Obtaining the kdd scalar

-draft-

January 2015

1 Scope and purpose

This document records how the kdd scalar value used in the TPS (more specifically the .rdu file) was determined.

The code below was written with Python 2.7

2 Approach and code

Import libraries

```
import matplotlib.pyplot as plt
import numpy as np
import struct
```

The kernel we are using is located at Z=61.

```
filePath = "D:\\Ceres\\Resource\\PlanEngine\\R8\\Kdd\\R8O0IQ00_Y000Z061C025.d3d"
f = open(filePath,'rb')
```

Read in the xsym, ysym, and zsym

```
xsym = struct.unpack('i',f.read(4))[0]
ysym = struct.unpack('i',f.read(4))[0]
zsym = struct.unpack('i',f.read(4))[0]
```

Read in nx, ny, and nz

```
nx = struct.unpack('i',f.read(4))[0]
ny = struct.unpack('i',f.read(4))[0]
nz = struct.unpack('i',f.read(4))[0]
```

Create boundary lists

```
xBoundary = []
yBoundary = []
zBoundary = []
```

Read in the boundaries

```
for i in range(0,nx+1):
    xBoundary.append(struct.unpack(’f’,f.read(4))[0])

for i in range(0,ny+1):
    yBoundary.append(struct.unpack(’f’,f.read(4))[0])

for i in range(0,nz+1):
    zBoundary.append(struct.unpack(’f’,f.read(4))[0])
```

Create dose matrix

```
dose25   = np.empty((nx,ny,nz))
dose25[:] = np.NAN
```

Read in the dose matrix

```
for i in range(0,nx):
    for j in range(0,ny):
        for k in range(0,nz):
            dose25[i,j,k]=struct.unpack(’f’,f.read(4))[0]
```

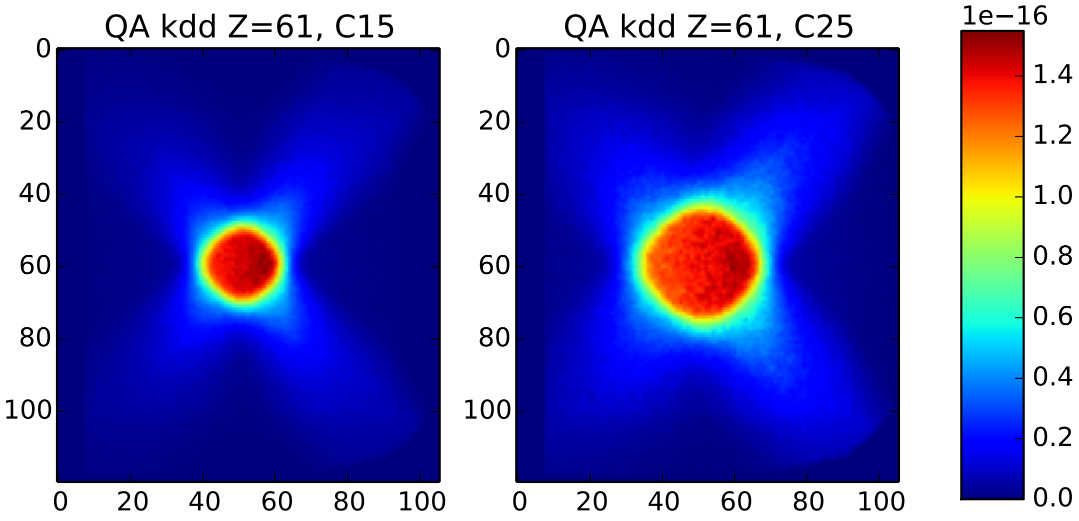
Repeat the same steps for the 15mm collimator

```
filePath15 = "D:\\Ceres\\Resource\\PlanEngine\\R8\\Kdd\\R8O0IQ00_Y000Z061C015.d3d"
f15 = open(filePath15,’rb’)


# read in the xsym, ysym, and zsym
xsym15 = struct.unpack(’i’,f15.read(4))[0]
ysym15 = struct.unpack(’i’,f15.read(4))[0]
zsym15 = struct.unpack(’i’,f15.read(4))[0]

# read in nx, ny, and nz
nx15 = struct.unpack(’i’,f15.read(4))[0]
ny15 = struct.unpack(’i’,f15.read(4))[0]
nz15 = struct.unpack(’i’,f15.read(4))[0]

# create boundary lists
xBoundary15 = []
yBoundary15 = []
zBoundary15 = []

# read in the boundaries
for i in range(0,nx15+1):
    xBoundary15.append(struct.unpack(’f’,f15.read(4))[0])

for i in range(0,ny15+1):
    yBoundary15.append(struct.unpack(’f’,f15.read(4))[0])

for i in range(0,nz15+1):
    zBoundary15.append(struct.unpack(’f’,f15.read(4))[0])
```

```
# create dose matrix
dose15   = np.empty((nx15,ny15,nz15))
dose15[:] = np.NAN


# read in the dose matrix
for i in range(0,nx15):
    for j in range(0,ny15):
        for k in range(0,nz15):
            dose15[i,j,k]=struct.unpack('f',f15.read(4))[0]
```

Displaying the central cuts

```
f, (ax1,ax2) = plt.subplots(1,2, figsize=(12,7))
img = ax1.imshow(dose15[0,:,:])
ax1.set_title('QA kdd Z=61, C15')

img = ax2.imshow(dose25[0,:,:])
ax2.set_title('QA kdd Z=61, C25')

f.subplots_adjust(right=0.8)
cbar_ax = f.add_axes([0.85, 0.25, 0.05, 0.5])
f.colorbar(img,cax=cbar_ax)
```



The maximum values in the kernels

```
QA15max = np.nanmax(dose15)
QA25max = np.nanmax(dose25)

print ("Max C15 = {0}\nMax C25 = {1}".format(QA15max,QA25max))
```

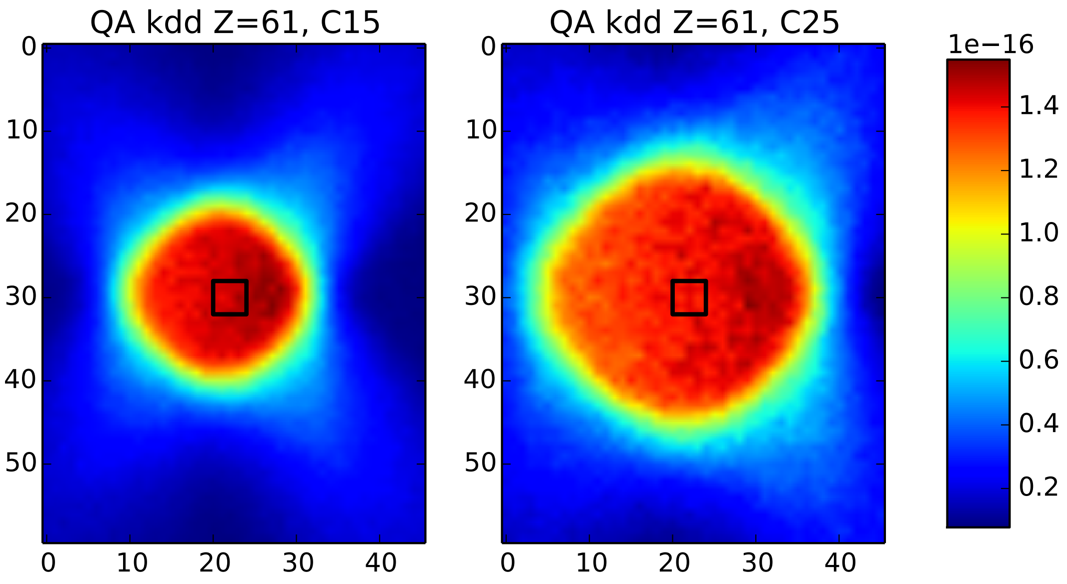
```
Max C15 = 1.42194451645e-16
Max C25 = 1.5489506218e-16
```

During calibration, the phantom is placed so that the ion chamber's sensitive area is located at a position corresponding to the center of the kernel.

```
QA15Centroid = dose15[0,60,52]
QA25Centroid = dose25[0,60,52]

print ("Centroid C15 = {0}\nCentroid C25 ={1}".format(QA15Centroid,QA25Centroid))
```

```
Centroid C15 = 1.31233885013e-16
Centroid C25 =1.40710317573e-16
```

For a more realistic value, the average of a 5x5 square around the center is used as the kdd scalar.



```
QA15CentroidAveraged = np.mean(dose15[0,58:62,50:54])
QA25CentroidAveraged = np.mean(dose25[0,58:62,50:54])
```

```
print ("Centroid averaged C15 = {0}\nCentroid averaged C25 = {1}".format(QA15CentroidAve
print ("Centroid averaged C15 / Centroid averaged C25 = {0:1.4f}".format(QA15CentroidAve
```

```
Centroid averaged C15 = 1.32605903852e-16
Centroid averaged C25 = 1.39012132319e-16
Centroid averaged C15 / Centroid averaged C25 = 0.9539
```

3 Result

```
For the 25mm collimator:

Value at center: 1.40710317573e-16
Value at center averaged: 1.39012132319e-16
```

Which results in a kdd scalar of

```
Center value: 7.10679939644e+15
Averaged values: 7.1936167248e+15
```