Independent dose calculation of Brachytherapy Dicom files

# Introduction

This document outlines the Dicom based TG43 calculation program I have written in python. It aims to independently calculate the dose based on the dwell positions within an Oncentra® RT Dicom Plan, and compare the resulting dose grid to an exported Oncentra® dose grid within an RT Dicom Dose file.

The code itself can be accessed from <https://github.com/SimonBiggs/teap-brachytherapy-portfolio/blob/master/TG43%20Implementation.ipynb> .

# Future direction

Collect a range of brachytherapy dicom files that are able to be placed within the Gihub repository that can be used for testing. Aim to support as many brachy dicom formats as possible.

Once egs\_brachy (<https://doi.org/10.1088/0031-9155/61/23/8214>) is available for use I would like to directly implement that model based dose calculation method to be able to provide checking for both TG43 and Monte Carlo based algorithms.

It is also planned that upper and lower 95% confidence interval doses will be reported when uncertainties due to catheter movement, catheter reconstruction, and calculation uncertainties are taken into account. Using the structure dicom file these 95% confidence interval doses can be converted to comparative DVHs.

The final stage is for this code to rerun a dwell time optimisation to see if any plan improvement is possible and compare the robustness of the original plan to positioning uncertainties with the calculated comparative plan.

# Cautionary warnings

Brachytherapy dicom files make use of a large number of private dicom tags. What some of these tags mean needs to be reverse engineered at times. This program has as of yet only been tested with Oncentra® dicom files, not BrachyVision™ dicom files. It has only been tested with a small subset of Oncentra® dicom files, with some of these files for certain configurations this code does not yet work. Be sure when using this code to investigate the testing figures produced to confirm that they represent what is expected within the plan.

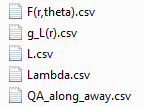
Pay particular attention to x,y,z definitions, catheter definitions, and source orientation.

# Method

## Source data

The current source data used was retrieved from <http://www.estro.org/binaries/content/assets/estro/about/gec-estro/tg43-sources/new/ir-192/192ir-hdr-nucletron-mhdr-v2.xls>. It is provided as the consensus TG43 source data.

The source data is supplied to the program in the form of a range of CSV files. These files look like the following within the file browser:



In that way by changing which version of these files the code is pointing to a different set of source data can be used.

### Geometry

The consensus source data is based on the revised mHDR-v2 source model which is given below in Figure 1:

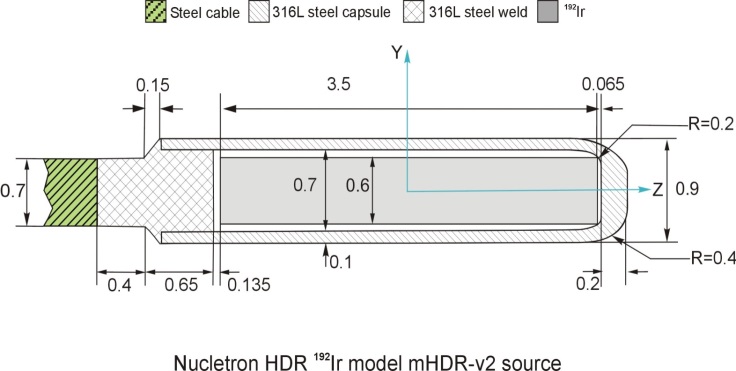


Figure 1: The revised Nucletron HDR Ir-192 mHDR-v2 source model

It has a length of 0.35 cm, according to this source data a of 1.109 cGy/(h U). The radial function data and the anisotropy function data are represented in Figure 2 and Figure 3.

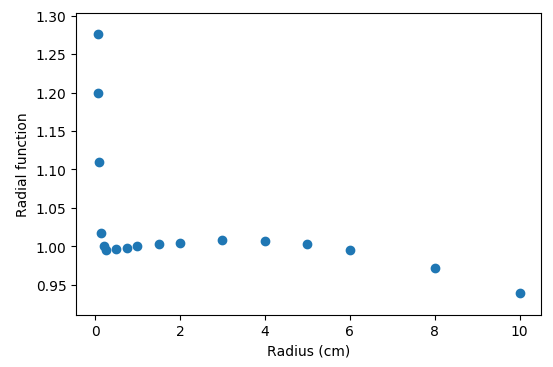


Figure 2: The radial function data

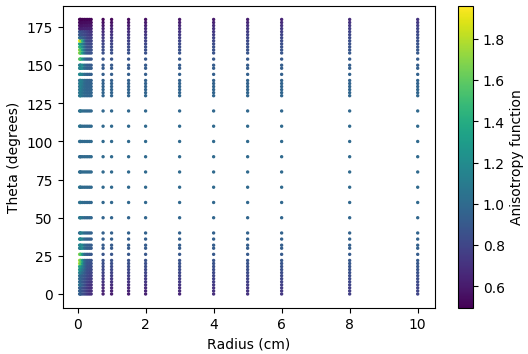


Figure 3: The anisotropy function data

## TG 43 implementation

TG-43 implementation follows the recommendations of TG-43U1S1.

### Geometry function implementation

To implement the geometry function the formulae provided by the following paper was used:

King, R. P., Anderson, R. S. and Mills, M. D. (2001), Geometry function of a linear brachytherapy source. Journal of Applied Clinical Medical Physics, 2: 69–72. [doi:10.1120/jacmp.v2i2.2615](https://dx.doi.org/10.1120/jacmp.v2i2.2615)

To aid in understanding the features of the geometry function the following interactive web app was also produced:

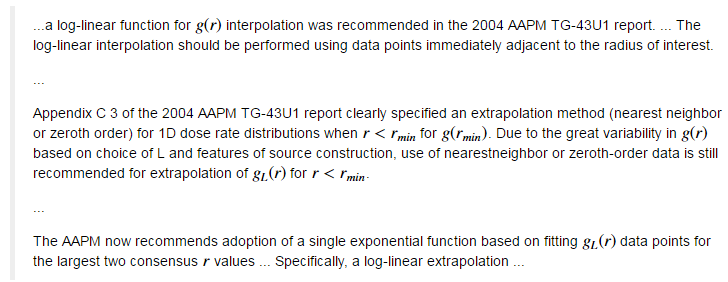
<https://www.geogebra.org/m/K25b5dNV>

### TG-43U1S1 recommendations

TG-43U1S1 (<https://www.aapm.org/pubs/reports/rpt_84S.pdf>) provides recommendations for the implantation of TG-43. It explicitly provides recommendations for how to interpolate and extrapolate the data.

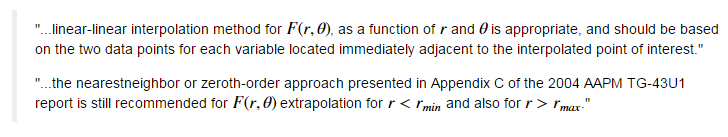
#### Radial function

The recommendations from TG-43U1S1 for the radial function are the following:



#### Anisotropy function

The recommendations from TG-43U1S1 for the anisotropy function are the following:



### Radial function interpolation

The resulting radial function interpolation can be seen in Figure 4.

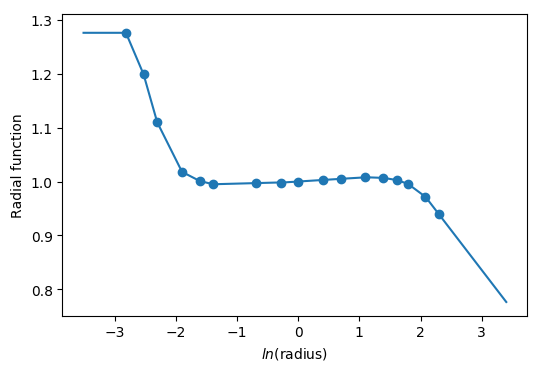


Figure 4: The interpolation of the radial function data

### Anisotropy function interpolation

The resulting anisotropy function interpolation can be seen in Figure 5.

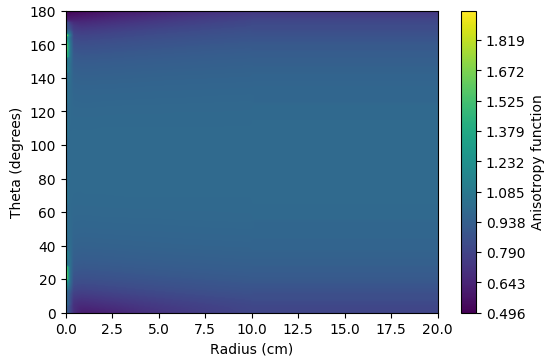


Figure 5: The interpolation of the anisotropy function

## Testing of TG 43 implementation

The source data provided also comes with a set of “QA Along Away” data points. These grid points were input into the TG-43 calculation and compared with the reference results. The result of this comparison is given in Figure 6.

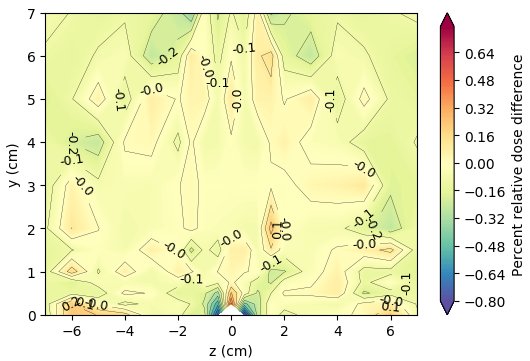


Figure 6: The results of the TG-43 implementation QA

### Concerning issue regarding length parameter

An issue of concern regarding this QA result is the features adjacent to the source. When the length value used within the geometry function is changed from the recommended 0.35 cm to 0.36cm the QA result looks as given in Figure 7.

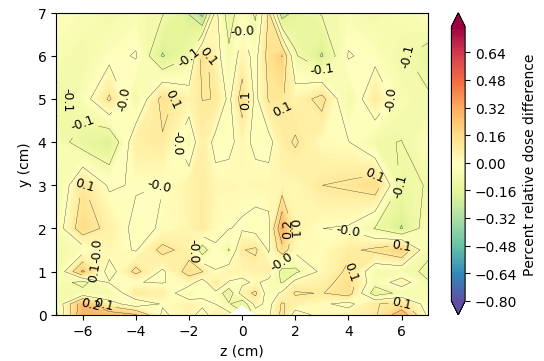
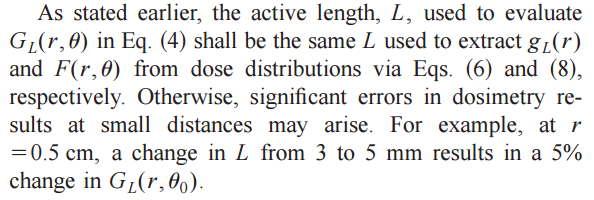


Figure 7: The result of the QA check when a length of 0.36 cm is used.

This is concerning given the following comment from [TG-43U1](http://dx.doi.org/10.1118/1.2184438):



Given that agreement begins to deviate when the recommended source length is used it is suspected that an incorrect source length may have been used in the production of the consensus data. Nevertheless, until further investigation is undergone the recommended source length of 0.35cm will be used.

## Dicom plan file reading

To read the Dicom plans a large number of processes are required. To best understand how this is done one would need to go through the import code itself. For the purpose of this document a short overview of where the required information is found will be provided.

### PyDicom

To read Dicom files within Python a module called pydicom is available. Documentation for this module is available at <http://pydicom.readthedocs.io>.

The loading of the plan and dose dicom files with pydicom can be achieved using the following code:

**import** **dicom**

dcm\_dose = dicom.read\_file("dicom\_dose\_filepath.dcm", force=**True**)

dcm\_plan = dicom.read\_file("dicom\_plan\_filepath.dcm", force=**True**)

### Reference Air Kerma Rate

Reference air kerma rate can by retrieved from the dicom header using the following code:

**import** **numpy** **as** **np**

reference\_air\_kerma\_rate = np.float(

dcm\_plan.SourceSequence[0].ReferenceAirKermaRate) / 360000

### Dwell positions and Dwell times

The dwell positions and dwell times are stored within the following Dicom tag:

dcm\_plan.ApplicationSetupSequence[0].ChannelSequence

A function is provided for the pulling of dwell positions, channels, and times. It is used in the following way:

dwell\_positions, dwell\_channels, dwell\_times = pull\_dwells(dcm\_plan)

An example result from using this function is given in Figure 8.

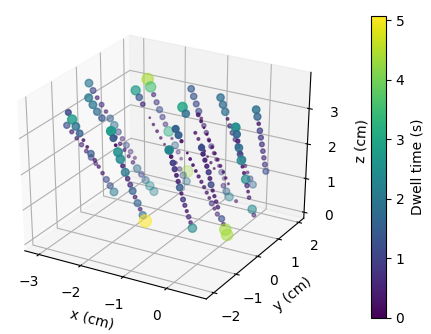


Figure 8: Extracted dwell positions and dwell times.

### Dwell orientation

Dwell orientation is determined using the catheter coordinates. A B-spline interpolation is made of the catheter coordinates and the derivative of this spline is then taken. The derivative along each axis is then converted into a three dimensional unit vector for each dwell position. This is the dwell direction.

A function is provided which determines the dwell directions. It is used as follows:

dwell\_directions = determine\_dwell\_directions(

dcm\_plan, dwell\_positions, dwell\_channels)

Dwell directions are printed out and displayed for each catheter. An example for the output provided for one catheter is given in Figure 9.

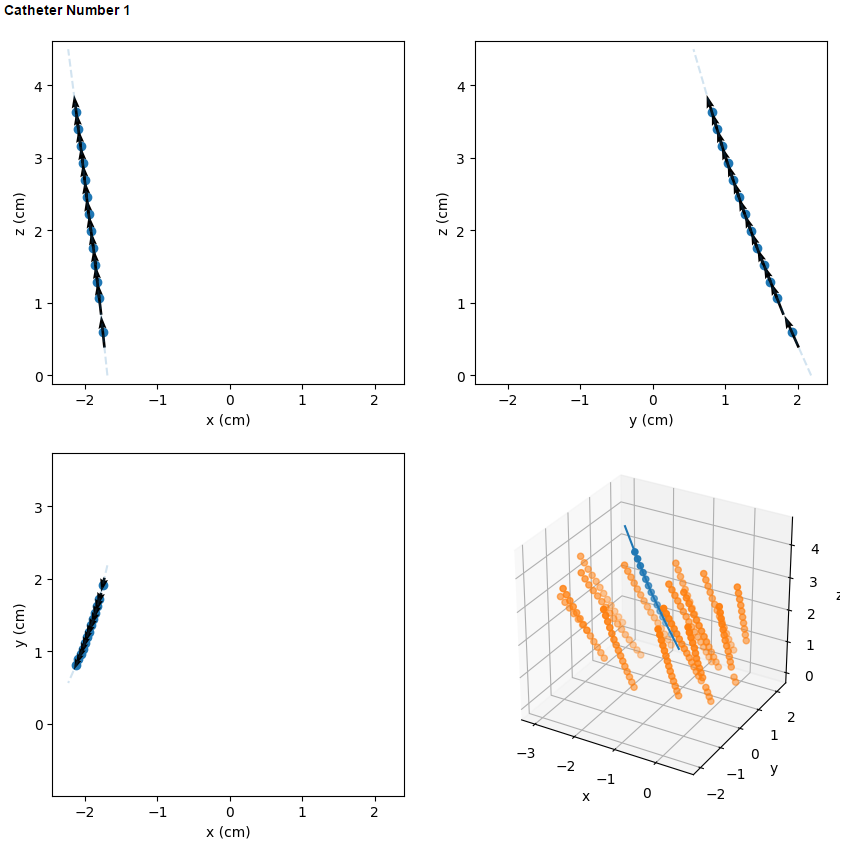


Figure 9: The dwell directions are displaced in x, y, and z directions for the user to verify dwell direction accuracy

## TG-43 usage

Once all of the required dwell information is extracted from the Dicom file the following implemented function can be used to calculate TG-43 dose:

tg43\_dose = tg43\_on\_grid(

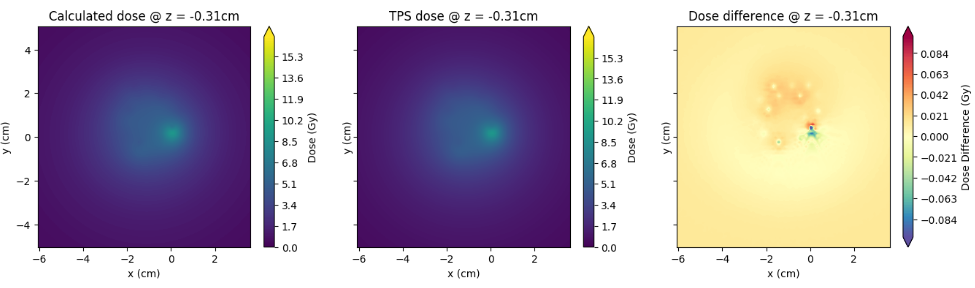
calc\_x, calc\_y, calc\_z, reference\_air\_kerma\_rate,

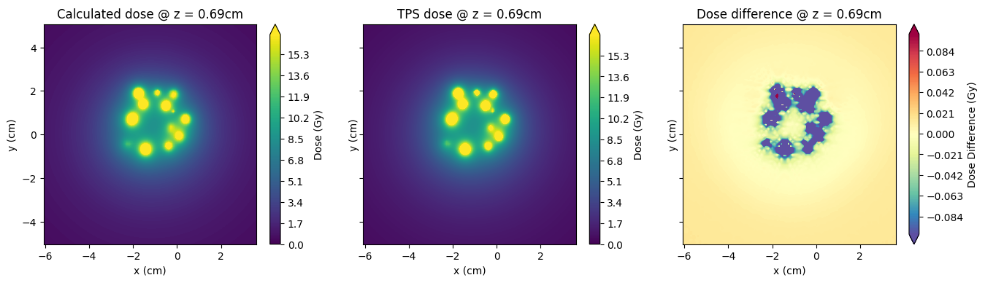
dwell\_times, dwell\_positions, dwell\_directions)

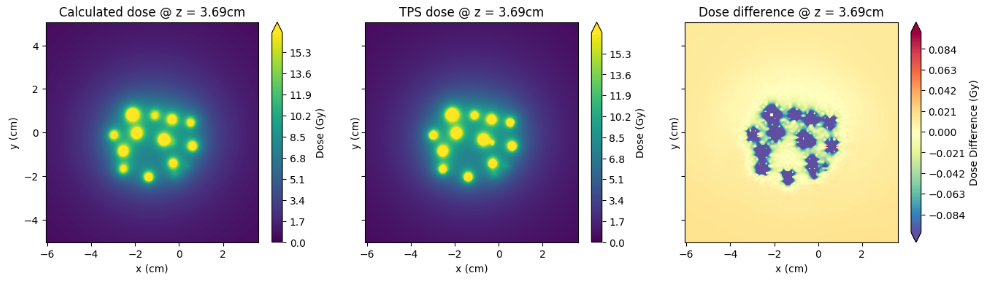
## Comparison with Dicom dose

The calculation grid is extracted from the Dicom dose file. This calculation grid is then used for the TG-43 calculation. Once the calculation is complete a direct comparison is undergone between the calculated dose and the TPS dose.

Examples of these comparisons are given in Figure 10.







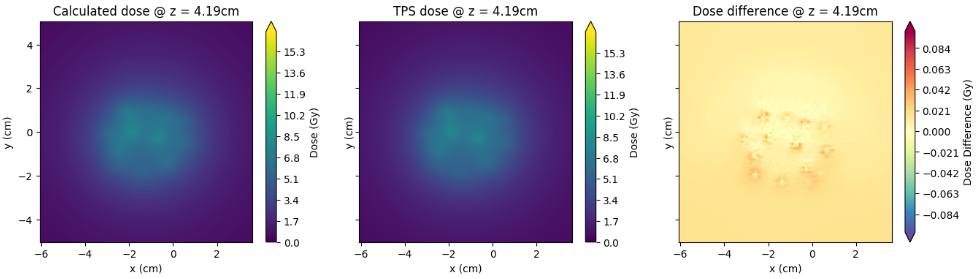


Figure 10: Dose differences between TPS and calculation for a range of slices.