

GammaLib - Action #539

Feature # 536 (Closed): Add HESS instrument response functions

HESS instrument response tests / examples / documentation.

10/10/2012 12:00 PM - Deil Christoph

Status:	Closed	Start date:	10/10/2012
Priority:	Normal	Due date:	
Assigned To:	Deil Christoph	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
The basic unit test would be checking the gammalib gives consistent results with the HESS software for the test scripts attached to issue #538 .			
A higher level test would be e.g. checking that a Crab spectrum for 100 runs is consistent with the HESS software, but that would first require implementing code similar to the HESS ReflectedBgMaker and SpectrumMaker.			
For documentation it could be nice to create a web page similar to http://www.slac.stanford.edu/exp/glast/groups/canda/lat_Performance.htm , only for HESS and CTA there will be too many parameters, so a dynamic page to investigate IRFs would be nice. We could write a tool that let's the user select observation parameters (zenith angle, ...) or a run list, and a target position, and then the effective area, energy resolution, PSF and sensitivity is computed. This would be actually useful for the scientists and allow us to show off all the hard work we'll put into IRFs.			

History

#1 - 10/10/2012 08:48 PM - Knödlseider Jürgen

- Target version set to HESS sprint #1

#2 - 10/10/2012 10:50 PM - Knödlseider Jürgen

Christoph Deil wrote:

The basic unit test would be checking the gammalib gives consistent results with the HESS software for the test scripts attached to issue #538 .

Right, we would need a one-to-one comparison. GammaLib now has GTestSuite for implementing unit test, which is pretty nice and automatically provides test reports in JUnit format. Any test should be inspired from the existing test that make use of this new class.

The test suite class is also available in Python, and unit testing can also be implement on the Python level (actually I do both, where the heavy class testing is done in C++ and the Python interface testing is done in the Python tests ; still, the Python test suite is very incomplete).

Overall, I like test driven development, i.e. first write the test and then develop the class. This helps even in thinking about how the class should be defined, as a use case is built before the class is coded.

A higher level test would be e.g. checking that a Crab spectrum for 100 runs is consistent with the HESS software, but that would first require implementing code similar to the HESS ReflectedBgMaker and SpectrumMaker.

I guess this would be part of an extended comparison that will be done once all code is implemented, but probably we won't put this in as unit test.

For documentation it could be nice to create a web page similar to http://www.slac.stanford.edu/exp/glast/groups/canda/lat_Performance.htm , only for HESS and CTA there will be too many parameters, so a dynamic page to investigate IRFs would be nice. We could write a tool that let's the user select observation parameters (zenith angle, ...) or a run list, and a target position, and then the effective area, energy resolution, PSF and sensitivity is computed. This would be actually useful for the scientists and allow us to show off all the hard work we'll put into IRFs.

This is indeed interesting, but a completely different field. I have in fact in mind to make somehow the ctools / GammaLib available through a web interface, and the first thing could consist of deriving the performance curves.

Before setting up such a service, some thinking is needed to decide upon the best web system that could support such a service (Drupal? something else?).

#3 - 11/03/2013 02:09 AM - Knödlseider Jürgen

- *Target version deleted (HESS sprint #1)*

#4 - 06/21/2016 10:22 PM - Knödlseider Jürgen

- *Status changed from New to Closed*

- *% Done changed from 0 to 100*