Department of Physics: MSc in Fusion Energy

ICF Data Laboratory

Week 1: Self Emission Imaging "First look"

- Script Updated: January 2021

Data Analysis procedures

To analyse this data you need to gain familiarity with a variety of software packages and are encouraged to spend time and play with the software. This is the main method to increase your familiarity and success at using it.

You will need to refer to the pre-reading script to extract information for this lab.

Aims

The objective here is to estimate the diameter (with errors) of the imploded core with time. You will all analyse an early time frame (where compression is minimal), an in addition will be assigned one later time frame from the GXD signal (with data from the compressed capsule) to analyse. In week 3, results from the entire class will be brought together, and the final result is a graph of capsule core radius (R), density (ρ and n_e), and density-radius (ρR) product with time.

To do this will take a number of cross-sections at various angles across the image. We could tackle this task with Python, but it is not trivial, and not easily automated, so instead we will generate the lineout with an image analysis package called ImageJ.

You should start out by completing this analysis with the first frame of data, and then, once you have compared our answer to the initial radius, and estimated errors, repeat analysis on your assigned frame. You should therefore make sure that your Python scripts are easy to reuse!

Taking lineouts in ImageJ

1. Make sure you are in your Linux partition on your laptop, and open ImageJ.
2. Via the menu item *File\Open*, open the GXI data image - be sure to choose the right image, and the correct data point (compare the images you have to the annotated version in the pre-reading script).
3. You can use *Image\LookupTables\Fire* etc. to convert your monochromatic image into colour. *Image\Adjust\Brightness\Contrast* is also useful. It is important to consider what these commands are doing. Do they change the image itself, or just how it is displayed?
4. You now need to take cross-sections from your image. From the tool bar select the straight line sections option. Draw a line selection by click-dragging a line across the image. *(Ctrl+k)* brings up a graph showing the cross-section along the line. You can draw a line at any angle across the image.
5. You might want to click the 'live' button on the lineout window so that it responds in real time to any changes you make to the line selection.
6. Double-click on the tool bar's straight line sections option to increase the width of your line. This averages the cross-section over the stated number of pixels reducing noise in the data.

7. It is important that you analyse a number of cross-sections (perhaps 4) through the centre of each image (i.e. a diameter). Think about how best to ensure that your line is along a diameter and not a chord.
8. Data from the cross-sections should be saved to a text file by using the 'Save' button on the lineout window.

At the end of this section you should have several lineouts ready to analyse. Remember, your lab book should have all the info you need to recreate these lineout files from scratch, as well as a record of which file on your computer each lineout is stored in.

Analysing Lineouts in Python

We will now analyse the lineouts we have created in Python. To do this, we will fit a function representing the imploded core density to the lineouts obtained above. If you have not already looked at the example Python code in 'fitting.py', and the lab script on use of Python, do so now!

1. First we must load the lineouts we generated in ImageJ. There is a command in the icf library for this

   ```
   xdata, ydata = icf.load_2col("filename")
   ```

   This will load the data into two numpy arrays, xdata and ydata, which will contain the pixel numbers and intensities respectively. Remember that you will need to import the icf library before using this command.
2. You could now plot the data to confirm that it has loaded correctly. See the example script reader.py to see how to do this (it only takes two extra lines of Python!)
3. The digitisation scale of the image is 1 pixel = 60 µm. To convert your xdata from pixel number to mm use either of these commands

   ```
   xdata = xdata * 60.0/1000.0
   xdata *= 60.0/1000.0
   ```

   These are just two syntaxes for doing exactly the same thing – choose the one you find clearest.
4. You can now try to fit a Gaussian to your data. As a starting point, you can look at the example 'fitting.py', and the lab script on use of Python. You can copy any lines from the example scripts into your own, but remember to also copy any required 'import' statements. To start, alter the fitting routine to use your loaded data rather than generate a Gaussian.
5. We need to make a small modification to the Gaussian function in that script as it assumes that the function tends to zero as we move away from the peak, wheres your data will be sat on a large, non-zero background. You can solve this by adding a fouth plotting parameter, Y_0, such that $G(x) = Y_0 + Ae^{(x-x0)^2/2c^2}$. To do this you need to make a few small modifications
   (a) Add another parameter to the gaussian function

   ```
   y0 = params[3]
   ```

   (b) Modify the function in the return statement to include Y_O

   ```
   return y0 + A*np.exp(-(x-x0)**2/(2*c*c))
   ```

   (c) Make sure add a fourth value for the initial guesses in the fitting routine

   ```
   guess = [1,1,1,1]
   ```

Obviously, you may also need to choose more reasonable initial guesses than this, based on your own data!

6. If you make reasonable initial guesses, you should now have a fit to the data. By following the above steps you can modify the fitting function to be anything you wish.

7. Our aim now is to make this fit as good as possible. The fitting.py script calculates R^2 for you, so this is one figure of merit you can use to find the best fit. There are two things you should try to get the best possible fit.

   (a) Firstly, it is unlikely that your baseline is flat as it consists of signal from the glowing hohlraum. So, try restricting your fit to a region which extends only slightly wider than your core. You can do this either by trimming your arrays (xdata and ydata) just after you have loaded them in, or, you can restrict the range of data passed to the curve_fit command. You may find the following command useful

   ```
   icf.find_closest(xdata, 1.2)
   ```

   This would return the index of the element in xdata closest to 1.2. So for example, to trim the xdata arrays to only include x values between 1 and 2 you could use

   ```
   xmin = icf.find_closest(xdata, 1.0)
   xmax = icf.find_closest(xdata, 2.0)
   xdata = xdata[xmin:xmax]
   ```

   Remember that you would need to do the same cropping to 'ydata' as both arrays must have the same length!

   (b) As discussed above, you can edit fitting.py to use your own function. Try fitting a super-Gaussian, see Figure 1, to your cross-section. One way to express the super-Gaussian has is $f(x) = y0 + A * exp(-((x - x0)/\sqrt{2}c)^n)$. So, n=2 is just a normal Gaussian, and c is related to the width. You should explore using n=4, 6, 8 etc. to find the best possible fit (it may not be the same function for each lineout)

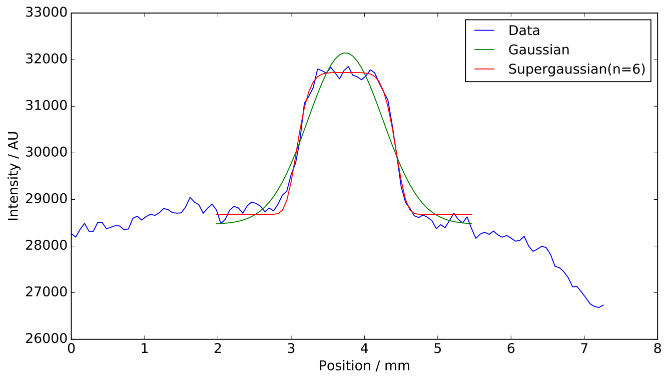

Figure 1: A Gaussian (green) and super-gaussian with n=6 (red) curve fit the to same WAX cross-section. Notice that the fit is only done in a narrow range about the core. In this case, the super-gaussian is a significantly better fit.

   (c) The FWHM, which is related to core diameter, is defined in terms of c. To extract the FWHM set $y - y_0 = A/2$, and rearrange for x.

You should have a number (say 4) of FWHMs for your chosen image. A convenient way to determine the average and standard deviation (and higher order moments) is to use statistics package (see stats.py for examples).

Analysis

Due to the GXI spatial resolution, which includes the size of pinholes, the microchannel-plate resolution (27 $\pm$ 6 µm) and digitisation size (60 µm), an image of a point source is blurred. The degree of spreading (blurring) of the point object is a measure for the quality of an imaging system. In general, any image is a convolution of the object (in this case the imploding core) and the instrument point spread function or PSF.

A simple approximation to the point spread function full-width-at-half-maximum (PSF_{FWHM}) is to add all sizes in quadrature such that

$$PSF_{FWHM} \approx \sqrt{pinhole\,size^2 + microchanel\,size^2 + digitisation\,size^2} \tag{1}$$

It is necessary to estimate the effect of your imaging system on a measurement by assuming the PSF approximates a Gaussian and finding the FWHM. Here we assume the object, the recorded image (I) and PSF may be represented by Gaussian's. With this assumption the object, or core diameter, FWHM (the result you are interested in obtaining) is

$$core\,diameter_{FWHM} = \sqrt{I^2_{FWHM} - PSF^2_{FWHM}} \tag{2}$$

You should also remember that your results to this point are a measure of the width of the image on the detector, which is not necessarily the same as the physical size of the core. Check back on the details of the experimental setup to look for any further corrections.

Use your estimate of core diameter FWHMs to determine (with estimates of errors) radius R. You can assume that the density (ion and electron) are constant throughout the core, and that negligible mass is lost via ablation. There is sufficient information to determine the initial core density in Table 1 of the pre-reading script.

Try to automate this process so that your second analysis, on later time data, is as streamlined as possible (in a real experiment you would have far more than two data points to analyse).

Writing up

Your laboratory notebook should contain details of all the analysis undertaken. This includes recording estimates, stating what images were used, the position of the cross-sections, width of the cross-sections, the parameters used for cross-section reduction etc. You should include appropriate images of the steps taken. In short, your laboratory notebook must contain a record of all the important experimental details to enable a reproduction of analysis.

Your final results should include an estimate of the imploded core radius or diameter for both the initial frame and your assigned frame. You must quote errors when reporting results. This is your opportunity to discuss parts of your analysis that requires further investigation (you can start trying to attempt one of these now if you have time, but we will spend more time on this in week 3). List these, be precise about the issues and identify possible solutions.