FITS Embedded Function Proposal

Jonathan McDowell

1998 Feb 9

1. This document describes a proposal the storage of a function or functional in a FITS binary table HDU. The HDU provides a recipe for returning a function or function value given the values of its parameters. The proposal would have the status of an ASC internal convention.

2. The HDU represents a function-valued function, called the table function. The table function is a mapping from an n-dimensional space, whose axes are called the table function parameters, to a space of functions, called the return function space. In other words, you give the HDU an n-tuple of numbers and it gives you back a function, which we call the return function. The return function space is not arbitrary; it is required to be the set of functions which are restrictions of a single, specified, function (the model function) of m+p parameters to an m-dimensional domain. We call the m parameters of the return function the 'independent variables' and the remaining p parameters of the model function the 'model function parameters'.

   For example: Consider modelling an instrument point spread function P(x,y) as a circular gaussian G(x,y,sigma,norm) whose sigma and norm depend further on the external parameters energy and off-axis angle theta. Then we have:

   - Model function $G : R^4 \rightarrow R = Gauss(x, y, sigma, norm)$ (i.e. G is a function from 4D space to 1D space).
   - Independent variables: x,y
   - Model function parameters: sigma, norm
   - Return function $P : R^2 \rightarrow R = P(x, y)$, the restriction of G(x,y,sigma,norm) to the 2-D subdomain given by a particular choice of sigma and norm.
   - Table function parameters: energy, theta
   - Table function: $T : R^2 \rightarrow F(R^2 \rightarrow R)$, (i.e. T is a function from 2D space to the space of functions which map 2D space to 1D); T(energy,theta) = P(x,y) returns a function of x and y for any given energy and theta.

3. The model function may be the constant function with the single parameter of its value, in which case the table function is a simple map from an n-tuple to a function value. In this case, if it is further true that the table function parameters are all tabulated on regular constant-spaced intervals, the standard FITS IMAGE storage mechanism can and should be used.

4. The HDU shall be recognized as containing a function by the (mandatory) presence of the keyword

   HDUCLAS1 = 'FUNCTION'

   The value of HDUCLASS may be 'ASC' and the value of HDUCLAS2 may indicate the function type.

5. A good, but not compulsory, choice of EXTNAME is 'FUNCTION'.

6. The mandatory keyword FUNCNAME gives the name of the table function. This name will be recognized by other software as the user name of this function, e.g. in the ASCFIT model syntax.

7. The keyword MODEL gives the definition in ASC model syntax of the model function. The special value 'None' for a constant function indicates that a function value is returned; alternatively, MODEL may be omitted. Use of COMMENT keywords to explain the model function definition is encouraged.

8. The mandatory keyword FNAXIS gives the number of table function parameters.

9. The mandatory indexed keywords FTYPEi (i = 1 to FNAXIS) give the name of the ith table function parameter. This name must be identical to one of the TTYPE values. (We key off the name rather than the column number so as to be more robust to changes by file manipulation tools.)

10. The optional indexed keywords FNAXISi give the number of tabulated values for the ith table function parameter. This only makes sense if the same number of values for FTYPE1 are tabulated at each value of FTYPE2, etc., in other words the grid of points at which the function is tabulated is regular like

```
FTYPE1 ---->
x...x.x..xx  |
...........  |
x...x.x..xx  |
...........  |
...........  V
x...x.x..xx FTYPE2
x...x.x..xx
```

   as opposed to

```
FTYPE1 ---->
x...x.x..x.  |
...........  |
x.....x..xx  |
..x........  |
...........  V
....x.x...x FTYPE2
x...x.x..xx
```

The latter form is more difficult to interpolate in, but may be necessary in practice as many measurements do not occur on fully sampled grids (e.g. a measurement at 6.5 keV may not have been done for one off-axis angle). Hence, the presence of the FNAXISi keywords indicates that for any tabulated value of the other parameters, you always get FNAXISi values of parameter i and moreover they will be the same values. Further, the total number of rows will be the product of all the FNAXISi.

11. The model parameters are given as table columns, using names consistent with the ASCFIT model syntax. However, if a model parameter has a constant value for all the rows, it may be stored as a keyword instead.

12. For the constant model, the column name giving the model value shall be the same as the table function name (value of FUNCNAME).

13. The keyword NPARAM gives the number of parameters.

14. The indexed keywords PARAMi repeat the names of the model parameters, so that the columns or keywords containing model parameters can be easily identified.

15. Other columns may be present in the table and do not affect the interpretation of the function. For instance, if the table is used to store the results of a fitting process, one of the table function parameters might be 'iteration number' and an additional column might be the goodness of fit parameter.

16. Note: ASCFIT distinguishes between a model's independent variables and its parameters. If we want to use the present convention as providing a user-defined table model for ASCFIT, we may need extra keywords to say which of the table function parameters are parameters in the ASCFIT sense, and which are independent variables. For example, suppose in example 1 below we want to fit for the focal length given measurements of the effective area. We then want Energy and Theta as independent variables and F as a parameter. We could add keywords to the file to say this. However, next time we might want to fit for theta too; it may be better to make ASCFIT able to choose on the fly which arguments to treat as parameters.

EXAMPLE 1: Simple function. We tabulate the HRMA effective area as a function of energy, off-axis angle, and assumed focal length: HRMA_AREA(ENERGY,THETA,F).

```
XTENSION = 'BINTABLE'
BITPIX   =         8
NAXIS    =         2
NAXIS1   =        16
NAXIS2   =        12
PCOUNT   =         0
GCOUNT   =         1
TFIELDS  =         4
TTYPE1   = 'ENERGY  '    / Energy
TUNIT1   = 'keV     '
TFORM1   = '1E      '
TTYPE2   = 'THETA   '     / Off-axis angle
TUNIT2   = 'arcmin  '
TFORM2   = '1E      '
TTYPE3   = 'F       '     / Focal length
TUNIT3   = 'mm       '
TFORM3   = '1E       '
TTYPE4   = 'HRMA_AREA'
TUNIT4   = 'cm**2   '
TFORM4   = '1E      '
EXTNAME  = 'AREA_TABLE'
HDUCLASS = 'ASC     '
HDUCLAS1 = 'FUNCTION'
HDUCLAS2 = 'TABLE'
COMMENT
COMMENT  Mission and instrument are relevant when units include counts
COMMENT  and pixels
COMMENT
MISSION  = 'AXAF    '
TELESCOP = 'AXAF    '
FUNCNAME = 'HRMA_AREA'   /
FNAXIS   =           3  /
FTYPE1   =  'ENERGY'    /
FTYPE2   =  'THETA'     /
FTYPE3   =  'F    '     /
FNAXIS1  =           2  /
FNAXIS2  =           3  /
FNAXIS3  =           2  /
MODEL    = 'None'       /

ENERGY      THETA     F      HRMA_AREA
```

1.5	0.0	10065.0	4.8
6.4	0.0	10065.0	2.3
1.5	1.0	10065.0	4.6
6.4	1.0	10065.0	2.1
1.5	2.0	10065.0	4.0
6.4	2.0	10065.0	1.8
1.5	0.0	10258.0	5.8
6.4	0.0	10258.0	3.3
1.5	1.0	10258.0	5.6
6.4	1.0	10258.0	3.1
1.5	2.0	10258.0	5.0
6.4	2.0	10258.0	3.8

EXAMPLE 2: Functional case. We give the ACIS PSF, which is modelled as a functional of summed gaussians. The table indicates the values of the gaussian parameters as a function of the independent variables.

```
XTENSION = 'BINTABLE'
BITPIX   =         8
NAXIS    =         2
NAXIS1   =        32
NAXIS2   =         6
PCOUNT   =         0
GCOUNT   =         1
TFIELDS  =         8
TTYPE1   = 'ENERGY  '    / Energy
TUNIT1   = 'keV     '
TFORM1   = '1E      '
TTYPE2   = 'THETA   '     / Off-axis angle
TUNIT2   = 'arcmin  '
TFORM2   = '1E      '
TTYPE3   = 'CORE.SIGMA'
TUNIT3   = 'pixel   '
TFORM3   = '1E      '
TTYPE4   = 'CORE.NORM'
TUNIT4   = 'count    '
TFORM4   = '1E      '
TTYPE5   = 'HALO.X0  '        / Halo center X0
TUNIT5   = 'pixel    '
TFORM5   = '1E      '
TTYPE6   = 'HALO.Y0  '        / Halo center Y0
TUNIT6   = 'pixel    '
TFORM6   = '1E      '
TTYPE7   = 'HALO.SIGMA'
TUNIT7   = 'pixel    '
TFORM7   = '1E      '
TTYPE8   = 'HALO.NORM'
TUNIT8   = 'count    '
TFORM8   = '1E      '
TTYPE9   = 'TRW_ID   '  / Test used to derive this data point
TFORM9   = '20A     '  / (Example of an extraneous column)
EXTNAME  = 'FUNCTION'
HDUCLASS = 'ASC     '
HDUCLAS1 = 'FUNCTION'
HDUCLAS2 = 'COMPOUND'
COMMENT
COMMENT  Mission and instrument are relevant when units include counts
COMMENT  and pixels
```

```
COMMENT
MISSION  = 'AXAF    '   /
TELESCOP = 'AXAF    '   /
INSTRUME = 'ACIS    '   /
DTYPE1   = 'CORE.X0'    / Function parameter
DVAL1    =        0.0   / [pixel]
DTYPE2   = 'CORE.Y0'    / Function parameter
DVAL2    =        0.0   / [pixel]
FUNCNAME = 'ACIS_PSF'   /
FNAXIS   =           2  /
FTYPE1   =  'ENERGY'    /
FTYPE2   =  'THETA'     /
FNAXIS1  =           2  /
FNAXIS2  =           3  /
MODEL    = 'Gauss2D[Core]+Gauss2D[Halo]'
NPARAM   =           8  /
PARAM1   = 'CORE.SIGMA' /
PARAM2   = 'CORE.NORM'  /
PARAM3   = 'CORE.X0  '  /
PARAM4   = 'CORE.Y0  '  /
PARAM5   = 'HALO.SIGMA' /
PARAM6   = 'HALO.NORM'  /
PARAM7   = 'HALO.X0  '  /
PARAM8   = 'HALO.Y0  '  /

ENERGY      THETA     CORE.SIGMA CORE.NORM  HALO.X0 HALO.Y0 HALO.SIGMA HALO.NORM  TRW_ID

1.5          0.0
6.4          0.0
1.5          1.0
6.4          1.0
1.5          2.0
6.4          2.0
```