

MATLAB commands in numerical Python (NumPy)

Copyright ©2006 Vidar Bronken Gundersen

Permission is granted to copy, distribute and/or modify this document as long as the above attribution is kept and the resulting work is distributed under a license identical to this one.

The idea of this document (and the corresponding XML instance) is to provide a quick reference¹ for switching from MATLAB to an open-source environment, such as Python, Scilab, Octave and Gnuplot, or R for numeric processing and data visualisation.

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

Time-stamp: 2007-11-09T16:46:36 vidar

1 Help

Desc.	MATLAB/Octave	Python	R
Browse help interactively	<code>doc</code> <code>Octave: help -i % browse with Info</code>	<code>help()</code>	<code>help.start()</code>
Help on using help	<code>help help or doc doc</code>	<code>help</code>	<code>help()</code>
Help for a function	<code>help plot</code>	<code>help(plot) or ?plot</code>	<code>help(plot) or ?plot</code>
Help for a toolbox/library package	<code>help splines or doc splines</code>	<code>help(pylab)</code>	<code>help(package='splines')</code>
Demonstration examples	<code>demo</code>		<code>demo()</code>
Example using a function			<code>example(plot)</code>

1.1 Searching available documentation

Desc.	MATLAB/Octave	Python	R
Search help files	<code>lookfor plot</code>		<code>help.search('plot')</code>
Find objects by partial name			<code>apropos('plot')</code>
List available packages	<code>help</code>	<code>help(); modules [Numeric]</code>	<code>library()</code>
Locate functions	<code>which plot</code>	<code>help(plot)</code>	<code>find(plot)</code>
List available methods for a function			<code>methods(plot)</code>

1.2 Using interactively

Desc.	MATLAB/Octave	Python	R
Start session	<code>Octave: octave -q</code>	<code>ipython -pylab</code>	<code>Rgui</code>
Auto completion	<code>Octave: TAB or M-?</code>	<code>TAB</code>	
Run code from file	<code>foo(.m)</code>	<code>execfile('foo.py') or run foo.py</code>	<code>source('foo.R')</code>
Command history	<code>Octave: history</code>	<code>hist -n</code>	<code>history()</code>
Save command history	<code>diary on [..] diary off</code>		<code>savehistory(file=".Rhistory")</code>
End session	<code>exit or quit</code>	<code>CTRL-D</code> <code>CTRL-Z # windows</code> <code>sys.exit()</code>	<code>q(save='no')</code>

2 Operators

Desc.	MATLAB/Octave	Python	R
Help on operator syntax	<code>help -</code>		<code>help(Syntax)</code>

¹References: 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).

2.1 Arithmetic operators

Desc.	MATLAB/Octave	Python	R
Assignment; defining a number	<code>a=1; b=2;</code>	<code>a=1; b=1</code>	<code>a<-1; b<-2</code>
Addition	<code>a + b</code>	<code>a + b</code> or <code>add(a,b)</code>	<code>a + b</code>
Subtraction	<code>a - b</code>	<code>a - b</code> or <code>subtract(a,b)</code>	<code>a - b</code>
Multiplication	<code>a * b</code>	<code>a * b</code> or <code>multiply(a,b)</code>	<code>a * b</code>
Division	<code>a / b</code>	<code>a / b</code> or <code>divide(a,b)</code>	<code>a / b</code>
Power, a^b	<code>a .^ b</code>	<code>a ** b</code> <code>power(a,b)</code> <code>pow(a,b)</code>	<code>a ^ b</code>
Remainder	<code>rem(a,b)</code>	<code>a % b</code> <code>remainder(a,b)</code> <code>fmod(a,b)</code>	<code>a %% b</code>
Integer division In place operation to save array creation overhead	Octave: <code>a+=1</code>	<code>a+=b</code> or <code>add(a,b,a)</code>	<code>a %/% b</code>
Factorial, $n!$	<code>factorial(a)</code>		<code>factorial(a)</code>

2.2 Relational operators

Desc.	MATLAB/Octave	Python	R
Equal	<code>a == b</code>	<code>a == b</code> or <code>equal(a,b)</code>	<code>a == b</code>
Less than	<code>a < b</code>	<code>a < b</code> or <code>less(a,b)</code>	<code>a < b</code>
Greater than	<code>a > b</code>	<code>a > b</code> or <code>greater(a,b)</code>	<code>a > b</code>
Less than or equal	<code>a <= b</code>	<code>a <= b</code> or <code>less_equal(a,b)</code>	<code>a <= b</code>
Greater than or equal	<code>a >= b</code>	<code>a >= b</code> or <code>greater_equal(a,b)</code>	<code>a >= b</code>
Not Equal	<code>a ~= b</code>	<code>a != b</code> or <code>not_equal(a,b)</code>	<code>a != b</code>

2.3 Logical operators

Desc.	MATLAB/Octave	Python	R
Short-circuit logical AND	<code>a && b</code>	<code>a and b</code>	<code>a && b</code>
Short-circuit logical OR	<code>a b</code>	<code>a or b</code>	<code>a b</code>
Element-wise logical AND	<code>a & b</code> or <code>and(a,b)</code>	<code>logical_and(a,b)</code> or <code>a and b</code>	<code>a & b</code>
Element-wise logical OR	<code>a b</code> or <code>or(a,b)</code>	<code>logical_or(a,b)</code> or <code>a or b</code>	<code>a b</code>
Logical EXCLUSIVE OR	<code>xor(a, b)</code>	<code>logical_xor(a,b)</code>	<code>xor(a, b)</code>
Logical NOT	<code>~a</code> or <code>not(a)</code> Octave: <code>~a</code> or <code>!a</code>	<code>logical_not(a)</code> or <code>not a</code>	<code>!a</code>
True if any element is nonzero	<code>any(a)</code>		
True if all elements are nonzero	<code>all(a)</code>		

2.4 root and logarithm

Desc.	MATLAB/Octave	Python	R
Square root	<code>sqrt(a)</code>	<code>math.sqrt(a)</code>	<code>sqrt(a)</code>
Logarithm, base e (natural)	<code>log(a)</code>	<code>math.log(a)</code>	<code>log(a)</code> $\ln a = \log_e a$
Logarithm, base 10	<code>log10(a)</code>	<code>math.log10(a)</code>	<code>log10(a)</code> $\log_{10} a$
Logarithm, base 2 (binary)	<code>log2(a)</code>	<code>math.log(a, 2)</code>	<code>log2(a)</code> $\log_2 a$
Exponential function	<code>exp(a)</code>	<code>math.exp(a)</code>	<code>exp(a)</code> e^a

2.5 Round off

Desc.	MATLAB/Octave	Python	R
Round	<code>round(a)</code>	<code>round(a)</code> or <code>math.round(a)</code>	<code>round(a)</code>
Round up	<code>ceil(a)</code>	<code>ceil(a)</code>	<code>ceil(a)</code>
Round down	<code>floor(a)</code>	<code>floor(a)</code>	<code>floor(a)</code>
Round towards zero	<code>fix(a)</code>	<code>fix(a)</code>	

2.6 Mathematical constants

Desc.	MATLAB/Octave	Python	R
$\pi = 3.141592$	<code>pi</code>	<code>math.pi</code>	<code>pi</code>
$e = 2.718281$	<code>exp(1)</code>	<code>math.e</code> or <code>math.exp(1)</code>	<code>exp(1)</code>

2.6.1 Missing values; IEEE-754 floating point status flags

Desc.	MATLAB/Octave	Python	R
Not a Number	<code>NaN</code>	<code>nan</code>	
Infinity, ∞	<code>Inf</code>	<code>inf</code>	
Infinity, $+\infty$		<code>plus_inf</code>	
Infinity, $-\infty$		<code>minus_inf</code>	
Plus zero, $+0$		<code>plus_zero</code>	
Minus zero, -0		<code>minus_zero</code>	

2.7 Complex numbers

Desc.	MATLAB/Octave	Python	R
Imaginary unit	<code>i</code>	<code>z = 1j</code>	<code>1i</code>
A complex number, $3 + 4i$	<code>z = 3+4i</code>	<code>z = 3+4j</code> or <code>z = complex(3,4)</code>	<code>z <- 3+4i</code>
Absolute value (modulus)	<code>abs(z)</code>	<code>abs(3+4j)</code>	<code>abs(3+4i)</code> or <code>Mod(3+4i)</code>
Real part	<code>real(z)</code>	<code>z.real</code>	<code>Re(3+4i)</code>
Imaginary part	<code>imag(z)</code>	<code>z.imag</code>	<code>Im(3+4i)</code>
Argument	<code>arg(z)</code>		<code>Arg(3+4i)</code>
Complex conjugate	<code>conj(z)</code>	<code>z.conj(); z.conjugate()</code>	<code>Conj(3+4i)</code>

$$i = \sqrt{-1}$$

2.8 Trigonometry

Desc.	MATLAB/Octave	Python	R
Arctangent, $\arctan(b/a)$	<code>atan(a,b)</code>	<code>atan2(b,a)</code>	<code>atan2(b,a)</code>
Hypotenuse; Euclidean distance		<code>hypot(x,y)</code>	

$$\sqrt{x^2 + y^2}$$

2.9 Generate random numbers

Desc.	MATLAB/Octave	Python	R
Uniform distribution	<code>rand(1,10)</code>	<code>random.random((10,))</code> <code>random.uniform((10,))</code>	<code>runif(10)</code>
Uniform: Numbers between 2 and 7	<code>2+5*rand(1,10)</code>	<code>random.uniform(2,7,(10,))</code>	<code>runif(10, min=2, max=7)</code>
Uniform: 6,6 array	<code>rand(6)</code>	<code>random.uniform(0,1,(6,6))</code>	<code>matrix(runif(36),6)</code>
Normal distribution	<code>randn(1,10)</code>	<code>random.standard_normal((10,))</code>	<code>rnorm(10)</code>

3 Vectors

Desc.
 Row vector, $1 \times n$ -matrix
 Column vector, $m \times 1$ -matrix

MATLAB/Octave
`a=[2 3 4 5];`
`adash=[2 3 4 5]';`

Python
`a=array([2,3,4,5])`
`array([2,3,4,5])[:,NewAxis]`
`array([2,3,4,5]).reshape(-1,1)`
`r_[1:10,'c']`

R
`a <- c(2,3,4,5)`
`adash <- t(c(2,3,4,5))`

3.1 Sequences

Desc.
`1,2,3, ... ,10`

MATLAB/Octave
`1:10`

Python
`arange(1,11, dtype=Float)`
`range(1,11)`
`arange(10.)`
`arange(1,11,3)`
`arange(10,0,-1)`
`arange(10,0,-3)`
`linspace(1,10,7)`
`a[::-1] or`
`a.fill(3), a[:] = 3`

R
`seq(10) or 1:10`
`seq(0,length=10)`
`seq(1,10,by=3)`
`seq(10,1) or 10:1`
`seq(from=10,to=1,by=-3)`
`seq(1,10,length=7)`
`rev(a)`

`0.0,1.0,2.0, ... ,9.0`
`1,4,7,10`
`10,9,8, ... ,1`
`10,7,4,1`
 Linearly spaced vector of n=7 points
 Reverse
 Set all values to same scalar value

`0:9`
`1:3:10`
`10:-1:1`
`10:-3:1`
`linspace(1,10,7)`
`reverse(a)`
`a(:) = 3`

3.2 Concatenation (vectors)

Desc.
 Concatenate two vectors

MATLAB/Octave
`[a a]`
`[1:4 a]`

Python
`concatenate((a,a))`
`concatenate((range(1,5),a), axis=1)`

R
`c(a,a)`
`c(1:4,a)`

3.3 Repeating

Desc.
`1 2 3, 1 2 3`
`1 1 1, 2 2 2, 3 3 3`
`1, 2 2, 3 3 3`

MATLAB/Octave
`[a a]`

Python
`concatenate((a,a))`
`a.repeat(3) or`
`a.repeat(a) or`

R
`rep(a,times=2)`
`rep(a,each=3)`
`rep(a,a)`

3.4 Miss those elements out

Desc.
 miss the first element
 miss the tenth element
 miss 1,4,7, ...
 last element
 last two elements

MATLAB/Octave
`a(2:end)`
`a([1:9])`
`a(end)`
`a(end-1:end)`

Python
`a[1:]`
`a[-1]`
`a[-2:]`

R
`a[-1]`
`a[-10]`
`a[-seq(1,50,3)]`

3.5 Maximum and minimum

Desc.
 pairwise max
 max of all values in two vectors

MATLAB/Octave
`max(a,b)`
`max([a b])`
`[v,i] = max(a)`

Python
`maximum(a,b)`
`concatenate((a,b)).max()`
`v,i = a.max(0),a.argmax(0)`

R
`pmax(a,b)`
`max(a,b)`
`v <- max(a) ; i <- which.max(a)`

3.6 Vector multiplication

Desc. Multiply two vectors Vector dot product, $u \cdot v$	MATLAB/Octave <code>a.*a</code> <code>dot(u,v)</code>	Python <code>a*a</code> <code>dot(u,v)</code>	R <code>a*a</code>
--	---	---	-----------------------

4 Matrices

Desc. Define a matrix	MATLAB/Octave <code>a = [2 3;4 5]</code>	Python <code>a = array([[2,3],[4,5]])</code>	R <code>rbind(c(2,3),c(4,5))</code> <code>array(c(2,3,4,5), dim=c(2,2))</code>
--------------------------	---	---	--

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

4.1 Concatenation (matrices); rbind and cbind

Desc. Bind rows	MATLAB/Octave <code>[a ; b]</code>	Python <code>concatenate((a,b), axis=0)</code> <code>vstack((a,b))</code>	R <code>rbind(a,b)</code>
Bind columns	<code>[a , b]</code>	<code>concatenate((a,b), axis=1)</code> <code>hstack((a,b))</code>	<code>cbind(a,b)</code>
Bind slices (three-way arrays)		<code>concatenate((a,b), axis=2)</code> <code>dstack((a,b))</code>	
Concatenate matrices into one vector	<code>[a(:), b(:)]</code>	<code>concatenate((a,b), axis=None)</code>	
Bind rows (from vectors)	<code>[1:4 ; 1:4]</code>	<code>concatenate((r_[1:5],r_[1:5])).reshape(2,1:4)</code>	<code>rbind(1:4,1:4)</code>
Bind columns (from vectors)	<code>[1:4 ; 1:4]'</code>	<code>vstack((r_[1:5],r_[1:5]))</code>	<code>cbind(1:4,1:4)</code>

4.2 Array creation

Desc. 0 filled array	MATLAB/Octave <code>zeros(3,5)</code>	Python <code>zeros((3,5),Float)</code>	R <code>matrix(0,3,5)</code> or <code>array(0,c(3,5))</code>
0 filled array of integers		<code>zeros((3,5))</code>	
1 filled array	<code>ones(3,5)</code>	<code>ones((3,5),Float)</code>	<code>matrix(1,3,5)</code> or <code>array(1,c(3,5))</code>
Any number filled array	<code>ones(3,5)*9</code>		<code>matrix(9,3,5)</code> or <code>array(9,c(3,5))</code>
Identity matrix	<code>eye(3)</code>	<code>identity(3)</code>	<code>diag(1,3)</code>
Diagonal	<code>diag([4 5 6])</code>	<code>diag((4,5,6))</code>	<code>diag(c(4,5,6))</code>
Magic squares; Lo Shu	<code>magic(3)</code>		
Empty array		<code>a = empty((3,3))</code>	

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \\ 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

4.3 Reshape and flatten matrices

Desc.	MATLAB/Octave	Python	R
Reshaping (rows first)	<code>reshape(1:6,3,2)'</code> ;	<code>arange(1,7).reshape(2,-1)</code> <code>a.setshape(2,3)</code>	<code>matrix(1:6,nrow=3,byrow=T)</code>
Reshaping (columns first)	<code>reshape(1:6,2,3)</code> ;	<code>arange(1,7).reshape(-1,2).transpose()</code>	<code>matrix(1:6,nrow=2)</code> <code>array(1:6,c(2,3))</code>
Flatten to vector (by rows, like comics)	<code>a'(:)</code>	<code>a.flatten()</code> or	<code>as.vector(t(a))</code>
Flatten to vector (by columns)	<code>a(:)</code>	<code>a.flatten(1)</code>	<code>as.vector(a)</code>
Flatten upper triangle (by columns)	<code>vech(a)</code>		<code>a[row(a) <= col(a)]</code>

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{bmatrix}$$

4.4 Shared data (slicing)

Desc.	MATLAB/Octave	Python	R
Copy of a	<code>b = a</code>	<code>b = a.copy()</code>	<code>b = a</code>

4.5 Indexing and accessing elements (Python: slicing)

Desc.	MATLAB/Octave	Python	R
Input is a 3,4 array	<code>a = [11 12 13 14 ... 21 22 23 24 ... 31 32 33 34]</code>	<code>a = array([[11, 12, 13, 14], [21, 22, 23, 24], [31, 32, 33, 34]])</code>	<code>a <- rbind(c(11, 12, 13, 14), c(21, 22, 23, 24), c(31, 32, 33, 34))</code>
Element 2,3 (row,col)	<code>a(2,3)</code>	<code>a[1,2]</code>	<code>a[2,3]</code>
First row	<code>a(1,:)</code>	<code>a[0,]</code>	<code>a[1,]</code>
First column	<code>a(:,1)</code>	<code>a[:,0]</code>	<code>a[,1]</code>
Array as indices	<code>a([1 3],[1 4]);</code>	<code>a.take([0,2]).take([0,3], axis=1)</code>	
All, except first row	<code>a(2:end,:)</code>	<code>a[1:,]</code>	<code>a[-1,]</code>
Last two rows	<code>a(end-1:end,:)</code>	<code>a[-2:,]</code>	
Strides: Every other row	<code>a(1:2:end,:)</code>	<code>a[::2,:]</code>	
Third in last dimension (axis)		<code>a[... ,2]</code>	
All, except row,column (2,3)			<code>a[-2,-3]</code>
Remove one column	<code>a(:,[1 3 4])</code>	<code>a.take([0,2,3],axis=1)</code>	<code>a[, -2]</code>
Diagonal		<code>a.diagonal(offset=0)</code>	

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

$$a_{23}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & & & \\ a_{31} & & & \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{bmatrix}$$

$$\begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{11} & a_{12} & a_{13} & a_{14} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{31} & a_{33} & a_{34} \\ a_{11} & a_{13} & a_{14} \\ a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{22} & a_{33} & a_{44} \end{bmatrix}$$

4.6 Assignment

Desc.	MATLAB/Octave	Python	R
Clipping: Replace all elements over 90	<code>a(:,1) = 99</code> <code>a(:,1) = [99 98 97]'</code> <code>a(a>90) = 90;</code>	<code>a[:,0] = 99</code> <code>a[:,0] = array([99,98,97])</code> <code>(a>90).choose(a,90)</code> <code>a.clip(min=None, max=90)</code>	<code>a[,1] <- 99</code> <code>a[,1] <- c(99,98,97)</code> <code>a[a>90] <- 90</code>
Clip upper and lower values		<code>a.clip(min=2, max=5)</code>	

4.7 Transpose and inverse

Desc.	MATLAB/Octave	Python	R
Transpose	<code>a'</code>	<code>a.conj().transpose()</code>	<code>t(a)</code>
Non-conjugate transpose	<code>a.' or transpose(a)</code>	<code>a.transpose()</code>	
Determinant	<code>det(a)</code>	<code>linalg.det(a) or</code>	<code>det(a)</code>
Inverse	<code>inv(a)</code>	<code>linalg.inv(a) or</code>	<code>solve(a)</code>
Pseudo-inverse	<code>pinv(a)</code>	<code>linalg.pinv(a)</code>	<code>ginv(a)</code>
Norms	<code>norm(a)</code>	<code>norm(a)</code>	
Eigenvalues	<code>eig(a)</code>	<code>linalg.eig(a)[0]</code>	<code>eigen(a)\$values</code>
Singular values	<code>svd(a)</code>	<code>linalg.svd(a)</code>	<code>svd(a)\$d</code>
Cholesky factorization	<code>chol(a)</code>	<code>linalg.cholesky(a)</code>	
Eigenvectors	<code>[v,1] = eig(a)</code>	<code>linalg.eig(a)[1]</code>	<code>eigen(a)\$vectors</code>
Rank	<code>rank(a)</code>	<code>rank(a)</code>	<code>rank(a)</code>

4.8 Sum

Desc.	MATLAB/Octave	Python	R
Sum of each column	<code>sum(a)</code>	<code>a.sum(axis=0)</code>	<code>apply(a,2,sum)</code>
Sum of each row	<code>sum(a')</code>	<code>a.sum(axis=1)</code>	<code>apply(a,1,sum)</code>
Sum of all elements	<code>sum(sum(a))</code>	<code>a.sum()</code>	<code>sum(a)</code>
Sum along diagonal		<code>a.trace(offset=0)</code>	
Cumulative sum (columns)	<code>cumsum(a)</code>	<code>a.cumsum(axis=0)</code>	<code>apply(a,2,cumsum)</code>

4.9 Sorting

Desc.	MATLAB/Octave	Python	R
Example data	<code>a = [4 3 2 ; 2 8 6 ; 1 4 7]</code>	<code>a = array([[4,3,2],[2,8,6],[1,4,7]])</code>	
Flat and sorted	<code>sort(a(:))</code>	<code>a.ravel().sort()</code> or	<code>t(sort(a))</code>
Sort each column	<code>sort(a)</code>	<code>a.sort(axis=0)</code> or <code>msort(a)</code>	<code>apply(a,2,sort)</code>
Sort each row	<code>sort(a')'</code>	<code>a.sort(axis=1)</code>	<code>t(apply(a,1,sort))</code>
Sort rows (by first row)	<code>sortrows(a,1)</code>	<code>a[a[:,0].argsort(),]</code>	
Sort, return indices		<code>a.ravel().argsort()</code>	<code>order(a)</code>
Sort each column, return indices		<code>a.argsort(axis=0)</code>	
Sort each row, return indices		<code>a.argsort(axis=1)</code>	

```

[ 4 3 2 ]
[ 2 8 6 ]
[ 1 4 7 ]
[ 1 2 2 ]
[ 3 4 4 ]
[ 6 7 8 ]
[ 1 3 2 ]
[ 2 4 6 ]
[ 4 8 7 ]
[ 2 3 4 ]
[ 2 6 8 ]
[ 1 4 7 ]
[ 1 4 7 ]
[ 2 8 6 ]
[ 4 3 2 ]
    
```

4.10 Maximum and minimum

Desc.	MATLAB/Octave	Python	R
max in each column	<code>max(a)</code>	<code>a.max(0)</code> or <code>amax(a [,axis=0])</code>	<code>apply(a,2,max)</code>
max in each row	<code>max(a')</code>	<code>a.max(1)</code> or <code>amax(a, axis=1)</code>	<code>apply(a,1,max)</code>
max in array	<code>max(max(a))</code>	<code>a.max()</code> or	<code>max(a)</code>
return indices, i	<code>[v i] = max(a)</code>		<code>i <- apply(a,1,which.max)</code>
pairwise max	<code>max(b,c)</code>	<code>maximum(b,c)</code>	<code>pmax(b,c)</code>
max-to-min range	<code>cummax(a)</code>	<code>a.ptp(); a.ptp(0)</code>	<code>apply(a,2,cummax)</code>

4.11 Matrix manipulation

Desc.	MATLAB/Octave	Python	R
Flip left-right	<code>fliplr(a)</code>	<code>fliplr(a)</code> or <code>a[:,::-1]</code>	<code>a[,4:1]</code>
Flip up-down	<code>flipud(a)</code>	<code>flipud(a)</code> or <code>a[::-1,]</code>	<code>a[3:1,]</code>
Rotate 90 degrees	<code>rot90(a)</code>	<code>rot90(a)</code>	
Repeat matrix: [a a a ; a a a]	<code>repmat(a,2,3)</code> <code>Octave: kron(ones(2,3),a)</code>	<code>kron(ones((2,3)),a)</code>	<code>kroncker(matrix(1,2,3),a)</code>
Triangular, upper	<code>triu(a)</code>	<code>triu(a)</code>	<code>a[lower.tri(a)] <- 0</code>
Triangular, lower	<code>tril(a)</code>	<code>tril(a)</code>	<code>a[upper.tri(a)] <- 0</code>

4.12 Equivalent to "size"

Desc.	MATLAB/Octave	Python	R
Matrix dimensions	<code>size(a)</code>	<code>a.shape</code> or <code>a.getshape()</code>	<code>dim(a)</code>
Number of columns	<code>size(a,2)</code> or <code>length(a)</code>	<code>a.shape[1]</code> or <code>size(a, axis=1)</code>	<code>ncol(a)</code>
Number of elements	<code>length(a(:))</code>	<code>a.size</code> or <code>size(a[, axis=None])</code>	<code>prod(dim(a))</code>
Number of dimensions	<code>ndims(a)</code>	<code>a.ndim</code>	
Number of bytes used in memory		<code>a.nbytes</code>	<code>object.size(a)</code>

4.13 Matrix- and elementwise- multiplication

Desc.	MATLAB/Octave	Python	R
Elementwise operations	<code>a .* b</code>	<code>a * b</code> or <code>multiply(a,b)</code>	<code>a * b</code>
Matrix product (dot product)	<code>a * b</code>	<code>matrixmultiply(a,b)</code>	<code>a %%% b</code>
Inner matrix vector multiplication $a \cdot b'$		<code>inner(a,b)</code> or	
Outer product		<code>outer(a,b)</code> or	<code>outer(a,b)</code> or <code>a %o% b</code>
Cross product			<code>crossprod(a,b)</code> or <code>t(a) %%% b</code>
Kronecker product	<code>kron(a,b)</code>	<code>kron(a,b)</code>	<code>kron(a,b)</code>
Matrix division, $b \cdot a^{-1}$	<code>a / b</code>		
Left matrix division, $b^{-1} \cdot a$ (solve linear equations)	<code>a \ b</code>	<code>linalg.solve(a,b)</code>	<code>solve(a,b)</code>
Vector dot product		<code>vdot(a,b)</code>	
Cross product		<code>cross(a,b)</code>	

$$\begin{bmatrix} 1 & 5 \\ 9 & 16 \end{bmatrix} \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix} = \begin{bmatrix} 55 & 22 \\ 123 & 352 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 4 & 8 & 12 & 16 \end{bmatrix} = \begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 4 & 6 & 8 \\ 3 & 6 & 4 & 8 \\ 9 & 12 & 12 & 16 \end{bmatrix}$$

$Ax = b$

4.14 Find; conditional indexing

Desc.	MATLAB/Octave	Python	R
Non-zero elements, indices	<code>find(a)</code>	<code>a.ravel().nonzero()</code>	<code>which(a != 0)</code>
Non-zero elements, array indices	<code>[i j] = find(a)</code>	<code>(i,j) = a.nonzero()</code> <code>(i,j) = where(a!=0)</code>	<code>which(a != 0, arr.ind=T)</code>
Vector of non-zero values	<code>[i j v] = find(a)</code>	<code>v = a.compress((a!=0).flat)</code> <code>v = extract(a!=0,a)</code>	<code>ij <- which(a != 0, arr.ind=T); v <- a[ij]</code>
Condition, indices	<code>find(a>5.5)</code>	<code>(a>5.5).nonzero()</code>	<code>which(a>5.5)</code>
Return values		<code>a.compress((a>5.5).flat)</code>	<code>ij <- which(a>5.5, arr.ind=T); v <- a[ij]</code>
Zero out elements above 5.5 Replace values	<code>a .* (a>5.5)</code>	<code>where(a>5.5,0,a)</code> or <code>a * (a>5.5)</code> <code>a.put(2,indices)</code>	

5 Multi-way arrays

Desc.	MATLAB/Octave	Python	R
Define a 3-way array	<code>a = cat(3, [1 2; 1 2],[3 4; 3 4]);</code> <code>a(1, :, :)</code>	<code>a = array([[1,2],[1,2]], [[3,4],[3,4]])</code> <code>a[0, ...]</code>	

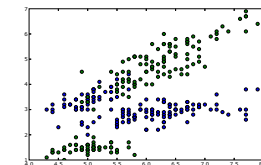
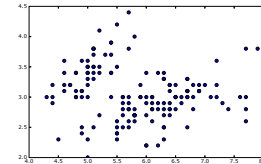
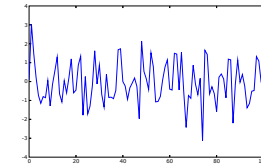
6 File input and output

Desc.	MATLAB/Octave	Python	R
Reading from a file (2d)	<code>f = load('data.txt')</code>	<code>f = fromfile("data.txt")</code> <code>f = load("data.txt")</code>	<code>f <- read.table("data.txt")</code>
Reading from a file (2d)	<code>f = load('data.txt')</code>	<code>f = load("data.txt")</code>	<code>f <- read.table("data.txt")</code>
Reading from a CSV file (2d)	<code>x = dlmread('data.csv', ';')</code>	<code>f = load('data.csv', delimiter=';')</code>	<code>f <- read.table(file="data.csv", sep=";")</code>
Writing to a file (2d)	<code>save -ascii data.txt f</code>	<code>save('data.csv', f, fmt='%.6f', delimiter=';')file="data.txt")</code>	<code>write(f,file="data.txt")</code>
Writing to a file (1d)		<code>f.tofile(file='data.csv', format='%.6f', sep=';')</code>	
Reading from a file (1d)		<code>f = fromfile(file='data.csv', sep=';')</code>	

7 Plotting

7.1 Basic x-y plots

Desc.	MATLAB/Octave	Python	R
1d line plot	<code>plot(a)</code>	<code>plot(a)</code>	<code>plot(a, type="l")</code>
2d scatter plot	<code>plot(x(:,1),x(:,2),'o')</code>	<code>plot(x[:,0],x[:,1],'o')</code>	<code>plot(x[,1],x[,2])</code>
Two graphs in one plot	<code>plot(x1,y1, x2,y2)</code>	<code>plot(x1,y1,'bo', x2,y2,'go')</code>	<code>plot(x1,y1)</code>
Overplotting: Add new plots to current	<code>plot(x1,y1)</code> <code>hold on</code>	<code>plot(x1,y1,'o')</code> <code>plot(x2,y2,'o')</code>	<code>matplot(x2,y2,add=T)</code>
subplots	<code>plot(x2,y2)</code> <code>subplot(211)</code>	<code>show() # as normal</code> <code>subplot(211)</code>	
Plotting symbols and color	<code>plot(x,y,'ro-')</code>	<code>plot(x,y,'ro-')</code>	<code>plot(x,y,type="b",col="red")</code>



7.1.1 Axes and titles

Desc.
 Turn on grid lines
 1:1 aspect ratio

MATLAB/Octave
`grid on`
`axis equal`
 Octave:
`axis('equal')`
`replot`

Python
`grid()`
`figure(figsize=(6,6))`

R
`grid()`
`plot(c(1:10,10:1), asp=1)`

Set axes manually
 Axis labels and titles

`axis([0 10 0 5])`
`title('title')`
`xlabel('x-axis')`
`ylabel('y-axis')`

`axis([0, 10, 0, 5])`

`text(2,25,'hello')`

`plot(x,y, xlim=c(0,10), ylim=c(0,5))`
`plot(1:10, main="title",`
`xlab="x-axis", ylab="y-axis")`

Insert text

7.1.2 Log plots

Desc.
 logarithmic y-axis
 logarithmic x-axis
 logarithmic x and y axes

MATLAB/Octave
`semilogy(a)`
`semilogx(a)`
`loglog(a)`

Python
`semilogy(a)`
`semilogx(a)`
`loglog(a)`

R
`plot(x,y, log="y")`
`plot(x,y, log="x")`
`plot(x,y, log="xy")`

7.1.3 Filled plots and bar plots

Desc.

MATLAB/Octave

Python

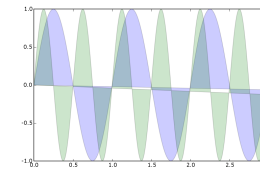
R

Filled plot

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

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

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



Stem-and-Leaf plot

`stem(x[,3])`

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

7.1.4 Functions

Desc.
 Defining functions

MATLAB/Octave
`f = inline('sin(x/3) - cos(x/5)')`

Python

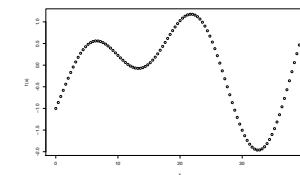
R
`f <- function(x) sin(x/3) - cos(x/5)` $f(x) = \sin\left(\frac{x}{3}\right) - \cos\left(\frac{x}{5}\right)$

Plot a function for given range

`ezplot(f, [0,40])`
`fplot('sin(x/3) - cos(x/5)', [0,40])`
 Octave: % no ezplot

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

`plot(f, xlim=c(0,40), type='p')`



7.2 Polar plots

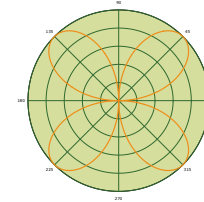
Desc.

```
MATLAB/Octave
theta = 0:.001:2*pi;
r = sin(2*theta);
```

```
Python
theta = arange(0,2*pi,0.001)
r = sin(2*theta)
```

R

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



```
polar(theta, rho)
```

```
polar(theta, rho)
```

7.3 Histogram plots

Desc.

```
MATLAB/Octave
hist(randn(1000,1))
hist(randn(1000,1), -4:4)

plot(sort(a))
```

Python

```
R
hist(rnorm(1000))
hist(rnorm(1000), breaks= -4:4)
hist(rnorm(1000), breaks=c(seq(-5,0,0.25), seq(0.5,5,0.5)), freq=F)
plot(apply(a,1,sort),type="l")
```

7.4 3d data

7.4.1 Contour and image plots

Desc.

MATLAB/Octave

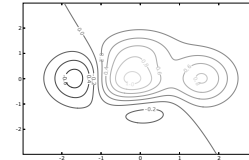
Python

R

Contour plot

`contour(z)`

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

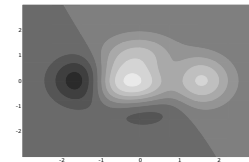


Filled contour plot

`contourf(z); colormap(gray)`

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

```
filled.contour(x,y,z,
              nlevels=7, color=gray.colors)
```



Plot image data

`image(z)`
`colormap(gray)`

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

```
image(z, col=gray.colors(256))
```

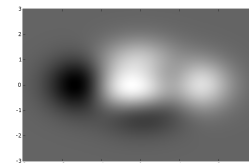
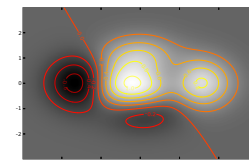


Image with contours
Direction field vectors

`quiver()`

```
# imshow() and contour() as above
quiver()
```



7.4.2 Perspective plots of surfaces over the x-y plane

Desc.

MATLAB/Octave

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

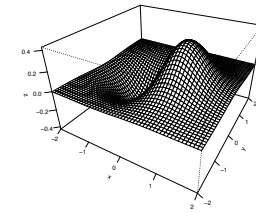
Python

```
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)
```

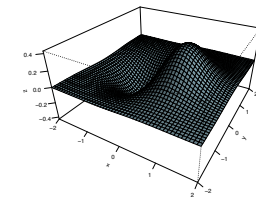
$$f(x,y) = xe^{-x^2-y^2}$$



Mesh plot

mesh(z)

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



Surface plot

surf(x,y,z) or surf1(x,y,z)
 Octave: % no surf1()

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

7.4.3 Scatter (cloud) plots

Desc.

MATLAB/Octave

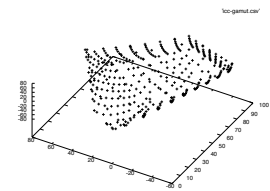
Python

R

3d scatter plot

plot3(x,y,z,'k+')

cloud(z~x*y)



7.5 Save plot to a graphics file

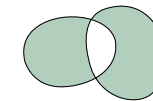
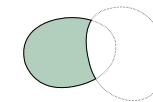
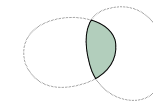
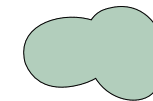
Desc.	MATLAB/Octave	Python	R
PostScript	<code>plot(1:10)</code> <code>print -depsc2 foo.eps</code> Octave: <code>gset output "foo.eps"</code> <code>gset terminal postscript eps</code> <code>plot(1:10)</code>	<code>savefig('foo.eps')</code>	<code>postscript(file="foo.eps")</code> <code>plot(1:10)</code> <code>dev.off()</code>
PDF		<code>savefig('foo.pdf')</code>	<code>pdf(file='foo.pdf')</code>
SVG (vector graphics for www)		<code>savefig('foo.svg')</code>	<code>devSVG(file='foo.svg')</code>
PNG (raster graphics)	<code>print -dpng foo.png</code>	<code>savefig('foo.png')</code>	<code>png(filename = "Rplot%03d.png"</code>

8 Data analysis

8.1 Set membership operators

Desc.	MATLAB/Octave	Python	R
Create sets	<code>a = [1 2 2 5 2];</code> <code>b = [2 3 4];</code>	<code>a = array([1,2,2,5,2])</code> <code>b = array([2,3,4])</code> <code>a = set([1,2,2,5,2])</code> <code>b = set([2,3,4])</code>	<code>a <- c(1,2,2,5,2)</code> <code>b <- c(2,3,4)</code>
Set unique	<code>unique(a)</code>	<code>uniqueid(a)</code> <code>unique(a)</code> <code>set(a)</code>	<code>unique(a)</code>
Set union	<code>union(a,b)</code>	<code>union1d(a,b)</code> <code>a.union(b)</code>	<code>union(a,b)</code>
Set intersection	<code>intersect(a,b)</code>	<code>intersect1d(a)</code> <code>a.intersection(b)</code>	<code>intersect(a,b)</code>
Set difference	<code>setdiff(a,b)</code>	<code>setdiff1d(a,b)</code> <code>a.difference(b)</code>	<code>setdiff(a,b)</code>
Set exclusion	<code>setxor(a,b)</code>	<code>setxor1d(a,b)</code> <code>a.symmetric_difference(b)</code>	<code>setdiff(union(a,b),intersect(a,b))</code>
True for set member	<code>ismember(2,a)</code>	<code>2 in a</code> <code>setmember1d(2,a)</code> <code>contains(a,2)</code>	<code>is.element(2,a)</code> or <code>2 %in% a</code>

[1 2 5]



8.2 Statistics

Desc.	MATLAB/Octave	Python	R
Average	<code>mean(a)</code>	<code>a.mean(axis=0)</code> <code>mean(a [,axis=0])</code>	<code>apply(a,2,mean)</code>
Median	<code>median(a)</code>	<code>median(a) or median(a [,axis=0])</code>	<code>apply(a,2,median)</code>
Standard deviation	<code>std(a)</code>	<code>a.std(axis=0) or std(a [,axis=0])</code>	<code>apply(a,2,sd)</code>
Variance	<code>var(a)</code>	<code>a.var(axis=0) or var(a)</code>	<code>apply(a,2,var)</code>
Correlation coefficient	<code>corr(x,y)</code>	<code>correlate(x,y) or corrccoef(x,y)</code>	<code>cor(x,y)</code>
Covariance	<code>cov(x,y)</code>	<code>cov(x,y)</code>	<code>cov(x,y)</code>

8.3 Interpolation and regression

Desc.	MATLAB/Octave	Python	R
Straight line fit	<code>z = polyval(polyfit(x,y,1),x)</code> <code>plot(x,y,'o', x,z ,'-')</code>	<code>(a,b) = polyfit(x,y,1)</code> <code>plot(x,y,'o', x,a*x+b,'-')</code>	<code>z <- lm(y~x)</code> <code>plot(x,y)</code> <code>abline(z)</code>
Linear least squares $y = ax + b$	<code>a = x\y</code>	<code>linalg.lstsq(x,y)</code>	<code>solve(a,b)</code>
Polynomial fit	<code>polyfit(x,y,3)</code>	<code>polyfit(x,y,3)</code>	

8.4 Non-linear methods

8.4.1 Polynomials, root finding

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

8.4.2 Differential equations

Desc.	MATLAB/Octave	Python	R
Discrete difference function and approximate derivative	<code>diff(a)</code>	<code>diff(x, n=1, axis=0)</code>	
Solve differential equations			

8.5 Fourier analysis

Desc.	MATLAB/Octave	Python	R
Fast fourier transform	<code>fft(a)</code>	<code>fft(a) or</code>	<code>fft(a)</code>
Inverse fourier transform	<code>ifft(a)</code>	<code>ifft(a) or</code>	<code>fft(a, inverse=TRUE)</code>
Linear convolution		<code>convolve(x,y)</code>	

9 Symbolic algebra; calculus

Desc.	MATLAB/Octave	Python	R
Factorization	<code>factor()</code>		

10 Programming

Desc.	MATLAB/Octave	Python	R
Script file extension	.m	.py	.R
Comment symbol (rest of line)	%	#	#
Import library functions	Octave: % or # % must be in MATLABPATH Octave: % must be in LOADPATH	from pylab import *	library(RSvgDevice)
Eval	string='a=234'; eval(string)	string="a=234" eval(string)	string <- "a <- 234" eval(parse(text=string))

10.1 Loops

Desc.	MATLAB/Octave	Python	R
for-statement	for i=1:5; disp(i); end	for i in range(1,6): print(i)	for(i in 1:5) print(i)
Multiline for statements	for i=1:5 disp(i) disp(i*2) end	for i in range(1,6): print(i) print(i*2)	for(i in 1:5) { print(i) print(i*2) }

10.2 Conditionals

Desc.	MATLAB/Octave	Python	R
if-statement	if 1>0 a=100; end	if 1>0: a=100	if (1>0) a <- 100
if-else-statement	if 1>0 a=100; else a=0; end		ifelse(a>0,a,0)
Ternary operator (if?true:false)			a > 0?a:0

10.3 Debugging

Desc.	MATLAB/Octave	Python	R
Most recent evaluated expression	ans		.Last.value
List variables loaded into memory	whos or who		objects()
Clear variable x from memory	clear x or clear [all]		rm(x)
Print	disp(a)	print a	print(a)

10.4 Working directory and OS

Desc.	MATLAB/Octave	Python	R
List files in directory	dir or ls	os.listdir(".")	list.files() or dir()
List script files in directory	what	grep.grep("*.py")	list.files(pattern=".r\$")
Displays the current working directory	pwd	os.getcwd()	getwd()
Change working directory	cd foo	os.chdir('foo')	setwd('foo')
Invoke a System Command	!notepad	os.system('notepad')	system("notepad")
	Octave: system("notepad")	os.popen('notepad')	

²This document is still draft quality. Most shown 2d plots are made using Matplotlib, and 3d plots using R and Gnuplot, provided as examples only.

³Version numbers 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/>.

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

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