

GammaLib - Feature #2382

Implement parametric CTA background fitting using splines.

03/02/2018 02:07 PM - Buehler Rolf

Status:	New	Start date:	03/02/2018
Priority:	Normal	Due date:	
Assigned To:	Buehler Rolf	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
We want to implement a GCTAModelSplineBackground class which can fit general background shapes in spectrum, offset (and later also phi). Splines seem well suited to do this.			

History

#1 - 03/07/2018 02:06 PM - Buehler Rolf

Regarding the implementation of this, here are our current plans. Any comments would be appreciated, thanks.

- 1) We want to create a new class "GCTAModelSplineBackground" and plan to inherit from "GModelData" (following "GCTAModelIrfBackground" as a template). Or would there be an advantage to inherit from "GCTABackground"?
- 2) The spline background model will not be factorisable into a spatial and temporal part, this is not a problem right?
- 3) We will need an external spline creation/evaluation library. We were thinking to use einspline (<http://einspline.sourceforge.net/>), as it allows one to set boundary conditions at the edges of the splines.

#2 - 03/07/2018 03:14 PM - Knödlseeder Jürgen

user#76 wrote:

Regarding the implementation of this, here are our current plans. Any comments would be appreciated, thanks.

- 1) We want to create a new class "GCTAModelSplineBackground" and plan to inherit from "GModelData" (following "GCTAModelIrfBackground" as a template). Or would there be an advantage to inherit from "GCTABackground"?

GCTABackground is the base class for all IRF background classes, hence this is not suited. You should follow the example of GCTAModelRadialAcceptance.

- 2) The spline background model will not be factorisable into a spatial and temporal part, this is not a problem right?

Yes, no problem with that. While the GCTAModelRadialAcceptance still assumes that factorisation, the GModelData base class is more generic.

- 3) We will need an external spline creation/evaluation library. We were thinking to use einspline (<http://einspline.sourceforge.net/>), as it allows one to set boundary conditions at the edges of the splines.

For testing purposes this is fine, but to make the code available in GammaLib we should add the spline computation there. I did not use splines since a long time, but looking into Numerical Recipes it looks like it's less than a hundred lines of code that are needed to get this going.

I can certainly provide you advice on how to organise the code.