

GammaLib - Bug #3072

Min-Max logic is inverted in GOptimizerPar for negative scale factors

11/26/2019 12:22 PM - Knödseder Jürgen

Status:	Closed	Start date:	11/26/2019
Priority:	Normal	Due date:	
Assigned To:	Knödseder Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.7.0		
Description			
The logic of minimum and maximum is inverted in GOptimizerPar for negative scale factors. The reason behind this is that the minimum and maximum is actually applied on the factor value, hence a minimum becomes a maximum and vice versa for a negative scale factor.			
This should be corrected since it is confusing.			

History

#1 - 01/28/2020 10:07 AM - Knödseder Jürgen

- Target version set to 1.7.0

#2 - 06/25/2020 12:36 PM - Knödseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödseder Jürgen

The following code

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

par = gammadlib.GModelPar("Test", 3.0, -1.0)
par.range(-10.0, 10.0)
```

leads to the following error

Traceback (most recent call last):

```
File "./test.py", line 5, in <module>
    par.range(-10.0, 10.0)
File "/usr/local/gamma/lib/python2.7/site-packages/gammadlib/opt.py", line 500, in range
    return _opt.GOptimizerPar_range(self, min, max)
```

ValueError: *** ERROR in GOptimizerPar::factor_value(double&): Invalid argument. Specified minimum factor 10 is larger than the value factor 3.

#3 - 06/26/2020 02:36 PM - Knödseder Jürgen

I implemented the correct handling of the minima and maxima.

The boundaries are now stored as values, and for the `factor_min()` and `factor_max()` now return the corresponding boundary based on the sign of the scale factor. The above test code now works, and adding a print function gives:

```
Test .....: -3 +/- 0 [-10,10] (free,scale=-1)
```

as expected.

All GammaLib tests are okay. I still need to check ctools now.

#4 - 06/26/2020 02:51 PM - Knödseder Jürgen

- Status changed from *In Progress* to *Pull request*

- % Done changed from 0 to 100

Also the ctools unit check is ok.

#5 - 06/26/2020 03:36 PM - Knödseder Jürgen

- Status changed from *Pull request* to *Closed*

Merged into devel.

#6 - 06/29/2020 09:24 AM - Knödseder Jürgen

- Status changed from *Closed* to *In Progress*

- % Done changed from 100 to 90

There is an integration error with the ctools unit test on OpenSolaris. 9 ctools unit tests fail. Here some examples of the error messages that occur:

ctools unit testing.Test cterror on command line

```
*** ERROR in GPythonTestSuite::test: <type 'exceptions.RuntimeError'> *** ERROR in GXml::load(std::string&): Unable to open file "cterror_cmd1.xml"
```

ctools unit testing.Test cterror from Python

```
*** ERROR in GPythonTestSuite::test: <type 'exceptions.ValueError'> *** ERROR in GOptimizerPar::factor_value(double&): Invalid argument. Specified value factor 1e-07 is smaller than the minimum boundary 1e-07.
```

cscripts unit testing.Test csbkgmodel from Python

```
*** ERROR in GPythonTestSuite::test: <type 'exceptions.ValueError'> *** ERROR in GOptimizerPar::factor_value(double&): Invalid argument. Specified value factor 1e-10 is smaller than the minimum boundary 1e-10.
```

cscripts unit testing.Test csbkgmodel pickeling

```
*** ERROR in GPythonTestSuite::test: <type 'exceptions.ValueError'> *** ERROR in GOptimizerPar::factor_value(double&): Invalid argument. Specified value factor 1e-10 is smaller than the minimum boundary 1e-10.
```

cscripts unit testing.Test csphasecrv from Python

```
*** ERROR in GPythonTestSuite::test: <type 'exceptions.ValueError'> *** ERROR in GOptimizerPar::factor_value(double&): Invalid argument. Specified value factor 1e-07 is smaller than the minimum boundary 1e-07.
```

#7 - 06/29/2020 05:55 PM - Knödseder Jürgen

I reimplemented the parameter boundary code, storing now the factor boundaries so that no accuracy problems should occur.

#8 - 06/30/2020 02:03 PM - Knödseder Jürgen

- *Status changed from In Progress to Closed*

- *% Done changed from 90 to 100*

The new code works on OpenSolaris. Merged into devel.