DOCTORAL THESIS

Improvements in full field techniques for the measurement, simulation and analysis of confined flows in complex geometry.

Gabriel Usera Velasco

Department of Mechanical Engineering

UNIVERSITAT ROVIRA I VIRGILI

Gabriel Usera Velasco

Improvements in full field techniques for the measurement, simulation and analysis of confined flows in complex geometry.

DOCTORAL THESIS

Supervised by
Dr. Josep Anton Ferré and
Dr. Anton Vernet

Department of Mechanical Engineering

UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2009

UNIVERSITAT
ROVIRA I VIRGILI

Departament d’Enginyeria Mecànica

Av. Països Catalans, 26
43007 Tarragona
Tel. 977 55 96 02
Fax. 977 55 96 91

I, Antón Vernet, associate professor in the Department d’Enginyeria Mecànica of the Rovira i Virgili University,

CERTIFY:

That the present study, entitled “Improvements in full field techniques for the measurement, simulation and analysis of confined flows in complex geometry”, presented by Gabriel Usera Velasco for the award of the degree of Doctor, has been carried out under my supervision at the Department of Mechanical Engineering of this university.

Tarragona, 17 June 2009

Table of contents

I. Introduction and Overall Conclusions

Improvements in full field techniques for the measurement, simulation and analysis of confined flows in complex geometry

G. Usera (gusera@fing.edu.uy) *
IMFIA, Facultad de Ingeniería, Universidad de la República, J. Herrera y Reissig 565, 11300 Montevideo, Uruguay

Departament d'Enginyeria Mecánica, Universitat Rovira i Virgili, Av. Paisos Catalans 26, 43007 Tarragona, Spain

Abstract.
This thesis deals with the improvement of techniques for the detailed analysis, experimental measurement and numerical simulation of flows that develop in time and space, confined within complex geometry. On the experimental front, interrogation algorithms for time resolved particle image velocimetry (PIV) image processing were designed and implemented. Special attention was given to the treatment of the near wall portions of the images to ensure appropriate resolution of the boundary layer [1, 5] by proper placement of the displacement vector when the interrogation area overlaps the image boundary. To produce numerical simulations, a three dimensional finite volume solver for the Navier-Stokes equations was created. This numerical model works with block structured grids and sliding interfaces, allowing the treatment of rather complex geometries [2]. For computational efficiency this solver has been fully parallelised for shared memory systems using OpenMP. The set of techniques was completed devising a novel conditional sampling method based on fuzzy clustering for the analysis of large-scale dynamics in turbulent flows [3]. A set of ensemble averages is produced and the time evolution of the flow can be analysed through the most probable sequence in time of the conditionally sampled averages. A case study was selected to drive the evolution of these techniques, concerning the flow inside a printed circuit board (PCB) enclosure. While the present work is restricted to the consideration of the velocity field, it belongs to a wider research effort aiming at both the dynamical and thermal fields within the PCB enclosure. The complete pack of techniques was applied to the description of the spatial and temporal evolution of this flow [4]. Special attention was given to flow structures that could hinder heat dissipation from the (PCB) surface.

Keywords: PIV, Finite Volume, Fuzzy Clustering, confined flows

A Flo, Maite y Sofi; nuestra pequeña Sofi.

[*] Supported by FPI/DPI2003-06725-C02-01 from DGI, Ministerio de Educación y Cultura y Fondos FEDER, Spain, and grant S/B/CE/33/07 from PDT and an I+D grant form CSIC, Uruguay.

1. Introduction

This thesis deals with the improvement of techniques for the detailed analysis, experimental measurement and numerical simulation of flows that develop in time and space, confined within complex geometry boundaries.

The main goal was to achieve the capacity to produce, both experimentally and numerically, and to analyse, data sets that span both spatial and temporal domains. These data sets would ideally contain complete descriptions of the velocity field, as well as other fields as, for instance, the temperature.

We are referring here to the concept of full field techniques [6], extended to both experimental observations and numerical simulations as means of obtaining such data sets. In the case of the velocity field, a full field technique would ideally provide instantaneous and simultaneous observations, spanning the full domain of interest at high enough spatial resolution and sampling rates to capture all flow details. These requirements depend on each particular flow and are still out of reach for turbulent flows at arbitrary large Reynolds numbers [7]. However, for low to moderate Reynolds numbers these requirements are less of an utopia.

1.1. General Objective

To develop an integrated system for the experimental measurement, numerical simulation and detailed analysis of transient confined flows in complex geometry.

1.2. Specific Objectives

1. To implement interrogation algorithms for time resolved PIV of confined flows, that properly treat the near wall region portions of the flow.

2. To construct a numerical flow solver for three dimensional, confined flows in complex geometry.

3. To produce an analysis technique that can synthesise the time history of the flow from data sets produced by the aforementioned techniques.

4. To asses the validity of this integrated approach by considering the case of the flow within a printed circuit board (PCB) enclosure model.

1.3. Methodology

To achieve the aforementioned goals, the work plan was divided in four stages. The first three modules were devoted to the development of the experimental, numerical and analysis techniques, keeping in mind the selected study case. Each development was tested and validated on its own, with frequent reference to this study case.

Finally all three techniques were applied jointly to examine the flow within the PCB enclosure model, providing an overall validation of the approach.

The next three sections present an overview of the techniques and the developments sought in each case. These are followed by a fourth section describing the study case and the application of these techniques to it.

1.4. Overview of the Experimental Technique

On the experimental front, time resolved PIV is one of the techniques approaching the requirements of a full field technique. Both sampling rates and spatial resolution have been enhanced greatly in the last decade, owing to hardware developments as well as better interrogation algorithms.

The sampling rates are largely limited only by the available hardware, mainly fast digital cameras and powerful pulsed Laser sources. This hardware is in turn rapidly evolving. Still some specialisation of the interrogation algorithms is required to optimise the information extraction process from time series of digital images, rather than pairs of images. In [5] a strategy for simultaneous interrogation of multiple images was proposed, based on a multiple image correlation approach, that could lead to better signal to noise ratios and thus generally more robust interrogation algorithms.

On the other hand, the achievable spatial resolution depends more evenly on both the hardware and the interrogation algorithms used. A big leap forward came from the introduction, about a decade ago, of efficient iterative image deformation algorithms [8]. These algorithms effectively eliminated the limitations to spatial resolution imposed by the size of interrogation windows in standard algorithms, where a trade off between achievable spatial resolution and obtained signal to noise ratio usually occurred.

Both the standard interrogation strategy and iterative image deformation algorithms were implemented in our system and adapted to automatically process time resolved PIV image strips. These algorithms were used as a basis for the development of the sought enhancements in near wall treatment.

Time resolved PIV recordings were obtained using an 420 pixels by 480 pixels Motion Scope PCI 1000 S digital camera capable of running at 125 and 250 Hz sampling rates for 512 image strips. The size of each image strip was limited by the internal memory of the camera, requiring a file transfer to the host computer at the end of each recording run. Each image strip was stored as single uncompressed AVI file, which was then fully processed sequentially by the interrogation system. Illumination was provided by a Monocrom DPSSL 532 nm pulsed laser source, with a cylindrical lens. Licopode spores were used as seeding material, which absorb water providing almost spherical, neutrally buoyant, particles.

Within our work, the consideration of PIV technique improvement was focused on the treatment of the interrogation process near wall boundaries. A novel correction method was devised, based on the accurate computation of the actual placement of the velocity vector estimation. This correction activates when the interrogation area overlaps the image boundary, and is based on the consideration of placement of the centre of mass of the truncated interrogation area. It thus enhances the accuracy in near boundary regions [1, 5].

Remarkably, this concept has quickly found its way through the specialised literature and was picked up by other authors [9] which found it useful.

1.5. Overview of the Numerical Technique

Turning now to numerical simulations, the availability of modern higher order methods for turbulent flows, like Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS), provides the opportunity to produce numerically full field data sets for low to moderate Reynolds number. In fact, due to the spectral nature of many numerical methods for DNS, it can be argued that the solutions provided by those methods are continuous in the spatial domain, although of course discrete and truncated in frequency domain.

The application of these techniques, and mainly LES, to engineering like problems requires numerical methods that can handle complex geometries efficiently and that can exploit the opportunity for parallelism available in modern computational hardware, as for instance the multi-core processors.

These requirements have been taken into account by integrating, into a finite volume method, different state of the art techniques. A block structured body fitted grid strategy was used, adapting the linear solvers to work with the resulting block heptadiagonal matrix structure [10]. Also, powerful algebraic multi-grid solvers were adapted to work with these block structured coefficient arrays [11]. To mitigate the effect of grid non-orthogonality in complex geometries, improved linear interpolation methods for non-orthogonal grids [12, 13] were devised and incorporated. The flow solver was instrumented with special grid block interfaces that allow for local grid refinement as well as treatment of sliding interfaces. Finally, the block structured framework was exploited for parallelisation in shared memory systems through the OpenMP programming model.

This effort leaded to the creation of numerical model *caffa3d.MB* [2], which evolved form a classical 2D version published by M. Peric [14]. The model *caffa3d.MB* has now been referred to in new editions of Peric's work [15].

The numerical model *caffa3d.MB* was implemented in Fortran 95, using OpenMP compilation directives for parallelisation. Further conditional compilation directives have been used to modularise the code, enabling integration of various physical models. The numerical simulations presented in [2] and [4] were run in several different hardware systems, among which the most powerful was a two processor SUN Fire V20z Server. This system runs AMD Opteron 248 processors, with 1 MB cache and 2 GB of RAM. It should be noted however that the OpenMP parallelisation strategy employed in this work is specially well suited for the new generation of multi core processors developed in recent years.

A few years ago, Dr. Zaleski suggested that fluid dynamics research would benefit from the availability of more public and open source codes [16]. I fully agree with this view and welcome the use and further development of caffa3d.MB by other researchers. For this purpose the code is made freely available to the academic community, through the website [1].

1.6. Overview of the Analysis Technique

The aforementioned experimental and numerical techniques provide the fluid dynamic researcher with databases with increased level of flow detail in the form of extended time series of velocity fields and possibly other fields as well. To efficiently use these types of data, specialised analysis methods need to be developed. Moreover, systematic comparison between experimental and numerical results for the transient dynamics of the flow needs to be done, for example to validate numerical simulation methods against time resolved PIV. Comparing time averaged fields or standard statistical properties will often not suffice.

[1] www.fing.edu.uy/imfia/caffa3d.MB

Some of the widely used techniques that can be found in the literature are, among others, the Proper Orthogonal Decomposition (POD) [17], the Pattern recognition (PR) [18], and, to a lesser extent, the Singular Spectral Analysis (SSA) [19], the Coherent Vortex Simulation (CVS) [20], and Fuzzy Clustering (FC) [21].

Proper orthogonal decomposition (POD) involves the computation of a subset of the eigenvectors and eigenvalues of the correlation tensor, in decreasing order of the eigenvalues magnitude. The eigenvectors with the largest eigenvalues can give an insight into the underlying large-scale structure of the flow [24]. In addition, the projection of the original data set onto the subset of eigenvectors, allows for reducing the dimensionality of the data set with a loss of information limited to the finer, and thus less energetic, scales of the flow. This is actually the way in which POD is used within this work, previous to the application of Fuzzy Clustering.

On the other hand, Fuzzy Clustering (FC) is a technique that allows the classification of all the instantaneous frames into a specified number of conditionally averaged or coherent subsets [21]. This technique can identify flow structures whose existence remains hidden within the usual unconditional averaging procedure [3].

The novel conditional sampling method that was devised, relays on (POD) to reduce the dimensionality of the data set and then on (FC) to produce a set of ensemble averages, characterising also the time history of the flow through the most probable sequences in time of those ensembles. This leads to a picture of the principal coherent structures in the flow and their evolution in time [3].

1.7. Overview of Study Case

The study case selected to drive our work was that of the flow inside a printed circuit board (PCB) enclosure model. The overall aim of the ongoing research that hosts this thesis work, is to investigate the velocity field, and its influence on the heat transport by natural and/or forced convection, in complex domains like those encountered in computers or other electronic subsystems with printed circuit boards (PCB). Two examples of such PCB that could require special attention to heat dissipation conditions encountered within their enclosures are pictured in figure 1.

A sketch of the experimental model is shown in figure 2. The entrance and outlet openings have square sections of side $h = 0.024m$. The upper and lower channels are $6h$ long by $3h$ wide, and of height h each. The central plate is only $5h$ long, leaving a passage from the upper channel to the lower channel.

Figure 1. Two examples of printed circuit boards that could require heat dissipation within their enclosures

For convenience water was used instead of air to run the experiments. Application of the PIV technique is generally simpler for water flows. Also the reduced kinematic viscosity of water with respect to air, enables to easily reach higher Reynolds numbers.

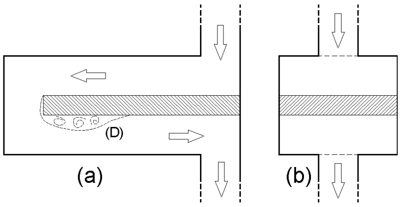


Figure 2. Sketch of experimental model with main paths of flow. (a) Side view with separation region (D) shown at the lower channel entrance. (b) Front view

An image of the experimental setup used for the time resolved PIV experiments is presented in figure 3. The Plexiglas model is shown at the front and the PIV camera at the back, mounted on a manual traversing system. The Laser source (not seen in the picture) is located to the right of the enclosure model.

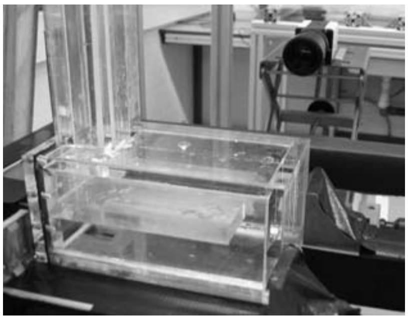


Figure 3. Image of experimental setup

Figure 4 presents a graphical representation of the block structured grid designed for this case [2]. Spatial resolution was set to $h/70$ near the walls and about $h/35$ at the core, using five stretched grid blocks, for a total of 3.1×10^6 nodes. This configuration occupies 1 GB RAM and was run on the two processor systems mentioned in 1.5.

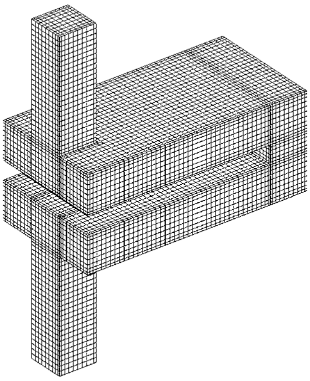


Figure 4. Block structured grid (adapted from [2]).

A general view of the flow is given in figure 5, where vortical structures are shown through visualisation of λ_2 iso-surfaces [28]. It can be seen that the flow in the upper channel is dominated by two counter rotating vortical structures. These structures break down at the entrance of the lower channel. The non stationary characteristic of the flow is also apparent in figure 5.

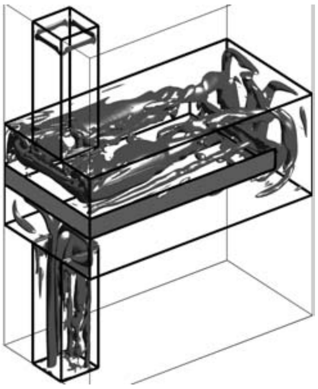


Figure 5. Instantaneous vortex structures, identified by means of λ_2 iso-surfaces. Adapted from [2]

The experimental and numerical simulation results obtained, as well as their analysis, are described in detail in [4]. Ongoing research on the dynamics of this flow, studying the influence of subtle changes to the geometry of the enclosure and the development of thermal field has been taking now on by Sylvana Varela as the matter for her PhD thesis [29]

1.8. Closing remarks for this introduction

The structure of this document reflects the systematic effort devoted to publish the results of our work. Following this introduction and the summary of conclusions, the thesis develops in four chapters. Each chapter is a self contained piece presented here in exactly the same way as it appeared published. Not even the page numbering was modified. [2] While this effort consumed quite a lot of time (publishing in Journals is usually a, painful, time consuming task), I do not regret having chosen to do so.

Chapter 1 deals with the design, implementation and validation, of the analysis technique based on fuzzy clustering to produce a set of ensemble averages. Chapter 2 presents the correction method devised to enhance the performance of PIV interrogation algorithms near wall boundaries. Chapter 3 concludes the presentation of the techniques, describing the development of the numerical model *caffa3d.MB* and its fundamental characteristics. Its validation against selected cases is also contained in this chapter. Finally, in chapter 4, all three techniques are put to work together applied to the selected study case.

This logical order for the presentation differs slightly from the chronological order of publication, since the contents of chapter 4 were written and published before those of chapter 3.

The work reported in these four chapters was carried out by Gabriel Usera, with the guidance of thesis supervisors Josep Anton Ferré and Anton Vernet. For chapter 1, we had the fundamental contribution of Jordi Pallares, whom developed the numerical simulation results for turbulent flow in a square channel [25, 26] used in that chapter to devise the conditional sampling technique.

The paper from chapter 1 was published in the *European Journal of Mechanics B/Fluids*, which currently holds a 5-year Impact Factor of 1.012, and ranks in place 72 out of 112 journals in category *Mechanics* with Aggregate Impact factor of 2.040.

The two papers from chapters 3 and 4 were published in *Flow, Turbulence and Combustion Journal*, whose 5-year Impact Factor is 1.193, and ranks 21 out of 43 journals in category *Thermodynamics* with Aggregate Impact factor of 1.132.

[2] This was in fact a condition requested by some of the Journals in order to grant permission for the articles to be used within the thesis.

2. Summary of Conclusions

In this section, the main conclusions, arising from the work contained in the four papers that follow, are summarised pointing out the original and/or most relevant achievements.

A novel conditional sampling technique was developed [3], which uses proper orthogonal decomposition and fuzzy clustering to identify coherent structures within the flow and analyse their time history. The proposed method holds two main advantages against other conditional sampling techniques, like for example pattern recognition. First, it involves all of the instantaneous fields in the time series instead of a typically small percentage of those. Also, unlike other techniques that are insensitive to the time ordering of the instantaneous field, it delivers a probabilistic map of transitions between the different ensemble averages which exposes a simplified time history of the flow. In this sense, it is somehow analogous to a phase averaging technique, but for non-periodical flows.

As for the time resolved PIV technique, it was shown that the standard interrogation procedure gives distorted results near image boundaries. This is due to the cropping of the interrogation area and of the weight function which implies a misplacing of the velocity vector estimate [1]. A simple, effective and computationally cheap correction method was devised, based on a first order approximation of this distortion, through the computation of the actual placement of the cropped weighting function centre of mass. The correction is applied after the interrogation process, not interfering with it. Quantitative tests showed that the error level in near boundary regions was reduced to the same level found far away from boundaries, clearly outperforming standard algorithms in this aspect. This enhancement can be specially significative in the study of confined flows, where boundary layers play a most important role.

The trilogy of technique developments was completed with the implementation of a numerical model for the simulation of transient three dimensional flows, based on the finite volume method with block structured body fitted grids [2]. Several state of the art techniques were incorporated to this flow solver. An improved linear interpolation formula for non-orthogonal grids, based on multidimensional Taylor series expansions, was devised and shown to preserve second order behaviour in non-orthogonal grids. The devised formula, casted in a compact expression involving mixed vector products, was in fact found to be equivalent to a previously published one [13]. The block structured grid framework was also the basis for parallelisation through OpenMP, whose performance was found to be dependant onto hardware implementations, presumably due to memory access bottleneck. To speed up pressure equation iterations, an algebraic multi-grid solver [11] was adapted into the same framework.

When the three techniques were applied together to the analysis of the study case, the validity of the fuzzy clustering conditional sampling technique was shown [4]. The application of this technique allowed the comparison of time resolved PIV measurements with transient numerical simulations, beyond that of mean statistical properties, so that the transient behaviour of the flow can be compared also. Specifically, the analysis revealed that not only the mean flow structures were well captured by the numerical model, but also the details of their time evolution were represented correctly.

Time passed by now, since the beginning of this work, and these techniques have proven useful not only to this author and his thesis, but also to other researchers and subjects [9, 15, 29, 30, 31, 32, 33]. That is about as much as I could have expected seven years ago.

2.1. *caffa3d.MB* AROUND THE WORLD

The first version of *caffa3d.MB* was available online by November 2004. Since then, more than 50 people around the world have decided to download it and, eventually, use it. No general effort has been done to follow up the experience of this people with the model, whether they found it useful at all or not. Some of them have spontaneously contacted back with comments, questions and even in some cases they have contributed new routines that now form part of a better *caffa3d.MB*. To express my gratitude to them all, the following pictures and tables summarise their names and affiliations.

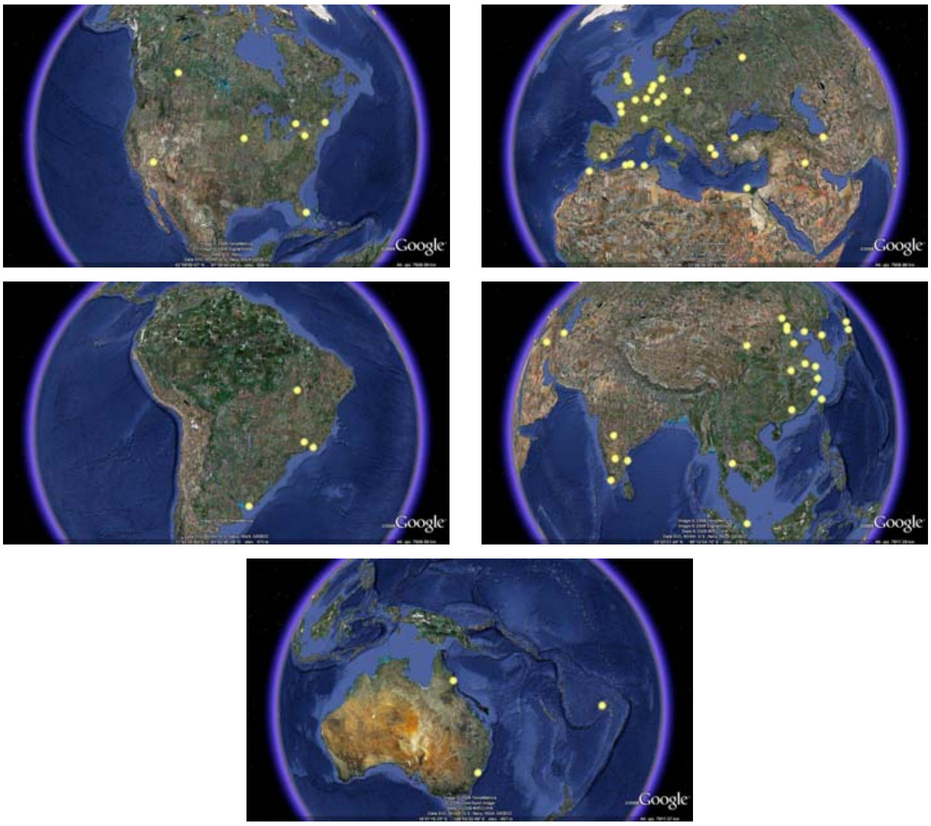


Figure 6. Sites that requested a copy of caffa3d.MB. 2004-2009

Table I. People and places around the world that downloaded 'caffa3d.MB'. 2004-2006

date : name : affiliation
22-Nov-2004 : Peiying Hsieh : Sr R&D Engineer, Bayer HealthCare, Germany
01-May-2005 : Tay Wee Beng : PhD student, National University of Singapore
19-May-2005 : Bensid Nadia : Universite de Constantine, Algeria
18-Jun-2005 : Desmond Aubery : PhD student, Chulalongkorn University, Thailand
18-Jun-2005 : Anindya Sengupta : DAE, Iowa State University, USA
18-Jun-2005 : Shibo Kuang : PhD student, SMSE, University of New South Wales, Australia
19-Jun-2005 : Merrouche Djemai : PhD student, University of Boumerdes, Algeria
19-Jun-2005 : Seok-Ki Choi : Korea Atomic Energy Research Institute
20-Jun-2005 : George Kosmadakis : National Technical University of Athens, Greece
20-Jun-2005 : John Dong : University of new south wales, Australia
20-Jun-2005 : Keyur Vyas : Dept. of Civil and Building Eng, Loughborough University, UK
23-Jun-2005 : Hongping Hu : IMST, Central South university, China
24-Jun-2005 : Simao Marques : University of Salford, UK
27-jun-2005 : D.Venkatesan : IIT-Madras
29-Jun-2005 : Enrum Anbudan : Regional Research Laboratory(CSIR), Trivandrum
07-Jul-2005 : Lijian Sun : University of Nevada, Las Vegas, USA
11-Jul-2005 : Zbigniew Mindziul : University of Technology Poland
16-Jul-2005 : Khaled Ben Nasr : PhD student, Technical University Hamburg-Harburg, Germany
23-Sep-2005 : Bystrik Cervenka : PhD student, Prvni Brnenska Strojirna, Velka Bites
26-Sep-2005 : Liu Huafei : PhD student, University of Science and Technology Beijing, China
23-Nov-2005 : Qinfu Hou : PhD student Northeastern University of China
25-Nov-2005 : Liu Cheng : PhD student, Dalian University of Technology
25-Nov-2005 : Zhang Chaolei : Jiangxi science and technology college, China
29-Nov-2005 : Jianhu Nie : University of Nevada, USA
30-Nov-2005 : Zhanfeng cui : WuHan University, WuHan city, China
29-Dec-2005 : Jian Ye : PhD student, SJP, Beijing Univ of Aero. and Astro, China
11-Jan-2006 : Ludovic Chatellier : Professor, LEPTEA, Universit de la Rochelle, France
06-Feb-2006 : Arthur Stueck : PhD student, FDST, Hamburg University of Technology, Germany
29-Mar-2006 : Gongal : PhD student, Tsinghua University
04-Apr-2006 : Bin Li : PhD student, University of aeronautics and astronautics of Beijing
17-Apr-2006 : Matthias Aechtner : PhD student, F. A. University Erlangen, Germany
18-Apr-2006 : Shumei Lou : Shandong University,Shandong, China
03-Jun-2006 : John Matthew : Past sessional lecturer at University of Alberta
09-Aug-2006 : Rohallah Tavakoli : Sharif University of Technology
17-Aug-2006 : Ravindra Pardeshi : Indian Institute of Science, Bangalore
09-Sep-2006 : Liu Zhaohui : Huazhong University of Science and Technology, China
23-Oct-2006 : Dornelles Vissotto Junior : Professor, Universidade Federal do Parana, Brasil
14-Nov-2006 : Xing Gailan : University of Science and Technology, Beijing, China

Table II. People and places around the world that downloaded 'caffa3d.MB'. 2007-2009

25-Ene-2007	Minghao Yuan	Shanghai Jiao Tong University, China
31-Jan:2007	Jutsusho Ryu	PhD student, Tohoku University, Japan
23-Feb-2007	Dimitris Fidaros	University of Thessaly, Volos, Greece
30-Apr-2007	Antonio Pons	University of Granada, Spain
08-May-2007	Djeghri Noureddine	ENP, Polytechnic School of Algiers
15-May-2007	Simon Nava	Professor, Florida International University, USA
16-May-2007	Ahmed Aly El-din	Alexandria University
22-May-2007	Muataz Steita	PhD student, University Tor Vergata, Roma, Italy
31-May-2007	Sylvain Faure	Paris-Sud XI Universty, France
26-Jun-2007	Jutsusho Ryu	Tohoku University, Japan
14-Aug-2007	Dong-Jun MA	Pennsylvania State University, USA
05-Oct-2007	A. Khourchafi	Univeristy Hassan II, Casablanca, Marruecos
22-Oct-2007	Kahlil Lasfer	University ENIM, France
16-Nov-2007	Stephane Zaleski	UPMC, France
20-Nov-2007	Abed Bouabdellah	Ecole Centrale de Nantes, France
26-Nov-2007	Jaime Peon	Profesor, Universidad de Vigo, Spain
07-Dec-2007	Gonski Kuang	University of Queensland
04-Jan-2008	Marcio de Oliveira	UNIVERSITAS - Itajuba -MG - Brasil
06-Feb-2008	Duane Baker	Atomic Energy of Canada Ltd.
09-May-2008	Cy Li	South China university of technology
14-May-2008	Ying Wei	Nanjing University of Aeronautics and Astronautics
14-May-2008	Asu	Tsinghua University, Beijing, China
14-May-2008	Sergei Strijhak	Moscow Technical State University, Russia
24-Jun-2008	Huseyin Yaltirik	Marmara University Faculty of Engineering
06-Jul-2008	Bhargav Lalith	Gulbarga University, Karnataka, India
17-Jul-2008	Mohsen G. Gawy	Babylon university
29-Jul-2008	Naseem Uddin	Instiute of Aerospace Thermodynamics, Stuttgart, Germany
09-Oct-2008	Arturo Gonzalez	PhD student, Ruhr Universitaet Bochum, Germany
21-Oct-2008	Antoine Rousseau	Professor, INRIA, France
27-Nov-2008	Bao-Feng Ma	Beihang University, China
20-Dec-2008	Rajnesh Lal	The University of the South Pacific, Fiji
25-Feb-2009	Pavel Nikitenko	Moscow Aviation Institute (Technical University)
11-Mar-2009	David S. Greeley	Applied Physical Sciences Corporation, Lexington, USA
16-Mar-2009	Renato Nascimento Elias	Federal University of Rio de Janeiro, Brazil

Acknowledgements

The work reported here was supported by FPI/DPI2003-06725-C02-01 and DPI2006-02477 from DGI, Ministerio de Educación y Cultura y Fondos FEDER, Spain, and grant S/B/CE/33/07 from PDT and an I+D grant from CSIC, Uruguay.

My thesis supervisors, Josep Anton Ferré and Anton Vernet, have supported me always and well beyond research matters, specially through the rough times. My deepest gratitude to both.

References

1. Usera G., Vernet A., Ferré J. A., Boundary treatment in PIV image analysis. *Models, experiments and computation in turbulent flows, R. Castilla, E. Oñate and J.M. Redondo (Eds.) CIMNE, Barcelona, Spain* (2007) ISBN: 978-84-96736.
2. Usera G., Vernet A., Ferré J. A., A Parallel Block-Structured Finite Volume Method for Flows in Complex Geometry with Sliding Interfaces. *Flow, Turbulence and Combustion*, 81 (2008) 471–495.
3. Usera G., Vernet A., Pallares J., Ferré J. A., A conditional sampling method based on fuzzy clustering for the analysis of large-scale dynamics in turbulent flows. *European Journal of Mechanics - B/fluids*, 25(2) (2006) 172–191.
4. Usera G., Vernet A., Ferré J. A., Use of time resolved PIV for validating LES/DNS of the turbulent flow within a PCB enclosure model. *Flow, Turbulence and Combustion*, 77 (2006) 77–95.
5. Usera G., Vernet A., Ferré J. A., Considerations and improvements of the analysing algorithms used for time resolved PIV of wall bounded flows *Proceedings of the 12th international symposium on applications of laser techniques to fluid mechanics* Lisbon, July 11-15
6. GonhXing-Shen On the approach to ideal observation & measurement techniques for fluid mechanics (and introduction of some recent work at BUAA) *Proceedings of the SPIE*, 3172 (1997) 540–550
7. Jiménez J. Computing high-Reynolds-number turbulence: will simulations ever replace experiments? *Journal of Turbulence*, 4(1) (2003) 22–30 (2003).
8. Nogueira J., Lecuona A., Rodríguez P. A., Local field correction PIV, implemented by means of simple algorithms and multigrid versions. *Measurement Science and Technology*, 12 (2001) 1911–1921.
9. Theunissen R., Scarano F., Riethmuller M. L. On improvement of PIV image interrogation near stationary interfaces. *Experiment in Fluids*, 45 (2008) 557–572.
10. Lilek Z., Muzaferija S., Peric M., Seidl V., An implicit finite-volume method using nonmatching blocks of structured grid. *Numerical Heat Transfer, Part B*, 32 (1997) 385–401.
11. Mora Acosta J., Numerical algorithms for three dimensional computational fluid dynamic problems. *PhD Thesis, Universidad Politécnica de Catalunya*, (2001).

12. Lehnhauser T. and Schafer M., Improved linear interpolation practice for finite-volume schemes on complex grids. *International Journal for Numerical Methods in Fluids*, 38 (2002) 625–645.
13. Lehnhauser T., Ertem-Muller S., Schafer M., Janicka J., Advances in numerical methods for simulating turbulent flows. *Progress in Computational Fluid Dynamics*, 4(3-5) (2004) 208–228.
14. Ferziger J. H., Peric M. Computational Methods for Fluid Dynamics *Springer, Berlin* (2002)
15. Ferziger J. H., Peric M. Numerische Stromungsmechanik *Springer, Berlin* (2008)
16. Zaleski S., Science and Fluid Dynamics should have more open sources. *http://www.lmm.jussieu.fr/ zaleski/OpenCFD.html*, (2001)
17. Berkooz G., Holmes P. and Lumeley J. L., The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review Fluid Mechanics*, 25 (1993) 539–575.
18. Kopp G. A., Ferré J. A. and Giralt F., The use of pattern recognition and proper orthogonal decomposition in identifying the structure of fully-developed free turbulence. *Journal of Fluids Engineering*, 119 (1997) 289–296.
19. Ghil M., Allen M. R., Dettinger M. D., Ide K., Kondrashov D., Mann M. E., Robertson A. W., Saunders A., Tian Y., Varadi F. and Yiou P., Advanced spectral methods for climatic time series. *Reviews of Geophysics*, 40 (2002) 1.
20. Farge M., Schneider K., Pellegrino G., Wray A.A., Rogallo R.S., Coherent vortex extraction in three-dimensional homogeneous turbulence: Comparison between CVS-wavelet and POD-Fourier decompositions. Physics of Fluids, 15 (10) (2003) 2886–2896.
21. Vernet A., Kopp G. A., Classification of turbulent flow patterns with fuzzy clustering. *Engineering Applications Of Artificial Intelligence*, 15:(3-4)(2002) 315–326.
22. Vernet A., Kopp G. A., Ferré J. A. and Giralt F., Three-dimensional structure and momentum transfer in a turbulent cylinder wake. *Journal of Fluid Mechanics*, 394 (1999) 303–337.
23. Xie X. L., Beni G., A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-13:(8) (1991) 841–847.
24. Reichert, R. S., Hatay, F. F., Biringen, S. and Huser, A., Proper orthogonal decomposition applied to turbulent flow in a square duct. *Physics of Fluids*, 6(9) (1994) 3086–3092.
25. Pallares J., Davidson L. Large-eddy simulation of turbulent flows in stationary and rotating channels and in a stationary duct *Rep 00/03. Department of Thermo and Fluid Dynamics, Chalmers University of Technology Gothenburg, Sweden, 2000 (http://www.tfd.chalmers.se/l̃ada/postscript_files/pallares_tfd00-03.pdf, as of 08/Feb/2005).*
26. Pallares J., Davidson L. Large-eddy simulations of turbulent heat transfer in stationary and rotating square ducts *Physics of Fluids*, 14 (8) (2002) 2804–2816.
27. Jeong J., Hussain F., Schoppa W., Kim J. Coherent structures near the wall in a turbulent channel flow *Journal of Fluid Mechanics*, 332 (1997) 185–214.
28. Jeong J. and Hussain F., On the identification of a vortex. *Journal of Fluid Mechanics*, 285 (1995) 69–94.
29. Varela S., Usera G., Vernet A., Ferré J.A. Experimental studies and numerical simulation of the processes of cooling electronic components by natural convection and forced *Mid-therm thesis work, available online at*

http://www.etseq.urv.es/doctorat/fotos/posters/Varela_Sylvana.pdf, as of April 05, 2009

30. Erturk N., Vernet A., Castilla R., Ferré J.A., Codina E. Time Resolved PIV for Flow Analysis in an External Gear Pump *Third Workshop on Turbulence Research* Sevilla (2008) Ercoftac
31. Ahusborde E., Glockner S. An implicit method for the Navier-Stokes equations on overlapping block-structured grids *International Journal for Numerical Methods in Fluids*, (2009) DOI:10.1002/fld.2044
32. Varela J., Araújo M., Bove I., Cabeza C., Martí A. C., Montagne R., Sarasúa L. G., Usera G. Instabilities developed in stratified flows over pronounced obstacles. *Physica A, Statistical Mechanics and its Applications*, 386 (2007) 681–685.
33. Cabeza C., Varela J., Bove I., Freire D., Marti A., Sarasua L.G., Usera G., Montagne R., Araujo M. Two-layer stratified flows over pronounced obstacles at low-to-intermediate Froude numbers *Physics of Fluids*, 21 (2009) DOI:10.1063/1.3110108

II. **Paper 1** : “A conditional sampling method based on fuzzy clustering for the analysis of large-scale dynamics in turbulent flows”

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

European Journal of Mechanics B/Fluids 25 (2006) 172–191

EUROPEAN JOURNAL OF MECHANICS B/FLUIDS

A conditional sampling method based on fuzzy clustering for the analysis of large-scale dynamics in turbulent flows

G. Usera [a,b], A. Vernet [a], J. Pallares [a], J.A. Ferré [a,*]

[a] *Departament D'Enginyeria Mecànica, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain*
[b] *Instituto de Mecánica de los Fluidos, Universidad de la República, J.H. Reissig 565, 11300, Montevideo, Uruguay*

Received 24 April 2004; accepted 26 June 2005

Available online 27 October 2005

Abstract

A conditional sampling technique, based on fuzzy clustering, is used to educe the organization of the secondary flow motions observed in the large-eddy simulation of a turbulent square channel flow. The data analysed are the multi-valued time series obtained from sampling the secondary velocity components at a fixed cross-section of the channel, over consecutive time steps. The mean values of the secondary flow motion velocities are one order of magnitude lower than their r.m.s values. The purpose of the conditional sampling scheme used is to replace the picture of the secondary flow motions provided by the unconditional time mean with several ensemble averages. In this way the whole variability of the instantaneous data can be split in two parts: one for the difference between the observed ensemble averages, and the other for the variability within each ensemble. Unlike other conditional sampling schemes which sort only part of the data into one or more families depending on an externally fixed condition, the fuzzy clustering approach used here first determines the optimum number of families or clusters and then classifies all the recorded time steps. The results show that the local turbulence intensities of the ensemble averages obtained from fuzzy clustering can be reduced by one order of magnitude. In addition, the classification of all the time steps into several clusters or families enables the large scale dynamics of the educed structures to be analysed.

Keywords: Fuzzy clustering; Conditional sampling; Turbulent flow

1. Introduction

The increasing availability of full field measurement techniques with good time resolution, such as time resolved Particle Image Velocimetry (PIV), and of higher order numerical simulation methods for turbulent flows, such as Large Eddy Simulations (LES) and Direct Numerical Simulations (DNS), provides the fluid dynamic researcher with databases that use extended time series to give more details about the flow. These databases tend to get larger and larger and specialised methods need to be developed if they are to be analysed efficiently. This paper analyses the extended time series of velocity fields in planar cross sections of high turbulence intensity flows. Turbulent flow in a square duct has been selected to illustrate the proposed method of analysis.

[*] Corresponding author. Tel.: +34977559602; fax: +34977559691.
E-mail address: josep.a.ferre@urv.net (J.A. Ferré).

0997-7546/$ – see front matter © 2005 Elsevier SAS. All rights reserved.
doi:10.1016/j.euromechflu.2005.06.002

Turbulent flow in a duct is the prototypical case of turbulent flow along a stream-wise corner, where secondary flows of Prandtl's second kind arise. Despite yielding a weak mean cross-stream velocity field, these secondary flows have a significant effect on flow properties of great practical interest (e.g. wall shear stress, and heat and mass transfer), as has been previously recognised by other authors (e.g. [1–3]). Some of the models used to solve the Reynolds-averaged Navier–Stokes equations are unable to reproduce secondary flows properly, so either LES or DNS are preferred. After more than a decade of numerical experiments on turbulent flow in square ducts, both DNS (e.g. [4,1,2]) and LES (e.g. [5–7]), have been established as valid techniques for gaining further insight into the dynamics of the flow. Once these simulations have been validated against experimental data, they provide a full and detailed picture of the flow, which can then be analysed by the appropriate techniques so that the flow dynamics can be understood better.

Proper orthogonal decomposition (POD) is one such technique that involves computing a subset of the eigenvectors and eigenvalues of the correlation tensor, in decreasing order of the magnitude of the eigenvalues. The eigenvectors with the largest eigenvalues can provide insight into the underlying large-scale structure of the flow in a square channel [8]. Kopp et al. [9] considered using these leading eigenvectors as starting templates for pattern recognition to search for coherent structures. In addition, the projection of the original data set onto the subset of eigenvectors makes it possible to reduce the dimensionality of the data set with a loss of information that is limited to the finer, and thus less energetic, scales of the flow. This is how POD is applied here. A summary of the applications of the POD technique has been given by Berkooz et al. [10]. More recently, Johansson and George [11] and Garnard et al. [12] further considered its application in the study of turbulent flows.

Another orthogonal decomposition technique has been proposed by Farge et al. [13] as an alternative to POD. The coherent vortex simulation (CVS) is a non-linear wavelet-based decomposition technique that adapts the number and selection of resolved modes, and keeps only the strongest wavelet coefficients at each time step. It thus rivals POD, which is a strategy with a fixed number of resolved modes, in terms of compression efficiency.

Pattern recognition (PR) is another set of techniques used to identify flow structures that govern the dynamics of the flow [14,15]. It involves cross-correlating initial templates, or prototypical structures, with the instantaneous events contained in the time series. Those events whose correlation factors exceed a user-defined threshold are accepted to create the ensemble average for the educed structure. However, usually only a small number of the instantaneous events contained in the time series are involved in the pattern classification process. For instance in Kopp et al. [9], the group of classified instantaneous events accounts for only 35–38% of the flow. On the other hand, fuzzy clustering (FC) has recently proved to be a useful technique that can classify all the instantaneous frames into a specified number of conditionally averaged or coherent subsets [16]. This technique is applied here to identify the flow structures that are hidden within the usual unconditional averaging procedure.

The purpose of this paper is to present a novel conditional sampling method based on a fuzzy clustering technique and its application to the analysis of secondary flow structures of the turbulent flow in a square duct. To overcome the limitations of unconditional time averaging, a set of conditionally sampled averages, or cluster averages, are obtained with an improved signal-to-noise ratio. The time evolution of the secondary flow organisation is also analysed, which provides the most probable sequence in time of the conditionally sampled averages. This two-step procedure (conditional sampling and time ordering) finally yields a simplified view of the flow dynamics of the secondary flow motions in the square channel. Following this introduction, the data set used and the analysis techniques will be described in detail in Sections 2 and 3. Section 4 presents some results of the analysis of the cross-stream velocity components, while Section 5 describes the time evolution analysis of the flow. Finally, the conclusions are summarised in Section 6.

2. Working data set

The data analysed in this paper was obtained from the large eddy simulation of the fully developed turbulent flow in a square duct at $Re_\tau = u_\tau h/\nu = 300$, which corresponds to $Re_h = u_b h/\nu = 4500$ [17]. A sketch of the computational domain and the nomenclature for the velocity components is shown in Fig. 1. The domain has an $h \times h$ square cross section and its length is $2\pi h$. In the cross section, the computational grid contained 66 × 66 points, non-uniformly spaced. However, the grid was uniform in the streamwise direction, with 66 nodes and periodic boundary conditions. The computational time step was $\Delta t = 5 \times 10^{-3} h/u_\tau \approx 8 \times 10^{-2} h/u_b$ and the data series was sampled at every time step.

The initial working data set was a time series of 1400 instantaneous, three-component, consecutive velocity fields, recorded at a planar cross section of the square duct in a 66 × 66 grid. Therefore, there were $66 \times 66 = 4356$ velocity

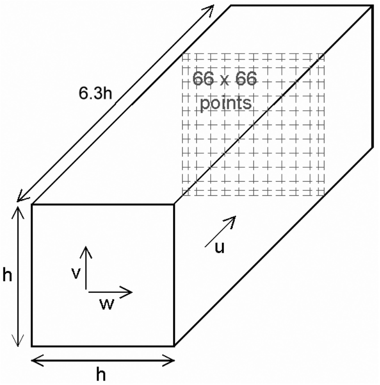


Fig. 1. Sketch of the computational domain and velocity components.

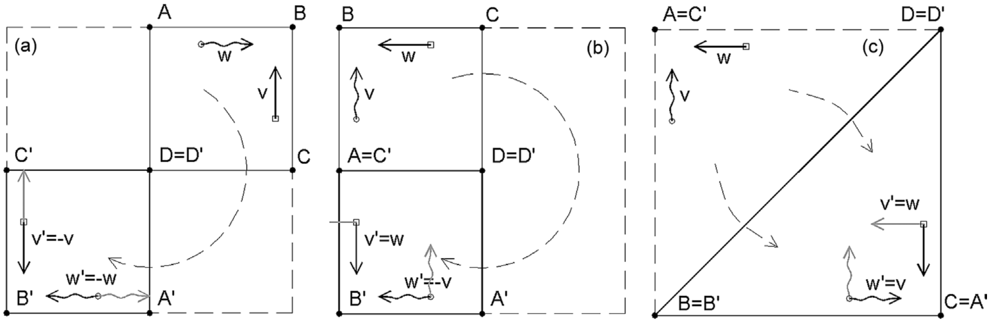


Fig. 2. (a) Transformation of the upper right quadrant into the lower left quadrant. (b) Transformation of the upper left quadrant into the lower left quadrant. (c) Transformation of the upper octant into the lower octant of the lower left quadrant.

vectors per time step. However, since the statistical properties of the flow are expected to be the same in any given octant, the full square duct data was used to generate a time series of $8 \times 1400 = 11200$ instantaneous velocity fields containing the data of the single octants of the square duct. Other authors have already proposed increasing the number of flow samples through this argument of symmetry (see [18]). Each frame corresponded to an octant of the square section, with a reduced number of only 561 velocity vectors per time step. Data from each octant was kept in a continuous sequence of 1400 time steps, in such a way that the time discontinuities of the whole data set were reduced to only 7 cuts in a series of 11200 time steps. The different octants were rotated and flipped and (v, w) were also transformed as needed, to make them all consistent. Only the u component of the velocity remained unchanged throughout these transformations (see Fig. 2). No spurious correlations were introduced using this procedure because the length of each continuous time series segment (1400 time steps) is two orders of magnitude larger than the integral time scale of the flow $T_\tau = 14.9\Delta t$, which is dominated by the streamwise velocity correlation, as can be observed in Fig. 3(c).

Fig. 3(a) presents the vector plot of the mean cross-stream velocity field and a contour plot of the mean stream-wise component. Indirectly, Fig. 3(a) also shows the non-uniform computational grid. Fig. 3(b) shows the mean stream-wise component of the wall shear stress along the wall (normalised by the mean wall shear stress). These results are consistent with those presented by other authors [4,18,1,2]. An extensive comparison between the results of the present study and previous DNS and experimental measurements can be found in Pallares and Davidson [17], and in further detail in Pallares and Davidson [19].

However, the averaged picture of the flow in Fig. 3(a) gives misleading information about the dominant characteristics of the instantaneous velocity fields, since the mean values displayed in Figs. 3(a) and 3(b) are up to one order of magnitude weaker than their respective r.m.s. values. Therefore, in this case one cannot expect to find a single instantaneous velocity field or shear stress profile that looks like the conventional time mean and these mean velocity fields say very little about the true dynamics of the secondary flow motions.

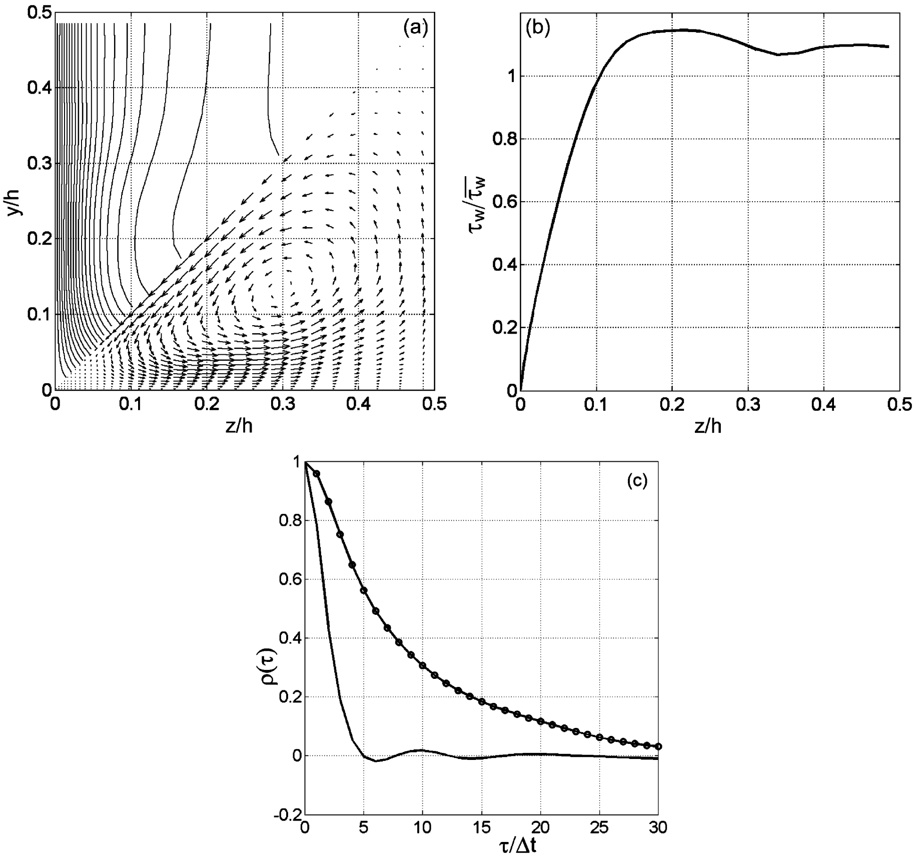


Fig. 3. (a) Octant mean velocity fields: u component contour lines (top) and (v, w) component secondary flow vector field (bottom). (b) Stream-wise component of mean wall shear stress along bottom half of the wall. (c) Auto-correlation coefficient $\rho(\tau)$ for the cross stream velocity field (—) and streamwise velocity (—○—), averaged over the octant section.

Fig. 3(c) displays the time auto-correlation of both the (v, w) cross stream velocity field and the streamwise velocity component. The correlation coefficient was computed at each grid point and then averaged over the octant. It can be observed that the first one goes to zero for a delay of five time steps, and that the time-memory of the (v, w) velocity field beyond that point fluctuates slightly below and above zero, but with very low values. As noted previously, the memory of the flow is dominated by the u component, which shows a much smoother auto-correlation decay. This relatively sudden decay of the (v, w) auto-correlation can be explained by the comparative analysis of several consecutive time steps. While the individual eddies retain their identity for time sequences larger than those expected from the information in Fig. 3(c), they move continuously within the square duct, mainly from the centre towards the walls. This continuous change in the relative position of the secondary eddies within the duct quickly generates a near zero correlation, as a result of zones of positive correlation pooling with zones of negative correlation. The auto-correlation of the streamwise velocity decays in a time delay of about 30 time steps which corresponds to a downstream distance of $x/h = 2.3$, using the relation $u_b \approx 15.0u_\tau$. This result is in agreement with the values reported in Fig. 3 of Gavrilakis [4].

3. Analysis technique

The main objective of the analysis technique applied here is to substitute the conventional time mean by several ensemble averages obtained from a reasonable conditional sampling procedure in such a way that the data shows reduced variability around these conditionally averaged ensembles. The procedure that enables us to do this is fuzzy clustering. However, although FC can be performed with the raw data set, a POD analysis of the raw data was first

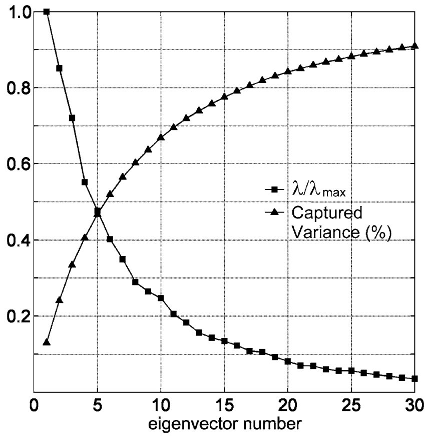


Fig. 4. POD normalised eigenvalues (—■—) and cumulative captured variance (—▲—).

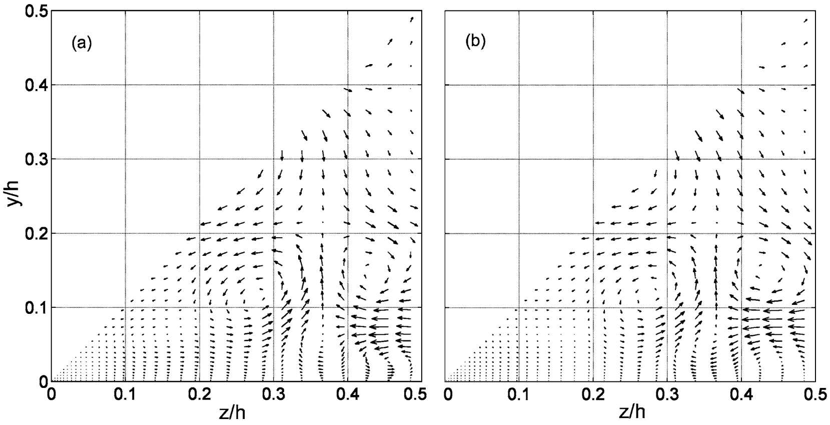


Fig. 5. (a) Instantaneous secondary flow velocity field. (b) POD projected field.

applied in order to compress the volume of data handled by the FC procedure, and to focus the FC classification on the larger scales of the secondary flow. POD was preferred to other alternative decomposition techniques, such as the wavelet transform based CVS technique [13], mainly because it is simpler. The classification operations that are performed afterwards by FC are also more straightforward.

The cross-stream velocity field within the octant consists of two velocity components sampled at 561 points. Therefore each time step can be thought of as a single point in a 1122-dimensional space. POD showed that the first 30 eigenvectors captured more than 90% of the total variance of the set, with the next eigenvector accounting for less than 5% of the total variance. These results are summarised in Fig. 4, which plots both the cumulative captured variance and the normalised eigenvalues for the first 30 eigenvectors. Gavrilakis [18], working with a reduced region near the wall from a DNS at a Reynolds number similar to the value used here, needed 17 modes to retain 90% of the turbulent kinetic energy. However, Reichert et al. [8] used 100 modes to retain 95% of the variance or turbulent kinetic energy of the DNS data computed at twice the Reynolds number used here.

Subsequently, each instantaneous velocity field is represented by its projection onto these eigenvectors, the data set being effectively reduced from its original 1122-dimensionality to a 30-dimensional space. This involves a reduction of more than one order of magnitude in the size of the data set, making it far more manageable and allowing the use of analysis techniques that still consider the data set as a whole. If the data set were not to be reduced, some operations on the whole set might be inefficient. This problem is greater if numerical simulations are performed at larger Reynolds numbers and/or if direct numerical simulations (DNS) are considered instead of LES, because in both cases the data sets get larger and larger. Still, the nature of POD ensures not only that the loss of information is small,

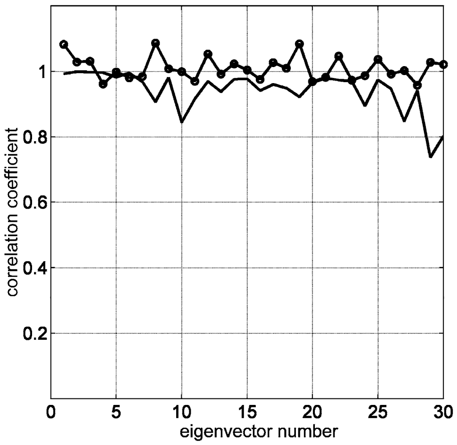


Fig. 6. Correlation between POD performed on independent halves of the data set. Corresponding eigenvector correlation (—-) and ratio between eigenvalues (—○—).

but also that it is constrained to the finer scales of the flow. Fig. 5(a) shows an instantaneous cross-stream velocity flow field, while Fig. 5(b) presents the reconstruction of this field using the first 30 eigenvectors. Note that there are only minor differences between Figs. 5(a) and 5(b): for example, the smearing out of a small vortex near the wall at $z/h = 0.45$ and $y/h = 0.05$.

While analysis of the POD eigenvectors can yield some insight into the flow structures, POD by itself cannot provide any information about the time history of the flow, since the cross-correlation matrix keeps no information on the time ordering of the velocity frames. However, this information can be accessed by analysing the time history of the projection components of the velocity frames onto the POD eigenvectors, as will be explained in Section 5. It should be pointed out that while the POD eigenvectors constitute an optimal, orthogonal base for decomposing each velocity frame into a linear combination of eigenvectors, they are not the best templates to be used to classify instantaneous velocity frames. This is due to their hierarchical and spectral nature. In other words, instantaneous velocity frames, or their given ensemble averages, should not be expected, in general, to reassemble any single POD eigenvector, but a suitable linear combination of eigenvectors. This is especially true for eigenvectors associated to low eigenvalues, because they tend to contribute to the finer scales.

To corroborate that the data set is large enough for our analysis, we performed an auxiliary POD analysis on two independent halves of the data set. The resulting sets of eigenvectors and eigenvalues were correlated to obtain the results presented in Fig. 6, which shows that both sets are nearly completely correlated, within 10% for most modes, with slight deterioration towards the higher modes associated with the finer scales.

Fig. 7 shows the first five eigenvectors plus the twentieth eigenvector to illustrate this. As can be observed in Fig. 4, the first five eigenvectors capture nearly 50% of the variance of the data while their individual normalised values are still higher than 0.5. The twentieth eigenvector was selected since it is the first one that displays a normalised value lower than 0.1 (that is to say, one order of magnitude weaker). While the first few eigenvectors might resemble some of the ensemble averages that will be presented later, it is clear that, as the eigenvector sequence advances, they increasingly reproduce the finer structures, as expected from the hierarchical and orthogonal nature of the POD decomposition. While these properties of POD are desirable in a decomposition method, they are neither necessary nor desirable in a classification method. Something else that has to be taken into account in the analysis of Fig. 7 is that these velocity fields can be fully sign-reversed without losing their condition of being the true eigenvectors of this POD analysis. Therefore, POD cannot distinguish between secondary flow motions with positive (into the plane) or negative (out of the plane) vorticity.

As stated above, the objective is to obtain a set of ensemble averages that substitute the conventional time mean, so that the data exhibit reduced variability, or fluctuating kinetic energy, around this set. Having been projected onto the first 30 eigenvectors, each instantaneous velocity field of the time series analysed (indexed as $j = 1, 2, \dots, n$, with $n = 11\,200$) is just a vector X_j in a 30-D space, with an unconditional mean $\overline{X}$. If the data is classified in $i = 1, 2, \dots, c$

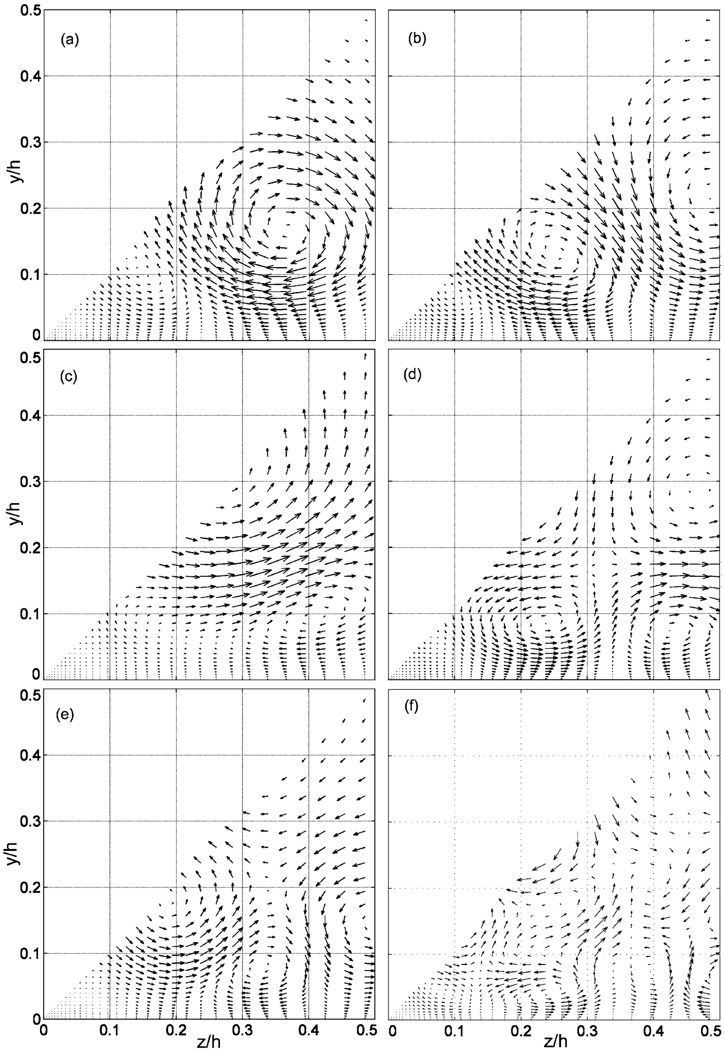


Fig. 7. Selected POD eigenvectors. (a)–(e) Eigenvectors #1–5. (f) Eigenvector #20.

different families (or clusters) each of them with their own mean V_i, then the whole variability of the data (the term on the left-hand side of Eq. (1)) is distributed among the new c clusters according to

$$\sum_{j=1}^{n} d^2(X_j, \overline{X}) = \sum_{i=1}^{c}\left(\sum_{k=1}^{n_i} d^2(X_{ki}, V_i)\right) + \sum_{i=1}^{c} n_i \cdot d^2(V_i, \overline{X}) \tag{1}$$

where the initial X_j $(j = 1, 2 \ldots, n)$ vectors on the left-hand side term have been sorted into $i = 1, 2 \ldots, c$ clusters in the terms on the right-hand side, and are now denoted as X_{ki} $(k = 1, 2 \ldots, n_i$ vectors in cluster $i)$. Also $d^2(\cdot,\cdot)$ is used here to note the Euclidean distance between two points

$$d^2(X_i, X_j) = \|X_i - X_j\|^2. \tag{2}$$

The first term on the right-hand side of Eq. (1) is the sum over all the clusters ($\sum$ for $i = 1, 2, \ldots, c$) of the variability within each cluster ($\sum$ for $k = 1, 2, \ldots, n_i$), while the second term is the contribution from the c cluster means. A classification that would suit the purpose of our analysis would be one that maximizes this second term on the right-hand side of Eq. (1), but minimises the first one, that is, the unexplained variability. Note that what we have been generically calling “variability” is, in fact, the kinetic energy of the fluctuating (v', w') velocity field since

$$d^2(X_j, \overline{X}) = \sum_{\text{whole field}} v'^2 + \sum_{\text{whole field}} w'^2. \tag{3}$$

Therefore, we can now say that our initial purpose of “reducing the variability” exhibited by the data around the conventional time-mean effectively means “reducing the kinetic energy” of the fluctuating velocity field not explained by the c cluster means.

The aim of some algorithms used for fuzzy clustering, and in particular Bezdek’s c-Means Clustering algorithm [20], is closely related to this purpose, since it tries to minimize an objective function that is very similar to the first term on the right-hand side of Eq. (1). Strictly, for a prescribed number of clusters c, and fuzziness index m $(m > 1)$, the c-Means Clustering algorithm tries to minimize an objective function J_m, defined as:

$$J_m = \sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^m d^2(X_j, V_i) \tag{4}$$

where X_j $(j = 1, 2, \ldots, n)$ is the data set to be clustered and $d^2(X_{j,} V_i)$ is a measure of the distance between the vectors X_j and the cluster centre V_i, taken here as the Euclidean distance, and μ_{ij} is the fuzzy membership function that verifies the condition:

$$\sum_{i=1}^{c}\mu_{ij} = 1. \tag{5}$$

For any partition, the cluster centroids are defined from:

$$V_i = \sum_{j=1}^{n}(\mu_{ij})^m X_j \Big/ \sum_{j=1}^{n}(\mu_{ij})^m. \tag{6}$$

Also, the fuzzy membership function can be computed from a specified set of centroids V_i from

$$\mu_{ij} = \left(\frac{1}{d^2(X_j, V_i)}\right)^{1/(m-1)} \Big/ \sum_{i=1}^{c}\left(\frac{1}{d^2(X_j, V_i)}\right)^{1/(m-1)} \tag{7}$$

J_m is minimized by iterating Eqs. (6) and (7) from a starting guess of either μ_{ij} or V_i until no further improvement in J_m (Eq. (4)) is observed.

The sum of squared distances in Eq. (4) will be equivalent to the first term on the right-hand side of Eq. (1) only if $\mu_{ij} = 1$ for the n_i vectors belonging to cluster i, and zero otherwise. This condition is verified approximately since, due to the quotient expression in Eq. (7), the membership function μ_{ij} of a vector X_j is near zero for all the clusters except for the one that is closest to this vector. Thus, during the iteration of Eqs. (6) and (7), the membership function can take any value in the range $[0, 1]$, although these values are usually closer to either 0 or 1 after the iteration converges. Once the procedure has finished, the membership function is binarised (for any data point X_j $\mu_{ij} = 1$ only for one cluster V_i and zero otherwise), so that each vector X_j is assigned to the cluster closest to it, which corresponds to the maximum value of $\mu_{ij} = 1$ for that particular X_j. Accordingly, the last clusters obtained are just conventional ensemble averages of the instantaneous (v, w) velocity fields that have been classified into the same family or cluster.

Still, the c-Means Clustering algorithm by itself gives no information about the number of clusters to be used in partitioning the set. In order to determine the optimum number of clusters to be used for each data set, as well as the value of the fuzziness index m, we applied the validity criterion for fuzzy clustering, proposed by Xie and Beni [21]. For any given partitioning, regardless of the algorithm used to determine it, a compactness and separation validity function S, can be defined as:

$$S = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^2 d^2(X_i, \overline{X}_j)}{n \cdot \min_{ij} d^2(X_i, \overline{X}_j)}. \tag{8}$$

The validity function S can be regarded as the ratio of the measure of the compactness of the clusters over the minimum separation among clusters. Optimum values of c and m are those for which S reaches an absolute minimum value.

The last step in the description of the fuzzy clustering algorithm used to analyse the data is to briefly discuss the initialisation of the iterative solution of Eqs. (6) and (7). Some authors simply recommend a random initialisation of μ_{ij}. To check the stability of the minimum of J_m (Eq. (4)) obtained in this way, we compared several hundreds of different random initialisations of μ_{ij}. The results obtained confirm that a random initialisation of μ_{ij} always leads to the same minimum value of J_m. However, we preferred an approach that took into account the fact that the raw data had previously been pre-processed with POD. Either the strongest eigenvectors or the ensemble averages of the vectors contained in the first few quadrants (n vectors define 2^n quadrants) were used as the initial cluster centroids, That is, we used either "pure" eigenvectors or their linear compounds. In both cases the results obtained from the iterative solution of Eqs. (6) and (7) were indistinguishable, and we finally decided to use the linear compounds as our standard starting strategy.

4. Secondary flow field analysis

The application of the fuzzy clustering validity criterion (Eq. (8)) yielded optimum values of $m = 1.12$ and $c = 17$ for the classification of the cross stream velocity field. As noted above, this criterion for optimal partitioning means that there is no need to arbitrarily specify the number of cluster ensembles to be constructed. Instead, the value of this parameter results from the intrinsic characteristics of the data set and the application of the validity criterion.

When the 11200 time steps of the octant data set were clustered with these parameters, about the same number of instantaneous frames were classified into each of the 17 clusters, giving a mean number of 660 frames in each of them (or about 5.9%, see Fig. 8). Figs. 9(b), 9(c) and 9(d) present only three cluster mean (v, w) velocity fields out of the seventeen obtained, together with the unconditional mean field (Fig. 9(a)), which has been added for purposes of comparison. All the vector fields plotted in the lower octant of the figures are given the same scale so that the relative weakness of the unconditional mean (v, w) velocity field can be assessed. For the sake of clarity, the unconditional mean field is symmetrically re-plotted on a larger scale in the upper octant of Fig. 9(a). Cluster mean velocity fields are up to one order of magnitude stronger than the local unconditional time mean. The gray scale background in Fig. 9 represents the ratio between the local energy of the fluctuating (v', w') field (Ec. 3) and the respective local mean field ($\overline{X}$ in Ec. 1). Thus, this ratio is the local turbulence intensity of the cluster ensemble. In the darker areas the high relative intensity of fluctuations reaches ratios that are above 20, while in the brighter areas this ratio can be below 1. The mean ratio, averaged over the octant section, for the unconditional time-mean (Fig. 9(a)) is about 40, while for any of the three clusters presented in Figs. 9(b), 9(c) and 9(d) it is below 2. The darkest regions in Fig. 9(a) are the result of the local unconditional time mean approaching zero faster than the fluctuating (v', w') field.

Several of the features in Fig. 9 merit some comment. First, the unconditional mean (Fig. 9(a)) is relatively weaker than any of the conditionally sampled or cluster mean averages (Figs. 9(b), 9(c), 9(d)). Second, the increase in the signal-to-noise ratio displayed by the cluster means relative to the unconditional time mean. But most strikingly, the

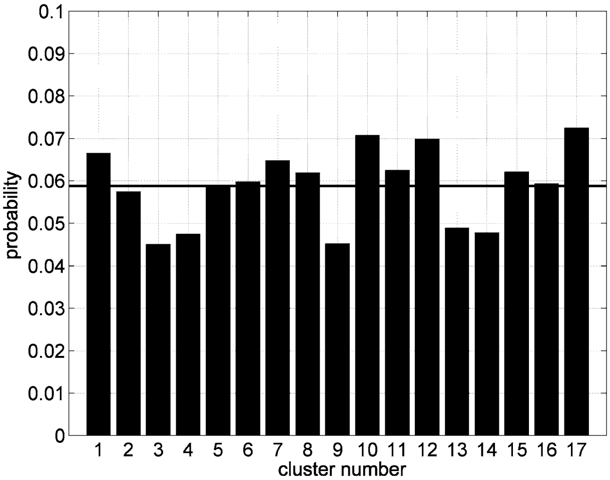


Fig. 8. Probability of occurrence of each cluster. Thick line indicates the mean probability (1/17) of equally probable clusters.

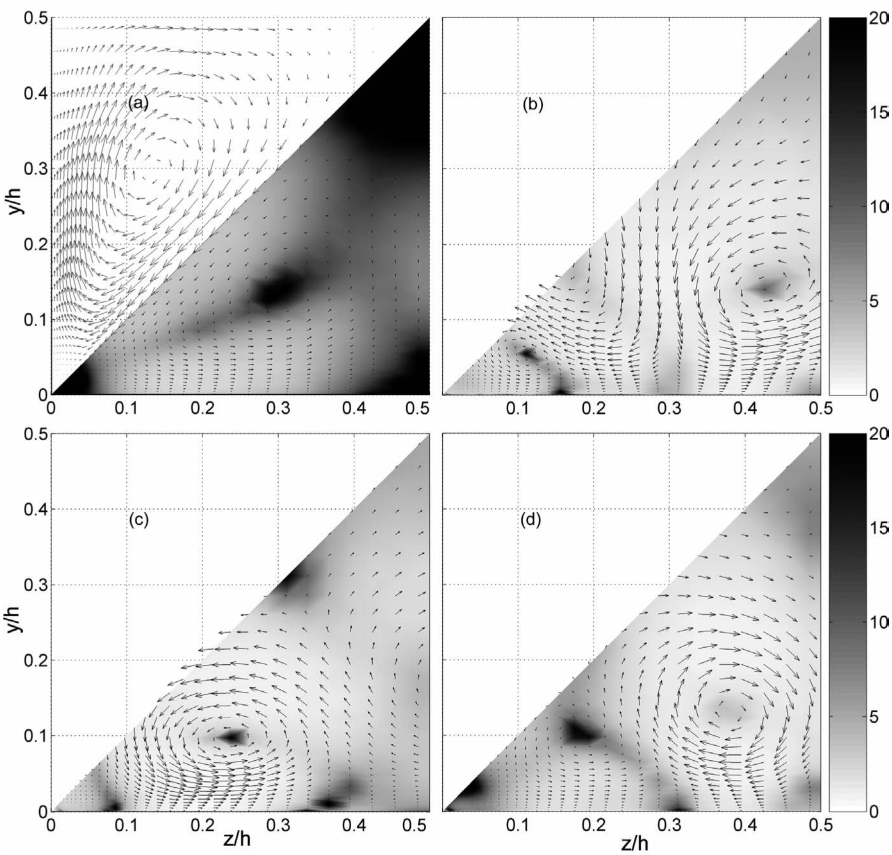


Fig. 9. Secondary flow field with local fluctuations intensity overlay. (a) Unconditional time mean. (b) (c) and (d) Selected clusters. Velocity vectors in the lower octants are plotted to the same scale in all frames. For purposes of comparison, in the upper octant of frame (a) the mean velocity field has been (symmetrically) replotted on a larger scale.

velocity fields of these true ensemble averages are very different from the unconditional mean which, in comparison, becomes meaningless. In fact the clusters in Figs. 9(b), 9(c) and 9(d) have been selected because they exhibit large-scale circulation with the same sign as the mean velocity field (Fig. 9(c)), with the opposite sign (Fig. 9(d)), or simultaneous zones with both signs (Fig. 9(b)).

Fig. 10 shows some instantaneous velocity fields selected from the clusters in Figs. 9(b), 9(c) and 9(d). While other instantaneous fields within each class show varying levels of similarity to the corresponding cluster mean, Fig. 10 emphasises the fact that, despite being ensemble averages, the cluster means do represent physically existing flow realisations.

The distribution of the kinetic energy among clusters, announced in Eq. (1), is presented in Fig. 11, where we have plotted the seventeen terms on the right-hand side of the equation divided by the left-hand side term: that is,

$$1 = \sum_{i=1}^{c}\left(\frac{\sum_{k=1}^{n_i} d^2(X_{ki}, V_i)}{\sum_{j=1}^{n} d^2(X_j, \overline{X})} + \frac{n_i \cdot d^2(V_i, \overline{X})}{\sum_{j=1}^{n} d^2(X_j, \overline{X})}\right). \tag{9}$$

For the sake of comparison, a horizontal line is plotted at $1/17$ (i.e. 5.9%), the expected mean contribution of each cluster to the total sum of squares. Another line is also plotted at about 2.1%, which corresponds to the mean explained variability of the data; that is, the second term on the right-hand side of Eq. (9). The difference between these two values, about 3.8%, is the mean contribution from the still unexplained variability of the data; that is, the first term on the right-hand side of Eq. (9). The relative weight of the explained kinetic energy to the total kinetic energy is only moderate, about 36% ($1.8/5.9 \approx 0.36$) if it is computed as the ratio of mean squared velocities, or 59% ($0.36/0.5 \approx 0.59$) computed as the ratio of the square root of mean squared velocities. However, it is still much higher

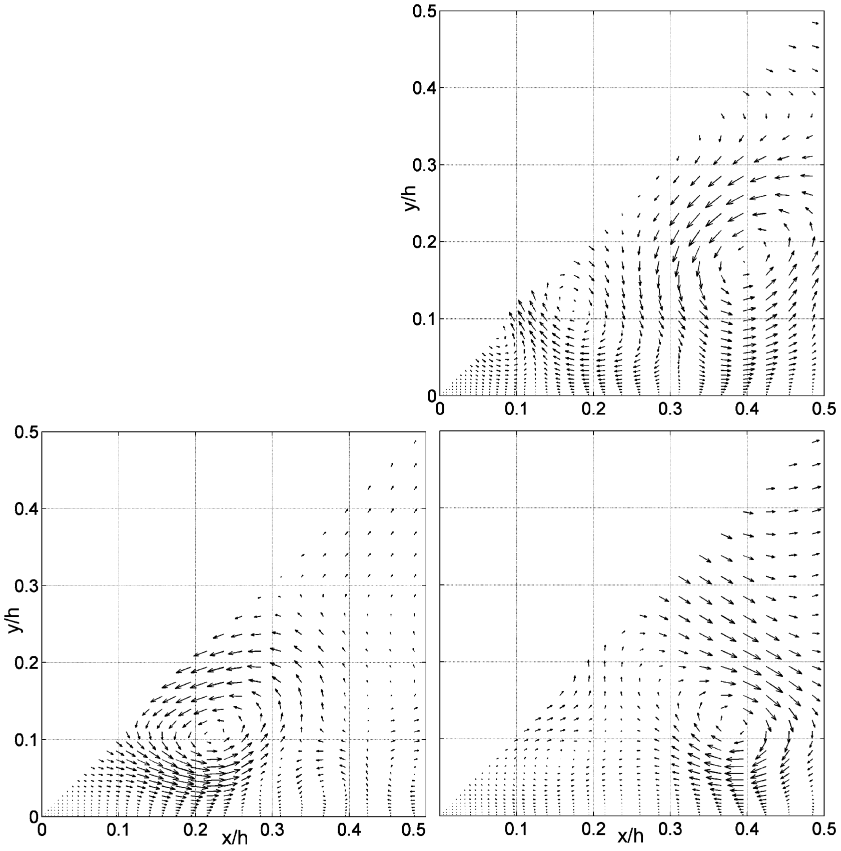


Fig. 10. Selected instantaneous velocity frames from those contributing to the cluster averages depicted in Figs. 9(b), 9(c) and 9(d).

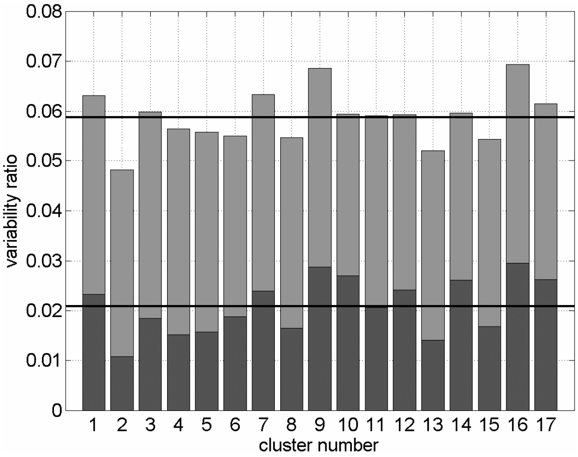


Fig. 11. Explained (dark gray, lower bars) and unexplained (light gray, upper bars) kinetic energy distribution among clusters.

than it was for the unconditional mean, where the kinetic energy of the fluctuating field was an order of magnitude above the kinetic energy of the mean field (for example, see Fig. 9).

In order to explore this subject further, we analysed how the kinetic energy content of the cluster mean velocity fields is distributed along the eigenvector spectrum. The results are presented in Fig. 12, where we compare the r.m.s. of each POD component of the whole data set and the r.m.s. of the POD components of the seventeen clusters. While

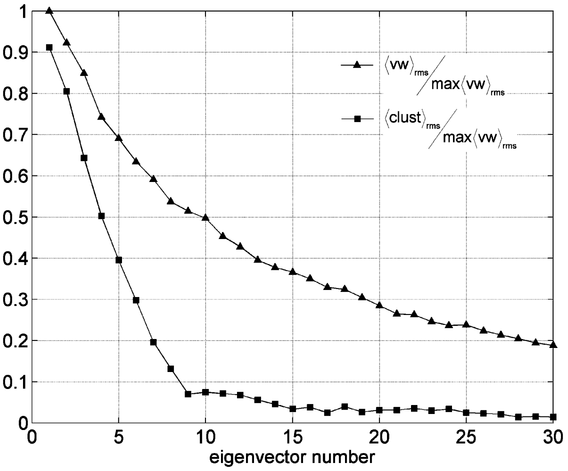


Fig. 12. Rapidly decaying energy content of cluster means along the eigenvector spectrum.

the energy content of the raw data slowly decays along the POD spectrum, the energy content of the cluster means is almost constrained to the first 10 eigenvectors, as can be expected for any ensemble average, where finer scales have been smeared out by the averaging procedure. Thus, the energy content associated to the remaining 20 eigenvectors corresponds to finer scale details and is out of the reach of the computed cluster means. This energy content can be as much as 31% of the total fluctuating energy in the cross-stream flow (see Fig. 4). Thus, from the remaining 69% of the energy associated to large scale motions, 36% has been captured by the computed cluster means or ensemble averages, while 33% is still attributed to the random character of the turbulent secondary motions.

Incidentally, Fig. 12 also justifies the choice, made in Section 3, of keeping just the first 30 eigen modes of the POD. Since the obtained cluster means only have significant projections onto the first 10 modes, the results of the present analysis would not be affected by the number of modes retained, as long as this number is comfortably larger than 10. The discarded modes would have no impact on the obtained cluster ensemble averages.

Below the fuzzy clustering (FC) procedure is compared with the pattern recognition (PR) and proper orthogonal (POD) techniques. For this purpose, PR was applied to the data set using the first few POD eigenvectors as initial templates for the PR analysis, as proposed in Kopp et al. [9]. The cross correlation threshold level for accepting individual elements was tuned to obtain ensemble averages consisting of a number of elements similar to those obtained by the FC technique. The resulting threshold level was 0.76. Fig. 13 displays the four ensemble averages obtained using the first four POD eigenvectors already presented in Fig. 7 as initial templates.

The ensemble averages in Fig. 13(a) and (b) are similar to their initial POD templates, while in cases (c) and (d) they are considerably different. As stated above, this behaviour is a result of the spectral nature of the POD decomposition. As we proceed higher into the POD modes, the POD eigenvectors should not be expected to resemble true ensemble averages of the flow. In other words, POD decomposes the flow into a spectral domain of orthogonal modes, while FC and PR decompose the flow into distinct ensemble averages within the spatial domain.

The circulation patterns in Fig. 13, obtained by PR, are similar to those of Fig. 9(b), (c) and (d), which were obtained by FC. Thus it seems that both PR and FC detect compatible sets of the ensemble averages which characterize the flow. However, PR requires starting templates to be provided while FC does not. FC also provides the means for obtaining a full set of disjoint ensemble averages that satisfy a given criterion of compactness and separation. Because PR determines the ensemble averages independently it is not possible, in principle, to enforce such conditions of compactness and separation. Notice that in case (a) and case (c) of Fig. 13 the ensemble averages are quite similar, even though the starting templates were quite different. Furthermore, the PR ensembles are not required to be disjoint and might share some instantaneous elements, and all the instantaneous velocity fields are not required to participate in the classification process. These properties of the fuzzy clustering process also suit the purpose of the section below, which analyses the temporal organisation of the data, since FC is able to classify all the instantaneous velocity fields in one family or another.

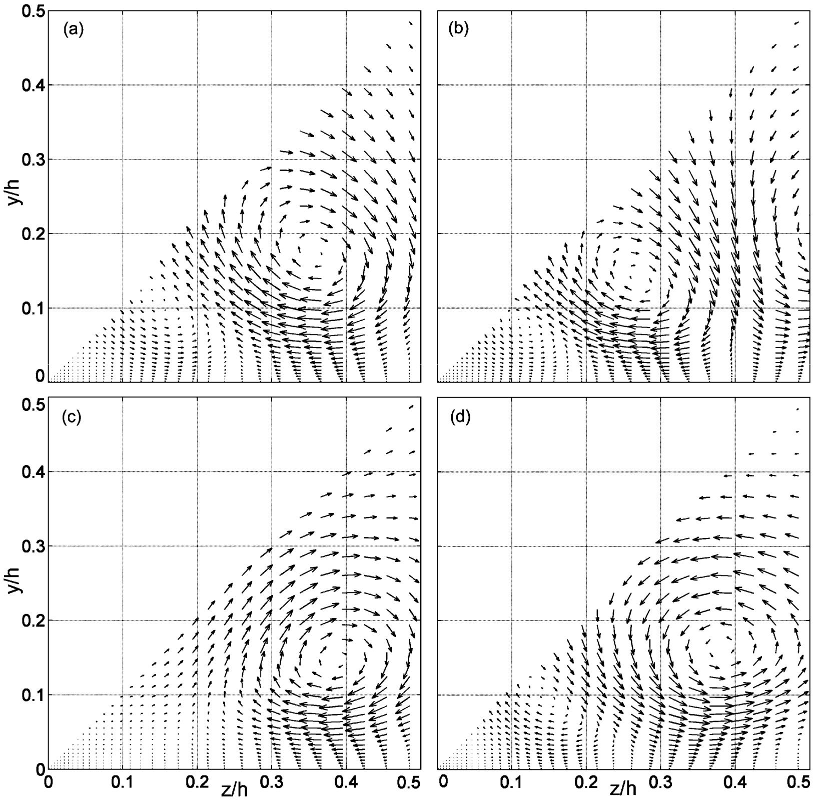


Fig. 13. Flow patterns obtained from pattern recognition using the first four POD eigenvectors from Fig. 7 as starting templates.

5. Time ordering of the clusters

All the results presented in the previous section are independent of the actual order of the time steps contained in the time series that has been analysed. That is, if they were to be in a random order, the clusters in Fig. 9, or the kinetic energy distribution analysed in Fig. 11, would be the same. Incidentally, we should also point out that the POD analysis that was reported in Figs. 4–7 is also insensitive to any random permutation in the time steps of our time series data, since the spatial correlation matrix on which POD is constructed is independent of the time sequence of the data.

To extract the data's temporal information, we will start by analysing how the instantaneous (v,w) data fields are classified into clusters. Suppose that each one of the 17 clusters obtained are labelled with a capital letter $(A,B,\dots,Q)$. Since all the individual velocity fields are classified as belonging to one and only one of the clusters, a capital letter can also be assigned to each time step. Therefore the time evolution of the secondary motions can be viewed as a sequence like:

$$\cdots KKKPPPPBJJJHHOFFEECCCI\cdots$$

where each letter represents the label of one time step.

At the beginning of this series it can be seen that for three successive time steps the instantaneous velocity fields are classified into cluster K, then into cluster P for four time steps, and so on. Successive time steps belonging to the same cluster is known as a "tie". Some ties are very short, like the B that appears between the P's and the J's, but others are 2, 3, 4 or even more time steps long. Fig. 14 shows the tie length distribution. The mean tie length is 2.6. The tie length distribution is consistent with the fast decay of the time auto-correlation of the (v,w) velocity field already presented in Fig. 3(c), which decays to almost zero values for time step displacements above 5.

However, more interesting than the length of the ties themselves are the "jumps" from one tie to another: that is to say, from being classified in one cluster at one time step they are then classified in another cluster at the following

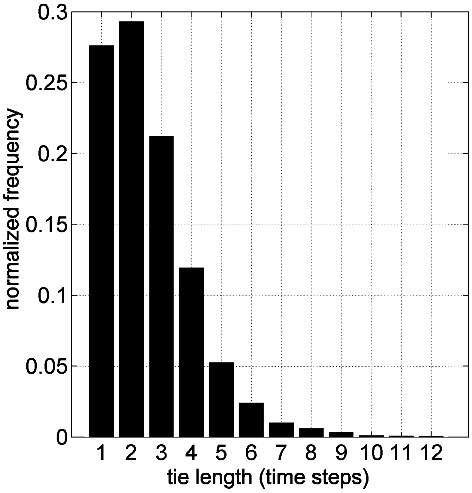


Fig. 14. Histogram of tie lengths.

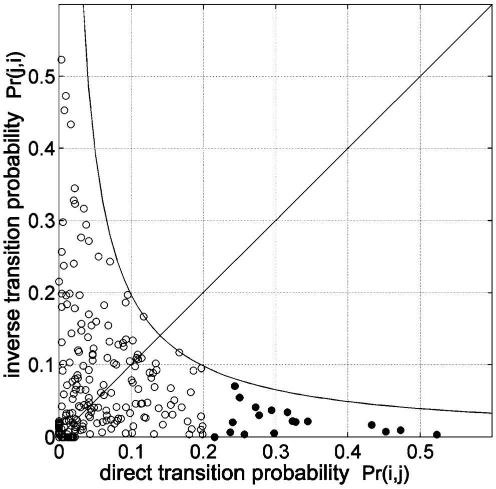


Fig. 15. Direct and inverse transition probabilities.

time step. The first jump that we observe in the short series given above as an example, is K to P, which is followed by a P-to-B jump, and so on. The last jump observed here is C to I. There are $17^2 - 17 = 272$ possible jumps, from one cluster to any other filling a matrix $Pr(i, j)$ with the probability of the (v, w) velocity field being classified into cluster V_i at time step t and into cluster V_j at time step $t + 1$. If the jumps between clusters occur randomly–that is, they are all equally probable–then all the terms in $Pr(i, j)$ should randomly fluctuate around the same value $(1/(17 - 1) \approx 6.25\%)$ and, in particular $Pr(K, P) \approx Pr(P, K)$, which is the probability of observing a K-to-P jump as above, should be of the same order of magnitude as the probability of observing a P-to-K jump.

The probability of each transition occurring was actually computed directly by counting the number of times that each transition occurred within the time series analysed. Instead of a uniform distribution of probabilities it was found that there are only a few jumps with a high probability of occurrence. Moreover, when a given jump occurs frequently, say from cluster K to cluster P, then the inverse jump, from P to K, tends to occur rarely. These results are displayed in Fig. 15, where the probability of one jump $(Pr(i, j))$ is plotted on the horizontal axis against the probability of the inverse jump $(Pr(j, i))$ on the vertical axis. Note that the plot in Fig. 15 is symmetric because both the points $(Pr(i, j), Pr(j, i))$ and $(Pr(j, i), Pr(i, j))$ are plotted. Therefore, the analysis of the results in Fig. 15 is limited to the points below the straight line $y = x$.

The first thing that can be observed in Fig. 15 is that the elements of $Pr(i, j)$ sometimes attain values as high as 0.5, but are often close to zero. Therefore, it seems that the jumps do not follow a uniform distribution. In fact only a few transitions have probabilities higher than 20%. These data points have been marked with filled circles on the horizontal axis. In addition, high values of $Pr(i, j)$ imply low values of $Pr(j, i)$, as supported by the fact that the data points with $Pr(i, j) > 0.2$ (filled circles) have $Pr(j, i)$ values well below 0.1. Only when $Pr(i, j)$ is small can $Pr(j, i)$ be of the same order.

The conclusion that can be drawn from the above observations is that some jumps between clusters are preferred, and also that they are oriented: that is to say, once it is known that a particular instantaneous (v, w) velocity field is classified into one cluster, say V_5, then it can be assessed with a probability of 0.52 that after some time steps it will fall into cluster V_8 (please see Fig. 16). And, if an instantaneous (v, w) velocity field is classified into cluster V_8, then it is fairly improbable that it will subsequently fall back into cluster V_5. This is the rightmost point in Fig. 15.

The above findings make it possible to order the clusters in the time direction. They are renumbered from 1 to 17 so that their natural time ordering is consistent with their increasing index. Fig. 16 presents the result of this ordering, which will be described in greater detail below, while Table 1 contains the matrix $Pr(i, j)$ with the most important probabilities of the jumps from cluster i to cluster j. Note that since the original order of the clusters obtained from FC is arbitrary, the original $Pr(i, j)$ matrix does not have the diagonal dominant structure of Table 1: instead it appears as random permutations of rows (and columns) of Table 1. This is why both Table 1 and Fig. 16 have not been presented until this point when they can both be presented in a more consistent way. In Fig. 16, the boxes contain the cluster index, and also the absolute probability of each centroid (in fact, these values have previously been plotted in Fig. 8).

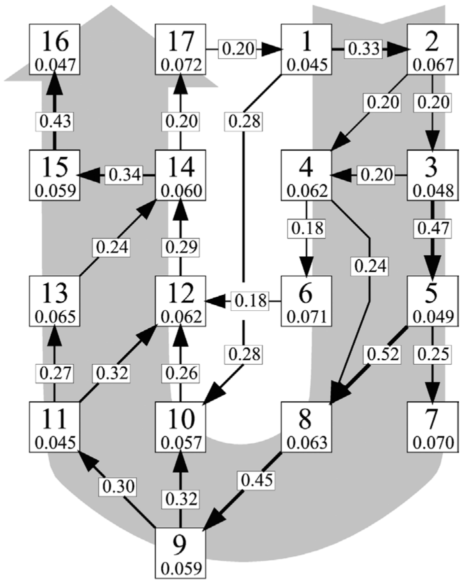


Fig. 16. Schematic representation of the most probable transitions, given arrows between pairs of clusters. The probability of these transitions is indicated in legends superimposed on the corresponding arrows. The clusters are identified with ordinal numbers (1–17) in each square box, together with the probability of each cluster (in smaller font).

Table 1
Probabilities of selected jumps. Level-1 jumps in dark cells, Level-2 jumps in grey cells

		To cluster **j**																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
From cluster **i**	1		0.33								0.28					0.01	0.11	
	2			0.20	0.20													
	3				0.20	0.47												
	4						0.18		0.24									
	5							0.25	0.52								0.05	
	6												0.18					
	7																0.02	
	8									0.46								
	9										0.32	0.30					0.01	
	10												0.26				0.01	
	11												0.32	0.27				
	12														0.29			
	13														0.24	0.34		
	14																	0.20
	15	0.10															0.43	
	16	0.11				0.12		0.24		0.18	0.13					0.02		
	17	**0.20**																

The arrows from one box to another point in the direction of the observed jumps, while the label over each arrow shows the probability of this jump. For example, we can see in Fig. 16 that the probability of randomly picking up an instantaneous (v, w) velocity field that is classified into cluster 5 is 4.9%. So, the probability that after some time steps (depending on the length of the tie) the velocity field will jump into cluster 7 is 25%, and the probability of jumping

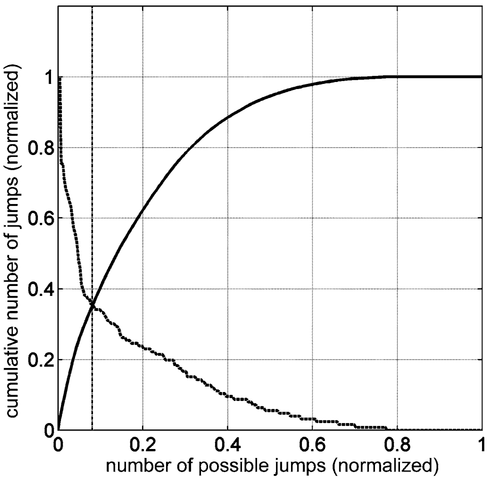


Fig. 17. Cumulative number of jumps (—-) and normalised frequency of jumps (— —), plotted in decreasing order of frequency.

into cluster 8 is 52%. In this latter case, there is a probability of 45% that the next jump observed is into cluster 9, and so on.

The general layout of Fig. 16 is rather arbitrary: we could mirror it left to right or upside down. What is less arbitrary is the information that we have included. To construct this figure we followed a simple procedure. We started by examining the matrix $Pr(i, j)$, which contains the conditional probability of each jump, and extracting the “strongest” jumps: that is to say, the $Pr(i, j)$ that were both maximum in their row and also in their column. We called this a level-1 search. Then we repeated the procedure but skipped over the values used in the previous search, in what we called a level-2 search. The lists obtained from these two searches are marked in Table 1 with dark and grey boxes, respectively, and contain the whole set of clusters, except cluster number 6. Therefore we decided to include this cluster arbitrarily in Fig. 16, and we also plotted the most probable jump to and from cluster 6 (see Table 1). We also tried to generate other plots equivalent to the one already displayed in Fig. 16, including “less strong” jumps (that is, level-3 jumps, level-4 jumps, etc). Two things happened. First, the jumps in these lists had smaller probabilities, below 0.2, and they were less oriented: that is to say, still $Pr(i, j) > Pr(j, i)$ but no longer $Pr(i, j) \gg Pr(j, i)$. Both effects can be easily observed in Fig. 15. But the most important thing was that it was impossible to keep Fig. 16 relatively simple because many cross-jumps started to appear. Still, when only level-1 and level-2 jumps were plotted, the sum of the absolute probabilities of all the jumps in Fig. 16 amounted to more than 35% of the jumps recorded in the time series. This means that if we monitored the flow for a while, one of every three jumps that we observed is in Fig. 16. Therefore, we can conclude that Fig. 16 displays the “essential dynamics” of the large-scale secondary flow motions in a square channel.

Fig. 17 plots, in decreasing order of probability, the cumulative portion of transitions as the number of possible jumps retained is increased. A vertical limit is plotted for the number of jumps retained in Fig. 16, which corresponds to 8.5% of the possible jumps and, as noted above, accounts for more than 35% of the jumps occurring in the time series. Also the decreasing normalised frequency of the jumps is plotted (please note that the fact that the three lines almost cross at the same point is a coincidence).

Fig. 17 reinforces the fact, already discussed in relation to Fig. 15, that there are a limited number of preferred or most frequent jumps, and a considerable number of jumps occur much less frequently. The most frequent jumps are the most significant for our analysis. Retaining more, but less frequent, jumps would add little significant information and increase the complexity of the analysis.

At this point the complete set of centroids obtained from the fuzzy cluster analysis of the (v, w) data can be plotted with a layout that is consistent with Fig. 16. The results can be seen in Fig. 18. In order to increase our understanding of the large-scale organisation of the flow, each centroid that was originally only the ensemble average of the data from the lower octant, has now been expanded with more points from the rest of the channel. In addition, contours of ω_x vorticity have been added to help identify the main vortices. A detailed analysis of Fig. 18 explains why the plot of the complete set of ensemble averages has been delayed until this point. Many pairs of averages linked by a

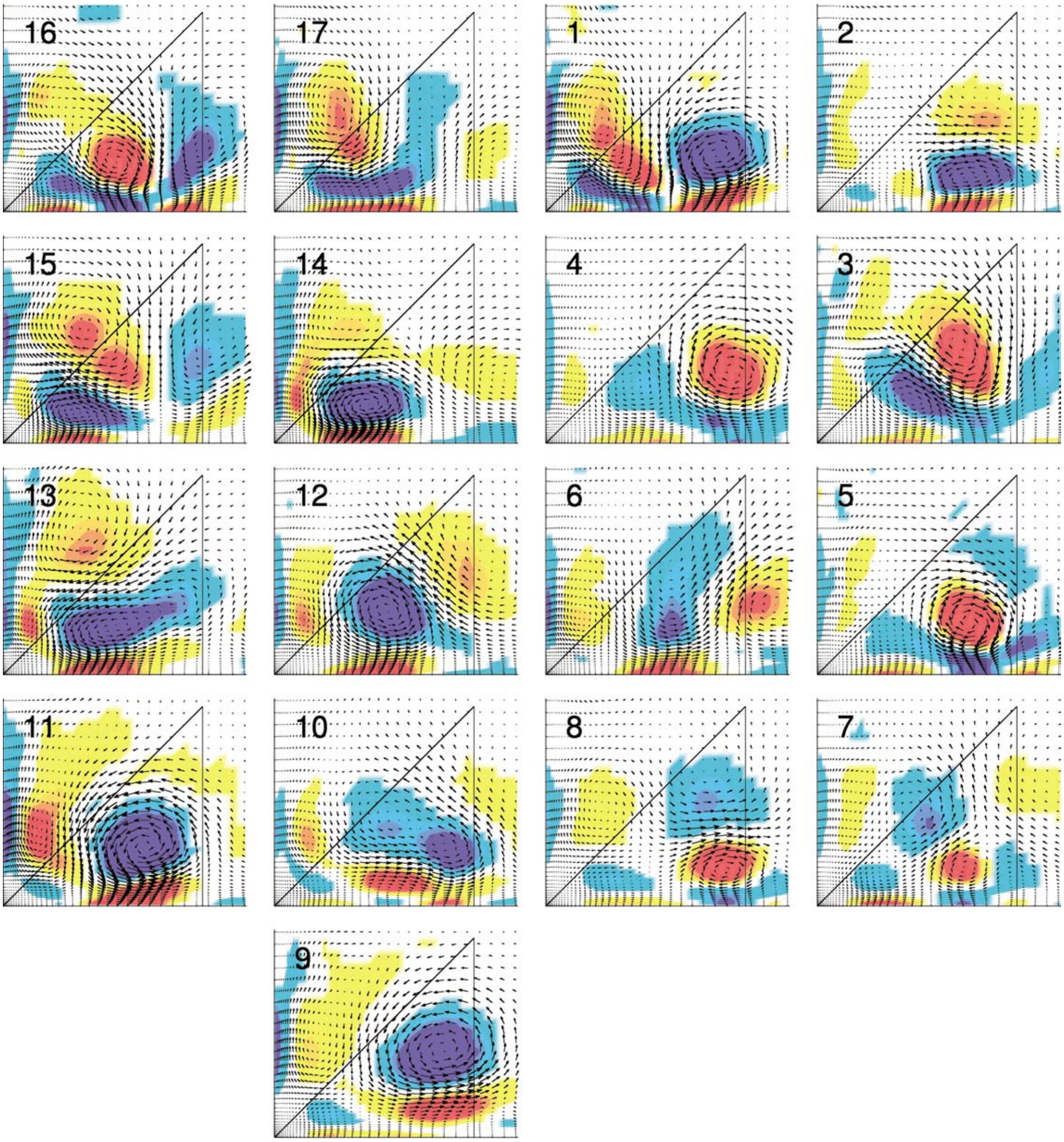


Fig. 18. Sequence of (v, w) centroids with vorticity background.

jump appear every time an eddy travels from one (z, y) position to another. This is clearly observed in the sequence 3-5-8, for example, and also in the longer sequences 9-11-12-14 and 9-11-13-14. These sequences are particularly interesting because when a large-scale vortex approaches the wall it generates a layer of sign-inverted vorticity that is no longer associated to large-scale circulation: rather it is wall-generated (boundary-layer) vorticity. Some of the conditionally sampled ensemble averages presented in Fig. 16 bear some similarity to the unconditional time mean. We can put centroids #9, #10, #11, #12, #13 and #14 in this list. Some of them, however, like centroids #4 and #5, have sign-opposed vorticity. Finally, another subset including centroids #15, #16, #17, #1 displays the traces of two counter-rotating vortices approaching the wall and, thus, lying over sign-inverted beds of vorticity. This is clearly observed in centroid #1.

Other sequences are less easy to interpret. For example, while it is clear that centroid #16 can follow #15, it is surprising that Fig. 16 shows no link between centroids #15 or #16 and centroid #1. We have to investigate the lower values of $Pr(i, j)$ to find a relationship between them. The values involving clusters #1, #15 and #16 have also

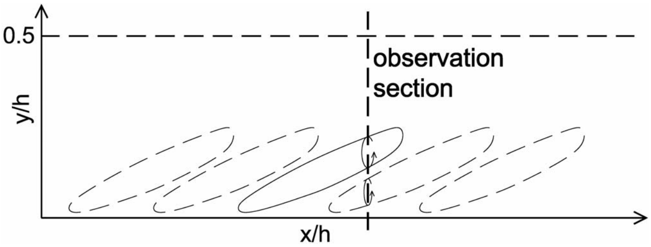


Fig. 19. Sketch of a vortical structure advected through the observation section.

been included in Table 1, which clearly shows the relationship (already displayed in Fig. 16) between #15 and #16: the probability of a 15-to-16 jump is high (43%) while the probability of a 16-to-15 jump is very low (only 1.6%). Centroid #1 has a non-symmetric relationship with #15 and #16. While 15-to-1, 16-to-1 and 1-to-16 jumps are equally probable (about 10%), the 1-to-15 jump is seldom observed (0.6%). Fig. 16 provides the explanation for these values. Centroid #1 always comes after centroids #15 and #16, and the joint probability of observing any of these jumps is 21% $(0.101 + 0.109 = 0.210)$. However, centroids #1 and #16 are too similar, and sometimes the clustering procedure misclassify an instantaneous (v, w) field and jumps back from 1-to-16.

Going further into Fig. 16's ability to represent the essential dynamics of the flow, we should point out that centroid #16 seems to be a dead end. However, a closer look at the values of $Pr(i, j)$ reveal that this is not true. Table 1 also shows the $Pr(16, j) > 0.1$ and also the corresponding $Pr(i, 16)$ values. Except for the $Pr(16, 1) \approx Pr(1, 16)$ case, discussed in the paragraph above, in all the other cases $Pr(16, j) \gg Pr(i, 16)$: that is to say, there continues to be a time directionality in the jumps. The reason why centroid #16 seems to be a dead end is clear. When the flow is classified in cluster #16 there are several possible jumps that follow this state, the most important of which is a 16-to-7 jump. This has not been included in Fig. 16 because the most important jump to cluster #7 is still from cluster #5, with probability of 25%, slightly over the 24% probability of the 16-to-7 jump.

The flow dynamics observed in Fig. 18 can be interpreted by considering the results obtained by Jeong et al. [22] in a turbulent channel, which revealed the existence of streamwise elongated vortices that originate at the wall and extend downstream and away from the wall.

Fig. 19 shows this structure in several downstream positions, as it is convected through the observation section where the data analysed here has been sampled. The intersection between the moving structure and the fixed observation section results in a vortex that approaches the wall as the structure is convected downstream, as can be seen in Fig. 18, in sequences 3-4-5-7-8 and 9-11-12-13-14-15.

This interpretation also gives further insight into the rapid loss of correlation undergone by the cross stream velocity field, as opposed to the streamwise velocity (as is shown in 3(c)). The displacement of the vortical structure produces a loss of correlation when evaluated at a given fixed cross section, since the intersection between the structure and the observation section moves and evolves. This issue is further illustrated by Fig. 20, which reproduces two instantaneous velocity frames, which follow one another within 3 time steps. They have been classified into clusters 9 and 11. While they can be interpreted as depicting the evolution of a single vortical structure like the one in Fig. 19, they have a relatively poor correlation of only 0.24.

We tried to verify this interpretation of the flow dynamics independently, without using POD and fuzzy clustering, or the λ_2 criterion used by Jeong et al. [22], and to make a quantitative estimation of the inclination angle of the structures. To this end, we computed unconditional, point to point correlation coefficients between selected locations at the wall and inner channel points. Seven points were selected along the wall at 0.05 $z/2h$ intervals. At the wall locations the streamwise component of the wall shear stress was measured, while at the inner points the selected magnitude was the streamwise vorticity, computed from the cross stream velocity components (v, w). The heights of the correlation peaks and the time delay were recorded. Finally, for each wall location, and at every wall distance, the maximum correlation peak was found. The time displacement for each of these maximum peaks was transformed into a distance using the mean convection velocity. The resulting correlation traces are plotted in Fig. 21, in normalised wall units:

$$y+ = \frac{y}{2h} \cdot Re_\tau, \qquad x+ = \frac{x}{2h} \cdot Re_\tau. \tag{10}$$

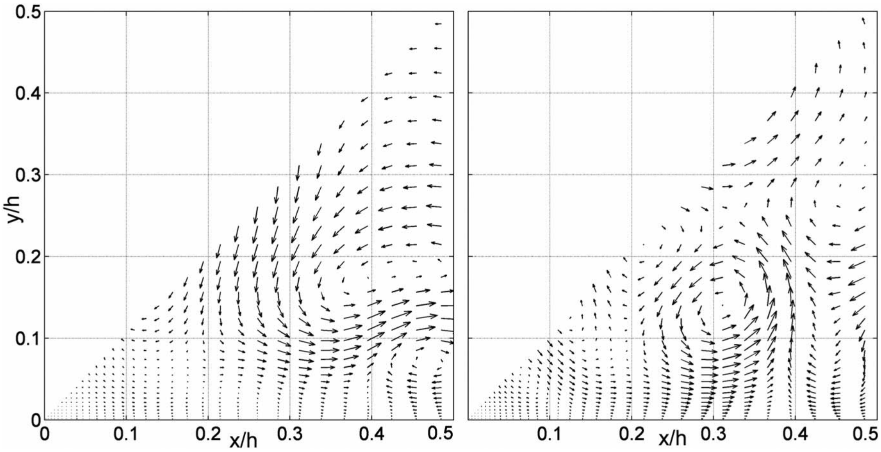


Fig. 20. Two instantaneous velocity frames, corresponding to cluster means #9 and #11, respectively. The time elapsed between both frames is 3 time steps.

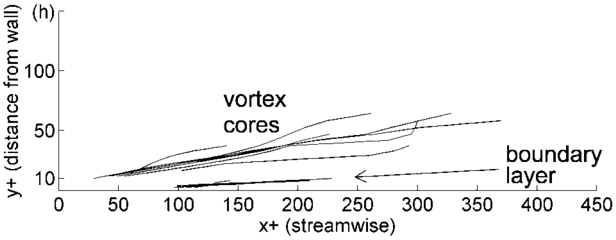


Fig. 21. Traces of the correlation peaks between streamwise vorticity and wall shear stress in the (x, y) plane.

Several aspects of this plot are worth mentioning. Firstly, two regions of collapsing traces can be seen. The main region is outside the viscous sub-layer, extending vertically from $y+ = 10$ to $y+ = 60$, and horizontally from $x+ = 50$ to $x+ = 350$. A smaller region is fully inside the viscous sub-layer, below $y+ = 10$ and from $x+ = 100$ to $x+ = 225$.

The main region is associated with the displacement of the vortical structures originated from the wall event, fixed at $x+ = 0$ in the plot. The mean inclination of these traces, in this region, is approximately 9°, in agreement with the observations of Jeong et al. [22]. The smaller region is a vorticity layer, of the opposite sign, that is forced by the structure onto the boundary layer as it moves away from the wall. These layers can be clearly seen in Fig. 18, as for example in clusters 5 and 11 which contain counter rotating vortices.

Thus, the existence of certain flow structures and their evolution over time was educed from the clustering analysis and then corroborated by an independent method. This seems a reasonable way to proceed since it exploits the unique advantages of clustering analysis, which reveals the time history of a non-periodic evolving flow but prevents any misleading conclusions that might be drawn from the probabilistic argument involved in interpreting the time relations between clusters.

6. Conclusions

A conditional sampling technique was developed, which used POD and fuzzy clustering to identify coherent structures within the flow and analyse their time history. This last property is an improvement on other techniques that are insensitive to the time ordering of the instantaneous fields. This conditional sampling technique involves all of the instantaneous fields in the time series, unlike pattern recognition or other conditional sampling schemes, for example, which typically use only a small percentage of the time series. Thus this technique is suitable for studying the large-scale flow dynamics of turbulent flows.

The turbulent flow in a square channel was analysed, with particular focus on the cross-stream secondary flow motions. The (v, w) cluster means, or ensemble averages, are much stronger than the single unconditional mean, with a much improved signal to noise ratio. This last feature makes it possible to interpret these ensemble averages directly as true representatives of the instantaneous turbulent large-scale secondary motions in this channel flow. While some of the ensemble averages obtained bear some similarity to the unconditional time mean, others clearly display sign-opposed vorticity or even traces of two counter-rotating vortices that approach the wall lying over sign-inverted beds of vorticity. The main vortical structures in the secondary flow were shown to originate at the wall, and to extend downstream and towards the centre of the channel. Vorticity layers were also observed within the viscous sub layer, forced by these vortical structures as they drift away from the wall.

When the time series formed from recording the jumps between clusters is analysed, the essential dynamics of the flow can be summarised in a relatively simple plot. This plot shows the probability of each cluster or "state" of the flow, as well as the probability of a transition to a new cluster. This makes it possible to further explore the contribution of each of the large-scale motions to the turbulent transport of momentum or heat transfer.

Acknowledgements

This work was financially supported by grants 2002 SGR00189 from AGAUR Catalan government, DPI2003-06725-C02-01 from DGI, Ministerio de Ciencia y Tecnología y Fondos FEDER, and a pre doctoral grant from the URV that was assigned to Gabriel Usera.

References

[1] A. Huser, S. Biringen, Direct numerical simulation of turbulent flow in a square duct, J. Fluid Mech. 257 (1993) 65–95.
[2] M. Piller, E. Nobile, Direct Numerical simulation of turbulent heat transfer in a square duct, Internat. J. Numer. Methods Heat Fluid Flow 12 (6) (2002) 658–686.
[3] M. Salinas Vázquez, O. Metais, Large-eddy simulation of the turbulent flow through a heated square duct, J. Fluid Mech. 453 (2002) 201–238.
[4] S. Gavrilakis, Numerical simulation of low-Reynolds-number turbulent flow through a straight square duct, J. Fluid Mech. 244 (1992) 101–129.
[5] B.K. Madabhushi, S.P. Vanka, Large eddy simulation of turbulence-driven secondary flow in a square duct, Phys. Fluids A 3 (11) (1991) 2734–2745.
[6] J. Pallares, L. Davidson, Large-eddy simulation of stationary and rotating channel and duct flows, in: Proceedings Svenska Mekanikdagar, Sweden, June 7–9, 1999, p. 4.3.
[7] H. Xu, A. Pollard, Large eddy simulation of turbulent flow in a square annular duct, Phys. Fluids 13 (11) (2001) 3321–3337.
[8] R.S. Reichert, F.F. Hatay, S. Biringen, A. Huser, Proper orthogonal decomposition applied to turbulent flow in a square duct, Phys. Fluids 6 (9) (1994) 3086–3092.
[9] G.A. Kopp, J.A. Ferré, F. Giralt, The use of pattern recognition and proper orthogonal decomposition in identifying the structure of fully-developed free turbulence, J. Fluids Engrg. 119 (1997) 289–296.
[10] G. Berkooz, P. Holmes, J.L. Lumeley, The proper orthogonal decomposition in the analysis of turbulent flows, Ann. Rev. Fluid Mech. 25 (1993) 539–575.
[11] P.B.V. Johansson, W.K. George, Proper orthogonal decomposition of an axisymmetric turbulent wake behind a disk, Phys. Fluids 14 (7) (2002) 2508–2514.
[12] S. Garnard, W.K. George, D. Jung, S. Woodwrad, Application of a "slice" proper orthogonal decomposition to the far field of an axisymmetric turbulent jet, Phys. Fluids 14 (7) (2002) 2515–2522.
[13] M. Farge, K. Schneider, G. Pellegrino, A.A. Wray, R.S. Rogallo, Coherent vortex extraction in three-dimensional homogeneous turbulence: Comparison between CVS-wavelet and POD-Fourier decompositions, Phys. Fluids 15 (10) (2003) 2886–2896.
[14] J.A. Ferré, F. Giralt, Pattern-recognition analysis of the velocity field in plane turbulent wakes, J. Fluid Mech. 198 (1989) 27–64.
[15] J.A. Ferré, J.C. Mumford, A.M. Savill, F. Giralt, Three-dimensional large eddy motions and fine-scale activity in a plane turbulent wake, J. Fluid Mech. 210 (1990) 371–414.
[16] A. Vernet, G.A. Kopp, Classification of turbulent flow patterns with fuzzy clustering, Engrg. Appl. Artificial Intelligence 15 (3–4) (2002) 315–326.
[17] J. Pallares, L. Davidson, Large-eddy simulations of turbulent heat transfer in stationary and rotating square ducts, Phys. Fluids 14 (8) (2002) 2804–2816.
[18] S. Gavrilakis, Turbulent-field modes in the vicinity of the walls of a square duct, Eur. J. Mech. B Fluids 16 (1997) 121–139.
[19] J. Pallares, L. Davidson, Large-eddy simulation of turbulent flows in stationary and rotating channels and in a stationary duct", Rep 00/03. Department of Thermo and Fluid Dynamics, Chalmers University of Technology Gothenburg, Sweden, 2000 (available online at http://www.tfd.chalmers.se/~lada/postscript_files/pallares_tfd00-03.pdf, as of 08/Feb/2005).
[20] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.
[21] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Machine Intell. PAMI-13 (8) (1991) 841–847.
[22] J. Jeong, F. Hussain, W. Schoppa, J. Kim, Coherent structures near the wall in a turbulent channel flow, J. Fluid Mech. 332 (1997) 185–214.

III. **Paper 2 :** “Boundary treatment in PIV image analysis”

MODELS, EXPERIMENTS AND COMPUTATION IN TURBULENCE
R. Castilla, E. Oñate and J.M. Redondo (Eds.)
© CIMNE, Barcelona, Spain 2004

BOUNDARY TREATMENT IN PIV IMAGE ANALYSIS

Gabriel Usera *
Instituto de Mecánica de los Fluidos e
Ingeniería Ambiental (IMFIA)
Universidad de la República
J. H. Reissig 565, 11300 Montevideo, Uruguay
Email: gusera@fing.edu.uy,
web page:http://www.fing.edu.uy/imfia

Anton Vernet and Josep A. Ferré
Departament de Enginyeria Mecánica
Universitat Rovira i Virgili
Campus Sescelades, Av. Paisos Catalans 26
43007 Tarragona, Spain
Email: anton.vernet@urv.net
web page: http://ecommfit.urv.es/

Abstract. A method to improve the accuracy of PIV image analysis near image boundaries is presented. The method, that can be applied to both iterative and non iterative correlation schemes, provides a rule to correct the location of the computed velocity vector when the interrogation area overlaps the image boundary. It is shown that the proposed method keeps the error level near boundary zones to the same level found far form the boundaries, outperforming standard algorithms in this aspect. The application of the method requires no modification of the correlation procedure since it is applied after each correlation step, furthermore it does no generate any significant deterioration of the time performance of the PIV analysis algorithm.

Keywords : Wall bounded flows, PIV

1. INTRODUCTION

Particle image velocimetry (PIV) has evolved into a well established whole field measurement technique in the last decade thanks to an intense development of both hardware and interrogation algorithms. For instance, high speed digital cameras have overcome an important limitation of traditional PIV systems, making time resolved PIV available at sampling frequencies of several hundred Hertz [1] and even higher [2].

The iterative pattern deformation algorithms have set the standard for increased accuracy and robustness in high gradient flows [3,4,5,6]. However, they suffer from high computational costs and the simpler, non-iterative, correlation algorithms still find a widespread use. The advent of time resolved PIV demand computational procedures that are less time consuming, since each record might contain several hundred pairs of images, increasing the need for faster algorithms. Thus, it seems that both iterative and non-iterative interrogation algorithms will coexist for some time and research should continue to address both cases.

While research aiming at algorithm development has been intense, it has been almost entirely centred in the conditions found away from the image boundaries, while less effort has been devoted to the peculiar conditions found when the interrogation process approaches the image boundary. This situation is of some concern since many important fluid mechanic problems require of accurate measurements near boundaries, as it is the case of boundary layers and wall bounded flows in general [1].

As it will be shown in this paper, a special treatment of the interrogation area near the image boundaries is required to obtain the same level of accuracy available at inner locations, since both iterative and non iterative standard interrogation algorithms introduce significant errors when the interrogation location is closer to the image boundary than half the interrogation area width. This situation is more critical in the case of the iterative pattern-deformation interrogation algorithms, where improper interrogation results obtained at the image boundaries might propagate through the iteration process, increasingly distorting the image and thus affecting also inner locations.

In the next section, the main source of interrogation errors near image boundaries will be identified and described, and the solution to this source of errors will be outlined. In section 3, an algorithm for the correct boundary treatment will be presented, and some implementation details will be discussed. In section 4, results obtained from the application of the proposed algorithm to both synthetic and real PIV images will be presented. Finally, conclusions will be derived in section 5.

2. INTERROGATION AREA NEAR IMAGE BOUNDARIES

As pointed out in the introduction, it is usually required to obtain reliable velocity (or displacement) field estimations at locations that are closer to the image boundaries than one half of the interrogation area size. That is the case in wall bounded flows and in general when using iterative pattern-deformation interrogation algorithms, as those presented in Nogueira et al.[4] and Scarano and Reithmuller[6]. There are several factors that limit the resolution of PIV measurements, and thus the capacity to obtain velocity vector estimations close to the image boundary. Ultimately the resolution will be limited by the size of particle images and the mean spacing between them. However, for the case of near image boundaries estimations, the much larger size of the interrogation area plays a determinant role.

In figure 1, a sketch of a generic PIV image is presented, with several interrogation areas placed at different locations. Interrogation areas (a) and (b) lay completely inside the image boundaries, since their geometric centres are not closer to any boundary than half the interrogation area size (IAs/2). Thus, in those cases no special treatment will be required. However, the interrogation areas (c) and (d) differ from the previous ones, since they lay only partially within the image boundaries, with the shaded parts of each area being placed outside the image boundaries.

In the case of the interrogation area (d) the geometric centre has been placed outside the PIV image. Therefore, it is clear that the resulting velocity vector estimation obtained form the interrogation process should be assigned to a point inside the PIV image, since the only valid information is provided by those pixels covered by the interrogation area that lays within the image boundary. From the discussion below it should also become clear that the geometric centre of the interrogation area is not the correct placement for the obtained velocity vector estimation in situations as those represented by cases (c) and (d).

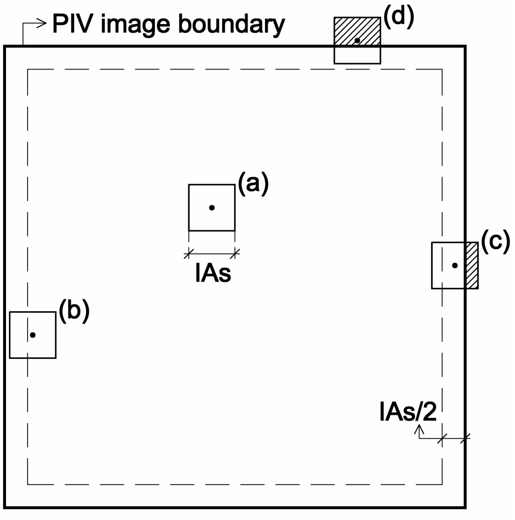


Fig 1. Sketch of a PIV image with boundaries, and samples of inner interrogation areas ((a) , (b)) and boundary affected interrogation areas ((c) , (d)).

Thus, when the situation described above occurs, caution must be exercised in order to correctly determine the actual location of the velocity vector estimation delivered by the interrogation process. In other words, the boundary proximity effect does not produces a necessarily bad estimate, but rather a *misplaced* estimate, whose location differs from the expected one. The sketch included in figure 2 illustrates this issue for a given weighting function. Weighting functions are applied in pattern correlation methods mainly to increase the influence of particle images closer to the interrogation area centre by augmenting its intensity relative to the particle images situated away from the interrogation area centre. In this work two different types of weighting functions, introduced below, are used. For inner locations, separated from the image boundary by more than half of the interrogation area size, the velocity vector estimation is correctly placed at the geometric centre of the interrogation area. Accordingly, the geometric centre P_1 of the interrogation area is coincident with the centre of mass G_1 of the weighting function.

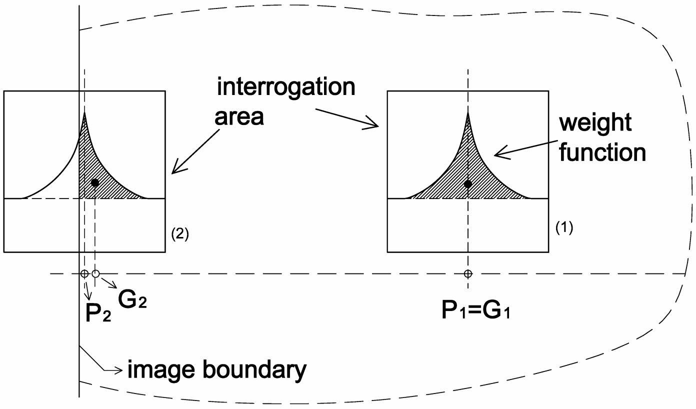
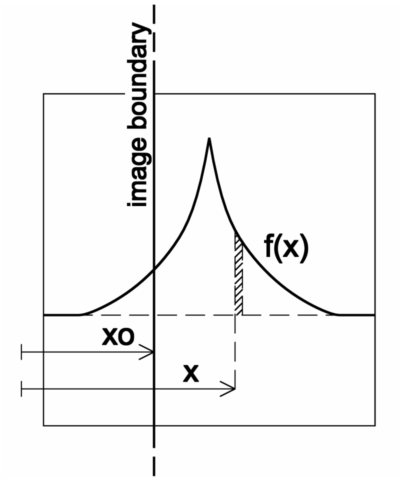


Fig 2. Sketch of the interrogation location displacement due to boundary cropping of the interrogation area.

However, at locations near the image boundary, the velocity vector estimation will be offset inwards due to the truncation of the weighted window. This effect can be approximated (at least at first order) by the actual placement of the centre of mass G_2 of the truncated weighted window, which would be the correct placement for a simple weighted mean estimator. Although the velocity vector estimation by interrogation algorithms is far from being a linear operation, the proposed modelling of the estimation location displacement seems a reasonable one, and will be proved to be accurate through testing with synthetic images in section 4.

The position of the centre of mass of the truncated weighting function is obtained through the first moment of the weighting function normalised by its total area, as shown in figure 3 and equation 1, where f(x) stands for the selected weighting function. Notice that the centre of mass **G** of the truncated weighted window will always fall within the image boundaries, even if the geometric centre of the weighted window falls outside the image. In fact, x(G) will always be larger than xo (see figure 3), as far as the weighting function is positive and the weighted area inside the image remains non null.



$$x(G) = \frac{\int_{x>xo} x \cdot f(x) \cdot dx}{\int_{x>xo} f(x) \cdot dx} \qquad \textbf{(1)}$$

Fig 3. Computation of the position of the centre of mass.

In figures 4 and 5, the results obtained by the application of this computation scheme are presented for two types of weighting functions. The first one will be referred to as the 'LFC-PIV' window, and is the weighting function proposed in Nogueira et al.[4] for their Local Field Correction PIV method. In fact, the iterative version of the algorithms used in this work follow largely the guidelines given in this work. The second one is the traditional Gaussian weighting function, and will be referred to as 'Gaussian'. The abscissa corresponds to the location estimations for the centre of mass of the weighting function, for each position of the geometric centre of the interrogation area, which is given in the ordinates. All co-ordinates are given relative to the boundary, positive values corresponding to locations inside the image, while negative values correspond to points outside the image.

In can be seen that significant displacements of the centre of mass relative to the geometric centre occur, and thus that the proposed correction will be significant. For instance, when the geometric centre of the interrogation area is placed at the image boundary, the actual position of the obtained estimate would be more than six pixels away for the case of 48 pixels wide LFC-PIV weighting function (figure 4) and almost 4 pixels away for the case of 32 pixels wide Gaussian weighting function (figure 5). That the proposed correction is also accurate will be shown later, in section 4, where results from tests with synthetic images are discussed.

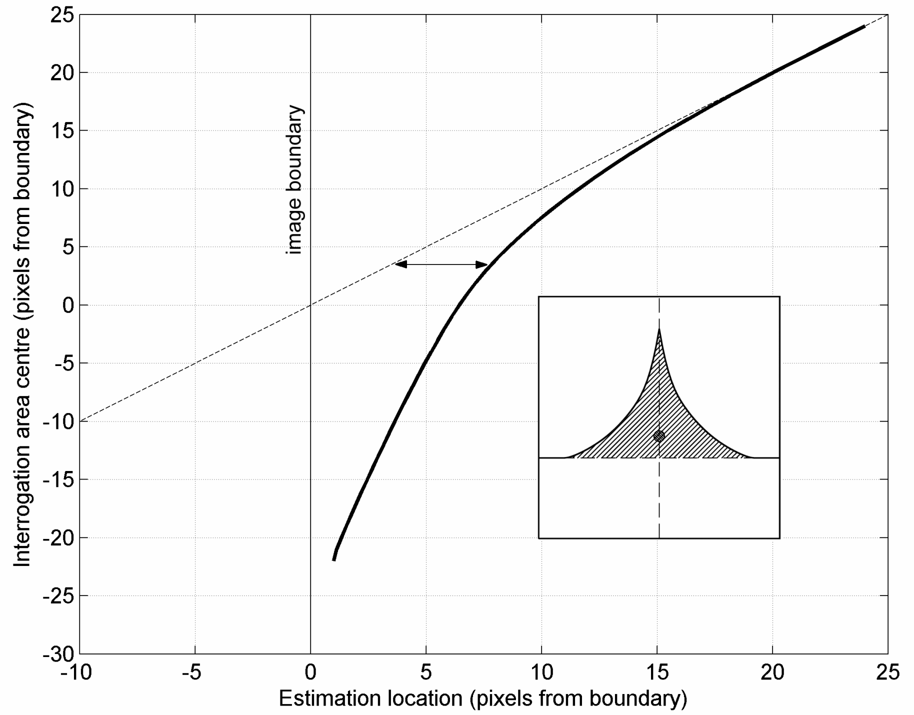


Fig 4. Displacement of the velocity vector estimation location due to the boundary cropping of the interrogation area. LFC-PIV-48 pixels wide weighting function.

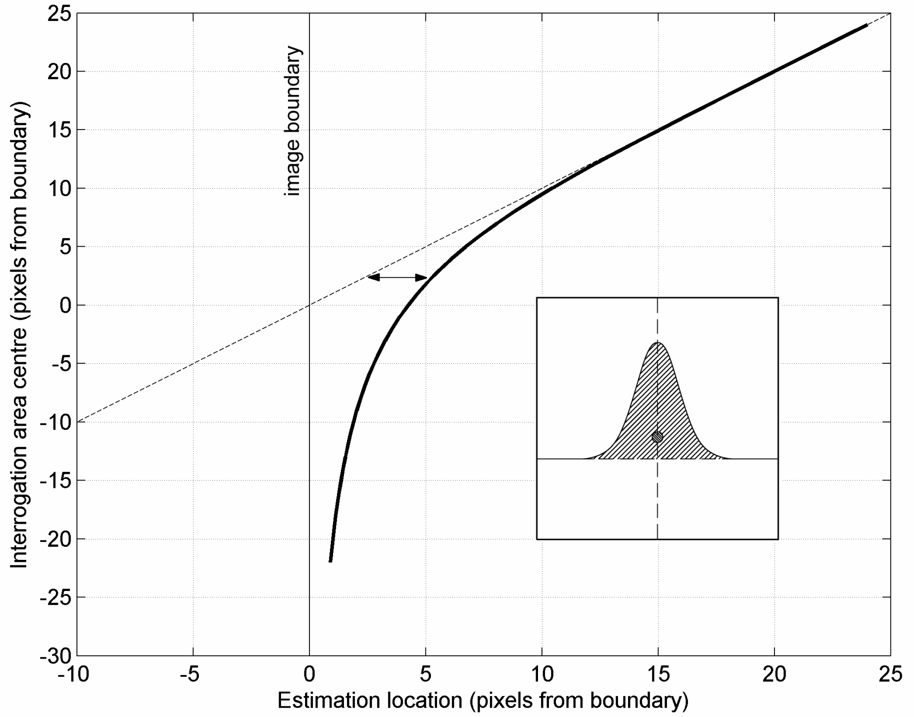


Fig 5. Displacement of the velocity vector estimation location due to the boundary cropping of the interrogation area. Gaussian-32 pixels wide weighting function.

The displacement lines in figures 4 and 5 end a few pixels away from the boundary, reflecting the fact that it is not possible to obtain a measurement arbitrarily close to the boundary. Actually, the interrogation area centre can hardly be pushed more than 10 pixels out of the image boundary without a significant loss of correlation and robustness of the peak detection process, although this limit depends on the width of the interrogation windows and weighting functions, and of course on the image properties. Still, displacement estimations closer to the boundary than five pixels are achievable with the proposed method, as will be shown in section 4. Of course, as mentioned above, the particle image size and mean spacing between particles will ultimately limit the reliability of the estimations obtained close to the image boundary.

In figure 6 a sample velocity profile is presented, with both the raw measurement and the profile compensated for the described effect being pictured. A systematic algorithm to produce such compensations will be described in the following section, while comprehensive results from tests with both real images and synthetic images will be presented in section 4. The aim of presenting here figure 6 is to clarify the boundary effect and how the displacement compensation works, while quantitative assessments on the benefits and weakness of the method will be deferred to section 4.

The velocity profile presented in figure 6 corresponds to a boundary layer, the wall being placed at the image boundary, and coincident with the abscissa origin. The image was processed with an 48 pixels wide LFC-PIV weighting function, with an 8 pixels grid spacing. Results from a non iterative run are presented. The ‘circles’ line (-o-) corresponds to the raw measured profile, while the ‘squares’ line (-□-) corresponds to the profile compensated for the estimation location displacement due to boundary cropping of the weighting function, following figure 4. The arrows indicate the corresponding points in the raw and compensated profiles.

The first aspect to be noticed in figure 6 is that compensated estimation locations lay within the flow domain, even if the interrogation area centre lies outside the flow domain, in this case, to the left of the image boundary or wall at negative pixel co-ordinates.

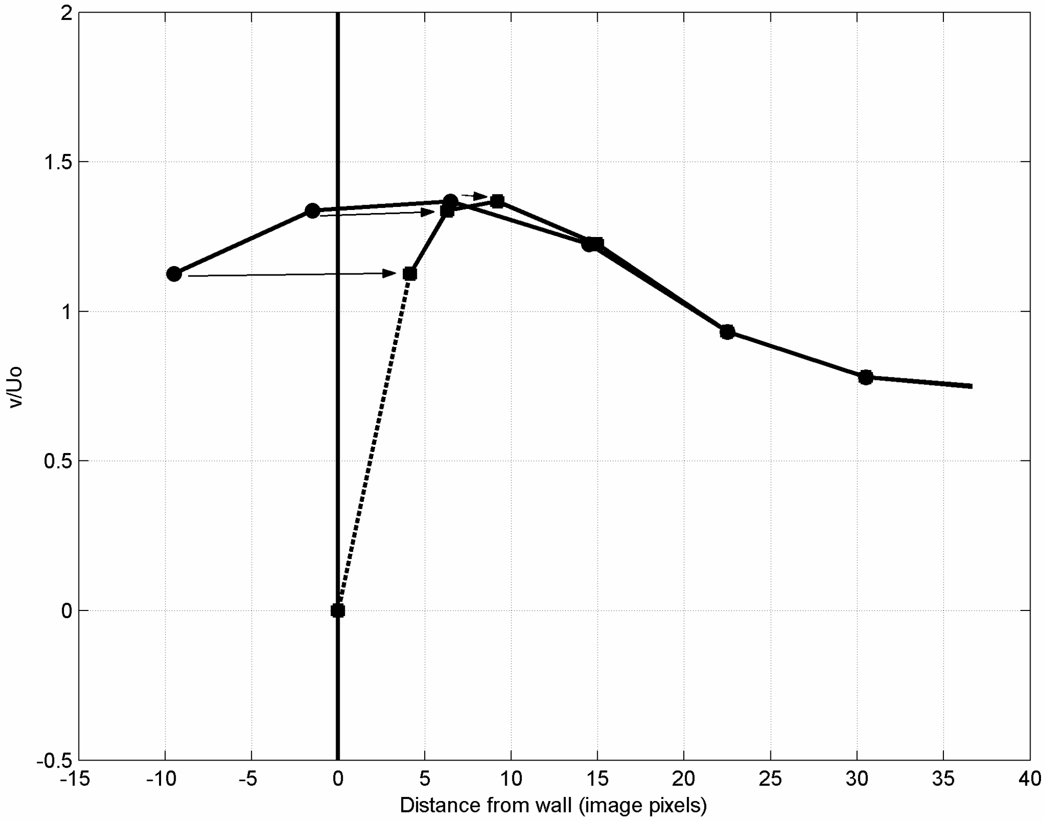


Fig 6. Example of the velocity profile correction due to estimation location displacement. Raw measured profile (-o-), and boundary compensated profile (-□-)

Also, in figure 6 it can be seen that negligible compensation is produced for points whose distance to the wall is greater than half the weighting function width, with the interrogation area cropping due to the image boundary affecting only the points closer to the image boundary.

One piece of information included in the compensated profile of figure 6 comes from a different source. The knowledge that the velocity is null at the wall, due to the non-slip condition, has been used to extend the profile up to the wall from the first compensated location. This section of the profile is presented with a dashed line. For non-iterative interrogation methods, this extension is of little interest, and might even be questioned since it introduces information that does not directly follows from the individual experiment. However, in the case of iterative pattern deformation algorithms, it is of greater importance since every pixel in the image has to be associated with a velocity vector estimation to produce a new deformed image pattern.

While in this work only wall boundaries will be considered, the proposed method can be extended to other type of boundary conditions, as specified flow inlets and null gradient open boundaries. Also the proposed treatment could be adapted to deal with inner flow boundaries and corners within the PIV image, as those resulting from the presence of obstacles or flow domain boundaries.

3. ALGORITHM FOR BOUNDARY TREATMENT

In this section, a systematic procedures to apply the ideas described in the preceding paragraphs in order to improve the interrogation process near image boundaries is presented. Two different cases are considered because of their different requirements. In the case of non iterative interrogation algorithms, a simple estimation location compensation scheme for the velocity vectors is proposed. For the case of iterative pattern-deformation algorithms, the procedure has to be further complemented with an appropriate interpolation scheme able to produce velocity vector estimations at all pixel locations within the image for the next pattern deformation step.

As stated above, for the case of non iterative interrogation algorithms it is only necessary to compensate the location of those velocity vector estimations affected by the boundary proximity. Following section 2, the algorithm should consist of the following steps :

1) Apply the interrogation algorithm at all selected locations (which usually are distributed in a regular grid).

2) Determine the locations that require treatment. These locations would be those that are closer to any image boundary than half the width of the interrogation area weighting function.

3) For each location identified in 2) as requiring treatment, obtain the actual placement of the velocity vector estimation. These values can be obtained directly from figures 4 and 5, for the weighting functions considered there, or computed as stated in equation 1 for any given weighting function.

It should be noted that the proposed procedure will, in general, produce a non regular grid of velocity vector estimations. Whether this is a desirable effect or not would, in general, depend on the specific post-processing to be applied to the data. However, it can be seen from figures 4 and 5, and from the example in figure 6, that the obtained grid will exhibit shorter distances between estimation locations as they approach the wall, which is in general a benefit for the description of boundary layers, since stronger gradients are present in this zone.

For the case of iterative pattern-deformation interrogation algorithms, the above steps have to be complemented by a fourth interpolation step :

4) Interpolate the displacements from the non regular grid obtained in 3), as required for the pattern deformation step. Usually this will mean to obtain displacement estimations at each pixel location in the image.

To successfully accomplish step 4) the velocity vector estimations at the image boundary are required. For the case analysed in this work, namely that of wall bounded flows, null velocity at the wall boundary should be assumed (as was suggested by the dashed line in figure 6).

Finally, it should be noted that the correlation procedure (step 1) requires no modification to implement the compensation steps (2-3). Furthermore, these compensation steps are not computational intensive. Thus the overall compensation procedure introduces no significant detriment of the overall time performance of the PIV analysis algorithm.

4. RESULTS AND DISCUSSION

In this section the results obtained from the evaluation of the proposed method for the treatment of PIV image interrogation near boundaries are preented. Quantitative results and error estimations will be drawn from tests with synthetic images, while, at the end of this section, an example of application to real PIV images obtained from an experimental model will be presented.

4.1 Synthetic linear velocity profile

Synthetic images used for the results presented below were generated with an in-house software. The selected image size was 256x256 pixels, with mean spacing between particles of 6.3 pixels and mean particle size of 2.2 pixels. Particles were randomly distributed, and so were their sizes about their mean value and the out of plane displacement in order to mimic the particle brightness variations and the loss of correlation between consecutive images. It should be kept in mind, however, that we are not pursuing here comprehensive testing of the robustness and performance of the interrogation procedure subject to different image qualities. Rather we intend to test a proposed method to correct a systematic error being made by standard interrogation algorithms in near boundary situations. The problem being addressed, as well as the proposed solution, are largely insensible to image properties, provided that the seeding is kept sufficient and homogeneous near the boundaries.

To start with, a simple linear displacement profile along a wall was considered, as pictured in figure 7. The constant gradient **k** was allowed to take values of 0.02, 0.05 and 0.10 pixels/pixels.

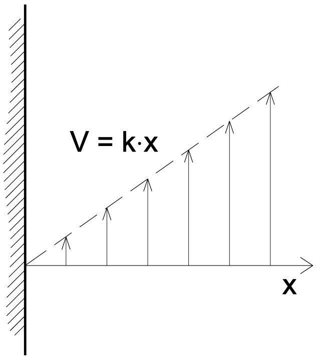


Fig. 7. Linear velocity profile.

The plots presented in figures 8, 9 and 10 correspond to the results obtained by the non iterative version of the interrogation algorithm, with different weighting functions: 48 and 32 pixels wide Gaussian window and 48 pixels wide LFC-PIV window, respectively. The expected behaviour near the boundary is verified in these figures. The profile obtained with the standard interrogation algorithm increasingly deviates from the exact result as the boundary is approached. Notice that, while all interrogation locations have been plotted, those obtained at negative positions are meaningless for the case of the standard algorithm. These same displacement estimations are used to construct the boundary compensated profile, as described in section 3, which shows no *systematic* deviation from the exact profile as the boundary is approached.

In each figure, it can be observed that the deviation for the non compensated case increases, at any location, with the displacement field gradient **k**. This behaviour was to be expected since, at any given location, the deviation will be roughly proportional to the gradient times the distance between the geometric centre of the interrogation window and its weighted centre of mass.

At the same time, this last distance increases for increasing weighting windows widths, and thus the deviation should also do so. Comparing figures 8 and 9, which correspond to the 48 and 32 pixels wide Gaussian windows, respectively, this effect is also observed. In figure 8 the deviation starts to be significant at about 20 pixels from the boundary, and reaches a magnitude of almost 0.6 pixels at the boundary location for the case **k**=0.10. On the other hand, in figure 9, the deviation is noticeable only 10 pixels away from the boundary and falls short of 0.5 pixels at the boundary for the same value of the gradient **k**. In fact the use of slender interrogation areas, has already being proposed before, as a mean to increase the accuracy of PIV measurements near walls[7].

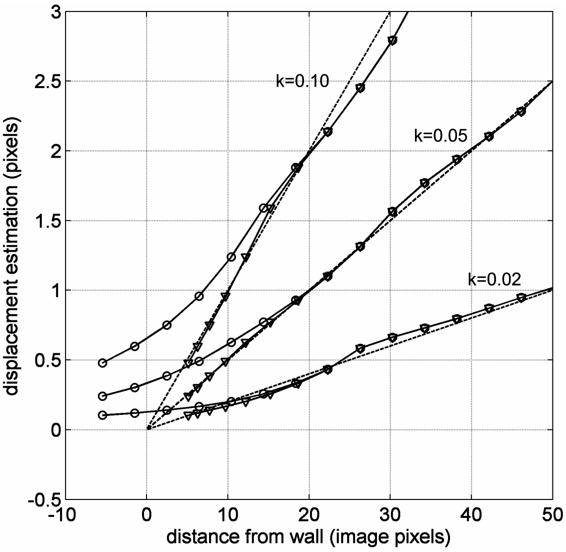


Fig 8. Linear velocity profiles processed with the 48 pixels wide Gaussian window, non iterative interrogation method. Standard interrogation (-o-), boundary compensated interrogation (-Δ-) and exact profile (--)

The preceding paragraphs stress the fact that the significance of the systematic error being addressed here largely depends on the gradient of the displacement field near the image boundaries. That is, measurements in flow fields which are mostly uniform near image boundaries would be little affected, while measurements in flows exhibiting large gradients near the boundaries, as boundary layer flows, for instance, would be affected the most.

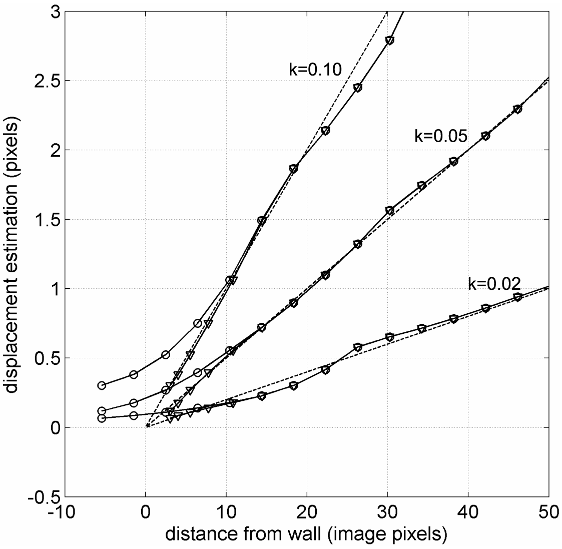


Fig 9. Linear velocity profiles processed with the 32 pixels wide Gaussian window, non iterative interrogation method. Standard interrogation (-o-), boundary compensated interrogation (-Δ-) and exact profile (--)

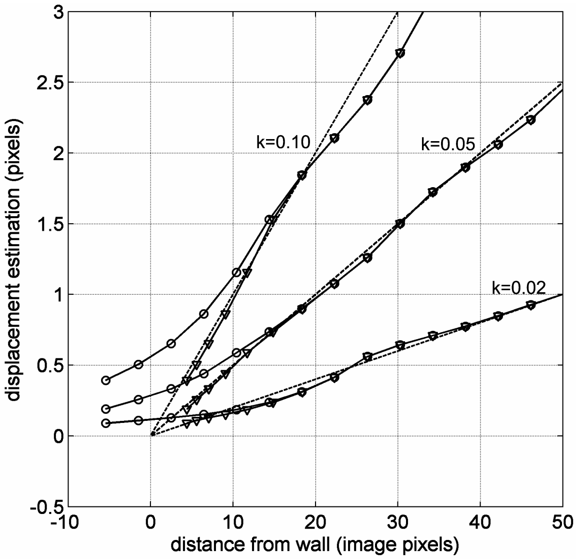


Fig 10. Linear velocity profiles processed with the 48 pixels wide LFC-PIV window, non iterative interrogation method. Standard interrogation (-o-), boundary compensated interrogation (-Δ-) and exact profile (--)

It must also be stressed that, in the proposed correction method, only the *placement* of the estimations is being altered, while their *values* are not. Thus estimation errors inherent to the interrogation process should remain, but only at roughly the same level they occur far from the boundary. This can be clearly observed in figure 11, where the errors, relative to the displacement value, have been plotted for all the test cases considered above, with 48 pixels wide weighting windows. It is seen that the relative error greatly increases, for the standard algorithm, as the boundary is approached far exceeding the 50% level. On the other hand, the relative error only moderately increases for the boundary compensated case, roughly remaining at a similar level to that found far from the boundary, almost entirely below the 20% level and mostly below 10%. A decrease of the error below that level was not to be expected, since the method used to determine the *value* of the velocity vector estimation has remained the same.

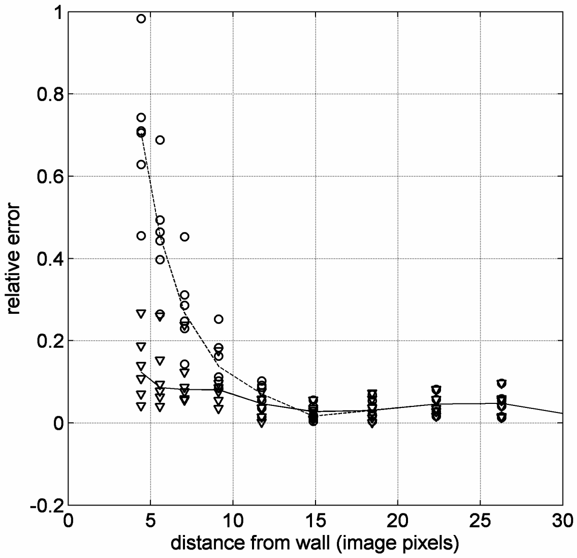


Fig 11. Relative error plot. Standard interrogation (o, dashed line) and boundary compensated interrogation (Δ, solid line)

4.2 Synthetic exponential velocity profile

Next, a displacement field with a profile closer to those found in typical boundary layers will be considered, as pictured in figure 12. The profile, modelled after a simple exponential function, verifies the non-slip condition at an imaginary wall placed at the image boundary, and reaches a free stream displacement of 3 pixels, at a distance of 20 pixels from the wall, as plotted in the dashed line of figure 13. With a mean gradient of 0.3 pixels/pixel in the first 10 pixels, the selected profile presents a challenge to non iterative interrogation algorithms and offers an opportunity to test the proposed boundary treatment method with both non iterative interrogation algorithms and iterative pattern-deformation interrogation algorithms. The latter are expected to outperform in moderate and high gradient situations as the one presented here. It will be seen, however, that they still suffer from the same type of boundary induced distortion, verified for the standard non iterative algorithm. Therefore, the use of the proposed boundary compensated method again results in an improved accuracy near the boundary.

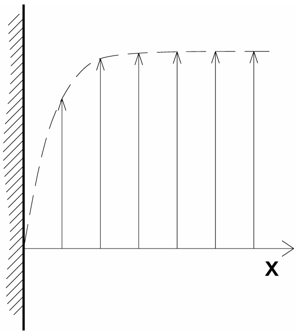


Fig. 12. Exponentially decaying velocity profile

Results from the interrogation with non iterative algorithms are presented in figure 13 for the present case. While the match to the prescribed profile is less accurate, as expected due to the increased gradient, the improvement due to the boundary compensation is still evident. It should be kept in mind that the boundary compensation method proposed here cannot recover from inaccuracies incurred at the interrogation time, as those related to moderate or high gradient situations. To overcame this issue iterative pattern deformation interrogation algorithms should be used, in combination with the boundary compensation method, as will be presented next.

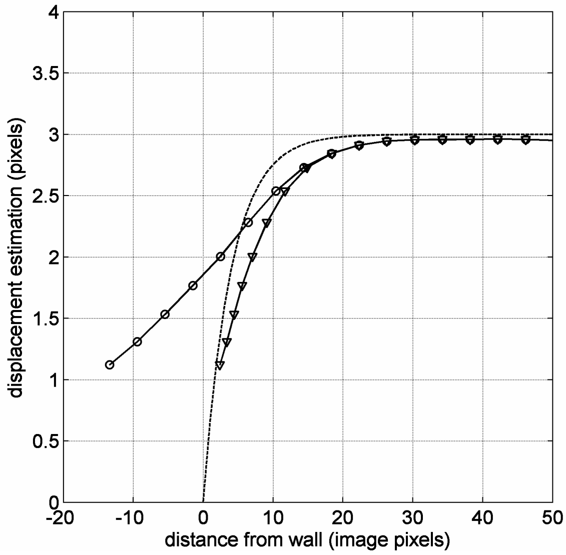


Fig 13. Exponential velocity profile processed with the 48 pixels wide LFC-PIV window, non iterative interrogation method. Standard interrogation (-o-), boundary compensated interrogation (-Δ-) and exact profile (--)

Figure 14 summarizes the results for the iterative algorithms. Results from the first iteration (figure 14a) and the fifth iteration (figure 14b) are presented. While the iteration could be continued further for increased accuracy, five iterations are enough to show the improvements delivered by the boundary compensation method.

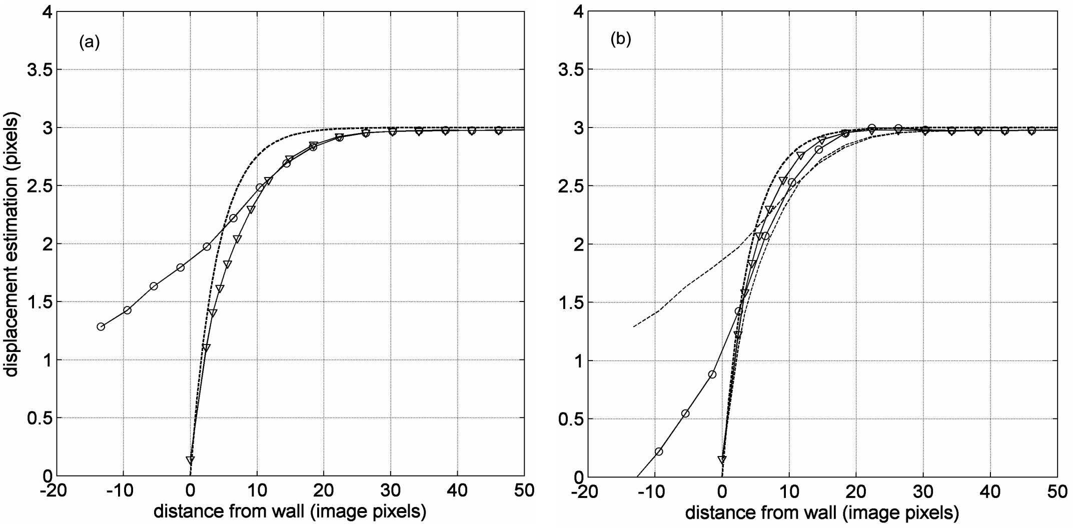


Fig 14. Exponential velocity profile processed with the 48 pixels wide LFC-PIV window, iterative pattern deformation interrogation method. Standard interrogation (-o-), boundary compensated interrogation (-Δ-) and exact profile (--). (a) First iteration. (b) After 5 iterations. In frame (b), first iteration profiles are reproduced in dashed line, for easier comparison.

First of all, an improved match to the prescribed profile is obtained, with respect to the non iterative case, for both the standard and boundary compensated interrogation. However, in the area closer to the wall, the match is much improved for the boundary compensated case. This can be quantitatively observed in figure 15, where the relative error has been plotted for both standard and boundary compensated methods, and for different iteration steps.

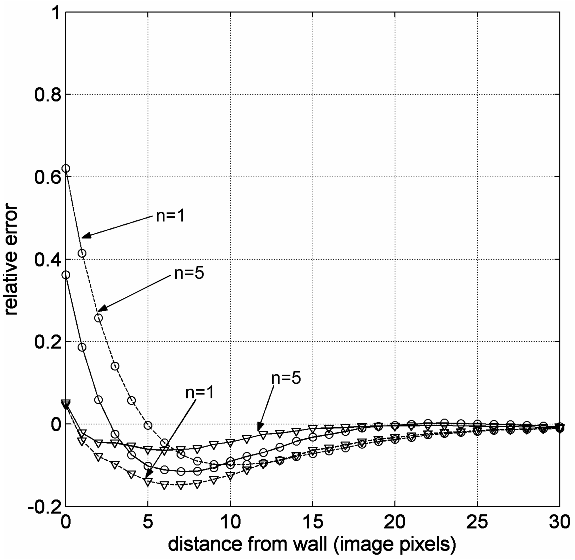


Fig 15. Relative error plot for iterative pattern deformation algorithm. Standard interrogation (o, dashed line n=1, solid line n=5) and boundary compensated interrogation (Δ, dashed line n=1, solid line n=5)

4.3 Experimental data analysis

In this section, the application of the proposed method to the analysis of real PIV images taken from an experimental set up is presented. In the sketches of figure 16, front and side views of the experimental model are shown, with the main paths of flow being indicated. The model fluid is water, it operates in closed circuit and results will be presented for a Reynolds number of Re=550 based on the width of the square entrance section which is 24 mm.

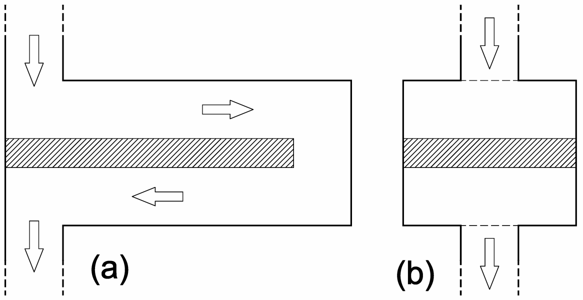


Fig 16. Sketch of experimental model. Side view (a) and front view (b)

Figure 17 is given solely with the purpose of showing the general features of the flow. Based on a numerical simulation, vortical structures are visualised by means of λ_2 iso-surfaces[8]. It can be seen that upper part of the flow, for which PIV results will be presented below, is dominated by two counter rotating vortical structures. The non stationary characteristic of the flow is also apparent in figure 17.

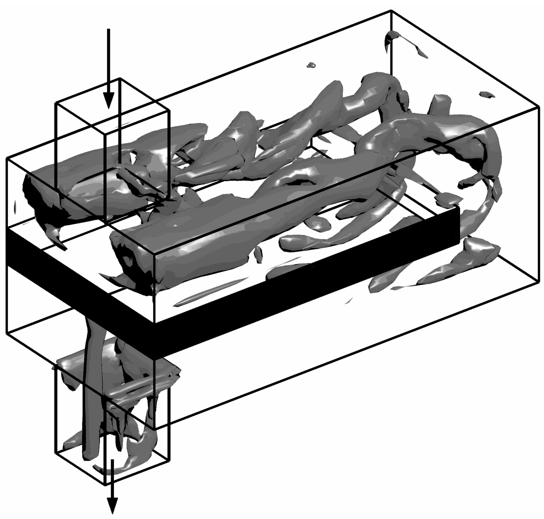


Fig 17. Vortical structures educed by λ_2 iso-surfaces from numerical simulation.

Results from PIV measurements in the central cross section of the upper channel will be presented next. The analysed images were 160 by 480 pixels and the sampling rate was 50 Hz. The iterative versions of the interrogation algorithm with a 48 pixels wide LFC-PIV window and a 4x4 pixels grid was used, with results being reported after 5 iterations. In figure 18 a full section velocity vector plot is presented, obtained with the proposed boundary effect correction in place. The resulting non uniformity of the grid near the boundaries can be appreciated. The two large counter rotating vortical structures are evident from figure 18, with smaller re-circulating zones appearing, specially near the lower left corner.

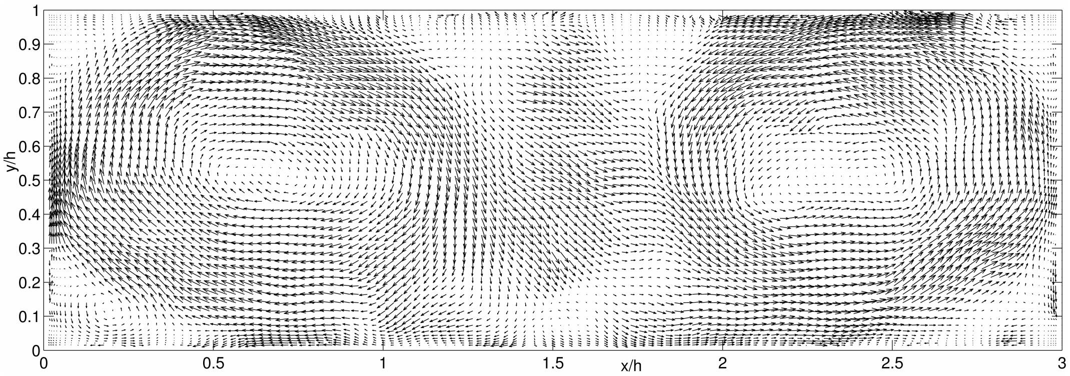


Fig 18. Upper channel central cross section vector plot, from PIV experiments.

In figures 19 and 20, selected vertical and horizontal velocity profiles obtained from both the standard version of the algorithm and the boundary compensated version are given. In figure 19 the normalised vertical velocity profile along the middle (y/h=0.5) horizontal line is presented, while in figure 20 the normalised horizontal velocity vertical profile along the vertical line passing through x/h=0.75 is presented.

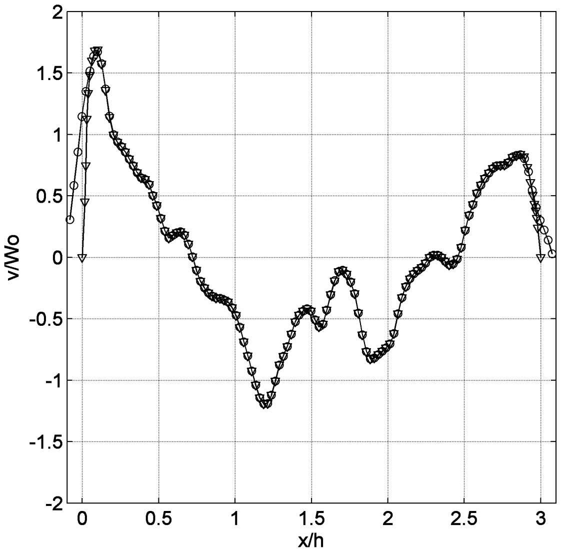


Fig 19. Normalised vertical velocity component profile at y/h=0.5. Standard interrogation (-o-) and boundary compensated interrogation (-Δ-).

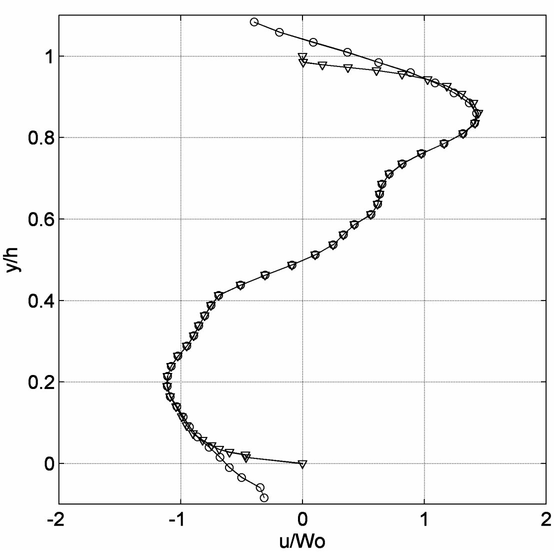


Fig 20. Normalised horizontal velocity component profile at x/h=0.75. Standard interrogation (-o-) and boundary compensated interrogation (-Δ-).

While a quantitative assessment of accuracy is not possible for this real flow cases, the overall improved aspect of the obtained boundary profiles supports the applicability of the proposed boundary compensation method to real PIV measurements.

5. CONCLUSIONS

The importance of careful treatment of interrogation near PIV image boundaries has been stressed. Standard algorithms produce distorted results near image boundaries due to the cropping of the interrogation area and weight function which produces a misplacing of the velocity vector estimate. A solution to this problem has been proposed based on a first order approximation to this distortion, through the computation of the actual placement of the cropped weighting function centre of mass. It has also been shown how this correction can be implemented for both non iterative interrogation algorithms and iterative pattern-deformation interrogation algorithms.

The algorithm has been tested with both synthetic and real PIV images of wall bounded flows, and for both iterative and non iterative algorithms. In all cases a significant improvement of the accuracy obtained near the wall has been verified. The proposed method is also straightforward to implement and implies no degradation of overall algorithm performance, since it requires no modification of the interrogation process, the correction being applied afterwards.

Similar methods to the one proposed here could be derived for use with other boundary conditions commonly encountered in fluid dynamic studies, as specified inlet profiles or null gradient outlets.

ACKNOWLEDGEMENTS

This work was financially supported by grants 2002 SGR00189 from AGAUR Generalitat de Catalunya, DPI2003-06725-C02-01 from DGI, Ministerio de Ciencia y Tecnología y Fondos FEDER, and a pre doctoral grants from Fundación URV and Programa de Desarrollo Tecnológico assigned to Gabriel Usera.

REFERENCES

[1] Usera G., Vernet A., Ferre J.A., (2004) *Considerations and improvements on analysing algorithms for time resolved PIV of turbulent wall bounded flows*. 12th International Symposium. Applications of Laser Techniques to Fluid Mechanics.

[2] Etoh G. T., Takehara K., Takano Y., (2002) *Development of high-speed video cameras for* dynamic *PIV*. Journal of Visualization, 5 (3) 213-224

[3] Nogueira J., Lecuona A., Rodríguez P. A., (1999*) Local field correction PIV : on the increase of accuracy of digital PIV systems*. Experiments in fluids, 27, 107-116.

[4] Nogueira J., Lecuona A., Rodríguez P. A., (2001) *Local field correction PIV, implemented by means of simple algorithms and multigrid versions*. Measurement and Science Technology, 12, 1911-1921

[5] Lecordier B., Trinité M., (2003) *Advanced PIV algorithms with image deformation – Validation with synthetice images of turbulent flow*. EuroPIV 2 Workshop on Particle Image Velocimetry, 13

[6] Scarano F., Riethmuller M. L., (2000) *Advances in iterative multigrid PIV image processing*. Measurement and Science Technology, S51-S60.

[7] Scarano F., (2003) *Development and assessment of non-isotropic resolution in PIV.* EuroPIV 2 Workshop on Particle Image Velocimetry, 12

[8] Jeong J., Hussain F., (1995) *On the identification of a vortex*. Journal of Fluid Mechanics, 285, 69-94

IV. **Paper 3 :** “A parallel block-structured finite volume method for flows in complex geometry with sliding interfaces”

Flow Turbulence Combust (2008) 81:471–495
DOI 10.1007/s10494-008-9153-3

A Parallel Block-Structured Finite Volume Method for Flows in Complex Geometry with Sliding Interfaces

G. Usera · A. Vernet · J. A. Ferré

Received: 13 April 2007 / Accepted: 7 April 2008 / Published online: 29 April 2008

Abstract An implementation of the finite volume method is presented for the simulation of three dimensional flows in complex geometries, using block structured body fitted grids and an improved linear interpolation scheme. The interfaces between blocks are treated in a fully implicit manner, through modified linear solvers. The cells across block interfaces can be matching *one-to-one* or *many-to-one*. In addition, the use of sliding block interfaces allows the incorporation of moving rigid bodies inside the flow domain. An algebraic multigrid solver has been developed that works with this block structured approach, speeding up the iterations for the pressure. The flow solver is parallelized by domain decomposition using OpenMP, based on the same grid block structure. Application examples are presented that demonstrate these capabilities. This numerical model has been made freely available by the authors.

Keywords Finite volume · Sliding interfaces · Block structured · OpenMP

G. Usera was supported by FPI/DPI2003-06725-C02-01 from DGI, Ministerio de Educación y Cultura y Fondos FEDER, Spain, grants 33/07 PDT, I+D CSIC, Uruguay.

G. Usera (✉)
IMFIA, Universidad de la República, J.H. Reissig 565,
11300 Montevideo, Uruguay
e-mail: gusera@fing.edu.uy

A. Vernet · J. A. Ferré
Departament d'Enginyeria Mecánica, Universitat Rovira i Virgili,
Av. Paisos Catalans 26, 43007 Tarragona, Spain

A. Vernet
e-mail: anton.vernet@urv.net

J. A. Ferré
e-mail: josep.a.ferre@urv.net

1 Introduction

Completely structured body fitted grids allow to adjust the spatial discretization to the domain boundaries only in cases of rather simple geometry. Building the grid out of blocks of curvilinear structured grids provides greater flexibility, somewhat in between that of completely structured grids and unstructured grids. While unstructured grids allow the treatment of very complex domains, they require greater computational effort specially through the use of special linear solvers. Body fitted, block structured grids are often enough for moderately complex domains, while keeping the efficiency of methods designed for structured grids as long as the interfaces between blocks are treated implicitly by the linear solvers [2, 10].

The block structured grid approach fits also other purposes like the local refinement of the grid [6], the computation of flows that interact with moving rigid bodies inside the domain, or the application of parallel programming through domain decomposition [2] and OpenMP.

One drawback in the use of body fitted grids is that standard linear interpolation practices, like the popular central differencing scheme (CDS), suffer from accuracy degradation in non-orthogonal grids. Improved linear interpolation practices, that do not increase the bandwidth of the coefficient matrices, have been recently proposed in [7, 9].

The three dimensional numerical method presented here has been developed jointly by Rovira i Virgili University and the University of the Republic, on the basis of the two dimensional flow solver *caffa* [10]. Added features include full block structured grid support, sliding interfaces, an improved linear interpolation scheme and an algebraic multigrid (AMG) solver. It has been suggested by other authors that fluid dynamics research would benefit from the availability of more public and open source codes [15]. The authors agree with this view and welcome the use and further development of this model by other researchers. For this purpose the code is freely available through the website.[1]

2 Mathematical Model

The mathematical model comprises the mass (1) and momentum (2) balance equations for an incompressible Newtonian fluid, with the Boussinesq approximation for buoyancy effects due to temperature induced small density variations:

$$\int_S \left(\vec{\boldsymbol{v}} \cdot \hat{\boldsymbol{n}}_s\right) dS = 0 \tag{1}$$

$$\begin{aligned}\int_\Omega \rho \frac{\partial u}{\partial t} d\Omega + \int_S \rho u \left(\vec{\boldsymbol{v}} \cdot \hat{\boldsymbol{n}}_s\right) dS = \int_\Omega \rho\beta \left(T - T_{ref}\right) \vec{\boldsymbol{g}} \cdot \hat{\boldsymbol{e}}_1 d\Omega \\ + \int_S -p\hat{\boldsymbol{n}}_S \cdot \hat{\boldsymbol{e}}_1 dS + \int_S \left(2\mu \boldsymbol{D} \cdot \hat{\boldsymbol{n}}_S\right) \cdot \hat{\boldsymbol{e}}_1 dS\end{aligned} \tag{2}$$

[1] www.fing.edu.uy/imfia/caffa3d.MB

These equations hold in any portion Ω of the domain, being S the boundary of Ω and $\hat{\boldsymbol{n}}_s$ the outward normal vector at the boundary S. The momentum balance equation (2) has been expressed for the first component u of the velocity vector $\vec{\boldsymbol{v}} = (u, v, w)$, with similar expressions holding for the other components. The buoyancy term in equation (2) involves the density ρ, thermal expansion coefficient β and temperature T of the fluid, a reference temperature T_{ref} and gravity $\vec{\boldsymbol{g}}$. The viscosity μ of the fluid and the symmetric deformation tensor $\boldsymbol{D}$ were used for the viscous term.

The conservation law (3) for a generic passive scalar ϕ with diffusion coefficient Γ is also considered, which of course includes the heat equation as a particular case. Other scalar transport equations can be easily incorporated, for example to construct turbulence models or to address wet air processes which include evaporation and condensation.

$$\int_{\Omega} \rho \frac{\partial \phi}{\partial t} d\Omega + \int_{S} \rho\phi \left(\vec{\boldsymbol{v}} \cdot \hat{\boldsymbol{n}}_s\right) dS = \int_{S} \Gamma \left(\nabla \boldsymbol{\phi} \cdot \hat{\boldsymbol{n}}_s\right) dS \tag{3}$$

These equations were presented in their integral form, in compliance with the finite volume method and promoting a conservative formulation. The discretized equations will be obtained by applying equations (1, 2 and 3) to each volume element.

3 Domain Discretization

The domain is covered by blocks of structured grid. Each block is composed by an arrangement of non regular hexahedra as the one of Fig. 1. Within each block, the hexahedra have each exactly six neighbours, do not overlap and leave no empty space in between.

The approximation of the integral expressions of equations (1, 2 and 3) can be done efficiently using precomputed geometrical properties. From the grid specification the coordinates of all eight vertices in each volume element are known. Other required properties are: the volume $\Delta\Omega$ of the element, the coordinates of its baricenter P, and the surface normal vectors $\vec{\boldsymbol{S}}_c$ and coordinates of baricenter c for each face $(c = w, e, s, n, b, t)$. The normal surface vector $\vec{\boldsymbol{S}}_c$ is defined with its norm equal to the surface area of the face. Each volume element face is a quadrilateral, *not necessarily planar*, so its properties are computed from the composition of two

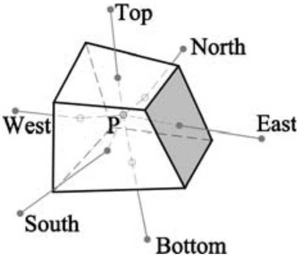


Fig. 1 Volume element with six neighbours

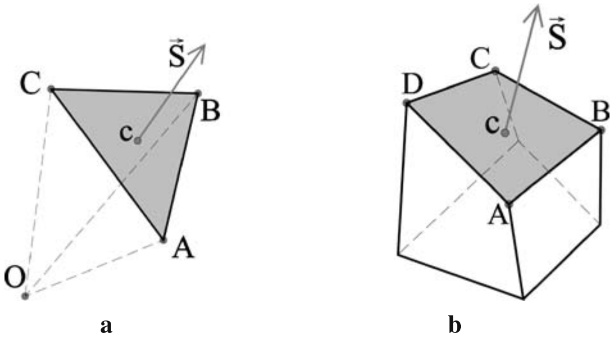


Fig. 2 **a** Triangular surface element with normal surface vector $\vec{S}$ at baricenter *c*. **b** Volume element with normal surface vector $\vec{S}$ at baricenter *c* of top face

triangular elements. Considering the triangular surface element ABC of Fig. 2a), equations (4) hold for the surface normal vector $\vec{\boldsymbol{S}}_{ABC}$ and the baricenter $\boldsymbol{c}_{ABC}$:

$$2\vec{\boldsymbol{S}}_{ABC} = \overrightarrow{\boldsymbol{AB}} \times \overrightarrow{\boldsymbol{AC}} \qquad 3\boldsymbol{c}_{ABC} = \left(\overrightarrow{\boldsymbol{OA}} + \overrightarrow{\boldsymbol{OB}} + \overrightarrow{\boldsymbol{OC}}\right) \tag{4}$$

Properties of face ABCD in Fig. 2b, are computed applying equations (5) and (6) to the triangular surface elements ABC and ACD:

$$\vec{\boldsymbol{S}}_{ABCD} = \vec{\boldsymbol{S}}_{ABC} + \vec{\boldsymbol{S}}_{ACD} \tag{5}$$

$$\boldsymbol{c}_{ABCD} \cdot \left|\vec{\boldsymbol{S}}_{ABCD}\right| = \boldsymbol{c}_{ABC} \cdot \left|\vec{\boldsymbol{S}}_{ABC}\right| + \boldsymbol{c}_{ACD} \cdot \left|\vec{\boldsymbol{S}}_{ACD}\right| \tag{6}$$

The volume $\Delta\Omega$ of an element Ω can be computed by application of the Gauss theorem, ensuring a conservative formulation where the total volume of the domain is the sum of volumes of each element:

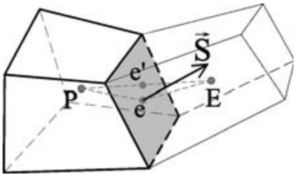
$$\Delta\Omega = \int_{\Omega} d\Omega = \int_{\Omega} \nabla \cdot \left(x\hat{\boldsymbol{e}}_1\right) d\Omega = \int_{S} x\hat{\boldsymbol{e}}_1 \cdot \hat{\boldsymbol{n}}_S dS = \sum_{c} x_c S_c^x \tag{7}$$

4 Interpolation and Gradient Approximation

4.1 CDS interpolation

The volume element of Fig. 3, centered in *P* and related to neighbour *E* through the *east* face, will be used to explain CDS interpolation and the computation of gradients. A collocated arrangement of variables (u, v, w, p, ϕ) is used so interpolation and

Fig. 3 Sketch of volume element *P* with neighbour *E*



gradient approximation are the same for all fields. The interpolated value ϕ_e of a scalar field ϕ at the baricenter of the *east* face, using cell center values ϕ_P and ϕ_E, is:

$$\phi_e = \phi_E \cdot \alpha_{PE} + \phi_P \cdot (1 - \alpha_{PE}) + (\nabla \boldsymbol{\phi})_e \cdot \overrightarrow{\boldsymbol{e'e}} \tag{8}$$

where the interpolation coefficient (α_{PE}) and the vector $(\overrightarrow{\boldsymbol{e'e}})$ verify:

$$\alpha_{PE} = \left|\overrightarrow{\boldsymbol{Pe}}\right| \cdot \left(\left|\overrightarrow{\boldsymbol{Pe}}\right| + \left|\overrightarrow{\boldsymbol{eE}}v\right|\right)^{-1} \tag{9}$$

$$\overrightarrow{\boldsymbol{e'e}} = \alpha_{PE} \cdot \overrightarrow{\boldsymbol{Ee}} + (1 - \alpha_{PE}) \cdot \overrightarrow{\boldsymbol{Pe}} \tag{10}$$

The last term in equation (8) compensates the error introduced in non-orthogonal grids due to the displacement of point e' away from the baricenter e. It depends on the precomputed gradient $\nabla \boldsymbol{\phi}$.

The gradient of field variables (u, v, w, p, ϕ) is also used explicitly in the discretization of diffusive terms in all conservation equations, and for the pressure term in the momentum equation. For a scalar field ϕ, the gradient at the cell center P is obtained using the Gauss theorem. For each component of the gradient the following expressions hold:

$$(\nabla \boldsymbol{\phi})_P \cdot \hat{\boldsymbol{e}}_i \approx \frac{\int_\Omega \nabla \cdot \left(\phi \hat{\boldsymbol{e}}_i\right) d\Omega}{\Delta\Omega} = \frac{\int_S \phi \hat{\boldsymbol{e}}_i \cdot \hat{\boldsymbol{n}}_S dS}{\Delta\Omega} \approx \frac{\sum_c \phi_c S_c^i}{\Delta\Omega} \tag{11}$$

4.2 Improved linear interpolation

The CDS scheme only partially accounts for the non-orthogonality in the grid through the explicit correction incorporated in (8). This could lead to significant loss of accuracy on non-orthogonal grids. Improved linear interpolation schemes based on multi-dimensional Taylor series expansions have been proposed to alleviate this problem [7, 9]. Following this approach the FTSE interpolation scheme is derived next, being actually a variation of that proposed in [9].

The CDS scheme was presented for the computation of value ϕ_e of a scalar field ϕ at the baricenter of the *east* face, regarding a sketch similar to Fig. 4a. The FTSE scheme for the interpolation of value ϕ_e involves also neighbour nodes S, N, B and T, as shown in Fig. 4b.

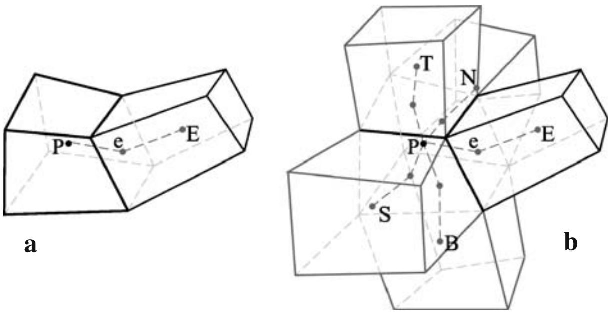


Fig. 4 **a** Volume element P with neighbour E, for CDS interpolation. **b** Volume element P with neighbours E, S, N, B, T, for improved linear interpolation

The starting point is the Taylor series expansion (12) for the field ϕ around point e, evaluated at a generic point A. With the notations of equations (13a–13c) the compact expression (14) replaces (12).

$$\phi_A = \phi_e + (x_A - x_e)\left(\frac{\partial \phi}{\partial x}\right)_e + (y_A - y_e)\left(\frac{\partial \phi}{\partial y}\right)_e + (z_A - z_e)\left(\frac{\partial \phi}{\partial z}\right)_e \tag{12}$$

$$x_{Ae} = (x_A - x_e) \qquad \phi_{Ae} = (\phi_A - \phi_e) \tag{13a}$$

$$\phi_e^x = \left(\frac{\partial \phi}{\partial x}\right)_e \qquad \nabla\boldsymbol{\phi}_e = \left(\phi_e^x, \phi_e^y, \phi_e^z\right) \tag{13b}$$

$$\overrightarrow{\boldsymbol{Ae}} = (x_{Ae}, y_{Ae}, z_{Ae}) \tag{13c}$$

$$\phi_{Ae} = x_{Ae}\phi_e^x + y_{Ae}\phi_e^y + z_{Ae}\phi_e^z = \overrightarrow{\boldsymbol{Ae}} \cdot \nabla\boldsymbol{\phi}_e \tag{14}$$

The geometric values in (14) are known, as well as the values of ϕ at the nodes P, E, S, N, B and T. Thus a system of linear equations, to determine the unknown values of the gradient $\nabla\boldsymbol{\phi}_e$, can be obtained by combining evaluations of (14) at these six nodes, as shown next.

Evaluating (14) at each node gives equations for the increments of ϕ, as (15a, 15b) for ϕ_{Pe} and ϕ_{Ee}. Similar expressions hold for ϕ_{Se}, ϕ_{Ne}, ϕ_{Be} and ϕ_{Te}. The unknown ϕ_e is eliminated combining these equations in pairs, as for example subtracting (15b) from (15a) to obtain (16).

$$\phi_{Pe} = \overrightarrow{\boldsymbol{Pe}} \cdot \nabla\boldsymbol{\phi}_e \tag{15a}$$

$$\phi_{Ee} = \overrightarrow{\boldsymbol{Ee}} \cdot \nabla\boldsymbol{\phi}_e \tag{15b}$$

$$\phi_{PE} = \phi_{Pe} - \phi_{Ee} = \overrightarrow{\boldsymbol{Pe}} \cdot \nabla\boldsymbol{\phi}_e - \overrightarrow{\boldsymbol{Ee}} \cdot \nabla\boldsymbol{\phi}_e = \overrightarrow{\boldsymbol{PE}} \cdot \nabla\boldsymbol{\phi}_e \tag{16}$$

Equations for ϕ_{SN} and ϕ_{BT} complete the linear system of algebraic equations (17), where the unknowns are the components of the gradient $\nabla\boldsymbol{\phi}_e$ and can be solved explicitly The solution of (17) is conveniently expressed in terms of the geometric coefficient θ given in (18). The compact expression (19) for θ results using the mixed product.

$$\begin{bmatrix} \overrightarrow{\boldsymbol{PE}} \\ \overrightarrow{\boldsymbol{SN}} \\ \overrightarrow{\boldsymbol{BT}} \end{bmatrix} \cdot \nabla\boldsymbol{\phi}_e = \begin{bmatrix} \phi_{PE} \\ \phi_{SN} \\ \phi_{BT} \end{bmatrix} \tag{17}$$

$$\begin{aligned} \theta = {} & x_{PE}y_{SN}z_{BT} + x_{SN}y_{BT}z_{PE} + x_{BT}y_{PE}z_{SN} \\ & - (x_{PE}y_{BT}z_{SN} + x_{SN}y_{PE}z_{BT} + x_{BT}y_{SN}z_{PE}) \end{aligned} \tag{18}$$

$$\theta = \overrightarrow{\boldsymbol{PE}} \cdot \left(\overrightarrow{\boldsymbol{SN}} \times \overrightarrow{\boldsymbol{BT}}\right) = \left\langle \overrightarrow{\boldsymbol{PE}}, \overrightarrow{\boldsymbol{SN}}, \overrightarrow{\boldsymbol{BT}} \right\rangle \tag{19}$$

From the solution of (17), equation (20) results for ϕ_e^x with similar expressions holding for ϕ_e^y and ϕ_e^z. In these expressions, the gradient components depend solely on geometrical values and the known values of ϕ at the nodes P, E, S, N, B and T.

$$\phi_e^x \cdot \theta = \phi_{PE}\left(y_{SN} z_{BT} - y_{BT} z_{SN}\right) \\ + \phi_{SN}\left(y_{BT} z_{PE} - y_{PE} z_{BT}\right) + \phi_{BT}\left(y_{PE} z_{SN} - y_{SN} z_{PE}\right) \tag{20}$$

An expression for ϕ_e at the baricenter of the *east* face, can be given now that the gradient components are known. Expression (21) is obtained directly from (15a). Since the gradient components depend linearly on the increments ϕ_{PE}, ϕ_{SN} and ϕ_{BT} between nodes, equation (21) can be arranged into the form of (22) where interpolation coefficients γ_E, γ_N and γ_T depend solely on the grid geometry.

$$\phi_e = \phi_P - x_{Pe}\phi_e^x - y_{Pe}\phi_e^y - z_{Pe}\phi_e^z = \phi_P - \overrightarrow{\boldsymbol{Pe}} \cdot \nabla \boldsymbol{\phi}_e \tag{21}$$

$$\phi_e = \phi_P + \gamma_E \cdot \phi_{EP} + \gamma_N \cdot \phi_{NS} + \gamma_T \cdot \phi_{TB} \tag{22}$$

Note that, for convenience, in (22) the increments of ϕ between nodes have been defined with opposite sign with respect to equations (20). Substituting equations (20) into (21) explicit expressions for the interpolation coefficients can be obtained, as in (23) for γ_E.

$$\gamma_E \cdot \theta = x_{Pe}\left(y_{SN} z_{BT} - y_{BT} z_{SN}\right) \\ + y_{Pe}\left(z_{SN} x_{BT} - z_{BT} x_{SN}\right) + z_{Pe}\left(x_{SN} y_{BT} - x_{BT} y_{SN}\right) \tag{23}$$

Similar expressions result for γ_N and γ_T. Again compact forms can be produced using the mixed product. Final expressions for FTSE interpolation coefficients are given in the set of equations (24), for γ_E, γ_N and γ_T. This set is complemented by equations (22) for the interpolated value ϕ_e and (19) for the denominator θ.

$$\gamma_E = \left\langle \overrightarrow{\boldsymbol{Pe}}, \overrightarrow{\boldsymbol{SN}}, \overrightarrow{\boldsymbol{BT}} \right\rangle \cdot \theta^{-1} \tag{24a}$$

$$\gamma_N = \left\langle \overrightarrow{\boldsymbol{PE}}, \overrightarrow{\boldsymbol{Pe}}, \overrightarrow{\boldsymbol{BT}} \right\rangle \cdot \theta^{-1} \tag{24b}$$

$$\gamma_T = \left\langle \overrightarrow{\boldsymbol{PE}}, \overrightarrow{\boldsymbol{SN}}, \overrightarrow{\boldsymbol{Pe}} \right\rangle \cdot \theta^{-1} \tag{24c}$$

5 Discretization of Equations

5.1 Momentum equation

The goal in this section is to obtain discretized and linearized expressions for the momentum equation projected onto the Cartesian coordinate directions. Each projection will be associated with one velocity component from $\vec{\boldsymbol{v}} = (u, v, w)$. This procedure will be explained for the first component u. For each velocity component, a system of linear equations will be obtained, with one equation for each volume element in the grid. The general form of this system will be:

$$\boldsymbol{A}^u \boldsymbol{u} = \boldsymbol{Q}^u \tag{25}$$

where $\boldsymbol{A}^u$ is an $N \times N$ coefficient matrix, $\boldsymbol{u}$ is the vector of unknowns and $\boldsymbol{Q}^u$ the independent vector term, being N the total number of grid elements. The coefficient matrix $\boldsymbol{A}^u$ is heptadiagonal, corresponding to the seven point computation molecule of Fig. 1, within each structured grid block. For a single generic element P we have:

$$A_P^u \cdot u_P + A_W^u \cdot u_W + A_E^u \cdot u_E + A_S^u \cdot u_S + A_N^u \cdot u_N + A_B^u \cdot u_B + A_T^u \cdot u_T = Q_P^u \tag{26}$$

where the coefficients A_i^u and the source term Q_P^u result from the discretization and linearization of the different terms in equation (2).

Equation (27) presents the approximation of convective u flux terms for a given volume element, through the *east* face (see Fig. 3).

$$\begin{aligned} F_e^{cu} &= \int_{S_e} \rho u \left(\vec{\boldsymbol{v}} \cdot \hat{\boldsymbol{n}}_S\right) dS \approx \dot{m}_e u_e = max\left(\dot{m}_e, 0\right) u_P + min\left(\dot{m}_e, 0\right) u_E \\ &\quad + \gamma_{CDS}\Big(\dot{m}_e u_e - max\left(\dot{m}_e, 0\right) u_P - min\left(\dot{m}_e, 0\right) u_E\Big) \\ &= A_P^{cue} \cdot u_P + A_E^{cue} \cdot u_E - Q_P^{cue} \end{aligned} \tag{27}$$

The mass flux through the *east* face ($\dot{m}_e$) participates explicitly in equation (27), allowing the linearization of the convective flux. The computation of mass fluxes will be considered in the next section when the mass balance equation is analysed. The approximation of the convective flux consists of an implicit, first order, *upwind* term and an explicit deferred correction [2, 10], which is of second order since the linearly interpolated velocity component u_e at the face center e is used. The interpolation scheme might be a standard CDS scheme or an improved FTSE scheme. The implicit term in equation (27) contributes to the coefficient matrix $\boldsymbol{A}^u$, while the explicit correction adds to the source term $\boldsymbol{Q}^u$. The use of an explicit deferred correction allows the inclusion of higher order approximations [6, 8], while preserving the simplicity and robustness of the implicit method.

The approximation of diffusive terms is considered next in equation (28), again for first Cartesian direction and through the *east* face:

$$\begin{aligned} F_e^{cu} &= \int_{S_e} \left(2\mu \boldsymbol{D} \cdot \hat{\boldsymbol{n}}_S\right) \cdot \hat{\boldsymbol{e}}_1 dS = \int_{S_e} \mu \left(\nabla u\right) \cdot \hat{\boldsymbol{n}}_S dS + \int_{S_e} \mu \frac{\partial \vec{\boldsymbol{v}}}{\partial x} \cdot \hat{\boldsymbol{n}}_S dS \\ &\approx \mu_e \left(u_E - u_P\right) \frac{\left|\vec{\boldsymbol{S}}_e\right|}{\left|\overrightarrow{\boldsymbol{PE}}\right|} + \Big(2\mu_e \boldsymbol{D}_e \cdot \vec{\boldsymbol{S}}_e\Big) \cdot \hat{\boldsymbol{e}}_1 - \mu_e \left(\nabla \boldsymbol{u}\right)_e \cdot \overrightarrow{\boldsymbol{PE}} \frac{\left|\vec{\boldsymbol{S}}_e\right|}{\left|\overrightarrow{\boldsymbol{PE}}\right|} \\ &= A_P^{due} \cdot u_P + A_E^{due} \cdot u_E - Q_P^{due} \end{aligned} \tag{28}$$

The approximation is split into an implicit first order term and an explicit deferred correction of higher order. The implicit term considers only the contribution of the given velocity component (u) to approximate the diffusive flux. The explicit correction takes in account the full expression of the interpolated symmetric tensor $\boldsymbol{D}_e$. Similar flux expressions hold for the other volume element faces w, n, s, t, b. The pressure term in equation (2) is approximated, using Gauss theorem, by the pressure gradient computed as in equation (11):

$$\int_S -p\hat{\boldsymbol{n}}_S \cdot \hat{\boldsymbol{e}}_1 dS = \int_\Omega -\nabla \cdot \left(p\hat{\boldsymbol{e}}_1\right) d\Omega \approx -\left(\frac{\partial p}{\partial x}\right)_P \Delta\Omega = Q_P^{pu} \tag{29}$$

Equation (29) contributes only to the source term $\boldsymbol{Q}^u$. The pressure itself is updated through the mass balance, as explained in Section 5.2

The remaining terms in equation (2) are given by volume integrals and their approximation is simple, through cell centered discretizations:

$$\int_\Omega \rho \frac{\partial u}{\partial t} d\Omega \approx \frac{\rho}{\Delta t}\left(u_P - u_P^{old}\right)\Delta\Omega = A_P^{tu} \cdot u_P - Q_P^{tu} \tag{30}$$

$$\int_\Omega \rho\beta\left(T - T_{ref}\right)\vec{\boldsymbol{g}} \cdot \hat{\boldsymbol{e}}_1 d\Omega \approx \rho\beta\left(T_P - T_{ref}\right)\vec{\boldsymbol{g}} \cdot \hat{\boldsymbol{e}}_1 \Delta\Omega = Q_P^{gu} \tag{31}$$

In equation (30) a first order backwards approximation was considered for the time derivative, while higher order terms can be easily considered, introducing multiple time step schemes.

5.2 Mass balance equation

The velocity field obtained from the discretized approximations to the momentum balance equation (2) is not subject to the incompressibility condition and thus it must be corrected to fulfil the mass balance equation (1). An equation to update the pressure field is also needed. With these purposes the SIMPLE method for pressure-velocity coupling [2] is used in this work. The current estimates for the velocity and pressure fields $(\vec{\boldsymbol{v}}^\star, p^\star)$ will be modified by adding velocity and pressure corrections $(\vec{\boldsymbol{v}}', p')$ to obtain new current estimations $(\vec{\boldsymbol{v}}, p)$:

$$\vec{\boldsymbol{v}} = \vec{\boldsymbol{v}}^\star + \vec{\boldsymbol{v}}' \qquad p = p^\star + p' \tag{32}$$

The correction fields $(\vec{\boldsymbol{v}}', p')$ are determined so that the new velocity field $(\vec{\boldsymbol{v}})$ verifies the mass balance. The relation between the velocity and pressure corrections is inspired in equations (26) and (29):

$$\vec{\boldsymbol{v}}_e' = -\left(\frac{\Delta\Omega}{A_P}\right)_e \left(\nabla \boldsymbol{p}'\right)_e \tag{33}$$

Equation (33) reflects that, in an incompressible fluid, the pressure field acts as a *link* enforcing the incompressibility condition onto the velocity field. Equation (33) neglects the influence of other terms from the momentum balance in equation (26) and thus the resulting new estimations $(\vec{\boldsymbol{v}}, p)$ will not fulfil exactly the momentum balance.

The mass flux through, for example, the *east* face of a given volume element centered in P can be approximated by the interpolated quantities at the face center e:

$$\dot{m}_e = \int_{S_e} \rho\left(\vec{\boldsymbol{v}}\cdot\hat{\boldsymbol{n}}_s\right) dS \approx \rho_e\left(\vec{\boldsymbol{v}}_e\cdot\vec{\boldsymbol{S}}_e\right) \tag{34}$$

Using equation (33) expressions for the corrected mass fluxes at each face of a volume element can be obtained. For the *east* face:

$$\dot{m}_e = \dot{m}_e^\star + \dot{m}_e' = \rho_e\left(\vec{\boldsymbol{v}}_e^\star\cdot\vec{\boldsymbol{S}}_e\right) - \left(\rho\frac{\Delta\Omega}{A_P}\right)_e \vec{\boldsymbol{S}}_e\cdot\left((\nabla\boldsymbol{p})_e - \overline{(\nabla\boldsymbol{p})_e}\right) - \left(\rho\frac{\Delta\Omega}{A_P}\right)_e \vec{\boldsymbol{S}}_e\cdot\left(\nabla\boldsymbol{p}'\right)_e \tag{35}$$

The first term in the right hand side of equation (35) corresponds to the uncorrected mass flux $\dot{m}_e^\star$, computed as in (34) using the current estimation of the velocity field ($\vec{\boldsymbol{v}}^\star$) obtained from the momentum balance equation. The second term is a correction to prevent oscillations in the pressure field [10, 13], based on the difference between the pressure gradient computed at the face center by central differences and the interpolated pressure gradient . The third term in the right hand side of equation (35) represents the mass flux correction $\dot{m}_e'$, thus accounting for the contribution of pressure correction (p') onto the velocity, according to equation (33). This last term produces the equation for the pressure correction (p'), when the expressions for the corrected mass fluxes are substituted into the discretized mass balance equation with $\left(\nabla\boldsymbol{p}'\right)_e$ approximated by central differences.

$$\sum_c \dot{m}_c = \sum_c \dot{m}_c^\star + \sum_c \dot{m}_c' = 0 \tag{36}$$

In this way equation (37) for the pressure correction is obtained, with the same structure as equation (26). This procedure might be adapted to deal with compressible flows, by allowing corrections to the density field and relating these corrections to the pressure corrections through appropriate equations of state [2].

$$A_P^p\cdot p_P' + A_W^p\cdot p_W' + A_E^p\cdot p_E' + \ldots + A_T^p\cdot p_T' = Q_P^p \tag{37}$$

5.3 Iteration scheme

Within each time step equations (26) for each component of the velocity and equation (37) for the pressure correction are solved alternately and successively. In addition equations for scalar transport, similar to (26) and which will not be detailed here, are solved. Thus, the solution to the non-linear, coupled, partial differential equation system (1, 2 and 3) is approximated by the solution of a succession of linear equation systems, as represented in Fig. 5. The model *caffa3d.MB* currently implements, for the iterative solution of these linear systems, the methods SIP and

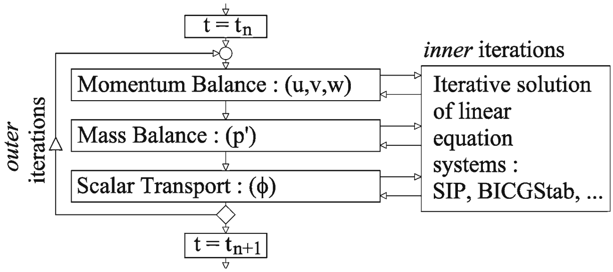

t = t_n
inner iterations
Momentum Balance : (u,v,w)
Mass Balance : (p')
Scalar Transport : (ϕ)
Iterative solution of linear equation systems : SIP, BICGStab, ...
outer iterations
t = t_n+1

Fig. 5 Iteration scheme for one time step (adapted from [12])

CGStab [2], and an AMG linear solver, that uses SIP as a smoother, presented below. Incorporating other methods for solving linear systems is straightforward.

6 Boundary Conditions and Block Interfaces

6.1 Boundary conditions

Boundary conditions are taken in account through a special treatment of flux terms in equations (1, 2 and 3), for the case of boundary volume elements, as those pictured if Fig. 6. The scheme for neighbours $P - E$ of Fig. 3 is replaced by the scheme $P - PB$ of Fig. 6a. The nodes PB are additional grid nodes placed at the center of boundary faces, as shown in Fig. 6b, to simplify the application of boundary conditions. Field values at these nodes do not participate as unknowns in the linearized systems of equations for inner nodes.

First, the contribution from inner cells and faces to the coefficients and source terms of equations (26) and (37) is considered following the structured arrangement within each grid block. Next, an indexed list to the boundary faces, as that of Fig. 6a, is looped through to consider the contribution of each boundary cell to these equations. The model currently implements inlet and outlet boundary conditions as well as symmetry surfaces and walls, both fixed or moving through sliding interfaces. Other boundary conditions can be incorporated easily.

6.2 Block interfaces

The consideration of *block* structured grids requires special attention to the volume elements placed at the interfaces between blocks. In Fig. 7a an example is presented, made of two grid blocks with a matching *one-to-one* interface. The scheme of Fig. 3 must be replaced for the case of block interface cells by the similar scheme of Fig. 7b.

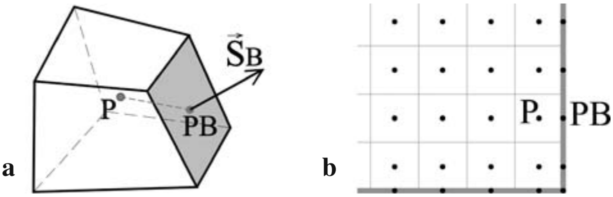


Fig. 6 **a** Boundary volume element. **b** Grid detail at the boundary

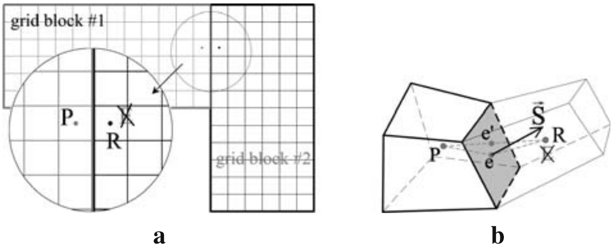


Fig. 7 **a** A 'one-to-one' block interface. **b** Volume element *P* at a block interface

The neighbour *R* belongs now to a different grid block. This is a non structured relation between *P* and *R*, so the information about block interface relations is stored in an indexed list.

Following the scheme of Fig. 7b, for a volume element *P* placed at the interface of the first block, equation (26) is replaced by equation (38), where the structured term $A^u_E \cdot u_E$ is replaced by the *unstructured* term $A^u_R \cdot u_R$, and the coefficient A^u_P is modified correspondingly:

$$A^u_P \cdot u_P + A^u_W \cdot u_W + \underbrace{A^u_R \cdot u_R}_{\text{interface}} + A^u_S \cdot u_S + \ldots + A^u_T \cdot u_T = Q^u_P \tag{38}$$

Analogously, equations for interface cells in the second block will incorporate non structured terms of the form $A^u_L \cdot u_L$ replacing the structured terms $A^u_W \cdot u_W$. The resulting coefficient matrix $\boldsymbol{A}^u$ is now block-heptadiagonal with non structured coefficients, corresponding to the interfaces between grid blocks, distributed in the off diagonal blocks of the matrix, as shown in Fig. 8. These off diagonal coefficients are taken in account to update the residual vector during the linear system iterations, although not to build the iteration matrix. Still the implicit character of the method is conserved [10]. The volume elements at the interfaces between grid blocks are not boundary nodes, but inner volume elements with special non structured information about their neighbour elements. They receive an essentially implicit treatment.

The case of *many-to-one* block interfaces, as the one shown in Fig. 9, can be treated analogously to that of the matching *one-to-one* block interface seen above. The relation between volume elements is *injective* if seen from the nodes *P* of the finer block, so that the construction of an indexed list and the consideration of

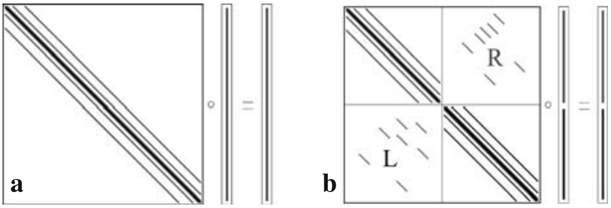


Fig. 8 Structure of linearized equation system $\boldsymbol{A}^u\boldsymbol{u} = \boldsymbol{Q}^u$. **a** For single block grids matrix is heptadiagonal. **b** In block structured grids the coefficient matrix becomes block heptadiagonal, with a *reduced* amount of non structured coefficients in off diagonal blocks corresponding to the interfaces between blocks

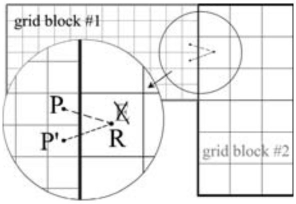


Fig. 9 Sketch of *many-to-one* block interface. The indexed neighbour R inherits contributions from more than one ‘P’ node (two in this case: *P* and *P’*)

additional coefficients in the linearized system of equations is done in the same way as before. In fact the implementation of non-matching *many-to-many* block interfaces, mostly differs in the increased complexity of building the indexed list of relationships through the interface and computing the geometrical properties involved. Once this is done, the process of building the coefficient matrices, and the iterative solution of the resulting linear systems scarcely require modifications. The case of *sliding* interfaces that will be considered next is a particular case of *many-to-many* non matching interface, where the referred process can be simplified.

6.3 Sliding block interfaces

The implementation of sliding block interfaces allows to incorporate moving rigid bodies within the domain, whose movement might be prescribed or can result from their dynamic interaction with the fluid. The precondition for this approach is that the interfaces between sliding blocks must keep their shape during the movement. This is appropriate for rotating bodies rather than freely moving ones. On the other hand the extra computational effort required is minimal, with no need to recompute the grid at each step, since only some connectivity information needs to be updated.

In Fig. 10 an example of sliding interface between two concentric grid blocks is given. The interior block is fixed, while the exterior block rotates. The distribution of cells in each block matches the other for the case of null relative displacement angle α. For other values of α, any given cell at the interface relates to at most two other cells in the opposite grid block. The indexed list of relationships between cells in the interface and the influence fractions between cells can be computed explicitly, at any time step, from the value of the angle α. The geometrical properties of the cells in the rotating block can also be updated applying a rotation through simple trigonometric relations.

To compute coefficients in the discretized of equations (26, 37) the relative movements of grid blocks must be considered. The local time derivative of the

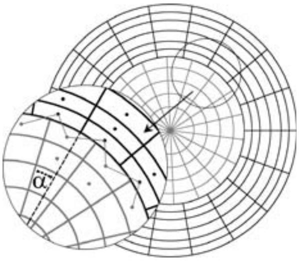


Fig. 10 Sketch of sliding interface between two concentric blocks, with detail of grid and relative displacement angle α

absolute velocity field is computed, for each moving block, in fixed points of the rotating system and the relative velocities are used for the mass fluxes. With these conditions equations (39) and (2) are equivalent:

$$\int_{\Omega} \rho \frac{\partial_R \vec{\boldsymbol{v}}_A}{\partial t} d\Omega + \int_S \rho \vec{\boldsymbol{v}}_A \left(\vec{\boldsymbol{v}}_R \cdot \hat{\boldsymbol{n}}_S\right) dS = \int_{\Omega} \rho\beta \left(T - T_{ref}\right) \vec{\boldsymbol{g}} \cdot \hat{\boldsymbol{e}}_1 d\Omega + \int_S -p\hat{\boldsymbol{n}}_S \cdot \hat{\boldsymbol{e}}_1 dS + \int_S \left(2\mu \boldsymbol{D} \cdot \hat{\boldsymbol{n}}_S\right) \cdot \hat{\boldsymbol{e}}_1 dS \quad (39)$$

where the subindex ($_A$) refers to the inertial reference system and the subindex ($_R$) refers to the reference system that rotates with the grid block. The relative movement between blocks is that of a rigid body. It is thus incompressible which has been implicitly considered in (39). Equation (39) is *not* the expression for the momentum balance equation in a non-inertial reference frame, since it still considers the absolute momentum ($\rho\vec{\boldsymbol{v}}_A$). It is rather the transformation of (2) under the conditions mentioned before.

The iterative solution of the resulting linearized systems require no other modifications than the consideration of additional non-structured coefficients for the computation of residual vectors, as in the case of *one-to-one* block interfaces. If the movement of the rigid parts is not prescribed, additional dynamic equations for the involved rigid bodies must be introduced to describe their movement. An example considering a simple mixer will be presented in Section 9.3

7 Block Structured AMG Solver

It is well known that the performance of usual linear solvers deteriorates as the number of nodes increases, since they only smooth efficiently the high spatial frequency components of the error, while the lower spatial frequency components are decreased much slower. Different multigrid strategies have been proposed to deal with this problem [2, 6, 11]. The common principle is to smooth the lower spatial frequency components of the error through a succession of coarsened grids [9]. The iterations in the coarser grids are inexpensive since, in three dimensional cases, the number of points is decreased by eight from one grid level to the next, while the number of iterations required in the finest grid is, for the multigrid approach, independent of the number of points. The AMG solver presented next is usually only applied to the pressure equation, since it is usually the hardest to converge for the linear solver, while the iterations for the momentum equations converge much rapidly. In this paper the AMG approach described in [11] was followed, adapting it to work in the block structured grid context presented before. The AMG strategy works directly with the linear system produced at the original grid. Instead of explicitly defining a set of coarser grids and building the corresponding linear system at each grid, a set of increasingly coarser linear systems is derived directly from the original linear system applying a blocking procedure through the use of transfer operators [11]. Not needing to explicitly define and compute the properties of a series of coarser grids is an advantage when dealing with non-orthogonal, body fitted grids, as is the case here. Still, coarser levels $l = 2, 3, ..., l_{max}$ might be regarded

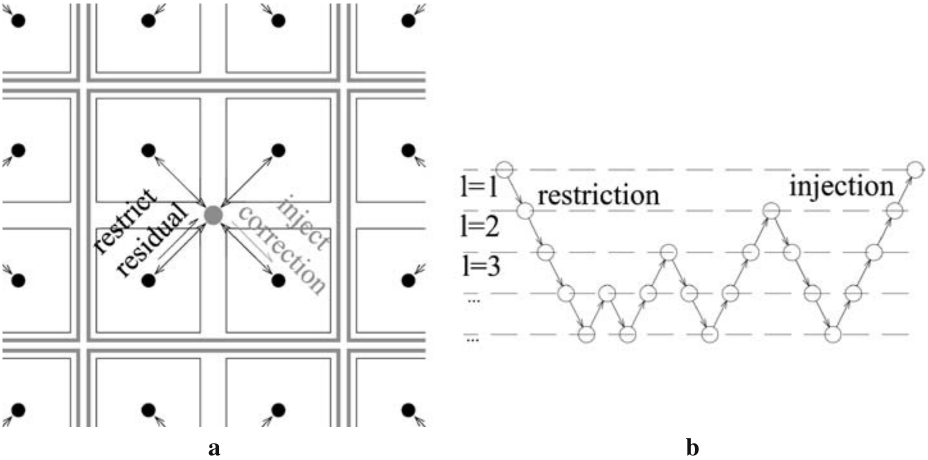


Fig. 11 **a** Relation between grid levels. Fine grid in *black* and coarse grid in *gray*. Restriction operator averages fine grid residuals into coarse grid, while injection operator copies coarse grid corrections into fine grid. **b** W-cycle in AMG method

as related to coarser grids obtained by grouping volume elements from the previous level in blocks. Figure 11a illustrates this for a schematic grid.

A linear system is defined at each level, with the first, finest, level corresponding to original grid. Coarser levels are constructed to damp the error at increasingly lower spatial frequencies:

$$A_1 x_1 = b_1\,, \quad A_2 x_2 = b_2\,, \quad \ldots\,, \quad A_l x_l = b_l\,, \quad \ldots \tag{40}$$

The coefficient matrices at each coarse level $l+1$ are derived from the previous finer level l by grouping the coefficients through a transfer operator. For the nine point scheme in three dimensions, a suitable transfer operation is the following:

$$\begin{aligned} A_{P(i,j,k)_{l+1}} = {} & \sum_{m=2i}^{2i+1}\sum_{n=2j}^{2j+1}\sum_{p=2k}^{2k+1} A_{P(m,n,p)_l} + \sum_{n=2j}^{2j+1}\sum_{p=2k}^{2k+1} A_{E(2i,n,p)_l} + A_{W(2i+1,n,p)_l} \\ & + \sum_{m=2i}^{2i+1}\sum_{p=2k}^{2k+1} A_{N(m,2j,p)_l} + A_{S(m,2j+1,p)_l} \\ & + \sum_{m=2i}^{2i+1}\sum_{n=2j}^{2j+1} A_{T(m,n,2k)_l} + A_{B(m,n,2k+1)_l} \end{aligned} \tag{41a}$$

$$A_{E(i,j,k)_{l+1}} = \sum_{n=2j}^{2j+1}\sum_{p=2k}^{2k+1} A_{E(2i+1,n,p)_l} \tag{41b}$$

with expressions similar to (41b) holding for A_W, A_N, $\ldots$, A_B.

The residual from the solution of the linear system at level l provides the independent term for the next coarser level $l+1$. The restriction operator selected simply adds up the residual at each $2 \times 2 \times 2$ block of cells in level l to obtain the independent term in level $l+1$.

$$r_l = b_l - A_l x_l \tag{42a}$$

$$r_l \rightarrow \underline{restriction} \rightarrow b_{l+1} \tag{42b}$$

For any location (i, j, k) in the coarser level $l+1$, the result of the restriction operation on the residual at the next finer level l is:

$$b_{(i,j,k)_{l+1}} = \sum_{m=2i}^{2i+1} \sum_{n=2j}^{2j+1} \sum_{p=2k}^{2k+1} r_{(m,n,p)_l} \tag{43}$$

At each level the *smoothing* of the solution is performed by iterating the linear system with an appropriate solver. Here the SIP solver, adapted to work with block structured grids as described in [10] and to deal with the multilevel indexing associated with AMG, has been incorporated to work as the smoother. The solution obtained at a coarser level $l+1$ is used to obtain a correction c_l for the next finer level l through an injection operation, which simply copies the coarse grid values to each corresponding block of $2 \times 2 \times 2$ cells in the fine grid. If x_l^{old} is the existing solution at level l, and x_{l+1} is the obtained solution at level $l+1$, then the new approximation x_l at level l is obtained by:

$$x_{l+1} \rightarrow \underline{injection} \rightarrow c_l \tag{44}$$

$$c_{(m,n,p)_l} = x_{(2i,2j,2k)_{l+1}}\,, \quad m=i,i+1 \;\; n=j,j+1 \;\; p=k,k+1 \tag{45}$$

$$x_l = x_l^{old} + c_l \tag{46}$$

At the interfaces between grid blocks, the interaction coefficients A_R and A_L, as for example in equation (38), are grouped through the same transfer operator used for the $A_E, A_W, \ldots, A_B$ coefficients. Thus, equations similar to (41b) apply. Iterations are started at the finest grid and proceed down to the coarsest level. The return path is done in successive V-cycles as shown in Fig. 11b, to progressively accumulate and smooth the corrections at each level.

8 Parallelization Using OpenMP

The OpenMP programming model allows parallel computation in shared memory computers (SMP) through compilation directives. Although it lacks the flexibility of the message passing interface (MPI) framework, since it cannot be applied directly to distributed memory systems, it has advantages in terms of simplicity and performance in SMP systems. The strategy of domain decomposition can be easily adapted to

OpenMP, specially in the case of block structured grids. The data structure designed to support block structured grids gives a natural framework to distribute the work among the different processors.

8.1 Domain decomposition

The domain decomposition strategy uses the same data structure of the block structured grid approach. For every task, each processor is assigned with one or more grid blocks. Only tasks associated with the interfaces between blocks are done by a single processor, since they potentially involve situations of interfering memory access.

The main advantages of this approach are the programming simplicity, the ability to control the domain decomposition through the design of the block structured grid, and the high efficiency attained when a good load balance is achieved. As opposed to MPI there is no effort wasted in sharing information between processors, since they all share the same memory. A significant drawback however is the difficulty to attain a good load balance in some cases, as for example a grid composed of only two blocks of very different size. A workaround is to arbitrary divide the bigger block into two conveniently sized blocks.

The introduction of OpenMP directives does not require the modification of the serial program structure. This ensures that the results of the computations will be the same, except for roundoff errors due to the different ordering when doing sums over the whole domain, for example for the computation of residuals.

8.2 Preliminary analysis of speed up

To evaluate the speed up factor obtained with OpenMP, a two processor SUN Fire V20z Server was used. The system runs AMD Opteron 248 processors, with 1 MB cache and 2 GB of RAM. The test case was the lid-driven flow in a cubical cavity at Re = 1000, which is analysed in Section 9.1. The grid was set up with two blocks, with a total of 61^3 nodes. The speed up factor was of 1.8× with respect to the serial execution.

The speed up factor is strongly dependent on the architecture of the multiprocessor system and on the choice of compiler. Similar results to those reported above (1.75×) were obtained in a two processor DELL system with two Intel Xeon processors (512KB cache and 1 GB RAM). However, using a generic two processor system with two AMD Atholon MP processors (256 KB cache and 1 GB RAM) the speed up factor was only 1.4×. The limiting factor was detected to be the memory access, due to a poor performance of the data bus, since two independent serial processes running simultaneously suffered a similar performance loss.

9 Application Examples and Results

9.1 Cubical lid-driven cavity at Re = 1000

The first application considered is that of the lid-driven flow inside a cubical cavity at Re = 1000. This is a traditional test case for Navier-Stokes solvers, with good

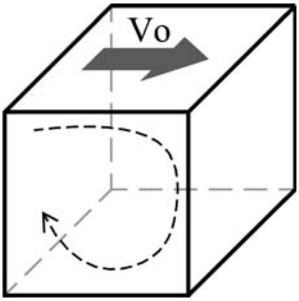


Fig. 12 Sketch of lid-driven cavity. Re = Vo · h/ν = 1000

numerical solutions being available in the bibliography [1, 3, 5] to validate the obtained results. The recent results from [1] will be used here for validation. Figure 12 presents a sketch of the domain. Non slip boundary conditions are applied to all walls, with the top one sliding at constant velocity Vo. Given the height h of the cavity and the lid velocity Vo, the fluid viscosity ν is set so that Re = 1000. The computations were run until steady state was achieved.

Simulations were run for two series of systematically refined grids, one with uniform distribution of grid points and the other with a finer resolution near the walls. The number of points on each side of the cavity was $N = 21, 41, 61, 81, 101$, and 121 respectively. A reference solution was computed applying Richardson extrapolation to the solutions obtained from the finer stretched grids. Figure 13 presents normalized velocity profiles at horizontal and vertical centerlines for uniform grids with $N = 41$ and 81, and stretched grid with $N = 121$. Comparison is provided against the results from spectral computations in [1]. In Table 1 the minimum and maximum

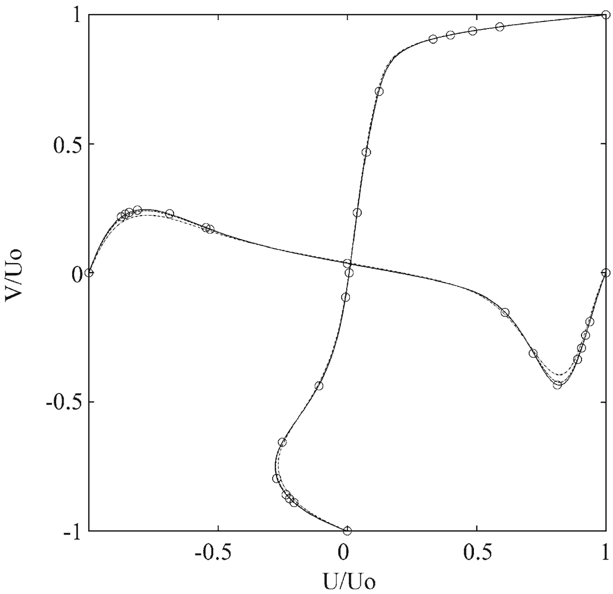


Fig. 13 Horizontal and vertical velocity profiles at centerlines. Results from uniform grids with $N = 41$ (*dash-dotted line*) and $N = 81$ (*dashed line*), and stretched grid with $N = 121$ (*solid line*), compared with results from [1] (symbols: o)

Table 1 Minimum and maximum velocity components on vertical and horizontal centerlines, for different grid resolutions

$N_x \times N_y \times N_x$	$max_y(v)$	y	$min_y(v)$	y	$min_x(v)$	x
41 × 41 × 41	0.2240	0.1134	−0.3949	0.9101	−0.2530	0.1317
81 × 81 × 81	0.2404	0.1110	−0.4241	0.9106	−0.2727	0.1267
121 × 121 × 121	0.2438	0.1101	−0.4301	0.9101	−0.2769	0.1253
41 × 41 × 41[a]	0.2384	0.1110	−0.4311	0.9116	−0.2710	0.1272
81 × 81 × 81[a]	0.2445	0.1096	−0.4340	0.9101	−0.2779	0.1249
121 × 121 × 121[a]	0.2457	0.1093	−0.4350	0.9098	−0.2793	0.1245
96 × 96 × 64[b]	0.2467	0.1091	−0.4350	0.9096	−0.2804	0.1242

Results given for uniform grids

[a] Stretched grids

[b] Spectral method computations from [1]

values for the velocity components along the centerlines are presented for selected grids, together with the results reported by [1]. Results for the solution on the $N = 121$ stretched grid are within 0.4% of those in [1]. Also both series of systematically refined grids show monotonic convergence towards these reference values.

The rate of error decay as the uniform and stretched grids are refined was computed. The error for the horizontal velocity component along the vertical centerline for both set of grids is shown in Fig. 14a. In both cases the error decay fits well the expected theoretical decay for a second order method, with the stretched grid performing better. To validate the FTSE interpolation, a series of block structured grids were configured. Three blocks were used, with an outer finer block near the wall interfacing with a coarser middle block and finally an orthogonal block at the

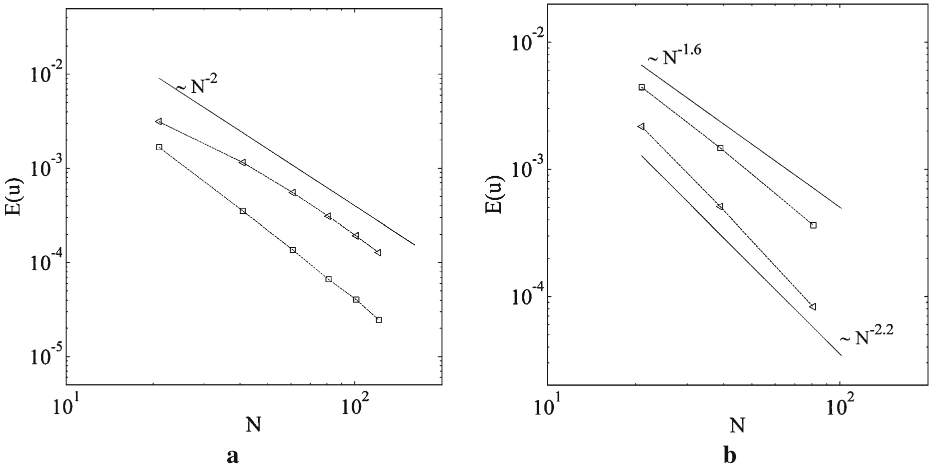


Fig. 14 **a** Error reduction with grid resolution for orthogonal grids. The error norm for horizontal velocity at vertical centerline is shown for uniform (◃) and stretched (□) grids respectively with $N = 21$, 41, 61, 81, 101 and 121. *Solid line* shows expected N^{-2} decay. **b** Error reduction with grid resolution in non-orthogonal grid, for CDS (□) and FTSE (◃). *Solid line* show fitting potential decays $N^{-1.6}$ and $N^{-2.2}$ respectively

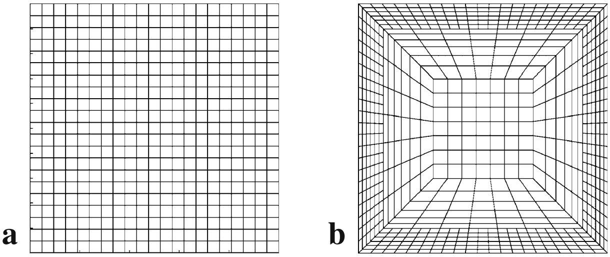


Fig. 15 **a** Uniform grid **b** Non-orthogonal O-type grid used to test the FTSE linear interpolation scheme. Pictured grids correspond to $N = 21$ resolution

center as pictured in Fig. 15b. The resulting composite grid is non orthogonal, and also exhibits sharp changes in grid line direction and sudden grid refinement across block interfaces, providing a challenge for standard CDS interpolation. Simulations in the O-type grids were run for different resolutions corresponding to $N = 21$, 41 and 81. In Fig. 14b error decay with increasing resolution is given for the O-type grids, using both the standard CDS and the improved FTSE interpolation schemes. The FTSE scheme preserves true second order error decay, while for CDS it reduces to 1.6.

9.2 Flow inside a PCB enclosure model

The simulation of the flow inside a printed circuit board enclosure is considered next. This study case motivated the development of the code and takes advantage of the block structured grid approach. Figure 16 presents a sketch of the domain (a), the block structured grid for this case (b) and an image of the experimental setup for the time resolved PIV measurements described in [14]. The Reynolds number at the entrance section was set to $Re = U.h/\nu = 1.2 \times 10^3$. Spatial resolution was set to $h/70$ near the walls and about $h/35$ at the core, using stretched grid blocks, for a total of 3.1×10^6 nodes. This configuration occupies 1 GB RAM and was run on the two processor systems mentioned in Section 8.2, under OpenMP. A general view of the flow is given in Fig. 17 through the visualization of λ_2 iso-surfaces [4]. The upper

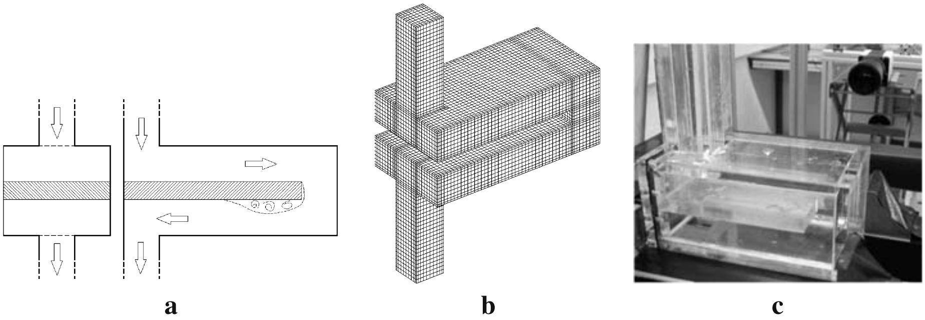


Fig. 16 **a** Sketch of domain. **b** Block structured grid. **c** Experimental setup

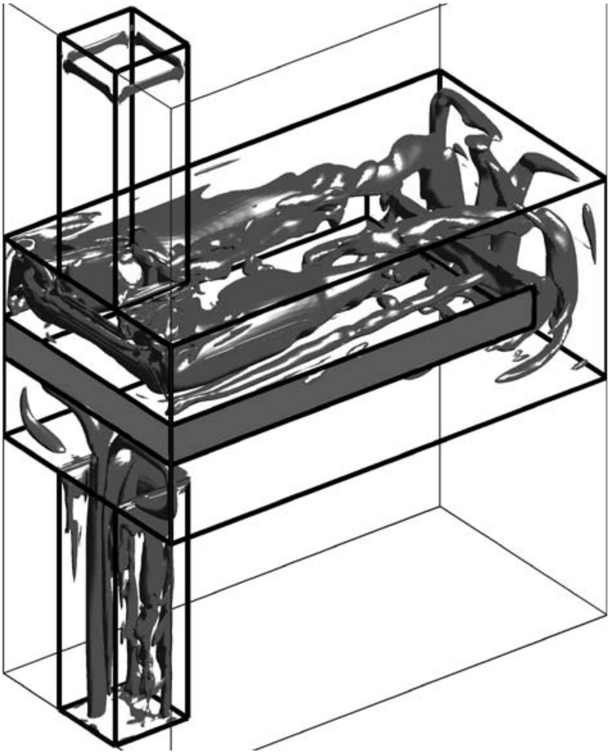


Fig. 17 Instantaneous vortex structures, identified by means of λ_2 iso-surfaces

channel is dominated by two large counter rotating vortical structures, which break down at the entrance to the lower channel.

Figures 18 and 19 present streamline patterns in the region indicated in Fig. 16a, at the entrance of the lower channel and beneath the plate, from PIV experiments and numerical simulations, respectively. In the experimental results a recirculation region is observed, centered at approximately ($y/h = 0.85, x/h = 4.35$) and with an horizontal extension of about h. For the numerical results the recirculation region is slightly displaced downstream, with center at ($y/h = 0.85, x/h = 4.15$), and has

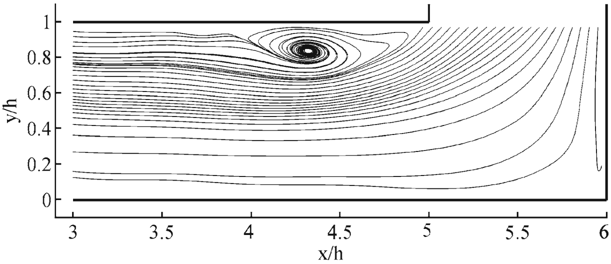


Fig. 18 Streamlines for the mean velocity field in the central vertical section at the entrance of the lower channel (see Fig. 16a), from time resolved PIV

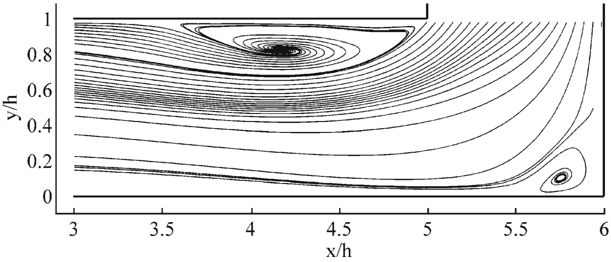


Fig. 19 Same as Fig. 18, but for numerical results

an horizontal extension of about 1.4*h*. Also a small recirculation region near the corner at $(y/h = 0.0, z/h = 6.0)$ barely appears in the experimental results, while it is clearly defined in the numerical results.

9.3 A simple mixer

The last application has been selected with the purpose of validating the sliding interface technique described in Section 6.3. The case is that of a simplified mixer with a single central rotating blade, as sketched in Fig. 20, inside a square container.

The body fitted block structured grid is shown in Fig. 21a). The grid contains five blocks. Two outer grid blocks (cyan and magenta) that are fixed and three inner grid blocks (red, blue and green) that rotate with the blade. The sliding interface exists between cyan and green blocks. The red block is an incomplete cylinder, while the blue block provides a simple representation of the blade geometry. The shape, size and number of blades could be easily modified by changing these two blocks or by decomposing them into several smaller blocks.

The fluid and blade are initially at rest, and a passive scalar profile is specified as shown in Fig. 21b in order to visualize the flow. The initial profile of this tracer has been slightly smeared in order to prevent oscillations due to a sharp edge. Non slip boundary conditions were applied to all vertical surfaces for the outer container and the blade, with their respective velocities. Symmetry boundary conditions hold on top and bottom surfaces, so the flow is essentially planar. The blade starts moving abruptly with angular velocity ω. The Reynolds number is $\mathrm{Re}=\omega L^2/\nu=500$, being

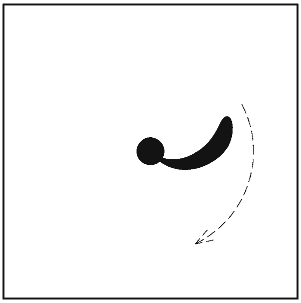


Fig. 20 Simplified mixer, with a single rotating central blade in a square container

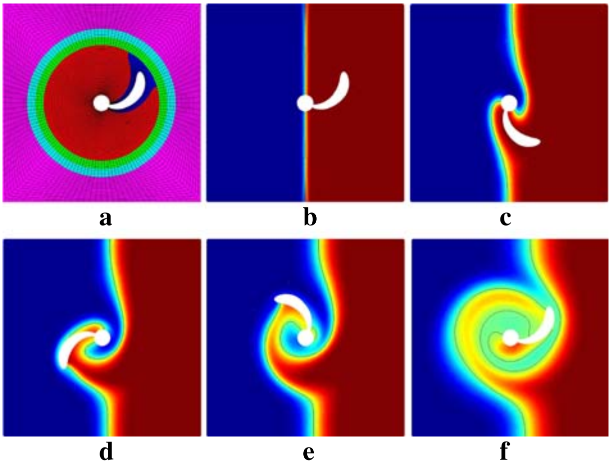


Fig. 21 **a** Block structured grid for the simplified mixer simulation. **b**–**f** Time evolution of passive scalar field (ϕ) through the first revolution of the blade and at the end of the second revolution (**f**). The *thin black line* is the iso-curve of $\phi = 0.5$

L the length of the blade. The remaining frames in Fig. 21 show the evolution during the first revolution of the blade (Fig. 21b–e) and after two revolutions (Fig. 21f). The iso-curve of $\phi = 0.5$ shows no singularities across the domain or block interfaces. There is no appreciable distortion due to the sliding interface.

Several simulations were run with different time step lengths to verify second order accuracy in time for a non stationary case with time dependent boundary conditions. Results for error decay with time step length are given in Fig. 22. The first two points correspond to time steps for which the blade rotates more than one grid block per time step. For smaller time steps truly second order decay is achieved.

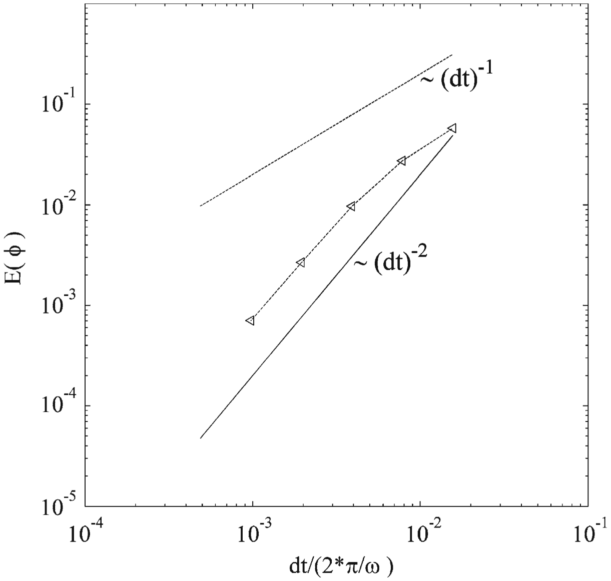


Fig. 22 Error reduction with time step. Error norm for scalar field ϕ is shown. *Solid line* shows expected theoretical dt^{-2} decay

10 Conclusions

An implementation of the finite volume method in block structured body fitted grids has been presented, for the simulation of three dimensional flows. The decomposition of the global grid into structured grid blocks allows to incorporate some advanced features. Qualitative as well as quantitative results for some applications were obtained that demonstrate these capabilities, as for example the incorporation of moving rigid bodies through sliding interfaces and the parallel computation through domain decomposition using OpenMP. Preliminary results show that the performance increase under OpenMP is dependant of specific hardware implementations, presumably those regarding the speed of memory access. Also, an AMG solver was incorporated and adapted to work under the block structured grid framework. An improved linear interpolation scheme for non-orthogonal grids, based on multidimensional Taylor series expansions, was shown to preserve the second order behaviour of the method, while the performance with a standard CDS interpolation in non-orthogonal grids deteriorates. Second order accuracy in time has also been verified. The numerical model presented here has been made freely available, to promote cooperative developments in the field. For turbulent flows, LES and κ-ϵ models are available, while a free surface module integrated to *caffa3d.MB* is currently being validated and will also be made available shortly.

Acknowledgements This work was supported by FPI/DPI2003-06725-C02-01 and DPI2006-02477 from DGI, Ministerio de Educación y Cultura y Fondos FEDER, Spain, and grants 33/07, 48/01 PDT and I+D CSIC, Uruguay. The authors are grateful to Dr. Milovan Peric for his authorization to use the 2D numerical model *caffa* as a basis to develop *caffa3d.MB*.

References

1. Albensoeder, S., Kuhlmann, H.C.: Accurate three-dimensional lid-driven cavity flow. J. Comput. Phys. **206**, 536–558 (2005)
2. Ferziger, J., Peric, M.: Computational Methods for Fluid Dynamics. Springer, Berlin (2002)
3. Iwatsu, R., Hyun, J.M., Kuwahara, K.: Analyses of three dimensional flow calculations in a driven cavity. Fluid Dyn. Res. **6**(2), 91–102 (1990)
4. Jeong, J., Hussain, F.: On the identification of a vortex. J. Fluid Mech. **285**, 69–94 (1995)
5. Jiménez, A.: Interaccio entre les estructures de flux i el transport de materia. Aplicacio a un doll pla turbulent. Ph.D. thesis, URV (2003)
6. Lange, C.F., Schäfer, M., Durst, F.: Local block refinement with a multigrid solver. Int. J. Numer. Methods Fluids **38**, 21–41 (2002)
7. Lehnhauser, T., Schäfer, M.: Improved linear interpolation practice for finite-volume schemes on complex grids. Int. J. Numer. Methods Fluids **38**, 625–645 (2002)
8. Lehnhauser, T., Schäfer, M.: Efficient discretization of pressure-correction equations on non-orthogonal grids. Int. J. Numer. Methods Fluids **42**, 211–231 (2003)
9. Lehnhauser, T., Ertem-Muller, S., Schafer, M., Janicka, J.: Advances in numerical methods for simulating turbulent flows. Prog. Comput. Fluid Dyn. **4**(3–5), 208–228 (2004)
10. Lilek, Z., Muzaferija, S., Peric, M., Seidl, V.: An implicit finite-volume method using nonmatching blocks of structured grid. Numer. Heat Transf. B **32**, 385–401 (1997)
11. Mora Acosta, J.: Numerical algorithms for three dimensional computational fluid dynamic problems. Ph.D. thesis, UPC (2001)
12. Peric, M.: Numerical methods for computing turbulent flows. Course notes (2001)

13. Rhie, C.M., Chow, W.L.: A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA J. **21**, 1525–1532 (1983)
14. Usera, G., Vernet, A., Ferré, J.A.: Use of time resolved PIV for validating LES/DNS of the turbulent flow within a PCB enclosure model. Flow Turbul. Combust. **77**, 77–95 (2006)
15. Zaleski, S.: Science and fluid dynamics should have more open sources. http://www.lmm.jussieu.fr/∼zaleski/OpenCFD.html (2001)

V. **Paper 4 :** “Use of time resolved PIV for validating LES/DNS of the turbulent flow within a PCB enclosure model”

Flow Turbulence Combust (2006) 77: 77–95
DOI 10.1007/s10494-006-9038-2

Use of Time Resolved PIV for Validating LES/DNS of the Turbulent Flow within a PCB Enclosure Model

G. Usera · A. Vernet · J. A. Ferré

Accepted: 12 January 2006 / Published online: 12 August 2006

Abstract In this paper an approach to the validation of transient numerical simulations of turbulent flows through the use of time resolved PIV data is presented and applied to the case of turbulent flow within a simplified model of printed circuit board (PCB) enclosure. The comparison between numerical and experimental data is not limited here to the time averaged fields, but transient flow structures are also addressed. With this purpose a set of conditionally sampled averages, or cluster averages, are obtained through a Fuzzy Clustering technique. Further, the time history of the flow is analyzed on the base of this set of cluster averages, yielding time sequence relations among them that simplify their interpretation in terms of flow dynamics. In this way, events of alternate vortex shedding have been identified in a selected portion of the flow, and showed to exist in both numerical and experimental sets of data.

Key words conditional sampling · fuzzy clustering · vortex shedding

Supported by FPI/DPI2003-06725-C02-01 from DGI, Ministerio de Educación y Cultura y Fondos FEDER, and grant S/B/CE/33/07 from PDT Uruguay.

G. Usera
IMFIA, Universidad de la República, J.H. Reissig 565, 11300 Montevideo, Uruguay
e-mail: gusera@fing.edu.uy

A. Vernet · J. A. Ferré (✉)
Departament d'Enginyeria Mecánica, Universitat Rovira i Virgili, Av. Paisos Catalans 26, 43007 Tarragona, Spain
e-mail: josep.a.ferre@urv.net

A. Vernet
e-mail: anton.vernet@urv.net

1. Introduction

The overall aim of the ongoing research is to investigate the velocity field in complex domains and its influence on the heat transport by natural and/or forced convection. The specific application target is the turbulent flow and heat transport within printed circuit boards (PCB) enclosures. In the present stage of the investigation only the velocity field is under study, while plans for simultaneous measurement of velocity and temperature, through two-color Laser Induced Fluorescence (LIF) [8], are being implemented. In this paper experimental measurements obtained with time resolved PIV and numerical results from DNS at moderate Reynolds number will be analyzed and compared, with an emphasis on the transient behavior of the flow.

The increasing availability of full field measurement techniques with good time resolution (e.g. time resolved PIV) and of higher order numerical simulation methods for turbulent flows, (e.g. LES or DNS) provides the fluid dynamic researcher with databases of increased level of flow detail in the form of extended time series. To efficiently use these types of data, specialized methods need to be developed. Moreover, the validation of numerical simulation methods against time resolved PIV also requires methods that allow to check the transient dynamics of the simulation rather than just the mean statistical properties. Some of the widely used techniques that can be found in the literature are, among others, the Proper Orthogonal Decomposition (POD), the Pattern recognition (PR), and, to a lesser extent, the Singular Spectral Analysis (SSA), the Coherent Vortex Simulation (CVS), and Fuzzy Clustering (FC).

Proper orthogonal decomposition (POD) involves the computation of a subset of the eigenvectors and eigenvalues of the correlation tensor, in decreasing order of the eigenvalues magnitude. The eigenvectors with the largest eigenvalues can give an insight into the underlying large-scale structure of the flow [15]. The use of these leading eigenvectors as starting templates for pattern recognition in the search of coherent structures has been considered by [10]. In addition, the projection of the original data set onto the subset of eigenvectors, allows for reducing the dimensionality of the data set with a loss of information limited to the finer, and thus less energetic, scales of the flow. This is actually the purpose of applying POD within this work. A summary of the applications of the POD technique can be found in [1]. More recently, extensions based on POD have been proposed by [6, 13].

While POD is typically applied in the spatial domain, a class of techniques exist, named generically Singular Spectral Analysis, which extend the application of the POD concept to the time domain [7]. Multivariate SSA allows for simultaneous decompositions of the spatial and temporal domains into data adapted basis functions. Wavelet based orthogonal decomposition techniques have also been proposed, as for instance the (CVS) in [2, 3] as alternatives to POD. The Coherent Wavelet Simulation is a non-linear wavelet-based decomposition technique that adapts the number and selection of resolved modes, and keeps only the strongest wavelet coefficients at each time step. It thus rivals POD, which is a strategy with a fixed number of resolved modes, in terms of compression efficiency, although scarifying simplicity.

Pattern recognition comprises another set of techniques used to identify flow structures governing the dynamics of the flow [4, 5]. In PR prototypical structures or templates are correlated against instantaneous events to form ensemble averages, through an iterative procedure. However, usually only a moderate percentage of the instantaneous frames contained in the time series is involved in the pattern classification [10, 19].

On the other hand, fuzzy clustering is a technique that allows the classification of all the instantaneous frames into a specified number of conditionally averaged or coherent subsets [20]. This technique can identify flow structures whose existence remains hidden within the usual unconditional averaging procedure [18].

In this paper a novel conditional sampling method based on a fuzzy clustering technique is applied to the analysis of the large-scale structures of the turbulent flow in an experimental model of a PCB enclosing. To overcome the limitations of the conventional (unconditional) time averaging, a set of conditionally sampled averages, or cluster averages, are obtained with an improved signal-to-noise ratio. Further, the time history of the flow is analyzed on the base of this set of cluster averages, yielding the time sequence relations among them that simplify and complement their interpretation in terms of flow dynamics. This approach is applied to the validation of numerical simulations of this flow against time resolved PIV measurements.

2. Experimental Setup and Technique

The specific application targeted is the turbulent flow and heat transport within PCB enclosures. With this purpose a simplified model of such enclosure was built in transparent Plexiglas to allow optical access for the PIV system. The model was run with water in a closed circuit. The water entering the model was gravity driven from an elevated deposit to minimize perturbations, with a pump downstream of the model driving the water back to the elevated deposit. Typical flow rates were in the $100 - 500\ l/h$ range with maximum attainable flow rate at about $2,300\ l/h$.

A sketch of the experimental model is shown in Figure 1. The entrance and outlet openings have square sections of side $h = 0.024m$. The upper and lower channels are $6h$ long by $3h$ wide, and of height h each. The central plate is only $5h$ long. Two sets of measurements at different locations were obtained, at Reynolds numbers of $Re = 1.16 \times 10^3$ and 5.79×10^3 relative to the entrance section, which correspond to mean velocities at this section of $Uo = 4.8 \times 10^{-2}$ and $2.4 \times 10^{-1} m/s$, respectively. In this paper only the results at the lower Reynolds number are presented, with an emphasis on the separation region near the lower channel entrance, as pictured by region (D) in Figure 1.

Time resolved PIV recordings were obtained using an 420 by 480 pixels Motion Scope PCI 1000 S digital camera capable running at 125 and 250 Hz sampling rates for 512 image strips. The size of each image strip was limited by the internal memory of

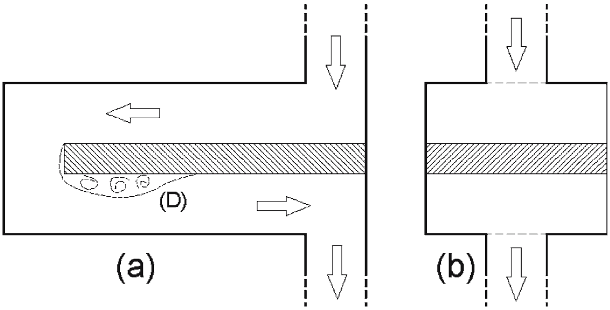


Figure 1 Sketch of experimental model with main paths of flow. **a** *Side view* with separation region (*D*) shown at the lower channel entrance. **b** *Front view*.

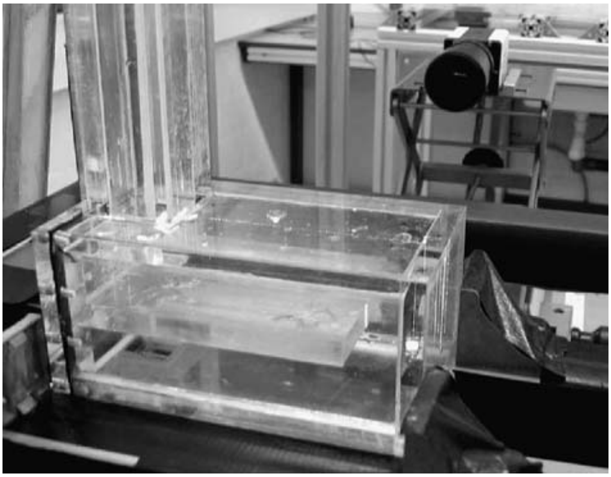


Figure 2 Image of experimental setup.

the camera. For the results presented here, at the lower Reynolds number, the lower sampling frequency was used. Illumination was provided by a Monocrom DPSSL 532 nm pulsed laser source, with a cylindrical lens. *Licopode* spores were used as seeding material, which absorb water providing almost spherical, neutrally buoyant, particles. An image of the experimental setup is presented in Figure 2 showing the Plexiglas model in the front and the PIV camera at the back. The Laser source is located to the right of the enclosure model, and reflections from the Laser beam onto the Plexiglas can be hinted at the outer shell of the model and at the front face of the inner plate.

The PIV processing algorithm used here is of the iterative pattern deformation type, following roughly the ideas of [14], and is described in detail in [17]. The iterative pattern deformation procedure allows for a spatial resolution that is almost independent of PIV interrogation area sizes and an enhanced behavior in strong gradient situations. Special care has been taken to improve the performance of the PIV interrogation near the wall boundaries, as well as to exploit the special characteristics of time resolved PIV recordings.

The time step between consecutive images is more than one order of magnitude smaller than representative integral time scales of the flow, as reported later in Section 5 and evidenced in Figure 9. Thus, the PIV recordings obtained in these experiments can be considered to be ‘time resolved’ in the sense discussed in [17]. A significant correlation will then exist between velocity fields obtained from consecutive pairs of images. This enables the application of multiple image correlation techniques together with reflection removal procedures to enhance the PIV processing [17]. The analysis of transient structures presented later also requires such degree of correlation between consecutive velocity fields.

3. Numerical Method

The numerical simulations considered here were obtained with the in-house flow solver *caffa3d.MB* developed at Rovira i Virgili University. It is an original Fortran95 implementation of a fully implicit finite volume method for solving the 3D incompressible Navier–Stokes equations in complex geometry. This three-dimensional solver is based on a two-dimensional solver described in [12].

Spatial discretization is based on block-structured, non-orthogonal, body fitted, collocated grids with first order (UDS) and second order (CDS) schemes for the convective terms. Improved linear interpolation schemes for non-orthogonal grids are also included following [11]. For the time discretization fully implicit two-level first order (implicit backward Euler) and three-level second order schemes are available. The SIMPLE algorithm is implemented for the coupling between velocity and pressure. Different interfaces between grid blocks are currently supported, among which only *one-to-one* interfaces are used in here. Domain decomposition on the basis of grid blocks is applied through OpenMP, with current simulation running on dual processor machines. A full description of the flow solver can be found in [16].

The grid for the current case was made of five regular blocks of resolution $h/40$, as pictured in Figure 3, for a total of 3.02×10^6 cells. Inlet and outlet duct blocks were used to minimize the effect of boundary conditions onto the region of interest of the flow. The dimensionless time step was $dt/(h/U_0) = 1 \times 10^{-2}$ and the simulation was allowed to reach a steady regime before collecting time series of results. Reynolds number was set to $Re = 1.16 \times 10^3$ as for experiments, and no turbulence model was applied.

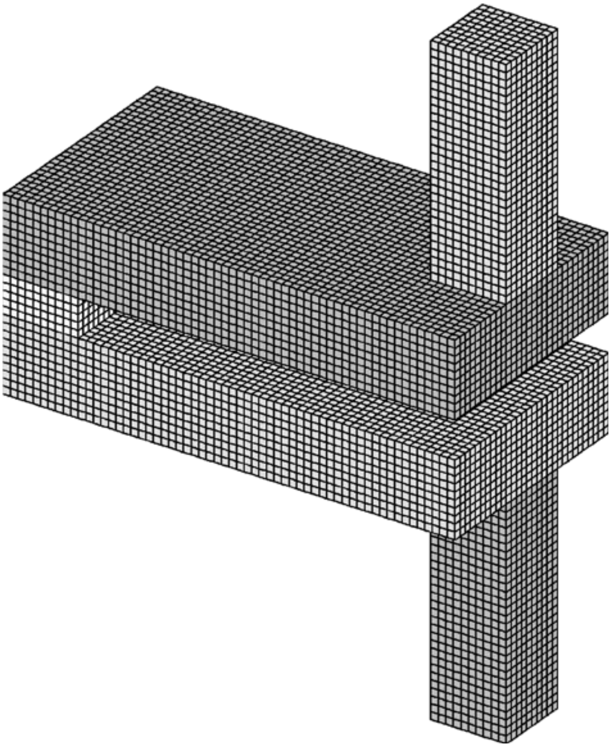


Figure 3 Numerical grid composed of five *matching* grid-blocks. Shown at $h/10$ resolution.

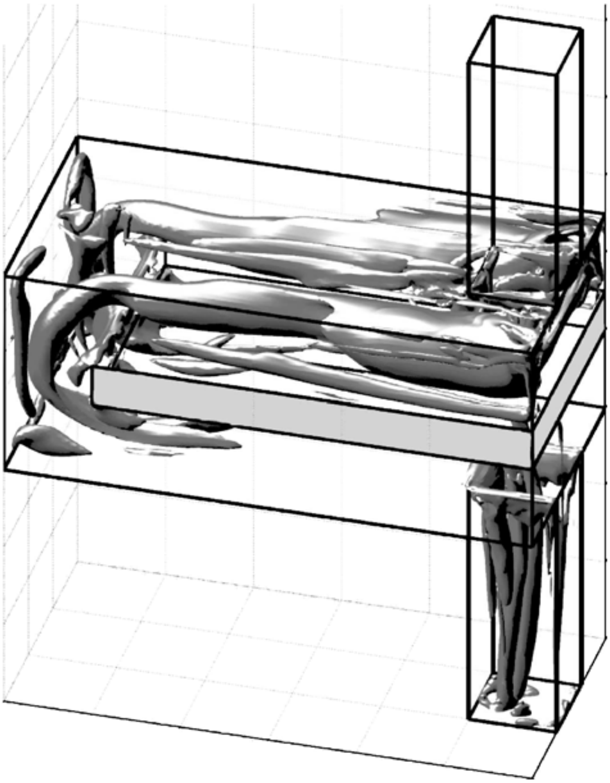


Figure 4 Vortical structures educed by λ_2 iso-surfaces from numerical simulation.

Figure 4 shows instantaneous vortical structures visualized by means of λ_2 iso-surfaces [9] obtained in the numerical simulation, in order to have an overview of the characteristics of the analyzed system. It can be seen that the flow in the upper channel is dominated by two counter rotating vortical structures. The non-stationary characteristic of the flow is also apparent from Figure 4.

4. Analysis of Mean Fields

The following analysis will be restricted to a portion of the flow, selected due to the interesting characteristics that the flow exhibits there and also because its expected role in the transport of heat away from the central plate, where the PCB would be located. The selected portion is the entrance to the lower channel in the vicinity of region (D) in Figure 1, beneath the central plate. Figure 5 shows the streamlines for the mean velocity field obtained from the ensemble averaged velocity field of the complete 2,550 frames PIV data set. Each PIV velocity field for the considered region contains $22 \times 64 = 1,408$ velocity vectors. It is seen that the mean velocity field is characterized, in the analyzed region, by a strong recirculation downstream from the leading edge of the plate, that separates the upper channel from the lower one, with the center of the re-circulation vortex at $x/h = 1.7$ and $y/h = 0.8$. Thus,

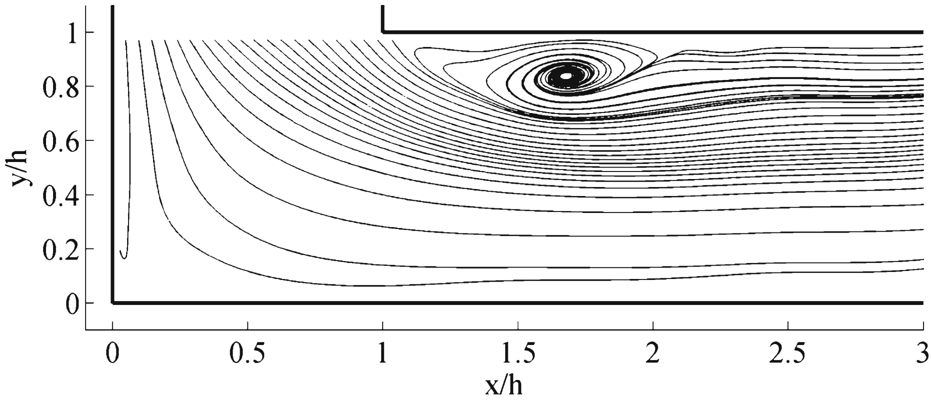


Figure 5 Mean velocity field streamlines in lower channel entrance region, showing re-circulation. Time resolved PIV experimental results.

the main flow entering the lower channel is deflected downwards, passing through a section of height about 2/3 of the total height of the channel.

This type of representation of the mean flow should not be pushed too far in drawing conclusions about the dynamics of the flow. For instance, if the instantaneous flow was to be permanently as pictured in Figure 5, the heat transfer from the plate near the re-circulation region could be adversely affected. This would be so because little exchange with the main flow entering the lower channel would be allowed. Thus, it is important to assess the transient evolution of the flow in this region, since transport of heat away from the plate might be enhanced by transient structures.

The reminder of this analysis will be based onto the spanwise vorticity field, since it is expected to be well suited to reveal the underlying structures in the re-circulation

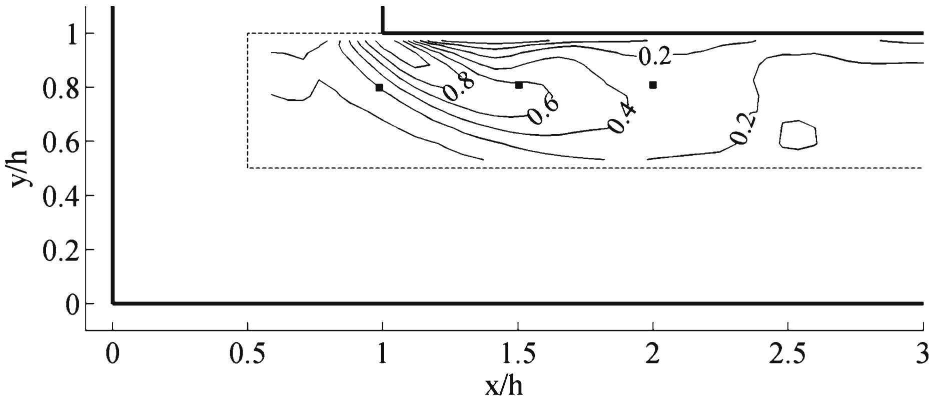


Figure 6 Normalized mean spanwise vorticity contours ($\omega_z/(2U_0/h)$). Time resolved PIV experimental results.

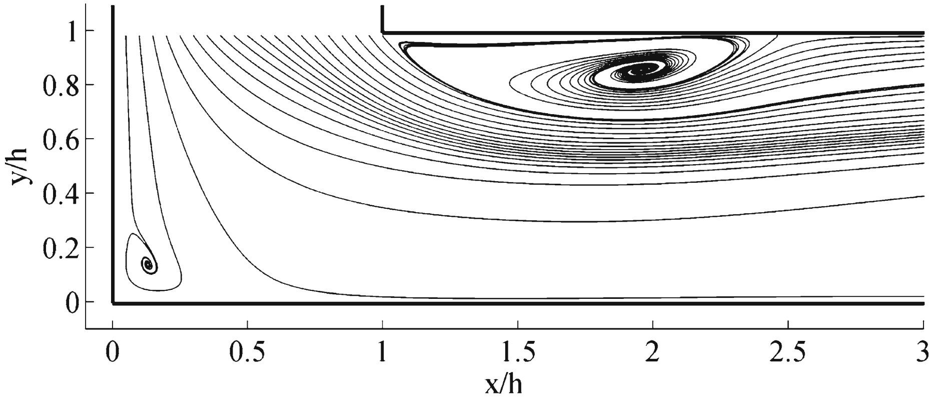


Figure 7 Mean velocity field streamlines in lower channel entrance region. Numerical simulation results.

region, which are expected to be mainly vortical structures. In Figure 6, the mean dimensionless spanwise vorticity field is pictured, as defined by:

$$\omega_z \cdot \frac{h}{2U_0} = \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) \cdot \frac{h}{2U_0} \tag{1}$$

where U_0 is the main velocity relative to the entrance section, u and v are the streamwise and vertical velocity components, and $h/2$ is the half height of the channel.

Figure 6 shows that the relative maximum of vorticity is roughly aligned with the shear layer defined between the main flow entering the lower channel and the re-circulation area. This layer weakens towards $x/h = 2.5$ marking the downstream extension of the re-circulation area. A region of interest around the re-circulation area has been marked with a dashed line in Figure 6. Further plots will be restricted only to this region of interest.

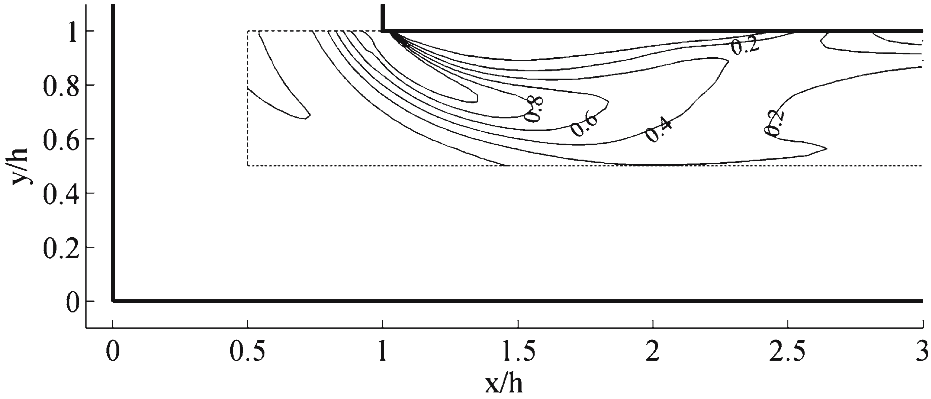


Figure 8 Normalized mean spanwise vorticity contours ($\omega_z/(2U_0/h)$). Numerical simulation results.

These results for the mean flow obtained from the PIV experimental measurements will be compared now with corresponding results from the numerical simulation, obtained after averaging over 1.3 × 10^2 integral time scales of the flow. Figure 7 presents mean flow streamlines while in Figure 8 mean normalized spanwise vorticity contours are given, corresponding, respectively, to Figures 5 and 6.

The patterns of streamlines pictured in Figures 5 and 7 are quite similar. The vertical extension of the recirculation regions is approximately $h/3$ in both cases, with the center of the recirculation pattern slightly displaced downstream in the numerical simulation results respect to the PIV experimental results. Also a small vortex appears at the lower left corner in the numerical results that is hidden in the PIV results, probably due to their lower resolution.

Mean spanwise vorticity contours of Figures 6 and 8 exhibit an even better agreement. Contour levels match very closely, being slightly higher for the numerical results, again probably due to their higher resolution. Still, the downstream weakening of the shear layer near $x/h = 2.5$ is quite well reproduced.

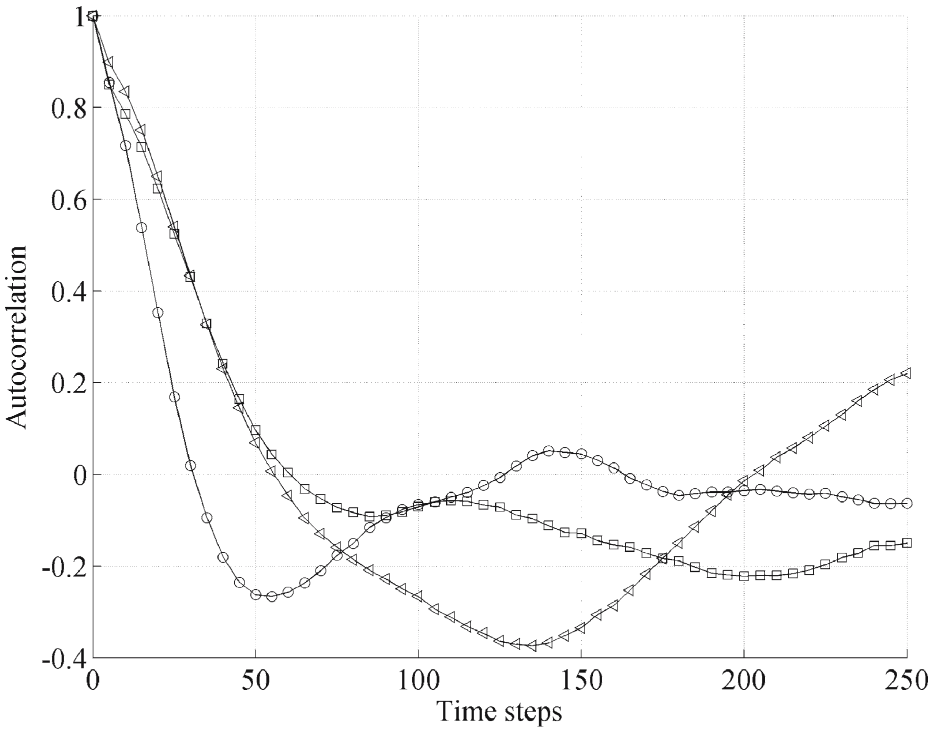


Figure 9 Auto-correlation of spanwise vorticity time series at $y/h = 0.8$ and $x/h = 1.0$ (-o-), $x/h = 1.5$ (-∇-) and $x/h = 2.0$ (-□-).

5. Analysis of Transient Structures

The analysis of the transient structures arising in the region marked in Figure 6 is considered next. This analysis will be first presented for the experimental data and then results from both experimental and numerical data sets will be compared. The analysis method followed here, combining POD and FC, was first described in [18].

First, the auto-correlation function for the spanwise vorticity time series was computed at selected locations, from the PIV data set. These auto-correlation

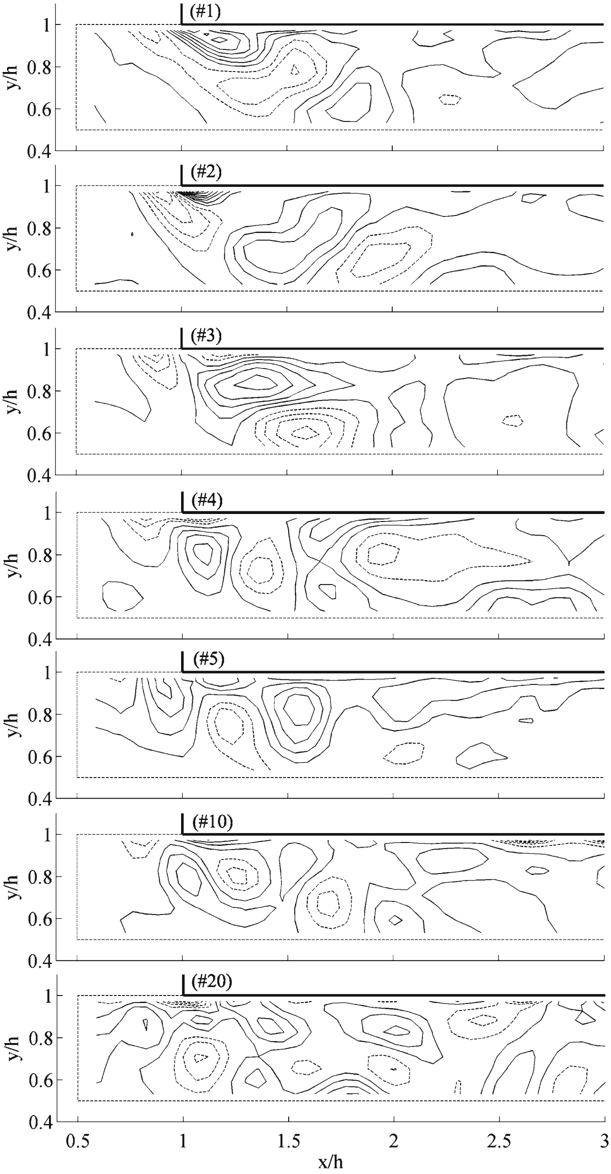


Figure 10 POD eigenvectors for spanwise vorticity field. From *top to bottom*: #1, #2, #3, #4, #5, #10, #20.

functions are plotted in Figure 9. The three selected locations are indicated in Figure 6 by small squares and their co-ordinates are $y/h = 0.8$ and $x/h = 1.0, 1.5, 2.0$. The first one falls in the main flow, right outside the re-circulation region, while the second and third lie within this region. Dimensionless integral time scales for vorticity fluctuations at these locations are $T/dt = 1.5 \times 10^1, 2.4 \times 10^1, 2.4 \times 10^1$, respectively, while the dimensionless sampling time for each PIV series was $Ts/dt = 510$. Thus, the whole PIV experimental time series spanned about 1.1×10^2 integral time scales.

In relation to Figure 9, it is worth noting that, while the auto-correlation functions at locations 1 and 3 oscillate around zero after a few integral time scales, in the case of location 2 the auto-correlation function exhibits a deeper oscillation of larger amplitude and duration. This might reveal the existence of cyclic vortex passing near this location, which in fact will be later confirmed by the clustering analysis below.

Next, proper orthogonal decomposition was applied to the time series of instantaneous spanwise vorticity fields within the dashed region of Figure 6, which consisted of $10 \times 48 = 480$ grid points and 2,550 time steps assembled in five series of 510 time steps each. The first 40 eigenvectors and eigenvalues were computed, accounting cumulatively for up to 70% of the total variance and with the last eigenvalue accounting for less than 1% of it. The *normalized* eigenvector vorticity fields for eigenvalues 1, 2, 3, 4, 5, 10 and 20 are given in Figure 10 that shows an increasingly

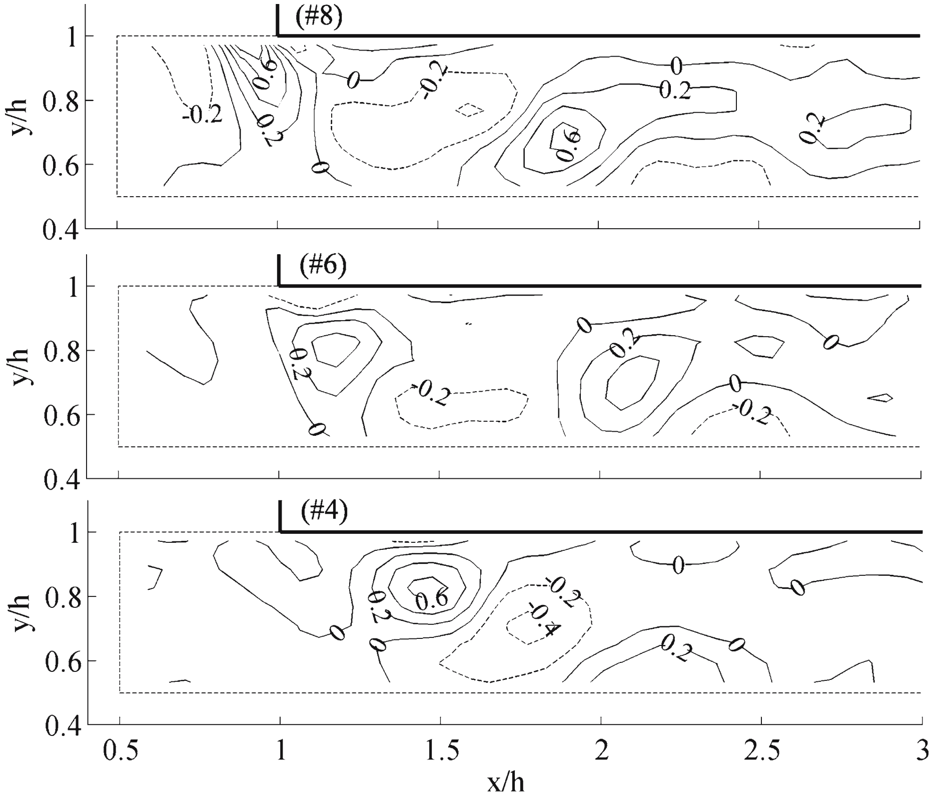


Figure 11 Cluster sequence #8-6-4. Time resolved PIV experimental results.

finer hierarchy of vortical structures, as expected from a spectral like decomposition, with alternate vorticity sign within the re-circulation region.

However, the analysis of the eigenvectors alone gives no indication of the temporal evolution of these structures. It should be noted that the results from POD analysis would be the same regardless of the temporal organization of the instantaneous time steps. The information regarding the temporal organization of the flow is kept then in the projections of the time series onto these eigenvalues and an analysis of these time series of projections is required in order to recover this information. This will be the objective of the following paragraphs where a fuzzy clustering algorithm will be first presented and then applied to determine a set of ensemble averages or clusters, which picture the most representative states of the flow and the transitions between them.

5.1. Fuzzy clustering algorithm

The fuzzy clustering algorithm applied here is the c-Means clustering algorithm coupled with a validity criteria proposed by Xie and Beni, both of which are described in a general context in [21]. Application of this method to time series of fields in fluid

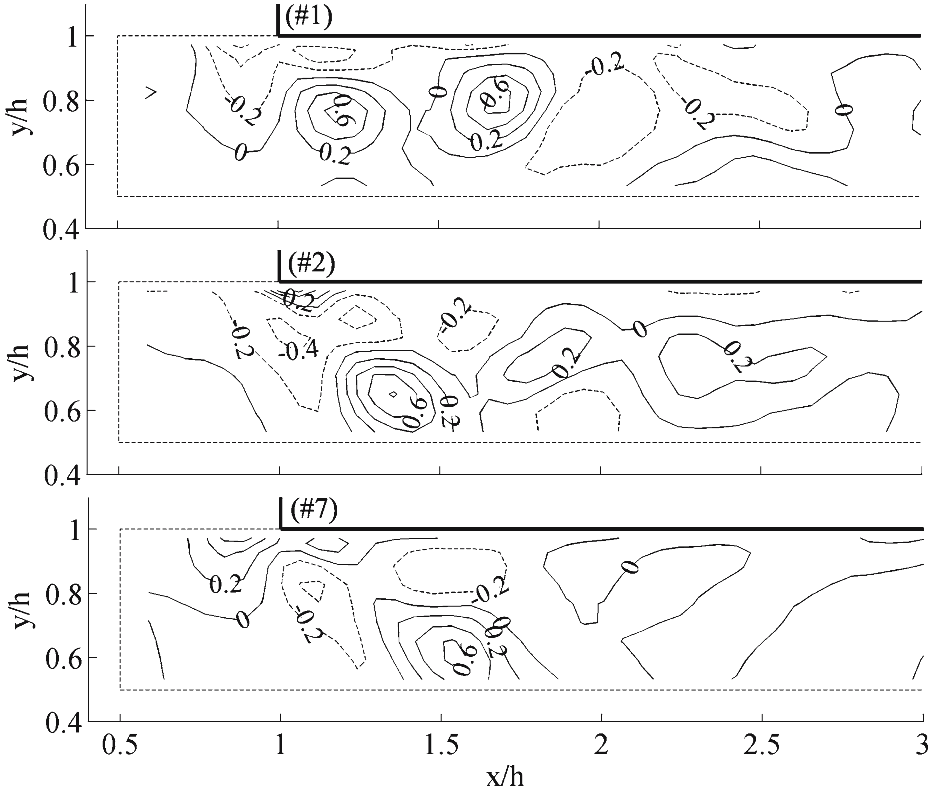


Figure 12 Cluster sequence #1-2-7. Time resolved PIV experimental results.

dynamics has been proposed in [18, 20]. Here a brief description of the method will be given following [21] and [18].

The c-Means Clustering Algorithm seeks the minimization, for a prescribed number of clusters c, and fuzziness index m ($m > 1$), of an objective function J_m, defined as:

$$J_m = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(\mu_{ij}\right)^m d^2\left(X_j, V_i\right) \tag{2}$$

where X_j (j=1,2,...,n) is the data set to be clustered and $d^2(X_j, V_i)$ is a measure of the distance between the vectors X_j and the cluster center V_i, usually taken as the Euclidean distance, and μ_{ij} is the fuzzy membership function that verifies the condition:

$$\sum_{i=1}^{c} \mu_{ij} = 1 \tag{3}$$

For any partition, the cluster centroids are defined from:

$$V_i = \frac{\sum_{j=1}^{n} \left(\mu_{ij}\right)^m X_j}{\sum_{j=1}^{n} \left(\mu_{ij}\right)^m} \tag{4}$$

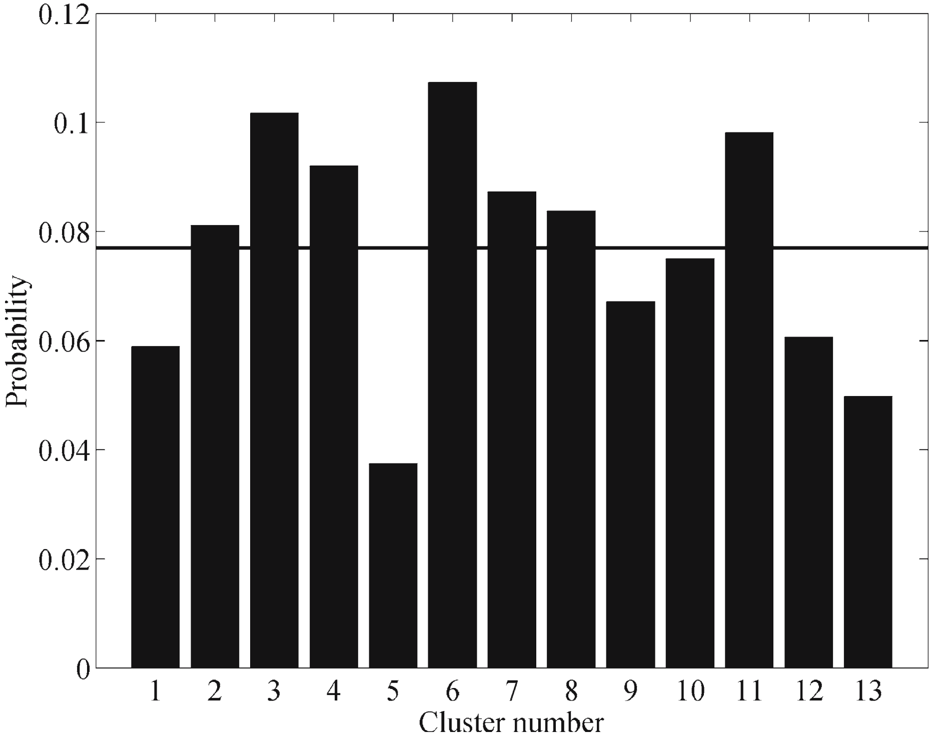


Figure 13 Probability of occurrence of each cluster. *Thick line* indicates the mean probability (1/13). Time resolved PIV experimental results.

Also, the fuzzy membership function can be computed from a specified set of centroids V_i from:

$$\mu_{ij} = \frac{\left(\frac{1}{d^2(X_j,V_i)}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c}\left(\frac{1}{d^2(X_j,V_i)}\right)^{\frac{1}{m-1}}} \tag{5}$$

The minimization of J_m is attained by iterating equations (4) and (5) from a starting guess of either μ_{ij} or V_i until no further improvement of J_m (Equation (2)) is observed.

Still, the c-Means Clustering Algorithm by itself gives no clue about the number of clusters to be used in partitioning the set. In order to determine the optimum number of clusters to be used for each data set, as well as the value of the fuzziness index m, the validity criterion for fuzzy clustering [21], was applied. For any given partitioning, regardless of the algorithm used to determine it, a compactness and separation validity function S, can be defined as:

$$S = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^2 d^2\left(X_j, V_i\right)}{n \cdot min_{ij} d^2\left(X_j, V_i\right)} \tag{6}$$

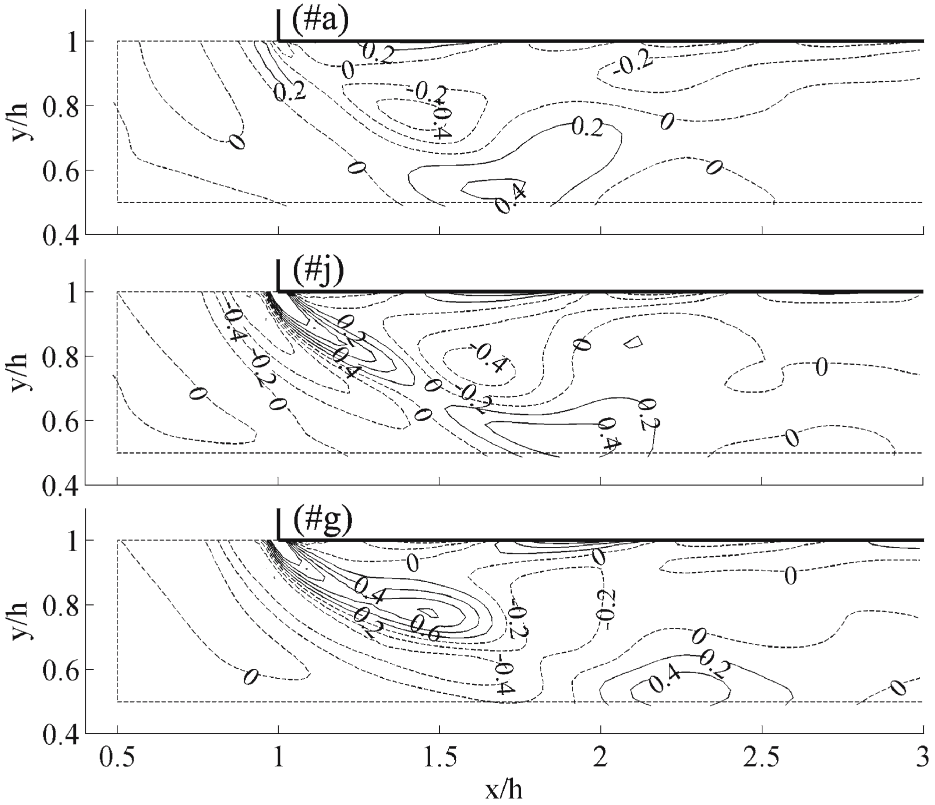


Figure 14 Cluster sequence #a-j-g. Numerical simulation results.

The validity function *S* can be regarded as the ratio of the measure of the compactness of the clusters over the minimum separation among clusters. Optimum values of *c* and *m* are those for which *S* reaches an absolute minimum value. In some cases however one might be content to find a relative minimum value.

5.2. Fuzzy clustering results

The FC algorithm described in the previous section, together with the partition validity criterion, was applied to the PIV data set. For the vorticity field data considered before an optimum value for *S* was found at $c = 13$ and $m = 1.16$. Not all cluster ensemble averages will be reproduced here, but only six of them selected due to the frequent transitions that the flow exhibits among them.

In Figure 11 a first set of three clusters, or ensemble averages for spanwise vorticity fluctuations, is presented comprising clusters #8, #6 and #4, while in Figure 12 the second set contains clusters #1, #2 and #7. The cluster index is assigned arbitrarily during the iteration process. The ordering of the clusters among each set was decided according to the sequence of the transitions of the flow between states corresponding to each cluster.

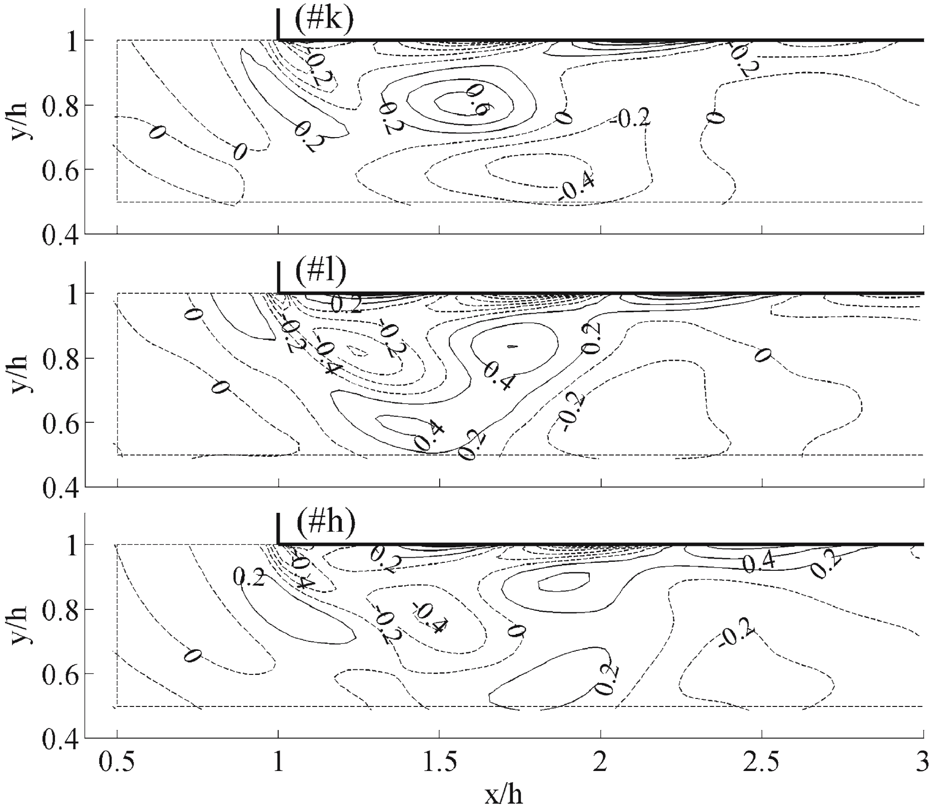


Figure 15 Cluster sequence #k-l-h. Numerical simulation results.

Figure 11 shows that an array of vorticity fluctuation spots with alternate sign evolve downstream. In particular a positive vorticity spot forms near the leading edge of the plate (clusters #8 and #6) and then drifts downstream (clusters #6 and #4). A similar picture emerges in Figure 12, although there the vortex forming at the leading edge has negative spanwise vorticity, while a vortex with positive spanwise vorticity is evolving downstream and away from the plate. The downstream drift of these structures is expected to be associated with a net transport of heat away from the plate that would enhance the limited transport of heat associated solely to the re-circulation pattern of the mean velocity field.

The probability of occurrence of each cluster is given in Figure 13. This figure can also be interpreted as the percentage of cumulative time spent by the flow in configurations classified as belonging to each cluster. From this point of view, the series presented in Figures 11 and 12 account, respectively, for 22% and 28% of the time, so that together they represent the evolution of the flow for about 50% of the total time.

Corresponding results obtained by applying the same FC procedure to the numerical simulation data set are presented now, for comparison with the experimental results. In this case, an optimum value for S was found at $c = 12$ and $m = 1.24$. While the near coincidence in the number of clusters found to be optimal for the classification is encouraging, it is most interesting to check whether the cluster ensemble

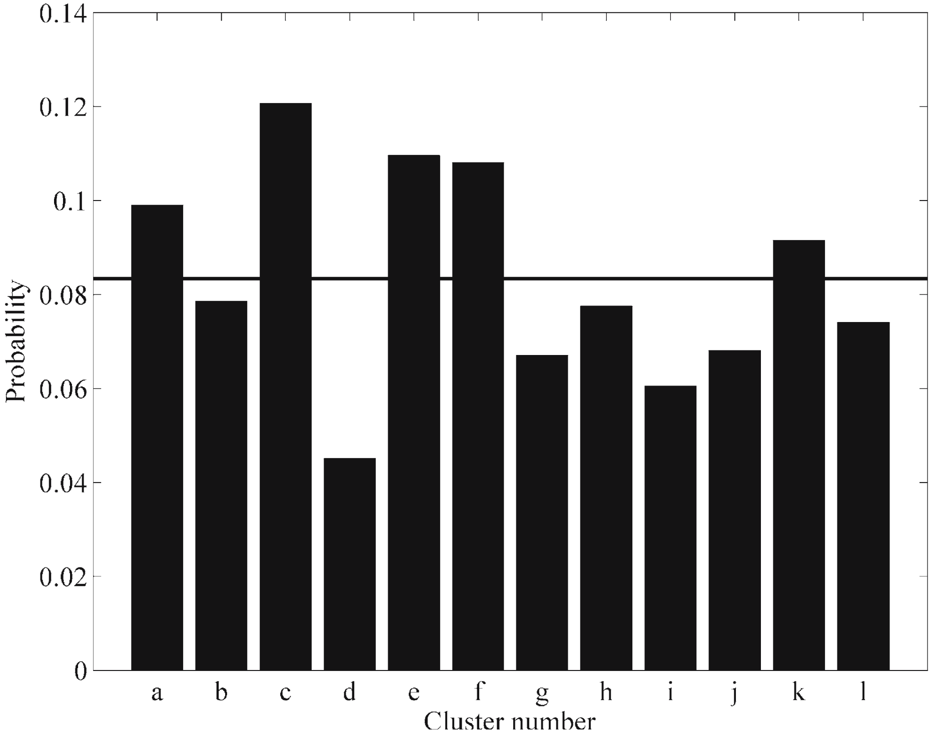


Figure 16 Probability of occurrence of each cluster. *Thick line* indicates the mean probability (1/12). Numerical simulation results.

averages themselves depict similar patterns as those found for the experimental data set.

Again two sequences of three clusters were selected, based on the frequent transitions between them observed in the flow. Since the initial indexing of clusters is arbitrary, letters have been assigned to the clusters obtained from the numerical data set to prevent confusion with the experimental ones. Thus, Figure 14 presents the cluster ensemble average sequence #a-j-g of spanwise vorticity fluctuation contours, while in Figure 15 sequence #k-l-h is presented. The selected cluster ensembles amount for up to 48% of the total time of simulated flow time series, as can be seen from Figure 16.

The evolution of vortical structures in Figures 14 and 15 resembles that of Figures 11 and 12, evidencing that the numerical simulation is in fact reproducing the same vortex shedding dynamics observed in the experimental results. The comparison between sequences is not necessarily on the basis of individual cluster ensembles, but rather on the trend expressed by the cluster ensemble sequences.

Figures 11, 12, 14 and 15 presented cluster ensemble averages of spanwise vorticity fluctuations. A complementary view can be given by adding the mean spanwise vorticity field to the spanwise vorticity fluctuations. This is pictured in Figure 17 for

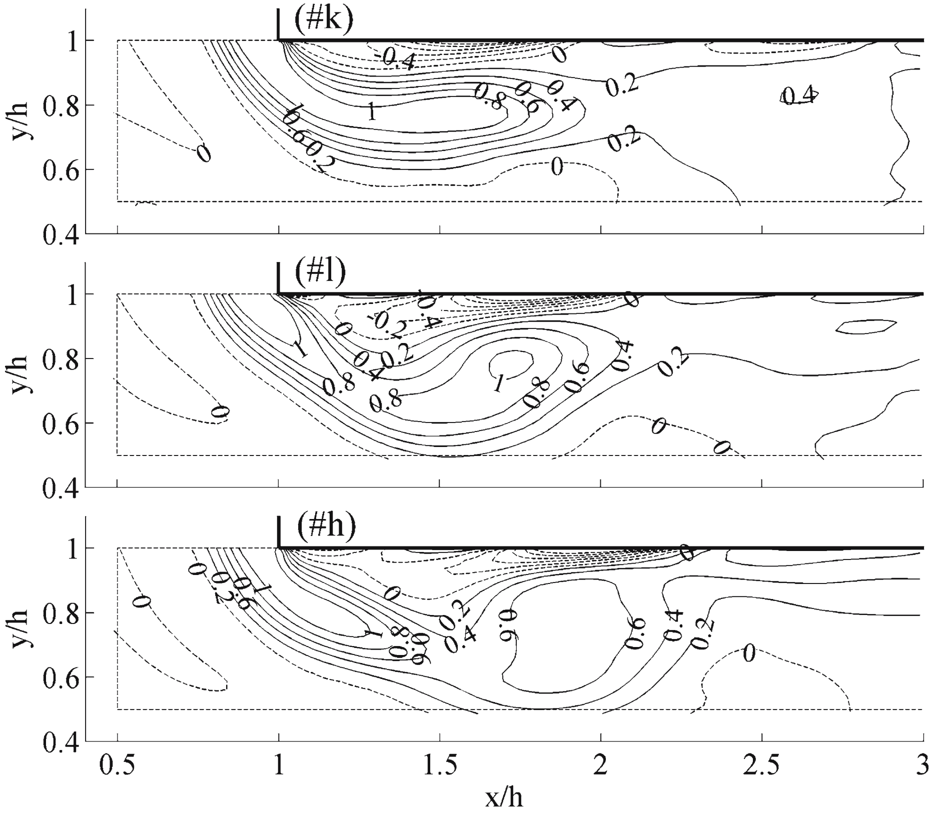


Figure 17 Cluster sequence #k-l-h, superimposed to the mean field. Numerical simulation results.

the case of cluster sequence #k-l-h obtained from the numerical data set. Vorticity contours in Figure 17 show a flapping shear layer that sheds a vortex, which then drifts downstream.

6. Conclusions

The flow within a simplified model of a printed circuit board (PCB) enclosing has been analyzed both experimentally and numerically, focusing on a selected portion of the flow.

A conditional sampling technique based on a Fuzzy Clustering algorithm has been applied to the analysis of time series of spanwise vorticity fields obtained from time resolved PIV and from the transient numerical simulations. This procedure allows for the comparison of not only mean average fields, but also of sequences of conditionally sampled ensemble averages. In this way validation of the numerical simulation against the experiments is not limited to the mean statistical properties of the flow, so that the transient behavior of the flow can be compared too.

The selected portion of the flow presented in this communication shows a recirculation region beneath the plate at the entrance of the lower channel. Events of alternate vortex shedding have been identified through the fuzzy clustering analysis, in both experimental and numerical data sets. These events, which were hinted by the auto-correlation function, may enhance the transport of heat beneath the plate, with respect to that which would be produced by the mean flow.

Future work will consider an evaluation of the temperature field and its relationship with the velocity through simultaneous measurements with two-color Laser Induced Fluorescence (LIF) in order to quantify the effect of these structures on the transport of heat.

References

1. Berkooz, G., Holmes, P., Lumeley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**, 539–575 (1993)
2. Farge, M., Schneider, K.: Coherent Vortex Simulation (CVS), a semi-deterministic turbulence model using wavelets. Flow Turbul. Combust. **66**, 393–426 (2001)
3. Farge, M., Schneider, K., Pellegrino, G., Wray, A.A., Rogallo, R.S.: Coherent vortex extraction in three-dimensional homogeneous turbulence: Comparison between CVS-wavelet and POD–Fourier decompositions. Phys. Fluids **15**(10), 2886–2896 (2003)
4. Ferré, J.A., Giralt, F.: Pattern-recognition analysis of the velocity field in plane turbulent wakes. J. Fluid Mech. **198**, 27–64 (1989)
5. Ferré, J.A., Mumford, J.C., Savill, A.M., Giralt, F.: Three-dimensional large eddy motions and fine-scale activity in a plane turbulent wake. J. Fluid Mech. **210**, 371–414 (1990)
6. Garnard, S., George, W.K., Jung, D., Woodwrad, S.: Application of a *slice* proper orthogonal decomposition to the far field of an axisymmetric turbulent jet. Phys. Fluids **14**(7), 2515–2522 (2002)
7. Ghil, M., Allen, M.R., Dettinger, M.D., Ide, K., Kondrashov, D., Mann, M.E., Robertson, A.W., Saunders, A., Tian, Y., Varadi, F., Yiou, P.: Advanced spectral methods for climatic time series. Rev. Geophys. **40**, 1 (2002)
8. Hishida, K., Sakakibara, J.: Combined planar laser-induced fluorescence-particle image velocimetry technique for velocity and temperature fields. Exp. Fluids **29**, 129–140 (2000)
9. Jeong, J., Hussain, F.: On the identification of a vortex. J. Fluid Mech. **285**, 69–94 (1995)

10. Kopp, G.A., Ferré, J.A., Giralt, F.: The use of pattern recognition and proper orthogonal decomposition in identifying the structure of fully-developed free turbulence. J. Fluids Eng. **119**, 289–296 (1997)
11. Lehnhauser, T., Schafer, M.: Improved linear interpolation practice for finite-volume schemes on complex grids. Int. J. Numer. Methods Fluids **38**, 625–645 (2002)
12. Lilek, Z., Muzaferija, S., Peric, M., Seidl, V.: An implicit finite-volume method using nonmatching blocks of structured grid. Numer. Heat Transf., Part B **32**, 385–401 (1997)
13. Maurel, S., Borée, J., Lumley, J.L.: Extended proper orthogonal decomposition: Application to Jet/Vortex interaction. Flow Turbul. Combust. **67**, 125–136 (2001)
14. Nogueira, J., Lecuona, A., Rodríguez, P.A.: Local field correction PIV, implemented by means of simple algorithms and multigrid versions. Meas. Sci. Technol. **12**, 1911–1921 (2001)
15. Reichert, R.S., Hatay, F.F., Biringen, S., Huser, A.: Proper orthogonal decomposition applied to turbulent flow in a square duct. Phys. Fluids **6**(9), 3086–3092 (1994)
16. Usera, G., Vernet, A., Ferré, J.A.: Simulación numérica de las ecuaciones de Navier Stokes en 3D, por volúmenes finitos en mallas curvilíneas estructuradas por bloques. Congreso de Métodos Numéricos en Ingeniería, Granada, Spain (2005)
17. Usera, G., Vernet, A., Ferré, J.A.: Considerations and improvements on analyzing algorithms for time resolved PIV of turbulent wall bounded flows. In: 12th International Symposium. Applications of Laser Techniques to Fluid Mechanics (2004)
18. Usera, G., Vernet, A., Pallares, J., Ferré, J.A.: A conditional sampling method based on fuzzy clustering for the analysis of large-scale dynamics in turbulent flows. Eur. J. Mech. B, Fluids **25**(2), 172–191 (2006)
19. Vernet, A., Kopp, G.A., Ferré, J.A., Giralt, F.: Three-dimensional structure and momentum transfer in a turbulent cylinder wake. J. Fluid Mech. **394**, 303–337 (1999)
20. Vernet, A., Kopp, G.A.: Classification of turbulent flow patterns with fuzzy clustering. Eng. Appl. Artif. Intell. **15**(3–4), 315–326 (2002)
21. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-13**(8), 841–847 (1991)