

Matlab vs. IDL

Michael Liebling
Biomedical Imaging Group
Institut d'Imagerie et Optique appliquée
Swiss Federal Institute of Technology, Lausanne
CH - 1015 Lausanne
Switzerland
michael.liebling@epfl.ch

February 28, 2002

1 Basic commands and synthax

MATLAB	IDL	Purpose
<pre>x'; x(3:5); A*B; A.*B; A.^2; A^2; % ... A=[1,2,3;4,5,6]; B=A(:,2:3); x=0:9; x=0.0:1.0:9.0; x=byte(0:255); x=0.0:1.0:9.0; sum(x); sum(sum(A)); x z=x;y=2*x; size(mat) length(vec) linspace(0,pi,100) fliplr(A) flipud(A) rot90(A) repmat 10*ones(N)</pre>	<pre>TRANPOSE(x) x[3:5] A##B A*B A^2 A##A ; \$ A=[[1,2,3],[4,5,6]] B=A[1:2,*] x=INDGEN(10) x=FINDGEN(10) x=BINDGEN(256) x=CINDGEN(10) TOTAL(x) TOTAL(A) PRINT, x z=x & y=2*x SIZE(mat,/DIMENSIONS) N_ELEMENTS(vec) REVERSE(A,1) REVERSE(A,2) ROTATE(A,1) ???? res=REPLICATE(10,N,N)</pre>	<pre>Transpose Vector portion Matrix multiplication Multiplication Exponentiation Exponentiation Comment Line continuation Matrix formation Submatrix extraction Integer vector Float vector Byte vector Complex vector Sum of vector Sum over all elements in matrix Print vector x Line continuation Size of a matrix Length of a vector Linearly spaced vector Flip columns Flip lines Linearly spaced vector Replicate a matrix Replicate a scalar</pre>
<pre>for k=1:10, disp(k), end for k=1:N x(k) = k; end</pre>	<pre>FOR k=1,10 DO PRINT, k FOR k=1,10 DO BEGIN x[k]=k ENDFOR</pre>	<pre>For Loop</pre>
<pre>if I == J A(I,J) = 2; elseif abs(I-J) == 1 A(I,J) = -1; else A(I,J) = 0; end</pre>	<pre>IF (I EQ J) THEN BEGIN A[I,J] = 2 ENDIF ELSE BEGIN A[I,J] = -1 ENDELSE</pre>	<pre>If Statements</pre>
<pre>switch lower(METHOD) case {'linear','bilinear'},... disp('linear') case 'cubic', disp('cubic') case 'nearest', disp('nearest') otherwise, disp('Unknown') end</pre>	<pre>CASE name OF 'Linda': PRINT, 'sister' 'John': PRINT, 'brother' 'Harry': PRINT, 'step-brother' ELSE: PRINT, 'Not a sibling.' ENDCASE</pre>	<pre>Cases</pre>
<pre>save 'image.mat' R G B I load 'image.mat' load 'file.txt' -ASCII save result 'result.txt' -ASCII whos whos x who help repmat lookfor spline type repmat why bad = find(data<0) txt=num2str(1.222) eval('sin(x)')</pre>	<pre>SAVE, FILENAME = 'image.dat', R, G, B,I RESTORE, 'image.dat' OPENW,1,'result.txt' &\$ PRINTF, 1, result, FORMAT='(13F10.3)'+&\$ CLOSE, 1 HELP HELP, x ? indgen bad = WHERE(data LT 0) txt=STRING(1.222) EXECUTE('SIN(x)') PRINT, 20.*!dpi/4., FORMAT='(F30.18)' t=STRING(\$ 1,2, FORMAT='(" ,I1," ,I2,")')</pre>	<pre>Save a color image in a file Restore the saved image Restore a matrix stored in ASCII Write a matrix as ASCII Get infos on variables Get infos on variable x Get help Lookfor word <i>spline</i> in help Type the source code If you complain ask for a response Get indices where data meets cond. Convert to text Execute (evaluate) the command Formatted printing print (1,2)</pre>

2 File manipulation

MATLAB	IDL	Purpose
	info=ROUTINE.INFO(sin) & PRINT, info.path	Get info on a function or procedure
pause(3)	WAIT(3)	Perform a 3 second pause
cd	CD	Change directory
cd ..	CD, '::'	Change to upper directory (MacOS)
ls	PRINT, FINDFILE('*')	List files
pwd	CD, C=c & PRINT, c	Path of current directory

3 Help generation

MATLAB	IDL	Purpose
	MK_HTML_HELP	
	DOC_LIBRARY	

4 Plotting nice graphs

MATLAB	IDL	Purpose
h=figure(1)	WINDOW, 0, TITLE=string, XPOS=value, \$ YPOS=value, XSIZE=pixels, YSIZE=pixels	open window
text(1,1,'1,1')	XYOUTS, 1, 1, "1,1" PSYM	put text at position x, y plot symbol

5 Constants

MATLAB	IDL	Purpose
pi	!PI, !DPI	π
	!RADEG	$180/\pi \approx 57.2958$
	!VALUES.F.INFINITY	∞
	!VALUES.F.NAN	Not a number
inf	!VALUES.D.INFINITY	
NaN	!VALUES.D.NAN	
A = [1,2,3]	A = [1,2,3]	Forming complex arrays
B = [4,5,6]	B = [4,5,6]	
C=A+i*B	C = COMPLEX(A, B)	
i or j	COMPLEX(0, 1.)	$\sqrt{-1}$
clear all	.RESET_SESSION	Clear all variables
clear A	DELVAR, A	Clear variable

6 Logics

MATLAB	IDL	Purpose
>	GT	
<	LT	
==	EQ	
>=	GE	
<=	LE	
~=	NE	

7 Procedures, functions, batchfiles

IDL code can take several forms:

1. a batch file
2. a *main* program
3. a procedure
4. a function

whereas in MATLAB, it can only take 2 forms:

1. script m-files
2. function

7.1 M-Script files, batch files, main files

MATLAB

A typical M-script file in MATLAB looks like this:

myplot.m

```
color='rgb'  
x=0:9;  
y=(x/10).^2;  
for k=1:length(color)  
    plot(x,y+k,color(k))  
    hold on  
end  
hold off
```

it is simply launched with the command line:

```
> myplot
```

IDL

In IDL this could be accomplished using a batch file like the one below.

mybatch.pro

```
linecolor=[255L,256L*(255L),256L*(256L*255L)]  
x=indgen(9)  
y=x^2  
plot, x,y, /nodata  
FOR k=0,2 DO oplot, x, y/10.+k,$  
color=linecolor[k],$  
background=255L+256L*(256L*255L)
```

Note that there is only a single command per line (in particular, the `for` construct has no `BEGIN` and `END` which is forbidden) and that the `$` character has been used to break the `oplot` command over several lines. A batch file is run with the command:

```
IDL>> @mybatch
```

Another possibility is to use a *main* .pro file like the one below:

mymain.pro

```
linecolor=[255L,256L*(255L),256L*(256L*255L)]  
x=indgen(9)  
y=x^2  
plot, x,y, /nodata  
FOR k=1,3 DO BEGIN  
    oplot, x,y, color=linecolor[k]  
END  
END
```

(notice the `END` at the last line) and is run with the command:

```
IDL>> .run mymain
```

or with the equivalent successive commands:

```
IDL>> .compile mymain
```

```
IDL>> .go mymain
```

7.2 Functions and procedures

MATLAB

MATLAB functions can take several arguments (their number may be variable) and can output several variables.

stat.m

```
function [mean,stdev] = stat(x)
n = length(x);
mean = avg(x,n);
stdev = sqrt(sum((x-avg(x,n)).^2)/n);

function mean = avg(x,n)
mean = sum(x)/n;
```

IDL

IDL functions can take several arguments (their number may be variable if they are specified as keywords) but have only one output. Arguments that are passed as keywords are modified inside the function.

average.pro

```
FUNCTION AVERAGE, arr
  RETURN, TOTAL(arr)/N_ELEMENTS(arr)
END
```

Next function returns the average and puts the standard deviation into variable `std` if and only if it is specified.

average2.pro

```
FUNCTION AVERAGE, arr, STDDEV=std
  aver=TOTAL(arr)/N_ELEMENTS(arr)
  IF KEYWORD_SET(std) THEN
    std=SQRT(TOTAL((arr-aver)^2)/N_ELEMENTS(arr))
  RETURN, aver
END
```

Still another possibility is to have a procedure. Procedure don't return a result.

hello.pro

```
PRO HELLO, name, UP=upcase
  IF KEYWORD_SET(upcase) THEN PRINT,
  STRUPCASE(name) $
  ELSE PRINT, name
END
```