MATLAB commands in numerical Python

Contributor: Gary Ruben

The idea of this document (and the corresponding XML instance) is to provide a quick reference[1] for switching to open-source mathematical computation environments for computer algebra, numeric processing and data visualisation. Examples of well known systems are MATLAB, IDL, R, SPlus, with their open-source counterparts Octave, Scilab, FreeMat, Python (NumPy and matplotlib modules), and Gnuplot. Or CAS tools like Mathematica, Maple, MuPAD, with Axiom and Maxima as open alternatives.

Where Octave and Scilab commands are omitted, expect Matlab compatibility, and similarly where non given use the generic command.

Time-stamp: 2007-06-15T17:27:54 vidar

1 Help

	Browse help interactively
MATLAB	`doc`
Octave	`help -i % browse with Info`
Scilab	`help`
R	`help.start()`
Python	`help()`
gnuplot	`help` or `?`
IDL	`?`
Axiom	`)hd`

	Help on using help
MATLAB	`help help` or `doc doc`
R	`help()`
Python	`help`
IDL	`?help`
Axiom	`)help ?`

	Help for a function
MATLAB	`help plot`
R	`help(plot)` or `?plot`
Python	`help(plot)` or `?plot`
gnuplot	`help plot` or `?plot`
IDL	`?plot` or `man,'plot`
Maxima	`describe(keyword)$`
Maple	`?keyword`
Mathematica	`?keyword`
MuPAD	`?keyword`

	Help for a toolbox/library package
MATLAB	`help splines` or `doc splines`
R	`help(package='splines')`
Python	`help(pylab)`

	Demonstration examples
MATLAB	`demo`
Scilab	`demoplay();`
R	`demo()`
IDL	`demo`

	Example using a function
R	`example(plot)`
Maxima	`example(factor);`

1.1 Searching available documentation

	Search help files
MATLAB	`lookfor plot`
R	`help.search('plot')`

	Find objects by partial name
R	`apropos('plot')`
Scilab	`apropos plot`
Axiom	`)what operations pattern`
Maxima	`describe(pattern)$`
Maple	`?keyword`
Mathematica	`?*pattern*`
MuPAD	`?*pattern*`

[1] References: Hankin, Robin. *R for Octave users* (2001), available from http://cran.r-project.org/doc/contrib/R-and-octave-2.txt (accessed 2005.07.24); Martelli, Alex. *Python in a Nutshell* (O'Reilly, 2003); Oliphant, Travis. *Guide to NumPy* (Trelgol, 2006); Hunter, John. *The Matplotlib User's Guide* (2005), available from http://matplotlib.sf.net/ (accessed 2005.07.31); Langtangen, Hans Petter. *Python Scripting for Computational Science* (Springer, 2004); Ascher et al.: *Numeric Python manual* (2001), available from http://numeric.scipy.org/numpy.pdf (accessed 2005.06.25); Moler, Cleve. *Numerical Computing with MATLAB* (MathWorks, 2004), available from http://www.mathworks.com/moler/ (accessed 2005.03.10); Eaton, John W. *Octave Quick Reference* (1996); Merrit, Ethan. *Demo scripts for gnuplot version 4.0* (2004), available from http://gnuplot.sourceforge.net/demo/ (accessed 2005.07.24); Woo, Alex. *Gnuplot Quick Reference* (2004), available from http://www.gnuplot.info/docs/gpcard.pdf (accessed 2005.07.14); Venables & Smith: *An Introduction to R* (2005), available from http://cran.r-project.org/doc/manuals/R-intro.pdf (accessed 2005.07.25); Short, Tom. *R reference card* (2005), available from http://www.rpad.org/Rpad/R-refcard.pdf (accessed 2005.07.24); Greenfield, Jedrzejewski & Laidler. *Using Python for Interactive Data Analysis* (2005), pp.125–134, available from http://stsdas.stsci.edu/perry/pydatatut.pdf (accessed 2005.07.29); Brisson, Eric. *Using IDL to Manipulate and Visualize Scientific Data*, available from http://scv.bu.edu/Tutorials/IDL/ (accessed 2005.07.31); Wester, Michael (ed). *Computer Algebra Systems: A Practical Guide* (1999), available from http://www.math.unm.edu/~wester/cas_review.html (accessed 2005.08.14).

	List available packages
MATLAB	`help`
R	`library()`
Python	`help(); modules [Numeric]`

	Locate functions
MATLAB	`which plot`
Scilab	`whereis plot`
R	`find(plot)`
Python	`help(plot)`

	List available methods for a function
R	`methods(plot)`

1.2 Using interactively

	Start session
Octave	`octave -q`
R	`Rgui`
Python	`ipython -pylab`
IDL	`idlde`
bc	`bc -lq`
gnuplot	`pgnuplot`

	Auto completion
Octave	`TAB` or `M-?`
Scilab	`! // commands in history`
Python	`TAB`

	Run code from file
MATLAB	`foo(.m)`
Scilab	`exec('foo.sce')`
R	`source('foo.R')`
Python	`execfile('foo.py')` or `run foo.py`
gnuplot	`load 'foo.gp'`
IDL	`@"foo.idlbatch"` or `.run 'foo.pro'`
Maxima	`batch("foo.mc")`

	Command history
Octave	`history`
Scilab	`gethistory`
R	`history()`
Python	`hist -n`
IDL	`help,/rec`
Axiom	`)history )show`

	Save command history
MATLAB	`diary on [..] diary off`
Scilab	`diary('session.txt') [..] diary(0)`
R	`savehistory(file=".Rhistory")`
IDL	`journal,'IDLhistory'`
Axiom	`)hist )write foo.input`

	End session
MATLAB	`exit` or `quit`
R	`q(save='no')`

Python	`CTRL-D`
	`CTRL-Z # windows`
	`sys.exit()`

gnuplot	`exit` or `quit`
IDL	`exit` or `CTRL-D`
Axiom	`)quit`
Maxima	`quit();`
Maple	`quit`
Mathematica	`Quit[]`
MuPAD	`quit`
Derive	`[Quit]`
REDUCE	`quit;`
bc	`quit`

2 Operators

	Help on operator syntax
MATLAB	`help -`
Scilab	`help symbols`
R	`help(Syntax)`

2.1 Arithmetic operators

	Assignment; defining a number
MATLAB	`a=1; b=2;`
R	`a<-1; b<-2`
Python	`a=1; b=1`
IDL	`a=1 & b=1`
bc	`a=1; b=1`

	Addition
Generic	`a + b`
MATLAB	`a + b`
R	`a + b`
Python	`a + b` or `add(a,b)`
gnuplot	`a + b`
IDL	`a + b`
bc	`a + b`

	Subtraction
Generic	`a - b`
MATLAB	`a - b`
R	`a - b`
Python	`a - b` or `subtract(a,b)`
gnuplot	`a - b`
IDL	`a - b`
bc	`a - b`

	Multiplication
Generic	`a * b`
MATLAB	`a * b`
R	`a * b`
Python	`a * b` or `multiply(a,b)`
gnuplot	`a * b`
IDL	`a * b`
bc	`a * b`

	Division
Generic	`a / b`
MATLAB	`a / b`
R	`a / b`
Python	`a / b` or `divide(a,b)`
gnuplot	`a / b`
IDL	`a / b`

	Power, a^b
MATLAB	`a .^ b`
R	`a ^ b`
Python	`a ** b`
	`power(a,b)`
	`pow(a,b)`
gnuplot	`a ** b`
IDL	`a ^ b`
Axiom	`a**b`
Maxima	`a^b` or `a**b`
bc	`a ^ b`

	Remainder
MATLAB	`rem(a,b)`
Scilab	`modulo(a,b)`
R	`a %% b`
Python	`a % b`
	`remainder(a,b)`
	`fmod(a,b)`
gnuplot	`a % b`
IDL	`a MOD b`
Axiom	`rem(a,b)`
Maxima	`mod(a,b)`
Maple	`a mod b`
Mathematica	`Mod[a,b]`
MuPAD	`a mod b`
Derive	`MOD(a,b)`
bc	`a % b`

	Integer division
R	`a %/% b`
bc	`a / b`

	Increment, return new value
Octave	`++a`
IDL	`++a` or `a+=1`
bc	`++a`

	Increment, return old value
Octave	`a++`
IDL	`a++`
bc	`a++`

	In place operation to save array creation overhead
Python	`a+=b` or `add(a,b,a)`
Octave	`a+=1`
IDL	`a+=1`
bc	`a+=b`

	Factorial, $n!$
MATLAB	`factorial(a)`
R	`factorial(a)`
Axiom	`factorial(a)`
Maxima	`a!`
Maple	`a!`

2.2 Relational operators

	Equal
Generic	`a == b`
MATLAB	`a == b`
R	`a == b`
Python	`a == b` or `equal(a,b)`
IDL	`a eq b`
bc	`a == b`

	Less than
Generic	`a < b`
MATLAB	`a < b`
R	`a < b`
Python	`a < b` or `less(a,b)`
IDL	`a lt b`
bc	`a < b`

	Greater than
Generic	`a > b`
MATLAB	`a > b`
R	`a > b`
Python	`a > b` or `greater(a,b)`
IDL	`a gt b`
bc	`a > b`

	Less than or equal
Generic	`a <= b`
MATLAB	`a <= b`
R	`a <= b`
Python	`a <= b` or `less_equal(a,b)`
IDL	`a le b`
bc	`a <= b`

	Greater than or equal
Generic	`a >= b`
MATLAB	`a >= b`
R	`a >= b`
Python	`a >= b` or `greater_equal(a,b)`
IDL	`a ge b`
bc	`a >= b`

	Not Equal
MATLAB	`a ~= b`
Scilab	`a ~= b` or `a <> b`
R	`a != b`
Python	`a != b` or `not_equal(a,b)`
IDL	`a ne b`
bc	`a != b`

2.3 Logical operators

	Short-circuit logical AND
MATLAB	`a && b`
Python	`a and b`
R	`a && b`
bc	`a && b`

	Short-circuit logical OR
MATLAB	`a || b`
Python	`a or b`
R	`a || b`
bc	`a || b`

	Element-wise logical AND
MATLAB	`a & b` or `and(a,b)`
R	`a & b`
Python	`logical_and(a,b)` or `a and b`
IDL	`a and b`

	Element-wise logical OR
MATLAB	`a | b` or `or(a,b)`
R	`a | b`
Python	`logical_or(a,b)` or `a or b`
IDL	`a or b`

	Logical EXCLUSIVE OR
MATLAB	`xor(a, b)`
R	`xor(a, b)`
Python	`logical_xor(a,b)`
IDL	`a xor b`

	Logical NOT
MATLAB	`~a` or `not(a)`
Octave	`~a` or `!a`
R	`!a`
Python	`logical_not(a)` or `not a`
IDL	`not a`
bc	`!a`

	True if any element is nonzero
MATLAB	`any(a)`

	True if all elements are nonzero
MATLAB	`all(a)`
Scilab	`and(a)`

2.4 root and logarithm

	Square root	
Generic	`sqrt(a)`	$\sqrt{a}$
MATLAB	`sqrt(a)`	
R	`sqrt(a)`	
Python	`math.sqrt(a)`	
gnuplot	`sqrt(a)`	
IDL	`sqrt(a)`	
Axiom	`sqrt(a)`	
Maxima	`sqrt(a)`	
Maple	`sqrt(a)`	
Mathematica	`Sqrt[a]`	
bc	`sqrt(a)`	

	Logarithm, base e (natural)	
Generic	`log(a)`	$\ln a = \log_e a$
MATLAB	`log(a)`	
R	`log(a)`	
Python	`math.log(a)`	
gnuplot	`log(a)`	
IDL	`alog(a)`	
Axiom	`log(a)`	
Maxima	`log(a)`	
Maple	`log(a)`	
Mathematica	`Log[a]`	
MuPAD	`ln(a)`	
bc	`l(a)`	

	Logarithm, base 10	
Generic	`log10(a)`	$\log_{10} a$
MATLAB	`log10(a)`	
R	`log10(a)`	
Python	`math.log10(a)`	
gnuplot	`log10(a)`	
IDL	`alog10(a)`	

	Logarithm, base 2 (binary)	
MATLAB	`log2(a)`	$\log_2 a$
R	`log2(a)`	
Python	`math.log(a, 2)`	

	Exponential function	
Generic	`exp(a)`	e^a
MATLAB	`exp(a)`	
R	`exp(a)`	
Python	`math.exp(a)`	
gnuplot	`exp(a)`	
IDL	`exp(a)`	
bc	`e(a)`	

2.5 Round off

	Round
MATLAB	`round(a)`
R	`round(a)`
Python	`around(a)` or `math.round(a)`
IDL	`round(a)`

	Round up
MATLAB	`ceil(a)`
R	`ceil(a)`
Python	`ceil(a)`
gnuplot	`ceil(a)`
IDL	`ceil(a)`

	Round down
MATLAB	`floor(a)`
R	`floor(a)`
Python	`floor(a)`
gnuplot	`floor(a)`
IDL	`floor(a)`

	Round towards zero
MATLAB	`fix(a)`
Python	`fix(a)`

2.6 Mathematical constants

	$\pi = 3.141592$
MATLAB	`pi`
Scilab	`%pi`
R	`pi`
Python	`math.pi`
IDL	`!pi`
Axiom	`%pi`
Maxima	`%pi`
Maple	`Pi`
Mathematica	`Pi`
MuPAD	`PI`

	$e = 2.718281$
MATLAB	`exp(1)`
Scilab	`%e`
R	`exp(1)`
Python	`math.e` or `math.exp(1)`
gnuplot	`exp(1)`
IDL	`exp(1)`
Axiom	`%e`
Maxima	`%e`
Maple	`exp(1)`
Mathematica	`E`
MuPAD	`E`

2.6.1 Missing values; IEEE-754 floating point status flags

	Not a Number
MATLAB	`NaN`
Python	`nan`
Scilab	`%nan`

	Infinity, ∞
MATLAB	`Inf`
Scilab	`%inf`
Python	`inf`

	Infinity, $+\infty$
Python	`plus_inf`
Axiom	`%plusInfinity`

	Infinity, $-\infty$
Python	`minus_inf`
Axiom	`%minusInfinity`

	Plus zero, $+0$
Python	`plus_zero`

	Minus zero, -0
Python	`minus_zero`

2.7 Complex numbers

	Imaginary unit	
MATLAB	`i`	
Scilab	`%i`	
R	`1i`	
Python	`z = 1j`	$i = \sqrt{-1}$
gnuplot	`{0,1}`	
IDL	`complex(0,1)`	
Axiom	`%i`	
Maxima	`%i`	

	A complex number, $3 + 4i$
MATLAB	`z = 3+4i`
Scilab	`z = 3+4*%i`
R	`z <- 3+4i`
Python	`z = 3+4j` or `z = complex(3,4)`
gnuplot	`{3,4}`
IDL	`z = complex(3,4)`
Axiom	`3+4*%i`
Maxima	`3+4*%i`

	Absolute value (modulus)
Generic	`abs(z)`
MATLAB	`abs(z)`
R	`abs(3+4i)` or `Mod(3+4i)`
Python	`abs(3+4j)`
gnuplot	`abs({3,4})`
IDL	`abs(z)`
Maxima	`abs(z);`

	Real part
Generic	`real(z)`
MATLAB	`real(z)`
R	`Re(3+4i)`
Python	`z.real`
gnuplot	`real({3,4})`
IDL	`real_part(z)`
Maxima	`realpart(z)`

	Imaginary part
Generic	`imag(z)`
MATLAB	`imag(z)`
R	`Im(3+4i)`
Python	`z.imag`
IDL	`imaginary(z)`
gnuplot	`imag({3,4})`
Maxima	`imagpart(z)`

	Argument
MATLAB	`arg(z)`
R	`Arg(3+4i)`
gnuplot	`arg({3,4})`

	Complex conjugate
Generic	`conj(z)`
MATLAB	`conj(z)`
R	`Conj(3+4i)`
Python	`z.conj(); z.conjugate()`
IDL	`conj(z)`

2.8 Trigonometry

	Sine
Generic	`sin(a)`

	Cosine
Generic	`cos(a)`

	Tangent
Generic	`tan(a)`

	Arcsine
Generic	`asin(a)` or `arcsin(a)`

	Arccosine
Generic	`acos(a)` or `arccos(a)`

	Arctangent
Generic	`atan(a)` or `arctan(a)`

	Arctangent, arctan(b/a)
MATLAB	`atan(a,b)`
R	`atan2(b,a)`
Python	`atan2(b,a)`

	Hyperbolic sine
Generic	`sinh(a)`

	Hyperbolic cosine
Generic	`cosh(a)`

	Hyperbolic tangent
Generic	`tanh(a)`

	Hypotenus; Euclidean distance	
Python	`hypot(x,y)`	$\sqrt{x^2+y^2}$

2.9 Generate random numbers

	Uniform distribution
MATLAB	`rand(1,10)`
Scilab	`rand(1,10,'uniform')`
R	`runif(10)`
Python	`random.random((10,))` `random.uniform((10,))`
IDL	`randomu(seed, 10)`

	Uniform: Numbers between 2 and 7
MATLAB	`2+5*rand(1,10)`
Scilab	`2+5*rand(1,10,'uniform')`
R	`runif(10, min=2, max=7)`
Python	`random.uniform(2,7,(10,))`
IDL	`2+5*randomu(seed, 10)`

	Uniform: 6,6 array
MATLAB	`rand(6)`
Scilab	`rand(6,6,'uniform')`
R	`matrix(runif(36),6)`
Python	`random.uniform(0,1,(6,6))`
IDL	`randomu(seed,[6,6])`

	Normal distribution
MATLAB	`randn(1,10)`
Scilab	`rand(1,10,'normal')`
R	`rnorm(10)`
Python	`random.standard_normal((10,))`
IDL	`randomn(seed, 10)`

3 Vectors

	Row vector, $1 \times n$-matrix
MATLAB	`a=[2 3 4 5];`
R	`a <- c(2,3,4,5)`
Python	`a=array([2,3,4,5])`
IDL	`a = [2, 3, 4, 5]`

	Column vector, $m \times 1$-matrix
MATLAB	`adash=[2 3 4 5]';`
R	`adash <- t(c(2,3,4,5))`
Python	`array([2,3,4,5])[:,NewAxis]`
	`array([2,3,4,5]).reshape(-1,1)`
	`r_[1:10,'c']`
IDL	`transpose([2,3,4,5])`

3.1 Sequences

	1,2,3, ... ,10
MATLAB	`1:10`
R	`seq(10)` or `1:10`
Python	`arange(1,11, dtype=Float)`
	`range(1,11)`
IDL	`indgen(10)+1`
	`dindgen(10)+1`

	0.0,1.0,2.0, ... ,9.0
MATLAB	`0:9`
R	`seq(0,length=10)`
Python	`arange(10.)`
IDL	`dindgen(10)`

	1,4,7,10
MATLAB	`1:3:10`
R	`seq(1,10,by=3)`
Python	`arange(1,11,3)`
IDL	`indgen(4)*3+1`

	10,9,8, ... ,1
MATLAB	`10:-1:1`
R	`seq(10,1)` or `10:1`
Python	`arange(10,0,-1)`

	10,7,4,1
MATLAB	`10:-3:1`
R	`seq(from=10,to=1,by=-3)`
Python	`arange(10,0,-3)`

	Linearly spaced vector of n=7 points
MATLAB	`linspace(1,10,7)`
R	`seq(1,10,length=7)`
Python	`linspace(1,10,7)`

	Reverse
MATLAB	`reverse(a)`
Scilab	`a($:-1:1)`
R	`rev(a)`
Python	`a[::-1]` or
IDL	`reverse(a)`

	Set all values to same scalar value
MATLAB	`a(:) = 3`
Python	`a.fill(3), a[:] = 3`

3.2 Concatenation (vectors)

	Concatenate two vectors
MATLAB	`[a a]`
R	`c(a,a)`
Python	`concatenate((a,a))`
IDL	`[a,a]` or `rebin(a,2,size(a))`

MATLAB	`[1:4 a]`
R	`c(1:4,a)`
Python	`concatenate((range(1,5),a), axis=1)`
IDL	`[indgen(3)+1,a]`

3.3 Repeating

	1 2 3, 1 2 3
MATLAB	`[a a]`
R	`rep(a,times=2)`
Python	`concatenate((a,a))`

	1 1 1, 2 2 2, 3 3 3
R	`rep(a,each=3)`
Python	`a.repeat(3)` or

	1, 2 2, 3 3 3
R	`rep(a,a)`
Python	`a.repeat(a)` or

3.4 Miss those elements out

	miss the first element
MATLAB	`a(2:end)`
Scilab	`a(2:$)`
R	`a[-1]`
Python	`a[1:]`

	miss the tenth element
MATLAB	`a([1:9])`
R	`a[-10]`

	miss 1,4,7, ...
R	`a[-seq(1,50,3)]`

	last element
MATLAB	`a(end)`
Scilab	`a($)`
Python	`a[-1]`

	last two elements
MATLAB	`a(end-1:end)`
Python	`a[-2:]`

3.5 Maximum and minimum

	pairwise max
MATLAB	`max(a,b)`
Python	`maximum(a,b)`
R	`pmax(a,b)`

	max of all values in two vectors
MATLAB	`max([a b])`
Python	`concatenate((a,b)).max()`
R	`max(a,b)`

MATLAB	`[v,i] = max(a)`
Python	`v,i = a.max(0),a.argmax(0)`
R	`v <- max(a) ; i <- which.max(a)`

3.6 Vector multiplication

	Multiply two vectors
MATLAB	`a.*a`
R	`a*a`
Python	`a*a`

	Vector cross product, $u \times v$
IDL	`crossp(u,v)`
Mathematica	`cross(u,v)`

	Vector dot product, $u \cdot v$
MATLAB	`dot(u,v)`
Python	`dot(u,v)`

4 Matrices

	Define a matrix	
MATLAB	`a = [2 3;4 5]`	
R	`rbind(c(2,3),c(4,5))`	
	`array(c(2,3,4,5), dim=c(2,2))`	
Python	`a = array([[2,3],[4,5]])`	$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$
IDL	`a = [[2,3],[4,5]]`	
Axiom	`a := matrix [[2,3],[4,5]]`	
Maxima	`matrix([2,3],[4,5])`	
Maple	`matrix([[2,3],[4,5]])`	
Mathematica	`{{2,3},{4,5}}`	
Derive	`[[2,3],[4,5]]`	

4.1 Concatenation (matrices); rbind and cbind

	Bind rows
MATLAB	`[a ; b]`
R	`rbind(a,b)`
Python	`concatenate((a,b), axis=0)`
	`vstack((a,b))`

	Bind columns
MATLAB	`[a , b]`
R	`cbind(a,b)`
Python	`concatenate((a,b), axis=1)`
	`hstack((a,b))`

	Bind slices (three-way arrays)
Python	`concatenate((a,b), axis=2)` `dstack((a,b))`

	Concatenate matrices into one vector
MATLAB	`[a(:), b(:)]`
Python	`concatenate((a,b), axis=None)`

	Bind rows (from vectors)
MATLAB	`[1:4 ; 1:4]`
R	`rbind(1:4,1:4)`
Python	`concatenate((r_[1:5],r_[1:5])).reshape(2,-1)` `vstack((r_[1:5],r_[1:5]))`

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    1    2    3    4
```

	Bind columns (from vectors)
MATLAB	`[1:4 ; 1:4]'`
R	`cbind(1:4,1:4)`

```
     [,1] [,2]
[1,]    1    1
[2,]    2    2
[3,]    3    3
[4,]    4    4
```

4.2 Array creation

	0 filled array	
MATLAB	`zeros(3,5)`	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
R	`matrix(0,3,5)` or `array(0,c(3,5))`	
Python	`zeros((3,5),Float)`	
IDL	`dblarr(3,5)`	

	0 filled array of integers
Python	`zeros((3,5))`
IDL	`intarr(3,5)`

	1 filled array	
MATLAB	`ones(3,5)`	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
R	`matrix(1,3,5)` or `array(1,c(3,5))`	
Python	`ones((3,5),Float)`	
IDL	`dblarr(3,5)+1`	

	Any number filled array	
MATLAB	`ones(3,5)*9`	$\begin{bmatrix} 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 9 \end{bmatrix}$
R	`matrix(9,3,5)` or `array(9,c(3,5))`	
Python		
IDL	`intarr(3,5)+9`	

	Identity matrix	
MATLAB	`eye(3)`	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Scilab	`eye(3,3)`	
R	`diag(1,3)`	
Python	`identity(3)`	
IDL	`identity(3)`	

	Diagonal	
MATLAB	`diag([4 5 6])`	$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \end{bmatrix}$
R	`diag(c(4,5,6))`	
Python	`diag((4,5,6))`	
IDL	`diag_matrix([4,5,6])`	
Axiom	`diagonalMatrix([4,5,6])`	

	Magic squares; Lo Shu	
MATLAB	`magic(3)`	$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$
Scilab	`testmatrix('magi',3)`	

	Empty array
Python	`a = empty((3,3))`

4.3 Reshape and flatten matrices

	Reshaping (rows first)	
MATLAB	`reshape(1:6,3,2)';`	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
Scilab	`matrix(1:6,3,2)';`	
R	`matrix(1:6,nrow=3,byrow=T)`	
Python	`arange(1,7).reshape(2,-1)` `a.setshape(2,3)`	
IDL	`reform(a,2,3)`	

	Reshaping (columns first)	
MATLAB	`reshape(1:6,2,3);`	$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$
Scilab	`matrix(1:6,2,3);`	
R	`matrix(1:6,nrow=2)` `array(1:6,c(2,3))`	
Python	`arange(1,7).reshape(-1,2).transpose()`	

	Flatten to vector (by rows, like comics)	
MATLAB	`a'(:)`	$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$
R	`as.vector(t(a))`	
Python	`a.flatten()` or	

	Flatten to vector (by columns)	
MATLAB	`a(:)`	
R	`as.vector(a)`	$\begin{bmatrix} 1 & 4 & 2 & 5 & 3 & 6 \end{bmatrix}$
Python	`a.flatten(1)`	

	Flatten upper triangle (by columns)	
MATLAB	`vech(a)`	
R	`a[row(a) <= col(a)]`	

4.4 Shared data (slicing)

	Copy of a	
MATLAB	`b = a`	
R	`b = a`	
Python	`b = a.copy()`	

4.5 Indexing and accessing elements (Python: slicing)

	Input is a 3,4 array	

MATLAB	`a = [ 11 12 13 14 ...`
	`      21 22 23 24 ...`
	`      31 32 33 34 ]`	

R	`a <- rbind(c(11, 12, 13, 14),`
	`           c(21, 22, 23, 24),`
	`           c(31, 32, 33, 34))`
		$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$

Python	`a = array([[ 11, 12, 13, 14 ],`
	`           [ 21, 22, 23, 24 ],`
	`           [ 31, 32, 33, 34 ]])`	

IDL	`a = [[ 11, 12, 13, 14 ], $`
	`     [ 21, 22, 23, 24 ], $`
	`     [ 31, 32, 33, 34 ]]`	

	Element 2,3 (row,col)	
MATLAB	`a(2,3)`	
R	`a[2,3]`	a_{23}
Python	`a[1,2]`	
IDL	`a(2,1)`	

	First row	
MATLAB	`a(1,:)`	
R	`a[1,]`	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \end{bmatrix}$
Python	`a[0,]`	
IDL	`a(*,0)`	

	First column	
MATLAB	`a(:,1)`	$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}$
R	`a[,1]`	
Python	`a[:,0]`	
IDL	`a(0,*)`	

	Array as indices	
MATLAB	`a([1 3],[1 4]);`	$\begin{bmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{bmatrix}$
Python	`a.take([0,2]).take([0,3], axis=1)`	

	All, except first row	
MATLAB	`a(2:end,:)`	
Scilab	`a(2:$,:)`	$\begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$
R	`a[-1,]`	
Python	`a[1:,]`	
IDL	`a(*,1:*)`	

	Last two rows	
MATLAB	`a(end-1:end,:)`	$\begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$
Python	`a[-2:,]`	

	Strides: Every other row	
MATLAB	`a(1:2:end,:)`	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$
Python	`a[::2,:]`	

	Third in last dimension (axis)	
Python	`a[...,2]`	

	All, except row,column (2,3)	
R	`a[-2,-3]`	$\begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{31} & a_{33} & a_{34} \end{bmatrix}$

	Remove one column	
MATLAB	`a(:,[1 3 4])`	$\begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \end{bmatrix}$
R	`a[,-2]`	
Python	`a.take([0,2,3],axis=1)`	

	Diagonal	
Python	`a.diagonal(offset=0)`	$\begin{bmatrix} a_{11} & a_{22} & a_{33} & a_{44} \end{bmatrix}$

4.6 Assignment

MATLAB	`a(:,1) = 99`	
R	`a[,1] <- 99`	
Python	`a[:,0] = 99`	

MATLAB	`a(:,1) = [99 98 97]'`
R	`a[,1] <- c(99,98,97)`
Python	`a[:,0] = array([99,98,97])`

Clipping: Replace all elements over 90

MATLAB	`a(a>90) = 90;`
R	`a[a>90] <- 90`
Python	`(a>90).choose(a,90)`
	`a.clip(min=None, max=90)`
IDL	`a>90`

Clip upper and lower values

Python	`a.clip(min=2, max=5)`
IDL	`a < 2 > 5`

4.7 Transpose and inverse

Transpose

MATLAB	`a'`
R	`t(a)`
Python	`a.conj().transpose()`
IDL	`transpose(a)`
Maxima	`transpose(a);`

Non-conjugate transpose

MATLAB	`a.'` or `transpose(a)`
Python	`a.transpose()`

Determinant

MATLAB	`det(a)`
R	`det(a)`
Python	`linalg.det(a)` or `determinant(a)`
IDL	`determ(a)`
Axiom	`determinant a`
Maxima	`determinant(a);`

Inverse

MATLAB	`inv(a)`
R	`solve(a)`
Python	`linalg.inv(a)` or `inverse(a)`
IDL	`invert(a)`
Axiom	`inverse a`
Maxima	`invert(a),detout;`

Pseudo-inverse

MATLAB	`pinv(a)`
R	`ginv(a)`
Python	`linalg.pinv(a)`

Norms

MATLAB	`norm(a)`
Python	`norm(a)`

Eigenvalues

MATLAB	`eig(a)`
Scilab	`spec(a)`
R	`eigen(a)$values`
Python	`linalg.eig(a)[0]`
	`eigenvalues(a)`
IDL	`hqr(elmhes(a))`
Mathematica	`Eigenvalues[matrix]`
Axiom	`eigenvalues a`

Singular values

MATLAB	`svd(a)`
R	`svd(a)$d`
Python	`linalg.svd(a)`
	`singular_value_decomposition(a)`
IDL	`svdc,A,w,U,V`
Mathematica	`SingularValueDecomposition[m]`

Cholesky factorization

MATLAB	`chol(a)`
Python	`linalg.cholesky(a)`

Eigenvectors

MATLAB	`[v,l] = eig(a)`
Scilab	`[v,l] = spec(a)`
R	`eigen(a)$vectors`
Python	`linalg.eig(a)[1]`
	`eigenvectors(a)`
Axiom	`eigenvectors a`

Rank

MATLAB	`rank(a)`
R	`rank(a)`
Python	`rank(a)`

4.8 Sum

	Sum of each column
MATLAB	`sum(a)`
Scilab	`sum(a,'c')`
R	`apply(a,2,sum)`
Python	`a.sum(axis=0)`
IDL	`total(a,2)`

	Sum of each row
MATLAB	`sum(a')`
Scilab	`sum(a,'r')`
R	`apply(a,1,sum)`
Python	`a.sum(axis=1)`
IDL	`total(a,1)`

	Sum of all elements
MATLAB	`sum(sum(a))`
Scilab	`sum(a)`
R	`sum(a)`
Python	`a.sum()`
IDL	`total(a)`

	Sum along diagonal
Python	`a.trace(offset=0)`

	Cumulative sum (columns)
MATLAB	`cumsum(a)`
R	`apply(a,2,cumsum)`
Python	`a.cumsum(axis=0)`

4.9 Sorting

	Example data	$\begin{bmatrix} 4 & 3 & 2 \\ 2 & 8 & 6 \\ 1 & 4 & 7 \end{bmatrix}$
MATLAB	`a = [ 4 3 2 ; 2 8 6 ; 1 4 7 ]`	
Python	`a = array([[4,3,2],[2,8,6],[1,4,7]])`	

	Flat and sorted	$\begin{bmatrix} 1 & 2 & 2 \\ 3 & 4 & 4 \\ 6 & 7 & 8 \end{bmatrix}$
MATLAB	`sort(a(:))`	
Scilab	`s=sort(a(:)); s($:-1:1)`	
R	`t(sort(a))`	
Python	`a.ravel().sort()` or	

	Sort each column	$\begin{bmatrix} 1 & 3 & 2 \\ 2 & 4 & 6 \\ 4 & 8 & 7 \end{bmatrix}$
MATLAB	`sort(a)`	
Scilab	`s=sort(a,'r'); s($:-1:1,:)`	
R	`apply(a,2,sort)`	
Python	`a.sort(axis=0)` or `msort(a)`	
IDL	`sort(a)`	

	Sort each row	$\begin{bmatrix} 2 & 3 & 4 \\ 2 & 6 & 8 \\ 1 & 4 & 7 \end{bmatrix}$
MATLAB	`sort(a')'`	
Scilab	`s=sort(a,'c'); s(:,$:-1:1)`	
R	`t(apply(a,1,sort))`	
Python	`a.sort(axis=1)`	

	Sort rows (by first row)	$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 8 & 6 \\ 4 & 3 & 2 \end{bmatrix}$
MATLAB	`sortrows(a,1)`	
Python	`a[a[:,0].argsort(),]`	

	Sort, return indices
R	`order(a)`
Python	`a.ravel().argsort()`

	Sort each column, return indices
Python	`a.argsort(axis=0)`

	Sort each row, return indices
Python	`a.argsort(axis=1)`

4.10 Maximum and minimum

	max in each column
MATLAB	`max(a)`
Scilab	`max(a,'c')`
R	`apply(a,2,max)`
Python	`a.max(0)` or `amax(a [,axis=0])`
IDL	`max(a,DIMENSION=2)`

	max in each row
MATLAB	`max(a')`
Scilab	`max(a,'r')`
R	`apply(a,1,max)`
Python	`a.max(1)` or `amax(a, axis=1)`
IDL	`max(a,DIMENSION=1)`

	max in array
MATLAB	`max(max(a))`
Scilab	`max(a)`
R	`max(a)`
Python	`a.max()` or
IDL	`max(a)`

	return indices, i
MATLAB	`[v i] = max(a)`
Scilab	`[v,i] = max(a,'c')`
R	`i <- apply(a,1,which.max)`

	pairwise max
MATLAB	`max(b,c)`
Python	`maximum(b,c)`
R	`pmax(b,c)`

MATLAB	`cummax(a)`
R	`apply(a,2,cummax)`

	max-to-min range
Python	`a.ptp(); a.ptp(0)`

4.11 Matrix manipulation

	Flip left-right
MATLAB	`fliplr(a)`
Scilab	`a(:,$:-1:1)` or `mtlb_fliplr(a)`
R	`a[,4:1]`
Python	`fliplr(a)` or `a[:,::-1]`
IDL	`reverse(a)`

	Flip up-down
MATLAB	`flipud(a)`
Scilab	`a($:-1:1,:)`
R	`a[3:1,]`
Python	`flipud(a)` or `a[::-1,]`
IDL	`reverse(a,2)`

	Rotate 90 degrees
MATLAB	`rot90(a)`
Scilab	`---`
Python	`rot90(a)`
IDL	`rotate(a,1)`

	Repeat matrix: [a a a ; a a a]
MATLAB	`repmat(a,2,3)`
Scilab	`mtlb_repmat(a,2,3)`
Octave	`kron(ones(2,3),a)`
Python	`kron(ones((2,3)),a)`
R	`kronecker(matrix(1,2,3),a)`

	Triangular, upper
MATLAB	`triu(a)`
R	`a[lower.tri(a)] <- 0`
Python	`triu(a)`
Mathematica	`UpperDiagonalMatrix[f, n]`

	Triangular, lower
MATLAB	`tril(a)`
R	`a[upper.tri(a)] <- 0`
Python	`tril(a)`

4.12 Equivalents to "size"

	Matrix dimensions
MATLAB	`size(a)`
R	`dim(a)`
Python	`a.shape` or `a.getshape()`
IDL	`size(a)`

	Number of columns
MATLAB	`size(a,2)` or `length(a)`
R	`ncol(a)`
Python	`a.shape[1]` or `size(a, axis=1)`
IDL	`s=size(a) & s[1]`
Axiom	`ncols(m)`
Maxima	`mat_ncols(m)`
Maple	`linalg[coldim](m)`
Mathematica	`Dimensions[m][[2]]`
Derive	`DIMENSION(m SUB 1)`

	Number of elements
MATLAB	`length(a(:))`
Scilab	`length(a)`
R	`prod(dim(a))`
Python	`a.size` or `size(a[, axis=None])`
IDL	`n_elements(a)`

	Number of dimensions
MATLAB	`ndims(a)`
Python	`a.ndim`

	Number of bytes used in memory
R	`object.size(a)`
Python	`a.nbytes`

4.13 Matrix- and elementwise- multiplication

	Elementwise operations	
MATLAB	`a .* b`	$\begin{bmatrix} 1 & 5 \\ 9 & 16 \end{bmatrix}$
R	`a * b`	
Python	`a * b` or `multiply(a,b)`	

	Matrix product (dot product)	
MATLAB	`a * b`	
R	`a %*% b`	
Python	`matrixmultiply(a,b)`	$\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$
IDL	`a # b` or `b ## a`	
Axiom	`a*b`	
Maxima	`a.b`	
Maple	`evalm(a &* b)`	
Mathematica	`a.b`	

	Inner matrix vector multiplication $a \cdot b'$	
Python	`inner(a,b)` or	$\begin{bmatrix} 5 & 11 \\ 11 & 25 \end{bmatrix}$
IDL	`transpose(a) # b`	

	Outer product	
R	`outer(a,b)` or `a %o% b`	$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 4 & 8 & 12 & 16 \end{bmatrix}$
Python	`outer(a,b)` or	
IDL	`a # b`	

	Cross product	
R	`crossprod(a,b)` or `t(a) %*% b`	$\begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}$

	Kronecker product	
MATLAB	`kron(a,b)`	$\begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 4 & 6 & 8 \\ 3 & 6 & 4 & 8 \\ 9 & 12 & 12 & 16 \end{bmatrix}$
Scilab	`kron(a,b)` or `a .*. b`	
R	`kronecker(a,b)`	
Python	`kron(a,b)`	

	Matrix division, $b{\cdot}a^{-1}$	
MATLAB	`a / b`	

	Left matrix division, $b^{-1}{\cdot}a$ (solve linear equations)	
MATLAB	`a \ b`	
Scilab	`linsolve(a,b)`	
R	`solve(a,b)`	$Ax = b$
Python	`linalg.solve(a,b)`	
	`solve_linear_equations(a,b)`	
IDL	`cramer(a,b)`	

	Vector dot product	
Python	`vdot(a,b)`	

	Cross product	
Python	`cross(a,b)`	

4.14 Find; conditional indexing

	Non-zero elements, indices
MATLAB	`find(a)`
R	`which(a != 0)`
Python	`a.ravel().nonzero()`
	`nonzero(a.flat)`

	Non-zero elements, array indices
MATLAB	`[i j] = find(a)`
R	`which(a != 0, arr.ind=T)`
Python	`(i,j) = a.nonzero()`
	`(i,j) = where(a!=0)`
	`(i,j) = nonzero(a)`
IDL	`where(a NE 0)`

	Vector of non-zero values
MATLAB	`[i j v] = find(a)`
R	`ij <- which(a != 0, arr.ind=T); v <- a[ij]`
Python	`v = a.compress((a!=0).flat)`
	`v = extract(a!=0,a)`
IDL	`a(where(a NE 0))`

	Condition, indices
MATLAB	`find(a>5.5)`
R	`which(a>5.5)`
Python	`(a>5.5).nonzero()`
IDL	`where(a GE 5.5)`

	Return values
R	`ij <- which(a>5.5, arr.ind=T); v <- a[ij]`
Python	`a.compress((a>5.5).flat)`
IDL	`a(where(a GE 5.5))`

	Zero out elements above 5.5
MATLAB	`a .* (a>5.5)`
Python	`where(a>5.5,0,a)` or `a * (a>5.5)`

	Replace values
Python	`a.put(2,indices)`

5 Multi-way arrays

	Define a 3-way array
MATLAB	`a = cat(3, [1 2; 1 2],[3 4; 3 4]);`
Python	`a = array([[[1,2],[1,2]], [[3,4],[3,4]]])`

MATLAB	`a(1,:,:)`
Python	`a[0,...]`

6 File input and output

	Reading from a file (2d)
MATLAB	`f = load('data.txt')`
R	`f <- read.table("data.txt")`
Python	`f = fromfile("data.txt")`
	`f = load("data.txt")`
IDL	`read()`

	Reading from a file (2d)
MATLAB	`f = load('data.txt')`
R	`f <- read.table("data.txt")`
Python	`f = load("data.txt")`
IDL	`read()`

	Reading fram a CSV file (2d)
MATLAB	`x = dlmread('data.csv', ';')`
R	`f <- read.table(file="data.csv", sep=";")`
Python	`f = load('data.csv', delimiter=';')`
gnuplot	`set datafile separator ";"`
IDL	`x = read_ascii(data_start=1,delimiter=';')`

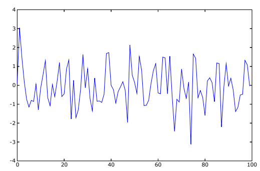
	Writing to a file (2d)
MATLAB	`save -ascii data.txt f`
R	`write(f,file="data.txt")`
Python	`save('data.csv', f, fmt='%.6f', delimiter=';')`

	Writing to a file (1d)
Python	`f.tofile(file='data.csv', format='%.6f', sep=';')`

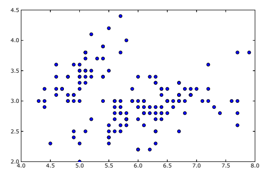
	Reading from a file (1d)
Python	`f = fromfile(file='data.csv', sep=';')`

7 Plotting

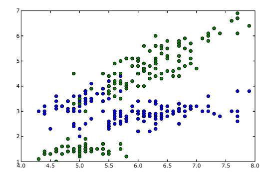
7.1 Basic x-y plots



	1d line plot
MATLAB	`plot(a)`
R	`plot(a, type="l")`
Python	`plot(a)`
IDL	`plot, a`



	2d scatter plot
MATLAB	`plot(x(:,1),x(:,2),'o')`
R	`plot(x[,1],x[,2])`
Python	`plot(x[:,0],x[:,1],'o')`
IDL	`plot, x(1,*), x(2,*)`



	Two graphs in one plot
MATLAB	`plot(x1,y1, x2,y2)`
Python	`plot(x1,y1,'bo', x2,y2,'go')`

	Overplotting: Add new plots to current

MATLAB	`plot(x1,y1)`
	`hold on`
	`plot(x2,y2)`

R	`plot(x1,y1)` `matplot(x2,y2,add=T)`

Python	`plot(x1,y1,'o')`
	`plot(x2,y2,'o')`
	`show() # as normal`

IDL	`plot, x1, y1` `oplot, x2, y2`

	subplots
MATLAB	`subplot(211)`
Python	`subplot(211)`
IDL	`!p.multi(0,2,1)`

	Plotting symbols and color
MATLAB	`plot(x,y,'ro-')`
R	`plot(x,y,type="b",col="red")`
Python	`plot(x,y,'ro-')`
IDL	`plot, x,y, line=1, psym=-1`

7.1.1 Axes and titles

	Turn on grid lines
MATLAB	`grid on`
R	`grid()`
Python	`grid()`

	1:1 aspect ratio
MATLAB	`axis equal`
Octave	`axis('equal')` `replot`
R	`plot(c(1:10,10:1), asp=1)`
Python	`figure(figsize=(6,6))`
gnuplot	`set size ratio -1`

	Set axes manually
MATLAB	`axis([ 0 10 0 5 ])`
Scilab	`plot('axis',[ 0 10 0 5 ])`
R	`plot(x,y, xlim=c(0,10), ylim=c(0,5))`
gnuplot	`set xrange [0:10]` `set yrange [0:5]`
Python	`axis([ 0, 10, 0, 5 ])`
IDL	`plot, x(1,*), x(2,*),` `   xran=[0,10], yran=[0,5]`

	Axis labels and titles

MATLAB	`title('title')`
	`xlabel('x-axis')`
	`ylabel('y-axis')`

R	`plot(1:10, main="title",` `  xlab="x-axis", ylab="y-axis")`
IDL	`plot, x,y, title='title',` `  xtitle='x-axis', ytitle='y-axis'`

	Insert text
Python	`text(2,25,'hello')`
IDL	`xyouts, 2,25, 'hello'`

7.1.2 Log plots

	logarithmic y-axis
MATLAB	`semilogy(a)`
R	`plot(x,y, log="y")`
Python	`semilogy(a)`
IDL	`plot, x,y, /YLOG` or `plot_io, x,y`

	logarithmic x-axis
MATLAB	`semilogx(a)`
R	`plot(x,y, log="x")`
Python	`semilogx(a)`
IDL	`plot, x,y, /XLOG` or `plot_oi, x,y`

	logarithmic x and y axes
MATLAB	`loglog(a)`
R	`plot(x,y, log="xy")`
Python	`loglog(a)`
IDL	`plot_oo, x,y`

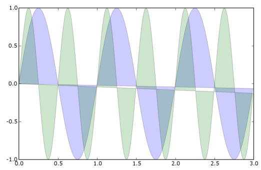
7.1.3 Filled plots and bar plots

	Filled plot
MATLAB	`fill(t,s,'b', t,c,'g')`
Octave	`% fill has a bug?`

R	`plot(t,s, type="n", xlab="", ylab="")`
	`polygon(t,s, col="lightblue")`
	`polygon(t,c, col="lightgreen")`

Python	`fill(t,s,'b', t,c,'g', alpha=0.2)`

gnuplot	`set xrange [0:3]`
	`set samples 3/.01`
	`plot sin(2*pi*x) with filledcurves,\`
	`sin(4*pi*x) with filledcurves`



	Stem-and-Leaf plot
R	`stem(x[,3])`

5	5
6	71
7	033
8	00113345567889
9	0133566677788
10	32674

7.1.4 Functions

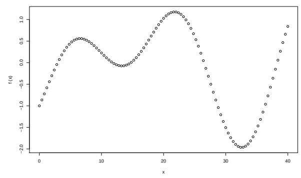
	Defining functions
MATLAB	`f = inline('sin(x/3) - cos(x/5)')`
Scilab	`deff('y = f(x)','y = sin(x/3) - cos(x/5)')`
R	`f <- function(x) sin(x/3) - cos(x/5)`

$f(x) = \sin\left(\frac{x}{3}\right) - \cos\left(\frac{x}{5}\right)$

	Plot a function for given range
MATLAB	`ezplot(f,[0,40])` `fplot('sin(x/3) - cos(x/5)',[0,40])`
Octave	`% no ezplot`
Scilab	`fplot2d([0:.5:40],f)`
R	`plot(f, xlim=c(0,40), type='p')`

Python	`x = arrayrange(0,40,.5)`
	`y = sin(x/3) - cos(x/5)`
	`plot(x,y, 'o')`

gnuplot	`set xrange [0,40]` `plot sin(x/3) - cos(x/5) with points`
Axiom	`draw( sin(x/3) - cos(x/5), x=0..40 )`

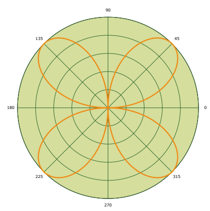


7.2 Polar plots

MATLAB	`theta = 0:.001:2*pi;` `r = sin(2*theta);`
Python	`theta = arange(0,2*pi,0.001)` `r = sin(2*theta)`

$\rho(\theta) = \sin(2\theta)$

MATLAB	`polar(theta, rho)`
Scilab	`polarplot(theta, rho)`
Python	`polar(theta, rho)`
gnuplot	`set polar` `plot sin(2*t)`



7.3 Histogram plots

MATLAB	`hist(randn(1000,1))`
R	`hist(rnorm(1000))`
IDL	`plot, histogram(randomn(5,1000))`

MATLAB	`hist(randn(1000,1), -4:4)`
R	`hist(rnorm(1000), breaks= -4:4)`

R	`hist(rnorm(1000), breaks=c(seq(-5,0,0.25), seq(0.5,5,0.5)), freq=F)`

MATLAB	`plot(sort(a))`
R	`plot(apply(a,1,sort),type="l")`

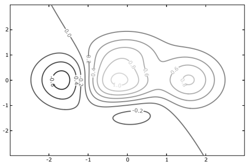
7.4 3d data

7.4.1 Contour and image plots

	Contour plot
MATLAB	`contour(z)`
R	`contour(z)`

Python	`levels, colls = contour(Z, V,`
	`    origin='lower', extent=(-3,3,-3,3))`
	`clabel(colls, levels, inline=1,`
	`    fmt='%1.1f', fontsize=10)`

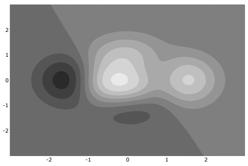
IDL	`contour, z`



	Filled contour plot
MATLAB	`contourf(z); colormap(gray)`
R	`filled.contour(x,y,z,` `    nlevels=7, color=gray.colors)`

Python	`contourf(Z, V,`
	`    cmap=cm.gray,`
	`    origin='lower',`
	`    extent=(-3,3,-3,3))`

IDL	`contour, z, nlevels=7, /fill` `contour, z, nlevels=7, /overplot, /downhill`



	Plot image data
MATLAB	`image(z)` `colormap(gray)`
R	`image(z, col=gray.colors(256))`

Python	`im = imshow(Z,`
	`    interpolation='bilinear',`
	`    origin='lower',`
	`    extent=(-3,3,-3,3))`

IDL	`tv, z` `loadct,0`

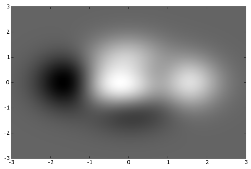
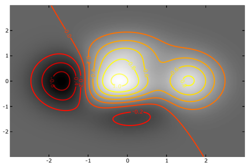


	Image with contours
Python	`# imshow() and contour() as above`



	Direction field vectors
MATLAB	`quiver()`
Scilab	`champ()`
Python	`quiver()`

7.4.2 Perspective plots of surfaces over the x-y plane

MATLAB	`n=-2:.1:2;`
	`[x,y] = meshgrid(n,n);`
	`z=x.*exp(-x.^2-y.^2);`

Python	`n=arrayrange(-2,2,.1)`
	`[x,y] = meshgrid(n,n)`
	`z = x*power(math.e,-x**2-y**2)`

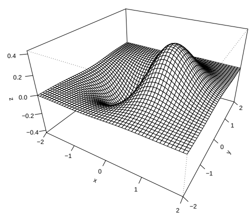
R	`f <- function(x,y) x*exp(-x^2-y^2)`
	`n <- seq(-2,2, length=40)`
	`z <- outer(n,n,f)`

$f(x,y) = xe^{-x^2-y^2}$

	Mesh plot
MATLAB	`mesh(z)`

R	`persp(x,y,z,`
	`  theta=30, phi=30, expand=0.6,`
	`  ticktype='detailed')`

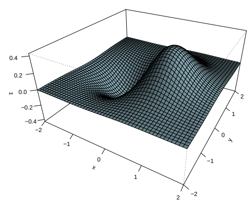
Python	
IDL	`surface, z`



	Surface plot
MATLAB	`surf(x,y,z)` or `surfl(x,y,z)`
Octave	`% no surfl()`

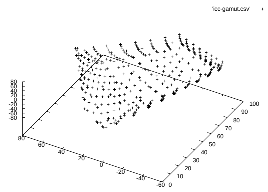
R	`persp(x,y,z,`
	`  theta=30, phi=30, expand=0.6,`
	`  col='lightblue', shade=0.75, ltheta=120,`
	`  ticktype='detailed')`

IDL	`shade_surf, z` `loadct,3`



7.4.3 Scatter (cloud) plots

	3d scatter plot
MATLAB	`plot3(x,y,z,'k+')`
R	`cloud(z~x*y)`
gnuplot	`splot 'icc-gamut.csv'`



7.5 Save plot to a graphics file

	PostScript
MATLAB	`plot(1:10)` `print -depsc2 foo.eps`

Octave	`gset output "foo.eps"`
	`gset terminal postscript eps`
	`plot(1:10)`

R	`postscript(file="foo.eps")`
	`plot(1:10)`
	`dev.off()`

Python	`savefig('foo.eps')`

gnuplot	`set terminal postscript enhanced eps color`
	`set output 'foo.eps'`
	`plot 1:10`

IDL	`set_plot,'PS'`
	`device, file='foo.eps', /land`
	`plot x,y`
	`device,/close & set_plot,'win'`

	PDF
Python	`savefig('foo.pdf')`
R	`pdf(file='foo.pdf')`
MATLAB	

	SVG (vector graphics for www)
Python	`savefig('foo.svg')`
R	`devSVG(file='foo.svg')`
gnuplot	`set terminal svg` `set output 'foo.svg'`
MATLAB	

	PNG (raster graphics)
MATLAB	`print -dpng foo.png`
Python	`savefig('foo.png')`
R	`png(filename = "Rplot%03d.png"`
gnuplot	`set terminal png medium` `set output 'foo.png'`

	Output TeX/LaTeX math
Axiom	`outputAsTex(e)`
Maxima	`tex(e);`
Maple	`latex(e);`
Mathematica	`TexForm[e]`
MuPAD	`generate::TeX(e);`

8 Data analysis

8.1 Set membership operators

	Create sets
MATLAB	`a = [ 1 2 2 5 2 ];` `b = [ 2 3 4 ];`
R	`a <- c(1,2,2,5,2)` `b <- c(2,3,4)`

Python	`a = array([1,2,2,5,2])`
	`b = array([2,3,4])`
	`a = set([1,2,2,5,2])`
	`b = set([2,3,4])`

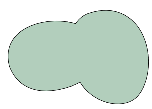
	Set unique
MATLAB	`unique(a)`
R	`unique(a)`

Python	`unique1d(a)`
	`unique(a)`
	`set(a)`

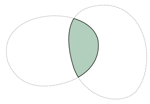
Maxima	`setify(a)`

$$\begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$$

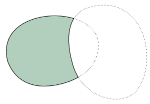
	Set union
MATLAB	`union(a,b)`
R	`union(a,b)`
Python	`union1d(a,b)` `a.union(b)`
Maxima	`union(a,b)`



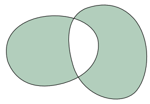
	Set intersection
MATLAB	`intersect(a,b)`
R	`intersect(a,b)`
Python	`intersect1d(a)`
	`a.intersection(b)`
Maxima	`intersect(a,b)`



	Set difference
MATLAB	`setdiff(a,b)`
R	`setdiff(a,b)`
Python	`setdiff1d(a,b)`
	`a.difference(b)`
Maxima	`setdifference(a,b)`
	`complement(b,a)`



	Set exclusion
MATLAB	`setxor(a,b)`
R	`setdiff(union(a,b),intersect(a,b))`
Python	`setxor1d(a,b)`
	`a.symmetric_difference(b)`



	True for set member
MATLAB	`ismember(2,a)`
R	`is.element(2,a)` or `2 %in% a`
Python	`2 in a`
	`setmember1d(2,a)`
	`contains(a,2)`

8.2 Statistics

	Average
MATLAB	`mean(a)`
R	`apply(a,2,mean)`
Python	`a.mean(axis=0)`
	`mean(a [,axis=0])`
IDL	`mean(a)`
Axiom	`mean a`

	Median
MATLAB	`median(a)`
R	`apply(a,2,median)`
Python	`median(a)` or `median(a [,axis=0])`
IDL	`median(a)`
Axiom	`median(a)`

	Standard deviation
MATLAB	`std(a)`
R	`apply(a,2,sd)`
Python	`a.std(axis=0)` or `std(a [,axis=0])`
IDL	`stddev(a)`

	Variance
MATLAB	`var(a)`
R	`apply(a,2,var)`
Python	`a.var(axis=0)` or `var(a)`
IDL	`variance(a)`

	Correlation coefficient
MATLAB	`corr(x,y)`
R	`cor(x,y)`
Python	`correlate(x,y)` or `corrcoef(x,y)`
IDL	`correlate(x,y)`

	Covariance
MATLAB	`cov(x,y)`
R	`cov(x,y)`
Python	`cov(x,y)`

8.3 Interpolation and regression

	Straight line fit
MATLAB	`z = polyval(polyfit(x,y,1),x)`
	`plot(x,y,'o', x,z ,'-')`
R	`z <- lm(y~x)`
	`plot(x,y)`
	`abline(z)`
Python	`(a,b) = polyfit(x,y,1)`
	`plot(x,y,'o', x,a*x+b,'-')`
IDL	`poly_fit(x,y,1)`

	Linear least squares $y = ax + b$
MATLAB	`a = x\y`
R	`solve(a,b)`
Python	`linalg.lstsq(x,y)`
	`(a,b) = linear_least_squares(x,y)[0]`

	Polynomial fit
MATLAB	`polyfit(x,y,3)`
Python	`polyfit(x,y,3)`

8.4 Non-linear methods

8.4.1 Polynomials, root finding

	Polynomial	
Scilab	`poly(1.,'x')`	
Python	`poly()`	
	Find zeros of polynomial	
MATLAB	`roots([1 -1 -1])`	$x^2 - x - 1 = 0$
R	`polyroot(c(1,-1,-1))`	
Python	`roots()`	
	Find a zero near $x = 1$	
MATLAB	`f = inline('1/x - (x-1)')`	$f(x) = \frac{1}{x} - (x - 1)$
	`fzero(f,1)`	
	Solve symbolic equations	
MATLAB	`solve('1/x = x-1')`	$\frac{1}{x} = x - 1$
	Evaluate polynomial	
MATLAB	`polyval([1 2 1 2],1:10)`	
Python	`polyval(array([1,2,1,2]),arange(1,11))`	

8.4.2 Differential equations

	Discrete difference function and approximate derivative	
MATLAB	`diff(a)`	
Python	`diff(x, n=1, axis=0)`	
	Solve differential equations	
MATLAB		

8.5 Fourier analysis

	Fast fourier transform	
Generic	`fft(a)`	
MATLAB	`fft(a)`	
R	`fft(a)`	
Python	`fft(a)` or `fft(a)`	
IDL	`fft(a)`	
	Inverse fourier transform	
MATLAB	`ifft(a)`	
R	`fft(a, inverse=TRUE)`	
Python	`ifft(a)` or `inverse_fft(a)`	
IDL	`fft(a),/inverse`	
	Linear convolution	
Python	`convolve(x,y)`	
IDL	`convol()`	

9 Symbolic algebra; calculus

	Decimal output	
Axiom	`numeric %`	
Maxima	`%,numer;`	
	Simplification	
Axiom	`simplify(e)` or `normalize(e)`	
Maxima	`ratsimp(e)` or `radcan(e)`	
Maple	`simplify(e)`	
Mathematica	`Simplify[e]` or `FullSimplify[e]`	
MuPAD	`simplify(e)` or `normal(e)`	
REDUCE	`e`	
Derive	`e`	
	Expand	
	Rectangular form	
Axiom	`rectform e`	
	Factorization	
MATLAB	`factor()`	
Axiom	`factor()`	
	Integration of functions	
Axiom	`integrate(f(x), x=0..1)`	
Maxima	`integrate(f(x), x, 0, 1)`	$\int_0^1 f(x)dx$
Maple	`int(f(x), x=0..1)`	
MuPAD	`int(f(x), x=0..1)`	
Mathematica	`Integrate[f[x], {x,0,1}]`	
	Differentiation	
Axiom	`differentiate(%,x)`	
Maxima	`diff(%,x)`	
	Taylor/Laurent/etc. series approxmation	
Axiom	`series(%,x=0)`	
	Solve equations	
Axiom	`solve(sys,vars)`	

	Laplace transform
Axiom	`laplace(e,t,s)`

10 Programming

	Script file extension
MATLAB	`.m`
Scilab	`.sce`
R	`.R`
Python	`.py`
gnuplot	`.gp` or `.plt`
IDL	`.idlbatch`
Maxima	`.mc` or `.mac`
bc	`bc`

	Comment symbol (rest of line)
MATLAB	`%`
Octave	`%` or `#`
Scilab	`//`
R	`#`
Python	`#`
gnuplot	`#`
IDL	`;`
Axiom	`--`
Mathematica	`(* .. *)`
Maple	`#`
Maxima	`/* .. */`

MuPAD	`//`
	`/* .. */`
	`# .. #`

Derive	`".."`
REDUCE	`%`
bc	`/* .. */`

	Import library functions
MATLAB	`% must be in MATLABPATH`
Octave	`% must be in LOADPATH`
Scilab	`getf('foo.sci')`
R	`library(RSvgDevice)`
Python	`from pylab import *`
Maxima	`load(SET);`

	Eval
MATLAB	`string='a=234';` `eval(string)`

Scilab	`eval()`
	`evstr()`
	`execstr()`

R	`string <- "a <- 234"` `eval(parse(text=string))`
Python	`string="a=234"` `eval(string)`

10.1 Loops

	for-statement
MATLAB	`for i=1:5; disp(i); end`
R	`for(i in 1:5) print(i)`
Python	`for i in range(1,6): print(i)`
IDL	`for k=1,5 do print,k`

Multiline for statements

MATLAB
```
for i=1:5
   disp(i)
   disp(i*2)
end
```

R
```
for(i in 1:5) {
    print(i)
    print(i*2)
}
```

Python
```
for i in range(1,6):
    print(i)
    print(i*2)
```

IDL
```
for k=1,5 do begin $
    print, i &$
    print, i*2 &$
end
```

10.2 Conditionals

	if-statement
MATLAB	`if 1>0 a=100; end`
R	`if (1>0) a <- 100`
Python	`if 1>0: a=100`
IDL	`if 1 gt 0 then a=100`

	if-else-statement
MATLAB	`if 1>0 a=100; else a=0; end`
IDL	`if 1 gt 0 then a=100 else a=0`

	Ternary operator (if?true:false)	
R	`ifelse(a>0,a,0)`	$a > 0?a : 0$
gnuplot	`a>0?a:0`	
IDL	`a>0?a:0`	

10.3 Debugging

	Most recent evaluated expression
MATLAB	`ans`
R	`.Last.value`
Axiom	`%`
Maxima	`%`

	List variables loaded into memory
MATLAB	`whos` or `who`
R	`objects()`
IDL	`help`

	Clear variable x from memory
MATLAB	`clear x` or `clear [all]`
R	`rm(x)`
Axiom	`)clear properties x`

	Print
MATLAB	`disp(a)`
R	`print(a)`
Python	`print a`
IDL	`print, a`

10.4 Working directory and OS

	List files in directory
MATLAB	`dir` or `ls`
R	`list.files()` or `dir()`
Python	`os.listdir(".")`
IDL	`dir`

	List script files in directory
MATLAB	`what`
R	`list.files(pattern="\.r$")`
Python	`grep.grep("*.py")`

	Displays the current working directory
MATLAB	`pwd`
R	`getwd()`
Python	`os.getwd()`
gnuplot	`pwd`
IDL	`sd`

	Change working directory
MATLAB	`cd foo`
Scilab	`chdir('foo')`
R	`setwd('foo')`
Python	`os.chdir('foo')`
gnuplot	`cd 'foo'`
IDL	`cd,'foo` or `sd,'foo`
Axiom	`)cd "foo"`

	Invoke a System Command
MATLAB	`!notepad`
Scilab	`host('notepad')`
Octave	`system("notepad")`
R	`system("notepad")`
Python	`os.system('notepad')`
	`os.popen('notepad')`
gnuplot	`!notepad`
IDL	`spawn,'notepad'`

[2] This document is still draft quality. Most 2d plot examples are made using Matplotlib, and 3d plots using R or Gnuplot.

[3] Version number and download URL for software used: Python 2.4.2, http://www.python.org/; NumPy 0.9.5, http://numeric.scipy.org/; Matplotlib 0.87, http://matplotlib.sf.net/; IPython 0.7.1, http://ipython.scipy.org/; R 2.1.1, http://www.r-project.org/; Octave 2.1.50, http://www.octave.org/; Scilab 4.0, http://www.scilab.org/; Gnuplot 4.0, http://www.gnuplot.info/; Maxima 5.9.1, http://maxima.sf.net/.

[4] For referencing: Gundersen, Vidar Bronken. *MATLAB commands in numerical Python* (Oslo/Norway, 2005), available from: http://mathesaurus.sf.net/

[5] Contributions are appreciated: The best way to do this is to edit the XML and submit patches to our tracker or forums.