GammaLib - Change request #1134

Flux calculation is wrong in an XML model of DiffuseSource x SpatialMap

02/06/2014 02:48 PM - Okumura Akira

Status:	Closed	**Start date:**	02/06/2014
Priority:	Urgent	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	00-09-00		

Description

When I simulate a diffuse source using "DiffuseSource" and "SpatialMap" with a 2D FITS image, the total flux is wrongly calculated. Because GammaLib automatically normalise the given FITS image, even though the flux should be calculated by multiplying <spectrum> in a unit of (/s/MeV/sr) to the intensity and the solid angle of each pixel of the image. It seems that GammaLib normalise the image to 1, and assumes that <spectrum> is written in a unit of (/s/cm2/MeV).

But I believe this is not compatible with the Fermi Science Tools definition.
http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/xml_model_defs.html#SpatialMap

I found that Jürgen had introduced this normalisation process intentionally a while ago.
https://cta-redmine.irap.omp.eu/issues/315
But I would like to remove this unnecessary normalisation for diffuse analyses. We, in the LAT diffuse group, usually calculate gamma-ray emissivity per H atom instead of integrated total flux. The intensity and the solid angle of a diffuse source are very important information for this.

In addition, I found that "Prefactor" of SpatialMap is ignored. Even if I change value from 1 to 2, the total number of photons generated in ctobssim remains same.

You can reproduce these behaviours with the attached FITS image and XMLs. The following is a list of the numbers of generated events with the attached XMLs.

model_point.xml -> 16032 events
model_extended.xml -> 12365 events (smaller than 16032 due to off-axis acceptance?)
model_diffuse.xml -> 12365 events (should NOT be same as ExtendedSource)
model_diffuse_x2.xml -> 12365 events (should be larger by a factor of 2)

```
$ ctobssim
Model [model_diffuse.xml]
Calibration database [$GAMMALIB/share/caldb/cta]
Instrument response function [kb_I_50h_v3]
RA of pointing (degrees) (0-360) [86.1716]
Dec of pointing (degrees) (-90-90) [-1.47746]
Radius of FOV (degrees) (0-180) [5]
Start time (MET in s) (0) [0]
End time (MET in s) (0) [360000]
Lower energy limit (TeV) (0) [0.1]
Upper energy limit (TeV) (0) [100]
Output event data file or observation definition file [events_diffuse.fits]
```

- tags/ctools-00-07-00
- tags/GammaLib-00-08-00

History

#1 - 02/06/2014 04:01 PM - Okumura Akira

Dear the system admin (Jürgen?)

Could you please remove co_OriB.fits from this page? I did not realize that this Redmine page was public.

#2 - 02/06/2014 04:41 PM - Okumura Akira

I took a look at the source code of GModelSky and GModelSpatialDiffuseMap.

```
void GModelSpatialDiffuseMap::load(const std::string& filename)
{
    // Initialise skymap
    m_map.clear();

    // Store filename of skymap (for XML writing). Note that we do not
    // expand any environment variable at this level, so that if we write
    // back the XML element we write the filepath with the environment
    // variables
    m_filename = filename;

    // Load skymap
    m_map.load(gammalib::expand_env(m_filename));

    // Prepare sky map
    prepare_map();

    // Return
    return;
}
```

```
void GModelSky::set_type(void)
{
    // Clear model type
    m_type.clear();

    // Continue only if we have a spatial model component
    if (m_spatial != NULL) {

        // Is spatial model a point source?
        if (dynamic_cast<const GModelSpatialPointSource*>(m_spatial) != NULL) {
            m_type = "PointSource";
        }

        // ... otherwise, is spatial model a radial source?
        else if (dynamic_cast<const GModelSpatialRadial*>(m_spatial) != NULL) {
            m_type = "ExtendedSource";
        }

        // ... otherwise we have a diffuse model
        else {
            m_type = "DiffuseSource";
        }

    } // endif: there was a spatial model component

    // Return
    return;
}
```

I would like to propose the following three items.

1. Rename GModelSpatialDiffuseMap::prepare_map to GModelSpatialDiffuseMap::normalize_map
2. Remove prepare_map(); in GModelSpatialDiffuseMap::load
3. Add (dynamic_cast<const GModelSpatialRadial*>(m_spatial))->normalize_map(); after m_type = "ExtendedSource";

Since I am not an expert of ctools or GammaLib, I may be writing something stupid. But I would appreciate it if you could review the implementation for diffuse sources.

#3 - 02/06/2014 05:36 PM - Martin Pierrick

- File deleted (co_OriB.fits)

#4 - 02/06/2014 09:23 PM - Knödlseder Jürgen

- Tracker changed from Bug to Change request

I relabeled this as Change Request.

The normalization is done in purpose so that the output of ctlike is correct (it gives a flux and units). I would argue that the thing is not done correctly in the Fermi Science Tools, as the number you get out of gtlike is in fact in arbitrary units. I had in fact several times people contacting me about the way how they have to prepare a map for Fermi analysis, and this is not necessarily a trivial task as the solid angles need to be computed correctly.

But I understand that you need some method or way to get the normalization factor, or to prevent normalization. So we have to think how to handle this. One could add an attribute to the XML file that prevents normalization (something like normalize="0" if no normalization is requested). In any case, I would store the normalization factor in the class so that it can be accessed for further usage. Alternatively, one could also add a header keyword in the FITS file to prevent normalization. But this would be somehow fancy, so I won't really recommend this.

The Prefactor is indeed ignored in the Monte Carlo simulations, as the spatial component is assumed to be always unit in GammaLib. I was in fact not expecting that somebody uses this Prefactor to scale things. I created bug #1135 for this issue.

#5 - 02/06/2014 10:23 PM - Okumura Akira

Hello Jürgen,

I suppose that you and your colleagues have already discussed on this spatial map design and philosophy. But please allow me to post my opinion here.

Knödlseder Jürgen wrote:

> I would argue that the thing is not done correctly in the Fermi Science Tools, as the number you get out of gtlike is in fact in arbitrary units.

I partially agree, but partially not. From a point of view as a software developer, the units used in a program should be consistent with each other. So I understand that calculating arbitrary units in gtlike or ctlike should be avoided. But from an astrophysical point of view, units used in FITS files are often arbitrary. For example, gas extinction map, e.g. E(B-V), is written in a "unit" of magnitude which cannot be expressed with MKSA units. CO maps, which are of course very important to analyse SNRs and diffuse emission, has a different unit (K km/s), N(HI) maps use (cm-2). If CFITSIO or GammaLib can handle these arbitrary units in a good unified way, I support your idea that we had better use the same unit in ctlike.

I have another reason to oppose your idea. Assuming that we have a diffuse gas template made from HI or/and CO when analysing the Galactic plane that is much larger than the CTA FOV, what is the physical meaning of the flux calculated in (/s/TeV/cm2)? Is this the total diffuse flux of the Galactic plane? Or is it the flux integrated in the FOV? Neither is the correct answer.

In addition, continuity and compatibility with Fermi Science Tools are important.

> I had in fact several times people contacting me about the way how they have to prepare a map for Fermi analysis, and this is not necessarily a trivial task as the solid angles need to be computed correctly.

I understand. So we had better have two options in ctools: "ExtendedSource" for SNRs, AGN lobes and so on, and "DiffuseSource" for literally diffuse sources such as molecular clouds and the Fermi bubble.

> But I understand that you need some method or way to get the normalization factor, or to prevent normalization. So we have to think how to handle this. One could add an attribute to the XML file that prevents normalization (something like normalize="0" if no normalization is requested). In any case, I would store the normalization factor in the class so that it can be accessed for further usage.

Adding a new option "normalize" is OK for me. But as I wrote above, differentiate "ExtendedSource" from "DiffuseSource" will work. What do you think?

> Alternatively, one could also add a header keyword in the FITS file to prevent normalization. But this would be somehow fancy, so I won't really recommend this.

~10% of the users do not know how to edit FITS files smile.png

> The Prefactor is indeed ignored in the Monte Carlo simulations, as the spatial component is assumed to be always unit in GammaLib. I was in fact not expecting that somebody uses this Prefactor to scale things. I created bug #1135 for this issue.

Thank you very much.

#6 - 02/06/2014 11:40 PM - Knödlseder Jürgen

Okumura Akira wrote:

> Hello Jürgen,
>
> I suppose that you and your colleagues have already discussed on this spatial map design and philosophy. But please allow me to post my opinion here.
>
> Knödlseder Jürgen wrote:
>
> > I would argue that the thing is not done correctly in the Fermi Science Tools, as the number you get out of gtlike is in fact in arbitrary units.
>
> I partially agree, but partially not. From a point of view as a software developer, the units used in a program should be consistent with each other. So I understand that calculating arbitrary units in gtlike or ctlike should be avoided. But from an astrophysical point of view, units used in FITS files are often arbitrary. For example, gas extinction map, e.g. E(B-V), is written in a "unit" of magnitude which cannot be expressed with MKSA units. CO maps, which are of course very important to analyse SNRs and diffuse emission, has a different unit (K km/s), N(HI) maps use (cm-2). If CFITSIO or GammaLib can handle these arbitrary units in a good unified way, I support your idea that we had better use the same unit in ctlike.

Ideally I would like to interpret the units of the FITS file, but almost nobody sets these correctly, hence we cannot reply on this possibility.

> I have another reason to oppose your idea. Assuming that we have a diffuse gas template made from HI or/and CO when analysing the Galactic plane that is much larger than the CTA FOV, what is the physical meaning of the flux calculated in (/s/TeV/cm2)? Is this the total diffuse flux of the Galactic plane? Or is it the flux integrated in the FOV? Neither is the correct answer.

Agree. This case is not well covered.

> In addition, continuity and compatibility with Fermi Science Tools are important.

Well, GammaLib provides it's own Fermi tools, and my goal was rather to convince Jim at some point to adopt a common XML format wink.png

> > I had in fact several times people contacting me about the way how they have to prepare a map for Fermi analysis, and this is not necessarily a trivial task as the solid angles need to be computed correctly.
>
> I understand. So we had better have two options in ctools: "ExtendedSource" for SNRs, AGN lobes and so on, and "DiffuseSource" for literally diffuse sources such as molecular clouds and the Fermi bubble.

The difference between ExtendedSource and DiffuseSource is that the first is a parametric model (where the parameters describe the extension and shape of the model), while the second is a map. SNRs and AGN lobes are thus a DiffuseSource, and from the code you cannot distinguish them from molecular clouds or Fermi bubbles.

> > But I understand that you need some method or way to get the normalization factor, or to prevent normalization. So we have to think how to handle this. One could add an attribute to the XML file that prevents normalization (something like normalize="0" if no normalization is requested). In any case, I would store the normalization factor in the class so that it can be accessed for further usage.
>
> Adding a new option "normalize" is OK for me. But as I wrote above, differentiate "ExtendedSource" from "DiffuseSource" will work. What do you think?

As described above, the logic behind is different, so I would prefer adding an attribute. What about adding a unit attribute? This can be even very general as an extension to the Fermi Science Tools format. If specified, GammaLib would read the unit from the attribute (each model parameter has in fact an attribute string), and we could use the attribute to switch between functionalities. For example unit="arbitrary" as in the example below could switch off the normalization:

```
<spatialModel type="SpatialMap" file="data/cena_lobes_parkes.fits">
   <parameter scale="1" name="Prefactor" min="0.001" max="1000.0" value="1" free="0" unit="arbitrary"/>
</spatialModel>
```

Alternatively, one may add a normalize attribute as shown below:

```
<spatialModel type="SpatialMap" file="data/cena_lobes_parkes.fits" normalize="0">
   <parameter scale="1" name="Prefactor" min="0.001" max="1000.0" value="1" free="0"/>
</spatialModel>
```

#7 - 02/06/2014 11:40 PM - Knödlseder Jürgen

Okumura Akira wrote:

> Dear the system admin (Jürgen?)
>
> Could you please remove co_OriB.fits from this page? I did not realize that this Redmine page was public.

We're doing open source smile.png

#8 - 02/07/2014 12:06 AM - Okumura Akira

Adding normalize and/or unit sounds reasonable. Then an XML itself will explain everything that the user is doing.

#9 - 02/07/2014 05:12 PM - Okumura Akira

http://gammalib.sourceforge.net/user_manual/modules/model.html

FYI. The very bottom of the above document made Tak and me very confused. While this explanation is not written for DiffuseSource but for FileFunction,

> cm?2s?1MeV?1sr?1 for a diffuse source

is a bit misleading for users who want to simulate diffuse sources with DiffuseSource. Adding a short note about the unit would be helpful.

#10 - 02/07/2014 06:02 PM - Knödlseder Jürgen

Okumura Akira wrote:

> http://gammalib.sourceforge.net/user_manual/modules/model.html
>
> FYI. The very bottom of the above document made Tak and me very confused. While this explanation is not written for DiffuseSource but for FileFunction,
>
> > cm?2s?1MeV?1sr?1 for a diffuse source
>
> is a bit misleading for users who want to simulate diffuse sources with DiffuseSource. Adding a short note about the unit would be helpful.

You're right. I think this statement is simply wrong and should be changed. Thank your for catching that.

#11 - 02/07/2014 06:08 PM - Knödlseder Jürgen

Well, there is one case where the units are per MeV, which is the constant diffuse model used for isotropic emission. This needs to be stated clearly somewhere when models are introduced, because this applies to all kinds of spectral models.

#12 - 02/19/2014 05:59 PM - Knödlseder Jürgen

- Project changed from ctools to GammaLib

#13 - 02/19/2014 05:59 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- Target version set to 00-08-02

- % Done changed from 0 to 80

Added normalize attribute to diffuse spatial model. Still needs some testing.

#14 - 02/19/2014 06:27 PM - Knödlseder Jürgen

- Status changed from In Progress to Feedback

- % Done changed from 80 to 100

Merge into 00-08-02.

#15 - 07/19/2014 02:04 AM - Knödlseder Jürgen

- Status changed from Feedback to Resolved

- Target version changed from 00-08-02 to 00-09-00

#16 - 07/20/2014 11:24 PM - Knödlseder Jürgen

- Status changed from Resolved to Closed

Files

model_point.xml	737 Bytes	02/06/2014	Okumura Akira
model_extended.xml	655 Bytes	02/06/2014	Okumura Akira
model_diffuse.xml	654 Bytes	02/06/2014	Okumura Akira
model_diffuse_x2.xml	654 Bytes	02/06/2014	Okumura Akira