

POLITECNICO DI TORINO

VON KARMAN INSTITUTE FOR FLUID DYNAMICS

Master Course
In Aerospace Engineering

Master thesis

Implementation of Constrained Radial Basis Functions for PIV fields
regression



Supervisors:

Prof. Sandra Pieraccini

Prof. Miguel Alfonso Mendez

Candidate:

Federica Spoto

Academic Year 2020/2021

Acknowledgements

I would like to thank my supervisors. Prof. Mendez, for his positivity and support despite the obstacles and Prof. Pieraccini, always helpful and with a goodness like few others.

Thanks to my short training program colleagues, for the inspiring atmosphere and the best brainstorming ever.

This master thesis was done in cooperation with the Von Karman Institute for Fluid Dynamics (VKI).

Abstract

In recent years, Particle Image Velocimetry (PIV), a non-intrusive optical technique that allows the evaluation of instantaneous velocity fields, has become one of the most widely used experimental fluid dynamics techniques. Due to the noise contained in the images and the difficult task to build the computational mesh to solve the Partial Differential Equations, especially near the boundaries of the problem, the post processing of the images is leaving the Computational Fluid Dynamics numerical techniques to provide space for Machine Learning tools.

This master thesis consists of an experimental part and a numerical part. Two test cases are considered for the study: a simple case, a free vortex flow, and an experimental case, a channel flow.

First of all, PIV images of the test cases are collected, in the first case generated synthetically with an algorithm and in the last case experimentally detected in Water Tunnel. Using Openpiv, a Python package for PIV image analysis, an algorithm calculates the PIV-based velocity fields. At this point, a Machine Learning method is implemented in Python. The algorithm performs a velocity regression that depends on a set of weights, via constrained optimization techniques, where the constraints include the boundary and initial conditions, trying also some factorization methods to improve the calculation time.

In this work, an RBF approximation of PIV fields is found, implementing a compact algorithm with which providing in input PIV images, RBF Approximation is obtained in output. Once having this, it will be simple to manipulate the function, becoming more accessible for an industrial context.

Keywords

Particle Image Velocimetry (PIV), Python, OpenPIV, Machine Learning, Radial Basis Functions (RBF).

Contents

Acknowledgements.....	0
Abstract.....	1
Keywords.....	1
List of Figures.....	4
List of Abbreviations.....	6
1. Introduction.....	8
1.1 Motivations and objectives.....	8
1.2 Methodology and framework.....	9
2. PIV evaluation.....	11
2.1 PIV concept.....	11
2.2 Definition of PIV and general components.....	14
2.3.1 Particles.....	15
2.3.2 Images.....	15
2.3.3 Velocity and derived quantities.....	16
2.3.4 Errors.....	17
2.3 PIV Campaign.....	19
2.3.1 Experimental set-up.....	19
2.4 PIV Synthetic Images.....	21
3. Function approximation via RBF.....	22

3.2	Introduction to Machine Learning	22
3.3	Advantages.....	24
3.4	Linear Regression via Nonlinear Basis.....	25
3.4.1	Finite Basis Functions and Linear Algebra Solution	25
3.4.2	Radial Basis Functions (RBFs).....	27
3.4.3	Optimization and Regularization	28
3.4.4	Solution for Tikhonov Regularization	29
3.5	Constrained Regression via RBFs	30
4.	Test cases	32
4.1	Sinusoidal distribution of points	32
4.2	Free Vortex Flow	33
4.3	Experimental test case, channel flow flowing from/to the bottom.....	34
5.	Python Implementation and Results	37
5.1	1D RBF Constrained Interpolation	38
5.2	2D RBF Free Vortex Approximation	40
5.3	2D RBF Channel Flow Approximation.....	45
6.	Conclusion and Future Work	53
	Bibliography	56

List of Figures

Figure 2.1: PIV process	11
Figure 2.2: Correlation map peak of an interrogation window.....	16
Figure 2.3: Average velocity	17
Figure 2.4: Error sources	18
Figure 2.5: Experimental set-up.....	19
Figure 2.6: Flow over a cube	21
Figure 2.7: Flow in 90 degree corner.....	21
Figure 2.8: Flow past a cylinder	21
Figure 3.1: Machine learning algorithms categories	23
Figure 3.2: Hybrid case of shifted gaussians plus a straight line.....	27
Figure 3.3: RBF approximation with different regularizations	29
Figure 4.1: Sinusoidal distribution of points.....	33
Figure 4.2: PIV Synthetic Images of a free vortex flow	33
Figure 4.3 Theoric free vortex flow.....	34
Figure 4.4: Image pair and motion field	36
Figure 5.1: 1D Interpolation Flowchart	38
Figure 5.2: 1D Radial Basis Functions	39
Figure 5.3: Constrained Interpolation.....	39
Figure 5.4: Free Vortex Flow RBF Approximation Flowchart	40
Figure 5.5: Vortex. PIV velocity field	41

Figure 5.6: Vortex. Theoretical Velocity Field	41
Figure 5.7: Vortex. Error between PIV and Theory	42
Figure 5.8: Vortex. RBF Approximation.....	43
Figure 5.9: Vortex. Error between Approximation and Theoretical	43
Figure 5.10: Vortex. Error between Approximation and PIV	44
Figure 5.11: Channel Flow. PIV Velocity Field.....	46
Figure 5.12: Channel Flow. RBF Approximation	47
Figure 5.13: Channel Flow. Error between PIV and Approximation.....	48
Figure 5.14: Best lambda for Ridge Regression	49
Figure 5.15: Best Lambda for LASSO Regression.....	49
Figure 5.16: Channel Flow. Best lambda Approximation with LASSO	51
Figure 5.17: Channel Flow. Error between LASSO and PIV.....	52

List of Abbreviations

PIV	Particle Image Velocimetry
RBF	Radial Basis Functions
PDE	Partial Differential Equation
CFD	Computational Fluid Dynamics
VKI	Von Karman Institute for Fluid Dynamics
PC	Personal Computer

Chapter 1

1. Introduction

1.1 Motivations and objectives

In nature, there are very complex motion fields with strong three-dimensional characteristics for which the magnitude and the direction of the velocity are locally unknown. For the measurement of the velocity vector, various techniques are available, from the simplest to the most complex.

There are different types of probes, the choice depends on the type of motion field to be analyzed. From the measurements of the probe, it is possible to evaluate the magnitude of the velocity, but not always the direction.

The velocity vector is fully defined with magnitude and direction, that is detected from two angles in two perpendicular planes.

The most used techniques for the measurement of the velocity vector are the pneumatic technique, in particular the Pitot probe, which measures the average velocity, and the hot wire probe, which measures the instantaneous vector. Both measurement techniques are intrusive and punctual, therefore they slightly modify the field of motion and it takes a long time to measure the entire velocity field [1].

In recent years, the PIV technique has become more and more widespread because it is a non-intrusive optical technique that allows the evaluation of instantaneous plans or volumes of velocity fields.

This work focuses on PIV, which is treated both numerically and experimentally.

The objective of this thesis is to observe how the approximation of a function works for an experimental case, in terms of accuracy, for different types of constrained regression and for different penalization term values. PIV measurements require expensive equipment and data processing. Moreover the measurements must be done under certain conditions, often requiring controlled pressure and temperature

environments, and the detected images are not always accurate, containing noise and uncertainties. For complicated geometries these problems are amplified. Another problem is related to the 2D flow assumption of 3D flows, in which errors due to these assumptions increase with Reynolds number [4].

In an industrial context, a trade-off between cost and performance is searched, PIV is therefore almost not reachable, because the geometry of prototypes produced by companies is very complicated and using CFD classic methods, sensitive to noise, requires to build a mesh, very difficult task near the boundaries, but also the grid changes with the problem, costing too much time.

To overcome the meshing problem, interest in Machine Learning tools is growing, in particular in meshless methods that use regressions based on function approximation to solve the Partial Differential Equations (PDEs).

The idea, in this work, is to find an RBF approximation of PIV fields, implementing a compact algorithm with which providing in input PIV images, RBF Approximation is obtained in output. Once having this, it will be simple to manipulate the function, decreasing noise, increasing resolution and integrating PDEs without the necessity of a mesh, being able to derive other fluid dynamics quantities.

1.2 Methodology and framework

This work starts from the experimental part, consisting of a PIV Campaign in Water Tunnel, to collect PIV experimental images of a channel flow. The numerical part starts with a PIV evaluation algorithm, using OpenPIV [2], a Python package for PIV image analysis, made for pre-processing the images, improving the quality of the images and removing the background noise, processing the images, correlating the images to find the displacement of the particles and, finally, post-processing them to find the velocity field.

After this, an overview of the machine learning landscape and some machine learning tools, Regression and Optimization problems in fluid mechanics, are introduced, ending up with RBF approximation and constrained regression, showing some test cases applications. Then, some advice to improve the

computational cost and some considerations about the kind of regression and the choice of the best hyperparameter are discussed.

Finally, the last chapter concludes the work and presents the perspectives.

Chapter 2

2. PIV evaluation

In this chapter, PIV technique will be introduced. The first part presents a typical set-up and the basics of the technique, the last part focuses on the PIV campaign taken place at the Von Karman Institute (VKI).

2.1 PIV concept

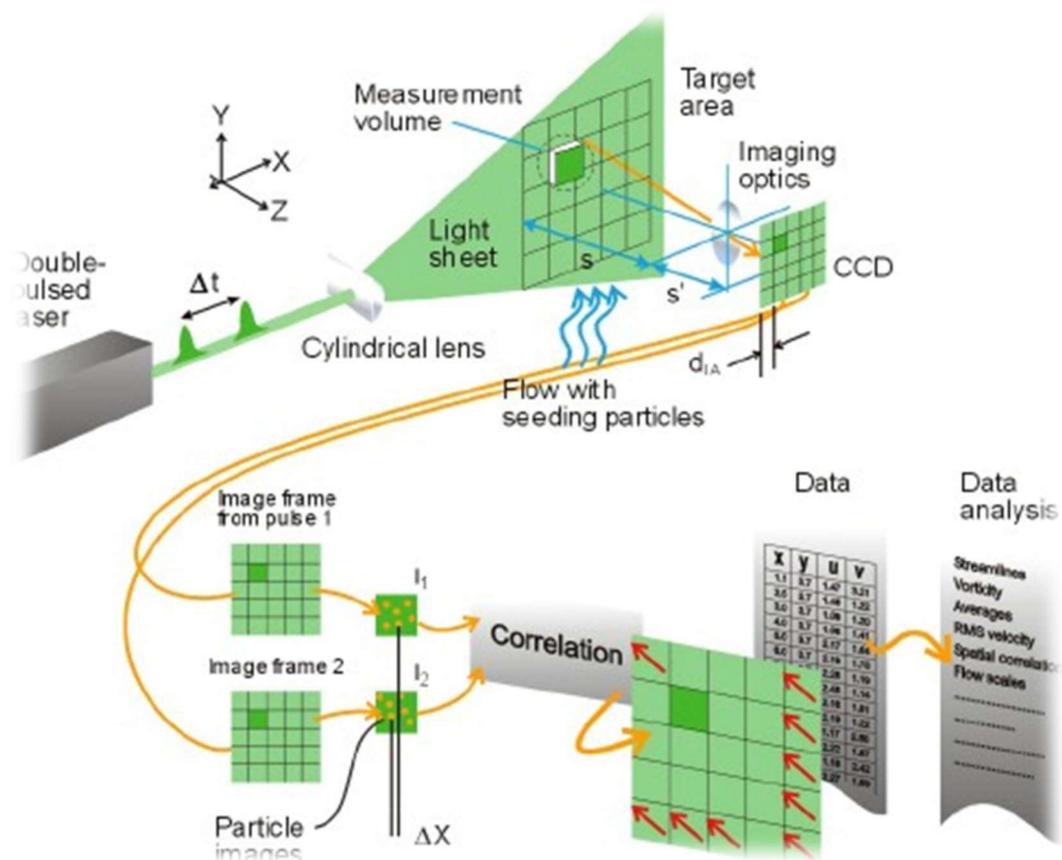


Figure 2.1: PIV process, [15]

PIV is the acronym for Particle Image Velocimetry. Particles are indeed used to seed and track the flow; the word “Image” refers to the fact that shots of the flow are made and “Velocimetry” is the measurement of the velocity of fluids [1].

In Figure 2.1, it is shown a typical general PIV experimental setup: the laser passes through different optical lenses and generates a green light sheet; this light laser sheet illuminates the model hit by the flow that is seeded with the particles. The laser emits two pulses, the second after a short time with respect to the first and a camera takes the shots of the two pulses of the laser. Instead, in the time resolved PIV version, the laser emits continuously. Capturing images take place in a dark environment.

This technique could be performed through two different types of cameras: the Charge Coupled Device (CCD), which is able to capture at a sampling frequency of about 100 Hz, and the Metal-Oxide Semiconductor (CMOS), with a higher resolution, capturing at almost 1 kHz. A PC with a dedicated software manages the synchronism system, the acquisition of PIV images and the recording on its memory.

The images could be single-exposed, where each image is illuminated only once by the laser, or multi-exposed, illuminated more than once.

In the first case a cross-correlation is made for each couple of frames A and B and this requires a higher frequency of sampling, as from each pair of PIV images the displacement of the particles is calculated, which means in obtaining a single instantaneous velocity field. Due to this fact, a CMOS camera is needed.

In the second case an auto-correlation is computed, a lower frequency of sampling is also acceptable, hence it is possible to use a CCM camera. Auto-correlation or cross-correlation algorithms allow the calculation of displacement between two subsequent PIV images.

In Figure 2.1 it is possible to observe the physical plane and the optical plane.

The whole PIV image is divided into interrogation windows of equal size.

The choice of the size of the interrogation windows is a trade-off between being able to capture the local gradients and keeping the flow uniform. A good compromise is represented by windows of 32x32 pixels.

Each window contains a certain number of particles, approximately 10, and the most probable displacement of these particles, accordingly one velocity vector is calculated on this area.

The image is defined by a distribution of gray levels, hence a function that describes the intensity of the gray level in the plane of the frame can be defined

$$I = F(x, y).$$

By dividing the interrogation windows into pixels p_x and p_y , the discrete auto-correlation function is defined as:

$$R(r_x, r_y) = \sum_{p_x=1}^{\Delta p_x} \sum_{p_y=1}^{\Delta p_y} F(p_x, p_y) F(p_x + r_x, p_y + r_y) \Delta p_x \Delta p_y$$

where r_x and r_y are the pairs of possible displacements in the x and y directions, the product $\Delta p_x \Delta p_y$ is the size of the interrogation window, $F(p_x + r_x, p_y + r_y)$ is the luminous intensity function evaluated after the displacement of the particles.

The cross-correlation function is defined as:

$$R(r_x, r_y) = \sum_{p_x=1}^{\Delta p_x} \sum_{p_y=1}^{\Delta p_y} F_i(p_x, p_y) F_{i+1}(p_x + r_x, p_y + r_y) \Delta p_x \Delta p_y$$

where $F_i(p_x, p_y)$ is the luminous intensity function at the t_i instant associated with frame A, and $F_{i+1}(p_x + r_x, p_y + r_y)$ is the same function at the $t_i + \Delta t$ instant associated with frame B.

The interrogation window is usually a square, indeed $\Delta p_x = \Delta p_y$.

The cross-correlation is applied to the image pairs collected for this thesis work.

The technique identifies the velocity fields by measuring the displacement of particles which the flow is seeded with. The evaluation of speed at each point of the field image is directly derived from the definition of speed:

$$v = \frac{\Delta s(x, y)}{\Delta t}$$

where the time interval Δt is imposed on the system by the operator having an estimation of the speeds involved. The unknown is the displacement Δs of the particles following the flow field, the velocity measured is therefore the one of the particles that, under precise requirements, can accurately follow the local behavior of the current.

This technique allows to distinguish the velocity vector in its entirety: magnitude, orientation, sense in every point of the captured field.

2.2 Definition of PIV and general components

PIV is an anemometric and experimental fluid dynamics optic non-intrusive measurement technique that finds application mostly in research centres. It is defined anemometric as it measures the velocity, providing qualitative and quantitative information on the instantaneous velocity fields of fluid flows. On the other hand, it is non-intrusive because between the particles and the laser light sheet no perturbations are introduced inside the test section.

The sensor of this technique is the particle that in-seminates and follows the flow. A Laser, that produces a high energy pulsed light, illuminates a plane of the flow and consequentially the seeding particles.

The illuminated plane can be shot at two subsequent times by one or more digital Cameras. A synchronizer regulates the double pulse laser and the double shot camera, the Acquisition System records all the images and a Processing System, through algorithms, converts the images into velocity fields.

Different PIV configurations are available for measuring instantaneous fields in the measurement plane, depending on the number of measurable velocity components:

- Standard PIV (2C2D). It can measure two components of the velocity vector in the cartesian plane, u , v , by placing one camera with its recording plane parallel to the laser sheet and capturing the parallel and perpendicular components of the plane for each point of the plane.

Evaluation of the velocity vector $V(x, y, t) = (u, v)$.

- Stereoscopic PIV (3C2D) resolves all of the three components of the velocity field u , v , w , but two cameras are needed, placed with an angle from the laser sheet. This configuration is a more complete version than the first, because it also allows to measure the third component of velocity, perpendicular to the captured plane.

The velocity vector $V(x, y, t) = (u, v, w)$ is evaluated.

- Tomographic PIV (3C3D) measures three components of the velocity field, u , v , w , using more cameras.

Through this configuration, it is possible to measure an entire "instant volume", evaluating the velocity vector $V(x, y, z, t) = (u, v, w)$ for each point of the volume.

With the PIV technique the velocity fields are perfectly described in space and time, but between a shot and another a short time interval passes.

In this work, in order to detect the field of motion shots, the standard PIV is used.

2.3.1 Particles

An effective Insemination System is needed. Indeed, the velocity measured in the PIV technique is the one of the particles the fluid is seeded with. Due to this aspect, the particles must have particular requirements to perfectly follow under certain conditions the behavior of the flow field. The aim of this technique is to measure the velocity of the flow, but what is actually measured is the velocity of the particles, because the particles have a scattering effect when they are illuminated by the laser that makes the flow visible. The seeding must be as uniform as possible, considering that the higher are the velocity gradients of the flow, the more difficult it is to reach a good level of uniform distribution of the particles.



Figure 2.1: Particles to inseminate water

The particles also have to be sufficiently small in order to follow the fluid motion properly without significant modifications of the properties of the flow.

To be sure that the particles adapt to the flow, following it, they must satisfy the requirement that the density ratio between the density of the seeding particle and the density of the flow particle has to be around one,

$$\frac{\rho_p}{\rho_f} \sim 1$$

so the particles should have similar densities. However, sometimes a perfect match between the two densities is not possible.

2.3.2 Images

In good PIV images, particles have 2-3 pixels in diameter.

The images often contain background noise because of illumination disuniformity or reflection, so the images need to be pre-processed to remove it.

As it is shown in picture 2.2, the two frames are divided into small areas called interrogation windows.

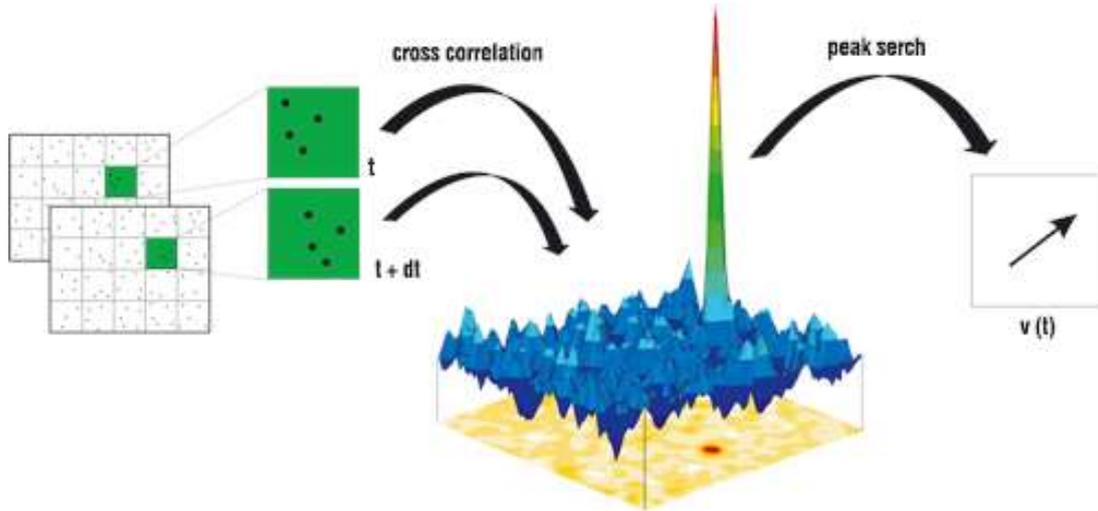


Figure 2.2: Correlation map of an interrogation window[18]

The cross-correlation to the corresponding interrogation windows from both frames allows to obtain the correlation map for each interrogation window, which has a peak located in the most probable displacement.

Knowing which frame was taken first, the cross-correlation between two interrogation windows of two frames provides exact information about the direction of a particle motion.

Object plane (real) and image plane (camera) are correlated by a linear mapping defined by the magnification factor M , that makes the conversion from optical scale [pixel] to real scale [mm]. Knowing pixel size and magnification factor, it is possible the conversion to a real scale:

$$\Delta s_{real} = M \Delta s_{PIV}$$

where Δs_{PIV} is the displacement of the particle captured in PIV image and Δs_{real} is the displacement of the same particle in the real plane.

2.3.3 Velocity and derived quantities

Knowing the size of a pixel and the time separation between the two images, the velocity can be calculated.

These calculations extract the velocity from the correlation map and for each interrogation window one velocity vector is obtained.

PIV results depend on the temporal separation between the two image frames: higher separations (higher displacement) allow to investigate higher dynamic ranges, but the correlation peak decreases.

Then, the particle images are processed through a cross-correlation based algorithm to get velocity field maps.

At each intermediate step of the iteration process a new velocity vector is computed, so data validation criteria must be applied.

A data sheet is obtained from the correlation and by performing the data analysis, raw data, calculated data, derived data and statistics can be obtained.

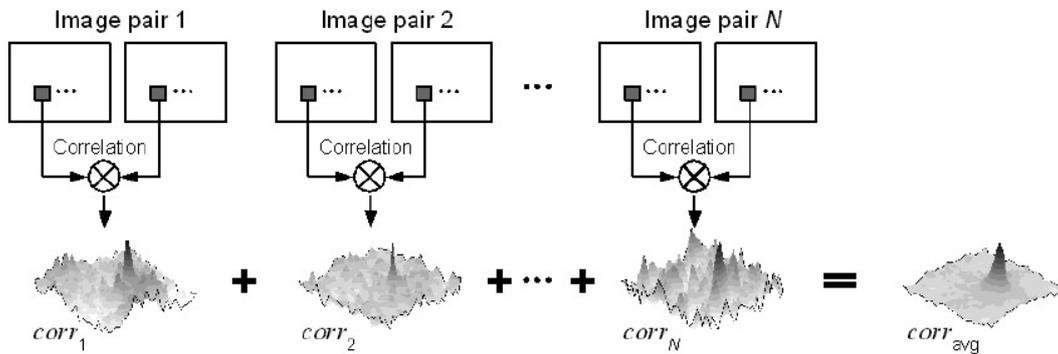


Figure 2.3: Average velocity [20]

Indeed, it is possible to derive pressure forces (like Lift and Drag) and acoustic emissions, to calculate the averages (Figure 2.3), the RMS, the vorticity, etcetera.

2.3.4 Errors

The main problems that occur when performing cross-correlation in PIV experiments are:

- **Edges.** Particle pairs near edges of the interrogation windows contribute less to correlation. This is why overlap between adjacent interrogation windows can be allowed, but too much overlap causes oversampling.
- **Background noise.** High-pass filtering can be used to eliminate the background noise. It can slightly affect the data.
- **Random correlations.** Random correlations originate from the correlation of different particles,
- **In-plane particle loss.** There are incomplete particle pairs due to the in-plane entering and leaving particles in the interrogation area, as they enter

and leave the domain (before the second frame) due to the nearness to the boundary.

Windows interrogation must be sufficiently large to include in the cross-correlation process a relatively large number of particles.

In Figure 2.4 an overview of the possible errors that can occur in the overall PIV process is shown.

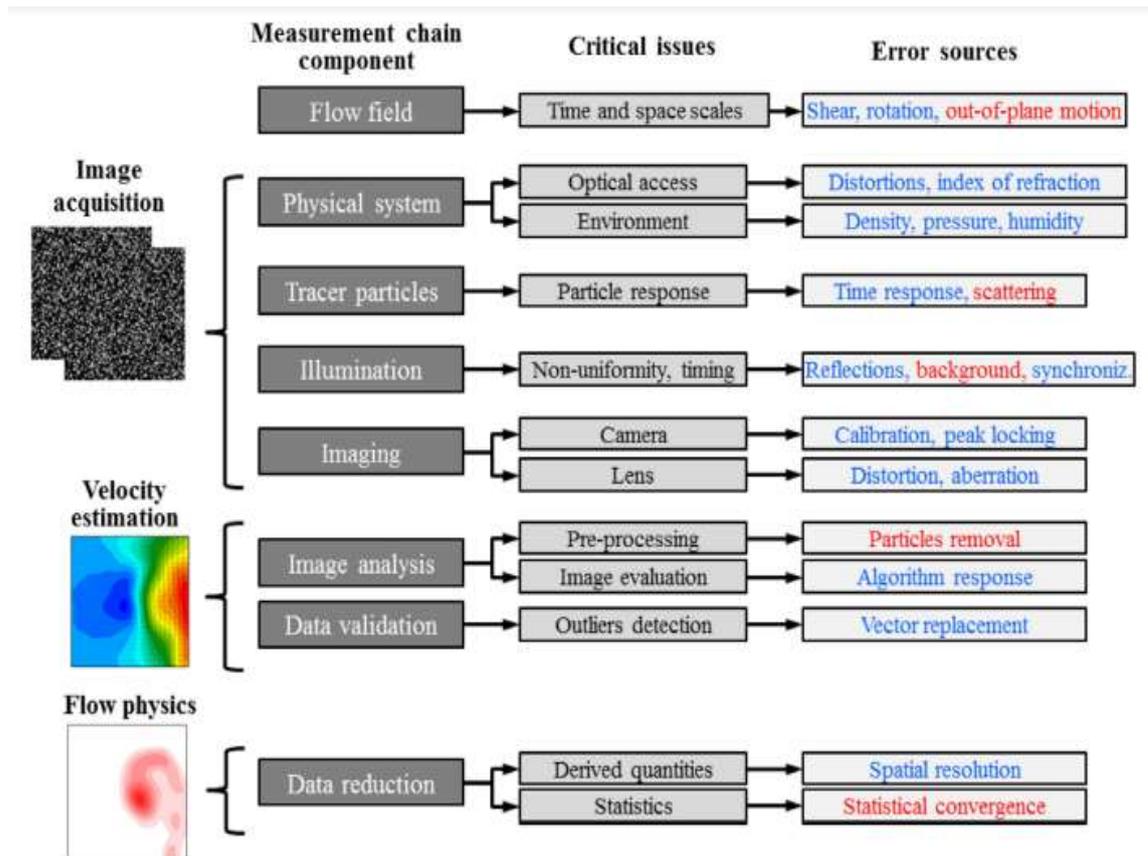


Figure 2.4: Error sources, [17]

2.3 PIV Campaign

A schematic representation of the experimental set-up for the PIV Campaign is shown in Figure 2.5.

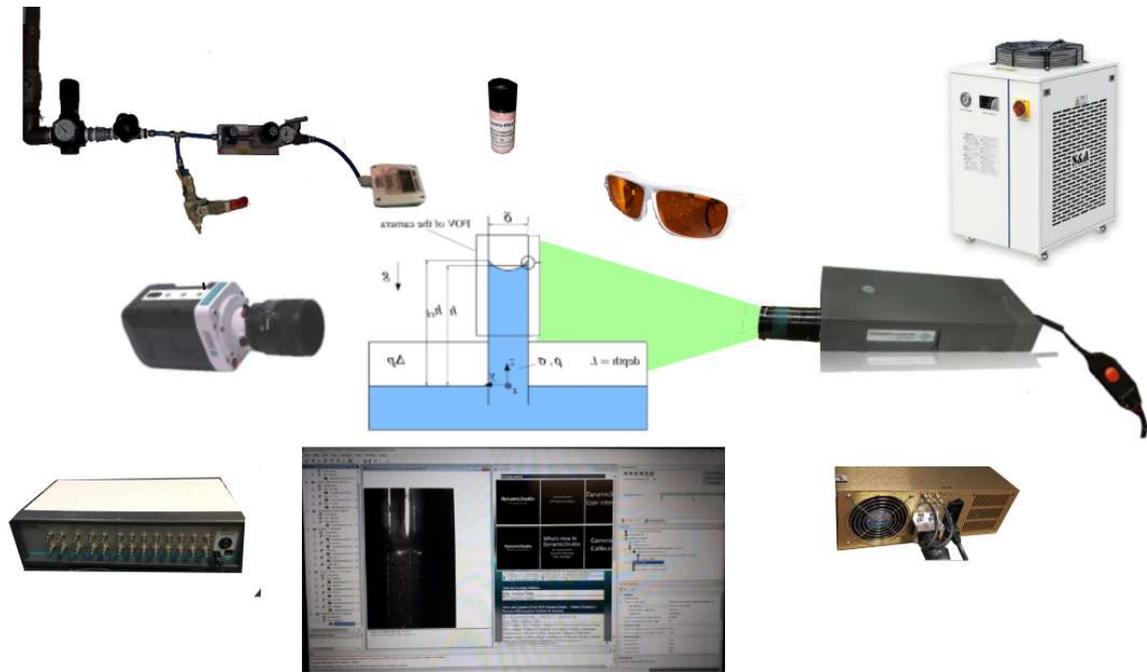


Figure 2.5: Experimental set-up

The set up is made up of various components and each of them contributes to the measurement process. Starting from the upper left image, clockwise are shown: the pressure regulation system, the seeding particles, the laser protection glasses, the cooler, the laser optics, the laser controller, the PC using Labview software, the synchronizer, the CMOS camera, and in the middle the reference model in the test section.

2.3.1 Experimental set-up

- Test Model. Tank with water and a vertical channel.
- Laser Head And Cooler. The laser head is connected to the laser controller and to the cooler. Before running the laser, a check of the humidity level is necessary, because a high humidity in the laser head cavity causes serious damage to the optics inside. The cooler has a filter that needs to be replaced when it starts to show a brown color. It also needs to have a minimum level

of a specific type of water, containing also an anti-algae in the deionized water. The laser must be supported on a very flat surface.

- Laser Optics. The laser contains some optical elements through which the light source passes and comes out like a laser sheet.
- Laser Controller. The laser controller is connected to the laser head and the sensors, to the synchronizer and to the computer.
- Camera. The camera is connected to the computer by an Ethernet cable. The objective can be regulated in aperture (the amount of light in the sensor) to focus the flow.
- Synchronizer. The synchronizer is connected to the computer via a USB cable. It is connected also to the triggers of the laser heads and the cameras. The Dantec Studio software [15] is used to manage this system.
- Pressure System. The valve can be opened until the desired pressure has been reached and then closed.
- Acquisition Process. The computer is connected to the synchronizer, the cameras and the laser controller. The process consists in switching ON the laser and setting the current at 21 A, launching the Dynamic Studio software and setting the command “acquisition mode”. Dynamic Studio is a software for scientific imaging [15].

To calibrate the camera is necessary to set the “Free Run mode”, while to align it, to click on “Trigger Mode” and then, by viewing the live image, it is possible to focus on the camera.

To acquire the images, the “Trigger Mode” of the synchronizer is shifted to “External” and the sampling parameters, the acquisition frequency and the number of images to take, are chosen. During the acquisition LabVIEW software is used, then the images can be saved in the Data Base. LabVIEW is a graphical programming environment engineers use to develop automated research, validation, and production test systems [16].

After every acquisition, a quick check to consider the quality of the measure is done: using PIVlab, a PIV tool to analyze, validate, postprocess, visualize and simulate PIV data, the velocity vectors are displayed; then a calculation of the pressure of the system is done. If the images are good, they can be exported and ready to be processed.

2.4 PIV Synthetic Images

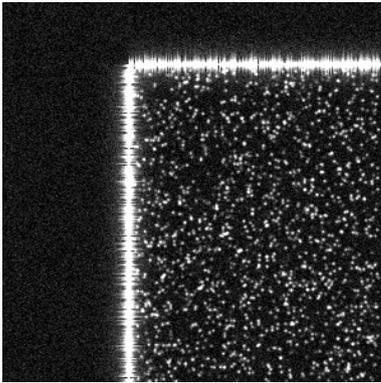


Figure 2.6: Flow in 90 degree corner

The free vortex test case will be studied starting from synthetic images. It is used a generator of PIV synthetic images to perform this. This algorithm allows to define the particles' features, like the displacement, the concentration of the particles and their position in the meshgrid.

It is possible also to add artificial background noise and reflection, phenomena that can be observed in real cases.

In addition, this image generator allows to define and modify the boundaries of the body hit by the flow, in order to simulate different kind of bodies. The program generates N images, that are ready to be used in the PIV process codes to find the velocity fields.

Figures 2.7, 2.8, 2.9 represent the synthetic PIV images created with three different geometries.

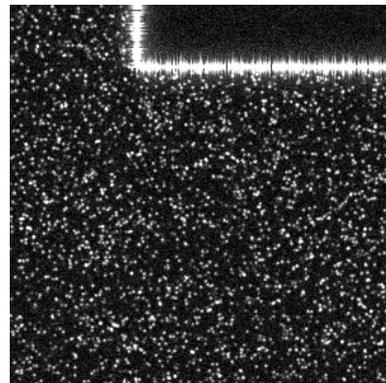


Figure 2.7: Flow over a cube

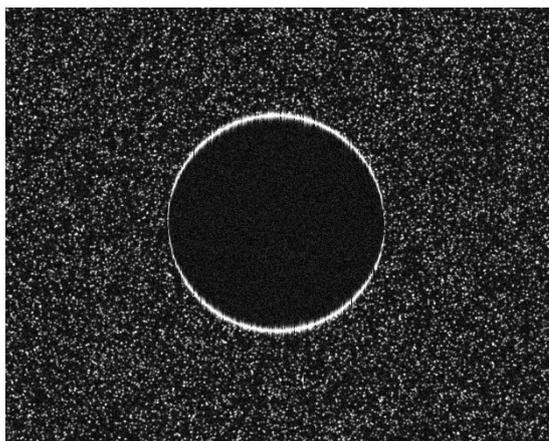


Figure 2.8: Flow past a cylinder

Chapter 3

3. Function approximation via RBF

In this chapter an overview about Machine Learning is introduced. Passing through the definitions of non-linear basis and the concepts of regression and optimization, ending up with constrained RBF (Radial Basis Functions) approximation for PIV fields regression.

3.2 Introduction to Machine Learning

Machine learning constitutes model-building automation for data analysis, data generation and taking actions, like face or object recognition, real time translation, generation of fake data, driving cars, robots.

Machine learning is part of Artificial Intelligence, studying how to use computers to simulate human learning activities, in which a computer program can learn and improve from data and experience.

If the old programming paradigm is about how to find answers from data and rules, the new programming paradigm, and consequently Machine Learning, revolves around how to find the rules from data and answers.

The learning process can be described as a functional risk [5]:

$$R(w) = \int L(y, \phi(x, y, w))p(x, y)dxdy$$

where $L(y, \phi(x, y, w))$ is the loss function, $\phi(x, y, w)$ defines the structure of the learning machine, w its parameters, and $p(x, y)$ is a probability distribution of x , that represent the input and y the output samples. This risk has to be minimized.

The concepts of Machine Learning, Optimization and Statistics are closely linked. Starting with the collection of two datasets, training and validation data, the challenge is to learn an objective function. Then, based on an hypothesis set and experience, the learning algorithm can learn the parameters and predict the final hypothesis, performing an optimization.

Another important element is statistics. This is necessary to understand the error entity of the weights prediction: how much the model depends on the data and so how much it can be generalized to other situations.

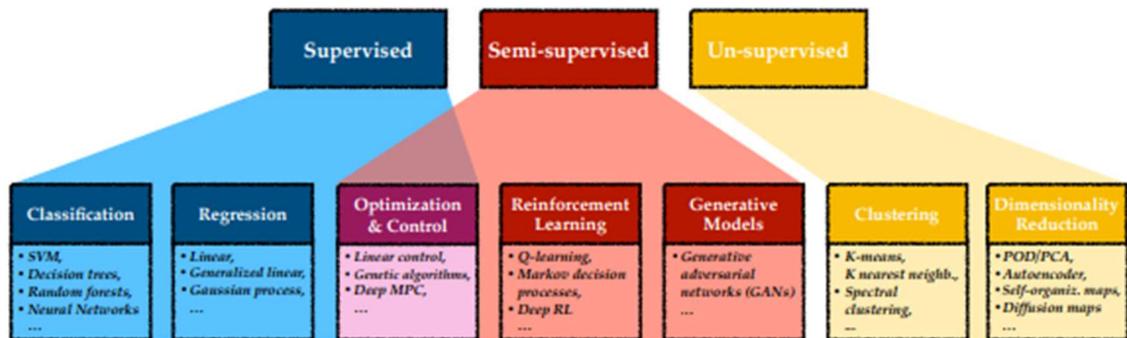


Figure 3.1: Machine learning algorithms categories, [5]

Depending on the extent and type of information available for the learning process, learning algorithms can be grouped in Supervised, Semi-supervised and Unsupervised [5, 6] (See Figure 3.1)

The Supervised learning makes predictions linking the function input with the output through some input and output data. The two tasks are Regression and Classification, where in the first the algorithm predicts a continuous variable, learning a curve that has to be as close as possible to some given points, minimizing uncertainty between old and new data. The second predicts a categorical data, the decision boundary separates and maximizes the width between classes, relying only on old data.

The Unsupervised learning maps an input to itself. The two tasks are Dimensionality Reduction and Clustering, which refers to groups of similar behaviors, performed by measurement techniques such as Image Processing.

Semi-supervised learning tasks are a mix of the first two, both predictive and descriptive. This includes another task, Reinforcement learning, making critics'

predictions without a model, trying to quantify the correctness of one action with respect to another.

Turbulence modeling problems are essentially regression problems. Turbulence modeling means mapping a term, relating it to a known variable, doing a regression. Flow control is instead based on observations without knowing the equation, hence the input is a state and the output is an action.

Deep learning algorithms usually show how to connect inputs and outputs by looking at how numerical schemes solve a problem step by step, then performing regression and trying to figure out what comes next. For more complex systems, such as dynamic systems, this approach is very difficult because the network learns by considering many solutions.

3.3 Advantages

Image-based velocimetry can provide velocity fields with sufficient resolution, but if this is not enough, it is possible to increase the resolution using a PIV-fields constrained regression, via RBF approximation of the solution.

Most approaches are based on “classic” numerical techniques, which require building a computational mesh to integrate the equations, as in CFD. This represents a difficult task, especially in presence of curved boundaries. Machine Learning attempts to overcome this problem by solving the partial differential equations from Image Velocimetry using meshless linear optimization methods. These are obtained via Radial Basis Functions parameterization of the solution, that depends on a set of weights. Weights are found using constrained optimization techniques, where the constraints include boundary and initial conditions.

These methods have several advantages, as the RBF can be interpolated over any grid to accurately compute the derivatives, solving the difficulties associated with geometric complexity [9]. This makes it easy to introduce constraints in linear least square regression.

For high Reynolds numbers, simulations ought to use the Navier-Stokes equations. Not being capable to solve them analytically, the alternative used consists of simulations based on approximations of these equations or laboratory experiments

for specific configurations, too slow for real-time control and expensive for iterative optimization. Machine learning opens up new methods for dimensionality reduction and reduced order modeling in fluid mechanics [5].

3.4 Linear Regression via Nonlinear Basis

This paragraph will introduce regression with a specific focus on the techniques to solve nonlinear regression problems. The matrix to be inverted contains nonlinear bases in its columns. Following, Polynomial Basis and Radial Basis are shown, ending up with the notion of regularization.

Referring to linear regression, consider n_p linearly related points $x, y \in \mathbb{R}^{n_p \times 1}$.

Accepting to make predictions with some uncertainties, assume that the regression has a deterministic part $f(x)$, that is the model, and a stochastic part $n(x)$, that is the error, e.g. noise. Assuming that the expected value of the mean of the stochastic part is zero, and the system is mostly deterministic, the overall prediction is made by the deterministic part

$$y = f(x) + n(x) \approx \tilde{f}(x).$$

3.4.1 Finite Basis Functions and Linear Algebra Solution

The function approximation $\tilde{f}(x) \in \mathbb{R}^{n_p \times 1}$ to be constructed, is a linear combination between a finite set of n_b nonlinear basis functions and weights $\boldsymbol{\phi}(x)\mathbf{w}$, where $\boldsymbol{\phi}(x) \in \mathbb{R}^{n_p \times n_b}$ is the matrix of non linear basis evaluated on the training set \mathbf{x} , and $\mathbf{w} \in \mathbb{R}^{n_b \times 1}$ is the vector containing the set of weights.

Training a data set \mathbf{x} (e.g. scattered points of a PIV image) means solving the linear system

$$\begin{aligned}
\mathbf{y} &\approx \tilde{f}(x) = \Phi(x)\mathbf{w} \\
&= \sum_{j=1}^{n_b} w_j \phi_j(x_k) \\
&= \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_{n_b}(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_{n_b}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_{n_p}) & \phi_2(x_{n_p}) & \cdots & \phi_{n_b}(x_{n_p}) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n_b} \end{bmatrix} \quad (1)
\end{aligned}$$

identifying the weights.

If $\Phi(x)$ is full rank, the solution is obtained as

$$\mathbf{w} = (\Phi^T(x)\Phi(x))^{-1}\Phi^T(x)\mathbf{y}$$

Uncertainties in weights will propagate in the final problem \mathbf{y}' , because the linear operator $\Phi(x')$ map scalars weights \mathbf{w} into functions \mathbf{y}' , hence they are correlated. Once the algorithm has been trained on a historical dataset, it is possible to apply new data, obtaining in output a “prediction”. Performing a prediction, projecting the target functions \mathbf{y} on the built non linear basis, permits to use the basis as a support for making estimates. The algorithm generates a possible value for the unknown variable for each item in the new data, allowing the modeler to decide what this value should be.

The identification of the weights is called training, and the data used for this scope is called training set. The identification of the function from known weights and new data is called prediction, and the data used for this scope is called validation set [6].

3.4.2 Radial Basis Functions (RBFs)

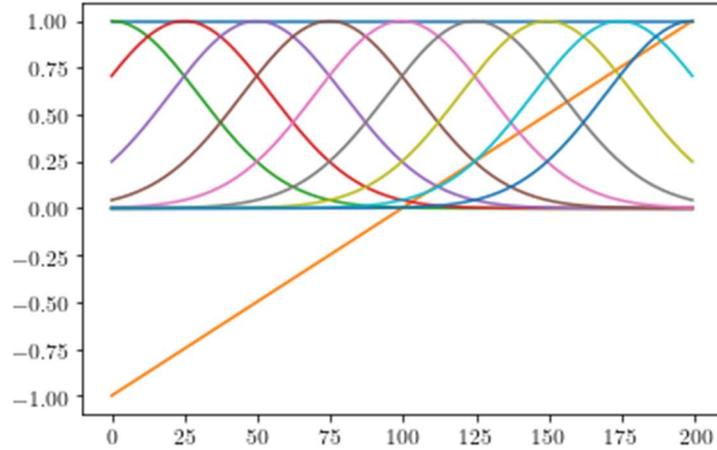


Figure 3.2: Hybrid case of shifted Gaussians plus a straight line, [6]

The support of a function is the subset of the domain containing the set of points where the function is not zero. Polynomial functions have an unbounded support, the trend is therefore not linear and the problem is ill-conditioned. This problem is similar to harmonic basis in the frequency domain, also with unbounded support, so it is useful to have localized basis with a limited support, such as Radial Basis [6].

RBFs are tools for approximating a function $f(x) \in \mathbb{R}^d$, with $d = 2$ or $d = 3$.

In literature different types of RBF can be found, the most common used are Gaussian and Multiquadratic, respectively expressed as

$$\phi(\|x - x_i\|_2) = e^{-t\|x-x_i\|_2^2}$$

$$\phi(\|x - x_i\|_2) = \sqrt{t^2 + \|x - x_i\|_2^2}$$

with $t > 0$ the shape parameter (choosing isotropic basis, only one shape parameter is considered) and x_i is the collocation point.

This function approximation is built from data and can be constrained. In this work, this allows to derive an approximation of velocity fields.

3.4.3 Optimization and Regularization

Following the literature, the problem is ill-posed, and this issue can be solved adding a regularization.

By applying a linear differentiation operator to the RBF functions, a linear relationship between the weights and the differentiated value is established. Assuming to minimize the function $f(x)$ subject to an equality constraint $g(x) = 0$, the minimum of the function must satisfy the condition

$$\nabla f + \lambda \nabla g = \nabla(f + \lambda g) = \nabla A = 0 \quad (2)$$

The minimum of the function $f(x)$ has to be along the function path which is parametrized by the implicit function $g(x) = 0$, as a constrained optimization.

The gradient of f and the gradient of g are parallel. In other words, one vector is a multiple of the other and the multiplier λ is a hyperparameter, that in Machine Learning is a parameter whose value is used to control the learning process. It is called Lagrange multiplier.

The difference between constrained optimization and regularization is how Lagrange multipliers are handled. In optimization the multiplier is found solving the system, in regularization, it is imposed.

Rewriting equation (2) as the norm of the residual of the linear system $\mathbf{y} = \Phi(x)\mathbf{w}$ plus a regularization term $R(w)$ multiplied by the Lagrange multiplier, we get

$$J(w) = \|\mathbf{y} - \Phi(x)\mathbf{w}\|_2^2 + \lambda R(w) \quad (3)$$

To obtain a good approximation the distance $\|\mathbf{y} - \Phi(x)\mathbf{w}\|_2^2$ between the function \mathbf{y} and prediction $\Phi(x)\mathbf{w}$ must be minimized.

The Lagrangian multiplier λ acts as a penalization and a right choice of the regularization function $R(w)$ is therefore fundamental.

Classic choices are the following:

- $R(w) = 0$ Ordinary Least Square
- $R(w) = \|\mathbf{w}\|_2$ Tikhonov regularization
- $R(w) = \|\mathbf{w}\|_1 = \sum |w_k|$ Lasso regularization
- $R(w) = \gamma_1 \|\mathbf{w}\|_1 + \gamma_2 \|\mathbf{w}\|_2$ Elastic net

The ordinary Least Square ends up with a linear regression.

The Tikhonov regularization leads to Ridge regression, promoting small weights.

Lasso regularization promote sparse weights and it is possible to tolerate some weights very large if most of the others are zero.

In Figure 3.3, an example of the effect of the regularization term on the regression is shown.

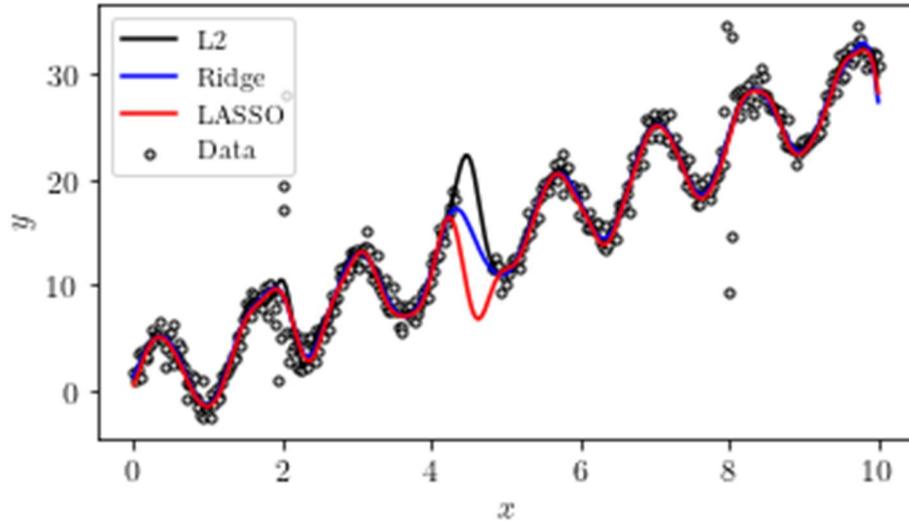


Figure 3.3: RBF approximation with different regularizations

3.4.4 Solution for Tikhonov Regularization

At this point, the Ridge regression will be considered, finding the solution for the Tikhonov Regularization, with $R(w) = \|w\|_2^2$.

Recalling the function (3) and by replacing the Tikhonov Regularization,

$$J(w) = \|y - \Phi(x)w\|_2^2 + \lambda\|w\|_2^2$$

$J(w)$ is therefore a quadratic function and minimizing it means simply finding its stationary point.

Computing the norm we get:

$$J(w) = \|y - \Phi(x)w\|_2^2 + \lambda\|w\|_2^2 = (y - \Phi(x)w)^T(y - \Phi(x)w) + \lambda w^T w$$

Then, computing the gradient, we have:

$$\begin{aligned} \nabla_w J(w) &= \nabla_w [(y - \Phi(x)w)^T(y - \Phi(x)w) + \lambda w^T w] \\ &= \nabla_w [w^T \Phi^T(x) \Phi(x) w - w^T \Phi^T(x) y - y^T \Phi(x) w - y^T y \\ &\quad + \lambda w^T w] = 2(\Phi^T(x) \Phi(x) w - \Phi^T(x) y + \lambda w) \end{aligned}$$

Setting $\nabla_w J(w) = 0$ we obtain

$$w = \left(\Phi^T(x)\Phi(x) + \lambda I_{n_p} \right)^{-1} \Phi^T(x)y.$$

The matrix $\Phi^T(x)\Phi(x) = K(x)$ is the Covariance Matrix and I_{n_p} is the Identity matrix, an extra term coming from the regularization. If $\lambda = 0$ it becomes an ordinary least square.

Computing the Hessian of J , we have $\mathbb{H}(J(w)) = 2\Phi^T(x)\Phi(x) + 2\lambda I_{n_p}$; it is clear that if $\lambda > 0$ the matrix \mathbb{H} is positive definite, and if $\lambda = 0$ and $\dim(\text{Ker}(\Phi)) = 0$ the matrix \mathbb{H} is positive definite. This ends up with the facts that the function $J(w)$ in both cases is convex, has a unique minimum and \mathbb{H} is invertible.

3.5 Constrained Regression via RBFs

In constrained regression, minimizing the error continues to be the target, but, in this case, constraints are added to the problem.

Let us rewriting equation (2) in the generic form $J(w) = \|y - \tilde{y}\|_p$,

where $\tilde{y} = Aw$ is a generic linear system, $\| \cdot \|_p$ is the L_p norm, defined as $L_p = (\sum |x_i|^p)^{\frac{1}{p}}$.

Choosing again the norm $L_2 = \| \cdot \|_2$ for simplicity, we have

$$J(w) = \|y - Aw\|_2 = (y - Aw)^T (y - Aw) = w^T A^T A w - w^T A^T y - y A w + y^T$$

$$\nabla_w J(w) = 2A^T (Aw - y)$$

that is zero if $A^T A w = A^T y$.

Imposing that the curve-fit must also satisfy other criteria, means setting some

$$\text{equality or inequality constraints } \begin{cases} R_e(w) = 0 \\ R_i(w) \leq 0 \end{cases}.$$

Boundary conditions yields equality constraints, and these are considered in this case.

Recalling the system (1), the function approximation is a linear combination of n_b RBFs

$$f(x) \approx \tilde{f}(x) = \sum_{k=0}^{n_b} w_k \phi_k(x | x_k, t_k)$$

where x_k is the collocation point and t_k is the shape parameter.

Boundary conditions can be written as $\mathbf{y}_B = \boldsymbol{\phi}_B \mathbf{w}$, where \mathbf{y}_B and $\boldsymbol{\phi}_B$ refers to boundary points.

Rewriting equation (2), we have

$$J(\mathbf{w}, \lambda) = \|\mathbf{y} - \Phi(\mathbf{x})\mathbf{w}\|_2^2 + \lambda R(\mathbf{w}),$$

where this time the regularization term is $R(\mathbf{w}) = \mathbf{y}_B - \boldsymbol{\phi}_B \mathbf{w}$

and carrying out the norm

$$\begin{aligned} J(\mathbf{w}, \lambda) &= \|\mathbf{y} - \Phi(\mathbf{x})\mathbf{w}\|_2^2 + \lambda(\mathbf{y}_B - \boldsymbol{\phi}_B \mathbf{w}) = \\ &= (\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^T (\mathbf{y} - \Phi(\mathbf{x})\mathbf{w}) + (\mathbf{y}_B - \boldsymbol{\phi}_B \mathbf{w}) = \\ &= (\mathbf{y} - \Phi(\mathbf{x})\mathbf{w})^2 + \lambda(\mathbf{y}_B - \boldsymbol{\phi}_B \mathbf{w}), \end{aligned}$$

taking the gradient in \mathbf{w} and λ

$$\begin{aligned} \nabla_{\mathbf{w}} J(\mathbf{w}, \lambda) &= 2(\Phi^T(\mathbf{x})\Phi(\mathbf{x})\mathbf{w} - \Phi^T(\mathbf{x})\mathbf{y}) + \lambda \boldsymbol{\phi}_B^T \\ \nabla_{\lambda} J(\mathbf{w}, \lambda) &= (\mathbf{y}_B - \boldsymbol{\phi}_B \mathbf{w}), \end{aligned}$$

looking for points vanishing the gradient, the constrained system is

$$\begin{pmatrix} 2\Phi^T(\mathbf{x})\Phi(\mathbf{x}) & \boldsymbol{\phi}_B^T \\ \boldsymbol{\phi}_B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \lambda \end{pmatrix} = \begin{pmatrix} 2\Phi^T(\mathbf{x})\mathbf{y} \\ \mathbf{y}_B \end{pmatrix} \quad (4)$$

Where $2\Phi^T(\mathbf{x})\Phi(\mathbf{x}) \in \mathbb{R}^{n_b \times n_b}$, $\boldsymbol{\phi}_B^T \in \mathbb{R}^{n_b \times n_c}$, $2\Phi^T(\mathbf{x})\mathbf{y} \in \mathbb{R}^{n_b \times 1}$, $\mathbf{y}_B \in \mathbb{R}^{n_c \times 1}$

The problems of velocity fields approximation and meshless pressure computation leads to constrained least square problems of this form.

Chapter 4

4. Test cases

In this short chapter one 1D problem, for practise, and two 2D problems on which the algorithms have been implemented are described. First, we consider a sinusoidal distribution of points, then a simple free vortex flow, which indeed represents a steady, incompressible, inviscid, irrotational case, and finally a more challenging case is considered, as it is an experimental channel flow.

4.1 Sinusoidal distribution of points

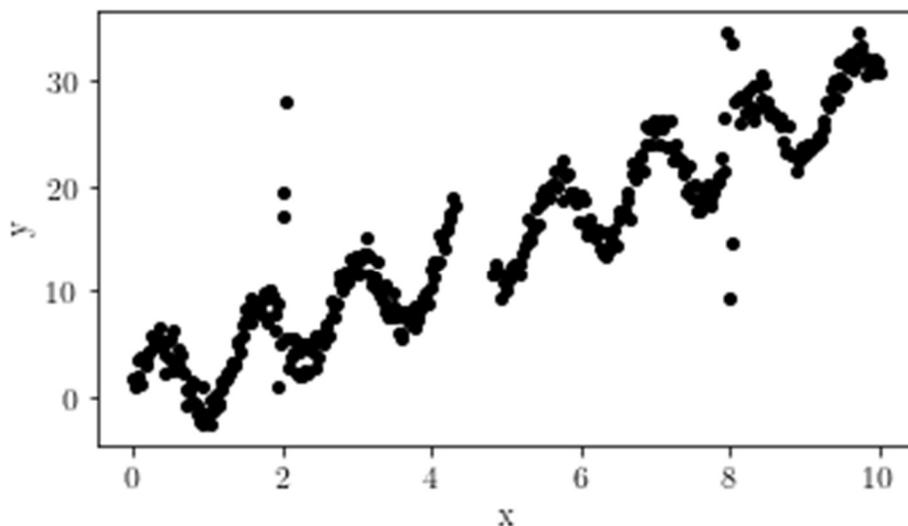


Figura 4.1: Sinusoidal distribution of points

This 1D case is the simplest. A set of points with a sinusoidal trend is given, on which a gaussian constrained RBF interpolation will be made. Boundary conditions are set to the left and right edges.

4.2 Free Vortex Flow

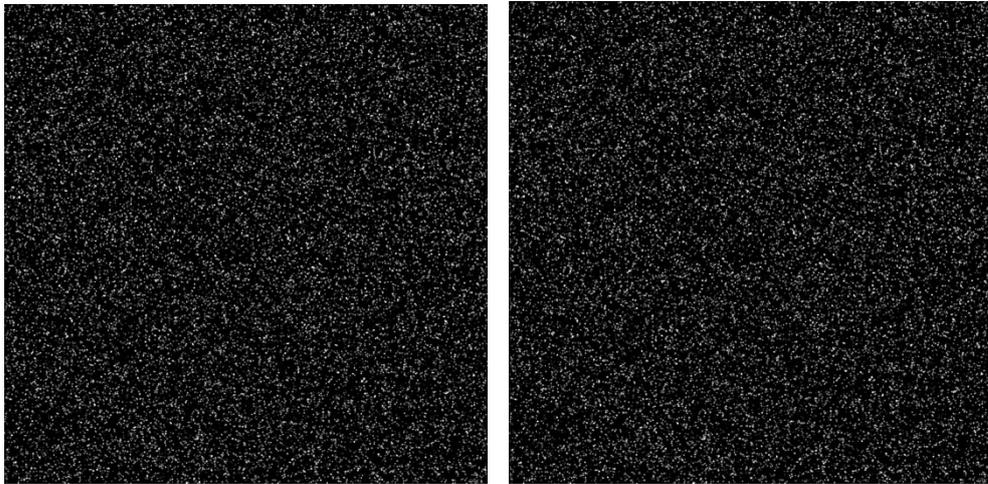


Figure 4.2: PIV Synthetic couple of Images of the present free vortex flow

A vortex is a region in a fluid in which the flow revolves around an axis line. Vortices are a major component of turbulent flow. In the absence of external forces, due to the viscous friction the vortex evolves quickly toward the irrotational flow pattern, represented by concentric circular about a given point, where the fluid flow velocity is greatest next to its axis and decreases in inverse proportion to the distance from the axis. The fluid elements do not rotate around themselves, but just go around the circular path. Irrotational vortices are also called free vortices.

Making some simplifications:

- the flow is steady
- the velocity remains smaller than the speed of sound (incompressible flow)
- the fluid is inviscid (viscous effects are normally confined to a very thin boundary layer)
- it has no vorticity (fluid particles are not rotating) everywhere (except at the location of the point vortex, where the derivative u_θ is infinite)

$$\omega = \nabla \times \mathbf{V} = 0$$

For an irrotational vortex, the circulation has a fixed value for any contour that does enclose the axis once.

$u_\theta(r, \theta) = \frac{const}{r}$ represents the tangential component of the particle velocity, constant along a (circular) streamline and inversely proportional to the radius r , derived from circulation Γ

$$\Gamma = \oint \mathbf{V} \cdot d\vec{l} = u_\theta 2\pi r \Rightarrow u_\theta(r, \theta) = \frac{\Gamma}{2\pi}$$

Instead, the radial component of the particle velocity is null $u_r(r, \theta) = 0$

Integrating the velocity components, we can get the stream function

$\psi(r, \theta) = \frac{\Gamma}{2\pi} \ln r$. In Cartesian coordinates it is expressed as

$$\psi(x, y) = \frac{\Gamma}{4\pi} \ln(x^2 + y^2)$$

while the velocity components are

$$u(x, y) = \frac{\Gamma}{2\pi} \frac{y}{(x^2+y^2)+c}, v(x, y) = -\frac{\Gamma}{2\pi} \frac{x}{(x^2+y^2)+c}, c = \text{const} \ll 1.$$

Figure 4.2 represents an example of a theoretic free vortex flow.

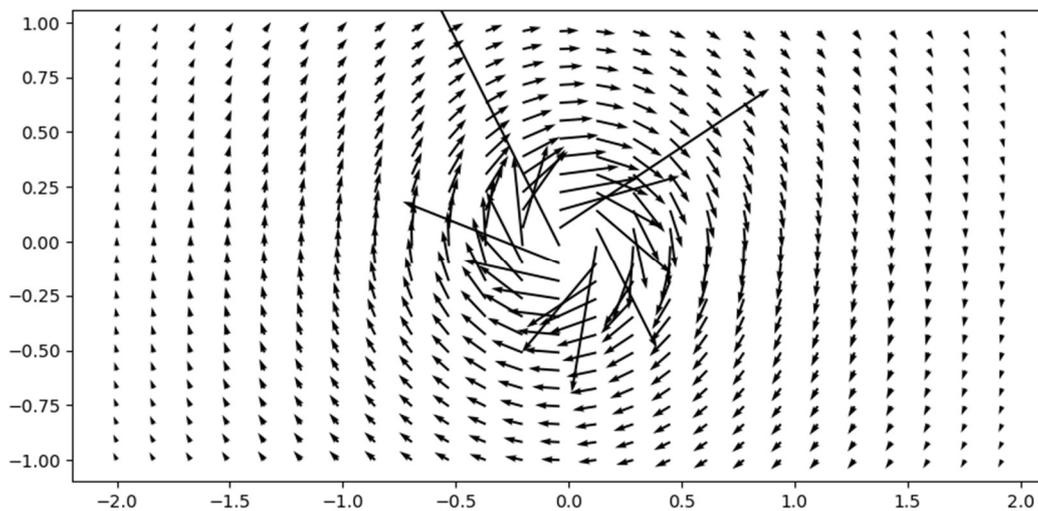


Figura 4.3 Theoretic free vortex flow

4.3 Experimental test case, channel flow flowing from/to the bottom

An internal flow is characterized by being confined by a surface. This means that the development of the boundary layer ends up being constrained by the geometric conditions.

The behavior of a channel flow is governed by the effects of viscosity and gravity relative to the inertial forces of the flow. Indeed, in this case, there is an additional source term F in the u -momentum equation, that include also the gravity force.

It is necessary to impose no-slip boundary conditions on the left and on the right.

Initial conditions $\Rightarrow u, v, p = 0$ everywhere

Boundary conditions $\Rightarrow u, v = 0$ at $x = 0, 2$, $\frac{\partial p}{\partial x} = 0$ at $x = 0, 2$,

p periodic at $x = 2$

In Figure 4.3, starting from the left, two frames A and B, a theoretical velocity computation and an animation representing velocity fields, obtained using all the 1200 PIV frames, are shown.

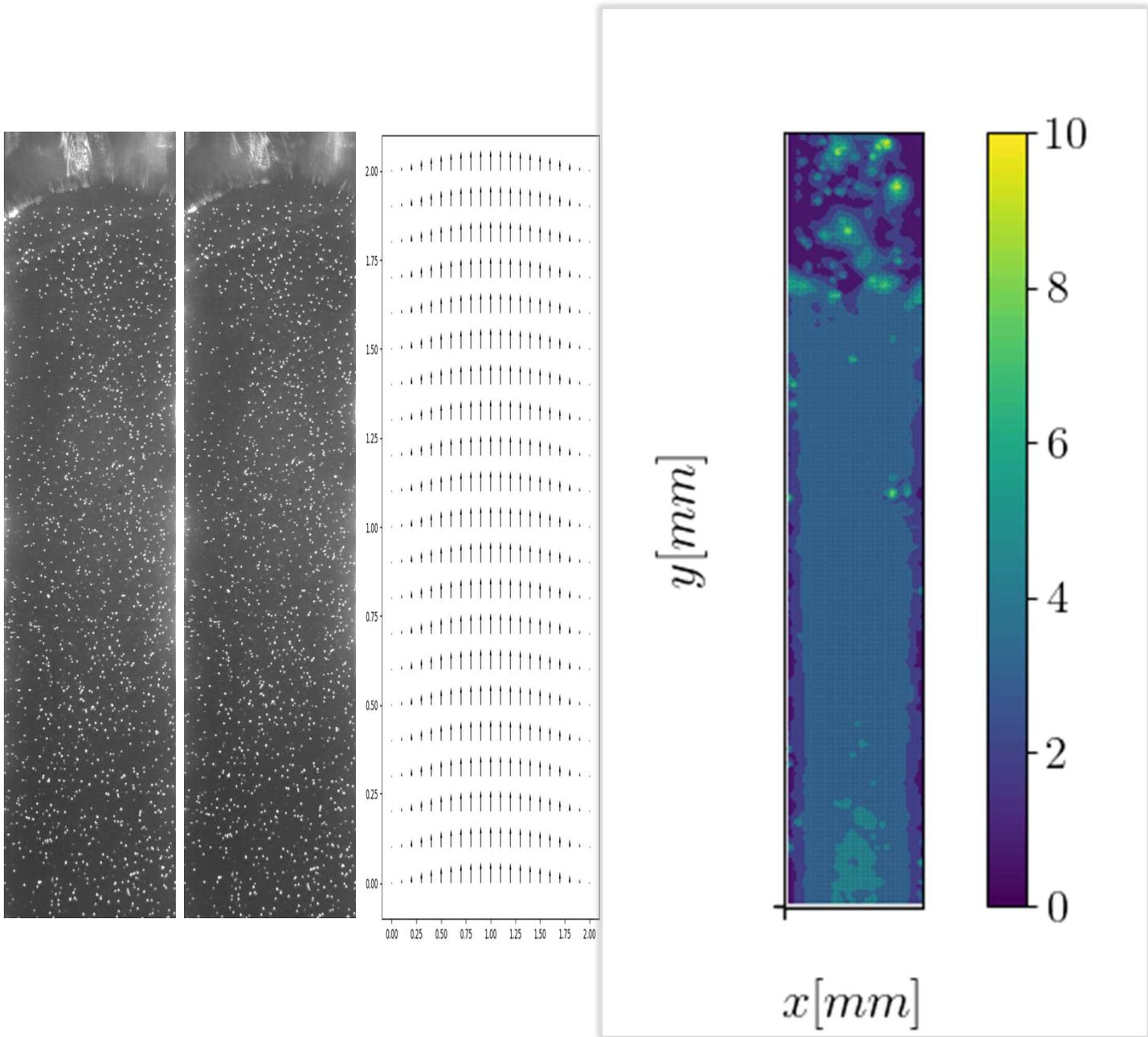


Figure 4.4: Image pair and motion field

Chapter 5

5. Python Implementation and Results

The methods described in previous chapters have been implemented in Python and are based on two sections.

The first section processes the PIV images and obtains the velocity fields using the open source Openpiv; all the arguments in a function are set to the most appropriate options and treshold values, in order to calculate the cross-correlation. After every calculation, a conversion from [pixel] to [mm] must be performed, through the Magnification Factor.

Velocity is expressed in [mm/s], images in [mm].

The second section performs a constrained Radial Basis Functions approximation PIV fields regression that depends on a set of weights, finding a good approximation of the PIV velocity fields:

- in the 2D vortex case, the velocity fields obtained from PIV, theory and RBF approximation are compared;
- in the 2D channel case, a Linear regression, a Lasso regression and a Ridge regression are implemented, increasing lambda from 0 to 1, and the mean error for each case is computed. After this, for the two best lambdas, $\lambda = 0$, $\lambda = 0.1$, the errors between PIV evaluation and RBF approximation are plotted.

A 1D RBF interpolation algorithm will be shown first. It is a simple case of 1D interpolation based on a set of given points, without the PIV processing section, to start practicing the constrained regression.

5.1 1D RBF Constrained Interpolation

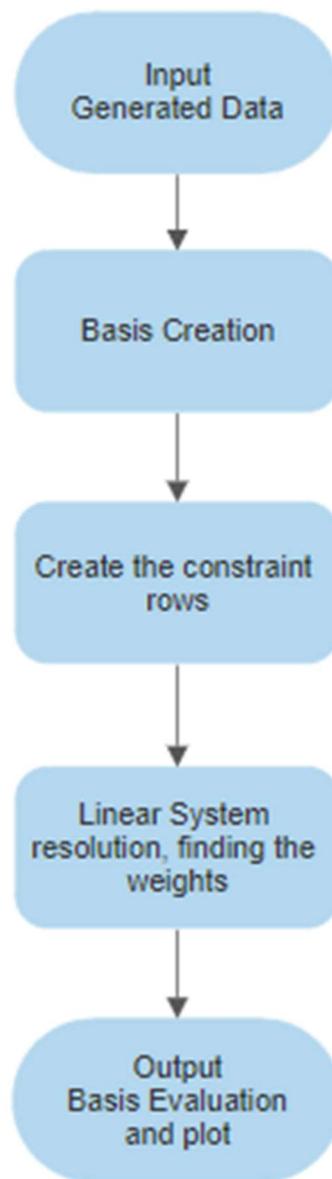


Figure 5.1: 1D Interpolation Flowchart

In Figure 5.1 the flowchart of the first algorithm is shown. In the sequel flowcharts, for the 2D test cases flowcharts will be considered (Figures 5.4 and 5.12).

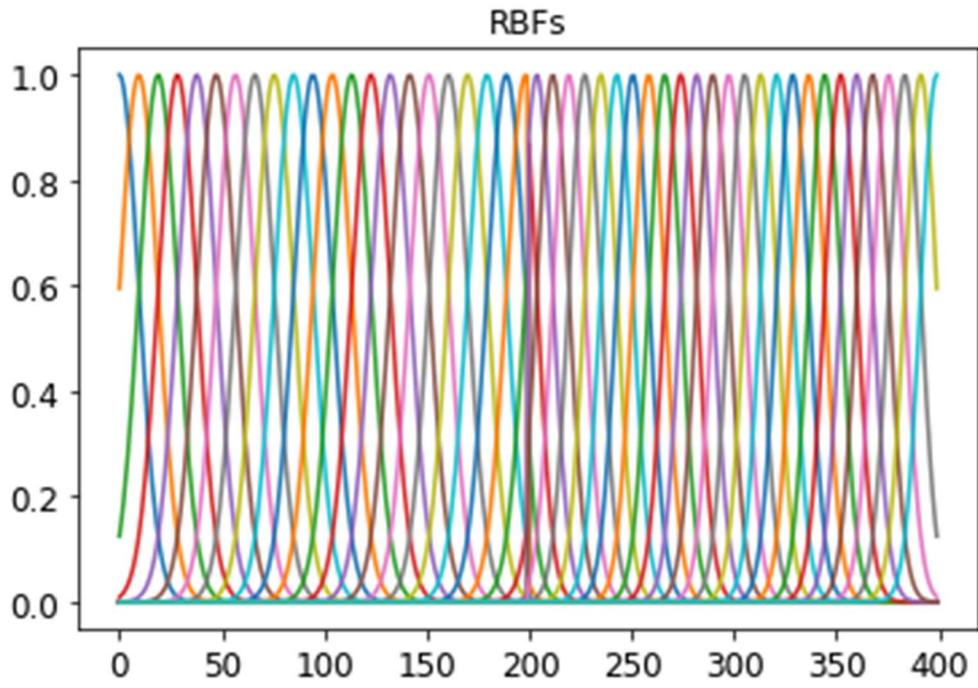


Figure 5.2: 1D Radial Basis Functions

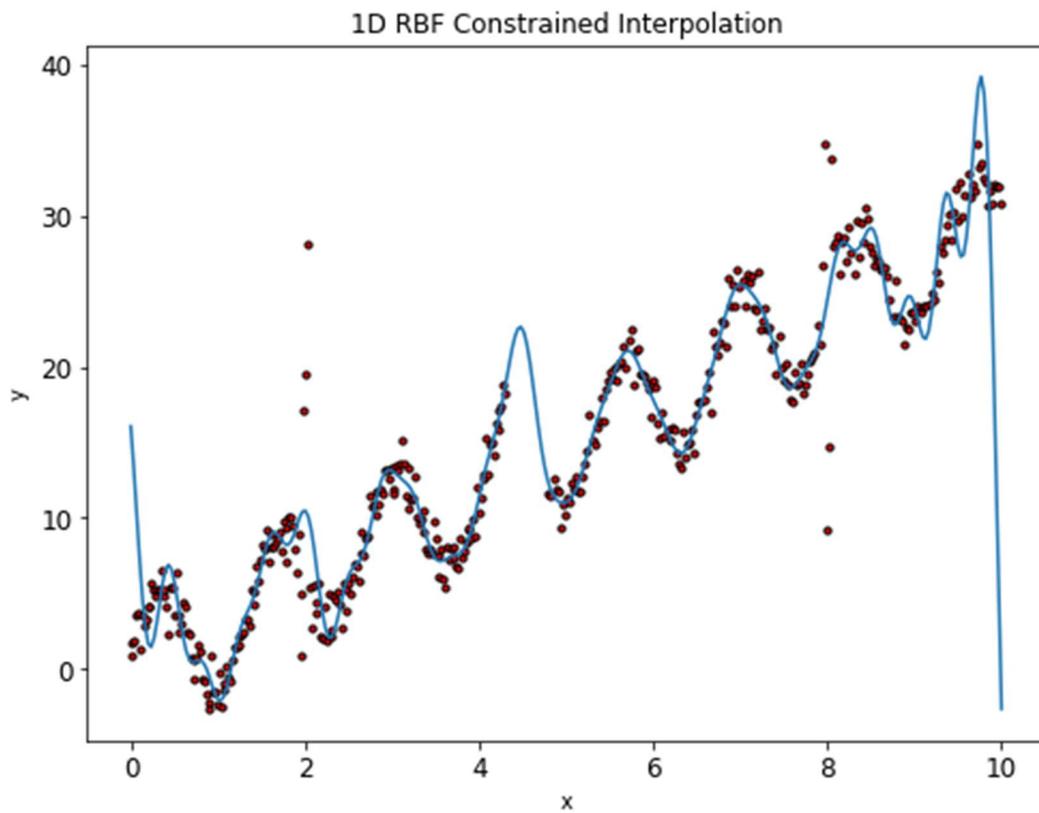


Figure 5.3: Constrained Interpolation

In Figures 5.2 the Gaussian RBFs used in this algorithm are shown.

Figure 5.3 shows the RBF interpolation of the given points. Boundary conditions are placed on the first point and on the last point of the function, in fact, in the constrained system of the algorithm, it has been imposed that the function assumes the value $y=15$ at the left boundary $x=0$ and the value $y=0$ at the right boundary $x=10$.

5.2 2D RBF Free Vortex Approximation

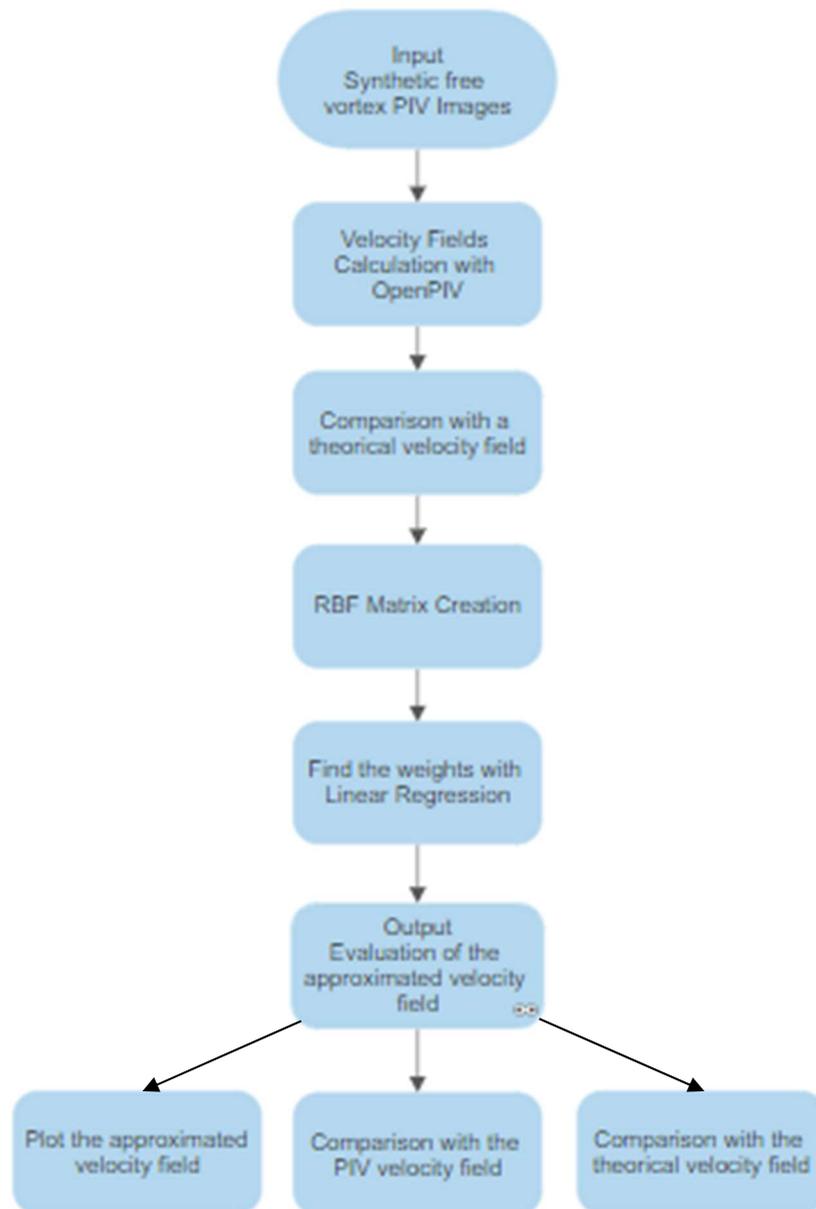


Figure 5.4: Free Vortex Flow RBF Approximation Flowchart

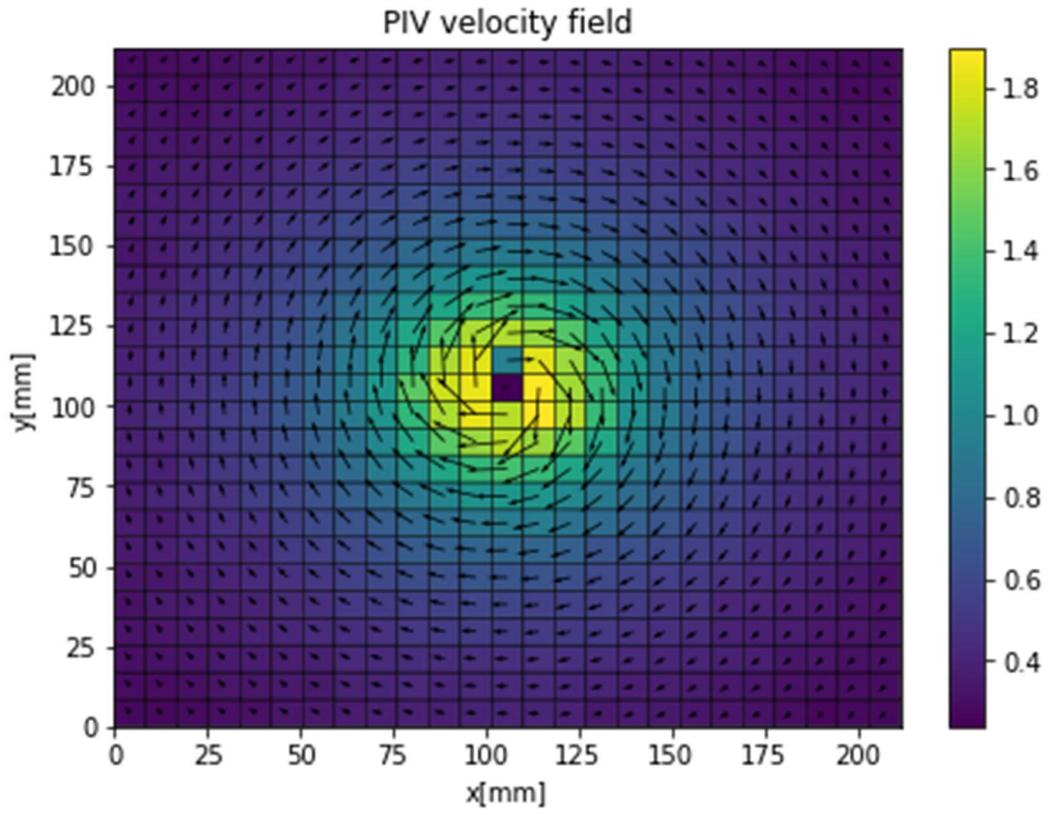


Figure 5.5: Vortex. PIV velocity field

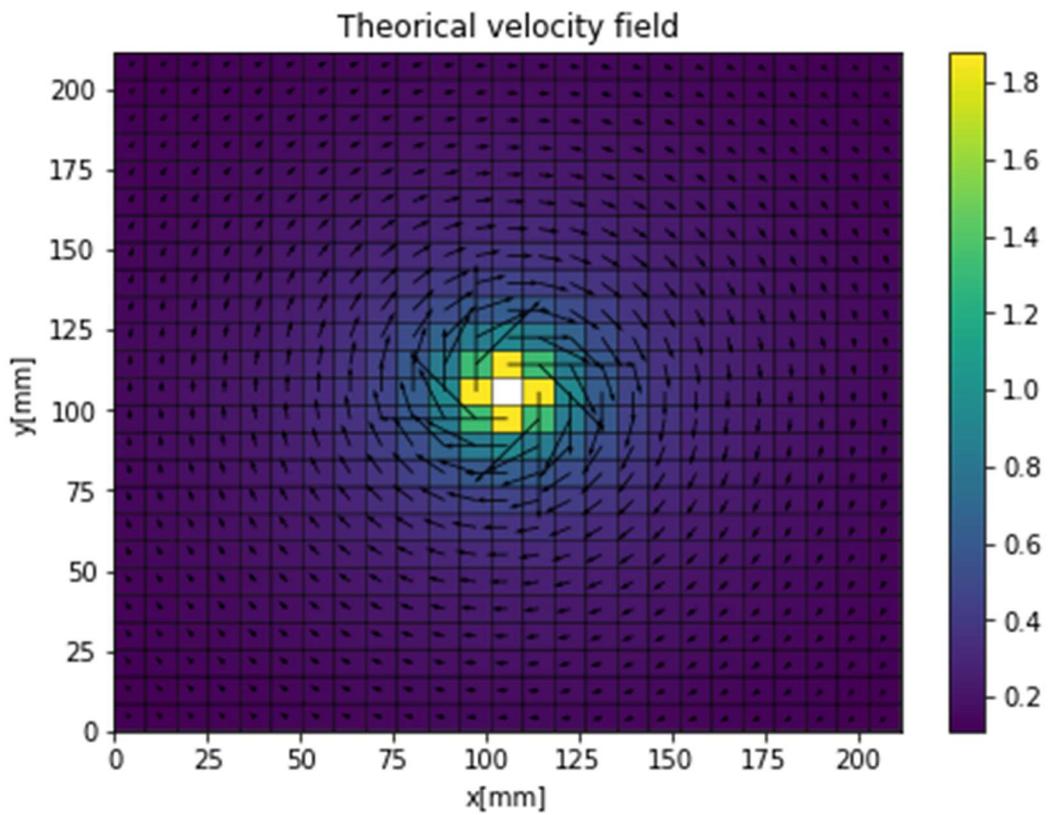


Figure 5.6: Vortex. Theoretical Velocity Field

In Figure 5.5, we report result obtained from PIV processing of vortex problem. It is worth noting that the theoretical calculation of the vortex, in Figure 5.6, is very different from the result obtained from PIV analysis, in Figure 5.5. This is due to the fact that the synthetic image generator builds the images inserting noise, reflections and other disturbances, simulating a real vortex, while the theoretical calculation is an exact calculation. The ideal irrotational vortex flow is not physically realizable, since it would imply that the particle speed would grow approaching the vortex axis. Indeed, in real vortices there is always a core region surrounding the axis where the particle velocity stops increasing and then decreases to zero as r goes to zero. Within that region the vorticity becomes non-zero. In fact, observing Figure 5.5, it is possible to see this behavior. As expected, in Figure 5.7, which represent the error between PIV evaluation and theoretical evaluation, it is shown that the error is high.

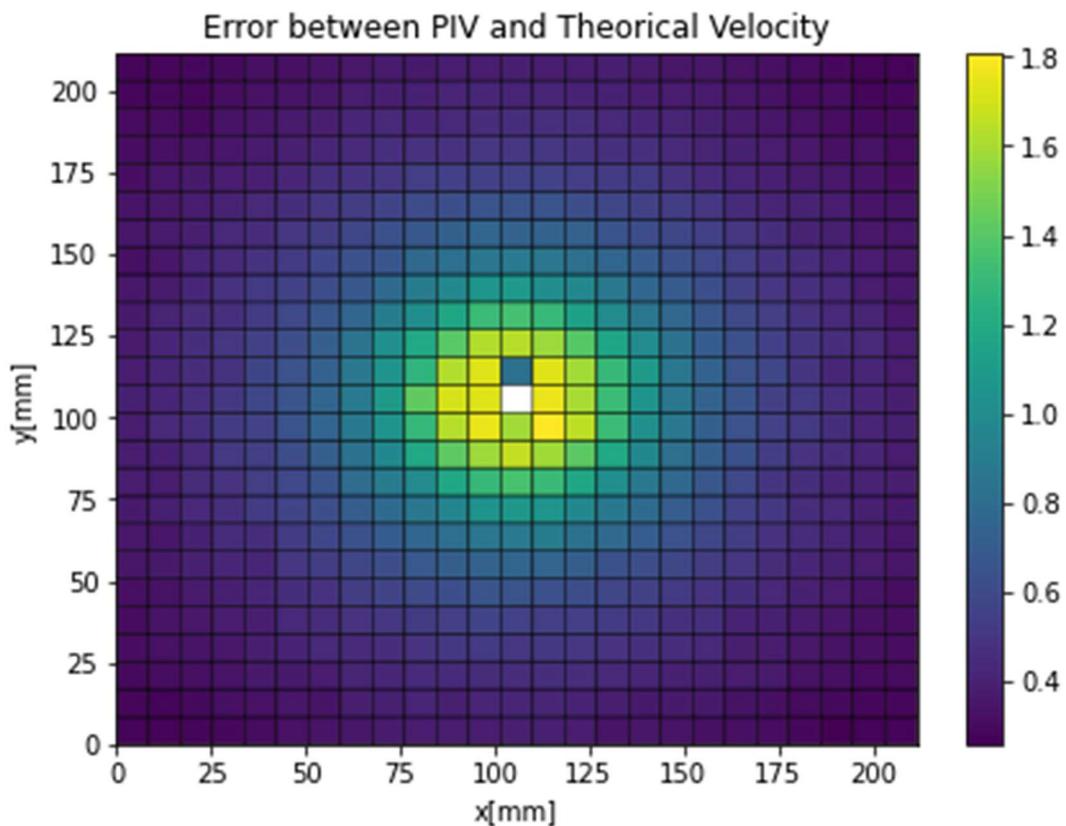


Figure 5.7: Vortex. Error between PIV fields and Theoretical fields

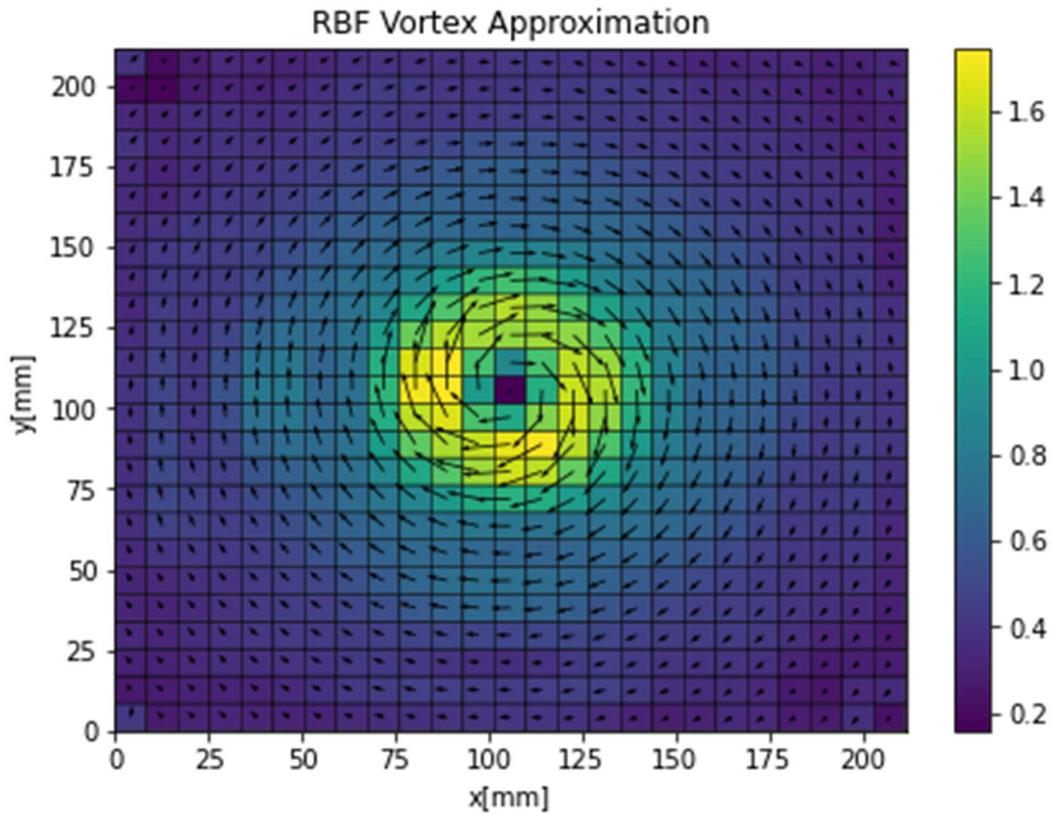


Figure 5.8: Vortex. RBF Approximation

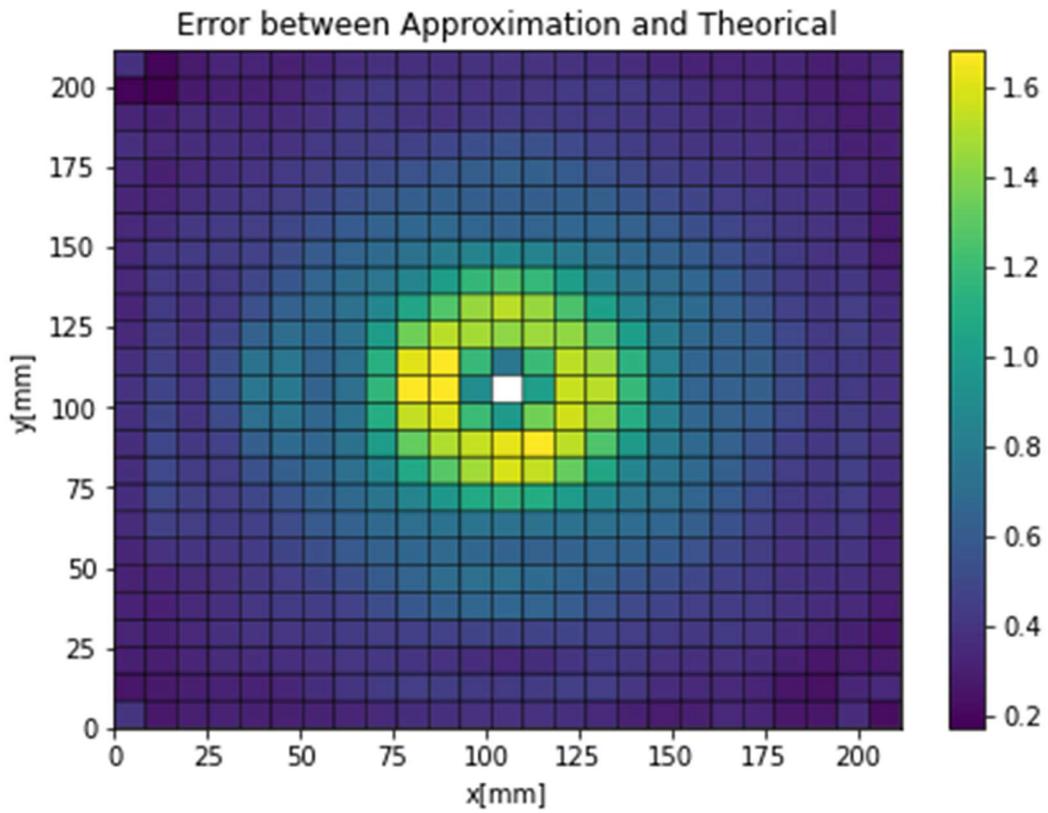


Figure 5.9: Vortex. Error between Approximation and Theoretical

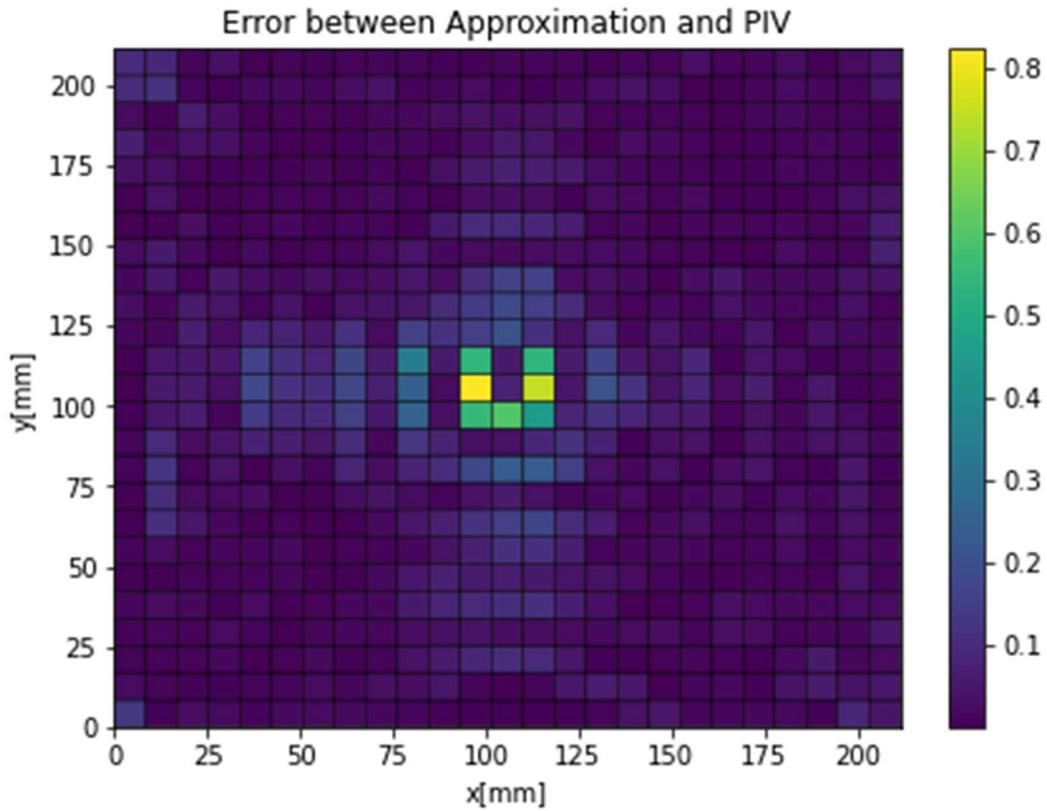


Figure 5.10: Vortex. Error between Approximation and PIV

Figure 5.8 represents the RBF Approximation of 2D vortex. We can note, in Figure 5.9, that the error between RBF approximation and theoretical computation is high too, as RBF approximation is a PIV-based approximation.

Instead, the RBF approximation fields are similar to the PIV fields, in fact, as it is shown in Figure 5.10, in this case the error is lower.

5.3 2D RBF Channel Flow Approximation

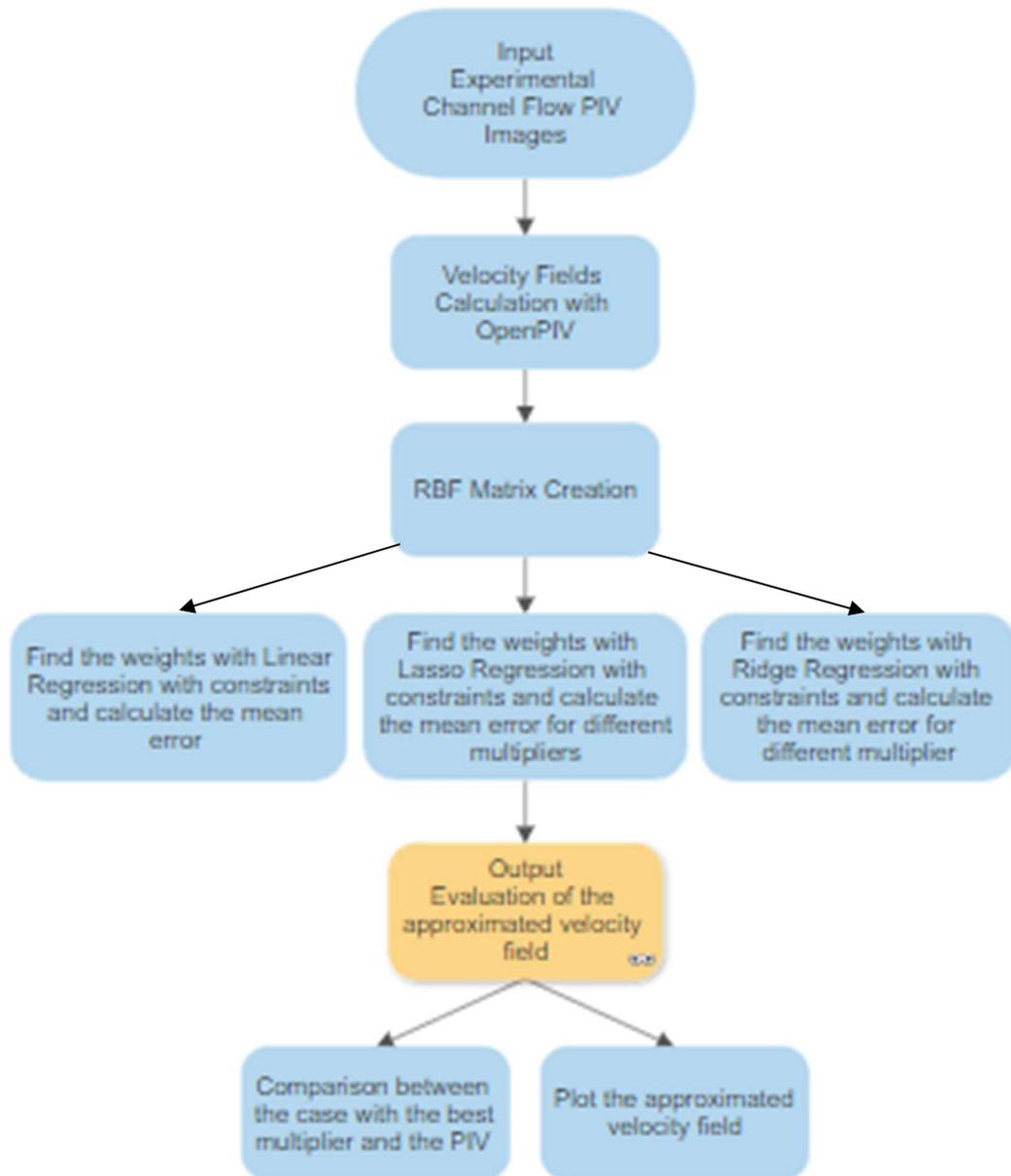


Figure 5.12: Channel Flow RBF Approximation Flowchart

The case considered in this paragraph relates to the laboratory experiment.

During the PIV campaign, for different conditions and parameters (pressure, sampling frequency, the rise / fall of the flow) a number of PIV images were collected with a certain frequency of sampling. In particular, the two PIV shot frames A and B considered for this analysis regards the “Fall” phase of the flow.

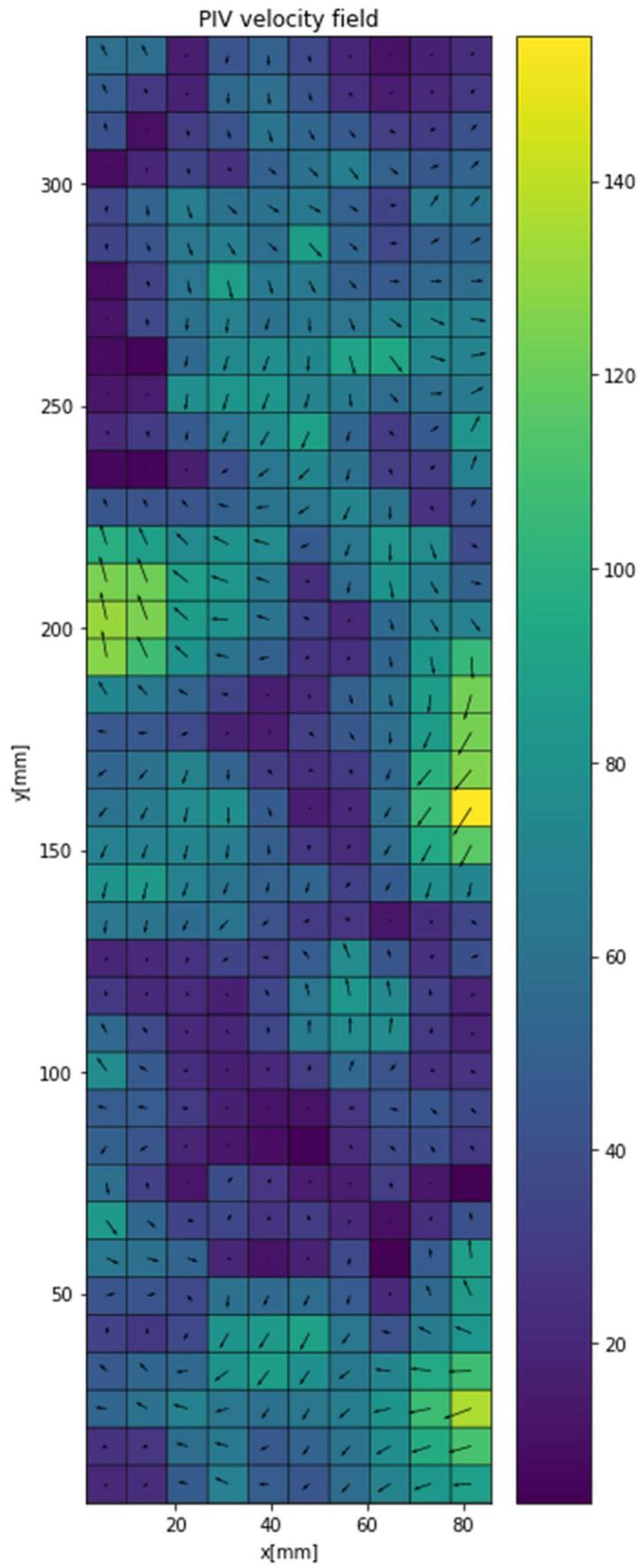


Figure 5.11: Channel Flow. PIV Velocity Field

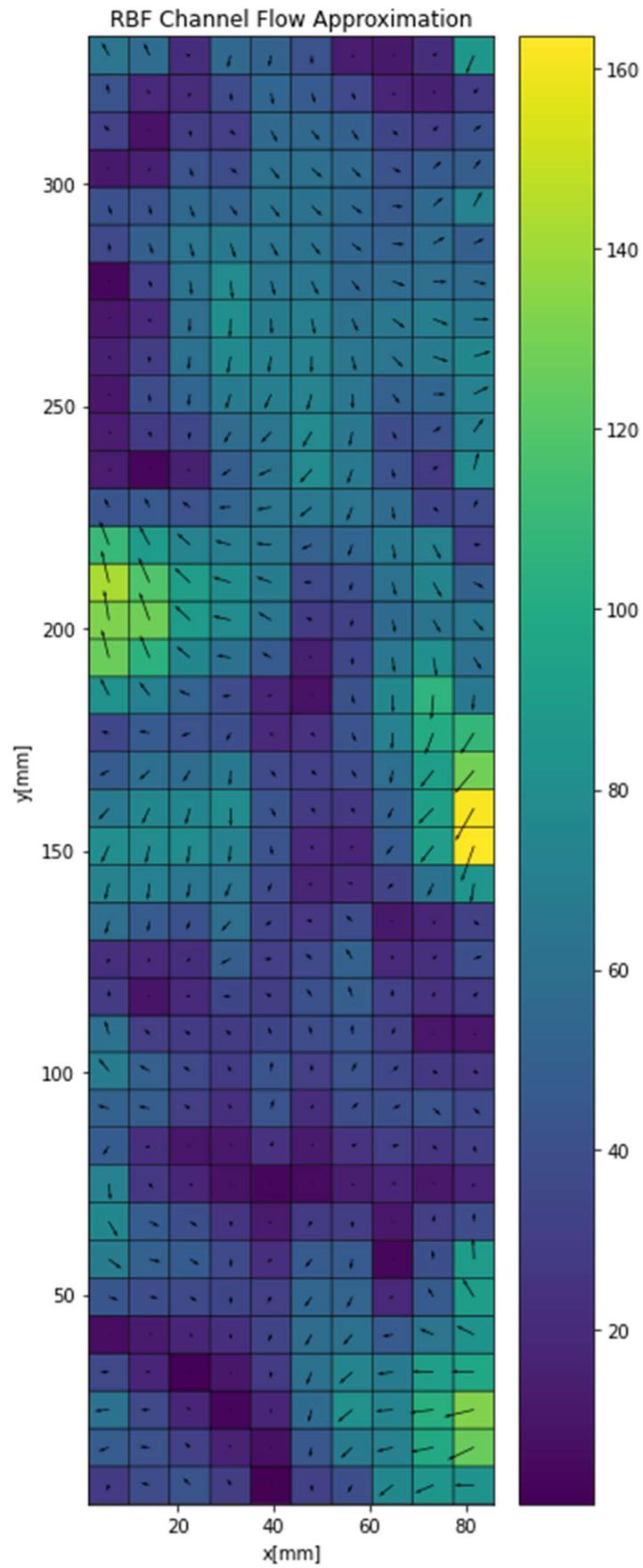


Figure 5.12: Channel Flow. RBF Approximation

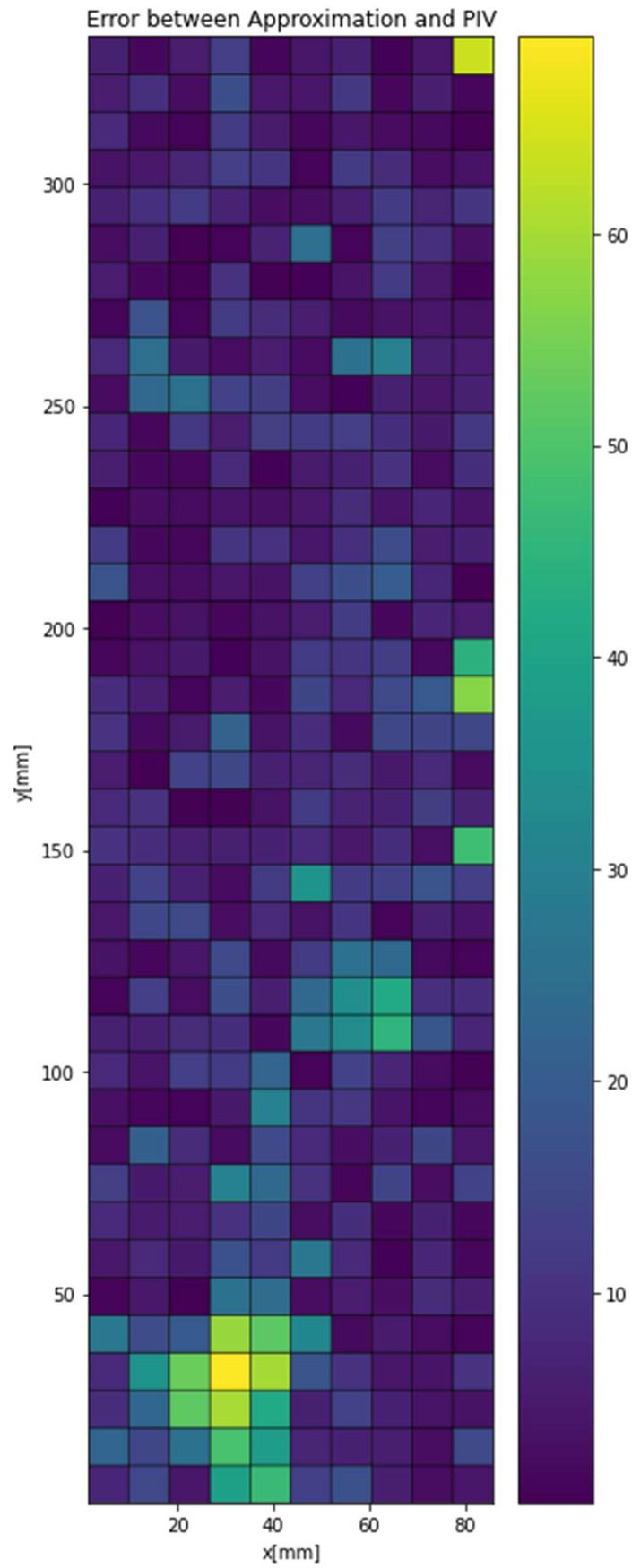


Figure 5.13: Channel Flow. Error between PIV and Approximation

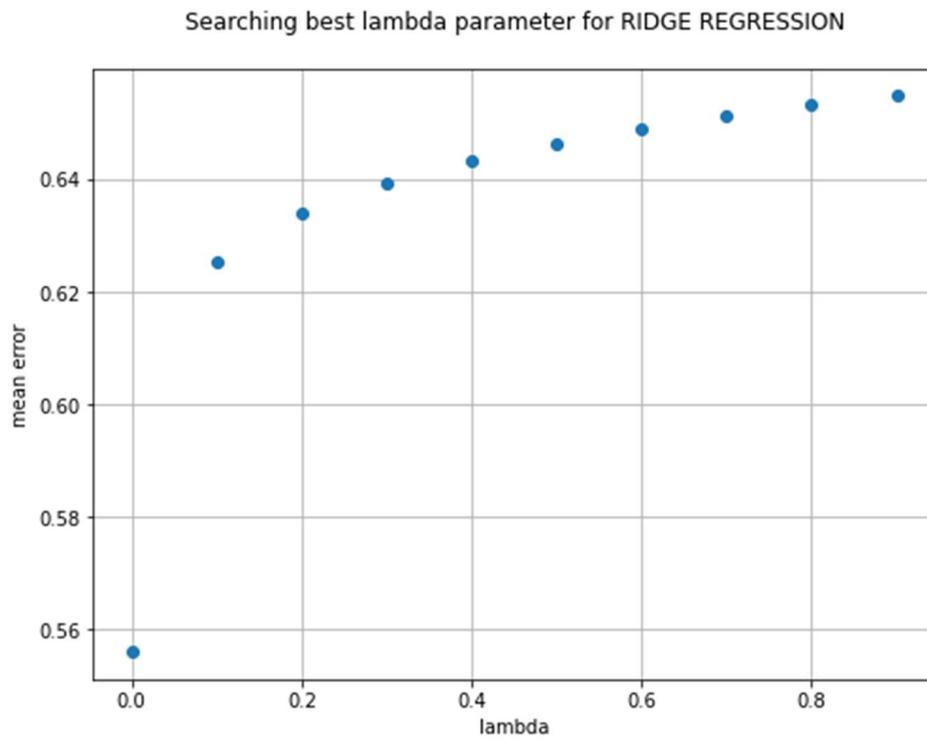


Figure 5.14: Best lambda for Ridge Regression

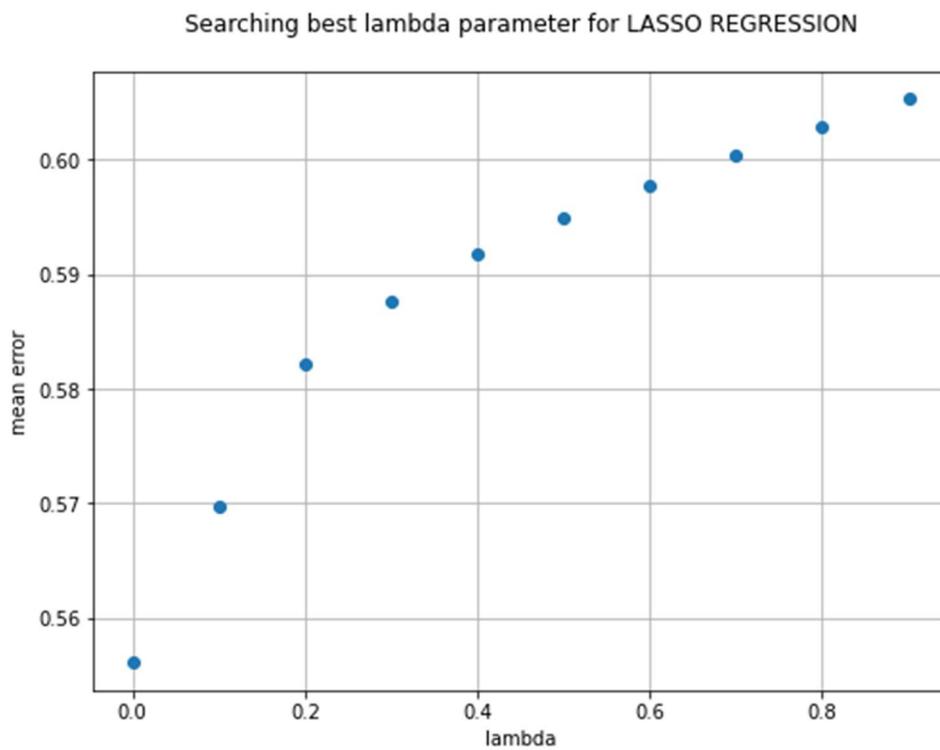


Figure 5.15: Best Lambda for LASSO Regression

From Figures 5.11 and 5.12, it possible to note some swirling motion of the flow.

In this case the error between the approximation and the PIV (Figure 5.14) is lightly more relevant than before. The reasons can be linked to the value of the multiplier and to the accuracy of the approximation, being a real case, hence more complex.

The best hyperparameter were searched for each type of regression performed, for the linear is $\lambda = 0$. It resulted that even for RIDGE and LASSO regressions the best lambda is null, then the best regression turns out to be linear.

Since in the case of LASSO, even $\lambda = 1$ gives an acceptable mean error, the RBF approximation with LASSO regression and the error between this one and the PIV are evaluated; the error is similar to the one given by the linear regression.

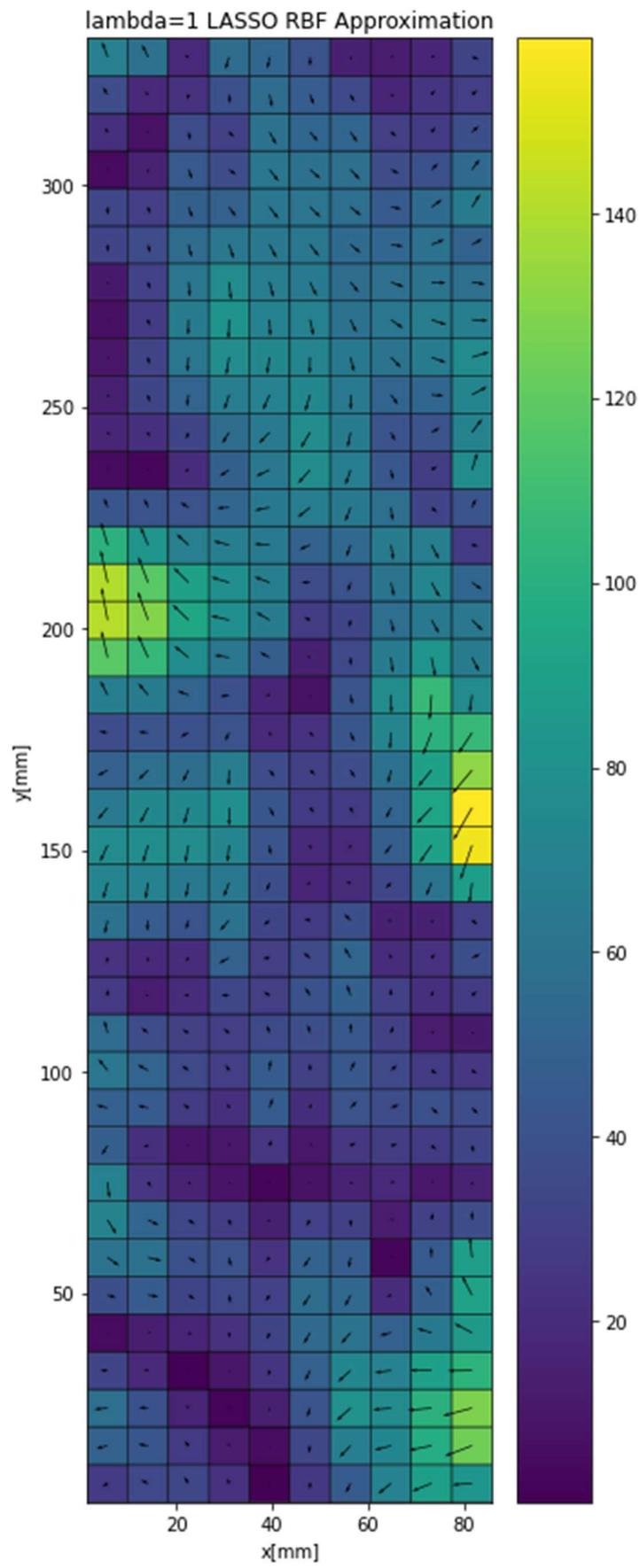


Figure 5.16: Channel Flow. Best lambda Approximation with LASSO

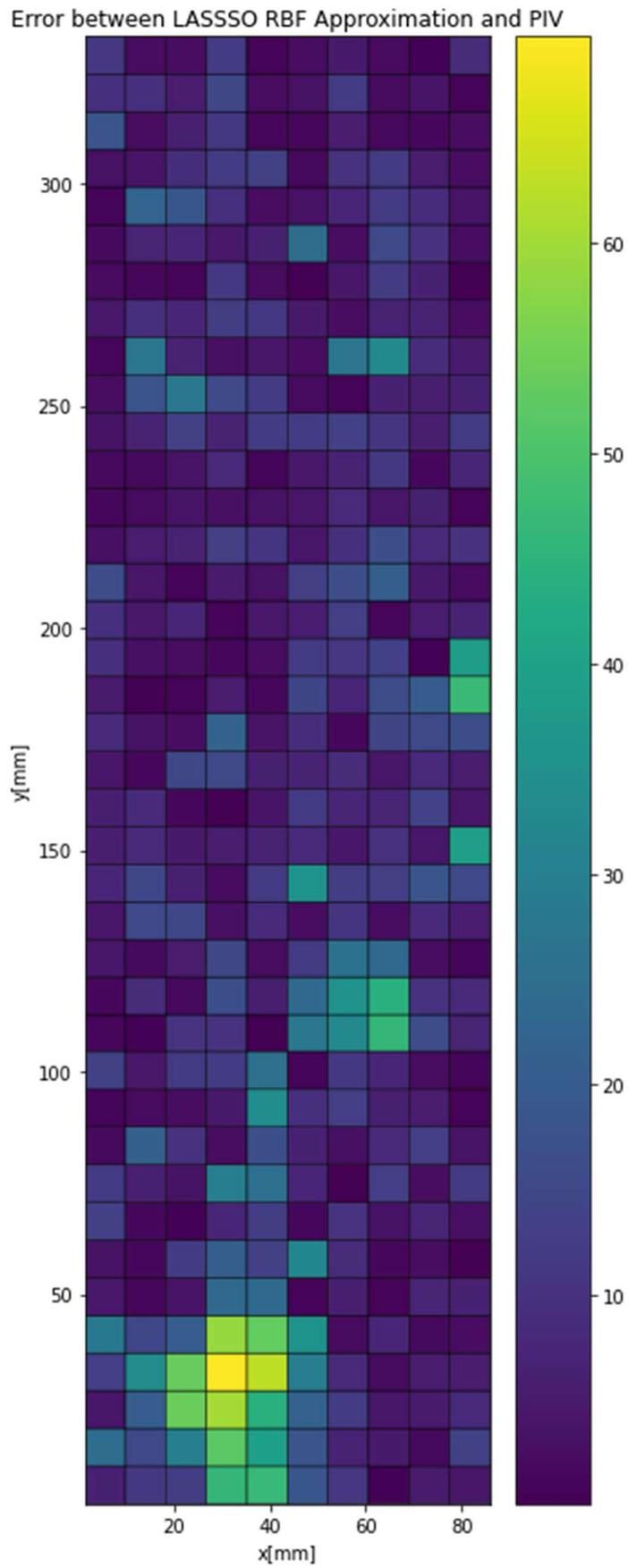


Figure 5.17: Channel Flow. Error between LASSO and PIV

Chapter 6

6. Conclusion and Future Work

The regression and approximation methods analyzed are the beginning of a revolution called Machine Learning, because improving the PIV-based velocity regression for real cases is a purpose of interest. At the moment, the function approximation of the velocity works very well in increasing the resolution and filter noise.

However, nowadays PIV finds application, especially in research centres whilst industrial applications are the objective.

Actually, it has not yet reached the industrial level because there are many issues that don't allow the generalization of this technique to industrial scales.

The principal barrier to the development of this technique are the noise that affects the PIV images and the fact that the definition of appropriate boundary conditions for each problem to evaluate the pressure fields is very challenging [4].

There is a great interest in techniques that estimate pressure from flow velocity, because this can provide many other important fluid dynamic quantities, completely characterizing the bodies considered. Algorithms on PIV-based pressure evaluation concern also spatially and temporally resolved information, critical dynamic phenomena such as aerodynamic loading, wake flow dynamics, and fluid-structure interaction.

The most diffused techniques use the Navier–Stokes equations, where all velocity terms can be measured directly through PIV technique, and solve for the pressure term with the Poisson formulation. Right now, other kind of methods are being implemented to integrate the pressure.

At the moment, industries have reached the RBF mesh morphing [13], a method that allows to update the shape of a model without rebuilding a new mesh. The aim

of the RBF morph technology is to perform fast mesh morphing using a mesh-independent approach based on state-of-the-art RBF techniques. The use of such a technology allows CFD users to perform shape modifications.

But the revolution is represented by meshless methods using Machine Learning tools, function approximation with RBF and constrained regressions and optimization, currently in development in research centres.

For the diffusion of industrial applications a balance between cost and performance has to be found to quantify how the loss of information in terms of time or space affects the accuracy of pressure estimation methods.

Moreover, the research is looking forward to end up with 3D meshless methods, even if the system calculation will be very expensive.

Bibliography

1. **Prof. Miguel Mendez & Prof. Delphine Laboureur.** *Introduction to Measurement Techniques - Lecture Series.* Bruxelles, Belgium : Von Karman Institute, 2020/2021
2. **Theo Käufer, Jörg König & Christian Cierpka.** *Stereoscopic PIV measurements using low-cost action cameras,* 2021.
3. **Mendez, Miguel Alfonso.** *Python PIV Process – Research Master Lesson.* Bruxelles, Belgium : Von Karman Institute, 2020/2021.
4. **J. W. Van der Kindere, A. Laskari, B. Ganapathisubramani & R. de Kat.** *Pressure from 2D snapshot PIV,* 2019.
5. **Miguel A. Mendez, A. Ianiro, B. R. Noack, S. L. Brunton.** *Data-Driven Fluid Mechanics: Combining First Principles and Machine Learning,* 2021.
6. **Mendez, Miguel Alfonso.** *Data Driven Fluid Mechanics and Machine Learning - Research Master Lectures.* Bruxelles, Belgium : Von Karman Institute, 2020/2021.
7. **<https://www.udacity.com/blog/2020/08/machine-learning-for-data-analysis.html>**
8. **Fasshauer, G. E..** *Solving partial differential equations by collocation with radial basis functions. In Proceedings of Chamonix,* 1997.
9. **Pietro Sperotto, Miguel A. Mendez, Sandra Pieraccini.** *A Meshless Method to Compute Pressure Fields from Image Velocimetry,* 2021.
10. **Barlow, Rae, Pope.** *Low– speed wind tunnel testing,* 2017

11. **Kansa, E. J.** *Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-i surface approximations and partial derivative estimates*, 1990.
12. **Nocedal, J. and Wright, S.** *Numerical optimization. Springer Science & Business Media*, 2006.
13. **M. E. Biancolini, A. Chiappa, U. Cella, E. Costa, C. Groth, S. Porziani.** *Radial Basis Functions Mesh Morphing*, 2020.
14. **Kim, Byeongsoo, Lee, Yongbin, Choi, Dong-Hoon.** *Construction of the radial basis function based on a sequential sampling approach using cross-validation*, 2009.
15. **Dantec Dynamics.** *Smart software for imaging solutions.*
<https://pdf.directindustry.com/>
16. **LabVIEW.** <https://www.ni.com/it-it/shop/labview.html>
17. **A. Sciacchitano.** *Uncertainty quantification in particle image velocimetry*, 2019.
18. https://www.piv.de/piv/measurement_principle/
19. **C. Brossard, J.C. Monnier, P. Barricau, F.X. Vandernoot, Y. Le Sant, et al.** *Principles and applications of particle image velocimetry. Aerospace Lab, Alain Appriou*, 2009, p. 1-11. fihal-01180587f
20. **Willert, Christian E..** *Adaptive PIV processing based on ensemble correlation*, 2008.