GammaLib - Bug #2112

GCTAPsfKing::containment_radius returns incorrect values

05/10/2017 07:41 PM - Kelley-Hoskins Nathan

Status:	Closed	**Start date:**	05/10/2017
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.3.0		

Description

I've got a fits table of gamma/sigma values loaded into GCTAPsfKing, but I'm running into problems when I try to check the containment radii.

- The value returned by containement_radius() (see c_radius column below) is an order of magnitude larger than a manually-integrated containment radius (i_radius below).
- Whenever gamma dips below 1.0, the containment_radius() tries to take the square root of a negative number, and defaults to zero.

I'm not sure if these are due to a bug in containment_radius(), or if I'm using an incorrect definition of sigma/gamma values.

Values sampled at THETA = 0.5deg
Columns:
energy: gamma-ray event energy
sigma: king function parameter, interpolated from GCTAResponseTable
gamma: king function parameter, interpolated from GCTAResponseTable
c_radius: value returned by GCTAPsfKing::containment_radius() for 68% containment
i_radius: manually integrated 68% containment radius using GCTAPsfKing::operator()

energy[TeV]	sigma[deg]	gamma	c_radius[deg]	i_radius[deg]
0.909242	0.00596238	4.67117	0.629914	0.00918249
0.928012	0.00568582	4.45451	0.607824	0.00959876
0.947169	0.00540927	4.23785	0.585994	0.0100258
0.966721	0.00513272	4.02119	0.564485	0.0104634
0.986678	0.00485617	3.80452	0.543376	0.0109114
1.00705	0.00457962	3.58786	0.522775	0.0113695
1.02783	0.00430307	3.3712	0.502837	0.0114514
1.04905	0.00402652	3.15454	0.483779	0.0115607
1.07071	0.00374997	2.93788	0.465931	0.0135165
1.09281	0.00347342	2.72122	0.44981	0.0246926
1.11537	0.00319687	2.50456	0.436273	0.0342252
1.13839	0.00292032	2.2879	0.426859	0.0410671
1.1619	0.00264377	2.07123	0.424625	0.0458772
1.18588	0.00236721	1.85457	0.436602	0.0494204
1.21036	0.00209066	1.63791	0.483171	0.0518064
1.23535	0.00181411	1.42125	0.654583	0.0536086
1.26085	0.00153756	1.20459	2.21022	0.0549816
1.28688	0.00126101	0.987928	0	0.0560364
1.31344	0.00098446	0.771267	0	0.0567733
1.34055	0.00070791	0.554605	0	0.0573576
1.36823	0.000431359	0.337944	0	0.0578272
1.39647	0.000154808	0.121283	0	0.0582093
1.4253	0.000138935	0.0736697	0	0.0585238
1.45472	0.00045454	0.241018	0	0.0587854
1.48475	0.000770144	0.408366	0	0.059005
1.5154	0.00108575	0.575714	0	0.0591911
1.54669	0.00140135	0.743062	0	0.0593501
1.57861	0.00171696	0.91041	0	0.0594868
1.6112	0.00203256	1.07776	259.986	0.0596052
1.64446	0.00234817	1.24511	2.15946	0.0597085
1.67841	0.00266377	1.41245	0.988206	0.059799

```
   1.71306    0.00297938   1.5798          0.751662       0.0598787
   1.74842    0.00329498   1.74715         0.669153       0.0599494
   1.78451    0.00353927   1.88506         0.637744       0.0600122
   1.82135    0.00343856   1.88054         0.62168        0.0600684
   1.85895    0.00333785   1.87602         0.605521       0.0601187
   1.89732    0.00323713   1.8715          0.589264       0.0601641
   1.93649    0.00313642   1.86698         0.572908       0.060205
   1.97647    0.00303571   1.86246         0.55645        0.0602422
   2.01727    0.00293499   1.85794         0.539889       0.0602759
   2.05891    0.00283428   1.85342         0.523223       0.0603067
   2.10141    0.00273356   1.8489          0.506449       0.0603348
   2.14479    0.00263285   1.84438         0.489565       0.0603606
   2.18907    0.00253214   1.83986         0.472569       0.0603843
   2.23426    0.00243142   1.83534         0.455458       0.0604061
   2.28038    0.00236844   1.83315         0.444463       0.0604263
   2.32745    0.00230954   1.83122         0.434107       0.0604449
   2.3755     0.00225064   1.82928         0.423719       0.0604622
   2.42454    0.00219173   1.82735         0.413299       0.0604782
   2.47459    0.00213283   1.82541         0.402848       0.0604931
============ ============ ========= ============== ==============
```

I've attached the fits file bad_gctapsfking.fits that contains the above GCTAPsfKing table. Anyone should be able to load it via:

```
import gammalib
a = gammalib.GCTAPsfKing()
a.load('bad_gctapsfking.fits')
```

The gamma/sigma values I get are fitted from my simulations. For each energy/offset block of the parameter space, gamma and sigma are found via:

1. From simulations, event positions p_mc and p_rec are found (vectors in camera x,y)
    - p_mc is the original monte carlo event position
    - p_rec is the reconstructed event position
2. a radially-binned histogram of abs(p_mc-p_rec) is made.
    - no scaling of bins based on radial bin area are done (?)
3. a cumulative histogram is then made from the above histogram, and is scaled to have a maximum of 1
4. the containment fraction equation is fitted to this cumulative histogram to find gamma and sigma
    - ContainmentFraction = 1 - (1 + (r*r/2*gamma*sigma*sigma))^(1-gamma)
    - derived from equations 7 and 9 in the 'gammalib maths' pdf

So does anyone have any ideas whats causing these problems? I suspect the histogram in step # 2 needs to be scaled by bin area, but I can't figure out the physical justification for it.

History

#1 - 05/10/2017 08:25 PM - Kelley-Hoskins Nathan

And for clarity, below is the function that calculates the 'manually integrated containment radius' (the i_radius column).

```
def containment_radius( run, en, th, contain=0.68, npts=100 ) :
  """For a given GCTAObservation 'run', energy 'en' and camera offset 'th',
  Calculate the containment radius that contains 'contain' fraction of the events.

  **Args:**
    run     : GCTAObservation object, should already have psf irf loaded
    en      : float, energy within psf parameter space, log10TeV
    th      : float, camera offset within psf parameter space, radians

  **Opts:**
    contain : float, 0.0-1.0, containment fraction to find radius for (contain=0.68 : 68% containment)
    npts    : int, number of integration sampling points to use

  **Returns:**
```

```
  float, radians
"""
samps = numpy.zeros(npts)
dmax  = run.response().psf().delta_max( en, th ) # radians
delta = dmax / npts # radians

# loop over each point
for i in range(npts) :
  d = i * delta # radians
  counts_per_sr = run.response().psf()( d, en, th )   # counts per sr
  if counts_per_sr == 0.0 :
    print('warning: containment_radius(en=%f, th=%f): 0 counts at radius %.3f deg, probably no psf info' % ( en, th, d * rad2deg ))
    return 0.0

  # integrate: counts at radius * perimeter of circle with that radius,
  # will get less accurate when d is large (>>10deg?)
  samps[i]      = counts_per_sr * 2 * math.pi * d

# total counts in counts
total_counts = numpy.sum(samps)

# loop over each sample, check if we cross the containment fraction
for j in range(npts-1) :
  partial_counts      = numpy.sum( samps[:j+1] )
  partial_counts_next = numpy.sum( samps[:j+2] )
  cfrac      = partial_counts      / total_counts
  cfrac_next = partial_counts_next / total_counts
  radius      =  j    * delta
  radius_next = (j+1) * delta

  # check if we have crossed the containment fraction
  if cfrac < contain and cfrac_next > contain :
    # if so, linearly interploate to find what radius has the target containment fraction
    # starting from http://en.wikipedia.org/wiki/Linear_equation , the Two-point form
    contain_radius = radius + ( (contain - cfrac) * (radius_next - radius) / (cfrac_next - cfrac ) )
    return contain_radius

raise LookupError('Could not find containment fraction %.2f within radii 0.0 and %.3f (radians)' % (contain, dmax) )
```

#2 - 06/06/2017 12:22 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- Target version set to 1.3.0

- % Done changed from 0 to 50

The problem is probably related to the fact that the King PSF you used is too large. Internally the radius of the King PSF is fixed to 0.7 deg.

I now changed the logic that no fixed maximum radius is used anymore. Instead the maximum radius is computed from the gamma and sigma parameters, making sure that 99.995% of the function is comprised within the maximum radius. However, if the maximum radius is larger than 2 degrees, the maximum radius will be limited to this number and the PSF area will be corrected.

#3 - 06/06/2017 04:20 PM - Knödlseder Jürgen

- Status changed from In Progress to Feedback

- % Done changed from 50 to 100

The code itself is correct, but the problem you encountered was probably related to a restriction of the PSF radius to 0.7 deg. The maximum PSF radius has been increased to 2 degrees, which should be really enough (otherwise you PSF becomes comparable to the field of view). Can you check if the code in devel is now okay for you?

#4 - 06/09/2017 02:06 PM - Knödlseder Jürgen

- Status changed from Feedback to Closed

Files

bad_gctapsfking.fits	8.44 KB	05/10/2017	Kelley-Hoskins Nathan