Matlab cheat sheet - Brian McGill Feb 2000

Basics

`expression<enter>` and `expression, expression <enter>` prints result while `expression;<enter>` does not
`var=expression` stores it (variable names are case sensitive)
... continues a line
`%` comment continues to end of line
expressions use: `+ - * / ^ () .* ./ ^ mod log log10 log2 exp sqrt pi eps realmin realmax flops`
logical expressions: `< <= > >= == ~= & | ~ xor(x,y) any(x) all(x)`
rounding: `fix` (towards 0) `floor ceil round` (nearest int)
imaginary numbers: `1+2i abs() angle() real() imag()`
`whos [-file file]` shows existing variables while `clear varname` removes
and `save filebase [var1, …] [-append]` saves to filebase.mat and `load filebase` [var1 …] brings it back
`clc` clears the screen and `format short/long/short e/long e/short g/long g/hex/+/bank` formats output
`dir` `which`('func') `exists`('name'[,'type'])

Arrays

`[a b c]` gives a row vector `[a; b; c]` gives a column vector `[a b c; d e f]` gives a 2X3 array
indexing: `a[idx1, idx2, …]` subscripts where idxn can be `n` or `[n1 n2 n3]` or `:` or `end` or `n1:n2` or `n1:step:n2`
(note step can be negative) also `a[:]` column vectorizes and `a[i]` gets ith entry
creating: `(start:end)` or `(start:end:step)` or `linspace(start,end,#)` or `logspace(start,end,#)` or
`zeros(n1,…)` or `ones(n1, …)` or `eye(n)` (identity) or `rand(n)` or `rand(n1,n2,…)` or `randn(n)` (normal) or
`repmat(val,n1,…)`
searching: `find(A rel val)` returns i indices, `[r,c]=find()` returns r,c `A(find(A rel))` returns entries meeting criteria just `find(A)` returns non-zero elements
functions: `‘` (transpose) `.’` (complex conj) `diag(vec) diag(ary) size(A) size(A,n) dot(v1,v2)`
updating: a subscripted array may be on LHS of =; a scalar is expanded; setting to [ ] deletes a row or column
`i=find(isnnan(A)); a(i)=zeros(size(i));`
set type functions: `unique(A)ismember(a,b) union(a,b) intersect(a,b) setxor(a,b) setdiff(a,b)`
`check: isempty isnumeric islogical isnan isinf isfinite isequal(x,round(x)) isreal`

Functions

funcname.m

```
function outvar=funcname(arg1,arg2,arg3)
%help text
% and more
if nargin<3, arg3=default3; end
if nargin<2, arg2=default2; end

statement;
outvar=2*3;
```

outvar can be `[outvar1, outvar2]`
`nargout` also useable
no outvars behave syntactically like commands
`error(string)` fails out
`fprintf(fmt,….)` or command w/ no ';' for output
`[x,y]=feval(‘funcname’,arg1,arg2,…)`
`GLOBAL var`
`edit func` (no .m needed)

Control structures

```
if expr,
  statement;
  statement;
elseif
     statements
else
     statements
end
```

```
while expression
   statement;
end
```

```
try
     statements
catch
     statements
end
```

```
for  var=rowvec
   statement;
end
```

e.g. `for n=1:10` picks up columns

`break` is usable in all

```
switch expression
  case testexpr
    statement;
  case {ex1, ex2}
    statement;
  otherwise
    statement;
end
```

Cells & Structures

Cell=array of subarrays	to access subarray use arry{i,j} [=arry2]	creation: { }	
conversion:	string/cell: `char/cellstr`	number/cell: {} or or [ary{:}] or `cat(1,ary{:})/num2cell`	
Struct=fields X records as cells	struct(i).field	fieldnames(struct)	struct('f1',val1,'f2',val2,…)
conversion:	struct2cell/cell2struct		

Strings

`‘this is a string’`
`help strfun`
strings are numerical arrays of ASCII values: e.g. `size(‘how long’)` gives 1 8 and subscripting works
`char` or `str2mat` gives a 2-D array w/ variable # of columns - `char(‘str1’, ‘str2 is longer’)`
access these by `mystrlist(n,:)`
concatenation: ['str1' 'str2' 'str3'] or for multirow strings: `strcat(a,b)` concatenates strings as long as they have the same # of rows
numeric conversion: `int2str(n) num2str(f)` and `sprintf(fmt,num)`

```
ischar(S) isletter(S) isspace(S) lower(S) upper(S) sttrep(s1,s2,s3)
findstr(S1,S2) strcmp(S1,S2)  strncmp(S1,S2,n) strtok(S1,D)    strmatch(s,sary)
eval(str)   str2num()    num2str()
startidx[,finish,tokens]=regexp[i](str,expr)   str=regexprep(str,expr,rep)
```

Basic analysis

`func(array)all array` or `func(array,1)` works across rows while `mean(array,2)` works across columns
`mean, max, min, cov, diff, std(a,[0/1[,dir]]), sum, prod, sort, rank, cumsum, cumprod`
`[val, index]=max()/min()` in std, 0 means n-1, 1 means n
`polyfit(x,y,n)   interpl(x,y,val[,’cubic’,’spline’,’nearest’])`
function functions: func can be m-file, string (w/ ‘x’), inline func (`var=inline(‘str’)`)
optimization: `fmin(func,min,max) fmin(ndfunc,initguessvec)`
zeros: `fzero(func[,start=0])` integration: quad(f,a,b[,tol])
difeq: `ode45(func,[beg end],init)` where func is a .m file taking `yprime=func(t,y)`
integration: `trapz(x,y) quad(inlfunc,lo,hi) quad8 dblquad` differentiation: `diff(y)./diff(x)`
`det(A) eig(A)[V,D]=eig(A)expm(A)inv(A)norm(A) norm(A,p)poly(A)rank(A)svd(A)trace(A)`
`jordan(A), colspace(A)`

Symbolic

`1) ‘expr’ or 2) syms a b` then `expr` w/ a,b or 3) sym(num) or 4) `symop(‘2,’+’,’x’)`

```
[n, d]=numden(se) compose(f1,f2) finverse(f) double(se) sym2poly(f) poly2sym(v)
subs(se,var,var/num) symsum(f) symsum(f,a,b) diff(f) diff(f,var) diff(f,n) int(f,a,b)
solve(se) solve(se,var) solve(se1,se2,…,var1,var2, …)
```

`dsolve(difeq[,var])` where difeq uses Dy,y, Dny w/ (0) for init and has multiple expr sep by ‘,’

```
det(matrixse) inv(matrixse) eig(matrixse) [V,D]=eig(matrixse) jordan, svd, colspace, null
taylor(se) jacobian(se,var1,var2) laplace(func,var,newvar) fourier(f,fv,nv) ztrans(f,v,n)
simplify(f) factor(f) expand(f) simple(f) vpa(se,digits) pretty(se) latex(se) ccode(se)
```

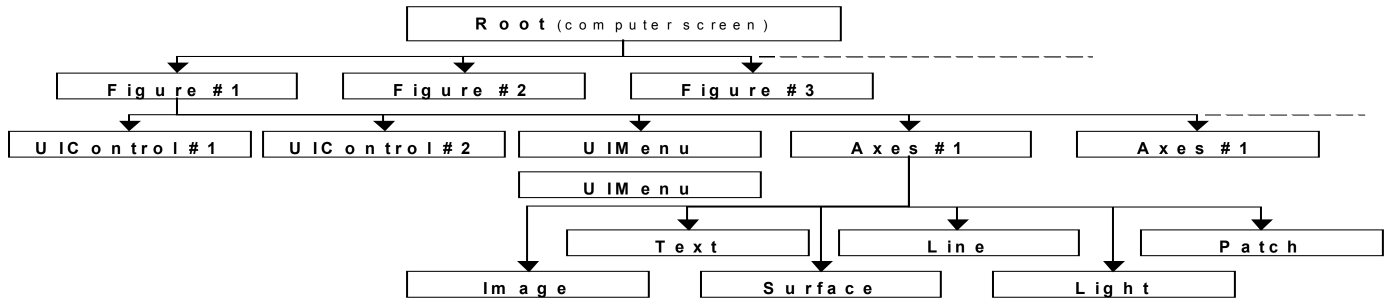
Graphing

`plot([x,]y[,style],…)` where style is string of color/symbol/line `bgrcmykw .ox+*sd^<>vph - : -. --`
`grid on/off;    xlabel(‘str’) ylabel(‘str’) title(‘str’) legend(‘str1’,’str2’,…)`
`glabel(X/Y/T,size[,color])` **color:** `ymcrgbwk` **symbols:** `.ox+*sdv^<>ph` **line:** `-, :, -., --`
`gbanner(haxis,string[,ptsz]);` **`Tex labels:`** `superscript: ^{}  subscript: _{}  \alpha …`
`axis([xmin xmax ymin ymax])   axis auto; axis manual` (freezes); `axis square; axis equal;`
`hold on; hold off; subplot(rows,cols,graph); figure;`
`loglog simlogx semilogy` all as plot `bar([x,]y[,width][,’stacked’])`
`pie(vec[,ispulledvec]) hist(vec[,binvec]) errorbar(x,y,e) plotmatrix(x,y)` as scatter
`fplot(f,[xmin xmax]) fplot(f,[xmin xmax ymin max]) ezplot(f)`
`3D:`
`xr=0:0.1:1; yr=0:0.1:1; [x y]=meshgrid(xr,yr); z=x+y.*z;`
then
`mesh(x,y,z) surf(x,y,z) countourf(x,y,z,n)=countour(x,y,z,n,’style’)+pcolor(x,y,z)`

Help topics

`help` topic	`lookfor` verbage	`more on`		
general	specfun	sparfun (sparse	graphics (handle)	datatypes
ops	matfun	mat)	uitools	dde
lang	datafun	graph2d	strfun	demos
elmat	polyfun	graph3d	iofun	symbolic
elfun	funfun (&ODE)	specgraph	timefun	signal
control	glmlab			
local				

Matlab Handle Graphics cheat sheet - Brian McGill Feb 2000



Object Type	Handles	Creators	Other utils	Comment
Root	0	N/A		Parent of all, holds defaults
Figure	gcf	figure	clf, close, refresh	window (contains axes, controls, menus)
UIControl		uicontrol, btngroup	btnstate, btnpress, btndown, btnup	Style
UIMenu		uimenu, makemenu, winmenu, uicontextmenu		
Axes	gca, get(hFig, 'CurrentAxes")	axes, sbuplot(rows,cols,#)	cla, box on/off, caxis, axis on/off, [xmin xmax ymin max] auto/manual/ij/xy/equal/square	main drawing areas xy places 0,0 in lower left ij places in upper left, y first
Text		text(x,y[,z],string)	xlabel, ylabel, zlabel, title, gtext	text on an axis
Line		line(vx,vy[,vz])		a set of lines
Patch		patch(vx,vy[,vz],color)	fill, fill3, rectangle	a colored polygon
Image		image(bitmapmatrix)	imagesc	a bitmap given by matrix
Surface		surface(x,y,z[,c])		adjacent polygons
Light		light		lighting of surfaces

Property manipulation & inheritance

At any level there is a "shadow" property that gives defaults for all children given by prefixing 'Default' – e.g. 'DefaultAxesFontSize'
Hierarchy: inbuilt, root, figure, axes - at creation time all non-specified propeties taken from first Default found up hierarchy
All the basic creator routines (axes, line, etc) all allow propname, propval, … at tend of function calls

set(h,propname,prop)	get(h,propname)	findobj([hvec,]propname1,propval1, …)
clruprop/setuprop/getuprop	allchild(h)	reset(h) delete(h) copyobj(h)

Common properties objects

BusyAction, ButtonDownFcn,CreateFcn, ChangeFcn, DeleteFcn,Interruptible

Children	Selected	UserData (any Matlab array)
Clipping	SelectionHighlight	Visible ('on'/'off')
HandleVisibility	Tag (any string)	
Parent	Type ('figure','axis','line', etc)	

Text properties: FontAngle (normal/italic/oblique); FontName; FontSize; FontUnits; FontWeight (light/normal/demi/bold); Extent*
Units: (pixels/normalized (0-1)/inches/centimeters/points)
Background Color: Color none/[r g b]/name except UIControl has BackgroundColor
Drawing objects: EraseMode (normal/none/xor/background) Marker (.0+etc) MarkerEdgeColor MarkeFaceColor MarkerSize
LineStyle LineWidth
Location: XData/YData/ZData (line, patch, surface) **Position** (text, uicontrol) [L B W H] - **Extent** is readonly & gives minimum
Axes (?=X/Y/Z): Title/Xlabel/Ylabel; ?Lim [min, max]; ?Dir normal/reverse; ?Grid on/off; ?LimMode auto/manual;
?Scale=linear/log; ?Tick vec; ?TickMode=auto/manual; ?TickLabel strings; ?TickLabelMode auto/manual;
?AxisLocation top/bottom/left/right; DataAspectRatio [dx dy dz]; DataAspectRatio Mode auto/manual;
GridLineStyle -/--/:/-./none; LineWidth n; Clim [cmin, cmax]; CLimMode auto/manual; TickDir in/out;
TickDirMode auto/manual; TickLength auto/manual; View [az, elev]
Figure: CurrentAxes, CurrentCharacter*, CurrentObject, CurrentPoint, Name string, NumberTitle on/off;
Pointer crosshair/arrow/watch etc; Resize off/on; WindowStyle normal/modal
UIControl: style (pushbutton/radiobutton/checkbox/edit/text/slider/frame/listbox/popupmenu); Min n; Max n; Value n; String str;
SliderStep [one page]

Array Indexing

3 modes:

1. a(commalist) where each element in comma separated list matches 1 dimension in array
2. a(idxlist) where the array is treated as a column vector regardless of shape (dim1 varies fastest)
3. a(boollist) where the array is treated as a column vector regardless of shape

commalist

idxlist1,idxlist2,idxlist3

s.field where s is an array of structures or s(idxlist).field

cell{:}

deal(ary**)**

reverse (commalist-> array) [commalist] or cat(dim,commalist) (array-> commalist) comamlist=deal(ary)

assign [commalist]=deal(commalist) including [var1,var2,…]=deal(commalist)

idxlist

: type

n:m represents elements n thru m
n:step:m represents incremental list
end may be used for n or m
: represents all elements

[ ] type

[n1 n2 n3 …]

n

a single #

boollist

vector of 0/1's nb: floats, x, converted to logicals by logical(x) while x>1 → logicals

Tricks

idxlist=find(boolarray)

find converts a boolean list into a idxlist

idxlist ommissions, repetitions and reordering is respected

add a row: `a(newrow,: )=0` ***remove a row*** `a(delrow,: )=[]`

Lookup

[b,idx]=ismember(vals,lookup) res=ceil(interp1(lookup,1:lookup,vals))

add a dimension

any dimension that is non-existent (or 1 position) can be subscripted with multiple 1's (e.g. [1 1 1])
e.g.: a=[1 2 3], a([1 1], **:**) → [1 2 3; 1 2 3]

concatenate arrays

[a1 a2] - horizontal (dim2) concatenation
[a1; a2] – vertical (dim1) concatenation

Stats

Distributions	ncx2	**Distr functions**	grpstats, crosstab,	ranksum, signrank, signtest,
:	norm*	dist*fit	tabulate, bootstrp	ztest, ttest, ttest2, *ks*
beta*	poiss*	distcdf	corrcoef, cov, *simprank,*	boxplot, gline, gname,
bino*	rayl	distpdf	*simpcorr(x,y,printp)*	normplot, refline,
chi2	t	distinv (of cdf)	anova1, anova2, leverage,	weibplot
exp*	unid	distrnd	polyfit(x,y,n),	refcurve, refline([slp,[int]]),
f	unif*	diststat (mean, var)	polyval, , stepwise,	*lllsline, llrefline*
gam*	weib*		regress(y,[1 x]	disttool, polytool, regst
geo		**Tools**	nlinfit(x,y,inline('a(1)*x.^	
hyge		(nan)mean, median,	a(2)','a','x',ainit)	
logn		min, max, std,	cluster, pdist, linkage, …	
nbin		sum	princomp, barttest, …	
ncf		prctile, range,	classify (LDA)	
nct		skewness		

Plot Properties

Set via the set(gca,'Propname',propval,'Propname',propval….)

Property	Values	Comments
Color	'none' | 'g' | 'green' | [0 1 0]	Color of background
Linewidth	N	Width in points of axis lines
TickDir	'in' | 'out'	Direction of ticklines
TickDirMode	'auto' | 'manual'	auto = in for 2-D, out for 3-D
TickLength	[2Dlength 3Dlength]	
Visible	'on' | 'off'	Hide axis
Tag	'string'	Tag usable in findobj
UserData	Matrix	Store data in graph
XAxisLocation	'top' | 'bottom'	
YaxisLocation	'left'|'right'	
?Color	'none' | 'g' | 'green' | [0 1 0]	Line & tick color
?Dir	'normal' | 'reverse'	Order of values (e.g. bottom to top)
?Grid	'on' | 'off'	Lines across graph
?Label	Textobject (text('Str','prop','val'…)	Axes labels
Title	"	Graph title
?LimMode	'auto' | 'manual'	Set to manual if set ?Lim
?Lim	[minimum maximum]	Set Axis scale
?TickLabelMode	'auto' | 'manual'	Set to manual if set ?TickLabel
?TickLabel	{'la1','la2'} | 'la1|la2' | [1 2]	Values displayed at ticks
?TickMode	'auto' | 'manual'	Set to 'manual' if set ?Tick
?Tick	Vector matrix – e.g. [1 3 5]	If [ ] then no ticks
?Scale	'log' | 'linear'	
Font*	As per text labels	no effect until ?Label set

LineStyles

Colors: ymcrgbwk
Markers: o . x+*sdv^<>ph
Styles: - : -. –
Line Properties: Marker, MarkerEdgeColor,MarkerFaceColor,MarkerSize, Color,LineStyle,LineWidth

Commands

2D:
```
hLines=plot([x,]y[,style],…)
gbanner(haxis,string[,ptsz]); manyplot(x,y)
hold on; hold off; subplot(rows,cols,graph); figure;
hBars=bar([x,]y[,width][,'stacked']) pie(vec[,ispulledvec]) hist(vec[,binvec]) myhistc()
errorbar(x,y,e) plotmatrix(x,y)
```
as scatter

Function:
```
fplot(f,[xmin xmax]) fplot(f,[xmin xmax ymin max]) ezplot(f)
```
3D:
```
xr=0:0.1:1; yr=0:0.1:1; [x y]=meshgrid(xr,yr); z=x+y.*z;
```
then
```
mesh(x,y,z) surf(xmat,ymat,zmat) or surf(xvec,yvec,zmat)
countourf(x,y,z,n)=countour(x,y,z,n,'style')+pcolor(x,y,z)
colormap(name(n)); name=jet/hot/cool/gray/bone/summer/autumn/spring
colorbar;
```
Dual axis:
```
ylims=get(gca,'YLim'); xlims=get(gca,'XLim');
newax=axes('position',get(gca,'position'));
set(newax,'YAxisLocation','right','color','none', ...
    'xgrid','off','ygrid','off','box','off','XTick',[],...
    'YLimMode','manual','YLim'ylims*scale);
```

Labels

set(get(gca,’?label’),’String’,’Prop1’,Val1,…))   or   title/?label(‘String’,’Prop1’,val,…)

Text properties

FontAngle	‘normal’ | ‘italic’ | ‘oblique’	
FontName	‘Courier’ | ‘Fixed-width’ | ….	
FontSize	Size in units	
FontUnits	‘points’ | ‘normalized’ | ‘inches’ | ‘centimeters’	Defaults to POINTS
FontWeight	‘light’ | ‘normal’ | ‘demi’ | ‘bold’	
VerticalAligment	‘middle’ | ’top’ | ‘cap’ | ‘baseline’ | ‘bottom’	
HorizontalAlignment	‘left’ | ‘center’ | ‘right’	
Rotation	Scalar	0=default

The string can also contain TEX

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
\alpha	α	\upsilon	υ	\sim	~
\beta	β	\phi	ϕ	\leq	≤
\gamma	γ	\chi	χ	\infty	∞
\delta	δ	\psi	ψ	\clubsuit	♣
\epsilon	ε	\omega	ω	\diamondsuit	♦
\zeta	ζ	\Gamma	Γ	\heartsuit	♥
\eta	η	\Delta	Δ	\spadesuit	♠
\theta	θ	\Theta	Θ	\leftrightarrow	↔
\vartheta		\Lambda	Λ	\leftarrow	←
\iota	ι	\Xi	Ξ	\uparrow	↑
\kappa	κ	\Pi	Π	\rightarrow	→
\lambda	λ	\Sigma	Σ	\downarrow	↓
\mu	∝	\Upsilon	ϒ	\circ	≡
\nu	ν	\Phi	Φ	\pm	±
\xi	ξ	\Psi	Ψ	\geq	≥
\pi	π	\Omega	Ω	\propto	∝
\rho	ρ	\forall	∀	\partial	∂
\sigma	σ	\exists	∃	\bullet	•
\varsigma		\ni	∍	\div	|
\tau	τ	\cong	≅	\neq	≠
\equiv	≡	\approx	≈	\aleph	ℵ
\Im	ℑ	\Re	ℜ	\wp	℘
\otimes	⊗	\oplus	⊕	\oslash	∅
\cap	∩	\cup	∪	\supseteq	⊇
\supset	⊃	\subseteq	⊆	\subset	⊂
\int	∫	\in	∈	\o	
\rfloor	⌋	\lceil	⌈	\nabla	∇
\lfloor	⌊	\cdot	·	\ldots	...
\perp		\neg	¬	\prime	∍
\wedge	∧	\times	x	\0	∅
\rceil	⌉	\surd	√	\mid	|
\vee	∨	\varpi		\copyright	©
\langle	〈	\rangle	〉		

_{} subscript   ^{} superscript   \fontname{name}   \fontsize{size}   \bf   \it   \rm

{‘line1’,’line2’}

Mapping toolbox in Matlab

Old Grid (Lat/Lon gridded, topographical, global satellite)

map=rxc matrix of z values
legend=[cells/angleunit north-latitude west-longitude] (nb: always lat/lon rhomboids)

file loading: dted, etopo5, globedem, gtopo30, satbath, tbase, usgs24kdem, usgsdem, avhrrgoode, avhrrlambert
creation: nanm,onem,zerom + population of legend
display: meshm, contourm, countour3m, contourfm

z=ltln2val(map,legend,lat,lon[,method]);[lat,lon]=setltln(map,leg,row,col);[row,col]=setpostn(map,legend,lat,lon);
[lat,lon]=findm(map boolexpr,legend);
[latlims,lonlims]=limitm(map,legend); vs. [r,c,legend]=sizem(latlims,lonlims,cellsperdeg);
gradientm(map,legend); viewshed, los2 (line of sight & view); areamat; maskm()
neworig, resizem

→Grid: refvec2mat(legend,size(map)); Z=map;
←Grid: refmat2vec(R,size(Z));map=Z; (only possible if R is for lat/lon data);

Grid (high resolution/projected in metric)

map=rxc matriz of z values
R=affine matrix such that [row col 1]*R=[x,y] coordinate (nb: may or may not be lat/lon & rhomboidal)

file loading: arcgridread, geotiffread (& geotiffinfo), sdtsdemread (&sdtsinfo), worldfileread
creation: matrix+makerefmat(x11center,y11center,xpixwidth,ypixwidth) (nb: ypixwidth<0 if y decereases w/ row)
display: mapshow (geoshow if lat/lon), also simple surf,mesh work if don't need coordinates

[x,y]=pix2map(R,r,c); [r,c]=map2pix(R,x,y); also latlon2pix & pix2latlon if coords in lat/lon (handles 360 wrap)
[x,y]=pixcenters(R,size(Z)[,'makegrid'])

→Geolocated: [x,y]=pixcenters(R,size(Z),'makegrid');[lat,lon]=projinv(mstruct,x,y); also meshgrat
←Geolocated: [Z,R]=geoloc2grid(lat,lon,z,cellsize)

Geolocated

lat,lon,z (values at anyshaped "grid" with centers at lat/lon) (may be a graticule if lat,lon smaller than z)
display: surfm, contourm, countour3m, contourfm
←old grid: latlim=[min(lt(:)) max(lt(:))]; lonlim=[min(lon(:)) max(lon(:))];
[map,legend]=nanm(latlim,lonlim,newpixpercurpix); map=imbedm(lat,lon,z,map,legend);
→old grid: [lat,lon]=meshgrat(map,legend);

Point/Line/Poly

lat,lon [z] ([la1 .. lan NaN la2 … la2n …] where NaN separates lines or patches (lan=la1 if patch)

file loading: usahi, usalo, worldhi,worldlo, coast
also .mat files: coast, oceanlo, usahi, usalo, worldhi, worldlo

display: linem (noreset), plotm (resets map), plot3m, fillm, fill3m, patchm (shading in patch), mapshow, geoshow
[x,y]=mfwdtrans(proj,lat,lon) for matlab maps projfwd(proj,lat,lon) for 3rd party
then can use: plot, line, etc
[mstruct,msg] = gcm;[x,y,z,savepts] = mfwdtran(mstruct,lat,lon,z,'surface'); h = patch('faces',tri,'vertices',[x(:) y(:) z(:)],'facevertexcdata',z(:), 'CDataMapping','scaled','facecolor','interp','edgecolor','none');

bufferm, reducem, interpm, interplat, interplon, nanclip, polybool, polycut, polyjoin, polymerge, polysplit, polyxpoly, areaint, areaquad, maptriml, maptrimp,
→ grid: vec2mtx, country2mtx, encodem

Geostructs

Version 2: struct(n).fields where fields are: Geometry='Line'|'Patch'|'Point', Lat/X, Lon/Y
BoundingBox:[minx minY;maxX maxY] if not Point, as many others as desired

fileload: shaperead (also shapeinfo);geotiff2mstruct
display: mapshow, geoshow, makesymbolspec (different symbols for each layer/attribute values)

updategeostruct, extractfield

Version 1: type='line'|'patch'|'text'|'surface' (geolocated grid)|'regular' (grid)
also: tag, lat,long,altitude,otherproperty and possibly map, maplegend, meshgrat, string depending on type
file loading: dcwdata,dcwgaz,dcwrdx,dcwread,dcwrhead,tgrline,tigermif,tigerp,vmapDdata,vmapDrdx
also: usalo, usahi, worldlo,world hi
display: displaym, mlayers

updategeostruct, extractm,country2mtx

Projections & display

maps; % lists all projections
axesm creates m=gcm; setm(m), getm(m); clma [all|purge] mfwdtrans,minvtrans; projfwd, projinv
framem; gridm; mlabel (meridians = lon), plabel (parallels=lat); scalerrule; axesmui (gui);tightmap;showaxes;
demcmap; polcmap (colormaps); colorbar; caxis([lo hi]); clrmenu (gui);
textm;gtextm;inputm;
Guis: axesmui, clrmenu, lightmui, origimui, panzoom, parallelui

Utilities

clipdata, trimdata (structs), maptriml, maptrimp (lat/lon)
equal area conversion: eqa2grn, grn2eqa
statistics: hista, histr,stddist, meanm, stdm,
conversion: deg/rad/nm/sm/km 2 deg/rad/nm/sm/km (e.g. deg2km)
hr2hms etc.
almanac('earth',['radius'|'volume'|'geoid'|'surfarea'[,'everest'|'clarke66'|…[,'km'|'deg'|'nm'|'sm'|'rad'|'meters']]])

.MAT files

coast.mat	polygons	(lat, long)
worldlo.mat	geostructs	Dnline (drainage), DNPatch, POline (political) POpatch, POtext, PPpoint (Populated places), Pptext
worldmtx.mat	grid (1°x1°)	map (195 countries), maplegend, nations (195 names),clrmap (useful colormap)
oceanlo.mat	grid	oceanmask
topo.mat	grid (1°x1°)	topo, toplegend, topomap1, topmap2 (colormaps)
usalo.mat	geostruct	conus, greatlakes, state, stateborder, gtlakelat, gtlakelon (patch), states (line)
usahi.mat	geostruct	statelin, statepatch, statetext
usamtx.mat	grid	mpa, maplegend, clrmap (useful colormap), states (names)

3rd Party

Worldwide 1°x1°

[map,maplegend]=etopo5(scale,latlim,lonlim)
[map,maplegend]=tbase(scale,latlim,lonlim)

Worldwide 1km x 1km

[map,maplegend]=dted(file)
[map,maplegend]=gtopo30(file,scale,latlim,lonlim) edcwww.cr.usgs.gov/landdac/gtopo30/gotopo30

Vegetation & AVHRR

[map,maplegend]=avhrrgoode('global',file,scale,latlim,lonlim)