

RADIUS OF GYRATION CALCULATOR

Advanced Undergraduate Physics Laboratory

Radius of Gyration

When studying DNA and other polymers, it is important to know how much space a strand takes up at various times. The radius of gyration is one way of parametrizing the “size” of a chain. It is a scalar quantity with units of length, defined as:

$$R = \sqrt{\frac{1}{N} \sum_{i=1}^N (\vec{r}_i - \vec{r}_{CM})^2}$$

where \vec{r}_i is the vector to each bead and \vec{r}_{CM} is the center of mass of the bead.

Plotting the radius of gyration for different configurations versus the time interval of the snapshots taken will show you how the radius of gyration changes over time. The maximum radius of gyration is when the chain is in a perfectly straight line, in which case it can be shown that

$$R_{\max} = \frac{L}{2\sqrt{3}},$$

where L is the length of the chain. It may be interesting to determine the radius of gyration as a fraction of this maximum value, and plot this versus time. Does the chain approach a certain value of R/R_{\max} ? If so, how does this value depend on the various physical properties of the chain and shaking process?

Overview of the Program

The program is a MATLAB script designed to work under MATLAB 7 and higher. The script calculates the radius of gyration of series of pictures taken using the video camera setup of the “Knots” experiment. The program will return the radius of gyration at each picture as well as the estimated error in the calculation.

Note: The program is far from fool proof. In order to avoid unexpected crashes of the program, please make sure that all the values are entered correctly, and no required info is missing. In case you did forget to enter a required value, or entered an incorrect value (e.g. a word instead of a number) the program will give out an error or crash without a warning.

Procedure

Taking Still Images

In the video capture program press the L key in order to record a still image. You can record a sequence of still images either using the program or manually, pressing L periodically keeping the track of time. The captured images are stored in “My Documents” folder. For each trial, store the pictures of the trial in a separate folder. Check that all the images are clear of any artifacts, and that only the images of the trial present in the folder.

Note: The file name for each captured image. It shows the date and time when the image was taken. For example: “TV2006050311471300.jpg” was taken on May 3rd 2006 at 11:47:13. The two last digits represent the sequential number of the images taken at that particular second. Make sure that all the files are named this way. Also note that there is no higher precision than seconds, thus fully relying on the filename to identify the time the picture was taken on won’t produce accurate results.

Running the Program

Download and unpack “gyration.zip” package. On your computer, open MATLAB. After MATLAB is loaded (which may take some time), change the working directory to where you have unpacked the package. Open the Radius of Gyration program by typing in:

```
>> gyration
```

A new pop-up window will open, as shown in Fig 1. This is the Radius of Gyration Program panel.

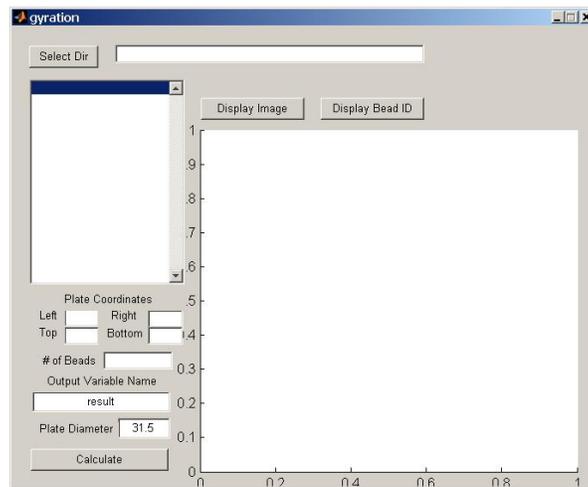


Fig 1. Radius of Gyration program window.

Radius of Gyration Panel

Note: Image processing is a lengthy process. If you click on a button and see no immediate effect, wait a couple of seconds before clicking it again. Some things take time to execute.

Start by clicking on “Select Dir” button and selecting the directory with the images for the trial you want to work with. After the directory is selected, the directory path will appear in the top text box and the images in the list box below. Also, another window (figure window) will open showing the first image in the sequence. An example of such a window is shown in Fig 2.

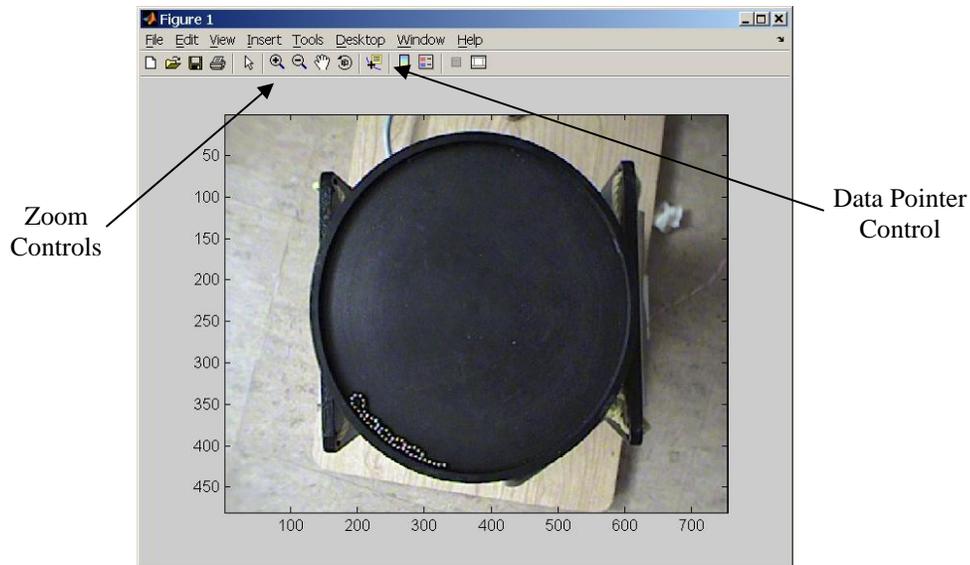


Fig 2. Figure window showing the first image of the trial.

Use the data pointer and zoom controls (as shown on the figure) in order to get the leftmost and rightmost (x-values), topmost and bottommost (y-values) values of the plate. Put the obtained values into the appropriate Plate Coordinates textboxes back on the panel.

Note: It is important that to do this procedure for each and every trial since the camera might move slightly changing the coordinates. Select the coordinates carefully, since incorrect selection may lead to unidentified beads, or other areas of the images wrongly identified as beads (Fig 3).

Also insert the number of beads in the chain in the appropriate textbox. The results of the calculations will be exported into the main MATLAB screen in a form of a matrix. Select the name for the variable you want to have the results stored in. Warning, if a variable under the selected name already exists, the data will be overwritten without any notifications. Also, enter the plate diameter in the appropriate box (without the units). The units the value is entered in will be the same units you receive your answers in (i.e. if the diameter is entered in cm the result will be shown in cm. Do not enter the units. The textbox will only work with numerical values).

Display Image and Display Bead ID

The two options can be used after all the information in the boxes is entered. Select an image from the list box to the left. The Display Image button will show the image in the window on the panel. The Display Bead ID button will show the image with the beads in blue circles and green line encircling the plate surface based on the coordinate selection. The program will not look for beads outside of the circle. Use those buttons to verify that the program works properly, and does not identify other areas of the image as beads. A couple of random checks are suggested for each trial. An example of right and wrong coordinate selection is shown in Fig 3.

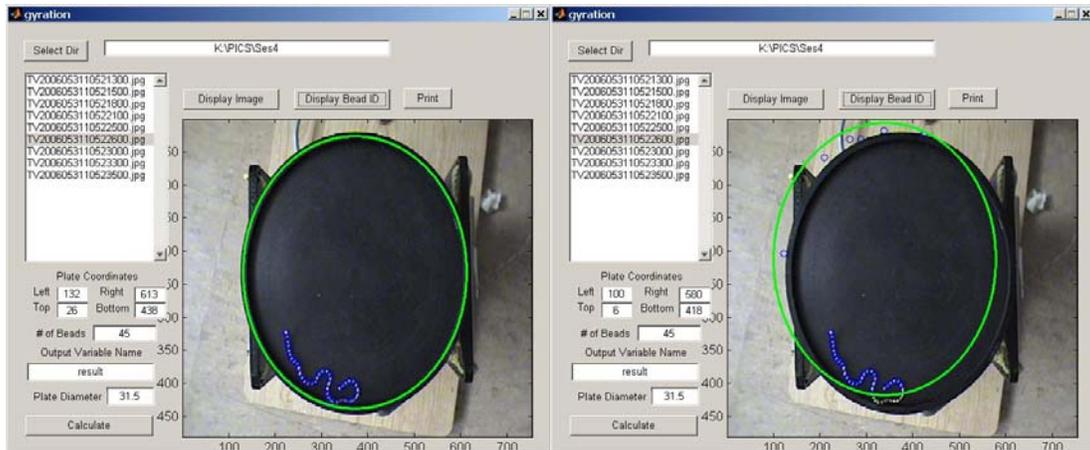


Fig 3. Left – Correctly selected coordinates lead to proper bead identification. Right – Incorrect coordinate selection may lead to identification of improper points and missing of the actually beads.

Printing

The “Print” button will send the panel and everything that appears on it at the moment of clicking to the default printer.

Calculate

When all the data is inserted and everything seems to work fine, press the “Calculate” button in order to perform the radius of gyration calculation. The progress is shown in the main MATLAB window. After the process is complete, a new variable with the name as specified earlier is created in the main MATLAB window containing the results of the calculation.

Working with the Results

The results are given in an Nx6 matrix form where N is the number of images analyzed. In order to work with the results you should have the basic knowledge of working with MATLAB matrices and numbers. Also, in case you want to plot the results, you should be familiar with the plotting function. If you have never worked with MATLAB before, I suggest you go through one of the numerous introductory tutorials you can find on the web. Here are some of them:

http://www.mines.utah.edu/gg_computer_seminar/matlab/matlab.html

<http://www.math.ufl.edu/help/matlab-tutorial/>

<http://www.math.mtu.edu/~msgocken/intro/intro.html>

The results for each picture is given in a separate row, while each column represents a piece of information about the picture:

First Column – The sequential number of the picture (in the order of their taking).

Second Column – The time the picture was taken (the first picture taken at time 0).

Third Column – The radius of gyration of the bead on the picture. Given in the units the diameter of the plate was provided in.

Fourth Column – The error in the calculation, given in the same units.

Fifth Column – Number of beads successfully identified in the picture.

Sixth Column – Threshold used. Shows how high the threshold was set in order to identify the beads. The value should be around 150. If the value is really high (greater than 200) you should take a look at the picture using the “Display Bead ID” button and ensure that everything is identified properly. Otherwise, you might consider on taking the image out of the sequence.

Examples on working with results:

Removing the data on the image number k from an array of N images:

```
>> results(:, :) = [results(1:k-1, :); results(k+1:N, :)]
```

Separating one big array into independent vectors:

```
>> radius_gyr = results(:, 3)
>> radius_gyr_err = results(:, 4)
>> time = results(:, 2)
```

Plotting:

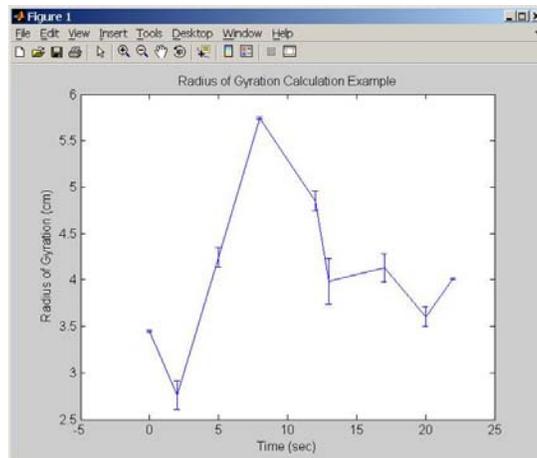
```
>> plot(time, radius_gyr)
or
>> plot(result(:, 2), result(:, 3))
```

In order to show error estimates on the plot, the “errorbar” function is needed. You can get help on using this function by typing in:

```
>> help errorbar
```

Eventually, you might want to have something that looks like this:

```
>> errorbar(result(:, 2), result(:, 3), result(:, 4))
or
>> errorbar(time, radius_gyr, radius_gyr_err)
```



Notes on taking images:

When taking pictures from the camera, it is wise to adjust the lighting in the room in order to maximize the contrast between the plate and the beads. Try turning off some of the lights in the room.

Focus the camera. Blurry beads are likely to remain unidentified.

Also, make sure that the plate is clear of any objects except for the chain. Otherwise, other artifacts may be identified as beads. An example is shown in Fig 5.

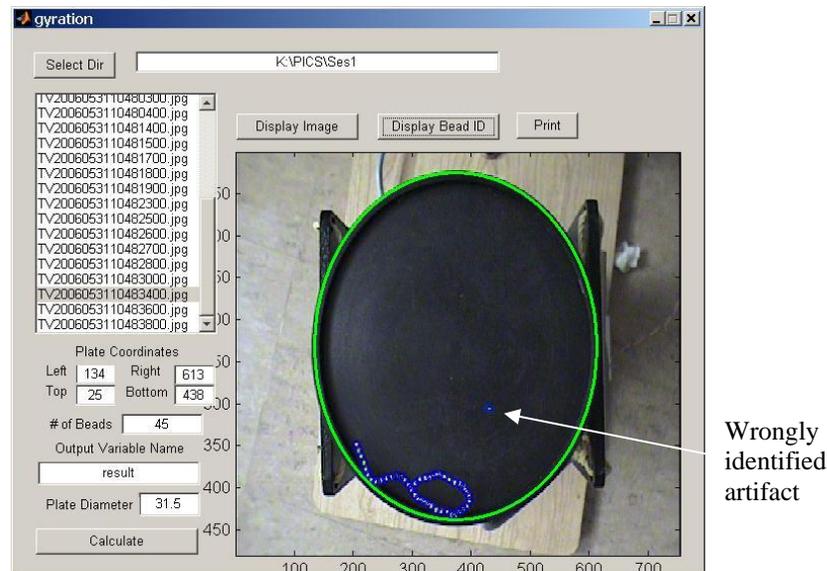


Fig 5. A bright enough artifact on the plate is wrongly identified as a bead.

Warning:

None of the information on the images (position of the bead chain, number of images taken, time intervals, e.t.c.) reflects the correct conditions that should be used for the experiment. Trying to mimic those will evidently lead to incorrect results.

A few notes on the program operation:

You are free to look and modify the .m file of the script. Although I have to admit, the program is most likely not written in the most effective and readable way.

The heart of the program is contained in the "picconv" function, while everything else is mostly concerned with the interface.

The main idea is to apply a threshold to a grayscale image, until the only thing that remains on the image are the beads. Afterwards, the beads (the regions that still remain on the binary image after threshold operation) are identified using "find" command and labeled. In case too many beads are found, the threshold is increased. Default value for the threshold is 150 (where 255 is white on grayscale).

Error Analysis:

The error analysis is done by the program. The program uses a standard error of +/- 1 pixel for CofM of each bead. In case not all the beads are found (this often happens because some beads are blurred as a result of motion), the error for each not found bead is +/- 10 pixels, since this is the mean diameter of the beads used, keeping in mind that the non-identified beads have to be adjacent to the ones identified. The error propagation formula used is a little simplified, providing a good error estimate nevertheless.