

# Part V

## Appendices

©Copyright, Todd Young and Martin Mohlenkamp, Department of Mathematics, Ohio University, 2025

# Appendix A

## Glossary of Matlab Commands

### Mathematical Operations

- + Addition. Type `help plus` for information.
- Subtraction. Type `help minus` for information.
- \* Scalar or matrix multiplication. Type `help mtimes` for information.
- / Scalar or right matrix division. Type `help slash` for information.  
For matrices, the command `A/B` is equivalent to `A*inv(B)`.
- ^ Scalar or matrix powers. Type `help mpower` for information.
- .\* Element by element multiplication. Type `help times` for information.
- .^ Element by element exponentiation. Type `help power` for information.
- ./ Element by element division.

### Built-in Mathematical Constants

- `eps` Machine epsilon, i.e. approximately the computer's floating point roundoff error.
- `i`  $\sqrt{-1}$ .
- `Inf`  $\infty$ .
- `NaN` Not a number. Indicates an invalid operation such as `0/0`.
- `pi`  $\pi = 3.14159\dots$

### Built-in Mathematical Functions

- `abs(x)` Absolute value  $|x|$ .
- `acos(x)` Inverse cosine  $\arccos x$ .
- `asin(x)` Inverse sine  $\arcsin x$ .

<code>atan(x)</code>	Inverse tangent $\arctan x$ .
<code>cos(x)</code>	Cosine $\cos x$ .
<code>cosh(x)</code>	Hyperbolic cosine $\cosh x$ .
<code>cot(x)</code>	Cotangent $\cot x$ .
<code>exp(x)</code>	Exponential function $e^x = \exp x$ .
<code>log(x)</code>	Natural logarithm $\ln x = \log_e x$ .
<code>sec(x)</code>	Secant $\sec x$ .
<code>sin(x)</code>	Sine $\sin x$ .
<code>sinh(x)</code>	Hyperbolic sine $\sinh x$ .
<code>sqrt(x)</code>	Square root $\sqrt{x}$ .
<code>tan(x)</code>	Tangent $\tan x$ .
<code>tanh(x)</code>	Hyperbolic tangent $\tanh x$ .
<code>max</code>	Computes maximum of the rows of a matrix.
<code>mean</code>	Computes the average of the rows of a matrix.
<code>min</code>	Computes the minimum of the rows of a matrix.

### Built-in Numerical Mathematical Operations

<code>fzero</code>	Tries to find a zero of the specified function near a starting point or on a specified interval.
<code>inline</code>	Define a function in the command window.
<code>ode113</code>	Numerical multiple step ODE solver.
<code>ode45</code>	Runga-Kutta 45 numerical ODE solver.
<code>quad</code>	Numerical integration using an adaptive Simpson's rule.
<code>dblquad</code>	Double integration.
<code>triplequad</code>	Triple integration.

### Built-in Symbolic Mathematical Operations

<code>collect</code>	Collects powers of the specified variable in a given symbolic expression.
<code>compose</code>	Composition of symbolic functions.
<code>diff</code>	Symbolic differentiation.
<code>double</code>	Displays double-precision representation of a symbolic expression.
<code>dsolve</code>	Symbolic ODE solver.
<code>expand</code>	Expands an algebraic expression.
<code>factor</code>	Factor a polynomial.

<code>int</code>	Symbolic integration; either definite or indefinite.
<code>limit</code>	Finds two-sided limit, if it exists.
<code>pretty</code>	Displays a symbolic expression in a nice format.
<code>simple</code>	Simplifies a symbolic expression.
<code>subs</code>	Substitutes for parts a a symbolic expression.
<code>sym</code> or <code>syms</code>	Create symbolic variables.
<code>symsum</code>	Performs a symbolic summation, possibly with infinitely many entries.
<code>taylor</code>	Gives a Taylor polynomial approximation of a given order at a specified point.

## Graphics Commands

<code>contour</code>	Plots level curves of a function of two variables.
<code>contourf</code>	Filled contour plot.
<code>ezcontour</code>	Easy contour plot.
<code>loglog</code>	Creates a log-log plot.
<code>mesh</code>	Draws a mesh surface.
<code>meshgrid</code>	Creates arrays that can be used as inputs in graphics commands such as <code>contour</code> , <code>mesh</code> , <code>quiver</code> , and <code>surf</code> .
<code>ezmesh</code>	Easy mesh surface plot.
<code>plot</code>	Plots data vectors.
<code>ezplot</code>	Easy plot for symbolic functions.
<code>plot3</code>	Plots curves in 3-D.
<code>polar</code>	Plots in polar coordinates.
<code>quiver</code>	Plots a vector field.
<code>semilogy</code>	Semilog plot, with logarithmic scale along the vertical direction.
<code>surf</code>	Solid surface plot.
<code>trimesh</code>	Plot based on a triangulation
<code>trisurf</code>	Surface plot based on a triangulation

## Special Matlab Commands

<code>:</code>	Range operator, used for defining vectors and in loops. Type <code>help colon</code> for information.
<code>;</code>	Suppresses output. Also separates rows of a matrix.
<code>=</code>	Assigns the variable on the left hand side the value of the right hand side.

**ans** The value of the most recent unassigned.  
**cd** Change directory.  
**clear** Clears all values and definitions of variables and functions. You may also use to clear only specified variables.  
**diary** Writes a transcript of a MATLAB session to a file.  
**dir** Lists the contents in the current working directory. Same as **ls**.  
**help**  
**inline** Define an inline function.  
**format** Specifies output format, e.g. `> format long`.  
**load** Load variables from a file.  
**save** Saves workspace variables to a file.

## Matlab Programming

**==** Is equal?  
**~=** Is not equal?  
**<** Less than?  
**>** Greater than?  
**<=** Less than or equal?  
**break** Breaks out of a **for** or **while** loop.  
**end** Terminates an **if**, **for** or **while** statement.  
**else** Alternative in an **if** statement.  
**error** Displays an error message and ends execution of a program.  
**for** Repeats a block of commands a specified number of times.  
**function** First word in a function program.  
**if** Checks a condition before executing a block of statements.  
**return** Terminates execution of a program.  
**warning** Displays a warning message.  
**while** Repeats a block of commands as long as a condition is true.

## Commands for Matrices and Linear Algebra

### Matrix arithmetic:

A = [ 1 3 -2 5 ; -1 -1 5 4 ; 0 1 -9 0] ..... Manually enter a matrix.

`u = [ 1 2 3 4]'`

`A*u`

`B = [3 2 1; 7 6 5; 4 3 2]`

`B*A` .....multiply  $B$  times  $A$ .

`2*A` .....multiply a matrix by a scalar.

`A + A` .....add matrices.

`A + 3` .....add a number to every entry of a matrix.

`B.*B` .....component-wise multiplication.

`B.^3` .....component-wise exponentiation.

### Special matrices:

`I = eye(3)` .....identity matrix

`D = ones(5,5)`

`O = zeros(10,10)`

`C = rand(5,5)` .....random matrix with uniform distribution in  $[0,1]$ .

`C = randn(5,5)` .....random matrix with normal distribution.

`hilb(6)`

`pascal(5)`

### General matrix commands:

`size(C)` .....gives the dimensions ( $m \times n$ ) of  $A$ .

`norm(C)` .....gives the norm of the matrix.

`det(C)` .....the determinant of the matrix.

`max(C)` .....the maximum of each row.

`min(C)` .....the minimum in each row.

`sum(C)` .....sums each row.

`mean(C)` .....the average of each row.

`diag(C)` .....just the diagonal elements.

`inv(C)` .....inverse of the matrix.

### Matrix decompositions:

`[L U P] = lu(C)`

`[Q R] = qr(C)`

`[U S V] = svd(C)` .....singular value decomposition.