GammaLib - Bug #1761

GModelSpatialDiffuseCube maps() and pixels() methods return 0

04/13/2016 03:00 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	04/13/2016
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.7.0		

Description

When constructing a GModelSpatialDiffuseCube object from a map cube file, the maps() and pixels() methods return 0 since the cube is only physically loaded on demand when the pixel values are accessed.

The map cube metadata such as number of pixels and number of maps should be loaded even if the cube itself is not loaded.

History

#1 - 10/06/2016 05:01 PM - Knödlseder Jürgen

- Target version set to 1.2.0

#2 - 03/03/2017 10:14 AM - Knödlseder Jürgen

- Target version changed from 1.2.0 to 1.3.0

#3 - 06/07/2017 12:37 AM - Knödlseder Jürgen

- Target version changed from 1.3.0 to 1.4.0

#4 - 07/31/2017 11:07 PM - Knödlseder Jürgen

- Target version deleted (1.4.0)

#5 - 06/24/2020 10:28 PM - Knödlseder Jürgen

- Assigned To set to Knödlseder Jürgen

- Target version set to 1.7.0

#6 - 06/26/2020 03:43 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- % Done changed from 0 to 10

The following test script

```
#!/usr/bin/env python
import gammalib

# Save map
map = gammalib.GSkyMap('CAR','GAL',0.0,0.0,1.0,1.0,360,180)
map.save('map.fits', True)

# Create diffuse map
diffuse_map = gammalib.GModelSpatialDiffuseMap('map.fits')
print('=== Diffuse map ===')
print(diffuse_map.value())

# Create diffuse cube
diffuse_cube = gammalib.GModelSpatialDiffuseCube('map.fits')
print('=== Diffuse cube ===')
print(diffuse_cube.value())
print(diffuse_cube.maps())
```

```
print(diffuse_cube.pixels())
```

gives the following output

```
=== Diffuse map ===
1.0
=== Diffuse cube ===
1.0
0
0
```

#7 - 06/26/2020 04:52 PM - Knödlseder Jürgen

- Status changed from In Progress to Pull request

- % Done changed from 10 to 100

I decided to remove the GModelSpatialDiffuseCube::maps() and GModelSpatialDiffuseCube::pixels() methods since the same information can be accessed through the GModelSpatialDiffuseCube::cube().nmaps() and GModelSpatialDiffuseCube::cube().npix() methods.

I added a fetch_cube() call in the GModelSpatialDiffuseCube::cube() method so that the cube actually gets fetched before any access. This fixes also the GModelSpatialDiffuseCube::cube() method.

Two ctools unit tests needed to be adapted since they were using the GModelSpatialDiffuseCube::maps() and GModelSpatialDiffuseCube::pixels() methods.

#8 - 06/30/2020 02:19 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Merged into devel.