

GammaLib - Bug #3012

Inconsistent handling of upper edge boundary in GSkyMap

09/30/2019 05:04 PM - Cardenzana Josh

Status:	Closed	Start date:	09/30/2019
Priority:	Normal	Due date:	
Assigned To:	Knödseder Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.7.0		

Description

Currently, GSkyMap does not consistently handle the upper edge boundary. Here's a quick example script demonstrating the issue. It creates a 10x10 GSkyMap and tries to access the value at the top-center border of the map:

```
#include <GammaLib.hpp>
#include <iostream>

int main()
{
    // Create a test skymap
    GSkyMap skymap("CAR","CEL",0, 0, -0.5, 0.5, 10, 10);
    skymap = 3.14159;

    // Create a test pixel on the top-center border of the map
    GSkyPixel pix(5.5, 9.5);

    std::cout << "Skymap Info:\n" << skymap.print() << std::endl;
    std::cout << "Pixel Info:\n" << pix.print() << std::endl;

    std::cout << "\nPixel is contained: ";
    if (skymap.contains(pix)) {
        std::cout << "yes\n\n";
    } else {
        std::cout << "no\n\n";
    }

    // Access the map value by pixel
    std::cout << "Trying to access map value by pixel...\n";
    try {
        double val(skymap(pix, 0));
        std::printf(" succeeded (val = %f)\n", val);
    } catch (...) {
        std::cout << " failed\n";
    }

    // Access the pixel by index
    int indx = skymap.pix2inx(pix);
    std::printf("Trying to access map value by index (index = %d)...\n", indx);
    try {
        double val(skymap(indx, 0));
        std::printf(" succeeded (val = %f)\n", val);
    } catch (...) {
        std::cout << " failed\n";
    }

    return 0;
}
```

which yields the following output:

Pixel is contained: yes

Trying to access map value by pixel...

succeeded (val = 0.000000)

Trying to access map value by index (index = 106)...

failed

It appears that the border counts as being contained in the map, however accessing directly via the pixel does not return a value from in the map and accessing by index actually throws an exception. Here's a breakdown of my understanding of what's going on:

- **skymap.contains(GSkyPixel&)** tests that x-pixel value is within [-0.5, m_num_x-0.5] with the upper bound inclusive (same for the y-pixel value), so pixel values that fall on any of the borders are considered as being inside the map.
- **skymap(int&)** simply returns the pixel value at indx, or throws an exception if indx is out of bounds.
- **skymap(GSkyPixel&)** first checks that the pixel is contained (which in the above example it is) then converts the pixel into an index and returns the pixel value at that index. However, in the above example the index is out-of-bounds, the pixel queried is beyond the end of the array and the returned value will be unpredictable.

The simplest change would be to make the upper bound on the skymap in x & y non-inclusive, which would prevent 'contains' from returning true resulting in both issues involving the contains() and skymap(GSkyPixel&) methods to be resolved. Alternatively, a check could be done for the special case when the pixel falls on the upper boundary and be handled accordingly by pix2inx.

History

#1 - 11/08/2019 05:25 PM - Knödlseider Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseider Jürgen

- Target version set to 1.7.0

- % Done changed from 0 to 10

I compiled the test code using

```
$ g++ test.cpp -lgamma -I/usr/local/gamma/include/gammlib -L/usr/local/gamma/lib
```

and got the following output

Skymap Info:

=== GSkyMap ===

Number of pixels: 100

X axis dimension: 10

Y axis dimension: 10

Number of maps: 1

Shape of maps: (1)

=== GWcsCAR ===

Number of axes: 2

Coordinate system: EQU

Projection code: CAR

Projection name: plate carree

Reference coordinate: (0, 0)

Reference pixel: (5.5, 5.5)

Increment at reference: (-0.5, 0.5)

(Phi_0, Theta_0): (0, 0) deg

(Phi_p, Theta_p): (0, 90) deg

Pixel Info:

(5.5,9.5)

Pixel is contained: yes

Trying to access map value by pixel...

succeeded (val =

Trying to access map value by index (index = 106)...

failed

It looks like the reading of a value beyond the boundary gives in my case (i.e. on Mac OS X) a huge number. Setting

```
GSkyPixel pix(5.5, 9.4);
```

hence a pixel value within the map gives

Pixel Info:
(5.5,9.4)

Pixel is contained: yes

```
Trying to access map value by pixel...
  succeeded (val = 3.141590)
Trying to access map value by index (index = 96)...
  succeeded (val = 3.141590)
```

It therefore looks reasonable to me to refuse the GSkyPixel pix(5.5, 9.5) as being invalid, which is the only way to guarantee a correct content.

I'll change the code.

#2 - 11/08/2019 09:26 PM - Knödlseider Jürgen

After excluding the upper bin boundary in GSkyMap::contains() the code returns the following:

Pixel Info:
(5.5,9.5)

Pixel is contained: no

```
Trying to access map value by pixel...
  failed
Trying to access map value by index (index = 106)...
  failed
```

and

Pixel Info:
(5.5,9.499999999999999)

Pixel is contained: yes

Trying to access map value by pixel...
succeeded (val = 3.141590)
Trying to access map value by index (index = 96)...
succeeded (val = 3.141590)

#3 - 11/08/2019 10:06 PM - Knödlseider Jürgen

- % Done changed from 10 to 80

I also needed to adjust the `GSkyMap::flux()` method since this method evaluated intensities at the upper pixel boundaries. This was in fact the reason why I changed in the past the code to include upper map boundaries in the test done in `GSkyMap::contains()`. Hence the code was actually reverted to what existed in the past, and the `GSkyMap::flux()` method now evaluates intensities just below the pixel boundaries.

#4 - 11/08/2019 11:42 PM - Knödlseider Jürgen

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

- % Done changed from 80 to 100

Merged into devel.