Symbolic calculations in Matlab:

You must first give MATLAB a list of the variable and function names that will appear in the symbolic expressions you will be working with. Do this using the command `syms`

```
>>syms x y z a;
```

This command basically tells the software that you will be using the symbols x, y, z, and f in forthcoming symbolic expressions.

One thing that is often useful is to define a function of the symbolic variables you have defined. Note that you do not need to define the function name using `syms` since MATLAB understands that when you define a function using symbolic variables you intend that function to be symbolic. An example function definition is

```
>>f = x^2 + sin(y);
```

If you would like to substitute numerical values into a symbolic function, use the function `subs` as in the following way

```
>>subs(f,{x,y},{1,pi/2});
```

Doing symbolic calculus is one of the more useful things that MATLAB can do for us. To integrate a function, use `int` as follows

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

to get the indefinite integral, or

```
>>int(f,x,1,a);
```

to get the definite integral from 1 to a. Note that the limits of integration can be either numeric or symbolic. To do double or triple integrals you need to manually use the `int` function multiple times.

You similarly can differentiate functions, though you can take derivatives of order higher than one in a single step. For example,

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

is the second derivative of f with respect to x.

Taking limits is also possible, using the function `limit` as follows

```
>>limit(f,x,0);
```

to take the limit of f as x goes to zero. Note that MATLAB recognizes `inf` as a symbolic representation infinity, which can be freely used in symbolic expressions. For example, if you want to take the limit as a variable approaches infinity you can use `inf` as the third input of `limit`.

Solving systems of equations is a very useful feature of MATLAB. Let's first define two equations

```
>>f1 = x + y;
>>f2 = x + 2*y + 1;
```

We can solve the system of equations $x + y = 0$ and $x + 2y = -1$ using the following command

```
>>xy_solution = solve(f1,f2,'x,y');
```

which also assigns the solution to the new symbolic variable `xy_solution`. This variable is actually something called a structure variable. To access the solutions for the individual variables, you must make assignments like this

```
>>x_solution = xy_solution.x;
>>y_solution = xy_solution.y;
```

Sometimes the results of the above operations will be quite complicated. Luckily `MATLAB` has the ability to simplify symbolic expressions. The two commands that do simplification are `simplify` and `simple`, both using the same syntax:

```
>>simple(f);
```

Not much to simplify in this case, but try it on a more complicated function and you will see the results.

Finally, if you ever need help using these functions, simply type `help` followed by the function name, for example,

```
>>help sym/int;
```

in which the sym/ part just ensures that you get help on the symbolic function when there exist symbolic and numerical functions with the same name. This occurs, for example, with `diff`. Try getting help on `diff` with and without the sym/ and see what you get. You may also simply type

```
>>help symbolic;
```

to get a list of all the other symbolic operations that are possible in `MATLAB`.