GammaLib - Bug #1457

GCTAAeff2D effective area interpolation behaves weirdly outside given fits file range

04/24/2015 01:51 PM - Kelley-Hoskins Nathan

Status:	Closed	**Start date:**	04/24/2015
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.1.0		

Description

I have effective area data saved to a fits file ('EFFECTIVE AREA', columns: ENERG_LO, ENERG_HI, THETA_LO, THETA_HI, EFFAREA, EFFAREA_RECO), and read in to a GObservation via GCTAAeff2D().

When I then get the effective area from the GObservations object, at different energies and theta offsets (from camera center), the interpolation works fine within the bounds of the given fits file (the green box in the attached plot). But, when you ask for the effective area where *both* the energy and offset are higher (the red box in the plot) than the ones given in the fits file, the interpolation starts to return very high, incorrect values. The further in energy/offset you go, the higher the incorrect values become.

Here's the code I used to get the effective area:

```
import gammalib

obs = load_runs() # custom loading function, loads event lists/irfs from fits files, returns GObservations() object

# pick energy range and # of bins
energy_min, energy_max = gammalib.GEnergy(), gammalib.GEnergy()
nenergy = 15
energy_min.GeV( 1.0)
energy_max.TeV(50000.0) # way larger than normal

# pick theta offset range and # of bins
ntheta, theta_min, theta_max = 15, 0.0, 3.5 # 3.5 also way larger than normal

# bin widths
edelta = ( energy_max.log10TeV() - energy_min.log10TeV() ) / nenergy
thetadelta = ( theta_max - theta_min ) / ntheta

# get the first observation
run = gammalib.GCTAObservation( obs[0] )

# loop over each bin in each theta/energy axis,
# and get the effective area at the bin center
for i in range(nenergy) :
  en = ( (i+0.5) * edelta ) + energy_min.log10TeV()
  for j in range(ntheta) :
    th = ( (j+0.5) * thetadelta ) + theta_min

    # get the effective area
    eff = run.response().aeff()( en, th*gammalib.deg2rad) # cm^2
    print 'en=%f log10(TeV)    theta=%f deg    effarea=%fcm^2' % ( en, th, eff )
```

I recognize that fov's and energies that large aren't physically possible with current-gen detectors, and I'm not sure if there's some checks to prevent accessing this region of the parameter space, but I thought I should post an issue about it, since that area should return 0.0 .

Related issues:

Related to GammaLib - Action # 1784: Use of the IRF with different energy bou...	**Closed**	**05/31/2016**

History

#1 - 05/19/2015 07:55 PM - Knödlseder Jürgen

The current logic is that beyond the data points a linear extrapolation is done. Typically this should lead to decreasing values (that may even become negative, through there is an explicit limitation to non-negative values in the code). But ultimately this depends on the value of the two last bins that are used for extrapolation.

We could think about returning zero in case that values beyond the tabulated range are requested.

#2 - 05/19/2015 07:55 PM - Knödlseder Jürgen

- Project changed from ctools to GammaLib

#3 - 05/19/2015 11:34 PM - Kelley-Hoskins Nathan

So, in the fits file, if I border the existing energy/offset parameter space with one or two rows of zeros, it should help reduce that extrapolated area?

#4 - 05/19/2015 11:50 PM - Knödlseder Jürgen

Kelley-Hoskins Nathan wrote:

> So, in the fits file, if I border the existing energy/offset parameter space with one or two rows of zeros, it should help reduce that extrapolated area?

In principle yes, but better test this and make sure that it does indeed work.

#5 - 06/21/2016 10:07 PM - Knödlseder Jürgen

- Related to Action #1784: Use of the IRF with different energy bounds with respect to the default added

#6 - 06/21/2016 10:07 PM - Knödlseder Jürgen

- Target version set to 1.1.0

#7 - 06/21/2016 11:53 PM - Knödlseder Jürgen

- Status changed from New to Closed

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 100

The effective area outside the boundaries covered by the FITS file are now set to zero.

Code has been merged into devel.

Files

plot.png	55.8 KB	04/24/2015	Kelley-Hoskins Nathan