Instability of radially spreading extensional flows Experimental analysis

Tal Kozakov

01/10/2022

The project focuses on the analysis of experimental data. The code is based on, and was written originally, for analyzing data obtained during the laboratory experiment described in the great paper: “Instability of radially spreading extensional flows. Part 1. Experimental analysis J. Fluid Mech.(2019),vol. 881,pp.722–738.Cambridge University Press 2019doi:10.1017/jfm.2019.77” by Dr. Roiy Sayag and Dr. M. Grae Worster.

1 Introduction

The displacement of one fluid by another in a quasi-two-dimensional geometry is common to a wide range of natural and industrial systems. In some systems, the evolving interfaces between the two fluids can maintain a smooth circular or planar shape, but in others they can develop fingering instabilities and consequently, evolve complex patterns. A large class of interfacial-stability problems, known as viscous fingering, involves flows that are dominated by shear. By finding a way to correlate, between the number of fingers developed in the fluid pattern, to the wave number that is associated with the dominant Fourier coefficient, we can model and predict, the behavior of various non Newtonian fluids that show the same patterns.

The code was made in attempt to prove the theoretical assumption, that fingering patterns, that emerge when circular interfaces of strain-rate-softening fluids displace less viscous fluids in extensionally dominated flows, can be monitored by tracing the wave number associated with the most dominant (the largest in absolute value) Fourier coefficient.
In other words, by extracting the wave number associated with the most dominant Fourier coefficient, we can track the development of the viscous fingering phenomenon. The phenomenon was witnessed in non Newtonian fluids, involves flows that are dominated by shear, typically due to the traction imposed by confining solid boundaries.

2 Method

In general, the code uses image processing and Fourier decomposition (via the FFT algorithm) of the fluid–fluid interface, to extract the wave number associated with the most dominant (largest in absolute value) Fourier coefficient.

2.1 Image processing

By using the OPENCV, python library, each image in the data set is converted to HSV format, masked and resized as preparation for the "draw contour" process.
By using the cv2.findContours function, a dictionary of possible contours is created. The wanted contour (in our case the one that encapsulates the largest area) is then selected and drawn on a copy of the original image and saved as the main contour for the rest of the analysis.

2.2 Creating axis and function $R(\theta)$

After obtaining the main contour, that is represented by a vector of (x,y) points in the processed image matrix, each point is processed to yield two new vectors. The first vector represents the distance R from the origin. The second vector represents the angle θ made between the line segment from the origin to each point and the positive x-axis.The angle is represented in degrees between 0 and 360. To represent the contour in polar coordinates (R,θ). The Cartesian axis origin is set to the pixel(267,252) in the original image. due to the comfortable nature of the data set, it is the same for all the images in the data. Creation of the vectors that represents the contour in polar coordinates is possible by two methods that are selected by the user.
Option 1
The first option uses the complex plane and the python library Numpy. By creating complex variable Z, that is defined as follows:

$$Z[i] = x[i] + jy[i]$$

and the two vectors of R and θ are created using the attributes that Z possess (size and angle).

$$R[i] = |Z[i]|$$
$$\theta[i] = \angle Z[i]$$

Option 2
The second option uses the definition of the Cartesian-polar transformation:

$$\theta[i] = \arctan\left(\frac{y[i]}{x[i]}\right)$$

$$R[i] = \sqrt{(y[i])^2 + (x[i])^2)}$$

To avoid unwanted behavior of the Fourier decomposition, due to harmonical oscillations which was caused by the tearing of the fluid near the end of the interface at each finger, the maximum radius was set to 60, thus the fingers were truncate to suggest the main shape of the fingers and avoid unwanted harmonics caused by the tearing.

2.3 Interpolating $R(\theta)$ function and "sampling" it in steady sampling frequency

After the creation of the two vectors that represents the contour in polar coordinates (R,θ),the vectors are sorted. A function is then interpolated based on the data received from the two vectors. Using the numpy.interp function. The interpolated function is then sampled by uniform sampling between 0 and 360 with intervals of 45/4096 between each sample.
The size of the interval was selected to best fit the FFT algorithm, that works best when the input vector has a length that is a whole power of 2. The vector is then sorted.

2.4 Fourier decomposition of the data

After creating and sorting the "sample" vector, it ran through the FFT algorithm using the numpy.fft.fft function which then returns the vector of Fourier coefficient associated with the given data (the sampled interpolated function). Each index in the coefficient vector, represents the wave number that corresponds to the coefficient. Due to the Nyquist–Shannon sampling theorem, the maximum wave number is N/2, where N is the number of samples in the vector that is given to the algorithm.

3 Images and plots

Data analysis of liquid with 5 fingers pattern

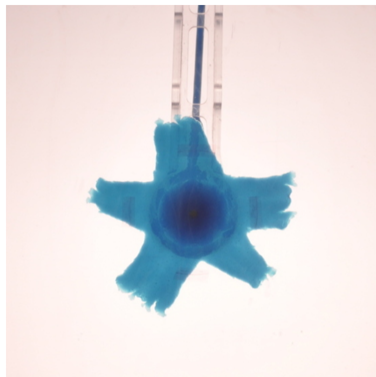
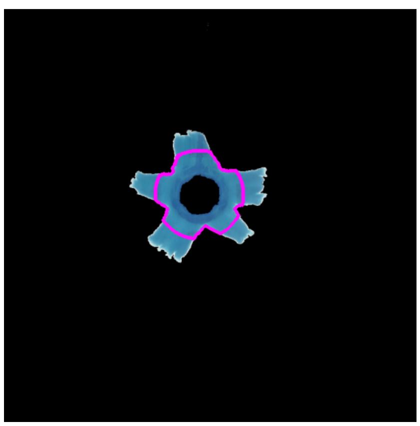


Figure 1: Original image with 5 fingers:



(a) Contour with max radius 60

(b) Contour with max radius 80

Figure 2: Contours of 5 fingers patterns with different maximum radius

.

Figure 3: Plots for analysis of 5 fingers patterns with different maximum radius

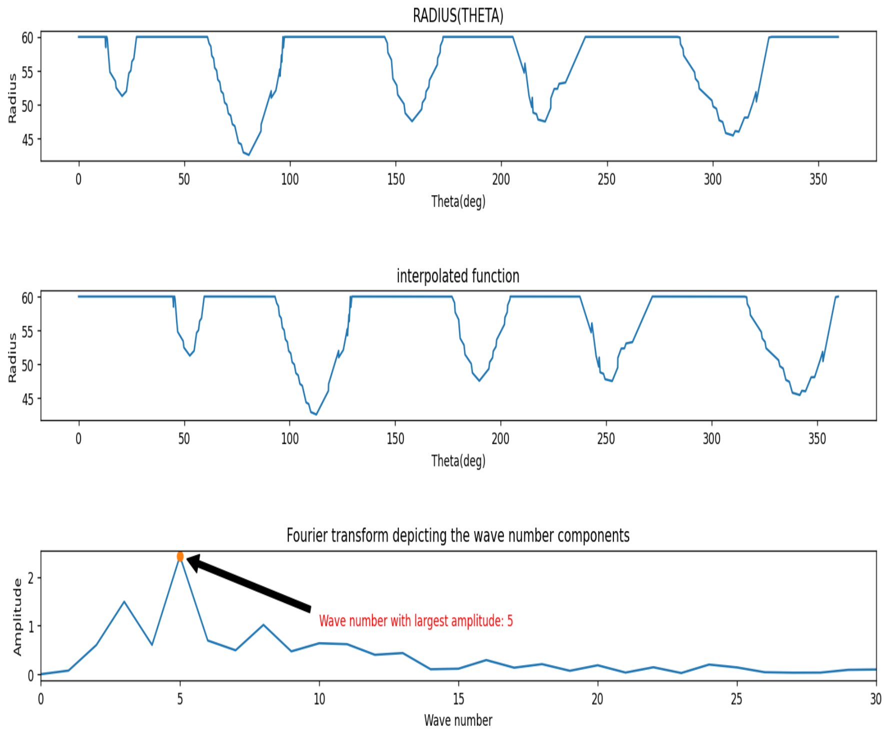


Figure 4: Plot with max radius 60

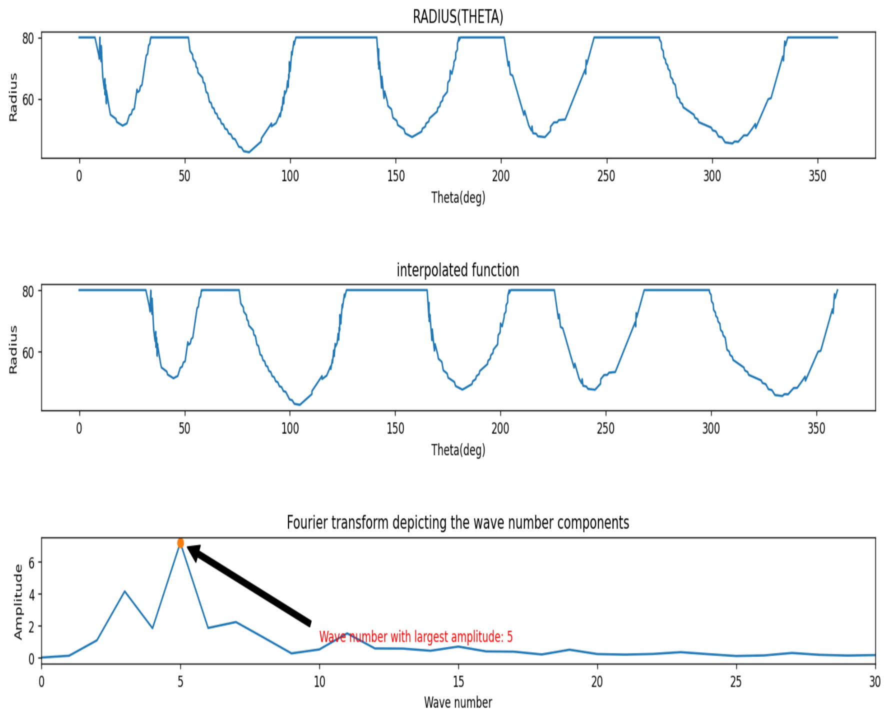


Figure 5: Plot with max radius 80

.

Data analysis of liquid with 3 fingers pattern

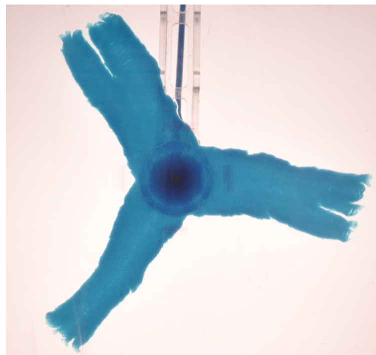
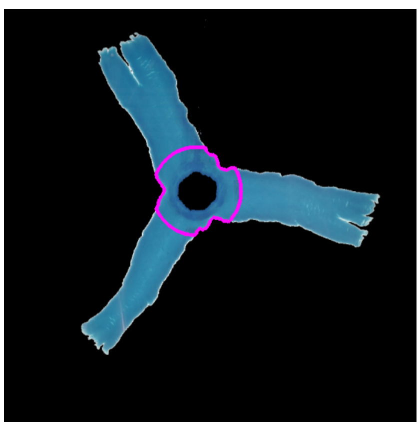
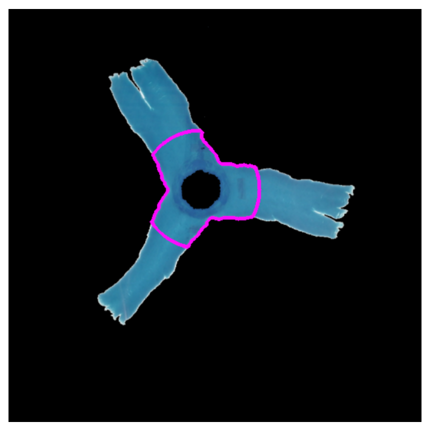


Figure 6: Original image with 3 fingers:



(a) Contour with max radius 60



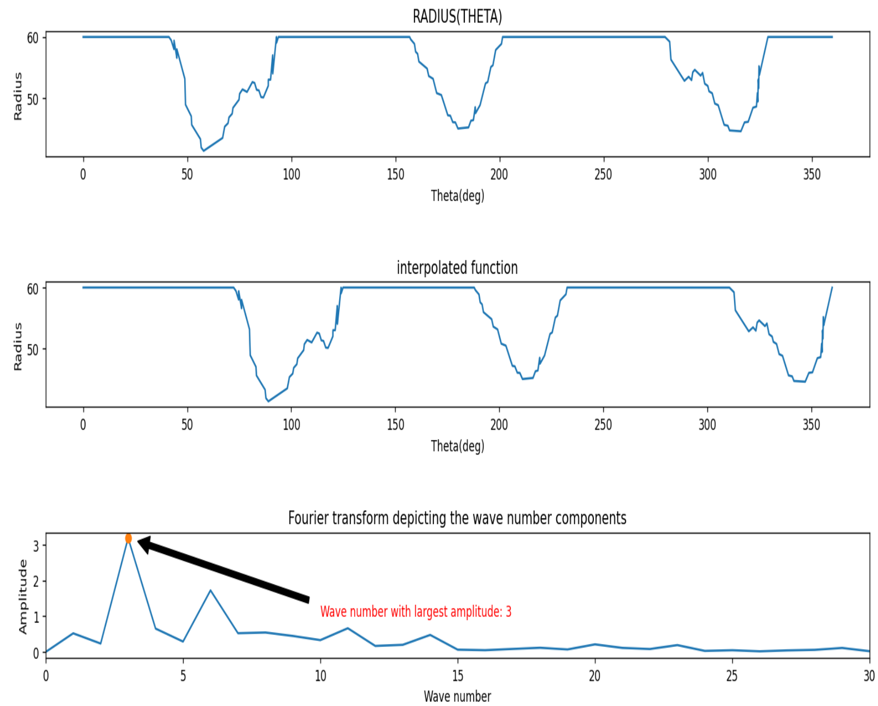
(b) Contour with max radius 80

Figure 7: Contours of 5 fingers patterns with different maximum radius

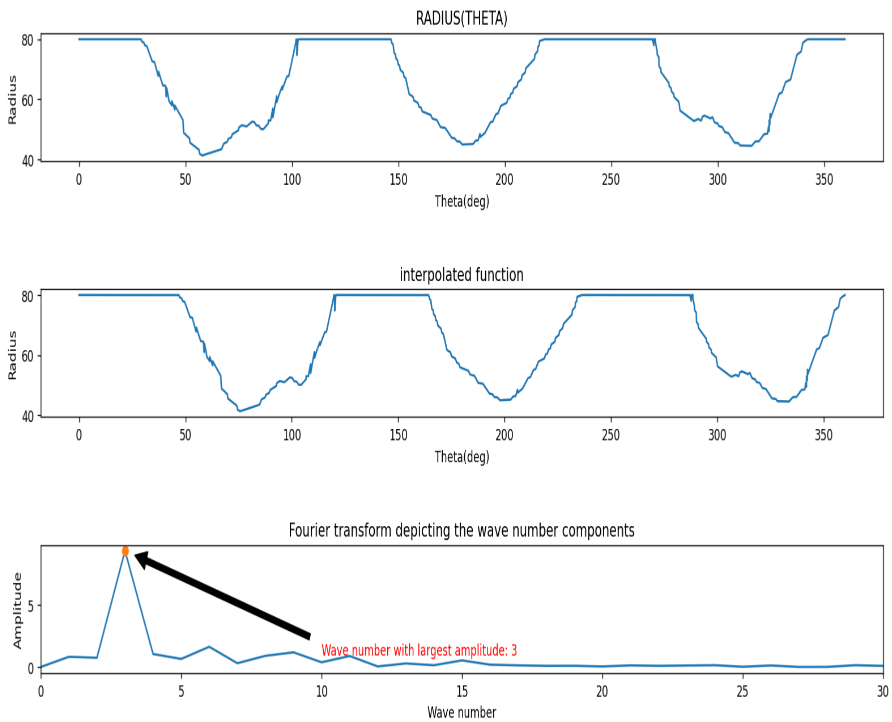
.

.

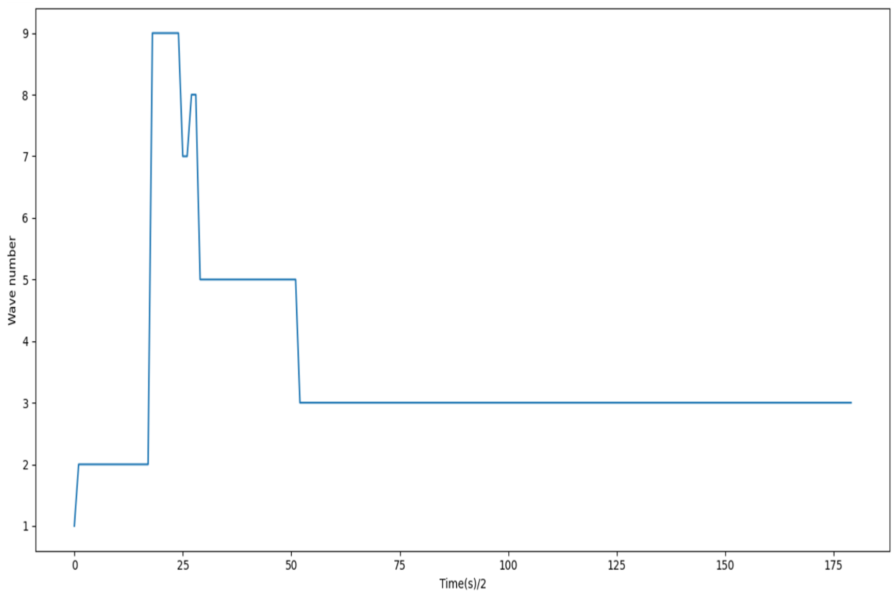
Figure 8: Plots for analysis of 3 fingers patterns with different maximum radius



(a) Plot with max radius 60



(a) Plot with max radius 80



(a) Development of the wave number over time (continues

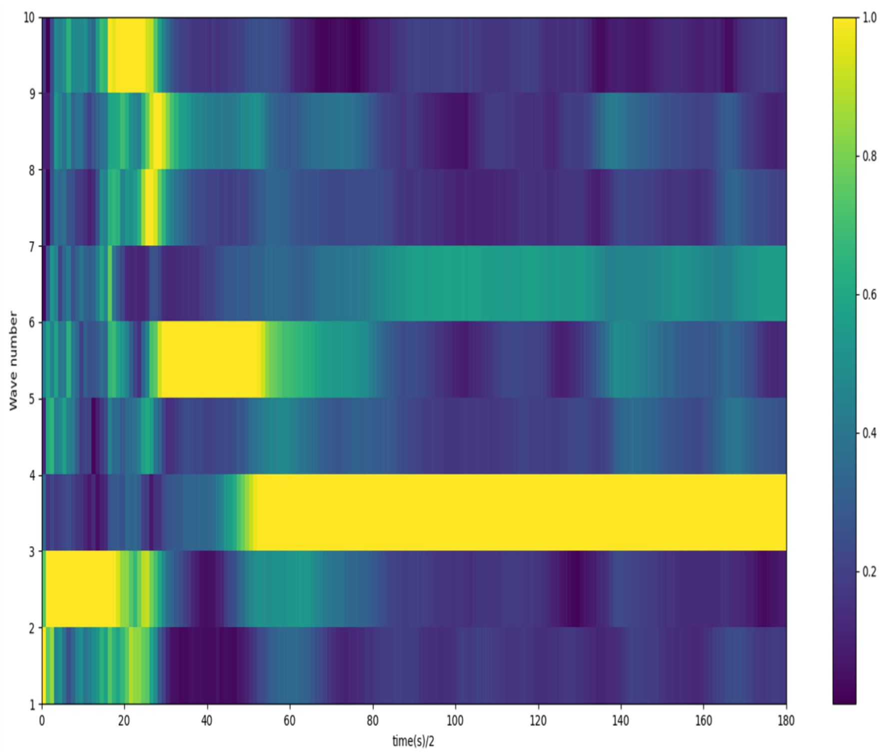


Figure 10: Normalized heat map of amplitude on K(t) plain

4 Results

In this project I have managed to show the correlation between the number of fingers developed in the fluid and the wave number associated with the dominant Fourier coefficient. After roughly 100 images from the data (which corresponds to 200 seconds of experiment), we can see that the link between the wave number and the number of fingers, has stabilized. Meaning, any change in the number of fingers developed, will lead to a change in the wave number.