G4TorusStack

Peter Gumplinger, TRIUMF

G4 Collaboration Meeting

ESTEC, October 2010

Slides adopted from information available at:
http://neutrino.phys.ksu.edu/~GLG4sim/docs/html_latest/FAQ.html

❑Original work GLG4sim for KamLAND
❑Author: Glenn Horton-Smith

G4TorusStack - to realistically describe the surface of a typical photo-tube (PMT). In a nutshell it allows the definition of a dome surface with varying curvature.

❑Used by DEAP/CLEAN, SNO+, KamLAND, MicroBooNE (to name a few).

•Employ GLG4TorusStack as their PMT model by ‘word-of-mouth’ distribution.

G4TorusStack: is quite useful volume definition and cannot necessarily be replaced by something existing in the G4 toolkit.

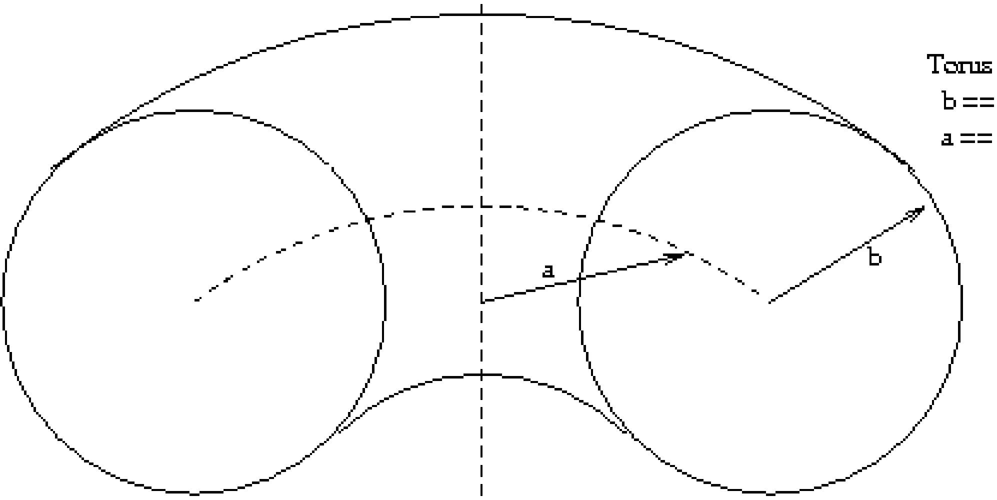
•Adopt it, bring it up to current G4 geometry standards and maintain it for our users in the future.

What is the “TorusStack” solid?

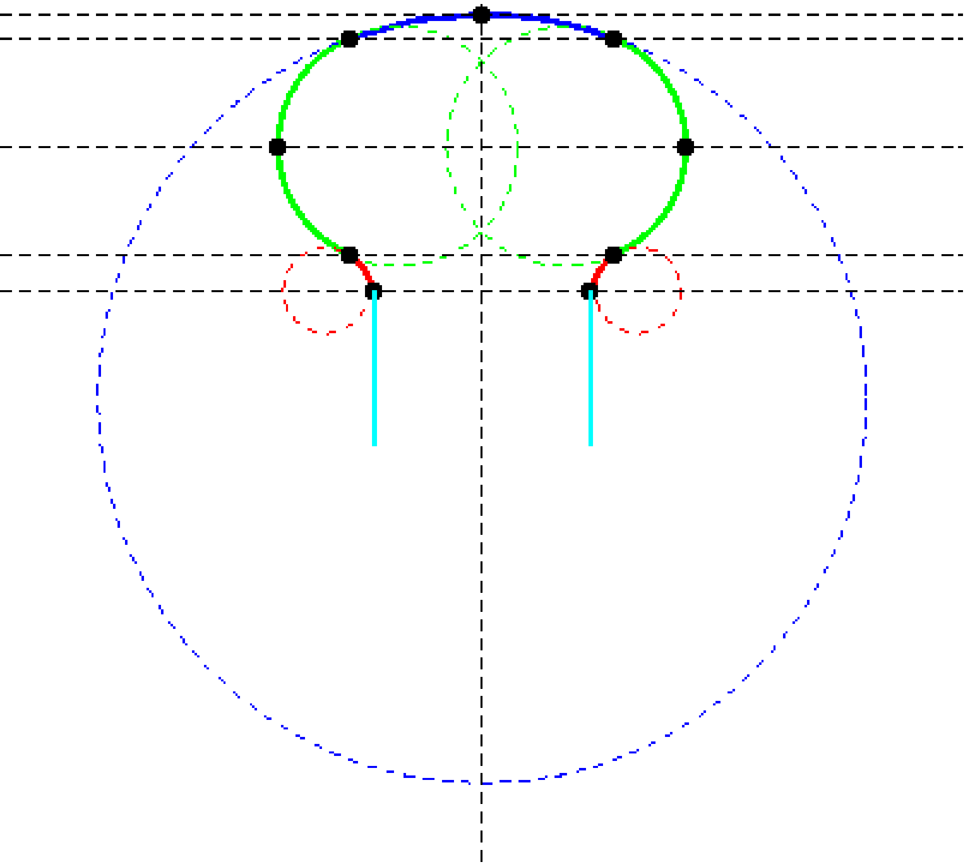
A TorusStack is a stack of toroidal segments, sliced parallel to their planes of symmetry and stuck together.

It is similar to a PolyCone, but with toroidal surfaces instead of conical.

(Note that the parameters “a” and “b” shown in the following graph are calculated internally, but are not used in defining the TorusStack in the code.)



Torus with circular cross-section
b == radius of circular section
a == "swept radius"



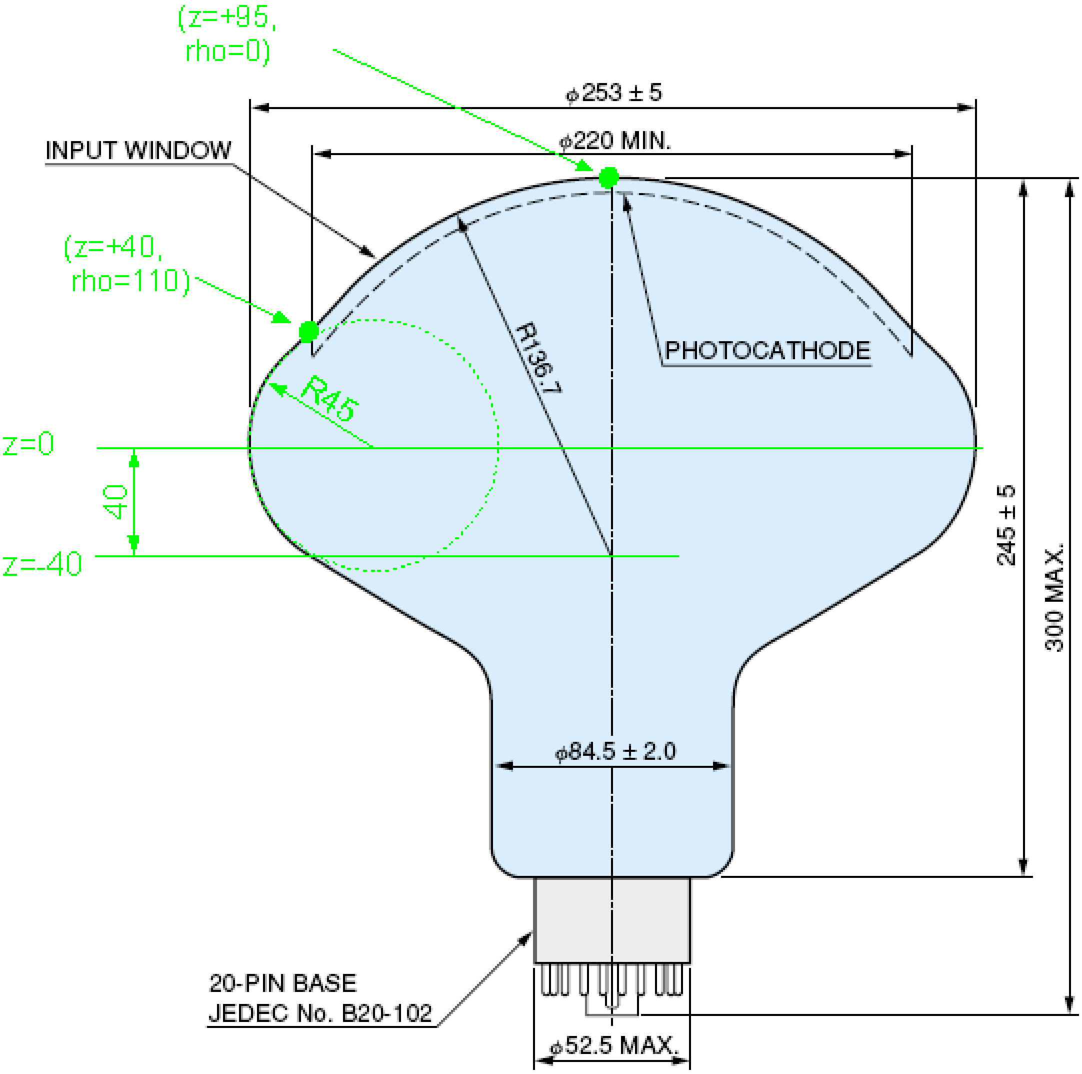
A "torus stack" made from torii
sliced at Z positions and stacked.
Each segment has its own a, b, and
displacement on the Z axis (z0).
KLTorusStack computes a and b
automatically from z and rho of points.

a=0, b>0 ==> sphere.
a>0, b=0 ==> cylinder.
a>0, b>0 ==> convex torus (b>a ok).
a>0, b<0 ==> bounded by inner surf.
of torus ("donut hole")

What are the parameters z_edge, rho_edge and z_o used to define a TorusStack solid?

The TorusStack class constructor must be passed some arrays that define the shape of the TorusStack, as well as the number **n** of segments in the stack. The array parameters are named **z_edge**, **rho_edge** and **z_o**.
The first two arrays are n+1 long, and define the points at which the segments join. The z_o array is n elements, and defines the Z-position of the center of curvature of each segment.

The following diagram (taken from the Hamamatsu PMT manual, with annotations added by hand in green), should help make this more clear:


(z=+95,
rho=0)
φ253 ± 5
φ220 MIN.
INPUT WINDOW
(z=+40,
rho=110)
R136.7
PHOTOCATHODE
R45
z=0
40
z=-40
245 ± 5
300 MAX.
φ84.5 ± 2.0
20-PIN BASE
JEDEC No. B20-102
φ52.5 MAX.

The PMT manufacturer tells us that the top segment of the TorusStack has a radius of curvature of 136.7mm. The center of curvature is about 40mm below the “equator” of the PMT, and the very top of the PMT is about 95mm above the equator. Taking the equator to be z=0, the value of z_o for the first segment z_o[0], should be -40mm. The first segment begins at about z=+95mm, rho=0. The first segment ends, and the second segment begins, at z=+40, rho=110. The center of curvature of the second segment is right on the equator, so z_o[1]=0. The second segment ends at z=0, rho=253/2=126.5. So the values for the first two segments are:

```
z_edge[] =     {95,        40,         0.    …
rho_edge[] = {  0,      110,      126.5  …
z_o[] =          {    -40,         0,           …
```

TorusStack is able to compute values of “a” and “b” for each segment from z_edge, rho_edge, and z_o. If there is some inconsistency in the input parameters, TorusStack does the best it can to resolve it and prints a warning

Acknowledgement: The graphs and the text in this presentation has been shamelessly copied from the GLG4TorusStack documentation available at:

http://neutrino.phys.ksu.edu/~GLG4sim/docs/html_latest/FAQ.html