GammaLib - Bug #3403

ctools unit test fails after implementation of analytical gradients in energy dispersion

10/16/2020 12:10 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	10/16/2020
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2.0.0		

Description

Here is what is happening

```
**********************
* ctools unit testing *
**********************
Test ctobssim on command line: ...... ok
Test ctobssim from Python: ........................................ ok
Test ctselect on command line: ........ ok
Test ctselect from Python: ............................................................ ok
Test ctphase on command line: ...... ok
Test ctphase from Python: ............. ok
Test ctfindvar on command line: ...... ok
Test ctfindvar from Python: ......... ok
Test ctbin on command line: ................. ok
Test ctbin from Python: ............................................................ ok
Test ctlike on command line: ............ ok
Test ctlike from Python: ........................................... ok
Test cttsmap on command line: ....................... ok
Test cttsmap from Python: ............................... ok
Test ctmodel on command line: ....... ok
Test ctmodel from Python: ............./test_python_ctools.sh: line 25: 73318 Segmentation fault: 11  ./test_python_ctools.py
FAIL test_python_ctools.sh (exit status: 139)
```

with the exception occurring in

```
Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0   libgamma.8.dylib                0x000000010893ff83 GResponse::convolve(GModelSky const&, GEvent const&, GObservation
 const&, bool const&) const + 531 (GResponse.cpp:242)
1   libgamma.8.dylib                0x00000001089500e8 GModels::eval(GEvent const&, GObservation const&, bool const&) const +
 168
2   libctools.7.dylib               0x000000010a5eaaf2 ctmodel::fill_cube(GCTAObservation const*, GModels&) + 1010
3   libctools.7.dylib               0x000000010a5e95c9 ctmodel::run() + 505 (vector:657)
4   libctools.7.dylib               0x000000010a5ce0d6 ctool::execute() + 22 (ctool.cpp:267)
```

which corresponds to the following code

```
double GResponse::convolve(const GModelSky&    model,
                           const GEvent&       event,
                           const GObservation& obs,
                           const bool&         grad) const
{
...
        } // endfor: looped over intervals

        // Initialise array index
        int index = 0;
```

```
            // Get probability
            prob = array(index++);  // <= Segfault occuring here

            // Set gradients
            if (grad) {
...
```

History

#1 - 10/16/2020 12:25 PM - Knödlseder Jürgen

- Status changed from New to Pull request

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 90

This was an easy one. The code allows the special case that the array object is empty, and adding a

```
            // Get probability
            if (array.size() > 0) {
                prob = array(index++);
            }
```

fixed the problem.

#2 - 10/16/2020 12:45 PM - Knödlseder Jürgen

- Status changed from Pull request to In Progress

- % Done changed from 90 to 50

Another issue occurs in the csscs script for the unit test with energy dispersion (the one that generates the log file csscs_py5.log):

```
Test csscs from Python: ..........................................................
 *** Break *** segmentation violation
 Generating stack trace...

 *** Break *** segmentation violation
 Generating stack trace...

 *** Break *** segmentation violation
 Generating stack trace...

 *** Break *** segmentation violation
 Generating stack trace...
 0x0000000106d52603 in GObservation::likelihood_poisson_binned(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (GObservation.cpp:0)
 0x0000000106d52603 in GObservation::likelihood_poisson_binned(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (GObservation.cpp:0)
 0x0000000106d52603 in GObservation::likelihood_poisson_binned(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (GObservation.cpp:0)
 0x0000000106d52603 in GObservation::likelihood_poisson_binned(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (GObservation.cpp:0)
 0x0000000106d506fa in GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (string:1421)
 0x0000000106d506fa in GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (string:1421)
 0x0000000106d506fa in GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (string:1421)
 0x0000000106d506fa in GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const (in libgamma.8.dylib) (string:1421)
```

```
 0x0000000106d4e42f in GObservations::likelihood::eval(GOptimizerPars const&) (in libgamma.8.dylib) (GObservations_likelihood.cpp:0)
 0x0000000106d4e42f in GObservations::likelihood::eval(GOptimizerPars const&) (in libgamma.8.dylib) (GObservations_likelihood.cpp:0)
 0x0000000106d4e42f in GObservations::likelihood::eval(GOptimizerPars const&) (in libgamma.8.dylib) (GObservations_likelihood.cpp:0)
 0x0000000106d4e42f in GObservations::likelihood::eval(GOptimizerPars const&) (in libgamma.8.dylib) (GObservations_likelihood.cpp:0)
 0x0000000106d2d0cc in GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) (in libgamma.8.dylib) (GOptimizerLM.cpp:0)
 0x0000000106d2d0cc in GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) (in libgamma.8.dylib) (GOptimizerLM.cpp:0)
 0x0000000106d2d0cc in GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) (in libgamma.8.dylib) (GOptimizerLM.cpp:0)
 0x0000000106d2d0cc in GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) (in libgamma.8.dylib) (GOptimizerLM.cpp:0)
 0x0000000106d4c503 in GObservations::optimize(GOptimizer&) (in libgamma.8.dylib) (GObservations.cpp:0)
 0x0000000106d4c503 in GObservations::optimize(GOptimizer&) (in libgamma.8.dylib) (GObservations.cpp:0)
 0x0000000106d4c503 in GObservations::optimize(GOptimizer&) (in libgamma.8.dylib) (GObservations.cpp:0)
 0x0000000106d4c503 in GObservations::optimize(GOptimizer&) (in libgamma.8.dylib) (GObservations.cpp:0)
 0x00000001088a903f in ctlike::optimize_lm() (in libctools.7.dylib) (ctlike.cpp:483)
 0x00000001088a903f in ctlike::optimize_lm() (in libctools.7.dylib) (ctlike.cpp:483)
 0x00000001088a903f in ctlike::optimize_lm() (in libctools.7.dylib) (ctlike.cpp:483)
 0x00000001088a903f in ctlike::optimize_lm() (in libctools.7.dylib) (ctlike.cpp:483)
 0x00000001088a7c56 in ctlike::run() (in libctools.7.dylib) (GObservations.hpp:359)
 0x00000001088a7c56 in ctlike::run() (in libctools.7.dylib) (GObservations.hpp:359)
 0x00000001088a7c56 in ctlike::run() (in libctools.7.dylib) (GObservations.hpp:359)
 0x00000001088a7c56 in ctlike::run() (in libctools.7.dylib) (GObservations.hpp:359)
 0x00000001087ee8f8 in _wrap_ctlike_run(_object*, _object*) (in _tools.so) (tools_wrap.cpp:1447)
 0x00000001087ee8f8 in _wrap_ctlike_run(_object*, _object*) (in _tools.so) (tools_wrap.cpp:1447)
 0x00000001087ee8f8 in _wrap_ctlike_run(_object*, _object*) (in _tools.so) (tools_wrap.cpp:1447)
 0x00000001087ee8f8 in _wrap_ctlike_run(_object*, _object*) (in _tools.so) (tools_wrap.cpp:1447)
Failed to start process notifications for pid 13131 (19)
Failed to start process notifications for pid 13129 (19)
Failed to start process notifications for pid 13128 (19)
Failed to start process notifications for pid 13130 (19)
```

#3 - 10/16/2020 12:52 PM - Knödlseder Jürgen

A Python error with the following backtrace occurred:

```
Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0   libgamma.8.dylib                  0x0000000106c55073 GResponse::convolve(GModelSky const&, GEvent const&, GObservation const&, bool const&)
 const + 803 (GOptimizerPar.hpp:402)
1   libgamma.8.dylib                  0x0000000106c46c88 GObservation::model(GModels const&, GEvent const&, GVector*) const + 248
2   libgamma.8.dylib                  0x0000000106c48603 GObservation::likelihood_poisson_binned(GModels const&, GVector*, GMatrixSparse*,
 double*) const + 275
3   libgamma.8.dylib                  0x0000000106c466fa GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const + 250
 (string:1421)
```

```
4   libgamma.8.dylib          0x0000000106c4442f GObservations::likelihood::eval(GOptimizerPars const&) + 735
5   libgamma.8.dylib          0x0000000106c230cc GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) + 1084
6   libgamma.8.dylib          0x0000000106c42503 GObservations::optimize(GOptimizer&) + 51
7   libctools.7.dylib       0x000000010879f03f ctlike::optimize_lm() + 831 (ctlike.cpp:483)
8   libctools.7.dylib       0x000000010879dc56 ctlike::run() + 214 (GObservations.hpp:359)
```

The code where the exception occurs is here:

```
inline
void GOptimizerPar::factor_gradient(const double& gradient) const
{
   m_factor_gradient = gradient;  // <= Exception occurs here
   return;
}
```

#4 - 10/16/2020 03:05 PM - Knödlseder Jürgen

- Status changed from In Progress to Pull request

- % Done changed from 50 to 90

I solved the problems by precomputing the size of the array:

```
        // Determine size of result array
        int size = 1;
        if (grad) {
           if (model.spectral() != NULL) {
              for (int i = 0; i < model.spectral()->size(); ++i) {
                 GModelPar& par = (*(model.spectral()))[i];
                 if (par.is_free() && par.has_grad()) {
                    size++;
                 }
              }
           }
           if (model.temporal() != NULL) {
              for (int i = 0; i < model.temporal()->size(); ++i) {
                 GModelPar& par = (*(model.temporal()))[i];
                 if (par.is_free() && par.has_grad()) {
                    size++;
                 }
              }
           }
           if (model.has_scales()) {
              for (int i = 0; i < model.scales(); ++i) {
                 GModelPar& par = const_cast<GModelPar&>(model.scale(i));
                 if (par.name() == obs.instrument()) {
                    if (par.is_free() && par.has_grad()) {
                       size++;
                    }
                 }
```

```
            }
        }
    }

    // Initialise Ndarray array
    GNdarray array(size);
```

#5 - 10/17/2020 03:33 PM - Knödlseder Jürgen

- Project changed from ctools to GammaLib

- Status changed from Pull request to Closed

- Target version changed from 2.0.0 to 2.0.0

- % Done changed from 90 to 100