GammaLib - Feature #2148

GSkyDir does not cache ra,dec or l,b values when they are computed

07/04/2017 11:54 AM - Cardenzana Josh

Status:	Closed	Start date:	07/04/2017	
Priority:	Normal	Due date:		
Assigned To:	Cardenzana Josh	% Done:	100%	
Category:		Estimated time:	0.00 hour	
Target version:	1.3.1			
Description				
When a GSkyDir object is created, it must be passed either ra,dec or glon, glat (i.e. l,b) values. As a result, the other coordinates are not cached. Here is an example:				
GSkyDir obs, dir; obs.radec(0, 0); dir.lb(0, 0); double dist = obs.dist(d	ir);			
Here, obs is created with celestial coordinates (as all observations do) and dir is created with galactic coordinates. When the 'dist()' is computed, dir has to have its coordinates converted to celestial before the cosine of the distance is computed. Here is the code snippet that does the computation:				
<pre>if (dir.m_has_rade4 #if defined(G_S if (!dir.m_has_radi7.m_has_radi7.m_sin_dec dir.m_cos_de } cosdis = m_sin_ m_cos_de std::cos(di #else cosdis = std::sin std::cos(di #endif } else { #if defined(G_S cosdis = m_sin_ m_cos_de std::cos(di #endif } }</pre>	c) { INCOS_CACHE) idec_cache = true; c = std::sin(dir.m_dec); c = std::cos(dir.m_dec); dec * dir.m_sin_dec + // This section show c * dir.m_cos_dec * // second time this fur r.m_ra - m_ra); (m_dec) * std::sin(dir.m_dec) + _dec) * std::cos(dir.m_dec) * r.m_ra - m_ra); INCOS_CACHE) dec * sin(dir.dec()) + // This section is run c * cos(dir.dec()) * // should only be run th r.ra() - m_ra); dec) * sin(dir.dec()) + c * cos(dir.dec()) *	EVERY time, but		
For each call to 'GSkyDir::dec()' and 'GSkyDir::ra()' the method has to run 'GSkyDir::gal2equ()' and take the cosine of the declination, both of which are expensive computations. In the above, that's three calls to 'gal2equ()' and two calls to 'cos()'. For map cube computations, which loop over every bin for every observation, this becomes very expensive.				
This is actually unnecessary as the data members 'm_ra' and 'm_dec' are filled when calling 'dir.ra()' or 'dir.dec()' the first time (similarly for galactic coordinates), so all that needs to be done is to update the call methods to set either 'm_has_radec' or 'm_has_lb' to true. This addition should be very low overhead, but potentially dramatic speedups in computation time when the bin sky directions are cached.				

Related issues:

Related to ctools - Feature # 2147: Reduce computation time of ctmodel and pa...

Closed 07/04/2017

History

#1 - 07/04/2017 03:30 PM - Cardenzana Josh

- Related to Feature #2147: Reduce computation time of ctmodel and parallelize using OpenMP. added

#2 - 07/04/2017 03:47 PM - Cardenzana Josh

- Status changed from In Progress to Pull request
- % Done changed from 0 to 100

After implementing the above change, the following time difference is produced in ctmodel running the following command:

ctmodel debug=yes inmodel=../1dc.south/models/model_bkg.xml inobs=../1dc.south/obs/obs_egal_baseline.xml incube=egalsrvy_01_datamap.fits outcube=test_ctmodel_2148.fits

This runs over the extragalactic survey data from the 1dc using a pre-generated cube (from ctbin) to define the bin information. The model cube is generated from only the background "CTAIrfBackground" model.

devel:

...

2017-07-04T13:31:07: +=========+
2017-07-04T13:31:07: Model cube
2017-07-04T13:31:07: +========+
2017-07-04T13:31:07: === GCTAEventCube ===
2017-07-04T13:31:07: Number of events: 21746400
2017-07-04T13:31:07: Number of elements: 67200000
2017-07-04T13:31:07: Number of pixels: 3360000
2017-07-04T13:31:07: Number of energy bins: 20
2017-07-04T13:31:07: Time interval 59215.5 - 59256.8548611111 days
2017-07-04T13:31:07: === GEbounds ===
2017-07-04T13:31:07: Number of intervals: 20
2017-07-04T13:31:07: Energy range 100 GeV - 100 TeV
2017-07-04T13:31:07: === GSkyMap ===
2017-07-04T13:31:07: Number of pixels 3360000
2017-07-04T13:31:07: X axis dimension 2800
2017-07-04T13:31:07: Y axis dimension: 1200
2017-07-04T13:31:07: Number of maps 20
2017-07-04T13:31:07: Shape of maps: (20)

2017-07-04T13:31:14: Application "ctmodel" terminated after 17168 wall clock seconds, consuming 16966.7 seconds of CPU time.

2148 branch:

...

2017-07-04T12:06:59: === GCTAEventCube ===
2017-07-04T12:06:59: Number of events 21746400
2017-07-04T12:06:59: Number of elements: 67200000
2017-07-04T12:06:59: Number of pixels 3360000
2017-07-04T12:06:59: Number of energy bins: 20
2017-07-04T12:06:59: Time interval 59215.5 - 59256.8548611111 days
2017-07-04T12:06:59: === GEbounds ===
2017-07-04T12:06:59: Number of intervals: 20
2017-07-04T12:06:59: Energy range 100 GeV - 100 TeV
2017-07-04T12:06:59: === GSkyMap ===
2017-07-04T12:06:59: Number of pixels 3360000
2017-07-04T12:06:59: X axis dimension 2800
2017-07-04T12:06:59: Y axis dimension 1200
2017-07-04T12:06:59: Number of maps 20
2017-07-04T12:06:59: Shape of maps: (20)

2017-07-04T12:07:07: Application "ctmodel" terminated after 6332 wall clock seconds, consuming 6284.84 seconds

This represents about 2.5x speed up with no difference in the produced model cubes. The biggest time computation in ctmodel in the above run is still in 'GSkyDist::dist()' at the end when computing 'gammalib::acos()' (which uses 'std::asin()'), followed closely by 'std::cos()'. However, I'm not sure there's any way to abstract this out without reducing precision in the computation.

Pull branch:

josh cardenzana / gammalib: 2148-improve_gskydir_caching

#3 - 07/06/2017 01:01 PM - Knödlseder Jürgen

- Target version changed from 1.4.0 to 1.3.1

I'm about to merge in the code. I moved the setting of the flags in the GSkyDir::gal2equ() and GSkyDir::equ2gal() methods since this simplifies the code and then also covers more methods.

#4 - 07/06/2017 01:56 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Merged into bugfix-1.3.1 branch. Will also merge into devel branch.