

# IDL commands in numerical Python

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.

Contributor: Gary Ruben

The idea of this document (and the corresponding XML instance) is to provide a quick reference<sup>1</sup> for switching from IDL to an open-source environment, such as GDL, Python, R or Octave and Gnuplot for numeric processing and data visualisation.

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

## 1 Help

Desc.	IDL	Python	MATLAB/Octave
Browse help interactively	?	help()	doc Octave: help -i % browse with Info
Help on using help	?help	help	help help or doc doc
Help for a function	?plot or man, 'plot	help(plot) or ?plot	help plot
Help for a toolbox/library package		help(pylab)	help splines or doc splines
Demonstration examples	demo		demo

### 1.1 Searching available documentation

Desc.	IDL	Python	MATLAB/Octave
Search help files			lookfor plot
List available packages		help(); modules [Numeric]	help
Locate functions		help(plot)	which plot

### 1.2 Using interactively

Desc.	IDL	Python	MATLAB/Octave
Start session	idlde	ipython -pylab	Octave: octave -q
Auto completion		TAB	Octave: TAB or M-?
Run code from file	@"foo.idlbatch" or .run 'foo.pro'	execfile('foo.py') or run foo.py	foo(.m)
Command history	help,/rec	hist -n	Octave: history
Save command history	journal, 'IDLhistory'		diary on [...] diary off
End session	exit or CTRL-D	CTRL-D CTRL-Z # windows sys.exit()	exit or quit

## 2 Operators

Desc.	IDL	Python	MATLAB/Octave
Help on operator syntax			help -

<sup>1</sup>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://stdas.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).

## 2.1 Arithmetic operators

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

## 2.2 Relational operators

Desc.	IDL	Python	MATLAB/Octave
Equal	a eq b	a == b <b>or</b> equal(a,b)	a == b
Less than	a lt b	a < b <b>or</b> less(a,b)	a < b
Greater than	a gt b	a > b <b>or</b> greater(a,b)	a > b
Less than or equal	a le b	a <= b <b>or</b> less_equal(a,b)	a <= b
Greater than or equal	a ge b	a >= b <b>or</b> greater_equal(a,b)	a >= b
Not Equal	a ne b	a != b <b>or</b> not_equal(a,b)	a ~= b

## 2.3 Logical operators

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

## 2.4 root and logarithm

Desc.	IDL	Python	MATLAB/Octave	
Square root	sqrt(a)	math.sqrt(a)	sqrt(a)	$\sqrt{a}$
Logarithm, base $e$ (natural)	alog(a)	math.log(a)	log(a)	$\ln a = \log_e a$
Logarithm, base 10	alog10(a)	math.log10(a)	log10(a)	$\log_{10} a$
Logarithm, base 2 (binary)		math.log(a, 2)	log2(a)	$\log_2 a$
Exponential function	exp(a)	math.exp(a)	exp(a)	$e^a$

## 2.5 Round off

Desc.	IDL	Python	MATLAB/Octave
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.	IDL	Python	MATLAB/Octave
$\pi = 3.141592$	<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.	IDL	Python	MATLAB/Octave
Not a Number		<code>nan</code>	<code>NaN</code>
Infinity, $\infty$		<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.	IDL	Python	MATLAB/Octave
Imaginary unit	<code>complex(0,1)</code>	<code>z = 1j</code>	<code>i</code>
A complex number, $3 + 4i$	<code>z = complex(3,4)</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(z)</code>
Real part	<code>real_part(z)</code>	<code>z.real</code>	<code>real(z)</code>
Imaginary part	<code>imaginary(z)</code>	<code>z.imag</code>	<code>imag(z)</code>
Argument			<code>arg(z)</code>
Complex conjugate	<code>conj(z)</code>	<code>z.conj(); z.conjugate()</code>	<code>conj(z)</code>

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

## 2.8 Trigonometry

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

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

## 2.9 Generate random numbers

Desc.	IDL	Python	MATLAB/Octave
Uniform distribution	<code>randomu(seed, 10)</code>	<code>random.random((10,))</code> <code>random.uniform((10,))</code>	<code>rand(1,10)</code>
Uniform: Numbers between 2 and 7	<code>2+5*randomu(seed, 10)</code>	<code>random.uniform(2,7,(10,))</code>	<code>2+5*rand(1,10)</code>
Uniform: 6,6 array	<code>randomu(seed,[6,6])</code>	<code>random.uniform(0,1,(6,6))</code>	<code>rand(6)</code>
Normal distribution	<code>randomn(seed, 10)</code>	<code>random.standard_normal((10,))</code>	<code>randn(1,10)</code>



## 3.6 Vector multiplication

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

## 4 Matrices

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

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

### 4.1 Concatenation (matrices); rbind and cbind

Desc.	IDL	Python	MATLAB/Octave
Bind rows		<code>concatenate((a,b), axis=0)</code>	<code>[a ; b]</code>
		<code>vstack((a,b))</code>	
Bind columns		<code>concatenate((a,b), axis=1)</code>	<code>[a , b]</code>
		<code>hstack((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>concatenate((a,b), axis=None)</code>	<code>[a(:), b(:)]</code>
Bind rows (from vectors)		<code>concatenate((r_[1:5],r_[1:5])).reshape((2,4))</code>	<code>[2;4] 1:4]</code>
Bind columns (from vectors)		<code>vstack((r_[1:5],r_[1:5]))</code>	<code>[1:4 ; 1:4]'</code>

### 4.2 Array creation

Desc.	IDL	Python	MATLAB/Octave
0 filled array	<code>dblarr(3,5)</code>	<code>zeros((3,5),Float)</code>	<code>zeros(3,5)</code>
0 filled array of integers	<code>intarr(3,5)</code>	<code>zeros((3,5))</code>	
1 filled array	<code>dblarr(3,5)+1</code>	<code>ones((3,5),Float)</code>	<code>ones(3,5)</code>
Any number filled array	<code>intarr(3,5)+9</code>		<code>ones(3,5)*9</code>
Identity matrix	<code>identity(3)</code>	<code>identity(3)</code>	<code>eye(3)</code>
Diagonal	<code>diag_matrix([4,5,6])</code>	<code>diag((4,5,6))</code>	<code>diag([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.	IDL	Python	MATLAB/Octave
Reshaping (rows first)	<code>reform(a,2,3)</code>	<code>arange(1,7).reshape(2,-1)</code> <code>a.setshape(2,3)</code>	<code>reshape(1:6,3,2)'</code> ;
Reshaping (columns first)		<code>arange(1,7).reshape(-1,2).transpose()</code>	<code>reshape(1:6,2,3);</code>
Flatten to vector (by rows, like comics)		<code>a.flatten()</code> or	<code>a'(:)</code>
Flatten to vector (by columns)		<code>a.flatten(1)</code>	<code>a(:)</code>
Flatten upper triangle (by columns)			<code>vech(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.	IDL	Python	MATLAB/Octave
Copy of a		<code>b = a.copy()</code>	<code>b = a</code>

## 4.5 Indexing and accessing elements (Python: slicing)

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

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

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

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

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

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

## 4.6 Assignment

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

## 4.7 Transpose and inverse

Desc.	IDL	Python	MATLAB/Octave
Transpose	<code>transpose(a)</code>	<code>a.conj().transpose()</code>	<code>a'</code>
Non-conjugate transpose		<code>a.transpose()</code>	<code>a.' or transpose(a)</code>
Determinant	<code>determ(a)</code>	<code>linalg.det(a) or</code>	<code>det(a)</code>
Inverse	<code>invert(a)</code>	<code>linalg.inv(a) or</code>	<code>inv(a)</code>
Pseudo-inverse		<code>linalg.pinv(a)</code>	<code>pinv(a)</code>
Norms		<code>norm(a)</code>	<code>norm(a)</code>
Eigenvalues	<code>hqr(elmhes(a))</code>	<code>linalg.eig(a)[0]</code>	<code>eig(a)</code>
Singular values	<code>svdc,A,w,U,V</code>	<code>linalg.svd(a)</code>	<code>svd(a)</code>
Cholesky factorization		<code>linalg.cholesky(a)</code>	<code>chol(a)</code>
Eigenvectors		<code>linalg.eig(a)[1]</code>	<code>[v,1] = eig(a)</code>
Rank		<code>rank(a)</code>	<code>rank(a)</code>

## 4.8 Sum

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

## 4.9 Sorting

Desc.	IDL	Python	MATLAB/Octave
Example data		<code>a = array([[4,3,2],[2,8,6],[1,4,7]])</code>	<code>a = [ 4 3 2 ; 2 8 6 ; 1 4 7 ]</code>
Flat and sorted		<code>a.ravel().sort() or</code>	<code>sort(a(:))</code>
Sort each column	<code>sort(a)</code>	<code>a.sort(axis=0) or msort(a)</code>	<code>sort(a)</code>
Sort each row		<code>a.sort(axis=1)</code>	<code>sort(a')'</code>
Sort rows (by first row)		<code>a[a[:,0].argsort(),]</code>	<code>sortrows(a,1)</code>
Sort, return indices		<code>a.ravel().argsort()</code>	
Sort each column, return indices		<code>a.argsort(axis=0)</code>	
Sort each row, return indices		<code>a.argsort(axis=1)</code>	

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

## 4.10 Maximum and minimum

Desc.  
 max in each column  
 max in each row  
 max in array  
 return indices, i  
 pairwise max

max-to-min range

IDL  
`max(a,DIMENSION=2)`  
`max(a,DIMENSION=1)`  
`max(a)`

Python  
`a.max(0)` or `amax(a [,axis=0])`  
`a.max(1)` or `amax(a, axis=1)`  
`a.max()` or

`maximum(b,c)`

`a.ptp(); a.ptp(0)`

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

## 4.11 Matrix manipulation

Desc.  
 Flip left-right  
 Flip up-down  
 Rotate 90 degrees  
 Repeat matrix: [ a a a ; a a a ]

Triangular, upper  
 Triangular, lower

IDL  
`reverse(a)`  
`reverse(a,2)`  
`rotate(a,1)`

Python  
`fliplr(a)` or `a[:,::-1]`  
`flipud(a)` or `a[::-1,:]`  
`rot90(a)`  
`kron(ones((2,3)),a)`

`triu(a)`  
`tril(a)`

MATLAB/Octave  
`fliplr(a)`  
`flipud(a)`  
`rot90(a)`  
`repmat(a,2,3)`  
 Octave: `kron(ones(2,3),a)`  
`triu(a)`  
`tril(a)`

## 4.12 Equivalents to "size"

Desc.  
 Matrix dimensions  
 Number of columns  
 Number of elements  
 Number of dimensions  
 Number of bytes used in memory

IDL  
`size(a)`  
`s=size(a) & s[1]`  
`n_elements(a)`

Python  
`a.shape` or `a.getshape()`  
`a.shape[1]` or `size(a, axis=1)`  
`a.size` or `size(a[, axis=None])`  
`a.ndim`  
`a.nbytes`

MATLAB/Octave  
`size(a)`  
`size(a,2)` or `length(a)`  
`length(a(:))`  
`ndims(a)`

## 4.13 Matrix- and elementwise- multiplication

Desc.  
 Elementwise operations

Matrix product (dot product)

Inner matrix vector multiplication  $a \cdot b'$

Outer product

Kronecker product

Matrix division,  $b \cdot a^{-1}$   
 Left matrix division,  $b^{-1} \cdot a$   
 (solve linear equations)  
 Vector dot product  
 Cross product

IDL

`a # b` or `b ## a`

`transpose(a) # b`

`a # b`

`cramer(a,b)`

Python

`a * b` or `multiply(a,b)`

`matrixmultiply(a,b)`

`inner(a,b)` or

`outer(a,b)` or

`kron(a,b)`

`linalg.solve(a,b)`

`vdot(a,b)`  
`cross(a,b)`

MATLAB/Octave

`a .* b`

`a * b`

`kron(a,b)`

`a / b`  
`a \ b`

$$\begin{bmatrix} 1 & 5 \\ 9 & 16 \end{bmatrix} \begin{bmatrix} 7 & 10 \\ 15 & 22 \\ 5 & 11 \\ 11 & 25 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 4 & 8 & 12 & 16 \end{bmatrix}$$

$$Ax = b$$



## 4.14 Find; conditional indexing

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

## 5 Multi-way arrays

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

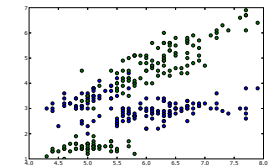
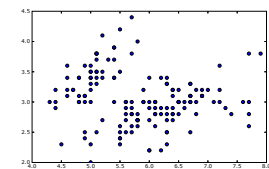
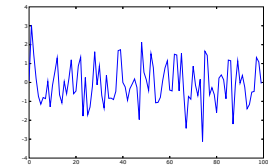
## 6 File input and output

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

## 7 Plotting

### 7.1 Basic x-y plots

Desc.	IDL	Python	MATLAB/Octave
1d line plot	<code>plot, a</code>	<code>plot(a)</code>	<code>plot(a)</code>
2d scatter plot	<code>plot, x(1,*), x(2,*)</code>	<code>plot(x[:,0],x[:,1], 'o')</code>	<code>plot(x(:,1),x(:,2), 'o')</code>
Two graphs in one plot Overplotting: Add new plots to current	<code>plot, x1, y1</code> <code>oplot, x2, y2</code>	<code>plot(x1,y1,'bo', x2,y2,'go')</code> <code>plot(x1,y1,'o')</code> <code>plot(x2,y2,'o')</code> <code>show()</code> # as normal	<code>plot(x1,y1, x2,y2)</code> <code>plot(x1,y1)</code> <code>hold on</code> <code>plot(x2,y2)</code>
subplots Plotting symbols and color	<code>!p.multi(0,2,1)</code> <code>plot, x,y, line=1, psym=-1</code>	<code>subplot(211)</code> <code>plot(x,y,'ro-')</code>	<code>subplot(211)</code> <code>plot(x,y,'ro-')</code>



#### 7.1.1 Axes and titles

Desc.	IDL	Python	MATLAB/Octave
Turn on grid lines 1:1 aspect ratio		<code>grid()</code> <code>figure(figsize=(6,6))</code>	<code>grid on</code> <code>axis equal</code> Octave: <code>axis('equal')</code>
Set axes manually	<code>plot, x(1,*), x(2,*),</code> <code>xran=[0,10], yran=[0,5]</code>	<code>axis([ 0, 10, 0, 5 ])</code>	<code>replot</code> <code>axis([ 0 10 0 5 ])</code>
Axis labels and titles	<code>plot, x,y, title='title',</code> <code>xtitle='x-axis', ytitle='y-axis'</code>		<code>title('title')</code> <code>xlabel('x-axis')</code> <code>ylabel('y-axis')</code>
Insert text	<code>xyouts, 2,25, 'hello'</code>	<code>text(2,25,'hello')</code>	

## 7.1.2 Log plots

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

IDL  
`plot, x,y, /YLOG` or `plot_io, x,y`  
`plot, x,y, /XLOG` or `plot_oi, x,y`  
`plot_oo, x,y`

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

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

## 7.1.3 Filled plots and bar plots

Desc.

IDL

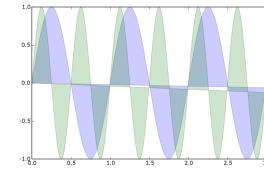
Python

MATLAB/Octave

Filled plot

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

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



## 7.1.4 Functions

Desc.  
Defining functions

IDL

Python

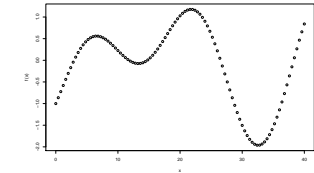
MATLAB/Octave  
`f = inline('sin(x/3) - cos(x/5)')`

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

Plot a function for given range

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

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



## 7.2 Polar plots

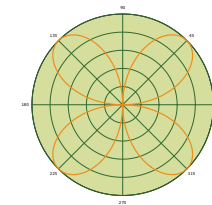
Desc.

IDL

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

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

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



`polar(theta, rho)`

`polar(theta, rho)`

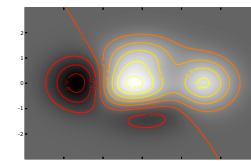
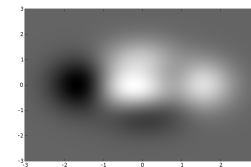
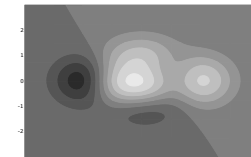
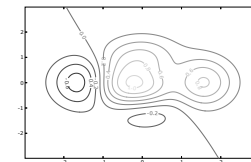
## 7.3 Histogram plots

Desc.	IDL	Python	MATLAB/Octave
	<code>plot, histogram(randomn(5,1000))</code>		<code>hist(randn(1000,1))</code> <code>hist(randn(1000,1), -4:4)</code> <code>plot(sort(a))</code>

## 7.4 3d data

### 7.4.1 Contour and image plots

Desc.	IDL	Python	MATLAB/Octave
Contour plot	<code>contour, z</code>	<code>levels, colls = contour(Z, V,</code> <code>    origin='lower', extent=(-3,3,-3,3))</code> <code>clabel(colls, levels, inline=1,</code> <code>    fmt='%1.1f', fontsize=10)</code>	<code>contour(z)</code>
Filled contour plot	<code>contour, z, nlevels=7, /fill</code> <code>contour, z, nlevels=7, /overplot, /downhildmap=cm.gray,</code>	<code>contourf(Z, V,</code> <code>    origin='lower',</code> <code>    extent=(-3,3,-3,3))</code>	<code>contourf(z); colormap(gray)</code>
Plot image data	<code>tv, z</code> <code>loadct,0</code>	<code>im = imshow(Z,</code> <code>    interpolation='bilinear',</code> <code>    origin='lower',</code> <code>    extent=(-3,3,-3,3))</code>	<code>image(z)</code> <code>colormap(gray)</code>
Image with contours Direction field vectors		<code># imshow() and contour() as above</code> <code>quiver()</code>	<code>quiver()</code>



## 7.4.2 Perspective plots of surfaces over the x-y plane

Desc.

IDL

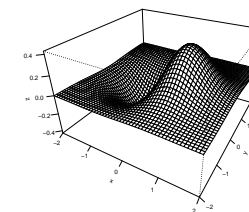
Python

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

MATLAB/Octave

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

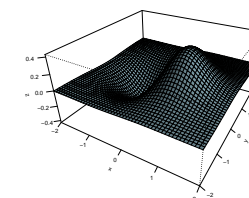
$$f(x,y) = xe^{-x^2-y^2}$$



Mesh plot

surface, z

mesh(z)



Surface plot

shade\_surf, z  
loadct,3

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

## 7.4.3 Scatter (cloud) plots

Desc.

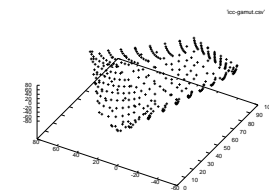
IDL

Python

MATLAB/Octave

3d scatter plot

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



## 7.5 Save plot to a graphics file

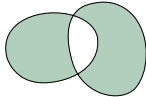
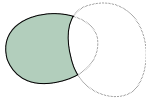
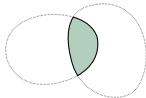
Desc. PostScript	IDL set_plot,'PS' device, file='foo.eps', /land plot x,y device,/close & set_plot,'win'	Python savefig('foo.eps')	MATLAB/Octave plot(1:10) print -depsc2 foo.eps Octave: gset output "foo.eps" gset terminal postscript eps plot(1:10)
PDF SVG (vector graphics for www) PNG (raster graphics)		savefig('foo.pdf') savefig('foo.svg') savefig('foo.png')	print -dpng foo.png

# 8 Data analysis

## 8.1 Set membership operators

Desc. Create sets	IDL	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])	MATLAB/Octave a = [ 1 2 2 5 2 ]; b = [ 2 3 4 ];
Set unique		uniqueid(a) unique(a) set(a)	unique(a)
Set union		union1d(a,b) a.union(b)	union(a,b)
Set intersection		intersect1d(a) a.intersection(b)	intersect(a,b)
Set difference		setdiff1d(a,b) a.difference(b)	setdiff(a,b)
Set exclusion		setxor1d(a,b) a.symmetric_difference(b)	setxor(a,b)
True for set member		2 in a setmember1d(2,a) contains(a,2)	ismember(2,a)

[ 1 2 5 ]



## 8.2 Statistics

Desc.	IDL	Python	MATLAB/Octave
Average	<code>mean(a)</code>	<code>a.mean(axis=0)</code> <code>mean(a [,axis=0])</code>	<code>mean(a)</code>
Median	<code>median(a)</code>	<code>median(a)</code> <b>or</b> <code>median(a [,axis=0])</code>	<code>median(a)</code>
Standard deviation	<code>stddev(a)</code>	<code>a.std(axis=0)</code> <b>or</b> <code>std(a [,axis=0])</code>	<code>std(a)</code>
Variance	<code>variance(a)</code>	<code>a.var(axis=0)</code> <b>or</b> <code>var(a)</code>	<code>var(a)</code>
Correlation coefficient	<code>correlate(x,y)</code>	<code>correlate(x,y)</code> <b>or</b> <code>corrcoef(x,y)</code>	<code>corr(x,y)</code>
Covariance		<code>cov(x,y)</code>	<code>cov(x,y)</code>

### 8.3 Interpolation and regression

Desc.	IDL	Python	MATLAB/Octave
Straight line fit	<code>poly_fit(x,y,1)</code>	<code>(a,b) = polyfit(x,y,1)</code> <code>plot(x,y,'o', x,a*x+b,'-')</code>	<code>z = polyval(polyfit(x,y,1),x)</code> <code>plot(x,y,'o', x,z ,'-')</code>
Linear least squares $y = ax + b$		<code>linalg.lstsq(x,y)</code>	<code>a = x\y</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.	IDL	Python	MATLAB/Octave
Polynomial		<code>poly()</code>	
Find zeros of polynomial		<code>roots()</code>	<code>roots([1 -1])</code>
Find a zero near $x = 1$			<code>f = inline('1/x - (x-1)')</code> <code>fzero(f,1)</code>
Solve symbolic equations			<code>solve('1/x = x-1')</code>
Evaluate polynomial		<code>polyval(array([1,2,1,2]),arange(1,11))</code>	<code>polyval([1 2 1 2],1:10)</code>

### 8.4.2 Differential equations

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

## 8.5 Fourier analysis

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

## 9 Symbolic algebra; calculus

Desc.	IDL	Python	MATLAB/Octave
Factorization			<b>factor()</b>

## 10 Programming

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

### 10.1 Loops

Desc. for-statement Multiline for statements	IDL for k=1,5 do print,k for k=1,5 do begin \$ print, i &\$ print, i*2 &\$ end	Python for i in range(1,6): print(i) for i in range(1,6): print(i) print(i*2)	MATLAB/Octave for i=1:5; disp(i); end for i=1:5 disp(i) disp(i*2) end
--	---	---	--

### 10.2 Conditionals

Desc. if-statement if-else-statement Ternary operator (if?true:false)	IDL if 1 gt 0 then a=100 if 1 gt 0 then a=100 else a=0 a>0?a:0	Python if 1>0: a=100	MATLAB/Octave if 1>0 a=100; end if 1>0 a=100; else a=0; end  $a > 0 ? a : 0$
--	---	-------------------------	--

### 10.3 Debugging

Desc. Most recent evaluated expression List variables loaded into memory Clear variable $x$ from memory Print	IDL  help  print, a	Python   print a	MATLAB/Octave ans whos or who clear x or clear [all] disp(a)
---	---------------------------------	---------------------------	--

### 10.4 Working directory and OS

Desc. List files in directory List script files in directory Displays the current working directory Change working directory Invoke a System Command	IDL dir  sd cd, 'foo or sd, 'foo spawn, 'notepad'	Python os.listdir(".") grep.grep("*.py") os.getcwd() os.chdir('foo') os.system('notepad') os.popen('notepad')	MATLAB/Octave dir or ls what pwd cd foo !notepad Octave: system("notepad")
---	--	---	--

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

<sup>3</sup>Version 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/>; Octave 2.1.50, <http://www.octave.org/>; Gnuplot 4.0, <http://www.gnuplot.info/>.

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

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