Basics

**MATRICES AND VECTORS**

- **[a:s:b]** — Vector from a to b in steps of length s.
- **linspace(a,b,n)** — Vector of n even-spaced numbers between a and b.
- **pi, exp(1)** — Values of π and e (3.14159... and 2.71828...).
- **clear, clear(x)** — Clear the values of all variables or only of x.
- **whos, who** — Describe or list the defined variables.

**ACCESSING ELEMENTS**

- **x(i)** — Element i-th of x.
- **x(i:j)** — Elements from the i-th to the j-th of vector x.
- **x(i:end)** — Elements from the i-th to the last one of vector x.
- **x(:)** — All the elements of x.

**FUNCTIONS AND UTILITIES**

- **sqrt** — Square root. **sqrt(x)**: square root of each element of x.
- **sin, cos, tan** — Sine, cosine, and tangent. For example, **sin(x)**, sine of each element of x.
- **asin, acos, atan** — Inverse functions of the previous ones.
- **exp** — Exponential. **exp(x)**: exponential of each element of x.
- **log, log10** — Natural logarithm (base e) and decimal (base 10). **log(x)**: natural logarithm of each element of x.

**FUNCTION DEFINITION**

- **f = @(x) sin(x) - exp(x) .* x.^2** — Defines the one-variable function f whose value is sin(x) - e^x * x^2 (always use .* , / and ^).
- **f = @(x,y,z) x .* y - z.^2 .* y ./ x** — Defines the three-variable function f whose value is x*y - z^2*y/x. (always use .* , / and ^).

**PLOTTING**

- **clf** — Clears the graphical window.
- **hold on/hold off** — Allows (hold on) or prevents (hold off) plotting on the same graph.
- **plot(y)** — If y is a vector, plot the sequence of values of y.
- **plot(x,y)** — If x and y are vectors of the same length, plot the graph of x against y given by those values.
- **plot(x,f(x))** — If x is a vector and f a function, plot the points (x,f(x)).
- **fplot(f,[x0 x1 y0 y1])** — If f is a function, plot f on the rectangle [x0, x1]×[y0, y1].
- **axis([x0 x1 y0 y1])** — Replot a graph on the rectangle [x0, x1]×[y0, y1].
- **xlim([x0 x1]), ylim([y0 y1])** — Replot a graph with x varying between x0 and x1 (or with y between y0 and y1).
Programming

`return` — Return immediately from a function.

```matlab
if(COND)
    ...
end
```
— If `COND` holds, perform the instructions . . .

```matlab
if(COND)
    ...
else
    ...
end
```
— If `COND` holds, do . . .1; if it does not, do . . .2.

```matlab
for k = a:s:b
    ...
end
```
— Loop from `a` to `b` in steps of `s`, in the variable `k`.

```matlab
for k = v
    ...
end
```
— If `v` is a vector, loop in the variable `k` traversing the elements of `v`.

```matlab
while(COND)
    ...
end
```
— Repeat the instructions in . . . while `COND` holds.

**Advanced matrix operations**

```matlab
function [x1 x2] = nombreF(a,b,c)
    ...
end
```
Defines function `nombreF`, which as, as input parameters, `a`, `b`, `c` and returns values `x1` `y` `x2`.

**Polynomial operations**

Assuming `v` is a vector `v=[v_1,...,v_n]`.

```matlab
polyval(v,x) — Value of the polynomial expression \(v_1 x^{n-1} + \ldots + v_{n-1} x + v_n\) (the degree decreases).
```

```matlab
roots(v) — Roots of the polynomial \(v_1 x^{n-1} + \ldots + v_{n-1} x + v_n\).
```

```matlab
polyderiv(v) — Derivative (as a vector expression) of the polynomial \(v_1 x^{n-1} + \ldots + v_{n-1} x + v_n\), that is, \([v_1/(n-1),\ldots,v_{n-2}/2,v_{n-1}]\).
```

**Numerical operations**

Assuming `v` is a vector `v=[v_1,...,v_n]`.

```matlab
sum(v), prod(v) — Sum and product of the elements of `v`.
```

```matlab
max(v), min(v) — Maximum and minimum of the elements of `v`.
```

```matlab
mean(v), var(v) — Mean and variance of the elements of `v`.
```

```matlab
diff(v) — Successive differences of `v`. That is, \([v_2 - v_1, v_3 - v_2, \ldots, v_n - v_{n-1}]\).
```

```matlab
diff(v,k) — Successive `k`-differences of `v`. One has that \(\text{length}(\text{diff}(v)) = \text{length}(v)-1\).
```

**Interpolation**

```matlab
interp1(x,y) — Given `x` and `y` two equally-sized vectors of length `n`, returns the interpolation polynomial (Lagrange) for the points `(x,y)`. It is returned as a vector `[v_1,...,v_n]` (interpreted as a polynomial of degree `n-1`).
```

```matlab
polyfit(x,y,n) — Polynomial (as a vector) of degree `n` which minimizes the quadratic error with respect to the cloud of points given by `x` and `y` (equally-sized vectors).
```

```matlab
spline(x,y) — Cubic spline interpolation function for the points `(x,y)` given by `x` and `y` (equally-sized vectors). It returns a piecewise polynomial which can be evaluated using `ppval`.
```