GammaLib - Action #85

Feature # 83 (Closed): Implement CTA interface for 1DC data format

Implement 1DC PSF file format

02/29/2012 06:34 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	03/07/2012
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	20.00 hours
Target version:	00-05-00		

Description

Related issues:

Blocked by GammaLib - Action # 96: Need final agreement of 1DC PSF format	**Closed**	**03/07/2012**

History

#1 - 03/07/2012 12:04 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- Start date set to 03/07/2012

#2 - 03/07/2012 12:23 PM - Knödlseder Jürgen

- % Done changed from 0 to 10

So far, GCTAResponse uses 3 methods for PSF handling: psf_dummy, psf_dummy_sigma, and psf_dummy_max. These methods are used in the following GCTAResponse methods:

psf_dummy:

- psf - PSF evaluation
- irf_kern_omega - kernel for PSF integration
- npsf_kern_rad_azsym - kernel for NPSF integration

psf_dummy_sigma:

- mc - here the Gaussian shape of the dummy PSF is explicitly used in the Random number generator; each shape needs thus a particular random number generator
- read_arf - here the sigma is used to determine the appropriate thetacut correction by calling npsf_kern_rad_azsym
- irf_extended - here the sigma is passed to irf_kern_rho for integration
- npred_extended - here the sigma is used to compute the maximum PSF radius using psf_dummy_max to set the integration boundary
- psf - PSF evaluation
- psf_delta_max - computes the maximal angular separation to be accounted for
- npsf - here the sigma is used for integration range computation; it is also passed to npsf_kern_rad_azsym for computations

psf_dummy_max:

- npred_extended - for integration range computation
- psf_delta_max - computes the maximal angular separation to be accounted for
- npsf - for integration range computation

So there are at the end 3 fonctionnalities that are required:

- setting a specific parameter set based on energy (actually psf_dummy_sigma)
- determining the maximum PSF radius for a given parameter set (actually psf_dummy_max)
- evaluation of PSF for a given parameter set (actually psf_dummy)

The simplest approach is to replace the double precision value double sigma by a specific PSF parameter vector std::vector<double> parameters that is then interpreted based on the actual PSF version. This can be done using a simple typedef:

typedef std::vector<double> GCTAPsfPars;

#3 - 03/07/2012 02:38 PM - Knödlseder Jürgen

- % Done changed from 10 to 30

- Remaining (hours) changed from 20.0 to 10.0

I now added the offset angle theta to the PSF computation routines. The theta values is handled in the following way in the various GCTAResponse methods:

Correct theta angle computation:

- GCTAResponse::npsf
- GCTAResponse::mc

Approximate theta angle computation (see notes in code):

- GCTAResponse::irf_extended (this should be okay, but needs verification)
- GCTAResponse::npred_extended (this should be okay, but needs verification)

Missing theta angle computation:
GCTAResponse::read_arf (this is not critical at this stage)

#4 - 03/07/2012 04:40 PM - Knödlseder Jürgen

- % Done changed from 30 to 70

- Remaining (hours) changed from 10.0 to 5.0

The response table support is now implemented.

What is still missing is the support of Monte Carlo simulations for the response table. This be done by decomposing the MC process in a two step process:

- First select for which Gaussian we should make the simulation
- Then draw from the specific Gaussian

#5 - 04/25/2012 11:28 AM - Knödlseder Jürgen

- Status changed from In Progress to Resolved

#6 - 04/25/2012 11:28 AM - Knödlseder Jürgen

- % Done changed from 70 to 100

#7 - 04/25/2012 11:28 AM - Knödlseder Jürgen

- Status changed from Resolved to Closed

- Remaining (hours) changed from 5.0 to 0.0