

GammaLib - Bug #1662

GCTAEdisp2D sometimes throws a warning on construction

02/11/2016 10:02 AM - Mayer Michael

<b>Status:</b>	Closed	<b>Start date:</b>	02/11/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Knödlseider Jürgen	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.1.0		
<b>Description</b>			
Here is what happens:			
<pre>import gammalib edisp = gammalib.GCTAEdisp2D("hess_edisp_023559.fits[EDISP_2D]") +++ WARNING in GIntegral::romberg(1.16536, 10, 5): Integration uncertainty 0.453982 exceeds absolute tolerance of 0.451215 after 21 iterations. Result 451215 is inaccurate. +++ WARNING in GIntegral::romberg(0.865363, 10, 5): Integration uncertainty 49.1817 exceeds absolute tolerance of 2.61044 after 21 iterations. Result 2.61044e+06 is inaccurate. +++ WARNING in GIntegral::romberg(0.578213, 10, 5): Integration uncertainty 22.0398 exceeds absolute tolerance of 10.9862 after 21 iterations. Result -1.09862e+07 is inaccurate. +++ WARNING in GIntegral::romberg(1.16536, 10, 5): Integration uncertainty 1.00613e-06 exceeds absolute tolerance of 1e-06 after 21 iterations. Result 1 is inaccurate. +++ WARNING in GIntegral::romberg(0.865363, 10, 5): Integration uncertainty 1.88404e-05 exceeds absolute tolerance of 1e-06 after 21 iterations. Result 1 is inaccurate. +++ WARNING in GIntegral::romberg(0.578213, 10, 5): Integration uncertainty 22.0398 exceeds absolute tolerance of 10.9862 after 21 iterations. Result -1.09862e+07 is inaccurate.</pre>			
The file is attached. I can't see any obvious problem when browsing through the file. Could it be too noisy?			

History

#1 - 02/11/2016 10:27 AM - Knödlseider Jürgen

I will look into that. Could indeed be the noise.

#2 - 02/11/2016 11:07 AM - Deil Christoph

We can add a smoothing step during FITS IRF production to reduce noise if that is the issue.

Concerning Gammalib / ctools, my request would be that the message is improved to say during which processing step the issue occurred (reading EDISP in this case).

#3 - 02/11/2016 11:36 AM - Knödlseider Jürgen

I agree it would be nice to know in which step the warning occurred, but this happens in a low level function which does not know who was the caller, and I'm not sure that you can easily recover the information.

#4 - 02/11/2016 11:43 AM - Deil Christoph

Python has a great exceptions / warning system.  
One can simply get stack traces of where warnings occurred.

I'm not very familiar with exceptions / warnings systems in Gammalib or C++ in general.  
It's probably possible to improve how Gammalib / ctools does this by using C++ exceptions, but I don't have a concrete suggestion.  
So Jürgen, maybe make a separate issue if you think there's something that could be improved, or we can just drop that more general question for now.

**#5 - 02/11/2016 12:29 PM - Knödseder Jürgen**

- *Status changed from New to In Progress*
- *Assigned To set to Knödseder Jürgen*
- *Target version set to 1.1.0*
- *% Done changed from 0 to 80*

Well, this is a warning, not an exception.

Anyways, warnings are there to indicate that something in the code is probably not correct. This was in fact also the case here: I discovered that energy boundaries were not computed correctly, so in some cases, this prevent the integrations to converge. I'm about to fix this and I will tell you when it's in devel.

**#6 - 02/11/2016 12:33 PM - Deil Christoph**

Well, this is a warning, not an exception.

Well, if you want more useful info for Gammalib warnings, namely the stack trace showing what caused the warning, you have to implement warnings as exceptions, no?

(There might be other ways ... I'm not a C++ expert)

Anyways, warnings are there to indicate that something in the code is probably not correct. This was in fact also the case here: I discovered that energy boundaries were not computed correctly, so in some cases, this prevent the integrations to converge. I'm about to fix this and I will tell you when it's in devel.

Thanks!

**#7 - 02/11/2016 12:41 PM - Knödseder Jürgen**

user#66 wrote:

Well, this is a warning, not an exception.

Well, if you want more useful info for Gammalib warnings, namely the stack trace showing what caused the warning, you have to implement warnings as exceptions, no?

(There might be other ways ... I'm not a C++ expert)

The idea of warnings is that they never should occur for production (I was even considering switching them off, but probably I forgot about that one).  
They are really there for code debugging.

**#8 - 02/11/2016 12:47 PM - Knödlseeder Jürgen**

- *Tracker changed from Support to Bug*
- *Status changed from In Progress to Closed*
- *% Done changed from 80 to 100*

Merged into devel.

**#9 - 02/11/2016 01:48 PM - Mayer Michael**

Thanks. Now the warnings are gone.

**Files**

---

hess_edisp_023559.fits	78.8 KB	02/11/2016	Mayer Michael
------------------------	---------	------------	---------------