## Input

* Measured cutout factors
* Shape defining boundary points

## Create width weighting

* Do a basic ellipse fit
* Spline fit the data
* Pull the aspect ratio one slice “circles”

## Clarkson “centre finding”

* Use the Clarkson method to find the dosimetric centre
  + For “indents” use the difference between open areas in a sector
  + Assume dose increase to be flat for regions larger than extrapolation region
  + Do not let middle point be defined closer than smallest cutout to the edge of shape
    - provide error if the minimisation algorithm states it should be closer to edge than smallest cutout.
  + Compare this method to the circles alone method to see how it compares at cutout factor prediction
* Make use of the circle fit provided by all of the data
* Iterate over possible centres , aiming to maximise relative dose with position
* Check to see if this is equivalent to the derivative weighting method used originally

## Shape straightening

* Straighten the shape around the calculated dosimetric centre

## Ellipse fit

* Fit the ellipse to the straightened shape

## Iteration

* Iterate the width weighting, centre finding, straightening, an ellipse fit, with the newly calculated widths and lengths.
* Do this until predicted ellipse parameters no longer change by more than 0.1 mm

Run this whole code all the way through with less input data points to estimate the uncertainty for varied number of input data points. Create a fit comparing number of data points measured to the uncertainty of prediction. Create this fit with uncertainty bars on the calculations to give a good idea of what you can expect if you measure a certain amount of points – will have to set this up to run for a decent amount of time.

Write the entire program within IPython Notebook. Do each section in a different notebook. Use the “%%writefile foo.py -a” magic in order to write up the modules. Have a top level controller script that defines where / when everything is run. I think there is no harm in running everything from module compilation through each time.

Package the whole thing up in Docker. See if “excel files” can be dragged and dropped into the IPython-dev interface.