LTHN PIV
Tutorial

Abril 2020

1 Introduction

This work aims the development of a program to obtain the particle velocity map in a fluid flow. This code was made to be applied to digital images obtained from the PIV method (Particle Image Velocimetry). These images are records of the positions of the particles in the flow at different time intervals.

In addition, the program was developed with object-oriented programming. This allows more organization of the code functions and greater versatility, as it makes it possible to add new pre-processing and processing methods without having to change the other parts of the code.

A graphic interface was also implemented aiming a greater usability for the program, since this allows the user to choose the desired parameters and methods.

In addition to the features mentioned, the code was written in python 3 and it is free and open source software.

2 The Code

Initially, to find the particle velocity map, it is necessary to determine the displacement map. From this and the time between captures of the images, it is possible to determine the desired velocity map.

One of the input parameters will be a set of images, which will be processed in pairs. Therefore, a displacement map is calculated for each pair of images. At the end of the code, all the maps found are averaged.

The code was divided into two parts: pre-processing and processing

2.1 Pre-processing

Pre-processing improves image quality and makes corrections.

- Sobel Filter;
- Background removal;
- Brightness homogenization.

2.2 Processing

After the images are treated in the pre-processing, they are divided into small parts called interrogation windows. Depending on how the interrogation windows will be used and which operations will be applied to them, a new processing method is determined. Among the methods developed in the program, are the Normal method and the Multigrid method.

- Normal method: The interrogation windows are moved with a constant displacement both horizontally and vertically, and they do not vary in size throughout the process.
- Multigrid method: It is an iterative procedure, in which the size of the interrogation windows is reduced between one iteration of the code and the other. And, in order not to have the result affected by this reduction, the interrogation windows are shifted in the images according to the displacement map found in the previous iteration.

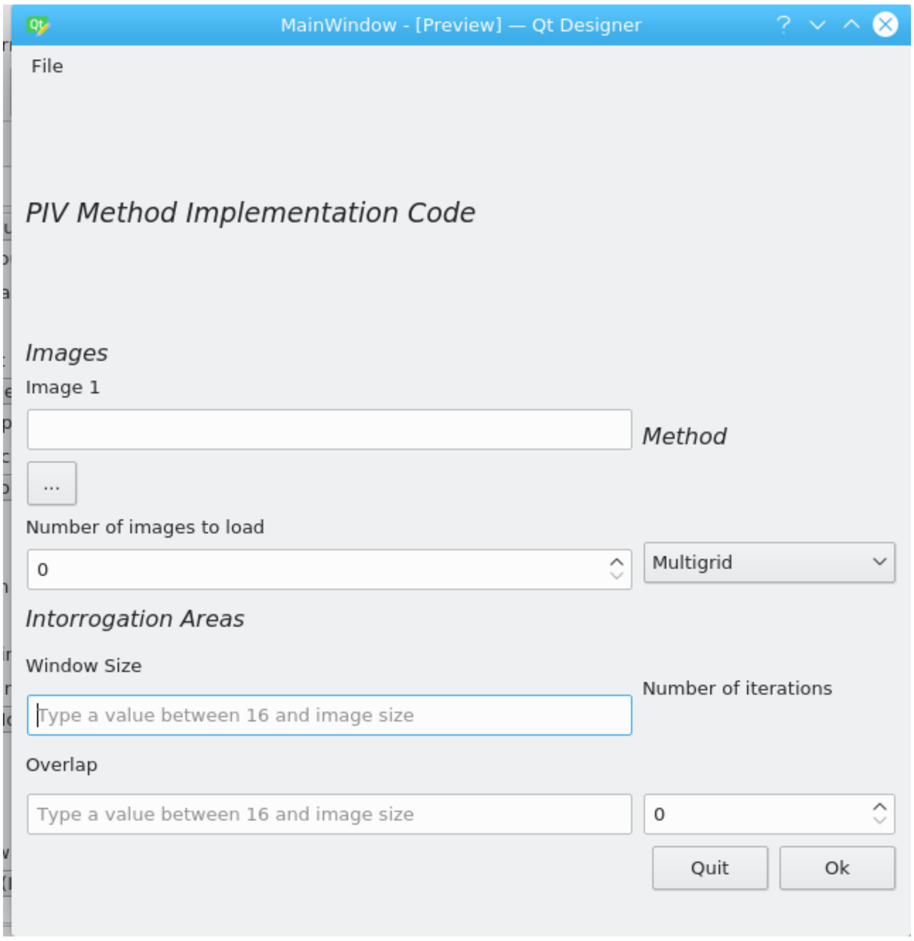
3 How to use

In order to obtain the desired velocity map, it must be ensured that:

- only the LTHN_PIV.py file have be executed.
- set of images must be in the same directory.

- The file names of the images in the set of images must be in the format image_set_number. The last part of the name of the image is the number of the image in the image set. So, for example, the first image must be named as image_set_1.

By doing this, the graphical interface will open and it will be possible to enter the code input parameters:


MainWindow - [Preview] — Qt Designer
File
PIV Method Implementation Code
Images
Image 1
...
Method
Interrogation Areas
Window Size
Type a value between 16 and image size
Overlap
Type a value between 16 and image size
Quit
Ok

Number of images to load
0

Multigrid

Number of iterations
0

3.1 Input parameters

- Image 1: From the button that has 3 points and is below the Image 1 label, it is possible to select the first image in the set of images. With that, the interface will take the string with the address of the first image in the image directory.

- Number of images to load label: Number of images in the directory the user wants to process.

- Window Size label: Size of interrogation windows used in processing.

- Overlap label: How much overlapping interrogation windows should be applied.
- Method label: If it is desired to use Normal or Multigrid methods.
- Number of iterations label: If the Multigrid method is chosen, how many iterations are desired.

After entering the input parameters, the Ok button must be pressed to start processing the images.

3.2 Output parameters

The code outputs are the components on the x and y axis of the found velocity map, which have the same dimensions as the processed images.

- dpx_resultado_multigrid.csv: a .csv file containing the velocity map on the x axis, if the Multigrid method was choosen.
- dpx_resultado_normal.csv: a .csv file containing the velocity map on the x axis, if the Normal method was choosen.
- dpy_resultado_multigrid.csv: a .csv file containing the velocity map on the y axis, if the Multigrid method was choosen.
- dpy_resultado_normal.csv: a .csv file containing the velocity map on the y axis, if the Normal method was choosen.

To obtain the map plotted from the output csv files, simply run the file plotter.py.

3.3 Comments

- To close the interface, use the Quit button.
- If values outside the possible range for processing are entered for the Window size and Overlap parameters, a warning will appear on the screen saying that other values must be entered.