3.433

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

0.87697

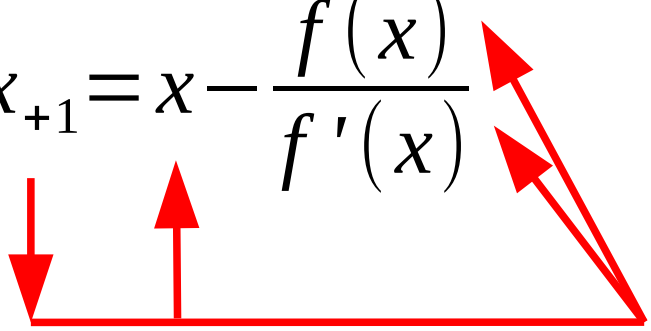
f'(3.433)=16.866



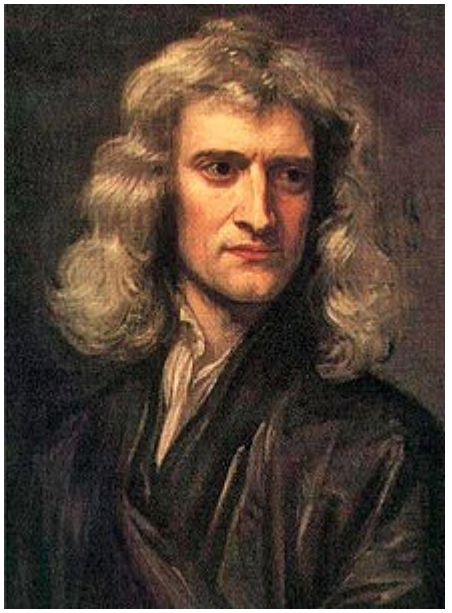
Now use the equation to keep improving the guess of x.

Finally he said if you keep repeating this process you will get better and better guesses of x

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



A loop



Let's try to implement this in MATLAB....

An example of Newton's method using a for loop

```
x=10                          %our guess for the answer
for i=1:10                    %loop ten times
    y=x*x+10*x-3;             %f(x)
    dy=2*x+10;                %f'(x)
    new_x=x-(y/dy);           %calculate new x
    x=new_x;                  %update x
    disp(x)                   %print the answer
end                           %repeat
```

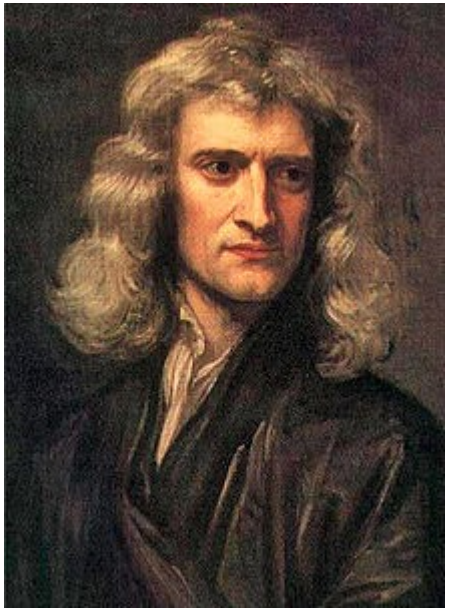
YouTube example

Newton's method summary

- You can change f(x) to ***any*** equation you want and you will get the answer.

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

A loop



- Tip – if you make a mistake and your program does not stop running you can press ctrl+C to stop it.

Outline of today's lecture

- Final important things that you should know
  - Newton's method
  - **Breaking loops early**
  - Combining if statements, and switch statement
  - Free software
- Last years exam

Breaking loops early 1/3

•Think about this piece of code, lets figure out what it does:

```
r=rand(100,1);
found=0
for i=1:100
    if (r(i)>0.5000)
       if (r(i)<0.53)
        found=1
       end
    end
end
if (found==1)
disp('Found')
end
```

How long will the for loop run for?

How could it be improved?

Breaking loops early 2/3

•Think about this piece of code:

```
r=rand(100,1);
found=0
for i=1:100
    if (r(i)>0.5000)
       if (r(i)<0.53)
        found=1
        break;
       end
    end
end
if (found==1)
disp('Found')
end
```

•The break command exits a loop early.

•It can be used to speed up your code.

Breaking loops early3/3

- The **break** command exits a loop early.
- It works with **for** and **while** loops.
- How long will this loop run for?

```
for i=1:100
    if (i==10)
      break;
    end
end
```

- The break command can save your program a lot of time especially with large data sets. Again have a play with it.

Outline of today's lecture

- Final important things that you should know
  - Newton's method
  - Breaking loops early
  - **Combining if statements, and switch statement**
  - Free software
- Last years exam

Combining conditions

- Think about this piece of code below
- When will it display the word OK?

```
if (x>10)
    if (x<15)
        disp('OK');
    end
end
```

This is one way but to express this in MATLAB but there is another way.....

Combining conditions the AND statement

- We could write it like this:

```
if (x>10)
    if (x<15)
    disp('OK');
    end
end
```

```
if (x>10)&&(x<15)
    disp('OK');
end
```

The && is an AND statement, i.e. in English:

if **(x>10) AND (x<15) the condition is met**

Combining conditions the OR statement

We can also do a logical or

```
if (x>10)||(x<5)
    disp('OK');
end
```

The || is an OR statement, i.e. in English:

if **(x>10) OR (x<15) the condition is met**

Combining conditions (summary)

Test	Description	Example
&&	AND	If (x>10)&&(x<15)
||	OR	If (x>10)||(x<5)

- Much neater than nested if statements
  btw. Nested if statements are bad.

Outline of today's lecture

- Final important things that you should know
  - Newton's method
  - Breaking loops early
  - Combining if statements, **and switch statement**
  - Free software
- Last years exam

Alternatives to the **if** statement

•If you look at this code (right), it looks a little complicated. Imagine if you had 100 statements, every statement would require an if and an elseif.

• There is a more elegant alternative.. the **switch** statement.

```
choice=3                          %speed in mph
if (choice==1)
        disp('The value equals one')
elseif (choice==2)
        disp('The value equals two')
elseif (choice==3)
        disp('The value equals three')
else
        disp('Not in range')
end
```

Alternatives to the switch statement

- Here is an example of a switch statement
- Notice you don't need to repeat the **elseif** command or the if command.
- They have several advantages over else-if-else statements....

```
a=input('Number')
switch a
        case 1
        disp ('you have entered one')
        case 2
        disp ('you have entered two')
        case 3
        disp ('you have entered three')
        case 4
        disp ('you have entered four')
        otherwise
        disp ('Entry not recognized.')
end
```

See example sheet: Q6g WS6.

Outline of today's lecture

- Final important things that you should know
  - Newton's method
  - Breaking loops early
  - Combining if statements, and switch statement
  - **Free software**
- Last years exam

Some comments from the Student evaluation of the module.

1.Hard to complete example worksheets from week 7 onwards, when doing coursework on work from weeks 3-6, **as not a lot of people have matlab on their personal devices**

2.Some way of having MatLab **full to download for free at home** would be useful, then coursework would be more likely to be completed at home and the worksheets done in the lab class.....

3.If, like we have for Creo, **we could download our own copies of Matlab, even if it's for a discounted fee and not for free,** it'd be very helpful so we could all have our own copies on our own laptops and PCs to try out new scripts in our own times too.

4.Can get a bit heavy for students who havent programmed before, and the fact that we cant **have matlab for free at home doesnt help for sure.**

Can I have MATLAB at home for free then?

Can I have MATLAB at home for free then?

No you can't have it for free.

That seems unreasonable, why?

That seems unreasonable, why?

- It's simple, **Mathworks** who distribute MATALB simply don't allow free copies to be distributed by the university – **they want money for it.**

- They also won't allow us to add students to the campus license. (Even though other companies like the people who make Creo do allow you to have your own copies).

- We are **bound by the terms of the MATLAB license. :(**

This was the problem faced by this man 1983:

•He wanted a FREE operating systems for his state of the art computer:

Ruben de Rijcke

Thesupermat

•At the time there were no free operating systems.

•So he starred the **Free Software Movement** to solve this problem.

This was the problem faced by this man 1983:

- And he wrote a very important **document** called the **General Public License (GPL):**

- In essence, this was a license for software. Rather than be restrictive like the **MATLAB license**, his license said that:

    - You can use my software that I have spent time writing **FOR FREE** and do **what ever you want with it**.

    - **BUT** you must allow other people to:
        a) **Change it**
        b) Also **redistribute** the improved software for **FREE**.

So it was a license that gives the user **freedom** rather than takes it away (like the MATLAB license - dig)

Free software today.

- Today the **Free Software Movement** is massive, there are millions of free programs written and distributed using Richard Stallman's license, here are three you may have heard of:
  - **Android**, **Moodle**, **Octave**, **Firefox**, The list is endless.

Free software today.

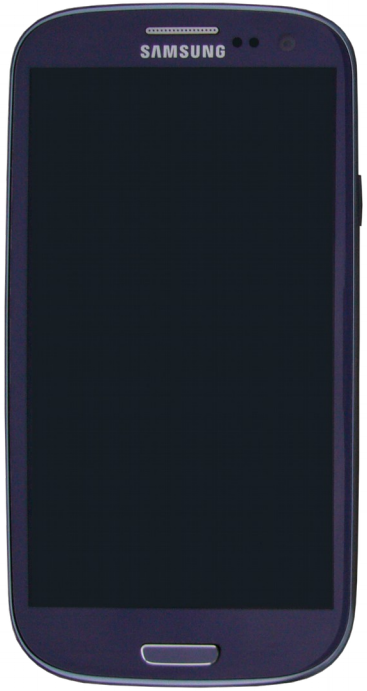
- Almost all large companies today rely on free software to run their critical infrastructure such as **Google** and **Wikipedia**.

- Because this software is **so important to these companies** they actually spend time and money improving it.

- But because of Richard Stallman's license they **have to give the improved software back to the community** so everybody can benefit form the improvements.

The power of free software

•Imagine you are an engineer developing the Samsung SIII mobile phone:



Illythr

•And by improving the software (Android) on the phone you find a way to save 1% battery power.

•You get a pat on the back from your boss.

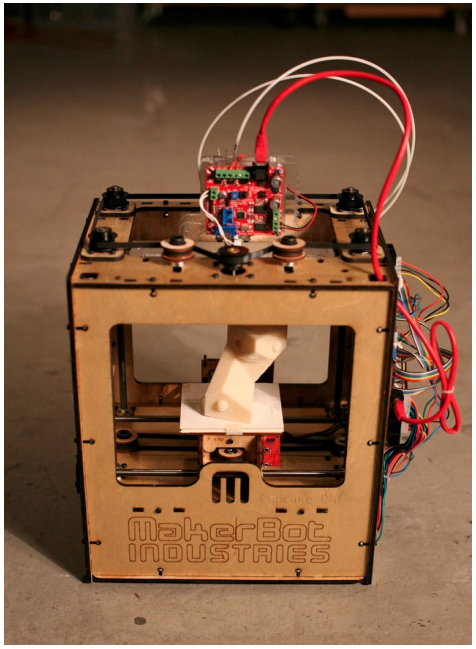
•You then upload your improved version of Android to the web (because you have to).

•Then other companies can also use this improved version of Android, and all of a sudden all phones in the world running Android save 1% battery life - even better.

That's good

•But it gets better, because the code to Android is also used in lots of other computers:

Steve Jurvetson



http://www.flickr.com/photos/bre/3458247336/

•wikipedia user sjr

•They also all save 1% power, which world wide adds up to a lot of energy.

•That's the power of free software.

The point I'm making is that..

- Free software is a huge **cultural movement**.

- The point is that, very often there is an alternative to a expensive proprietary product that is just as good if not better.

Outline of today's lecture

- Final important things that you should know
  - Newton's method
  - Breaking loops early
  - Combining if statements, and switch statement
  - Free software
- **Last years exam**

The Exam

- The exam will be a **written exam (i.e. with a pen).**
- It will **not be computer based.**
- It will be 1.5 hours long.
- There will 4 sections
- There are no optional questions
- You will **not** be allowed to bring notes/books into the exam.

The Exam

- There will be lots of marks for commenting your code. Regularly space comments throughout your code.

- If you write a function make sure you put comments above it – as I showed you how to when covering functions.

- Your comments must be understandable by an English speaker.

- However, there will be **no marks at all for grammar or spelling** – it is a matlab exam. What you write must be however understandable.

The Exam

- MM1CPM was a new module last year. So there is only one past paper for MM1CPM which I wrote.

- However, MM1CPM is ***identical*** to MM2CPM which was taught in 2012,2011,2010.

- The exam ***format/scope*** is also ***identical*** to that of previous MM2CPM exams.

- The questions on this year's paper will be ***very slightly easier*** than previous MM2CPM exams ***but not much***.

The Exam

- In summary you can use MM2CPM papers as a very very useful revision tool for MM1CPM.

- You can find the past papers on the moodle page for this module.

- I recommend you use the example sheets first then do the practice papers last.

- Passing this exam will be about practicing MATLAB **not cramming** before the exam.

- An example paper....

The University of Nottingham

DEPARTMENT OF MECHANICAL, MATERIALS AND MANUFACTURING ENGINEERING

A LEVEL 2 MODULE, SPRING SEMESTER 2012-2013

COMPUTER PROGRAMMING

Time allowed ONE Hour <<<you get 1.5 hrs

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced.

Answer ALL questions

Only silent, self contained calculators with a Single-Line Display, or Dual-Line Display are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

ADDITIONAL MATERIAL: None

Section A

- Short answer questions where you have to write some simple MATLAB.

SECTION A: Command-line instructions

For each of the following, state what MATLAB commands you would use to

1. Define the following matrix in MATLAB $x = \begin{bmatrix} 1 & 2+i & 3 \\ 4 & 5 & 6+i \end{bmatrix}$

2. Establish a 6x6 array called 'a' containing random numbers , then establish a 1x6 array called 'b' also containing random numbers. Finally, multiply matrix 'b' by matrix 'a' storing the result in z.

3. Plot a graph of sin(x) between x=0 and x= 2π using 1000 points.

4. Evaluate the equation $y = \frac{\left(e^{-x} + e^{x}\right)^2}{2}$ for x=1.0

5. Write the words "Hello World!" to a file using a script containing the ***fopen*** command.

SECTION A: Command-line instructions

For each of the following, state what MATLAB commands you would use to

1. Define the following matrix in MATLAB $x = \begin{bmatrix} 1 & 2+i & 3 \\ 4 & 5 & 6+i \end{bmatrix}$

2. Establish a 6x6 array called 'a' containing random numbers , then establish a 1x6 array called 'b' also containing random numbers. Finally, multiply matrix 'b' by matrix 'a' storing the result in z.

3. Plot a graph of sin(x) between x=0 and x= 2π using 1000 points.

4. Evaluate the equation $y = \frac{\left(e^{-x} + e^{x}\right)^2}{2}$ for x=1.0

5. A very simple question about computer hardware.

Section B:

- Short answer questions where you have to remember what MATLAB commands do and give short examples of how you would use them.

SECTION 2: Usage of MATLAB features

For each of the following features within MATLAB, state how it might be used – using illustrations where possible.

7. while
8. round (.)
9. disp(.)
10. break
11. size(.)
12. surf(.)

Tip: if it is in **my lecture notes/example sheets** the function could appear in this section.

If it's not in my lecture notes/example sheets – it probably won't appear.

Section C:

- This section will test your ability to understand short MATLAB programs.
- You will have to write what each line does
- Then interpret the overall meaning of the program.
- **This year there will be a single mistake in the code that you will have to find.**

Section C: Second question

Q13a	`a=load('data.dat');`
Q13b	`orig=a;`
Q13c	`L=size(a);`
Q13d	`out=zeros(L);`
Q13e	`for nn=1:5`
Q13f	`for n=2:(L-1)`
Q13g	`s=(a(n-1)+a(n)+a(n+1))/3;`
Q13h	`out(n)=s;`
Q13i	`end`
Q13j	`a=out;`
Q13k	`end`
Q13l.	`plot(out);`

Section C: Second question

14.

Q14a	max=input('max')
Q14b	failed=0; count=0; z=zeros(max,max)
Q14c	while (count<10)
Q14d	rx=floor(rand(1)*max)+1; ry=floor(rand(1)*max)+1;
Q14e	if z(ry,rx)==0
Q14f	z(ry,rx)=1;
Q14g	count=count+1;
Q14h	else
Q14i	failed=failed+1;
Q14j	end
Q14k	end
Q14l	disp('Failed:');disp(failed);z

Section D

- This section will require you to write longer MATLAB programs.

- The question normally has some type of engineering or science context.

Section D (Winter 2014):

SECTION 4: Writing MATLAB code

15. The location of ships in the sea is recorded by a satellite 100 km above the earth and the positions of the ships are transmitted to earth every minute. The locations of the ships are represented by x,y coordinates on a 180x100 km grid (see Figure 1). A computer on earth receives the positions of the ships and writes the data to a file called <**ship_locations.dat**>. Each line in the file holds the position of one ship. The file has two columns of data, column 1 holds the ***x*** position of the ship and column 2 hold the **y** position of the ship. When new data is received the file is over written.

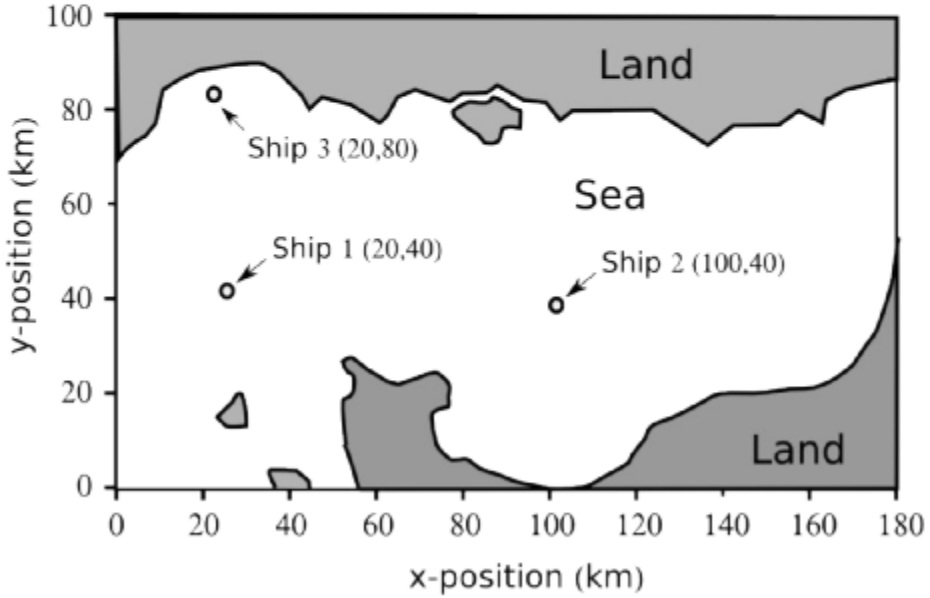


Figure 1: The satellites view of earth. The ships and the x-y grid have been overlaid.

Section D (Winter 2014):

(a) Write a MATLAB script to read the data file, load the ***x*** and ***y*** coordinates into array <***Cx***> and <***Cy***>. Calculate the number of ships recorded in the file and output this to the screen. Then plot the locations of the ships in a labeled graph using black circles.

[25 marks]

(b) Write a Matlab function ***<collision>*** that accepts the position of two ships and then calculates and returns the distance between them in km. This function should be stored in it's own script file file called <***collision.m***>. The function ***collision*** *should have the following format*
» d=collision(ship1_x,ship1_y,ship2_x,ship2_y);

[25 marks]

Section D (Winter 2014):

(c) Write a new Matlab script which begins with the word script that uses uses the function ***collision*** *to check the distance between each ship in the file and prints this to screen. Use* ***while*** *loops to do this* ***not for loops***.

[30 marks]

(d) *Change the script so that if two ships are closer than 0.5km it prints the word "Warning!", the locations of both ships and the distance by which they are separated. (Note: The script should not give false warning by*

Section D (Winter 2014):

(e) Before new ship locations are received from the satellite, the computer moves the current 'ship_locations.dat' file to a back up file called 'old_ship_locations.dat'. Because we know that the satellite transmits new ship locations every 60 seconds we can use these two files to calculate the speed and direction of the ships. Write a MATLAB script to load the two files in to matrices **<ship_locations>** and **<old_ship_locations>**, then calculate the x-y vectors representing their change in position in the last 60 seconds and print each vector the scree in the format.

"The ship has moved 1km in the x-axis and 1km in the y-axis"

where 1km would be replaced by the calculated distance traveled.

[22 marks]

Questions?

- Next week, it's the last Friday of term.....
- Turn out is usually quite low.
- I will therefore, come to the lecture to answer any questions you may want to
  - On past exam papers
  - On example sheets
  - Etc etc...
- There will be no new material.