

## GammaLib - Bug #1876

### GCTAPsf2D::containment\_radius calculation

11/04/2016 02:35 PM - Ziegler Alexander

<b>Status:</b>	Closed	<b>Start date:</b>	11/04/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Knödlseeder Jürgen	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.3.0		

**Description**

I just stumbled over the member function GCTAPsf2D::containment\_radius() and recognized two strange things:

1.) The function GCTAPsf2D::containment\_radius is described to be calculated with a global normalization, see doxygen docu code lines 559-562 in GCTAPsf2D.cpp (e.g. here: [http://cta.irap.omp.eu/gammalib/doxygen/GCTAPsf2D\\_8cpp-source.html#l00570](http://cta.irap.omp.eu/gammalib/doxygen/GCTAPsf2D_8cpp-source.html#l00570)). This would follow the standard treatment like in the rest of the code.  
However, in the implementation of this function it seems that it is evaluated in a different way, as m\_norm being the normalization of only the first gaussian, see lines 607-610.  
Is this wrong implemented or a wrong understanding of me (it would probably also affect the calculation of the derivative)?  
From having a quick look the implementation seems to contradict the doxygen docu of the function.

2.) I recalculated quickly, and potentially found a missing '1-'? As its done at the moment, following the doxygen docu, for a -> infinity, the fraction goes to zero? But shouldn't it go to 1?  
Think more correct would be fraction = 1-fraction\*, with fraction\* the fraction which is used at the moment.

Maybe I am wrong (I am not too familiar with the PSF class), but from a first quick look, the described issues seem strange to me.

#### History

##### #1 - 11/10/2016 04:18 PM - Knödlseeder Jürgen

Thanks for catching that. I think that neither the documentation not the implementation is correct sad.png

I think the correct code should be

```
// Calculate f(a)
double fa(0.0);
fa += 1.0 - std::exp(m_width1 * a2);
fa += m_norm2 * (1.0 - std::exp(m_width2 * a2));
fa += m_norm3 * (1.0 - std::exp(m_width3 * a2));
fa *= m_norm;
fa -= fraction;
```

```
// Calculate f'(a)
double fp(0.0);
fp += std::exp(m_width1 * a2);
fp += m_norm2 * std::exp(m_width2 * a2);
fp += m_norm3 * std::exp(m_width3 * a2);
fp *= -2.0 * a * m_norm;
```

Could you please double check if you would agree with that?

**#2 - 11/10/2016 04:31 PM - Knödseder Jürgen**

- *Target version set to 1.2.0*
- *% Done changed from 0 to 80*

I implemented the change in the branch 1876-correct-containment-radius. Can you check if everything is okay now?

**#3 - 11/21/2016 11:57 AM - Knödseder Jürgen**

- *Status changed from New to In Progress*

For the record: the change has not yet been integrated

**#4 - 01/26/2017 03:53 PM - Knödseder Jürgen**

- *Status changed from In Progress to Feedback*
- *% Done changed from 80 to 100*

Change has been merged in the code. Still needs some testing!!!

**#5 - 03/03/2017 10:15 AM - Knödseder Jürgen**

- *Target version changed from 1.2.0 to 1.3.0*

**#6 - 03/16/2017 11:54 PM - Knödseder Jürgen**

- *Status changed from Feedback to Closed*

Checked and closed