Mathematical Modelling – assignments week 2:
Complex numbers & Vector and matrix calculations

The MATLAB Desktop is a Graphical User Interface, which consists of several windows, like:

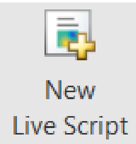
- Command Window,
- Current Directory of Workspace,
- Workspace.

In principle, you could enter all your commands in the Command Window. We will use the Command Window to try some commands. However, we will not use this approach for assignments the labs, since we need to keep them.

Handing in your assignments (a)?:

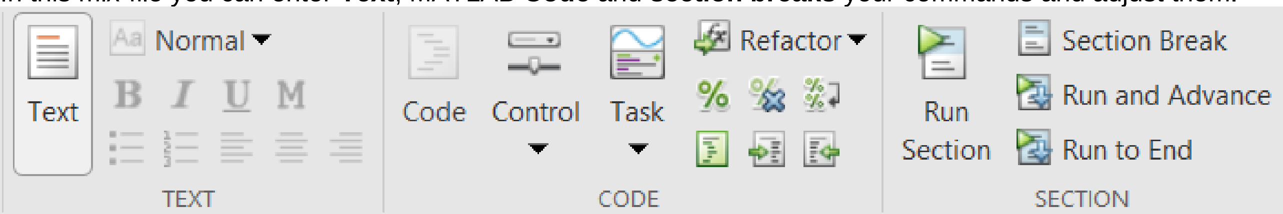
Since the command window of MATLAB has limited editing capabilities, and since you want to keep the assignments at the end of the lab, we place all assignments in an **mlx-file.** This mlx-file is a file, in which you can place text, MATLAB commands and section breaks (and even more fancy things like images, formulas in LaTeX etc.).

To create such an mlx-file, navigate to the the Home tab, where you can find the New Live Script Icon:



If you click on it, a new mlx-file will be opened in the editor.

In this mlx-file you can enter **Text**, MATLAB **Code** and **section breaks** your commands and adjust them.



In this document there are multiple assignments. You should start each assignment with Text-block in which you indicate the assignment (or subassignment). Then there should be a Code-block with MATLAB commands and you should end with a Section Break. In this way you can execute a single section by pushing the “Run Section”-icon. You still can run a complete live script by pushing the “Run”-icon. That might be useful if you wan to export the final result as a pdf-file.

Handing in your assignments (b)?:

Start each mlx-file with a Text-block, which contains your name and student number, e.g.

```
Name: Jane Doe
Student number: 123456
Assignments week 2
```

Handing in your assignments (c)?:

Create a section per separate assignment (or subassignment, if that makes sence), and start it also with a Text-block, e.g.

```
Assignment 3c
```

The naming conventions of an mlx-file should be according to the following requirements:

- Use only letters, numbers and/or underscores (_), so no point, space, or commas.
- Begin with a letter!

Good names are e.g.: assignments_week_2.m, or assignmentsWeek2.m

The default folder for MATLAB is already created. It is also useful to use this folder for all your mlx-files!

Regular rows of numbers

There are two ways to quickly define a regular row of numbers in MATLAB. In the first way you give a start value, a step size and an end value. In the second way you give a start value, an end value and the number of values you want to have in total. In this case MATLAB itself will calculate the step size.

First way

```
x = [start:stepsize:end]
```

Second way

```
x = linspace(start,end,number of values)
```

Examples:

```
x = [-10:1:10]
```

is a row of 21 numbers, from `-10` to `10`, with increments of 1 between each following number (check it). Note that an increment of 1 is the default, so the following has the same result:

```
x = [-10:10]
```

```
y = linspace(-10,10,20)
```

is a row of 20 numbers from `-10` to `10` with mutually equal distances.

The seventh number in x can be retrieved with the following command (note the parentheses):

```
X(7)
```

Note that x(1) is the first number in x, unlike most programming languages (e.g. C), where the first number (in an *array*) is at index 0.

The range of numbers at positions 10 through 15 in y can be retrieved with the command:

```
y(10:15)
```

The number of elements in the row x can be retrieved with the command:

```
length(x)
```

The first half of a row with an even number of elements, for example row y can be obtained with:

```
y(1:length(y)/2)
```

With the command `clear`, all variables you have defined will be cleared.
It is a good idea to start new assignments with a `clear` command, so that you will start with a clean slate.
A useful command for the command window is `clc` which will clear the command window.

Assignment 1

Make a row of numbers from 1 to 5 with step size 0.2, and store this in the variable x.
What is the 19^{th} element of this row? How many elements does the row contain? You can include the answers to these two questions as comments in the m-file (a comment line starts with a % sign)

Assignment 2

Make a row (again use x as the variable) of 100 numbers in ascending order, beginning with 35 and ending with 47, at equal intervals.
What are the 19^{th} up to the 22^{nd} elements of this row? Again, you can include the answers to these questions as comments in the m-file.

Complex numbers

MATLAB can calculate with complex numbers as well. There are a couple of ways of how to use complex numbers in MATLAB, shown below:

```
z = 2 + 3i
```

or

```
z = 2 + 3*i
```

or

```
z = 2 + 3j
```

or

```
z = 2 + 3*j
```

Please note that in MATLAB the complex number 'j' gives the same result as using an 'i', but the answer will be shown using an 'i'. The * sign is not mandatory in the example above, but: if you would use a variable in front of i the * sign **is** mandatory, however.

```
a=3;
z=2+aj    % This will raise an error
z=2+a*j
```

You can ask MATLAB to give you the real and imaginary part of the complex number z = a +bj, as well as the modulus (the absolute value of the length) and the argument (the angle).
Beware! When using the modulus or argument function, MATLAB will calculate rounded values, and not exact values.

```
z = (3-5j)*(4-2j)
a = real(z)
b = imag(z)
r = abs(z)
phi = angle(z)
```

Assignment 3

Given are the complex numbers $z_1 = 3+4j$ and $z_2 = -1+j$

a. Calculate the modulus and argument of z_1 and z_2 using MATLAB

b. Calculate $z_1 \cdot z_2$ and $\frac{z_1}{z_2}$ using MATLAB

c. $\frac{1}{z_3} = \frac{1}{z_1} + \frac{1}{z_2}$. Calculate z_3 using MATLAB

d. Calculate the modulus and argument of z_3 using MATLAB

Plotting graphs

Open MATLABs help function browser (Shift F1) and search for help on the commands **`grid`, `axis`, `legend`, `xlabel`** and **`ylabel`**
Beware: these commands apply to the previously made plot, so you'll need to run these commands **AFTER** you have run the plot command itself.
E.g.:

```
x=linspace(0,2*pi,100);
y=sin(x);
z=cos(x);
plot(x,y,x,z)
axis([0,2*pi,-1,1])
grid on
title('A sine and a cosine')
legend('Graph of sin(x)','Graph of cos(x)')
xlabel('0 <= x <= 2\pi') % x-axis label
ylabel('sine and cosine values') % y-axis label
```

Handing in your assignments (d)?:

You should always use `title`, `xlabel` and `ylabel` when making a plot, not only to beautify the plot, but also to make it more readable. If you plot more than a single output, you should also use `legend`.

`title`, `xlabel` and `ylabel` can also understand LaTeX.
Try e.g. `xlabel('0 \leq x \leq 10')` instead of `xlabel('0 <= x <= 10')`

Plotting complex numbers

You can also plot complex numbers.

Beware:
When using the plot-command you need **TWO** input variables (x and y). But if you plot a **complex number** you only need **one** input variable (z); MATLAB will use the real part of the complex number as x-coordinate and the imaginary part as y-coordinate.

Example:
```
z=2+3j;
plot(z)
```

To make the points in the plot better readable, you can assign a symbol to use for a point (like a circle, a plus or a cross symbol). For the circle you should use the letter o (not the number 0), for a plus you should use a '+', and for a cross you should use an 'x'.

Example:
```
plot(z,'o')
```

The following plot contains several points. These points are the complex number $z = \frac{1}{2}\sqrt{3} + \frac{1}{2}\,\mathrm{j}$, raised to the powers 1 through 12.

```
z=.5*sqrt(3) + .5i;
plot(z.^[1:12],'o')
```

Beware of the dot before the ^ sign.

A better result can be obtained using the `axis equal` statement:

```
plot(z.^[1:12],'o')
axis equal
```

If you use the plot-command using a real number x and complex number y, MATLAB will give you a warning that the imaginary part of y will be ignored, since only a two-dimensional picture will be plotted.

For example:
```
x = linspace(0,10,1000);
y = log(x-2);
plot(x,y)
title('The real values of ln(x-2)')
xlabel('0 \leq x \leq 10') % x-axis label
ylabel('ln(x-2)') % y-axis label
```

```
Warning: Imaginaryparts of complex X and/or Y arguments ignored
```

You'll notice that only the real part of y has been plotted, but not the imaginary part.

You might have seen this warning in the previous assignments.

Creating multiple plots simultaneously

If you want to plot multiple functions in the same plot window simultaneously, you can use the `hold on` command. To disactivate this option, you should use the `hold off` command:
```
x=[0.01:0.01:3];
y1=x.^2;
y2=log(x);
y3=exp(x);
plot(x,y1)
hold on
```

```
plot(x,y2)
plot(x,y3)
hold off
title('3 functions together')
legend('Graph of x^2','Graph of ln(x)','Graph of e^x')
xlabel('0.01 \leq x \leq 3') % x-axis label
ylabel('x^2, ln(x) and e^x values') % y-axis label
```

If you want to have multiple plots in multiple separate windows you can open a new window for each new plot by using the `figure` command just before the new plot-command:

```
x=[0.01:0.01:3];
y1=x.^2;
y2=log(x);
y3=exp(x);
plot(x,y1)
title('Graph of x^2')
xlabel('0.01 \leq x \leq 3') % x-axis label
ylabel('x^2 values') % y-axis label
figure
plot(x,y2)
title('Graph of ln(x)')
xlabel('0.01 \leq x \leq 3') % x-axis label
ylabel('ln(x) values') % y-axis label
figure
plot(x,y3)
title('Graph of e^x')
xlabel('0.01 \leq x \leq 3') % x-axis label
ylabel('e^x values') % y-axis label
```

If you want to create multiple plots, but combined in one single window, you can use the `subplot` command:

```
subplot(2,2,1)
x = linspace(0,10,1000);
y1 = sin(x);
plot(x,y1)
title('Subplot 1: sin(x)')

subplot(2,2,2)
y2 = sin(2*x);
plot(x,y2)
title('Subplot 2: sin(2x)')

subplot(2,2,3)
y3 = sin(4*x);
plot(x,y3)
title('Subplot 3: sin(4x)')

subplot(2,2,4)
y4 = sin(8*x);
plot(x,y4)
title('Subplot 4: sin(8x)')
```

Assignment 4

The variable x contains 100 values between -5 and 5.

$y = x^2$

Make two separate graphs (using `figure`), one with the command `plot(x,y)` and one with the command `plot(y)` instead and compare the two graphs. In your m-file, give an explanation (use a comment) of the differences between these two graphs.

Assignment 5

Given is the complex number $z = \dfrac{1 + j\omega}{1 + j\omega - \omega^2}$.

ω consists of values from 0.1 to 10, with a step size of 0.05

Make three different plots in two different windows.

a. A plot of z itself should be in a separate window,
b. A plot of the modulus of z as a function of ω and a plot of the argument of z as a function of ω should be in a separate window, consisting of two sub windows, one on the top and the other on the bottom.

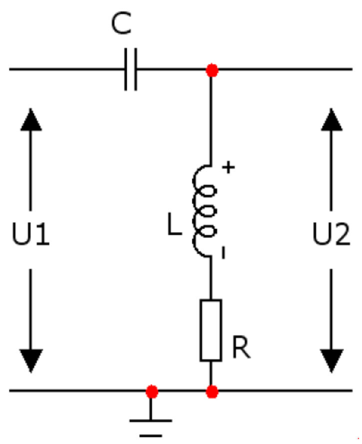
Hint: use both `figure` and `subplot`.

Assignment 6

Search the MATLAB documentation for `logspace` and `semilogx`.

Create a vector f with 1000 values, logarithmic spaced. The smallest value for f must be 10; the biggest vale for f must be 100000.

In the given circuit the following transfer function, $H(f)$, applies:



$$H(f) = \frac{z_2(f)}{z_1(f) + z_2(f)} \text{ with } z_1 = \frac{1}{j\omega C},\ z_2 = R + j\omega L,\ \omega = 2\pi f,$$

$$R = 10,\ L = \frac{R}{2000\pi}, \text{ and } C = \frac{1}{2000\pi R}.$$

Make a plot of the **modulus** of the transfer function $H(f)$ as a function of f. The horizontal scale of the plot must be logarithmic.

Complex alternating signals

Thanks to complex numbers we can calculate the sum of a sine and a cosine, if both have the same angular frequency, and rewrite this sum as a single sine or cosine

Application: adding different alternating signals with the same angular frequency ω.

Example:

$$u(t) = 2\cos\left(3t + \frac{1}{4}\pi\right) + 5\cos\left(3t - \frac{1}{6}\pi\right)$$

In this example we would like to write this sum as a single cosine of the form $r\cos(\omega t + \varphi)$

Notice that in our case ω = 3.

The complex alternating signal (containing both the cosine and the imaginary sine) can be written as:

$$\hat{u}(t) = 2e^{3tj + \frac{1}{4}\pi j} + 5e^{3tj - \frac{1}{6}\pi j}$$

Get the common multiplication factor, $\exp(3tj)$, outside of the brackets.

$$\hat{u}(t) = e^{3tj}\left(2e^{\frac{1}{4}\pi j} + 5e^{-\frac{1}{6}\pi j}\right)$$

We are now going to simplify the constant part $2e^{\frac{1}{4}\pi j} + 5e^{-\frac{1}{6}\pi j}$ (i.e. the part without the t)

In MATLAB this would be:

```
c=2*exp(pi/4*j)+5*exp(-pi/6*j);
r=abs(c)
fi=angle(c)
```

This will give the following results:

```
r = 5.8461
fi = -0.1868
```

Hence: $\hat{u}(t) = e^{3tj} \cdot r \cdot e^{\varphi j} = r \cdot e^{(3t+\varphi)j} = r \cdot (\cos(3t + \varphi) + j\sin(3t + \varphi))$

The real part of this expression is the signal we were looking for, hence the final answer is:
$u(t) = r \cdot \cos(3t + \varphi)$

To check the result, we can plot both graphs and compare them.

Assignment 7
Given are the following alternating signals:

$f(t) = 3\cos\left(5t + \frac{1}{4}\pi\right) \quad g(t) = 2\sin\left(5t - \frac{1}{3}\pi\right)$

$k(t) = f(t) - g(t)$ can be written in the form $r\cos(\omega t + \varphi)$

Calculate the values, in four decimals, of r, ω en φ.

Plot two different graphs in two separate windows:

one graph of $k(t)$ and one graph of the calculated $r\cos(\omega t + \varphi)$

Hint: use $\sin(x) = \cos\left(x - \frac{\pi}{2}\right)$ to be able to calculate the subtraction.

Mathematical vector notation
The point O (point of origin) is represented by a null-vector: all numbers in this vector are 0 (zero).

A random point A is represented by the vector $\vec{\boldsymbol{a}} = \overrightarrow{\boldsymbol{OA}}$ (Starts in O and ends in A)

The vector that starts in a point P and ends in a point Q is the vector $\overrightarrow{\boldsymbol{PQ}}$.

And this applies: $\overrightarrow{\boldsymbol{PQ}} = \vec{\boldsymbol{q}} - \vec{\boldsymbol{p}}$

The middle of the line from P to Q can be calculated as the vector: $\vec{\boldsymbol{m}} = \frac{1}{2}(\vec{\boldsymbol{p}} + \vec{\boldsymbol{q}})$

The number of values of a vector is also called the vector's dimension. In MATLAB you can get the dimension of a vector `x` by means of the command `length(x)`

There are multiple ways to multiply two vectors (with equal dimensions)
The first and most common way is to multiply them using the dot operator.
We call this: **element-by-element-operation**.

Example:
```
clear
x = [2, 5, 3];
y = [4, 1, 3];
x.*y
```
gives as result `[8, 5, 9]`

In vector calculations we know the inner product of a vector, also known as dot-product. The result of this is inner product is a **NUMBER**.
There are multiple ways to calculate this dot product in MATLAB. We´ll just choose this method:
```
dot(x,y)
```

This gives the correct result: 2*4 + 5*1 + 3*3 = 22
Please note: if two vectors are perpendicular, then the dot-product is equal to zero. This property is often used to prove that two vectors are perpendicular.

Assignment 8

Given are the points A(2, 1, 4), B(1, -2, 3) and C(4, 5, 2).

Make the vectors $\vec{\boldsymbol{a}} = \overrightarrow{\boldsymbol{OA}}$, $\vec{\boldsymbol{b}} = \overrightarrow{\boldsymbol{OB}}$, and $\vec{\boldsymbol{c}} = \overrightarrow{\boldsymbol{OC}}$.

Calculate the vector $\vec{\boldsymbol{d}} = \vec{\boldsymbol{a}} + \vec{\boldsymbol{b}} - 3\vec{\boldsymbol{c}}$

Calculate the vectors $\overrightarrow{\boldsymbol{AB}}$, $\overrightarrow{\boldsymbol{AC}}$ and $\overrightarrow{\boldsymbol{BC}}$

Point M is in the middle of the line AB. Calculate the vector $\overrightarrow{\boldsymbol{CM}}$

Calculate the inner product of the two vectors $\overrightarrow{\boldsymbol{AB}}$ and $\overrightarrow{\boldsymbol{AC}}$ (this should give just a number as result!)

Cross and norm

For three-dimensional vectors, there´s another way of multiplication: the outer product, also known as cross product.
In MATLAB:

```
z = cross(x,y)
```

The result is again a vector, but one that is perpendicular to both $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{y}}$. The length of this vector is equal to the area of the parallelogram where $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{y}}$ make up the sides.

Beware: the length of the vector $\vec{\boldsymbol{x}}$ is calculated in MATLAB by means of the command `norm(x)`

Assignment 9

Given are the points A(2, 1, 4), B(1, -2, 3), and C(4, 5, 2).
These three points together form a triangle.
Calculate the area of this triangle.

Matrix calculation

A matrix in MATLAB is entered with an opening square bracket on the first row, then ENTER, then the second row and so on and so on. The last row should be ended with another square bracket].

Example:

```
A = [1 2
     3 4]
```

or you could put a semicolon to start a new row:
Example:

```
A = [1 2; 3 4]
```

You'll get the transposed matrix $\mathbf{A}^T$ by adding the accent.

```
AT = A'
```

The command `size(x)` will give you the precise dimensions of the matrix.

Multiplying matrices is only possible when the dimensions agree, this means that:
If **A** is a p×q matrix, **B** is a r×s matrix, then matrix **A*****B** only exists when q=r.
The result of this matrix multiplication will be a p×s matrix.
If you try to multiply matrices that don't agree you'll receive an error message:
??? Error using ==> *
Inner matrix dimensions must agree.

Assignment 10

Given are the matrices $\mathbf{A} = \begin{pmatrix} 1 & -6 & 5 \\ -2 & 0 & 2 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 3 & -1 \\ 4 & 1 \end{pmatrix}$

Calculate (if possible): **AB, BA, A^T^B**

If the calculation is not possible, give an explanation in the form of MATLAB comment why the calculation is not possible according to you.

Also give the dimensions of the matrices **A** , **B** , and the dimensions of the results of the previous matrix multiplications (if they were possible).

Solving a system of linear equations

Matrix calculations are often used to solve a system of linear equations.
We can transform linear equations to a matrix notation:

$$\mathbf{A} \cdot \underline{\mathbf{x}} = \underline{\mathbf{b}}$$

A is the coefficient matrix
x is the column vector of the variables x_1 to x_n
b is the column vector of the right side of the equation.

Example: $\begin{cases} 3x_1 - 2x_2 = 12 \\ 2x_1 + 5x_2 = 17 \end{cases}$ can be written as **A** · **x** = **b**, with $\mathbf{A} = \begin{pmatrix} 3 & -2 \\ 2 & 5 \end{pmatrix}$, $\underline{\mathbf{b}} = \begin{pmatrix} 12 \\ 17 \end{pmatrix}$

The nice thing about this notation is that we can describe the solution for x with this as well.

```
A = [3 -2; 2 5]
b = [12; 17]       Beware: This is a column vector.
x = A\b
```

Note: this is not the same as `b/A`.

Assignment 11

Solve the following system of linear equations in MATLAB:

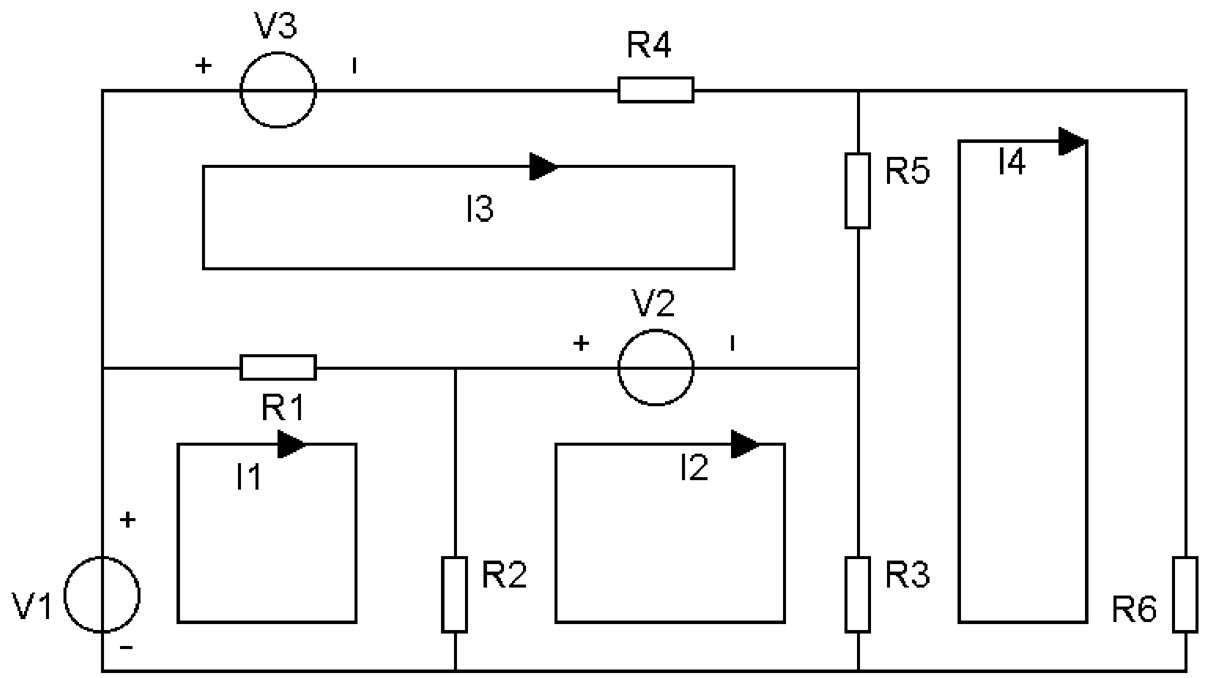
5x + 4y = 4
3x + 6y = 16

Assignment 12

Solve the following system of linear equations in MATLAB:

5x + 4y - 2z + 6w = 4
3y + 6w – 13 = z + x
6x + 12z – 2y + 16w = 20
42y + 2z – 4w = 6

Application to electrical engineering



In the electrical system given above, the voltages and resistances are known, but the currents need to be calculated.
This system can be translated into a system of linear equations, which can be solved by MATLAB.
You'll find the corresponding equations by choosing a current direction, see the image.
The voltage will be counted as positive if the current enters the + pole of the voltage supply, the voltage for a resistance will be counted positive in the chosen direction, this is Kirchhoff's Law. Note that resistances and voltage supplies can be used for each current if that current passes through the voltage supply and or resistance.
For example, applying Kirchhoff's voltage law to the top left loop of the system displayed above gives:

$$V3 + R4 \cdot I3 + R5 \cdot I3 - R5 \cdot I4 - V2 + R1 \cdot I3 - R1 \cdot I1 = 0$$

Assignment 13a

Write down the other equations in the form of a MATLAB comment.

You should get a total of four different equations, each corresponding to one of the loops.
Now you'll have to translate these in a way so that you can make a coefficient matrix where the variables are: I1, I2, I3 and I4.
This means you need to combine all the terms with I1 and all terms with I2 etc.
And all constants need to be assigned to the right side of the equation.

Assignment 13b

Work out each of the equations in this way. Then solve the system of equations in MATLAB with the following values:
V1=20, V2=12, V3=40, R1=18, R2=10, R3=16, R4=6, R5=15, R6=8.
What are the four currents you have found?

Logical Operators

Advice: prefer using .* over &

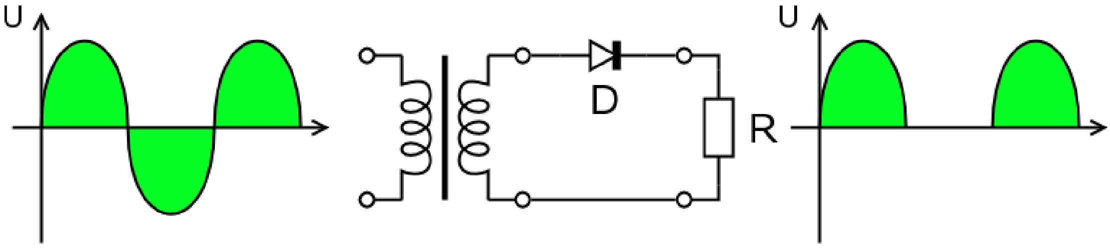
These operators can be used for when you want to plot a graph of a function with different definitions for different parts of the domain.
Please beware of the precedence rules and use brackets where necessary.
Example:

$$f(x)=\begin{cases}\sin(\pi x) & \text{if} \quad x<1\\ 1-x & \text{if} \quad 1\le x<2\\ x^2-4x+3 & \text{if} \quad x\ge 2\end{cases}$$

You'll get the graph of this function on the interval [-5,5] by the following:

```
clear
x = -5:0.1:5;
y1 = (x<1).*sin(pi*x);
y2 = (x>=1).*(x<2).*(1-x);
y3 = (x>=2).*(x.^2-4*x+3);
y = y1 + y2 + y3
plot(x,y)
```

You can also use this to draw the graph of a single rectified voltage function
Take: $U=10\sin(50t)$



The corresponding MATLAB commands to get the graph on the right (Ua) are as follows:

```
clear
t = linspace(0,0.2,100);
U = 10*sin(50*t);
Ua = (U>0).*U;
plot(t,Ua)
```

Intermezzo: the function diff

Open MATLABs help function browser (Shift F1) and then search for the help-information about the command **`diff`**.

For example:

```
x = [1,2,5,3,6,10];
dx = diff(x)
dx = 1   3   -2   3   4
```

Note that: x is a row of 6 numbers, diff(x) is a row of 5 numbers. Why?

Assignment 14

Make a vector for x of 100 numbers between 0 and 6 using the `linspace` command.
Then calculate the vector y = sin(x)
Calculate the vector dx = diff(x) as well as the vector dy = diff(y).

Calculate the vector y_derivative, being equal to $\frac{dy}{dx}$

This vector y_ derivative is a vector of only 99 numbers.
The graph of this can't just be plotted versus x, because x is a vector of 100 numbers.
That's why you should make a vector x_help that is equal to x, except for the last number.
This can be done for example with the commands: `n = length(x)-1;`

```
x_help = x(1:n);
```

Make two different plots screens with the graphs of y versus x and of y_ derivative versus x_help.
Make use of the `figure` command and add useful descriptions for the graphs and axes.
Is there something you notice in comparing both graphs?

Assignment 15

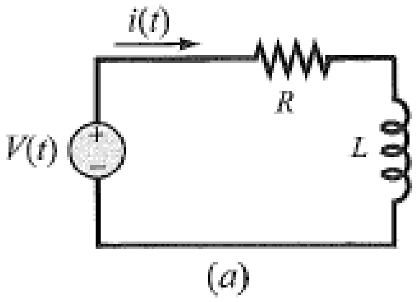
The place-time-function of a given object is given by the following vector:

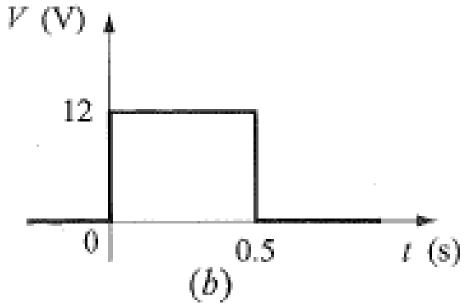
$$s(t) = \begin{cases} t & \text{if} \quad 0 \le t < 1 \\ t^2 - t + 1 & \text{if} \quad 1 \le \mathrm{t} < 2 \\ 3t - 3 & \text{if} \quad 2 \le t < 3 \\ -t^2 + 9t - 12 & \text{if} \quad 3 \le \mathrm{t} \le 8 \end{cases}$$

Make a vector t of 801 numbers from 0 to 8 using the `linspace` command.
Make a speed-time-function v(t) using differentiation like you also did in the previous assignment.
Make two different plot screens with one graph of s as function of t, and another graph of v as function of t
Make use of the `figure` command and add useful descriptions for the graphs and axes.

Assignment 16

See figure. R = 4 Ω , L =1.3 H





The current i as function of time is given by the following formula:

$$i = \frac{12}{R}\left(1 - e^{-\frac{Rt}{L}}\right) \quad \text{if } 0 \le \mathrm{t} \le 0.5$$

$$i = e^{-\frac{Rt}{L}} \frac{12}{R}\left(e^{\frac{R}{2L}} - 1\right) \quad \text{if } 0.5 < \mathrm{t} \le 2$$

Make a plot of current *i* as function of time *t* for *t* between 0 and 2 seconds with a step size of 0.01 add a useful description for the graphs and axes.