Laboratory 1: Introduction in MAPLE

Computation in Maple

Maple can make simple computations using the operators:
+ - addition
- - difference
* - multiplication
/ - division
^ - power
! - factorial
sqrt() - square root
exp() - the exponential function
ln() - the logarithm function
sin(), cos(), tan(), cot() - trigonometric functions sinus, cosine, tangent, cotangent

```
> 1+2-3;
```

$$0$$

```
> 2*3/7+3^2;
```

$$\frac{69}{7}$$

```
> sqrt(100);
```

$$10$$

```
> 3^2;
```

$$9$$

```
> sqrt(5);
```

$$\sqrt{5}$$

```
> sqrt(5.0);
```

$$2.236067977$$

When an integer is entered in the square root expression, MAPLE performs a *symbolic calculation*, if a decimal number is entered, MAPLE executes a numerical calculation with a precision of 10 digits. The **evalf** function returns the numeric value of the specified expression.

```
> evalf(sqrt(5));
```

$$2.236067977$$

```
> (1/5)^3;
```

$$\frac{1}{125}$$

```
> (0.2)^3;
```

$$0.008$$

```
> evalf((1/5)^3);
```

$$0.008000000000$$

```
>
```

Variables

You can assign values to variables using the command ": ="

```
> x:=1;y:=2;
```

$$x := 1$$
$$y := 2$$

```
> (x^2+y^2)/(2*x*y);
```

$$\frac{5}{4}$$

```
> evalf(%);
```

$$1.250000000$$

When we need the numerical evaluation of the previous expression we can use the command **evalf(%)**

We can use the Greek letters

```
> alpha, beta, gamma, Alpha, Beta, Gamma;
```

$$\alpha, \beta, \gamma, \mathrm{A}, \mathrm{B}, \Gamma$$

Remark: The expression **Pi** has the numerical value of this number, while the expression **pi** returns the respective greek letter

```
> pi; evalf(pi);
```

$$\pi$$
$$\pi$$

```
> Pi; evalf(Pi);
```

$$\pi$$
$$3.141592654$$

```
> alpha:=3*Pi/4;
```

$$\alpha := \frac{3}{4}\pi$$

```
> sin(alpha);cos(alpha);tan(alpha);cot(alpha);
```

$$\frac{1}{2}\sqrt{2}$$
$$-\frac{1}{2}\sqrt{2}$$
$$-1$$
$$-1$$

```
> ln(alpha),exp(alpha);
```

$$\ln\left(\frac{3}{4}\pi\right), e^{\frac{3}{4}\pi}$$

```
> evalf(%);
```

$$0.8570478133, 10.55072407$$

Functions and graphical representation

```
> restart;
```

The **restart** command clears the memory from the used values.

A single variable function can be defined as follows:

```
> f:=x->sin(x)/x;
```

$$f := x \rightarrow \frac{\sin(x)}{x}$$

```
> f(3*Pi/2),f(1.5);
```

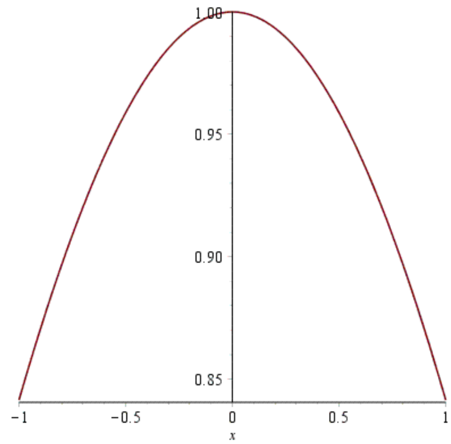
$$-\frac{2}{3\,\pi},\ 0.6649966577$$

```
> f(a+b);
```

$$\frac{\sin(a+b)}{a+b}$$

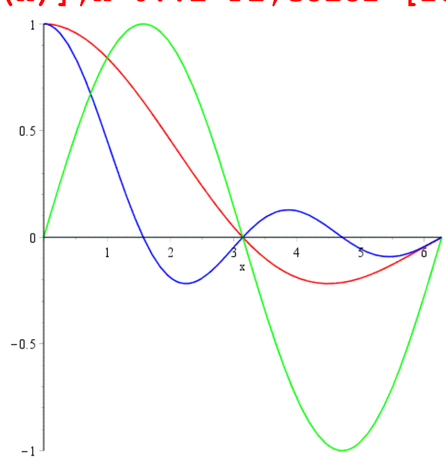
For graphical representation we need to load **plots** package using **with** command

```
> with(plots):
> plot(f(x),x=-1..1);
```



We can plot more than one function in the same window specifying the functions list $[f_1(x), f_2(x), f_3(x)]$ and the corresponding color list $[c_1, c_2, c_3]$:

```
> plot([f(x),f(2*x),sin(x)],x=0..2*Pi,color=[red,blue,green]);
```



If the functions list is larger, the operator **$** in the form (**expr$i=m..n**) can be used to generate it. It returns the list obtained by replacing **i** between **m** and **n** in the expression **expr**. For example, for the function $f_n(x) = \dfrac{x}{(1+x^2)^n}$ if we want to generate the graphical representations of the functions $f_1(x)$, ..., $f_{10}(x)$ first we construct the list and then the graphs using **plot** command:

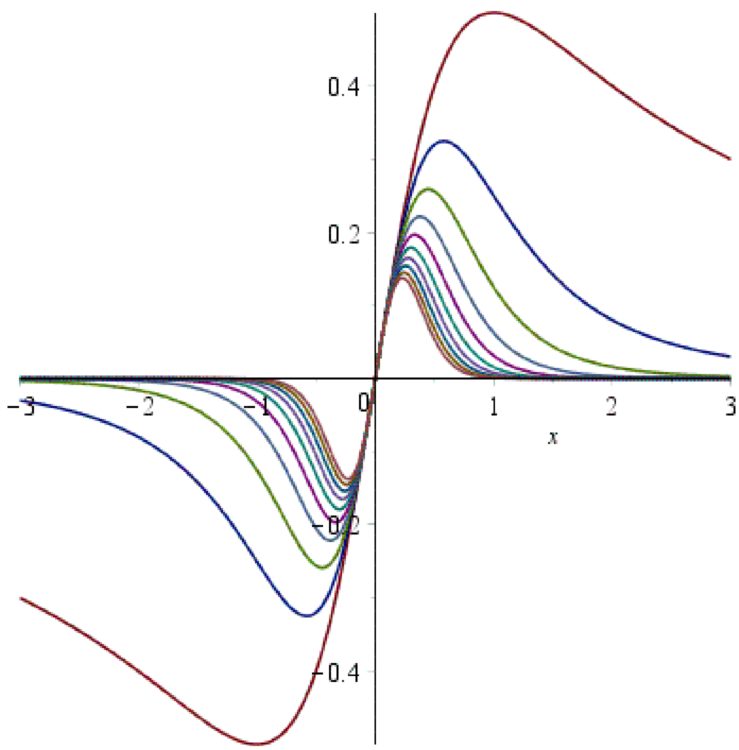
```
> f:=(x,n)->x/(1+x^2)^n;
```

$$f := (x, n) \rightarrow \frac{x}{(1+x^2)^n}$$

```
> list_f:=f(x,i)$i=1..10;
```

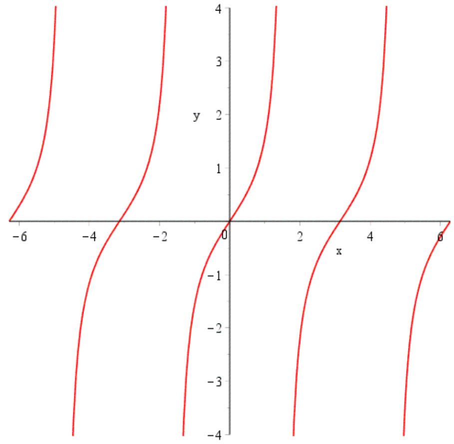
$$list\_f := \frac{x}{x^2+1}, \frac{x}{(x^2+1)^2}, \frac{x}{(x^2+1)^3}, \frac{x}{(x^2+1)^4}, \frac{x}{(x^2+1)^5}, \frac{x}{(x^2+1)^6}, \frac{x}{(x^2+1)^7}, \frac{x}{(x^2+1)^8}, \frac{x}{(x^2+1)^9}, \frac{x}{(x^2+1)^{10}}$$

```
> plot([list_f],x=-3..3);
```



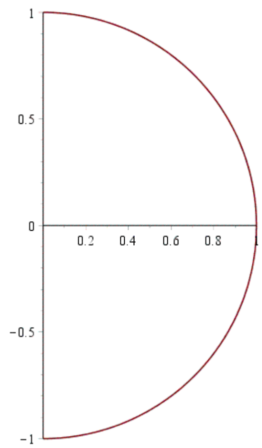
In the case of discontinuous points we need to use the option **discont = true**:

```
> plot(tan(x), x = -2*Pi..2*Pi, y = -4..4, discont = true);
```



If a curve is given in a parametric form (for example: $x(t) = \sin(t)$, $y(t) = \cos(t)$, $t = 0..\pi$;) we use the instruction:

```
> plot([sin(t),cos(t),t=0..Pi]);
```

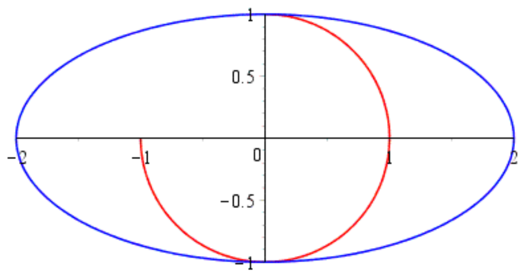


In this case, the `plot` function argument is a list of 3 components: `[x(t),y(t),t=a..b]` the first variable represents the **x** coordinate, the second **y** coordinate, the third variable the parameter range . In order to represent in the same graph several curves given in parametric form the `plot` function argument will be a list of curves list, for example for curves:

(C1): $x(t) = \sin(t)$
$y(t) = \cos(t)$, $t = 0..\pi$

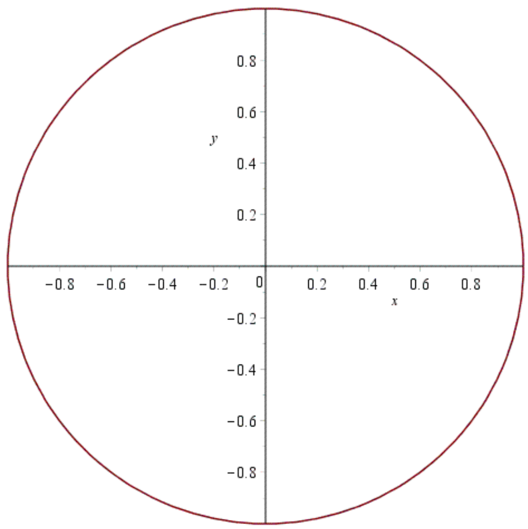
(C2): $x(t) = 2\sin(t)$, $t = 0..2\pi$
$y(t) = \cos(t)$, $t = 0..2\pi$

```
> plot([[sin(t),cos(t),t=0..3/2*Pi], [2*sin(t),cos(t),t=0..2*Pi]],
color=[red,blue], scaling=constrained);
```



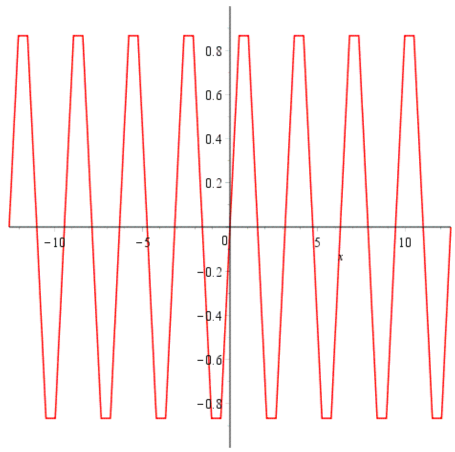
In the case of a curve given by the implicit equation we use the instruction **implicitplot**:

```
> implicitplot(x^2+y^2=1,x=-1..1,y=-1..1);
```



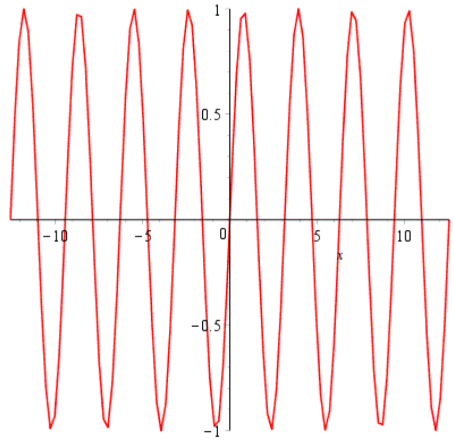
If we need to visualize the parameter dependence of a function we can use the command **animate** (right click on the image, select *Animation* and *Play*)

```
> animate(sin(x*t),x=-4*Pi..4*Pi,t=0..2,color=red);
```



If we nedd a higher precision we specify the number of points and the number of frames:

```
> animate(sin(x*t), x=-4*Pi..4*Pi, t=0..2, color=red, numpoints=100,
frames=100);
```



Limits, Derivates, Integrals

```
> restart:
```

The sequences limits and functions limits can be obtained using the **limit** command:

```
> limit(1/n,n=infinity);
```

$$0$$

```
> limit(sin(x)/x,x=0);
```

$$1$$

```
> limit(exp(x), x=infinity);
```

$$\infty$$

```
> limit(1/x, x=0, real);
```

$$\mathit{undefined}$$

The derivation of the functions can be made in two ways: using **diff** command or using the derivation operator **D** :

```
> f:=x->exp(x^2)+3;
```

$$f := x \to \mathrm{e}^{x^2} + 3$$

The **diff** command executes the derivation of the given expression with respect to the specified variable. The derivation operator **D** returns the derivate as a function.

```
> diff(f(x),x);
```

$$2\,x\,\mathrm{e}^{x^2}$$

the second order derivate is given by

```
> diff(f(x),x,x);
```

$$2\,\mathrm{e}^{x^2} + 4\,x^2\,\mathrm{e}^{x^2}$$

also we can use the option **x$n** to get n-order derivative

```
> diff(f(x),x$2);
```

$$2\,\mathrm{e}^{x^2} + 4\,x^2\,\mathrm{e}^{x^2}$$

```
> diff(f(x),x$3);
```

$$12\,x\,\mathrm{e}^{x^2} + 8\,x^3\,\mathrm{e}^{x^2}$$

Using the derivation operator:

```
> D(f)(x);
```

$$2\,x\,e^{x^2}$$

```
> D(f)(1);
```

$$2\,e$$

```
> (D@D)(f)(x);
```

$$2\,e^{x^2} + 4\,x^2\,e^{x^2}$$

```
> (D@D)(f)(1);
```

$$6\,e$$

```
> (D@@2)(f)(x);
```

$$2\,e^{x^2} + 4\,x^2\,e^{x^2}$$

```
> (D@D@D)(f)(x);
```

$$12\,x\,e^{x^2} + 8\,x^3\,e^{x^2}$$

```
> (D@@3)(f)(x);
```

$$12\,x\,e^{x^2} + 8\,x^3\,e^{x^2}$$

The computation of the indefinite integral or antiderivative can be made using the **int** command:

```
> int(cos(x),x);
```

$$\sin(x)$$

if the integration limits are specified **x=a..b** then the definite integral value is obtained:

```
> int(cos(x),x=0..Pi);
```

$$0$$

```
> int(1/x,x=1..infinity);
```

$$\infty$$

Not always MAPLE can compute the definite integral value:

```
> int( sin( sqrt(1 - x^3) ), x = 0..1 );
```

$$\int_0^1 \sin\left(\sqrt{-x^3+1}\right)\,dx$$

but it can obtain an approximate value using numerical approximation methods:

```
> evalf( int( sin( sqrt(1 - x^3) ), x = 0..1 ) );
```

$$0.7315380065$$

Algebraic equations and systems of algebraic equations. Linear algebra

Algebraic equations and systems of algebraic equations can be solved using the **solve** command:

```
> restart:
> solve( x^2 + 3*x + 2=0 );
```

$$-1,\ -2$$

```
> solve( x^2 + x + 1=0 );
```

$$-\frac{1}{2} + \frac{1}{2}\,I\,\sqrt{3},\ -\frac{1}{2} - \frac{1}{2}\,I\,\sqrt{3}$$

The **fsolve** command solves the equations using numerical methods to obtain approximating solutions:

```
> solve( x^3 + x = 27 );
```

$$\frac{1}{6}\left(2916+12\sqrt{59061}\right)^{1/3}-\frac{2}{\left(2916+12\sqrt{59061}\right)^{1/3}},$$
$$-\frac{1}{12}\left(2916+12\sqrt{59061}\right)^{1/3}+\frac{1}{\left(2916+12\sqrt{59061}\right)^{1/3}}$$
$$+\frac{1}{2}\,I\sqrt{3}\left(\frac{1}{6}\left(2916+12\sqrt{59061}\right)^{1/3}\right.$$
$$\left.+\frac{2}{\left(2916+12\sqrt{59061}\right)^{1/3}}\right), -\frac{1}{12}\left(2916+12\sqrt{59061}\right)^{1/3}$$
$$+\frac{1}{\left(2916+12\sqrt{59061}\right)^{1/3}}-\frac{1}{2}\,I\sqrt{3}\left(\frac{1}{6}\left(2916\right.\right.$$
$$\left.\left.+12\sqrt{59061}\right)^{1/3}+\frac{2}{\left(2916+12\sqrt{59061}\right)^{1/3}}\right)$$

```
> fsolve( x^3 + x = 27 );
```

$$2.888941572$$

```
> solve( tan(x) - x = 2 );
```

$$RootOf(-\tan(\_Z)+\_Z+2)$$

```
> fsolve( tan(x) - x = 2 );
```

$$1.274392662$$

In the case of the equation systems the same command is used, the equations are placed between the braces

```
> solve({x+2*y=1,x-y=3},{x,y});
```

$$\left\{x=\frac{7}{3},\, y=-\frac{2}{3}\right\}$$

Linear Algebra

In the case of operations with vectors and matrices the linear algebra package `linalg` must be loaded:

```
> with(linalg):
```

Vectors can be defined as follows:

```
> v := vector( [1, -1] );
```

$$v := \begin{bmatrix} 1 & -1 \end{bmatrix}$$

```
> v[1]; v[2];
```

$$1$$
$$-1$$

```
> print( v );
```

$$\begin{bmatrix} 1 & -1 \end{bmatrix}$$

```
> w := vector( [1, 1] );
```

$$w := \begin{bmatrix} 1 & 1 \end{bmatrix}$$

```
> evalm(v+w);
```

$$\begin{bmatrix} 2 & 0 \end{bmatrix}$$

```
> dotprod( v, w ); #   scalar product
```

$$0$$

The matrices are defined using the command `matrix`:

```
> A := matrix([[1,0],[3,2]]);
```

$$A := \begin{bmatrix} 1 & 0 \\ 3 & 2 \end{bmatrix}$$

```
> B := matrix([ [1, 0], [2, 1] ]);
```

$$B := \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

We have the following operations for matrices:

```
> evalm(A + B);  # matrices summation
```

$$\begin{bmatrix} 2 & 0 \\ 5 & 3 \end{bmatrix}$$

```
> evalm(2*A);  #   multiplying a matrix by a scalar
```

$$\begin{bmatrix} 2 & 0 \\ 6 & 4 \end{bmatrix}$$

```
> evalm(A &* B);  # matrices multiplication
```

$$\begin{bmatrix} 1 & 0 \\ 7 & 2 \end{bmatrix}$$

```
> evalm(A &* (v+w));  # multiplying a matrix by a vector
```

$$\begin{bmatrix} 2 & 6 \end{bmatrix}$$

```
> det(A);  # determinant of A
```

$$2$$

```
> evalm(A^(-1));  #   calculate the inverse matrix of A
```

$$\begin{bmatrix} 1 & 0 \\ -\frac{3}{2} & \frac{1}{2} \end{bmatrix}$$

```
> eigenvals(A);  # calculate the eigenvalues of A
```

$$1, 2$$

```
> eigenvects(A);  # calculate the eigenvectors of A
```

$$[2, 1, \{[\,0\ \ 1\,]\}], [1, 1, \{[\,1\ \ -3\,]\}]$$