GammaLib - Feature #1768

Investigate whether we can interface GammaLib with NumPy

04/18/2016 12:03 PM - Knödlseder Jürgen

Status:	In Progress	**Start date:**	04/18/2016
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	50%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Following discussions with Christoph, it would be useful to interface GammaLib with NumPy. NumPy is pretty standard in the Python community to handle ndarrays, and it could be useful for users that want to dive deeply into the internal working of the library to access data in that format.

However, since NumPy is not part of the Python standard library we want (at least for the moment) to keep the NumPy interface as an optional dependency. We have to see then how this evolves over time.

So this feature is about investigating how the interfacing with NumPy can be done properly / best.

I know that there is a NumPy swig interface, and maybe this can be simply added to the actual code using the appropriate swig typemaps / typechecks. It needs to be investigate how this can be done optionally (probably needs a NumPy detection step during configuration).

There is also the possibility (which I guess Christoph won't like in a first place :-)) to implement ndarrays in GammaLib, and to interface these with NumPy. This would be the GammaLibonic way. I know that some LSST guy wrote a ndarray C++ package that is compatible with NumPy, maybe this can be adapted? See: https://github.com/lsst/ndarray and https://github.com/ndarray/ndarray (only the first is GPL3, the other is BSD license). The advantage of this approach is that we gradually could use the GammaLib ndarray to store stuff (maps, data, etc.), which then would make the information immediately available in the right format to NumPy. In that way we could both support NumPy and keep the formal independency to it.

Related issues:

Related to GammaLib - Action # 1731: Use low-level FFT classes to implement G...	**Closed**	**03/03/2016**

History

#1 - 10/07/2016 10:17 PM - Knödlseder Jürgen

- Related to Action #1731: Use low-level FFT classes to implement GSkyMap smoothing added

#2 - 10/07/2016 10:56 PM - Knödlseder Jürgen

Here is a possible layout of the class:

```
class GNdarray : public GBase {
   friend GNdarray sin(const GNdarray& array);
   friend GNdarray cos(const GNdarray& array);
   friend GNdarray tan(const GNdarray& array);
   ...
public:
   GNdarray(void);
   GNdarray(const int& nx);
   GNdarray(const int& nx, const int& ny);
   GNdarray(const int& nx, const int& ny, const int& nz);
   GNdarray(const std::vector<int>& n);
   ...
   double& operator()(const int& ix);
   double& operator()(const int& ix, const int& iy);
   double& operator()(const int& ix, const int& iy, const int& iz);
   double& operator()(const std::vector<int>& i);
   ...
   void               clear(void);
   GNdarray*          clone(void) const;
   std::string        classname(void) const;
   int                size(void) const;
```

```
    double&                 at(const int& ix);
    double&                 at(const int& ix, const int& iy);
    double&                 at(const int& ix, const int& iy, const int& iz);
    double&                 at(const std::vector<int>& i);
    const std::vector<int>& shape(void) const;
    void                    shape(const std::vector<int>& shape);
    ...
protected:
    std::vector<int>    m_shape;
    std::vector<int>    m_strides;
    std::vector<double> m_data;
};
```

#3 - 10/08/2016 03:08 AM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- Target version set to 1.2.0

- % Done changed from 0 to 50

An initial version of this class has been implemented.

Next step is to see whether the class is useable by the FFT class.

#4 - 03/03/2017 10:17 AM - Knödlseder Jürgen

- Target version deleted (1.2.0)