**Improve KEYS for trickier options**

Update May 17, 2018 PW: I think these were all implemented and just kept for the record. I changed all $ to ? since that is the letter we were using.

Here is the list of secondary inputs and outputs that require nontrivial keys. In some cases, it may make more sense to introduce a backwards compatible newer syntax that we will insist must be used in mex.

**filter1d -Ff<table>**

If filter type **f** is given, then we expect a filename with a table of weights in one column. These could come from mex, hence **–Ff** should become **–Ff?** in GMT\_Encode\_Options. Also, grdfilter allows a leading minus-sign to indicate high-pass filter. We should extend this concept to filter1d I think but we will switch to using a modifier, +h. Hence the new syntax would be **–Ff***arg***[+h**].

Possible mex syntax **–Ff**[**+***modifiers*] After encoding: -**Ff**?[**+***modifiers*]

**gmtmath -A[-]<ftable>[+s|w|e]]**

The problem here is the optional leading hyphen which means “only load in the rhs of Ax = b”. Also there are modifiers **+s**, **+w**, and **+e** to skip. New syntax seems preferable and I suggest **–A***file***[+e**][**+s**|**w**][**+r**].

Possible mex syntax **–A**[**+s**|**w**][**+e**][**+r**] After encoding: **–A**?[**+s**|**w**][**+e**][**+r**]

**gmtselect -C[-|+]<dist>[<unit>]/<ptfile> -L[p][-|+]<dist>[<unit>]/<lfile>**

These both have similar syntax in the sense a filename is required and added after a trailing slash. In external interfaces these would be missing. Old convoluted syntax best replaced with newer method: **–C***file***+d**<*distance*> **-L***file***+d**<*distance*>[**+p**]

Possible mex syntax –C+d<distance> After encoding: –C?+d<distance>

Possible mex syntax –L+d<distance>[+p] After encoding: –L?+d<distance>[+p]

**gmtspatial -D[+f<file>][+a<amax>][+d[-|+]<dist>[<unit>]][+c|C<cmax>][+l][+s<sfact>][+p]]**

**-N<pfile>[+a][+p<ID>][+r][+z]]**

In the first case a file is given via **+f**<name> and this is not a required modifier. In the second the file is given after **–N** but type changes with modifier **+r**. We have that in the key but not tested fully.

Possible mex syntax: -D[+f][+…] After encoding: -D[+f?][+…]

Possible mex syntax: -N[+…]] After encoding: -N?[+…]

**grdcontour -A[-|[+]<annot\_int>]+t|T<file> -D<template>**

The **–D** should work if nothing is given in mex and all contours would be returned in one dataset. The deep annotation option to save the annotation strings could be recognized but these records are currently written to a local file via fopen. We would need to accumulate these to a TEXTSET and write this at the end of the program. Also, the label-placing mechanisms –Gx<file> and –Gf<file> would need to use GMT\_Read\_Data to allow this to be passed via mex.

Possible mex syntax: -A…+t|T[+…] After encoding: -A…+t|T?[+…]

**grdfilter -F<type>[-]<grdfile>**

See filter1d. Replace the – with +h but backwards compatible of course.

Possible mex syntax –Ff[+modifiers] After encoding: -Ff?[+modifiers]

**grdimage -A<out\_img=driver>**

If we wanted this to be possible we would need to add code to write this image directly with GMT\_Write\_Data but only when the file is a memory file. In this case there is no driver as we just return the image structure.

Possible mex syntax –A After encoding: -A?

**grdtrack -S[<method>]+sstackedfile**

One issue is that if stackedfile is not given (on the command line) then it defaults to stacked\_profile.txt instead. But in mex we would want **+s** to mean “return me”. May be best to remove the default output name and require it?

Possible mex syntax -S[<method>]+s After encoding: -S[<method>]+s?

**greenspline -A[<format>,]<gradientfile> -C[n|v]<cut>[/<file>]**

Nonstandard syntax with a leading integer and comma before the file. The –C option places the file after a slash. Best to change the syntax to avoid the comma so I propose–A<filename>+<fmt> and C[n|v]<cut>[+f<file>]. Note: This must propagate to new module gpsgridder as well.

Possible mex syntax –A+f<fmt> After encoding: -–A?+f<fmt>

Possible mex syntax -C[n|v]<cut>+f After encoding: -C[n|v]<cut>+f?

**mapproject -L<ltable>[/[+|-]<unit>]][+]**

Filename is given right after option but additional metadata can follow a slash. With a filename and directory path this is ugly anyway. Propose new syntax mapproject -L<ltable>[+u[-|+]<unit>][+p]

Possible mex syntax -L[+u[-|+]<unit>][+p] After encoding: -L?[+u[-|+]<unit>][+p]

**pscontour -A[-|[+]<annot\_int>]+t|T<file> -D<template>**

Same issue and solution as for grdcontour.

**psconvert -L<listfile>**

The list of files to process could come from a list but we would first need to read the list with GMT\_Read\_Data. No syntax change needed but add LT( to the keys.

**psmask -D<template>**

Same issue and solution as for grdcontour.

**psrose –C[<name>]**

in mex we cannot have –C mean two different things (which works on the command line), so I suggest we use –Cm to indicate “calculate mean direction”, as in gmtvector -Am.

Possible mex syntax -C After encoding: -C?, but if –Cm then nothing.

**gmtgravmag3d -T<[d]xyz\_file>/<vert\_file>[/m]|<[r|s]raw\_file>**

Unconventional option **–T** with two files given. This would need to be split into two arguments to enable mex input, e.g. –Td and –Tv perhaps.

**grdredpol <anomgrid> -G<rtp\_grdfile> [-E<dip\_grd>/<dec\_grd>]**

Unconventional option **–E** with two grid files given. This would need to be split into two arguments to enable mex input, e.g. **–Ei** and **–Ee** perhaps. Key EG(2.

**x2sys\_cross <files> -A<combi.lis>**

**x2sys\_datalist -I<ignorelist>**

**x2sys\_list -I<ignorelist>**

**x2sys\_report -I<ignorelist>**

All the x2sys modules use a read list function that returns an array of char strings. This could be replaced with a GMT\_Read\_Data call that returns a textset. Then these would just need a key AT( etc. depending on option letter.

**Actions:**

1. Allow high-pass modifier +h in filter1d. Then, both filter programs would have the syntax –Ff<file>[+h]. Apply same syntax with backwards support in grdfilter.
2. Let GMT\_Encode\_Options handle optional key-modifiers +<mod>?, meaning if +<mod> is found with no argument then we insert an implicit ?.
3. Let GMT\_Encode\_Options handle key-modifiers =+ which means “strip off anything from the first + and see if there is an argument before it”.
4. Change gmtmath **–A** syntax to pass the “-“ modifier via a new **+r** for “only place for right-hand side”. New syntax –A<file>[+e][+|s|w][+r].
5. Modernize gmtselect syntax to be –C<file>+d<distance> -L<file>+d<distance>[+p]
6. Rewrite the gmt\_contlabel\_save\_begin/end functions to avoid fopen/fprintf and instead add records to a textset, then write the textset to specified file. Add keys AT)+t+T or something.
7. Enable the label-specifications via –Gf and –Gx to be handed by GMT\_Read\_Data. This extends to both quoted and decorated lines as well as contours.
8. Introduce –Cm in psrose to calculate mean direction. This will be backwards compatible for commandline version of psrose.
9. Disable default outputname in grdtrack –S…+s when called from mex. Alternatively, avoid default names and require a name. Not really happy with either solution.
10. Change greenspline syntax to –A<filename>+<fmt>
11. Change mapproject syntax to –L<file>[+u[-|+]<unit>[+p] for easier parsing.
12. Let psconvert read its list via GMT\_Read\_Data into a textset. Add key LT(
13. Split –T option in gmtgravmag.c to allow mex input.
14. Split –E option in grdredpol.c to allow mex input.
15. Mgd77list: Switch to using GMT i/o with GMT\_Put\_Record, and because output is either text or data depending on choices the KEYS needs to have a ? and a special check is needed in GMT\_Encode\_Data.
16. X2sys\_\*: Use GMT\_Read\_Data when building those file lists so they could come from mex. Add suitable key.

I think actions 2-3 would allow all GMT syntax to be handled if the other changes are done as well.