

GammaLib - Support #2808

Problem in Swig interface of GModel scale method?

01/30/2019 10:16 AM - Tibaldo Luigi

| | | | |
|------------------------|-------------------|------------------------|------------|
| Status: | In Progress | Start date: | 01/30/2019 |
| Priority: | Normal | Due date: | |
| Assigned To: | Knödlseher Jürgen | % Done: | 10% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | | | |

Description

The scale method of the GModel class has two different implementations in the Swig interface

```
GModelPar    scale(const std::string& instrument) const;
void         scale(const GModelPar& par);
```

I'd have expected that both enable the user to set the value of the scale parameter, but I did not manage to use the second one to do so successfully. Here's a minimal example.

```
import gammalib

# position
ra = 83.633
dec = 22.015
# log parabola pars
prefactor = 3.23e-17 # ph cm-2 s-1 MeV-1
index = -2.47
pivot_e = 1 # TeV
curvature = -0.25
srcname = 'Crab'

src_dir = gammalib.GSkyDir()
src_dir.radec_deg(ra, dec)
spatial = gammalib.GModelSpatialPointSource(src_dir)
pivot = gammalib.GEnergy(pivot_e, 'TeV')
spectral = gammalib.GModelSpectralLogParabola(prefactor, index, pivot, curvature)
model = gammalib.GModelSky(spatial, spectral)
model.name(srcname)

print(model)
=== GModelSky ===
Name .....: Crab
Instruments .....: all
Instrument scale factors ...: unity
Observation identifiers ...: all
Model type .....: PointSource
Model components .....: "PointSource" * "LogParabola" * "Constant"
Number of parameters .....: 7
Number of spatial par's ....: 2
RA .....: 83.633 deg (fixed,scale=1)
DEC .....: 22.015 deg (fixed,scale=1)
Number of spectral par's ...: 4
Prefactor .....: 3.23e-17 +/- 0 [0,infy] ph/cm2/s/MeV (free,scale=3.23e-17,gradient)
Index .....: -2.47 +/- 0 [10,-10] (free,scale=-2.47,gradient)
Curvature .....: -0.25 +/- 0 [10,-10] (free,scale=-0.25,gradient)
PivotEnergy .....: 1000000 MeV (fixed,scale=1000000,gradient)
Number of temporal par's ...: 1
Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
```

All the instrument scales are set to unity as expected. Trying to change the scales using the second version of the method.

```
In [5]: model.scale('HESS').value(0.5)
```

```
In [6]: model.has_scales()
```

```
Out[6]: False
```

```
In [7]: model.scale("HESS").value()
```

```
Out[7]: 1.0
```

```
In [8]: print(model)
```

```
==== GModelSky ====
```

```
Name .....: Crab
```

```
Instruments .....: all
```

```
Instrument scale factors ...: unity
```

```
Observation identifiers ...: all
```

```
Model type .....: PointSource
```

```
Model components .....: "PointSource" * "LogParabola" * "Constant"
```

```
Number of parameters .....: 7
```

```
Number of spatial par's ...: 2
```

```
RA .....: 83.633 deg (fixed,scale=1)
```

```
DEC .....: 22.015 deg (fixed,scale=1)
```

```
Number of spectral par's ...: 4
```

```
Prefactor .....: 3.23e-17 +/- 0 [0,infty] ph/cm2/s/MeV (free,scale=3.23e-17,gradient)
```

```
Index .....: -2.47 +/- 0 [10,-10] (free,scale=-2.47,gradient)
```

```
Curvature .....: -0.25 +/- 0 [10,-10] (free,scale=-0.25,gradient)
```

```
PivotEnergy .....: 1000000 MeV (fixed,scale=1000000,gradient)
```

```
Number of temporal par's ...: 1
```

```
Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
```

So nothing happens. Now trying the first method.

```
In [9]: scale = gammalib.GModelPar('HESS', 0.5)
```

```
In [10]: model.scale(scale)
```

```
In [11]: model.has_scales()
```

```
Out[11]: True
```

```
In [12]: model.scale("HESS").value()
```

```
Out[12]: 0.5
```

```
In [13]: print(model)
```

```
==== GModelSky ====
```

```
Name .....: Crab
```

```
Instruments .....: all
```

```
Instrument scale factors ...: HESS=0.5, others unity
```

```
Observation identifiers ...: all
```

```
Model type .....: PointSource
```

```
Model components .....: "PointSource" * "LogParabola" * "Constant"
```

```
Number of parameters .....: 7
```

```
Number of spatial par's ...: 2
```

```
RA .....: 83.633 deg (fixed,scale=1)
```

```
DEC .....: 22.015 deg (fixed,scale=1)
```

```
Number of spectral par's ...: 4
```

```
Prefactor .....: 3.23e-17 +/- 0 [0,infty] ph/cm2/s/MeV (free,scale=3.23e-17,gradient)
```

```
Index .....: -2.47 +/- 0 [10,-10] (free,scale=-2.47,gradient)
```

```
Curvature .....: -0.25 +/- 0 [10,-10] (free,scale=-0.25,gradient)
```

```
PivotEnergy .....: 1000000 MeV (fixed,scale=1000000,gradient)
```

```
Number of temporal par's ...: 1
```

```
Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
```

This works as expected.

I'm not sure if there is a misunderstanding from my side on the usage of this method or something that really doesn't work as it should.

History

#1 - 03/06/2019 12:13 PM - Knödseder Jürgen

- *Tracker changed from Bug to Support*

The method

```
GModelPar scale(const std::string& instrument) const
```

returns a parameter but does not allow to set it (the method is declared const). Since the const declarations are however meaningless in Python, you can write a code as you did that attempts setting the scale. This does however not set the scale in the model, but in the local copy of the parameter:

```
>>> p=model.scale('HESS')
>>> print(p)
HESS .....: 1 (fixed,scale=1)
>>> p.value(0.5)
>>> print(p)
HESS .....: 0.5 (fixed,scale=1)
```

#2 - 03/06/2019 12:14 PM - Knödseder Jürgen

- *Status changed from New to In Progress*

- *% Done changed from 0 to 10*