Much of the following is taken from the on-line documentation for the symbolic algebra toolbox in MATLAB

MATLAB incorporates commands from the symbolic language MuPAD. One works with *symbolic variables* to perform these commands. Thus we need to create such variables:

```
>> syms x y
```

makes symbolic variables `x` and `y` . We can then manipulate these in various ways.

Calculus: differentiation, integration, limits and Taylor series

---

```
>> syms x y n
>> diff(sin(x))                % gives cos(x)
>> diff(sin(x),2)              % gives -sin(x)
>> diff(sin(x+2*y),y)          % gives 2*cos(x+2y) (partial derivative wrt y)
>> diff(x^n)                   % gives n*x^(n-1)
>> pretty(diff(x^n))           % gives pretty output

>> simplify(cos(x)^2-sin(x)^2) % gives cos(2x)
>> [r,h]=simple(cos(x)+i*sin(x)) % gives 'shortest' version of the expression,
                                 % i.e. r=exp(ix), with method used in h
>> int(sin(x))              % gives -cos(x)

>> taylor(exp(x),6)         % gives 6 terms in the Taylor expansion of e^x
                            % around 0 (the default)
>> taylor(exp(x),6,1)       % expands around one instead of zero

>> limit(sin(x)/x,x,0,'right') % the limit, x->0+ sin(x)/x
```

We can plot these results directly, e.g.

```
ezplot(diff(sin(x)),[0 2])  % plots the derivative on the interval [0,2]
```

Solving equations

```
>> syms a b c d e f x
>> quad=a*x^2+b*x+c          % set up a quadratic equation
>> solns=solve(quad,x)       % find the two solutions
>> a=1;b=1;c=1;eval(solns)   % find the two solutions of x^2+x+1=0
>> syms a b c                % reset the symbolic variables
>> cubic=a*x^3+b*x^2+c*x+d   % set up a cubic equation (with a=b=c=1)
>> solns=solve(cubic,x)      % find the general solution of the cubic
>> a=1;b=0;c=-1;d=0;eval(solns); % find the roots of x^3-x=0
>> factor(x^3-x)             % factorize x^3-x
>> syms a b c d              % reset symbolic variables
>> quintic=a*x^5+b*x^4+c*x^3+d*x^2+e % set up a quintic
>> solve(quintic,x)          % find the general solution?
>> dsolve('D2x=-x')          % Solve ODE: gives C3*cos(t)+C4*sin(t)
>> syms a b c d              % reset symbolic variables
>> eig([a b; c d])           % find eigenvalues of general 2x2 matrix
>> inv([a b; c d])           % inverse of 2x2 matrix
```

Precision

There are three different kinds of arithmetic operations in this toolbox:
Numeric: MATLAB floating-point arithmetic
Rational: MuPAD's exact symbolic arithmetic
VPA: MuPAD's variable-precision arithmetic

```
>> format long
>> 1/2+1/3                % MATLAB floating point: gives 0.83333333333333
>> sym(1/2)+1/3           % uses symbolic computation to give 5/6

>> digits(25)
>> vpa('1/2+1/3')         % uses 25 digits to give .83333333333333333333333333
>> vpa(pi,780)            % shows six consecutive 9's near digit 770 in the
                          % decimal expansion of pi
>> sin(vpa(pi,780))       % evaluates sin(pi) to 780 significant digits
```

MATLAB uses 'format rat' for approximating reals as the ratio of two small(ish) integers:

```
format rat
>> 1/2+1/3         %  returns 5/6
>> 1/2+1/3+1e-8     % returns 5/6 (approximate rational)
>> 1/2+1/3+1e-6    % returns 138891/166669
```