GammaLib - Action #3253

Create uniform ring spatial model

06/18/2020 11:43 AM - Martin Pierrick

Status:	Closed	**Start date:**	06/18/2020
Priority:	Normal	**Due date:**	
Assigned To:	Martin Pierrick	**% Done:**	100%
Category:		**Estimated time:**	6.00 hours
Target version:	1.7.0		

Description

Create GModelSpatialRadialRing class, which implements a ring-shaped uniform intensity distribution. Will be inspired from GModelSpatialRadialShell. Parameters would be inner and outer radius, center position, and total flux.

History

#1 - 06/18/2020 12:09 PM - Knödlseder Jürgen

It's maybe easier to adapt the GModelSpatialRadialDisk class, since a ring is a disk with a hole.

#2 - 06/18/2020 02:27 PM - Martin Pierrick

Good point ! Will do that.

#3 - 06/18/2020 10:43 PM - Martin Pierrick

- % Done changed from 0 to 90

Almost done. Compiles and passes all tests. Need to play with it a bit to make sure everything is alright. Would need to update the description above: ring parameters are radius and width (plus direction), similar to shell model, while inner/outer radii are computed and used internally. And there is no flux parameter of course as the model is normalized.

#4 - 06/20/2020 09:01 AM - Martin Pierrick

- Status changed from New to Pull request

- % Done changed from 90 to 100

Should be OK. Available on branch 3253-add-ring-model on my gitlab fork.
Not tested extensively, just a few simulations and fits with rings of various sizes and widths, with a Crab-like spectrum.

Side note for memory (mostly for myself): in a first implementation, the model was not properly fitted because parameters radius and width were found to have zero curvature and thus fixed. This first implementation relied in many places on internal variables for inner and outer radii. Changing that to using methods radius() and width() instead seems to have solved the problem but I am not sure to have understood why...

#5 - 06/22/2020 10:00 AM - Knödlseder Jürgen

I guess that with inner and outer radii as parameters you can have the case that an inner radius becomes larger than an outer radius, which leads to a zero ring (I guess). Radius and width avoids this (like for the shell model).

#6 - 06/22/2020 10:19 AM - Knödlseder Jürgen

I see that you have defined the radius as the outer radius, which still poses a problem when the width becomes larger than the radius. I think it should be inner radius. I'll change this during integration.

#7 - 06/22/2020 10:48 AM - Martin Pierrick

Indeed. I did implement a check on the values in one of the constructors, but it is not checked later on if you pass values through radius() and width(). By the way, I noted this morning that the shell model is defined via inner radius so better be consistent with that. I can make the modifications if you want...

#8 - 06/22/2020 11:01 AM - Martin Pierrick

OK. I am on it. Should be done within 30min...

#9 - 06/22/2020 11:47 AM - Martin Pierrick

Done and available on my branch, but apparently you moved forward and made it yourself...

#10 - 06/22/2020 11:56 AM - Knödlseder Jürgen

- Target version set to 1.7.0

You were too fast, I did the changes also while integrating the code. Anywhere, I'm currently testing the code in the integration branch, code will be available soon in devel for further testing (in particular a pull distribution would be important to check that also the MC code is okay).

#11 - 06/22/2020 01:31 PM - Knödlseder Jürgen

- Status changed from Pull request to Feedback

Merged into devel. I leave the issue open to get some pull distribution results.

#12 - 07/09/2020 03:17 PM - Knödlseder Jürgen

- Status changed from Feedback to Closed

I verified the pull distributions. There were some issues with the CTA response convolution, related to the fact that intensities drop to zero below the inner radius, but after adding nodes for the inner radius to the GCTAResponseIrf::irf_radial and GCTAResponseIrf::nroi_radial methods, the fitting was okay and the pull distributions correct.

I added the pull distribution generation to the ctools science verification step. All code is merged into devel.