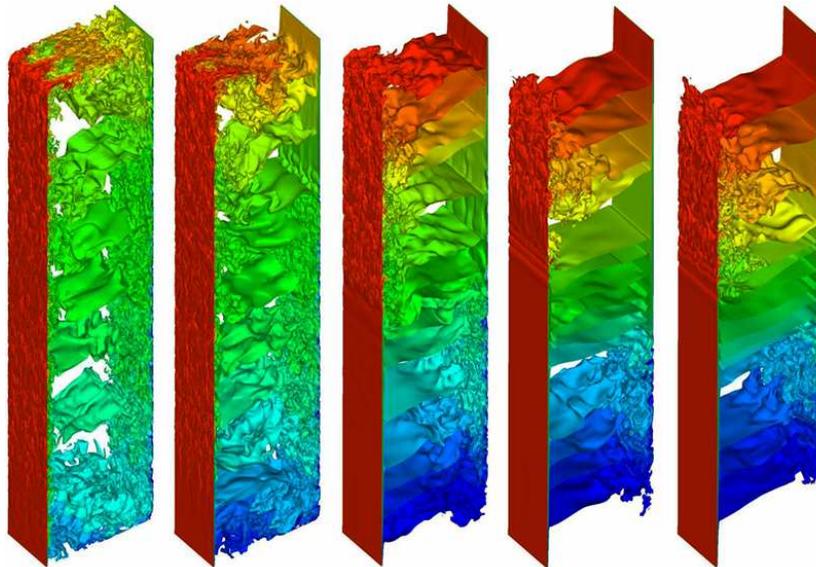


UPC

CTTC

Andrey V. Gorobets

Parallel algorithms for direct
numerical simulations of
incompressible turbulent flows on
supercomputers



DOCTORAL THESIS

Centre Tecnològic de Transferència de Calor
Departament de Màquines i Motors Tèrmics
Universitat Politècnica de Catalunya

Andrey V. Gorobets
Doctoral Thesis

UPC – October 17, 2008

**Parallel algorithms for direct numerical simulations
of incompressible turbulent flows on
supercomputers**

Andrey V. Gorobets

TESI DOCTORAL

presentada al

Departament de Màquines i Motors Tèrmics
E.T.S.E.I.A.T.
Universitat Politècnica de Catalunya

per a l'obtenció del grau de

Doctor Enginyer Industrial

Terrassa, October 17, 2008

**Parallel algorithms for direct numerical simulations
of incompressible turbulent flows on
supercomputers**

Andrey V. Gorobets

Directors de la Tesi

Dr. Manel Soria Guerrero

Dr. Francesc Xavier Trias Miquel

Dr. Assensi Oliva Llena

Tribunal Qualificador

Dr. Carlos David Pérez-Segarra

Universitat Politècnica de Catalunya

Dr. Joaquim Rigola Serrano

Universitat Politècnica de Catalunya

Dr. Roel Verstappen

University of Groningen

Dr. José Fernández Seara

Universidad de Vigo

Dr. Cristóbal Cortés Gracia

Universidad de Zaragoza

Acknowledgements

One day several years ago I was about to leave the science. I was finishing my previous work without considering to be involved in science any second longer. At this moment came Manel Soria proposing cooperation with CTTC. Here it begins...

The work would be hardly possible or at least much more difficult without kind assistance of many people. In particular...

First of all, I would like to thank Prof. Assensi Oliva for giving me the opportunity to do this PhD at CTTC and for the permanent support of this work.

Thanks to Manel Soria, the thesis director. His help and guidance in numerous subjects especially during the first years of my stay in Spain can hardly be overestimated.

Thanks to Xavi Trias, the thesis director as well as a partner in the work the thesis based on. His assistance and guidance was indispensable especially in the last years of the work.

Thanks to Ricard Borrell. His help and support in a wide range of aspects have been playing an important role since the very beginning.

There are many colleagues and friends who taught me and helped with rather annoying problems like bureaucracy I could hardly deal with. Thanks to them all!

Thanks to Roel Verstappen who arranged my stay in Holland in the University of Groningen. His helpful guidance substantially accelerated the work on the surface mounted cube DNS.

Thanks to the Generalitat de Catalunya for its financial support by a FI grant.

Thanks to the Institute of mathematical modelling of RAS for cooperation with CTTC and support of my participation in the foreign work.

MareNostrum supercomputer of the BSC has been primarily used for DNS carried out within this work. Thanks to this institution.

MVS 100000 supercomputer of the Russian academy of science has also been used for solver development and several simulations. Thanks to this institution.

Abstract

The thesis is mainly focused on high-performance computing algorithms in computational fluid dynamics (CFD). The purpose of this work is to contribute to the development of numerical algorithms and parallelization methods for the Direct Numerical Simulation (DNS) of incompressible flows on wide range of parallel systems from loosely coupled clusters to supercomputers. Nowadays DNS is becoming more and more important area of CFD with the rapid growth of the computing power of modern computer systems. Supercomputers are approaching fast the Petaflop boarder. The number of CPU cores in a supercomputer becoming of 10^5 scale. In this conditions efficient scalable parallel algorithms are of the first priority. Meanwhile low-cost loosely coupled clusters are still widely used due to their high performance/cost ration. Hence flexible algorithms that can be efficiently used on both supercomputers and small clusters are of high importance. Development and implementation of such algorithms are within the primary subjects of the thesis. DNS of the turbulent flows which requires huge computing power is one of the main fields of application of the developed algorithms. This class of CFD simulations provides not only new insights into the physics of turbulence but also provides a calibration basis for the future progresses on turbulence modelling. High resolution DNS of complex 3D turbulent flows with two and three wall-bounded directions are of high importance for verification of numerous turbulent models. Application of the developed algorithms for large DNS of such complex flows is also a substantial part of the thesis¹.

In the first chapter, a scalable parallel Poisson solver, named Krylov-Schur-Fourier Decomposition (KSFD), is presented. It is based on fast Fourier decomposition method (FFT) with Krylov method of conjugate gradients (CG) preconditioned with the Direct Schur Decomposition (DSD) algorithm [1, 2]. FFT provides uncoupling of 3D problem in one direction leading to the set of independent 2D problems, each of them is further solved with CG and DSD method. Numerical experiments showing the scalability and the flexibility of the method on both the MareNostrum supercomputer and a PC cluster with a conventional 100 Mbits/s network are presented and discussed.

In chapter 2, the set of large DNS of a differentially heated cavity is presented. The biggest case of the set for Rayleigh number $Ra = 10^{11}$ was solved with 4-th order scheme on a mesh of 1.1×10^8 nodes using up to 1024 CPU of the Marenostrum

¹This work consists of several chapters which content is based on the papers (that have been published or currently under revision in international journals) and proceedings of international conferences. Hence, all these chapters are written to be self-contained and only minor changes have been introduced with respect the originals. As a consequence, some contents such those describing the governing equations, the numerical algorithms or the description of test cases are repeated.

supercomputer of Barcelona supercomputing center. The set of DNS presented in this chapter covers configurations with high Rayleigh numbers 10^{10} , 3×10^{10} and 10^{11} , $Pr = 0.71$). Together with previous results [3] [4] it gives a relatively wide range of Ra -numbers from weak to fully developed turbulence. The main features of the flow, including the time-averaged flow structure, the turbulent statistics, the global kinetic energy balances and the internal waves motion phenomenon are described and discussed.

In the chapter 3, a way to extend the KSFD solver to fully 3D geometries is proposed. The use of FFT decomposition implies some limitations on geometry such as uniform mesh step on the FFT direction. For this reason KSFD solver is only applicable to so-called extruded geometries - 3D problems derived from 2D geometry by extrusion with uniform mesh step. This chapter is devoted to extend the previous algorithm to eliminate these limitations. The "fully-3D" algorithm presented is based on combination a two-level multigrid (MG) and the KSFD method as solver for second level. This way fully-3D geometry is approximated by geometry suitable for KSFD solver. A typical fully-3D DNS problem of high importance, a surface mounted cube in a channel flow, is considered as an example of solver application. Illustrative performance results presented.

Contents

| | |
|--|-----------|
| Acknowledgements | 7 |
| Abstract | 9 |
| 1 Krylov-Schur-Fourier Decomposition algorithm for high-order Poisson equation | 19 |
| 1.1 Introduction | 20 |
| 1.1.1 Motivation and summary of the present work | 22 |
| 1.2 Overview of the mathematical model | 23 |
| 1.2.1 Governing equations | 23 |
| 1.2.2 Time-integration method | 24 |
| 1.2.3 Spatial discretization | 26 |
| 1.2.4 Discretization of the Poisson equation | 26 |
| 1.3 Krylov-Schur-Fourier Decomposition algorithm | 28 |
| 1.3.1 Fourier Decomposition | 28 |
| 1.3.2 Overview of the Direct Schur Decomposition (DSD) algorithm | 30 |
| 1.3.3 Alternative: use of the DSD as a preconditioner for a Krylov-subspace method | 34 |
| 1.3.4 Properties of $\hat{\mathbf{A}}_i^{2D}$ that affect the convergence of the iterative method | 37 |
| 1.3.5 Residual criteria and initial guess | 42 |
| 1.4 Scalable KSFDF solver model | 46 |
| 1.4.1 Flexible solver configurations | 47 |
| 1.4.2 Illustrative parallel performance test | 48 |
| 1.4.3 Parallelisation in the periodic x -direction | 52 |
| 1.5 Conclusions | 55 |
| 2 Direct numerical simulation of a differentially heated 3D cavity on MareNostrum supercomputer | 57 |
| 2.1 Introduction | 58 |
| 2.1.1 Motivation and summary of the present work | 61 |
| 2.2 Governing equations and numerical method | 63 |
| 2.2.1 Governing equations | 63 |
| 2.2.2 Numerical method | 64 |
| 2.2.3 Code and simulation verifications | 67 |
| 2.3 Results and discussion | 69 |
| 2.3.1 Time-averaged flow | 69 |
| 2.3.2 Heat transfer | 75 |

| | | |
|----------|---|------------|
| 2.3.3 | Flow dynamics | 76 |
| 2.4 | Conclusions | 88 |
| 3 | Extension of the KSFD Poisson solver for fully-3D DNS of turbulent flows on supercomputers | 91 |
| 3.1 | Introduction | 92 |
| 3.2 | Numerical methods for DNS | 94 |
| 3.3 | Extension of the KSFD algorithm for fully-3D problems | 94 |
| 3.3.1 | Solution of the second level | 95 |
| 3.3.2 | Remapping between the two levels | 95 |
| 3.3.3 | Modification of the stencil for solid wall BC | 96 |
| 3.4 | Convergence and performance issues | 98 |
| 3.5 | Problem under consideration: surface mounted cube | 100 |
| 3.5.1 | Boundary conditions | 100 |
| 3.5.2 | Mesh construction | 101 |
| 3.6 | Set of stages performed to complete the DNS | 102 |
| 4 | Conclusions and future research | 107 |
| 4.1 | Concluding remarks | 107 |
| 4.2 | Future research | 108 |
| 4.2.1 | Implementation aspects | 108 |
| 4.2.2 | Numerical methods aspects | 108 |
| | Appendix A: Parallel CFD terminology | 109 |
| | Bibliography | 111 |
| | References | 111 |

List of Figures

| | | |
|------|--|----|
| 1.1 | Domain uncoupling on x -direction (periodic) by means of Fourier decomposition. | 29 |
| 1.2 | Partition and reordering of the unknowns for the fourth-order discrete Poisson equation. | 31 |
| 1.3 | Decomposition of one plane into 4 blocks. Structure of matrixes \mathbf{A}_D and \mathbf{A}_I | 36 |
| 1.4 | Top: comparison of convergence properties of the planes using second- and fourth-order discretizations. Middle: estimated convergence rate ω . Bottom: same plot for different number of planes right. | 43 |
| 1.5 | Iteration number for different initial divergence norm reduction ϵ (case B3D). | 46 |
| 1.6 | Solver configurations example. | 48 |
| 1.7 | Comparison of the solver configurations for different systems: JFF (left) and MareNostrum supercomputer (right). Computing time for each plane is represented using each of the four proposed configurations. This allows to obtain the optimised configuration choosing for each plane the best one. Results for $\xi = 0$, several orders of magnitude larger, have not been displayed. | 50 |
| 1.8 | Optimised configurations for different parallel systems | 51 |
| 1.9 | Speedup results | 52 |
| 1.10 | Speed-up results parallelising in the periodic x -direction. | 55 |
| 2.1 | Differentially heated cavity problem. | 62 |
| 2.2 | Numerical errors versus mesh size for convective (top) and divergence (bottom) operators for second-, fourth-, sixth- and eighth-order discretizations. | 68 |
| 2.3 | Averaged solutions. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} . Top: averaged temperature field. Bottom: streamlines of the averaged flow. For temperature fields, the isotherms are uniformly distributed from 0 to 1. | 71 |
| 2.4 | Averaged vertical temperature profiles at mid-width. | 72 |
| 2.5 | Dimensionless time-averaged vertical wall shear stress scaled by $Ra^{-1/4}$ | 73 |
| 2.6 | Averaged temperature (left-hand side of pair) and vertical velocity (right-hand side of pair) profiles at $z = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$. Each vertical subdivision represents 0.5 units for temperature and 0.2 units for vertical velocity. For the pair of plots on the left, the abscissa scale factor is $4y$ and for the pair on the right, is $4yRa^{1/4}$ | 74 |

| | | |
|------|--|-----|
| 2.7 | Local Nusselt number distribution (solid lines) and their standard deviation (dashed lines): (a) $Ra = 6.4 \times 10^8$, (b) $Ra = 2 \times 10^9$, (c) $Ra = 10^{10}$, (d) $Ra = 3 \times 10^{10}$ and (e) $Ra = 10^{11}$ | 77 |
| 2.8 | Representative instantaneous isotherms. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} . The isotherms are uniformly distributed from 0 to 1. | 79 |
| 2.9 | Representative instantaneous temperature iso-surfaces for highest Ra number 10^{11} . Time evolution of the flow to statistically stationary state (top), Zoom view of the turbulent transition area (bottom left), 2D section with zoom to transition area (bottom right) | 81 |
| 2.10 | Normalized density power spectra of Nusselt number at the vertical mid-plane: (a) $Ra = 6.4 \times 10^8$, (b) $Ra = 10^{10}$, (c) $Ra = 3 \times 10^{10}$ and (d) $Ra = 10^{11}$ | 82 |
| 2.11 | Distribution of turbulent kinetic energy k . From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} | 83 |
| 2.12 | Distribution of turbulent viscous dissipation ϵ_ν . From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} | 84 |
| 2.13 | Distribution of temperature variance $\overline{T'T'}$. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} | 84 |
| 2.14 | Distribution of the turbulent heat flux $\overline{u'_z T'}$. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} | 85 |
| 2.15 | Distribution of the Reynolds stress component $\overline{u'_y u'_y}$. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} | 85 |
| 2.16 | Distribution of the Reynolds stress component $\overline{u'_z u'_z}$. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} | 86 |
| 2.17 | Horizontal profiles at $z = 0.8$ of the turbulent kinetic energy k (solid line) and its viscous dissipation rate ϵ_ν (dashed line): (a) $Ra = 6.4 \times 10^8$, (b) $Ra = 2 \times 10^9$, (c) $Ra = 10^{10}$, (d) $Ra = 3 \times 10^{10}$ and (e) $Ra = 10^{11}$ | 87 |
| 3.1 | Surface mounted square cylinder in a channel - example of extruded geometry | 93 |
| 3.2 | 1D example of conservative interpolation. The bars represent values in the control volumes. | 96 |
| 3.3 | Stencil modification near the boundary | 97 |
| 3.4 | Scalability test: number of iterations with mesh growth | 99 |
| 3.5 | Surface mounted cube in a channel | 100 |
| 3.6 | Mesh overview | 102 |
| 3.7 | Choice of the cube position. Average vertical velocity profile in the middle of the channel. Inflow is placed where vertical velocity fades to zero | 104 |

3.8 Surface mounted cube in a channel. Instantaneous maps of kinetic energy averaged on periodic direction (top) and iso-surfaces (middle), pressure iso-surfaces with streamlines (bottom) 105

List of Tables

| | | |
|-----|--|----|
| 1.1 | Physical and numerical parameters of the tested cases. | 45 |
| 2.1 | Physical and numerical simulation parameters. | 69 |
| 2.2 | Summary of the averaged flow results. By rows, from top to bottom, the magnitudes are: the overall averaged Nusselt number, the maxima of the averaged local Nusselt and the standard deviation of local Nusselt and their respective z positions at the vertical hot wall, the maxima of $\overline{u_z}$ and \overline{T} at the horizontal mid-height plane and their respective y positions, the maxima of $\overline{u_y}$ and \overline{T} at the vertical mid-width plane and their respective z positions. | 70 |
| 2.3 | Nusselt number and correlations | 75 |
| 2.4 | From left to right: fundamental frequencies of the Nu_c , dimensionless stratification in the core of the cavity and Brunt-Väisälä frequency . . | 80 |
| 2.5 | Global kinetic energy balances | 88 |
| 3.1 | Typical DNS cases | 92 |

Chapter 1

Krylov-Schur-Fourier Decomposition algorithm for high-order Poisson equation

Main contents of this chapter have been published in:

A. Gorobets, F. X. Trias, M. Soria and A. Oliva, A scalable Krylov-Schur-Fourier Decomposition for the efficient solution of high-order Poisson equation on parallel systems from small clusters to supercomputers, *Computers and Fluids*, (submitted).

A. Gorobets, Scalable algorithm for incompressible flow simulation on parallel computer systems, *Mathematical modelling*, vol. 19, #10, 105-128, 2007.

Abstract.

A parallel algorithm for direct numerical simulation (DNS) of incompressible turbulent flows that provides a fairly good scalability for a wide range of parallel computer architectures has been developed. Since the time integration is fully explicit, from a parallel point of view, the main bottleneck is the Poisson equation. In the previous works a Direct Schur-Fourier Decomposition (DSFD) algorithm [5, 2] conceived for low-cost PC clusters with poor network performance, was proposed. Such method, that is very efficient for PC clusters, can not be used with relatively large number of processors and mesh sizes, mainly due to the RAM memory requirements [2]. The new scalable parallel Poisson solver, named Krylov-Schur-Fourier Decomposition (KSFD), is presented in this chapter. It is based on fast Fourier decomposition method (FFT) with Krylov method of conjugate gradients (CG) preconditioned with the Direct Schur Decomposition (DSD) algorithm [1, 2]. FFT provides uncoupling of the original 3D problem into a set of independent 2D problems, then each of them is solved using CG preconditioned with DSD method. Numerical experiments showing the scalability and the flexibility of the method on both the MareNostrum supercomputer and a PC cluster with a conventional 100 Mbits/s network are presented and discussed.

1.1 Introduction

Many important applications in the computational fluid dynamics (CFD) field, such as direct numerical simulation (DNS) or large-eddy simulation (LES) of transition and turbulent flows demand huge computing power and need parallel computers to be feasible. The Poisson equation, which arises from the incompressibility constraint and has to be solved at least once at each time step, is usually the main bottleneck from a parallel point of view. In this context, efficient and scalable algorithms for the solution of the Poisson equation on a wide range of parallel systems are of high interest.

Structured Cartesian discretization in a rectangular domain is considered. The Poisson equation is to be solved at least once on each time step even using explicit algorithms and this equation is the main bottleneck for the parallel simulations of the turbulent incompressible flows. The poisson operator has infinite speed of propagation of information in the spatial domain. This tight coupling of computational domain is a serious obstacles for the parallelization. For this reason efficient solution of the Poisson equation on parallel systems is of first priority.

Roughly speaking, contemporary parallel systems vary from cheap small loosely-coupled PC clusters made of standard office computer equipment and conventional networks to huge supercomputers with thousands of tightly coupled CPUs. First group has very good CPU-performance/cost ratio but, in general, the total number of CPUs is relatively small and the network performance is relatively poor (high latency and low bandwidth). Therefore, on such systems the size and number of the messages exchanged must be minimized in order to increase the efficiency of the algorithms. In contrast, in supercomputers the number of CPUs is much bigger and network performance much better (specially in terms of latency). It is important to note that depending on the computer architecture, the number of processors and the scale of the problem being solved, these considerations may become even more important than the arithmetical complexity of the algorithm [2].

Therefore, for the aforementioned reasons, algorithms that work well on supercomputers may not work efficiently on PC clusters due to the poor network performance. For instance, Krylov-subspace methods demand several communication episodes (for matrix-vector products, scalar products and pre-conditioning) that can lead to poor efficiency on loosely coupled parallel systems. And vice-versa, algorithms that are designed to work well on PC clusters, such as Schur complement-based methods [5, 2, 1, 6, 7, 8], can suffer strong limitations of scalability to bigger number of CPUs [2].

Another example, the Multigrid method which is one of the most powerful methods for sequential computations requires a lot of communication episodes when iterating

on coarser levels. It is rather difficult (if possible) to apply multigrid efficiently to solve Poisson equation on parallel systems with rather big CPU number or with high-latency network. Some implementations of multigrid has better scalability but it is still far from hundreds and thousands of CPU. Details about the problems of parallel multigrid solver implementations can be found in [9].

FFT methods [10] applied for more that one direction can provide very low computational cost $O(N \log N)$ but it also have specific limitations, in particular, FFT requires uniform mesh step. Chebyshev based methods [11] allow to use non-uniform mesh step but it also requires specific mesh step distribution. Placing obstacles in a flow it is another problem because presence of obstacle must not affect the Poisson matrix otherwise these methods cannot be applied.

For a more detailed review of the existing parallel solvers and their scalability limitations the reader is referred to [5, 2].

Moreover, most of the existing parallel Poisson solvers do not properly combine the flexibility required for the wide variety of parallel systems and CFD applications to be run on.

This chapter is focused on the development of a parallel Poisson solver scalable and flexible enough to run efficiently on both PC clusters and supercomputers. In the context of DNS and LES time-accurate incompressible flow simulations, the scalability of the solver is of high importance as it allows to deal with large size problem (like hundreds of millions of unknowns and more) and exploit efficiently the computing power of moderns supercomputers.

Furthermore, the following four aspects, which are also relevant in the context of this paper, are commonly present in these applications:

- The Poisson equation has to be solved repeatedly¹, with different right-hand side terms, while the system matrix remains constant. Hence, a pre-processing stage with large computing demands can be accepted.
- Wall-bounded flows or/and flows around internal obstacles are common in most of the applications [12, 13, 4, 3, 14]. Therefore, in order to solve all relevant turbulent scales near the walls arbitrary non-uniform discretizations are required.
- The solution obtained in the previous time step(s) can be used as an initial guess for iterative solvers in order to accelerate the convergence. This becomes specially relevant if an explicit time advancement method is being used.
- Periodicity in at least one direction is usually of interest for many LES and DNS applications.

¹The number of time-steps required to solve our benchmark problems from initial conditions is $\propto 10^6$.

The last property makes the Fourier diagonalization [15] in the periodic direction(s) the best choice. The uniformity of the grid in each such direction imposed by the method is suitable with the isotropic nature of the flow in the periodic direction(s). Fourier diagonalization allows to decompose the original three-dimensional (3D) Poisson equation into a family of independent two-dimensional (2D) systems of equations.

The previous work was oriented on low-cost PC clusters with poor network performance. In that context, the Schur complement method was a good choice for the parallel solution of this set of 2D problems because systems arising from non-uniform meshes can be solved without increasing the computational costs. Moreover, the Schur complement matrix only needs to be computed in a pre-processing stage and it allows to solve all the planes to machine accuracy using only one all-to-all communication episode [1]. Furthermore, since the parallelisation of FFT is not efficient on loosely coupled parallel computers parallelisation in the periodic direction was discarded. Such parallel Poisson solver, named Direct Schur-Fourier Decomposition (DSFD) algorithm [5, 2], have been successfully used to perform DNS simulations [4, 3] of turbulent natural convection flows in enclosed cavities.

1.1.1 Motivation and summary of the present work

However, the DSFD algorithm, that is very efficient on PC clusters, can not be used for an arbitrarily large number of processors and problem size, mainly due to the RAM memory requirements [2] and the size of communications that grow fast with the number of processors and mesh size. These problems limit the DSFD solver scalability specially for high-order numerical schemes making it not applicable for large-scale problems on supercomputers using hundreds (or thousands) of processors.

To do so, a new version of the solver, named Krylov-Schur-Fourier Decomposition (KSFD), is presented. Again, a Fourier decomposition is used to uncouple the original 3D Poisson equation into a set of 2D planes. Then, each 2D problem is solved using a Conjugated Gradient (CG) method [16] preconditioned by a Direct Schur Decomposition (DSD) algorithm [1]. To do that, each plane is decomposed into *blocks* and each of them is solved with the DSD or a local solver depending on size.

Therefore, one of the main features of the algorithm is its flexibility: the number of blocks governs the convergence properties of the algorithm. The smaller is the number of blocks, the CG algorithm is better preconditioned and consequently the number of iterations reduced. However, bigger blocks implies more RAM memory and computational costs. Hence, the block size of each plane is chosen in order to minimize the global computational cost accomplishing with the memory limitations and communication time imposed by our computer architecture.

Convergence properties of the 2D problems are considered from both analytical and numerical point of views. Based on these results, a flexible solver model governed by the number of blocks (solved using a direct DSD method or a local LU method depending on a block size) is proposed and tested. In addition, the problem of the choice of the residual criteria and the initial guess are also discussed and analyzed.

The numerical tests presented in this paper have been carried out on a PC cluster with 40 standard PCs running Linux and a conventional 100 Mbits/s Ethernet network and on the IBM MareNostrum supercomputer at the Barcelona Supercomputing Center. Performance tests for meshes of up to $\approx 111 \times 10^6$ nodes using up to 1024 processors illustrating the robustness and scalability of the method are presented and discussed.

The rest of the chapter is organized as follows. In section 1.2, the numerical methods for space and time discretization are briefly exposed. The KSFD algorithm is described in section 1.3. In section 1.4, a scalable solver model for KSFD is introduced and tested for a real application on the PC cluster and on the MareNostrum supercomputer. Finally, conclusions are given in section 1.5.

1.2 Overview of the mathematical model

1.2.1 Governing equations

The non-dimensional incompressible Navier-Stokes (NS) equations coupled with the thermal transport equation in a bounded parallelepipedic domain of height L_z , width L_y and depth L_x are considered

$$\nabla \cdot \mathbf{u} = 0 \quad (1.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = Pr \nabla^2 \mathbf{u} - \nabla p + \mathbf{f} \quad (1.2)$$

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \nabla^2 T \quad (1.3)$$

where Pr is the Prandtl number and \mathbf{f} is an externally applied body force (e.g. gravity). Periodic boundary conditions are prescribed in the x -direction,

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x} + L_x \mathbf{e}_x, t) \quad (1.4)$$

$$T(\mathbf{x}, t) = T(\mathbf{x} + L_x \mathbf{e}_x, t) \quad (1.5)$$

because it allows to study the 3D effects due to intrinsic instability of the main flow and not due to the boundary conditions [4, 3].

1.2.2 Time-integration method

In order to simplify the notation, momentum equation (1.2) can be rewritten as

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{R}(\mathbf{u}) - \nabla p$$

where $\mathbf{R}(\mathbf{u})$ represents the right-hand-side terms of the momentum equation except for the pressure gradient,

$$\mathbf{R}(\mathbf{u}) \equiv Pr \nabla^2 \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{u} + \mathbf{f} \quad (1.6)$$

For the temporal discretization, a central difference scheme is used for the time derivative term, a fully explicit second-order one-leg scheme [12] for $\mathbf{R}(\mathbf{u})$ and a first-order backward Euler scheme for the pressure-gradient term. Incompressibility constraint is treated implicit. Thus, we obtain the semi-discretized NS equations

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (1.7)$$

$$\frac{(\beta + 1/2) \mathbf{u}^{n+1} - 2\beta \mathbf{u}^n + (\beta - 1/2) \mathbf{u}^{n-1}}{\Delta t} = \mathbf{R}((1 + \beta) \mathbf{u}^n - \beta \mathbf{u}^{n-1}) - \nabla p^{n+1} \quad (1.8)$$

where the parameter β is computed each time-step to adapt the linear stability domain of the time-integration scheme to the instantaneous flow conditions in order to use the maximum Δt possible. For further details about the time-integration method the reader is referred to [3].

To solve the velocity-pressure coupling a classical fractional step projection method [17, 18] is used. In the projection methods, solutions of the unsteady NS equations are obtained by first time-advancing the velocity field \mathbf{u} without regard for its solenoidality constraint (1.7), then recovering the proper solenoidal velocity field, \mathbf{u}^{n+1} ($\nabla \cdot \mathbf{u}^{n+1} = 0$). This projection is derived from the Helmholtz-Hodge vector decomposition theorem [19], whereby the velocity \mathbf{u}^{n+1} can be uniquely decomposed into a solenoidal vector, \mathbf{u}^p , and a curl-free vector, expressed as the gradient of a scalar field, $\nabla \tilde{p}$. This decomposition is written as

$$\mathbf{u}^p = \mathbf{u}^{n+1} + \nabla \tilde{p} \quad (1.9)$$

where the predictor velocity \mathbf{u}^p is

$$\mathbf{u}^p = \frac{2\beta\mathbf{u}^n - (\beta - 1/2)\mathbf{u}^{n-1}}{\beta + 1/2} + \frac{\Delta t}{\beta + 1/2} \mathbf{R}((1 + \beta)\mathbf{u}^n - \beta\mathbf{u}^{n-1}) \quad (1.10)$$

and the pseudo-pressure is $\tilde{p} = \Delta t / (\beta + 1/2) p^{n+1}$. Taking the divergence of (1.9) yields a Poisson equation for \tilde{p}

$$\nabla \cdot \mathbf{u}^p = \nabla \cdot \mathbf{u}^{n+1} + \nabla \cdot (\nabla \tilde{p}) \longrightarrow \nabla^2 \tilde{p} = \nabla \cdot \mathbf{u}^p \quad (1.11)$$

The question of what boundary condition to use for the pressure equation (1.11) in the non-periodic directions has led to much discussion. The main ideas were shortly reviewed in [20]. The use of the normal component of the momentum equation is commonly accepted as the most appropriate boundary condition, see for example [21]. However, at the discrete level on staggered grids with prescribed velocity boundary conditions, as in our case, the incompressibility condition occurs naturally and no specific boundary condition for the pressure needs to be specified as pointed out in [22].

Finally, once the solution is obtained, \mathbf{u}^{n+1} results from the correction

$$\mathbf{u}^{n+1} = \mathbf{u}^p - \nabla \tilde{p} \quad (1.12)$$

Therefore, the algorithm for the integration of each time-step is

1. Evaluate $\mathbf{R}((1 + \beta)\mathbf{u}^n - \beta\mathbf{u}^{n-1})$.
2. Evaluate the predictor velocity \mathbf{u}^p from Eq.(1.10).
3. Evaluate $\nabla \cdot \mathbf{u}^p$ and solve the discrete Poisson (2.12) equation.
4. Obtain the new velocity field with Eq.(1.12).

It is well-known that due to stability reasons explicit temporal schemes introduce severe restrictions on the time-step, while implicit discretization would improve the overall stability. However, for the use of implicit methods in DNS of turbulent flows the computational costs are rather high compared to those of explicit methods. This is because of the underlying restrictions to time-step that are required to fully resolve all temporal scales in the NS equations. Therefore, we have only considered explicit methods in the view of the lower costs.

1.2.3 Spatial discretization

Governing equations (1.1-1.3) are discretized on a staggered grid in space by second- or fourth-order symmetry-preserving discretizations [12]. Following the same notation, the symmetry-preserving discretization of the Navier-Stokes equations becomes

$$\mathbf{M}\mathbf{u}_h = \mathbf{0} \tag{1.13}$$

$$\Omega \frac{d\mathbf{u}_h}{dt} = -\mathbf{C}(\mathbf{u}_h)\mathbf{u}_h + \mathbf{D}\mathbf{u}_h + \mathbf{f}_h - \mathbf{M}^t \mathbf{p}_h \tag{1.14}$$

where \mathbf{u}_h stands for the discrete velocity vector, Ω is a positive-definite diagonal matrix representing the sizes of the control volumes, the convective coefficient matrix $\mathbf{C}(\mathbf{u}_h)$ is skew-symmetric, the discrete diffusive operator \mathbf{D} is a symmetric negative-definite matrix and \mathbf{M} is the discrete divergence operator. The discrete gradient operator is the transpose of the discrete divergence multiplied by a diagonal scaling $\mathbf{G} = -\Omega^{-1}\mathbf{M}^t$.

The main feature of such discretization is that it preserves the underlying symmetry properties of the continuous differential operators. These global discrete operator properties ensure both stability and that the global kinetic-energy balance is exactly satisfied even for coarse meshes if incompressibility constraint is accomplished [12]. Energy transport equation is also discretized using a symmetry-preserving discretization.

It must be noted that periodic boundary conditions are prescribed in the x -direction because it allows to study the 3D effects due to intrinsic instability of the main flow and not due to the boundary conditions [4, 3]. This case is also more convenient from a computational point of view because the resulting flows are free from boundary layers on the x -direction and therefore, the mesh can be coarser and *uniform* in this direction. This allows to apply Fourier-based methods to solve the Poisson equation. See section 1.3.1 for further details.

1.2.4 Discretization of the Poisson equation

Let us briefly outline those aspects concerning the discretization of the Poisson equation (2.12) that are relevant in our context. The discrete Laplacian operator of the Poisson equation can be viewed as the product of the discrete divergence operator \mathbf{M} by the discrete gradient operator, which is the transpose of the discrete divergence multiplied by a diagonal scaling $\mathbf{G} = -\Omega^{-1}\mathbf{M}^t$. So, the Laplacian operator is approximated by the matrix product

$$\mathbf{L} = -\mathbf{M}\Omega^{-1}\mathbf{M}^t \tag{1.15}$$

where $\mathbf{A}_{j,k}^{n_r}$, $\mathbf{A}_{j,k}^{s_r}$, $\mathbf{A}_{j,k}^{t_r}$ and $\mathbf{A}_{j,k}^{b_r}$ are $N_x \times N_x$ diagonal matrices

$$\mathbf{A}_{j,k}^{n_r} = a_{j,k}^{n_r} \mathbf{I} \in \mathbb{R}^{N_x \times N_x} \quad (1.21)$$

and $\mathbf{A}_{j,k}^p$ are $N_x \times N_x$ symmetric real-valued circulant [23] matrices. For further details about the discretization of the Poisson equation the reader is referred to [5, 2].

1.3 Krylov-Schur-Fourier Decomposition algorithm

1.3.1 Fourier Decomposition

Diagonalization problem of generic symmetric real-valued circulant matrices and their correspondence with the discrete Fourier transforms (DFT) was considered in detail in our previous work [2]. Following the same notation we define $\mathbf{Q}_{\mathbb{R}} \in \mathbb{R}^{N_x \times N_x}$ as an inverse Fourier transform matrix for real-valued problems

$$z = \mathbf{Q}_{\mathbb{R}} \hat{z} \quad (1.22)$$

and consequently $\mathbf{Q}_{\mathbb{R}}^{-1}$ as the forward DFT matrix.

Furthermore, $\mathbf{A}_{j,k}^p$ are symmetric real-valued circulant [23] matrices of the form³

$$\mathbf{A}_{j,k}^p \equiv \begin{pmatrix} a_{j,k}^p & a_{j,k}^{e_1} & a_{j,k}^{e_2} & \cdots & a_{j,k}^{e_1} \\ a_{j,k}^{e_1} & a_{j,k}^p & a_{j,k}^{e_1} & \cdots & a_{j,k}^{e_2} \\ a_{j,k}^{e_2} & a_{j,k}^{e_1} & a_{j,k}^p & \cdots & a_{j,k}^{e_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{j,k}^{e_1} & a_{j,k}^{e_2} & a_{j,k}^{e_3} & \cdots & a_{j,k}^p \end{pmatrix} \in \mathbb{R}^{N_x \times N_x} \quad (1.23)$$

therefore, they can be diagonalized by means of a DFT algorithm,

$$\mathbf{Q}_{\mathbb{R}}^{-1} \mathbf{A}_{j,k}^p \mathbf{Q}_{\mathbb{R}} = \Lambda_{j,k} \quad (1.24)$$

where $\Lambda_{j,k} \in \mathbb{R}^{N_x \times N_x}$ is a diagonal matrix with the eigenvalues of $\mathbf{A}_{j,k}^p$ given by

$$(\lambda_i)_{j,k} = a_{j,k}^p + 2 \sum_{r=1}^{N_x/2-1} a_{j,k}^{e_r} \cos(r\alpha_i) \quad (1.25)$$

where $\alpha_i = [2\pi(i-1)]/N_x$ (for further details the reader is referred to [2]).

This allows to decompose the original block diagonal system (1.17) into a set of N_x block diagonal equations. To do so, each of the block diagonal equations

³The number of non-zero coefficients $a_{j,k}^{e_r}$ depends on the stencil size of the numerical scheme. However, we have adopted this notation to keep the method as general as possible.

is pre-multiplied by $\mathbf{Q}_{\mathbb{R}}^{-1}$ and the sub-vectors $\mathbf{x}_{j,k}$ are expressed as $\mathbf{Q}_{\mathbb{R}} \hat{\mathbf{x}}_{j,k}$. After these operations, the $\mathbf{A}_{j,k}^p$ matrices become diagonal while the $\mathbf{A}_{j,k}^{nb}$ matrices, that are equal to the identity matrix multiplied by a scalar (1.21), are not affected (e.g. $\mathbf{Q}_{\mathbb{R}}^{-1} a_{j,k}^n \mathbf{I} \mathbf{Q}_{\mathbb{R}} = a_{j,k}^n \mathbf{I}$).

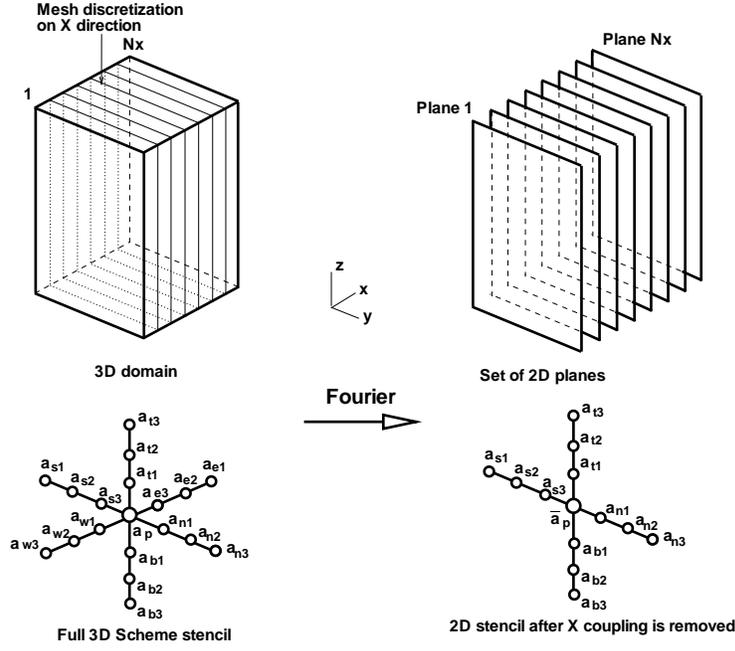


Figure 1.1: Domain uncoupling on x -direction (periodic) by means of Fourier decomposition.

In conclusion, the Fourier diagonalization eliminates all the non-diagonal entries of matrices $\mathbf{A}_{j,k}^p$ (see figure 1.1). Thus, the original system (1.17) is now decoupled into a set of N_x systems where the unknowns $\hat{x}_{i,j,k}$ are only coupled with unknowns in the same plane i ,

$$\hat{\mathbf{A}}_i^{2D} \hat{\mathbf{x}}_i^{2D} = \hat{\mathbf{b}}_i^{2D}, \quad i = 1, \dots, N_x \quad (1.26)$$

where $\hat{\mathbf{A}}_i^{2D} \in \mathbb{R}^{N_{yz} \times N_{yz}}$ is a block diagonal matrix associated with the transformed equation for plane i . Further details of the method can be found in [5, 2]. In summary, the operations to be performed to solve the original system are

1. Calculate the N_x transformed right-hand-side sub-vectors,

$$\hat{\mathbf{b}}_{j,k} = \mathbf{Q}_{\mathbb{R}}^{-1} \mathbf{b}_{j,k}.$$

2. Solve the N_x decoupled block diagonal equations systems $\hat{\mathbf{A}}_i \hat{\mathbf{x}}_i = \hat{\mathbf{b}}_i$.
3. Carry out the anti-transformation of the N_x solution sub-vectors $\mathbf{x}_{j,k} = \mathbf{Q}_{\mathbb{R}} \hat{\mathbf{x}}_{j,k}$

Matrix-vector product requires $\mathcal{O}(N^2)$ operations, so for both the inverse and forward DFT (steps 1 and 3), the cost would be $\mathcal{O}(N_x^2 N_y N_z)$. However, in our case they can be carried out using a FFT algorithm that is only $\mathcal{O}(N_x (\log_2 N_x))$ (see [24], for instance). Therefore, their cost grows only with $\mathcal{O}(N_x (\log_2 N_x) N_y N_z)$ and it is small compared with the solution of the N_x decoupled linear systems. Moreover, note that the optimised FFT for real-valued problems outlined in our previous work [2] allows us to half the computational cost compared with the straightforward implementation.

Therefore, once the FFT algorithm has been applied, we must focus on the efficient solution of the set of decoupled block diagonal systems (1.26). This will be addressed in the following sections.

1.3.2 Overview of the Direct Schur Decomposition (DSD) algorithm

In this section the Direct Schur Decomposition [1] (DSD) algorithm is briefly described. This parallel method, that allows to solve each bidimensional decoupled problem (1.26) to almost machine accuracy, is based on the fact that the matrix of the coefficients remains constant during all the fluid flow simulation. This allows us to evaluate and store the inverse of the interface matrix of each equation in a pre-processing stage. The combination of the Fourier diagonalization, outlined in the previous section, and the DSD algorithm as a direct solver for each plane was named Direct Schur-Fourier Decomposition (DSFD) algorithm [5, 2]. Then, in the solution stage all the systems are solved together. Thus, only one all-to-all communication episode is needed to solve each 3D Poisson equation (1.17) to machine accuracy. This method allowed us to perform DNS simulations of buoyancy driven flows on a low-cost PC cluster [4, 3]. For further details about the DSFD algorithm the reader is referred to our previous works [5, 2, 1].

Since all the planes are fully decoupled after the Fourier decomposition, in this section only the solution of one arbitrary 2D problem is considered. To simplify the notation, the hats and sub-indices are dropped and each N_x block diagonal equations (1.26) to be solved is simply denoted as

$$\mathbf{Ax} = \mathbf{b} \tag{1.27}$$

where $\mathbf{A} \in \mathbb{R}^{N_{yz} \times N_{yz}}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^{N_{yz}}$.

The unknowns of vector \mathbf{x} are partitioned into a family of P subsets, called inner domains, that are labelled from 0 to $P-1$, plus one interface (see figure 1.2), labelled s . Inner nodes are those that are not directly coupled with other inner subsets.

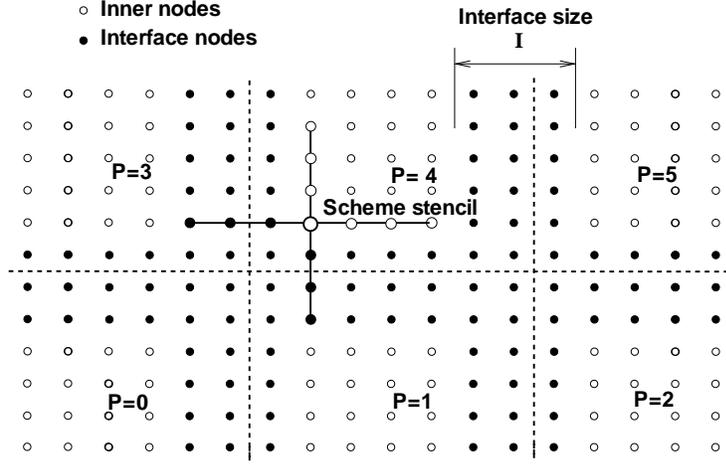


Figure 1.2: Partition and reordering of the unknowns for the fourth-order discrete Poisson equation.

Therefore, all the nodes are divided into $P+1$ subsets and renumbered according to the partition chosen. Thus, with this partition the system (1.27) can be expressed using block matrices as follows

$$\begin{pmatrix} \mathbf{A}_{0,0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{0,s} \\ \mathbf{0} & \mathbf{A}_{1,1} & \cdots & \mathbf{0} & \mathbf{A}_{1,s} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{P-1,P-1} & \mathbf{A}_{P-1,s} \\ \mathbf{A}_{s,0} & \mathbf{A}_{s,1} & \cdots & \mathbf{A}_{s,P-1} & \mathbf{A}_{s,s} \end{pmatrix} \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{P-1} \\ \mathbf{x}_s \end{pmatrix} = \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{P-1} \\ \mathbf{b}_s \end{pmatrix} \quad (1.28)$$

The interface unknowns are isolated by means of block Gaussian elimination. Then, system (1.28) results

$$\begin{pmatrix} \mathbf{A}_{0,0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{0,s} \\ \mathbf{0} & \mathbf{A}_{1,1} & \cdots & \mathbf{0} & \mathbf{A}_{1,s} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{P-1,P-1} & \mathbf{A}_{P-1,s} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{A}}_{s,s} \end{pmatrix} \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{P-1} \\ \mathbf{x}_s \end{pmatrix} = \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{P-1} \\ \tilde{\mathbf{b}}_s \end{pmatrix} \quad (1.29)$$

The last block equation, involving only unknowns in $\mathbf{x}_s \in \mathbb{R}^{N_s}$, is the interface equation

$$\tilde{\mathbf{A}}_{s,s} \mathbf{x}_s = \tilde{\mathbf{b}}_s \quad (1.30)$$

with the modified right-hand side

$$\tilde{\mathbf{b}}_s = \mathbf{b}_s - \sum_{p=0}^{P-1} \mathbf{A}_{s,p} \mathbf{A}_{p,p}^{-1} \mathbf{b}_p \quad (1.31)$$

and the Schur complement matrix given by

$$\tilde{\mathbf{A}}_{s,s} = \mathbf{A}_{s,s} - \sum_{p=0}^{P-1} \mathbf{A}_{s,p} \mathbf{A}_{p,p}^{-1} \mathbf{A}_{p,s} \quad (1.32)$$

Hence, interface equation (1.30) is solved before the inner domain equations. Once \mathbf{x}_s is known, each of the \mathbf{x}_p can be obtained independently by its own p , solving its original equation

$$\mathbf{A}_{p,p} \mathbf{x}_p = \mathbf{b}_p - \mathbf{A}_{p,s} \mathbf{x}_s \quad (1.33)$$

Finally, note that $\tilde{\mathbf{A}}_{s,s} \in \mathbb{R}^{N_s \times N_s}$ is a dense matrix. Both $\tilde{\mathbf{A}}_{s,s}$ and $\tilde{\mathbf{b}}_s$ can be evaluated without the explicit calculation of $\mathbf{A}_{p,p}^{-1}$, as described in [1].

The solution of the interface equation (1.30) is critical for the efficiency of the Schur decomposition algorithm. For this reason the inverse of the matrix $\tilde{\mathbf{A}}_{s,s}^{-1}$ is obtained at the preprocessing stage to solve the equation explicitly:

$$\mathbf{x}_s = \tilde{\mathbf{A}}_{s,s}^{-1} \tilde{\mathbf{b}}_s \quad (1.34)$$

A parallel block LU decomposition of dense matrices, based on the distributed evaluation and storage of the inverse of the Schur complement matrix, is used (see references [5, 2, 1] for details). In summary, the DSD algorithm for the direct solution of Eq.(1.27) is as follows:

1. Solve $\mathbf{A}_{p,p}\mathbf{t} = \mathbf{b}_p$
2. Evaluate the local contribution to the *r.h.s.* of the interface equations, $\tilde{\mathbf{b}}_s^p = \mathbf{A}_{s,p}\mathbf{t}$
3. Carry out the global summation $\mathbf{t} = \sum_{p=0}^{P-1} \tilde{\mathbf{b}}_s^p$
4. Evaluate the *r.h.s.* of the interface equations, $\tilde{\mathbf{b}}_s = \mathbf{b}_s - \mathbf{t}$
5. Evaluate the interface nodes $\mathbf{x}_s = \tilde{\mathbf{A}}_{s,s}^{-1}\tilde{\mathbf{b}}_s$ (where needed)
6. Solve locally the inner nodes from $\mathbf{A}_{p,p}\mathbf{x}_p = \mathbf{b}_p - \mathbf{A}_{p,s}\mathbf{x}_s$

Note that vector \mathbf{t} represents a temporary storage area. Only a *global reduction communication episode* is needed on the stage 3.

Limitations of the DSD algorithm

Although the DSFD algorithm has been successfully used to perform DNS simulations on a low-cost PC cluster using up to 40 processors [4, 3] several scalability problems arise with the growth of the number of CPUs and the mesh size [2].

The first and most important problem is related to the RAM memory requirements. Under certain simplifications (for a detailed deduction, see [2]) the total number of array positions that each processor needs is approximately

$$N_t^{2d} \approx 3M \left(\frac{N_{yz}}{P} \right)^{3/2} + 4I^2 N_{yz} \frac{P^{1/2} - 1}{P^{1/2}} \quad (1.35)$$

where P is the number of processors, $N_{yz} = N_y \times N_z$ is the number of unknowns in each plane, M is the scheme stencil size and I is the interface size (see figure 1.2). Note that the main difficulty arises from the interface equation (second term of *r.h.s.* of Eq. 1.35) whose RAM memory requirements grows quadratically with I and remains almost independent of the number of processors P . Thus, the use of the DSD algorithm on a parallel computer with any number of CPUs is limited by the RAM memory of one single CPU node. Moreover, since M and I are increasing functions of the order of the scheme o^4 , RAM memory requirements may also become a serious obstacle to use DSD for high-order schemes.

⁴For the family of discretizations here considered:

$$M = o - 1 \quad (1.36)$$

$$I = o - 1 \quad (1.37)$$

Apart from the memory problems, there are other issues that may prevent the efficient use of DSD solver for large-scale problems. Firstly, DSD method needs at least one big collective communication for global reduction operation for the summation of the interface nodes (stage 3 of the algorithm). The data from all the interface nodes must be spread between all the CPUs. Under certain approximations, the size of data transmission can be estimated as

$$CS = \mathcal{O}\left(N_x \log_2(P) \sqrt{N_{yz}P}\right) \quad (1.38)$$

The previous expression shows that data exchanged for each CPU grows with both the number of processors and the number of unknowns. When the number of CPUs is relatively small this communication is not specially time consuming, but with the growth of the number of processors, and depending on the computer architecture, it may affect negatively the scalability of the method.

Secondly, computational cost per CPU of the matrix-vector product $x_s = \tilde{\mathbf{A}}_{s,s}^{-1} \tilde{\mathbf{b}}_s$ (step 5 of the DSD algorithm) demands $\mathcal{O}(N)$ floating point operations. Thus, the cost only depends on the problem size not the numbers of processors. Although for small- and medium-size meshes this is not really a serious problem, for really large problems this stage may eventually limit the scalability of the algorithm.

Finally, the computational cost related with the distributed evaluation of the inverse of the Schur complement matrix, $\tilde{\mathbf{A}}_{s,s}$, grows fast with P and N_{yz} . Thus, for large meshes that demand a large number of CPUs, the cost of the pre-processing stage can become not negligible at all with respect the cost of the DNS simulation itself.

It must be noted that the weight of all these scalability limitations would depend on the mesh size, the number of CPUs, the computer architecture and the total time-integration period demanded for our applications⁵. In conclusion, all these scalability limitations motivate the necessity to investigate a different approach for really large-scale problems.

1.3.3 Alternative: use of the DSD as a preconditioner for a Krylov-subspace method

If due to the aforementioned scalability limitations the DSD method cannot be applied efficiently (or cannot be applied at all) to couple all the planes then it can be used to couple only some planes of the set (1.26). In this case, the rest of planes can be solved

⁵Based on our experience, for real CFD applications the use of the DSFD algorithm is feasible for problems with mesh sizes up to $\sim 6 \times 10^6$ points and 30 \sim 50 processors

using an iterative method. Moreover, DSD itself can be used as an auxiliary direct solver or preconditioner inside of the iterative method. In the later case, the DSD algorithm is used to couple some parts of the plane and the iterative method provides coupling of the whole plane. Henceforth, we will refer those parts of the plane that are coupled with the DSD solver as *blocks*. For a detailed survey about the existing preconditioning techniques for sparse linear systems the reader is referred to [25].

Algorithm of the preconditioned CG method

Each plane corresponds to the system of equations in the set (1.26). Once again, since all planes are fully decoupled, only the solution of one arbitrary plane is considered. Thus, the system to be solved is⁶

$$\mathbf{Ax} = \mathbf{b} \quad (1.39)$$

Since the matrices \mathbf{A} are, by construction (1.15), a symmetric positive definite non-singular⁷ matrix (see section 1.3.4 for details) the Conjugate Gradient (CG) method [16] is chosen as the most appropriate. The plane is decomposed into blocks and each block is coupled with the DSD solver. Thus, the matrix \mathbf{A} is splitted as follows

$$\mathbf{A} = \mathbf{A}_D + \mathbf{A}_I \quad (1.40)$$

where matrix \mathbf{A}_D results from matrix \mathbf{A} after discarding elements that correspond to the coupling between different blocks. An example of the decomposition of one plane into 4 blocks is displayed in figure 1.3.

Then, iterative CG solver uses DSD algorithm as a direct solver for each block, $\mathbf{DSD}(\mathbf{A}_D, \mathbf{b})$, as a preconditioner which provides exact solution for the system $\mathbf{A}_D \mathbf{x} = \mathbf{b}$. Therefore, the preconditioned CG algorithm results

First iteration:

1. $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0$
2. $\mathbf{z}^0 = \mathbf{DSD}(\mathbf{A}_D, \mathbf{r}^0)$
3. $\mathbf{p}^1 = \mathbf{z}^0$

⁶Indexes are dropped for the sake of clarity.

⁷Originally there is one plane whose corresponding \mathbf{A} matrix is singular. In this case, singularity is removed preserving the symmetry by changing one element on the main diagonal that corresponds to an arbitrary inner node. This modification fixes the value of this particular node to zero.

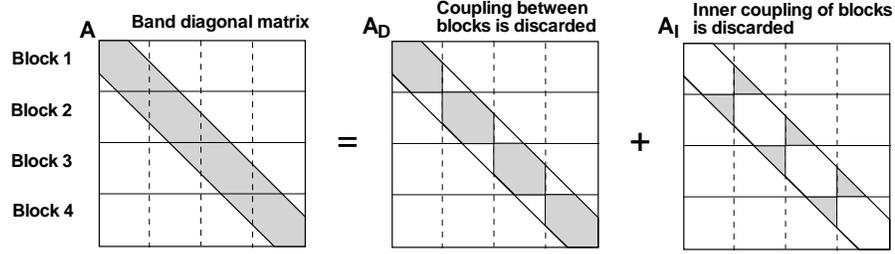


Figure 1.3: Decomposition of one plane into 4 blocks. Structure of matrixes \mathbf{A}_D and \mathbf{A}_I

i -th iteration:

1. $\mathbf{z}^{i-1} = \text{DSD}(\mathbf{A}_D, \mathbf{r}^{i-1})$
2. * $\rho_{i-1} = \mathbf{r}^{i-1T} \cdot \mathbf{z}^{i-1}$
3. $\beta_{i-1} = \rho_{i-1} / \rho_{i-2}$
4. $\mathbf{p}^i = \mathbf{z}^{i-1} + \beta_{i-1} \mathbf{p}^{i-1}$
5. Update halos for \mathbf{p}^i
6. $\mathbf{q}^i = \mathbf{A} \mathbf{p}^i$
7. * $\alpha_i = \rho_{i-1} / (\mathbf{p}^{iT} \cdot \mathbf{q}^i)$
8. $\mathbf{x}^i = \mathbf{x}^{i-1} + \alpha_i \mathbf{p}^i$
9. $\mathbf{r}^i = \mathbf{r}^{i-1} - \alpha_i \mathbf{q}^i$
10. * Calculation of residual norm

where * operations imply collective communication for global reduction operation:

- Summation of a single value: steps 2 and 7
- Maximum of a single value: step 10

1.3.4 Properties of $\hat{\mathbf{A}}_i^{2D}$ that affect the convergence of the iterative method

The number of iterations needed to converge a Krylov-subspace method is closely related with the spectral condition number, $\kappa(\hat{\mathbf{A}}_i^{2D})$. Well-posed systems (κ keeps close to unity) converge easily whereas they tend to degrade fast when the system becomes ill-posed ($\kappa \gg 1$). In our particular case, as the matrices $\hat{\mathbf{A}}_i^{2D}$ are symmetric and positive semidefinite, the condition number κ is given by

$$\kappa(\hat{\mathbf{A}}_i^{2D}) = \frac{\max_j \left\{ \lambda_j(\hat{\mathbf{A}}_i^{2D}) \right\}}{\min_j \left\{ \lambda_j(\hat{\mathbf{A}}_i^{2D}) \right\}} \quad (1.41)$$

where $\lambda_j(\hat{\mathbf{A}}_i^{2D}) \in \mathbb{R}_0^+$ denotes any of the eigenvalues of $\hat{\mathbf{A}}_i^{2D}$.

A complete analytical approach is beyond the scope of this paper. Nevertheless, the essential ideas and results can be conveyed in a more intuitive manner without the mathematical machinery required for the most general cases. In the following subsections relevant theoretical properties are analysed.

Singularity of $\hat{\mathbf{A}}_1^{2D}$

After the Fourier decomposition, singularity of \mathbf{A}^{3D} (see [2], for details) is inherited only by the transformed system of the first plane (Eq. 1.26 with $i = 1$). According to (1.25) the sum of all non-null terms of matrix $\hat{\mathbf{A}}_1^{2D}$ in each row is zero

$$\begin{aligned} \hat{a}_{1,j,k}^p &= (\lambda_1)_{j,k} = a_{j,k}^p + 2 \sum_{r=1}^{N_x/2-1} a_{j,k}^{e_r} \\ &= - \sum_{r=1}^{N_x/2-1} \left(a_{j,k}^{b_r} + a_{j,k}^{s_r} + a_{j,k}^{n_r} + a_{j,k}^{t_r} \right) \end{aligned} \quad (1.42)$$

Recalling that matrix $\hat{\mathbf{A}}_1^{2D}$ exactly corresponds to the 2D discrete Poisson equation [1], it can be shown (by construction) that $\hat{\mathbf{A}}_1^{2D}$ is a positive semi-definite matrix

$$\mathbf{x}^t \hat{\mathbf{A}}_1^{2D} \mathbf{x} \geq 0, \quad \forall \mathbf{x} \neq \mathbf{0} \quad (1.43)$$

Therefore, former expressions (1.42) and (1.43) imply that minimum eigenvalue of $\hat{\mathbf{A}}_i^{2D}$ is equal to zero,

$$\min_j \left\{ \lambda_j \left(\hat{\mathbf{A}}_1^{2D} \right) \right\} = 0 \quad (1.44)$$

whose associated eigenvector is the unity vector

$$\hat{\mathbf{A}}_1^{2D} \mathbf{1} = \mathbf{0} \quad (1.45)$$

and consequently its spectral condition number is infinite

$$\kappa \left(\hat{\mathbf{A}}_1^{2D} \right) = +\infty \quad (1.46)$$

Relation between two matrices $\hat{\mathbf{A}}_{i_1}^{2D}$ and $\hat{\mathbf{A}}_{i_2}^{2D}$

By construction, all the matrices of the set of systems (1.26) are equal except their diagonal elements,

$$\mathbf{D}_{i_1, i_2} \equiv \hat{\mathbf{A}}_{i_1}^{2D} - \hat{\mathbf{A}}_{i_2}^{2D} \quad (1.47)$$

where difference diagonal matrix, \mathbf{D}_{i_1, i_2} , can be directly evaluated from Eq.(1.25),

$$\mathbf{d}_{i_1, i_2}^{j, k} = 2 \sum_{r=1}^{N_x/2-1} a_{j, k}^{e_r} (\cos(r\alpha_{i_1}) - \cos(r\alpha_{i_2})) \quad (1.48)$$

where $\alpha_i = [2\pi(i-1)]/N_x$ and $\mathbf{d}_{i_1, i_2}^{j, k}$ stands for the diagonal element of \mathbf{D}_{i_1, i_2} associated with the (j, k) -position on the yz -plane⁸. Using the trigonometric double angle formula previous expression results into a more convenient form

$$\mathbf{d}_{i_1, i_2}^{j, k} = -4 \{g_{j, k}(\alpha_{i_1}) - g_{j, k}(\alpha_{i_2})\} \quad (1.50)$$

where

$$g_{j, k}(\alpha) \equiv \sum_{r=1}^{N_x/2-1} a_{j, k}^{e_r} \sin^2 \left(r \frac{\alpha}{2} \right) \quad (1.51)$$

⁸Therefore, for the standard rowwise lexicographical ordering $\mathbf{d}_{i_1, i_2}^{j, k}$ is given by

$$\mathbf{d}_{i_1, i_2}^{j, k} \equiv [\mathbf{D}_{i_1, i_2}]_{jN_y+k, jN_y+k} \quad (1.49)$$

Moreover, it can be shown that for the family of discretizations here considered following properties are hold⁹

$$\frac{\partial}{\partial \alpha} g_{j,k}(\alpha) \begin{cases} < 0 & , & 0 < \alpha < \pi \\ > 0 & , & \pi < \alpha < 2\pi \end{cases} \quad (1.52)$$

$$g_{j,k}(\alpha) < 0 , \quad 0 < \alpha < 2\pi \quad (1.53)$$

Taking $i_2 = 1$, expression (1.50) becomes

$$\mathbf{d}_{i_1,1}^{j,k} = -4g_{j,k}(\alpha_n) \quad (1.54)$$

that combined with property (1.53) guarantees that matrices $\mathbf{D}_{i_1,1}$ are positive-definite

$$\mathbf{x}^t \mathbf{D}_{i_1,1} \mathbf{x} > 0 , \quad \forall \mathbf{x} \neq \mathbf{0}, \quad i_1 > 1 \quad (1.55)$$

for any value of N_x . Finally, expressions (1.43) and (1.55) automatically imply that matrices $\hat{\mathbf{A}}_i^{2D}$ (with $i > 1$) are also definite-positive

$$\min_j \left\{ \lambda_j \left(\hat{\mathbf{A}}_i^{2D} \right) \right\} > 0 , \quad i > 1 \quad (1.56)$$

Moreover, from (1.54) it can be viewed that matrices $\hat{\mathbf{A}}_i^{2D}$ are paired as follows

$$\hat{\mathbf{A}}_{i_1}^{2D} = \hat{\mathbf{A}}_{i_2}^{2D} , \quad \text{if } (i_1 + i_2 - 2) = N_x \quad (1.57)$$

⁹Proof for the second-order discretization is straightforward. However, a formal proof for generic order case is beyond the scope of this paper. Anyhow, exhaustive numerical experiments have been successfully performed to assess the validity of these identities.

Hence, later property allows us to almost¹⁰ halve the number of different systems to be solved. Now, each system have to be solved twice. Consequently, the set of systems (1.26) reduces

$$\hat{\mathbf{A}}_1^{2D} \mathbf{x}_1^{2D} = \mathbf{b}_1^{2D} \quad (1.58)$$

$$\left. \begin{aligned} \hat{\mathbf{A}}_i^{2D} \mathbf{x}_i^{2D} &= \mathbf{b}_i^{2D} \\ \hat{\mathbf{A}}_i^{2D} \mathbf{x}_{N_x-i+2}^{2D} &= \mathbf{b}_{N_x-i+2}^{2D} \end{aligned} \right\} i = 2, \dots, \frac{N_x}{2} \quad (1.59)$$

$$\hat{\mathbf{A}}_{N_x/2+1}^{2D} \mathbf{x}_{N_x/2+1}^{2D} = \mathbf{b}_{N_x/2+1}^{2D} \quad (1.60)$$

where matrices $\hat{\mathbf{A}}_i^{2D}$ are ordered descending condition number. That is,

$$\kappa \left(\hat{\mathbf{A}}_i^{2D} \right) > \kappa \left(\hat{\mathbf{A}}_{i+1}^{2D} \right) \quad i = 1, \dots, \frac{N_x}{2} \quad (1.61)$$

which follows from the property (1.52) of $g_{j,k}(\alpha)$ function.

Halving the number of linear systems to be solved is an important issue. Although it is apparently irrelevant in terms of computational cost it may become important in terms of RAM memory requirements and subsequently reduce the computational cost of the global algorithm. Therefore, further we only consider the set of matrices from 1 to $N_x/2 + 1$.

Spectral condition number of $\hat{\mathbf{A}}_i^{2D}$: a simplified analytical approach

Nevertheless, to know how well-posed are our systems we need more information about their spectral condition number, $\kappa \left(\hat{\mathbf{A}}_i^{2D} \right)$. A complete analytical approach for the general case is beyond the scope of this paper. However, the essential ideas and results can be conveyed for the second-order scheme on a uniformly distributed mesh. For this simplified case, matrix $\mathbf{D}_{i,1}$ reduces

$$\mathbf{D}_{i,1} = 4h \sin^2 \left(\frac{\pi(i-1)}{N_x} \right) \mathbf{I} \quad (1.62)$$

where h is the mesh spacing. Consequently all matrices $\hat{\mathbf{A}}_i^{2D}$ will have the same eigenvectors and their eigenvalues will be given by

$$\lambda_j \left(\hat{\mathbf{A}}_i^{2D} \right) = \lambda_j \left(\hat{\mathbf{A}}_1^{2D} \right) + 4h \sin^2 \left(\frac{\pi(i-1)}{N_x} \right) \quad (1.63)$$

¹⁰Note that matrices $\hat{\mathbf{A}}_1^{2D}$, corresponding to the zero-frequency plane, and $\hat{\mathbf{A}}_{N_x/2+1}^{2D}$, corresponding to the highest-frequency plane, are not paired with any other matrix.

and the corresponding spectral condition number

$$\kappa\left(\hat{\mathbf{A}}_i^{2D}\right) = \frac{\max_j \left\{ \lambda_j \left(\hat{\mathbf{A}}_1^{2D} \right) \right\} + 4h \sin^2 \left(\pi(i-1)/N_x \right)}{\min_j \left\{ \lambda_j \left(\hat{\mathbf{A}}_1^{2D} \right) \right\} + 4h \sin^2 \left(\pi(i-1)/N_x \right)} \quad (1.64)$$

Gershgorin circle theorem gives an upper bound for $\max_j \left\{ \lambda_j \left(\hat{\mathbf{A}}_1^{2D} \right) \right\}$

$$\max_j \left\{ \lambda_j \left(\hat{\mathbf{A}}_1^{2D} \right) \right\} \leq 8h \quad (1.65)$$

that combined with (1.44) leads

$$\kappa\left(\hat{\mathbf{A}}_i^{2D}\right) \leq 1 + \frac{2}{\sin^2 \left(\pi(i-1)/N_x \right)} \quad (1.66)$$

that gives an approximate idea of how well-posed are the systems to be solved as a function of the relative number of plane defined as

$$\xi(i, N_x) \equiv \frac{2(i-1)}{N_x} \quad (1.67)$$

Convergence analysis of CG algorithm provide an upper bound on the convergence rate (see [16], for instance)

$$\|\mathbf{e}_i^n\|_{\hat{\mathbf{A}}_i^{2D}} \leq 2 \underbrace{\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)}_{\omega}^i \|\mathbf{e}_i^0\|_{\hat{\mathbf{A}}_i^{2D}} \quad (1.68)$$

where $\mathbf{e}_i^n = \mathbf{x}_i^n - \left(\hat{\mathbf{A}}_i^{2D} \right)^{-1} \mathbf{b}_i^{2D}$ is the solution error after n iterations and the \mathbf{A} -norm is defined as $\|\mathbf{e}\|_{\mathbf{A}} = \left(\mathbf{e}^t \mathbf{A} \mathbf{e} \right)^{1/2}$. Then, after some straightforward calculations, the convergence rate ω can be upper bounded as a function of ξ

$$\omega(\xi) \leq \frac{1}{1 + S\sqrt{S^2 + 2} + S^2} \quad (1.69)$$

where $S = \sin(\xi\pi/2)$. Former function is displayed in figure 1.4 (solid line).

Preliminary numerical experiments

To assess the validity of the simplified analytical approach described in the previous sections some numerical experiments have been performed for a real CFD application. Test case B3D (see table 1.1) has been chosen as starting point. It corresponds to

a natural convection flow in an air-filled differentially heated cavity of aspect ratio 4 and $Ra = 10^9$ (a detailed description of this case is given further in chapter 2). Residual criteria for each plane is set to

$$\frac{\|\mathbf{r}_i\|_\infty}{\|\mathbf{r}_i^0\|_\infty} \leq 0.01, \quad i = 1, \dots, N_x \quad (1.70)$$

where \mathbf{r}_i^0 is the initial residual. Each plane is decomposed into 32 blocks, dividing y - and z -directions into 4 and 8 parts, respectively. Each block is solved directly with a local band-LU solver. Number of iterations needed to obtain solution of each plane is averaged over 1000 time-steps.

In figure 1.4 (top), the number of iterations required to reach the prescribed accuracy in function of ξ are displayed. As expected, these results show very significant differences in the number of iterations. Number of iterations remains almost equal for both second- and fourth-order discretizations. In figure 1.4 (middle), the estimated convergence rates ω are compared with the idealised second-order upper-bound (1.69). It is observed that qualitatively both plots resemble quite well the idealised limit. Quantitative differences are due to local band-LU preconditioning.

In figure 1.4 (bottom), the same test is performed for there different values of N_x keeping the same spatial discretization in the two non-periodic directions and using a fourth-order discretization. The qualitative behaviour remains the same. However, the number of iterations tends to reduce with the growth of N_x (except for the first plane). This effect may be due to the availability of better initial guesses when finer meshes are used. In conclusion, numerical experiments corroborate the idea of using various strategies to obtain solution on different planes.

1.3.5 Residual criteria and initial guess

The final performance of an iterative method strongly depends on both the level of accuracy demanded for the application it is being used for and the availability of “good” initial guesses. Respect the first, it is generally accepted that a “relatively” high level of accuracy is demanded for time-accurate DNS/LES simulations. However, despite the great importance of this issue very few works have focussed his attention to elucidate which is the minimum level of accuracy required for such applications. According to authors’ opinion this situation responds to the availability of direct solvers for most of DNS/LES applications or very good initial guesses, that allow to keep the Poisson equation overresolved without a significant increasing of the computational cost, for iterative methods. In the following paragraphs we briefly analyse these key aspects. Firstly, a non-dimensional mesh-independent criteria for residual is derived from the discrete incompressibility constraint. Then, such criteria is verified for a

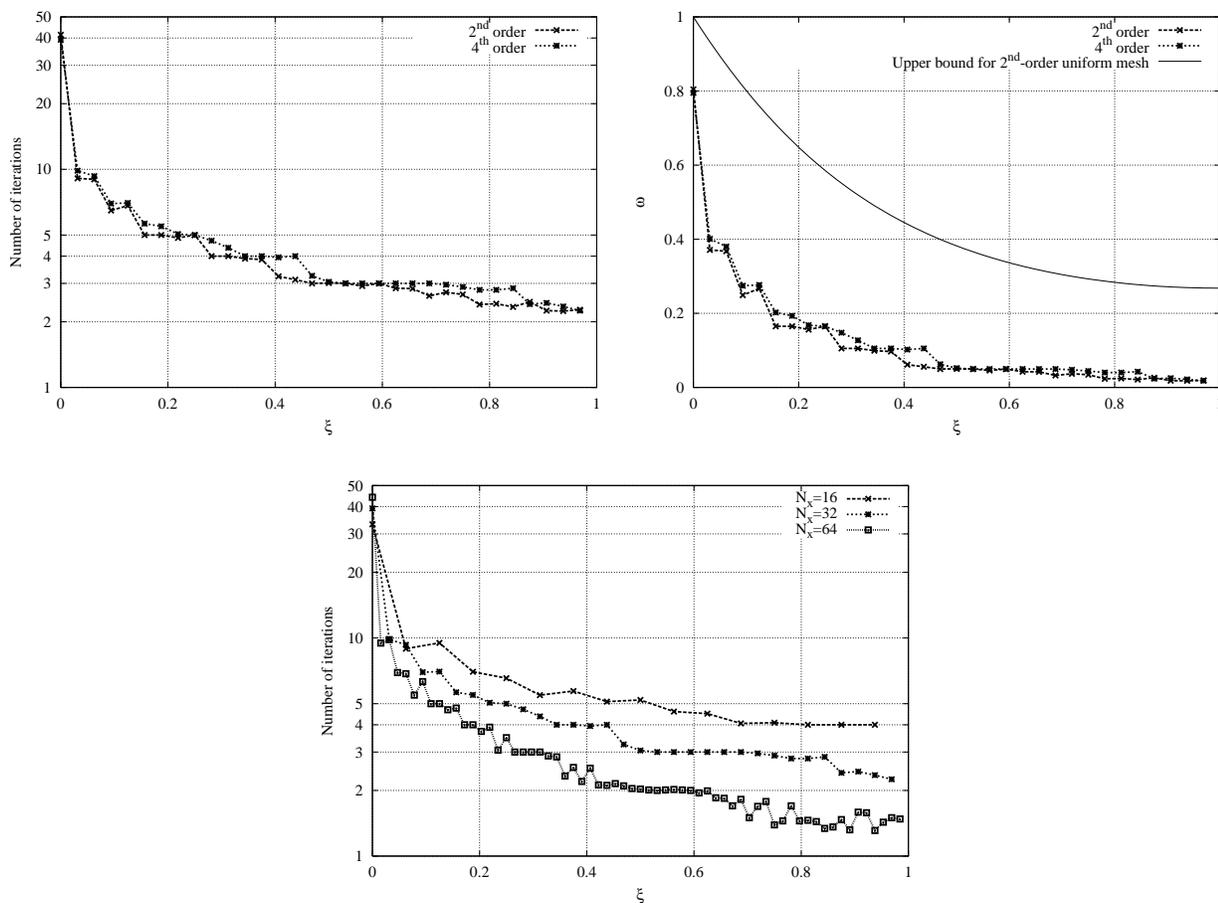


Figure 1.4: Top: comparison of convergence properties of the planes using second- and fourth-order discretizations. Middle: estimated convergence rate ω . Bottom: same plot for different number of planes right.

series of simulations by direct comparison with reference solutions obtained using a direct solver. Finally, the initial guess problem is discussed.

Choice of residual criteria

Once the discrete Poisson equation (1.16) is solved the new discrete velocity field \mathbf{u}_h^{n+1} results from the correction

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^p - \mathbf{G}\mathbf{p}_h = \mathbf{u}_h^p + \Omega^{-1}\mathbf{M}^t\mathbf{p}_h$$

taking the divergence of the previous expression leads

$$\mathbf{M}\mathbf{u}_h^{n+1} = \mathbf{M}\mathbf{u}_h^p + \mathbf{M}\Omega^{-1}\mathbf{M}^t\mathbf{p}_h = \mathbf{M}\mathbf{u}_h^p - \mathbf{L}\mathbf{p}_h$$

Therefore, it is shown that, after the correction, the residual of discrete Poisson equation (1.16)

$$\mathbf{r}_h = -\mathbf{L}\mathbf{p}_h + \mathbf{M}\mathbf{u}_h^p$$

is exactly equal to minus the discrete divergence of the velocity field \mathbf{u}_h^{n+1} (1.71)

$$\mathbf{r}_h = \mathbf{M}\mathbf{u}_h^{n+1} \quad (1.71)$$

Thus, residual of Poisson solver is directly related to a physical quantity. After analysing results of multiple DNS simulations using different meshes and Ra -numbers the following criteria is proposed - to fix the level of decrease of the discrete divergence norm:

$$\frac{\|\mathbf{M}\mathbf{u}_h^{n+1}\|_\infty}{\|\mathbf{M}\mathbf{u}_h^p\|_\infty} \leq \epsilon \quad (1.72)$$

where $\mathbf{M}\mathbf{u}_h^p$ is the value of the discrete divergence of the velocity field before velocity correction. Fixing this ϵ can provide sufficiently accurate solution for different meshes¹¹.

Verification of the criteria and evaluation of ϵ

The criteria is to be used with the same ϵ with no dependence on the number of nodes.

Firstly, series of DNS simulations using different meshes and Ra -numbers have been accomplished in order to find out which ϵ level provides sufficiently accurate

¹¹Corresponding criteria for residual norm for each plane is following: $\|\mathbf{r}_i\|_\infty \leq \bar{\epsilon}$, $i = 1, \dots, N_x$. The value $\bar{\epsilon}$ is corrected automatically during computations: if obtained solution does not satisfy criteria (1.72) then solution is repeated with smaller $\bar{\epsilon}$. This way proper $\bar{\epsilon}$ is set within few first time steps and then only rare corrections may be needed during simulation.

solution and to ensure that this level is equal for all cases. The test cases chosen are a set of 2D and 3D turbulent natural convection flow in air-filled differentially heated cavity of height aspect ratio 4 (detailed description of the problem is given in chapter 2). Case parameters are presented in table 1.1.

| Case | N_x | N_y | N_z | Ra | Pr | Order |
|------|-------|-------|-------|--------|------|----------|
| C2D | - | 39 | 78 | 10^9 | 0.71 | 2^{nd} |
| B2D | - | 78 | 156 | 10^9 | 0.71 | 2^{nd} |
| C3D | 16 | 39 | 78 | 10^9 | 0.71 | 4^{th} |
| B3D | 32 | 78 | 156 | 10^9 | 0.71 | 4^{th} |

Table 1.1: Physical and numerical parameters of the tested cases.

Averaged first and second-order statistics are compared. Since they correspond to time-integration of random processes special care must be taken. We must be sure that differences observed are only due to insufficiently accurate solution of the Poisson equation. Hence, finite time integration period must be long enough (in practice, about 50-100 periods of the lowest significant frequencies suffices). To evaluate acceptable level of difference in results for an iterative method two DNS using DSFD algorithm (direct solver) have been carried out for each case. These two DNS simulation have different random seeds to generate the initial random temperature distribution providing results for two different random processes. Since the Poisson equation is solved directly the differences observed in averaged results are only due to finite integration period and it gives the idea of the acceptable range for solutions using iterative solver.

The numerical tests have shown that reduction of initial norm of divergence by a factor of 10 is sufficient to obtain accurate enough solutions. It must be noted that this value may be valid only for this particular explicit algorithm (see section 2.2.2) and differ for other CFD applications.

Initial guess for iterative solver

Initial guess $(\mathbf{x}_i^0)^{n+1}$ at instant $n + 1$, is obtained by linear extrapolation using two previous solutions:

$$(\mathbf{x}_i^0)^{n+1} = 2(\mathbf{x}_i)^n - (\mathbf{x}_i)^{n-1}, \quad i = 1, \dots, N_x \quad (1.73)$$

This way allows to reduce substantially initial residual for an iterative solver. Moreover, this prediction has an important property that should be noted.

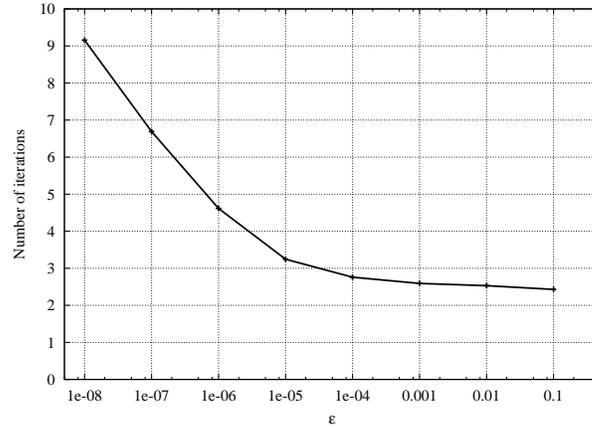


Figure 1.5: Iteration number for different initial divergence norm reduction ϵ (case B3D).

The number of iterations increases with the accuracy imposed for the Poisson equation, but the accuracy enhances the prediction of the initial guess for the next time step and this reduces the next number of iterations. Therefore, changes in the accuracy do not translate immediately to changes in the cost, as can be seen in figure 1.5. This results may explain why, in the context of DNS/LES applications, relatively little attention has been paid to determine the required level of accuracy for Poisson equation.

1.4 Scalable KSFD solver model

The set of two-dimensional problems to be solved (Eqs. 1.58-1.60) are numbered descending the condition number. Hence, a plane with smaller number would require more iterations. Since conditioning number of first plane is infinity an special treatment may be required. According to these features different methods can be applied for different planes.

Planes are decomposed into *blocks*. The size of block governs the number of iterations: the smaller is block the bigger is number of iterations because the smaller parts of plane are coupled directly. But bigger block size requires more memory and computations. For this reason the block size can be chosen independently for each plane to minimise computational price needed to perform all iterations and obtain solution. This way gives a flexible solver model that can be efficiently used on different

parallel systems.

The number of blocks in the i -th plane is denoted by S_i . This parameter can be varied from 1 to N_{yz} . Configuration with only one block ($S_i = 1$) means that the entire plane is solved using the direct DSD method; on the opposite extreme, $S_i = N_{yz}$, means that iterative CG method using a Jacobi preconditioner is being used. Intermediate values corresponds to configurations where CG method is preconditioned using DSD algorithm to solve each block. We denote $P_{yz} = P_y \times P_z$ as the total number of CPUs that cooperate to solve one plane. Thus, each plane is decomposed into P_{yz} subdomains. Then, two different situations are possible,

1. $S_i < P_{yz}$; blocks are bigger than subdomains. Then, to solve each block a DSD parallel direct solver is used.
2. $S_i \geq P_{yz}$; blocks are equal or smaller than subdomains. In this case, an incomplete band-LU factorisation with a drop tolerance of 10^{-4} is used to solve each block directly.

1.4.1 Flexible solver configurations

The general idea behind is demonstrated on a simplified example of the KSFD solver configuration. The set of planes is divided into three groups:

1. $S_i = 1$; planes corresponding to the first group are solved directly using DSD method described in section 1.3.2.
2. $S_i = P_{yz}$; blocks of planes of second group are set equal to subdomains.
3. $S_i > P_{yz}$; for the third group blocks are set smaller than subdomains.

Hence, this simplified configuration is governed by 3 parameters:

- D_1 ; the first delimiter $0 \leq D_1 \leq N_x$.
- D_2 ; the second delimiter $D_1 \leq D_2 \leq N_x$.
- B ; number of blocks within subdomain for the third group.

Therefore, planes with number $i \leq D_1$ belong to first group. Second group includes planes with number $D_1 < i \leq D_2$. Rest of planes belong to third group, whose number of blocks is given by $S_i = B \times P_{yz}$.

In the figure 1.6, three basic types of KSFD solver configurations are displayed:

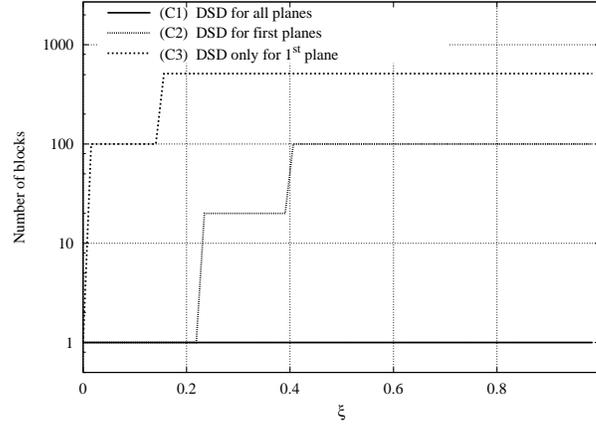


Figure 1.6: Solver configurations example.

1. All the planes are solved directly using the DSD algorithm ($D_1 = D_2 = N_x$) (this situation corresponds to our previous approach [5, 2]). This solver configuration works well on systems with relatively small number of CPUs and moderate large problems. It is specially designed for high-latency parallel systems. Main limitation is the RAM memory requirements (see section 1.3.2).
2. Direct solution with DSD of first few planes and iterative solution of other planes using blocks of subdomain size or smaller ($1 < D_1 < N_x$). This can be used on systems with bigger range of CPU number. In general, iterative solution of the last planes is more efficient than DSD solver. This configuration can be also used on relatively small number of CPU when mesh size is too big to use previous configuration due to RAM memory limitations.
3. Direct solution with DSD of only the first plane ($D_1 = 1$) and iterative solution for the rest of planes using blocks of subdomain size or smaller. This configuration can be used on the edge of capabilities of DSD method on systems with for example 64 – 256 CPUs (or even larger). In this case application of DSD even for only one plane is mostly limited by preprocessing stage computing time. The first plane is especially problematic for iterative solution, hence using DSD for only the first plane makes solver substantially faster.

1.4.2 Illustrative parallel performance test

The test case chosen to measure the performance of the KSFD algorithm corresponds to a DNS simulation of a turbulent natural convection flow in air-filled ($Pr = 0.71$)

differentially heated cavity of aspect ratio 4 and $Ra = 10^{11}$. More details of this configuration can be found in chapter 2. Residual criteria is set to $\epsilon = 10^{-5}$. Parallel performance tests were carried out on two different parallel systems:

- MareNostrum supercomputer of the Barcelona Supercomputing Center (BSC). It is an IBM BladeCenter with 10240 PowerPC 970MP processors at 2.3 GHz. Dual nodes with 8 Gb (4 Gb per node) are coupled with a high-performance Myrinet network. Auxiliary 1 Gbit Ethernet network is used for shared file system.
- JFF cluster of the Heat and Mass Transfer Technological Center (CTTC). It is a Beowulf PC cluster based on common office computer equipment. 40 single CPU nodes with AMD Athlon 2.6 GHz processors and 1 Gb of RAM are coupled with an Ethernet 100 Mbits/s network.

Roughly speaking, the two main differences between these two parallel systems are the network performance (Myrinet network latency has smaller latency and bigger bandwidth than JFF Ethernet network) and the number of CPUs (BSC has allowed us to perform simulations using up to 1024 processors).

Demonstration of configuration choice for two different parallel systems

The following test has been chosen to illustrate the way the KSFD solver can be easily adapted for different kind of parallel systems to provide maximal performance. The mesh size is $64 \times 240 \times 460$ ($\approx 7 \times 10^6$ nodes). Test is performed using 32 CPUs and domain is consequently decomposed into 32 subdomains ($P_y = 4$ and $P_z = 8$, respectively). Then, four basic solver configurations (see figure 1.8) are tested:

1. $D_1 = 64$; direct solution with DSD.
2. $D_1 = 0, D_2 = 64$; blocks are equal to subdomains.
3. $D_1 = 0, D_2 = 0, B = 4$; each subdomain is divided into 4 blocks.
4. $D_1 = 0, D_2 = 0, B = 16$ each subdomain is divided into 16 blocks.

To measure the averaged computing time of each plane t_i , we measure the total computing time, T , to solve the entire set of planes and the number of iterations to solve the i -th plane, I_i . These quantities are obtained after averaging over 200 time-steps. Since for the tested configurations the computational cost per iteration is equal for all planes, computing time per plane is proportional to the number of iterations

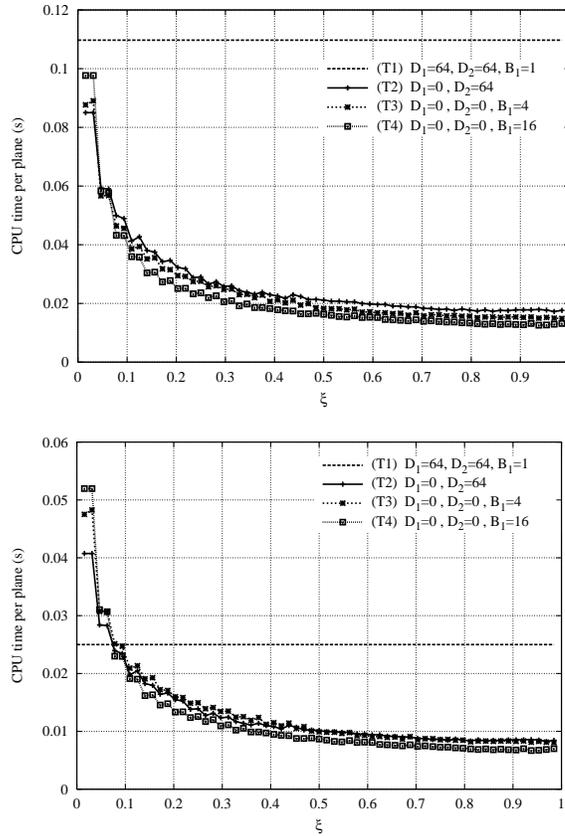


Figure 1.7: Comparison of the solver configurations for different systems: JFF (left) and MareNostrum supercomputer (right). Computing time for each plane is represented using each of the four proposed configurations. This allows to obtain the optimised configuration choosing for each plane the best one. Results for $\xi = 0$, several orders of magnitude larger, have not been displayed.

$$t_i = \frac{I_i T}{\sum_{r=1}^{N_x} I_r} \quad (1.74)$$

Then, direct comparison of the computing time of each plane allows to obtain the optimised configuration built from these four tested configurations. In figure 1.7, results for both parallel systems are displayed. As expected, solution of first plane

requires so many iterations that its results have been omitted. Hence, using the computing time measurements represented in figure 1.7 following optimised configurations are obtained:

1. $D_1 = 1, D_2 = 3, B = 4$ for JFF cluster.
2. $D_1 = 5, D_2 = 5, B = 4$ for MareNostrum.

It must be noted that DSD solver, in this particular case, is about 4 times faster on MareNostrum than on JFF cluster while overall computing time of the optimised configuration is only about twice faster. This is due to the substantial advantage of MareNostrum in network bandwidth which becomes especially important for DSD global summation operation. For this reason more planes are solved directly in configuration optimised for MareNostrum. Optimised configurations are displayed in figure 1.8.

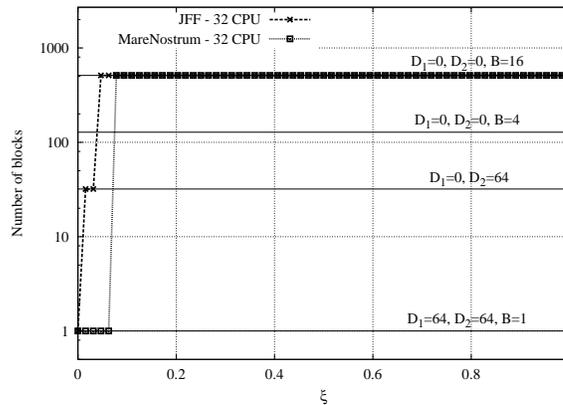


Figure 1.8: Optimised configurations for different parallel systems

Speed-up test

The speed-up, S , the ratio between the computing time needed by one processors and P processors, is traditionally employed as a measure of the parallel performance of a parallel algorithm. However, S is difficult to measure in a realistic situation because the algorithm is designed for large problems that usually can not be solved by a single processor (neither for a reduced number of processors). A speedup test has been carried out on the MareNostrum supercomputer. Mesh size is $32 \times 170 \times 320$ ($\approx 1.7 \times 10^6$ points). This test shows how KSF D solver can be adapted for different

number of CPUs to provide optimal performance and scalability. Following solver configurations are considered:

1. $D_1 = 32$; direct solution with DSD (type C1 in figure 1.6).
2. $D_1 = 5, D_2 = 10, B = 9$; direct solution of planes from 1 to 5; preconditioned CG for the rest (type C2 in figure 1.6).
3. $D_1 = 1, D_2 = 10, B = 9$; direct solution of first plane; preconditioned CG for the rest (type C3 in figure 1.6).

The speed-up results for these three configurations are displayed in figure 1.9. Measurements start from 1 CPU (sequential configuration) for the first configuration where DSD algorithm reduces to band-LU¹² algorithm.

For a relatively low number of CPUs configuration S1 provides optimal results (see figure 1.6). Then, as expected, it begins losing efficiency and is replaced by configuration S2. Finally, configuration S3 provides optimal scalability for large numbers of CPUs: it reaches a speed-up of 153 for 200 processors.

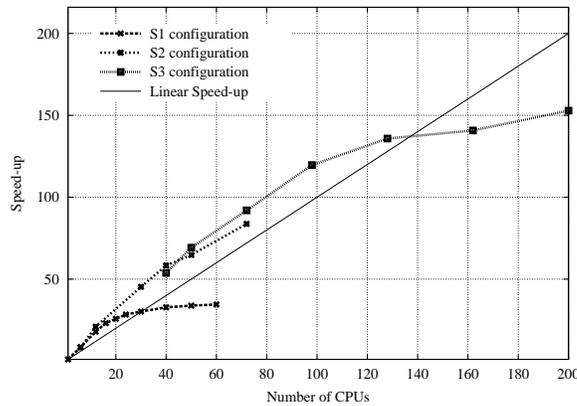


Figure 1.9: Speedup results

1.4.3 Parallelisation in the periodic x -direction

It has been shown that an efficient direct solution of the first plane, whose condition number is infinite (1.46), is crucial for the scalability of the KSF solver to very large-scale problems. However, the huge amount of RAM memory demanded for the DSD

¹²Band-LU decomposition can not be fitted in memory of 1 CPU, thus computing time for only one plane was measured and then scaled by a factor of 32.

may eventually limit its scalability. Moreover, increasing the number of processors do not help too much in terms of memory (see Eq. 1.35) and increase the amount of data to be transferred (see Eq. 1.38).

Therefore, in this context, parallelisation in the x -periodic direction becomes necessary. Hence, computational domain is now decomposed into $P = P_x \times P_y \times P_z$ subdomains. Coordinates of each subdomain will be referred as p_x , p_y and p_z , respectively. Then, two different kind of subsets are considered:

1. 1D-subsets; that is, groups of processors with the same p_y and p_z . Size of each subset is P_x and there are $P_y \times P_z$ 1D-subsets.
2. 2D-subsets; that is, groups of processors with the same p_x . Size of each subset is $P_y \times P_z$ and there are P_x 2D-subsets.

Then, this strategy allows to divide that total number of processors P into P_x 2D-subsets that solve its own set of planes. Doing so, more memory resources are available for those planes solved using the direct DSD algorithm. Additionally, the computational cost of the preprocessing stage to compute the inverse of Schur complement matrices, $\tilde{\mathbf{A}}_{s,s}^{-1}$, is significantly reduced.

Therefore, the set of 2D problems to be solved (1.58-1.60) is now splitted into P_x parts. Each part, that consists¹³ of N_x/P_x planes, is then solved by a 2D-subset of processors. However, the straightforward implementation (planes assigned to 2D-subsets according to the ordering defined in Eq. 1.58 and 1.59) is not convenient because a substantial load imbalance appears: first 2D-subset would have planes with the worst convergence properties meanwhile the last subset would have to deal with the best conditioned systems of the set.

Thus, it becomes necessary to re-order the planes in such a way the load imbalance be minimised. Here, we propose to use the following mapping function

$$i \leftrightarrow \{(i-1) \bmod P_x\} P_x + \left\lceil \frac{i}{P_x} \right\rceil \quad (1.75)$$

where $\lceil \cdot \rceil$ denotes the ceiling function. The same configuration is applied to each 2D-subsets of planes. That is, the same solver configuration will be used for all the i -th planes of each 2D-subset. Doing so, load imbalance between subsets of processors is minimised.

Parallelisation in periodic x -direction is performed by just replication of data for Poisson equation. Then, modified algorithm results

1. Replication of initial guess vector \mathbf{x}_0^{3D} and right-hand-side vector \mathbf{b}^{3D} within 1D-subsets.

¹³Here, we assume that N_x is divisible by P_x .

2. FFT transforms original system (1.17) into (1.26).
3. Solution of Eq.(1.26) by P_x 2D-subsets of processors.
4. Replication of the solution within 1D-subsets.
5. Inverse FFT transforms solution of (1.26) into the solution of (1.17).

Additional broadcast communications are required within each of 1D-subset to provide data replication in stages 1 and 4. Once the data is replicated forward and inverse FFT algorithm can be applied in stages 2 and 5 without parallelisation to vectors of size of N_x elements; like if there were no domain decomposition on x -direction. Thus, on the solution stage 3, each of 2D-subsets has all N_x planes but only N_x/P_x of them to be solved. Then, solution is replicated again within 1D subgroup on stage 4. Finally, inverse FFT transform is applied to obtain the solution of the Poisson equation.

However, the cost of these broadcast operations needed on stage 1 and 4 grows¹⁴ fast with P_x . Therefore, the strategy to be followed should be, for a given problem, try to keep the size of 2D-subsets ($P_y \times P_z$) as big as possible within limitations of the DSD method in order to minimise P_x . Keeping P_x relatively small means that big broadcast operations are not especially time consuming and they do not substantially affect the solver parallel efficiency. This is because communication is done only within small 1D-subsets of P_x processors. Moreover, it must be noted that time consumption of forward and inverse FFT operation is negligible in comparison with overall Poisson solver. Hence, in our context, there is no need to parallelise the FFT algorithm.

In practice, parallelisation in the periodic x -direction has substantially expanded the range of fairly good scalability. It has allowed the KSFD solver to be efficiently used for realistic applications using more than a thousand processors.

Speed-up results from 256 up to 1024 CPUs are displayed in figure 1.10. They correspond to a mesh of more than 111×10^6 points ($128 \times 680 \times 1280$). Details of the simulation are given in the next section. For this test $P_y = 8$ and $P_z = 16$ have been kept constant whereas P_x has varied (2 for 256 processors, 4 for 512 and 8 for 1024). Parallel efficiency for 512 CPUs is nearly 100% whereas for 1000 processors it goes down to 70% approximately. This lost of efficiency is due to the increasing cost of broadcast communications in stages 1 and 4.

¹⁴Of course, this cost would depend on the network performance of our parallel computer.

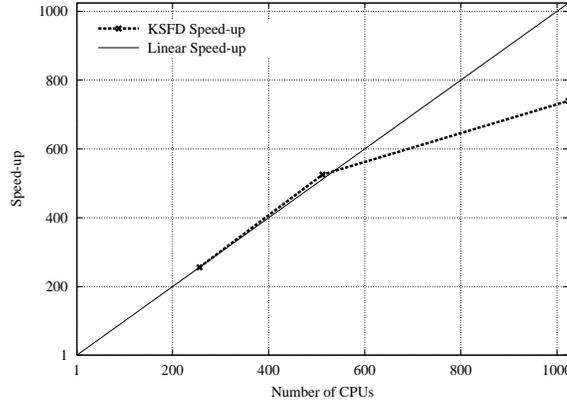


Figure 1.10: Speed-up results parallelising in the periodic x -direction.

1.5 Conclusions

A new parallel Poisson solver that provides a fairly good scalability for a wide range of problem sizes and computer architectures have been presented. The previous version of the solver, called Direct Schur-Fourier Decomposition (DSFD) algorithm [5, 2], was conceived for low-cost PC cluster and allowed us to perform DNS simulations of turbulent natural convection flows [4, 3]. However, such method can not be used for arbitrarily large number of processors and mesh size, mainly due to RAM memory limitations [2]. To do so, a new version of the solver, named Krylov-Schur-Fourier Decomposition (KSFD) has been presented and described in this paper. It basically consists on solving the set of 2D problems using a CG method preconditioned by the DSD algorithm [1]. Each plane is decomposed into blocks that are solved with the DSD solver. The number of blocks governs the convergence properties of the algorithm. Therefore, different strategies can be used for different 2D problems of the set depending on how well-posed is the system to be solved.

The integration of the algorithm in an explicit parallel CFD solver, the problem of the residual criteria and the scalable solver model have been described. For the sake of clarity, main ideas behind the method have been exemplified for several problems corresponding to real CFD applications. The scalability and flexibility of the KSFD algorithm have been successfully shown by performing several numerical experiments on both the MareNostrum supercomputer and a low-cost PC cluster. Speed-up results up to 1024 processors have proved that the algorithm scalability for large-scale problems is good. Finally, KSFD algorithm performance was demonstrated on a DNS simulation of a differentially heated cavity with $Ra = 10^{11}$, $Pr = 0.71$ (air) and height

aspect ratio 4. The DNS has been carried out using mostly 512 processors.

Acknowledgements

This work has been financially supported by the *Ministerio de Educación y Ciencia*, Spain, Project: “Development of high performance parallel codes for the optimal design of thermal equipments”. Contract/grant number ENE2007-67185.

Calculations have been performed on the JFF cluster at the CTTC and on the IBM MareNostrum supercomputer at the Barcelona Supercomputing Center. The author thankfully acknowledge these institutions.

Chapter 2

Direct numerical simulation of a differentially heated 3D cavity on MareNostrum supercomputer

Main contents of this chapter have been published in:

F. X. Trias, A. V. Gorobets, M. Soria and A. Oliva, DNS of natural convection flows on MareNostrum supercomputer - *In Parallel Computational Fluid Dynamics*, Elsevier, 2007.

F. X. Trias, A. V. Gorobets, M. Soria and A. Oliva, Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Ra -number up to 10^{11}

Submitted to *International Journal of Heat and Mass Transfer* 2008.

Abstract.

A set of direct numerical simulations (DNS) of a differentially heated cavity of aspect ratio 4 with adiabatic horizontal walls was presented in [3] [4] for Rayleigh number based on cavity height 6.4×10^8 , 2×10^9 and 10^{10} . The set of DNS presented in this work covers two more configurations with higher Rayleigh numbers 3×10^{10} and 10^{11} , $Pr = 0.71$). Together with previous results it gives a relatively wide range of Ra -numbers from weak to fully developed turbulence. The DNS were performed on Marenostrum supercomputer using scalable parallel Poisson solver represented in chapter 1. Up to 1024 processors were used for solving meshes of up to 110 millions of nodes.

The main features of the flow, including the time-averaged flow structure, the turbulent statistics, the global kinetic energy balances and the internal waves motion phenomenon are described and discussed.

Nomenclature

| | |
|--------------------------------|---|
| A_1, A_3 | depth and height aspect ratios |
| g | gravitational acceleration |
| β | thermal expansion coefficient |
| ρ | fluid density |
| T | temperature |
| T_H | temperature of the hot wall |
| T_C | temperature of the cold wall |
| \mathbf{f} | body force |
| Ra_2 | Rayleigh number based on cavity width, $(g\beta\Delta TL_y^2)/(\nu\alpha)$ |
| Ra_3 | Rayleigh number based on cavity height, $(g\beta\Delta TL_z^2)/(\nu\alpha)$ |
| Pr | Prandtl number, ν/α |
| x, y, z | coordinates |
| \mathbf{u} | velocity field |
| p | dynamics pressure |
| \tilde{p} | pseudo-pressure, $\Delta t/(\beta + 1/2)p$ |
| L_x, L_y, L_z | cavity depth, width and height |
| ν | kinematic viscosity |
| α | thermal diffusivity |
| ΔT | temperature difference, $(T_h - T_c)$ |
| C | dimensionless stratification, |
| f_{BV} | Brunt-Väisälä frequency, $(CPr)^{0.5}/(2\pi)$ |
| \mathbf{M} | discrete divergence operator |
| \mathbf{u}_h | discrete velocity vector |
| Ω | size of control volumes |
| \mathbf{G} | discrete gradient operator |
| \mathbf{D} | discrete diffusive operator |
| $\mathbf{C}(\mathbf{u}_h)$ | convective coefficient matrix |
| Nu | Nusselt number |
| Nu_c | Nusselt number through the vertical mid-plane |
| Δt | time step |
| t | time |
| $\Delta x, \Delta y, \Delta z$ | mesh size |

2.1 Introduction

Natural convection in differentially heated cavities (DHC) has been the subject of numerous studies over the past decades. This configuration models many engineering applications such as ventilation of rooms, cooling of electronics devices or air flow in buildings. Simultaneously, this configuration has served as prototype for the

development of numerical algorithms. A summary of previous direct numerical simulations of air-filled ($Pr = 0.71$) differentially heated cavity relevant in our context is presented in the next paragraphs. The coordinate system used here is: x for the periodic direction and y (horizontal) and z (vertical) for the two wall-normal directions. Ra_2 and Ra_3 are the Rayleigh numbers based on the cavity width and height respectively. $A_3 = L_z/L_y$ and $A_1 = L_x/L_y$ are the height and depth aspect ratios. Unless otherwise mentioned, all cases use Boussinesq approximation.

The early numerical studies concentrated on configurations characterized by small Rayleigh numbers in the steady laminar regime. After the pioneering work of Vahl Davis & Jones [26], where the original benchmark formulation was established for a set of square two-dimensional cavities with $10^3 \leq Ra \leq 10^6$, Hortmann *et al.* [27] used a multigrid method to solve the problem with finer meshes up to 640×640 . Latter, solutions for the full range of two-dimensional steady-state solutions ($Ra_3 \leq 10^8$) have been obtained using different methods by Le Quéré [28], Ravi *et al.* [29] and Wan *et al.* [30]. The three-dimensional cubic cavity ($A_1 = A_3 = 1$), with adiabatic horizontal walls and solid vertical walls in the third direction is also a well-known configuration, but has received comparatively less attention (see Fusegi *et al.* [31]; Tric *et al.* [32]). For large height aspect ratio cavities, in a certain range of Ra numbers, a steady-state multicellular flow is obtained (see Lartigue *et al.* [33]; Le Quéré [34]; Schweiger *et al.* [35]).

Beyond a critical Rayleigh number, the two-dimensional differentially heated cavity flows become time-dependent (periodic, chaotic and eventually fully turbulent). Due to the presence of high temperature areas at the bottom of the cavity, the configuration with perfectly conducting horizontal walls is more unstable than the configuration with adiabatic ones. Its transition to non-steadiness was studied by Winters [36], obtaining a critical number of $Ra_3 = 2.109 \times 10^6$, later confirmed by Henkes [37]. For the square cavity with adiabatic horizontal walls, Le Quéré & Behnia [38] determined the critical number as $Ra = 1.82 \pm 0.01 \times 10^8$ and studied the time-dependent chaotic flows up to $Ra = 10^{10}$. For the case of cavities also with adiabatic horizontal walls and height aspect ratio $A_3 = 4$, Le Quéré [39] determined that there is a Hopf bifurcation at $Ra_3 = 1.03 \times 10^8$ and that a chaotic behaviour is first observed at $Ra_3 = 2.3 \times 10^8$. Two-dimensional chaotic flows have been studied by Farhangnia *et al.* [40], who carried out a direct simulation for $Ra_3 = 6.4 \times 10^{10}$ and by Xin & Le Quéré [41], who studied the situations with $Ra_3 = 6.4 \times 10^8$, 2×10^9 and 10^{10} . The cavity with $A_3 = 8$, with $Ra_2 = 3.4 \times 10^5$ (unsteady), has been chosen as a test problem (see Christon *et al.* [42], for example). For this configuration, the critical Ra_2 number for the transition to unsteadiness is $Ra_2 = 3.0619 \times 10^5$. Time-dependent two-dimensional flows have also been studied without using the Boussinesq approximation by Paolucci & Chenoweth [43] and Paolucci [44], for Rayleigh numbers up to 10^{10} and different aspect ratios.

The transition from a two-dimensional steady laminar to a three-dimensional time-dependent regime was first considered by Jansen *et al.* [45], for the case of a perfectly conducting horizontal walls in a cubic cavity with solid vertical walls and numerically imposing flow symmetries. The critical Ra number was estimated to be between 2.25×10^6 and 2.35×10^6 , larger than the equivalent critical number of a two-dimensional cavity. An oscillatory flow regime was studied by Fusegi *et al.* [46] in the same conditions, without assuming symmetry, for $Ra = 8.5 \times 10^6$. For the case of the adiabatic horizontal walls in a cubic cavity, the transition to unsteadiness was studied by Jansen & Henkes [47], assuming symmetry too. The critical Ra number obtained was between 2.5×10^8 and 3×10^8 . However, the same configuration was later studied without assuming symmetry by Labrosse *et al.* [48], obtaining a non-symmetric transition for $Ra = 3.19 \times 10^7$, significantly lower than in the equivalent two-dimensional case. A direct simulation of a turbulent flow with $Ra = 10^{10}$ was carried out by Fusegi *et al.* [49], using a $62 \times 122 \times 62$ mesh.

In a general differentially heated cavity problem with periodic vertical boundary conditions where the boundary conditions do not force the flow to be three-dimensional, there are three possible flow configurations: two-dimensional steady, two-dimensional unsteady and three-dimensional unsteady. A question relevant in our context is if there is a range of Rayleigh numbers where the flow is two-dimensional but unsteady, this is, if the transitions to unsteadiness and three dimensionality are simultaneous. This problem was considered in detail by Henkes & Le Quéré [50], for square differentially heated cavities ($A_3 = 1$) with adiabatic horizontal walls and perfectly conducting horizontal walls, using periodic vertical boundary conditions. In both cases, it was found that three-dimensional perturbations are less stable than two-dimensional perturbations, concluding that the assumption of bidimensionality is not correct (in time-dependent square differentially heated cavity). A three-dimensional simulation was carried out for $Ra = 10^8$, with perfectly conducting horizontal walls, using $A_1 = 0.1$ with four Fourier modes in the x -direction. Statistics of the flow were recorded and compared with the statistics of the two-dimensional flow. The most significant difference found was an increase of the heat transfer coefficient in the three-dimensional flow. For cavities of height aspect ratio 4, the same question was considered by Penot *et al.* [51]. Experimentally, it was found that in a cavity of $A_1 = 1.33$ there is a transition to unsteadiness at $Ra_3 \approx 10^8$, in good with the results reported by Xin & Le Quéré [41]. Two-dimensional numerical simulations confirmed this result. However, in a three-dimensional periodic vertical boundary conditions simulation with a slightly lower Rayleigh number, $Ra_3 = 9.6 \times 10^7$, with $A_1 = 1$, the flow was found to be three-dimensional. Hence, these results seem to indicate that for $A_3 = 1$ and $A_3 = 4$ (and large enough A_1), the flows would never be unsteady and two-dimensional. But this conclusion is not valid for other aspect ratios: Xin & Le Quéré [52] showed that for $A_3 = 8$, the critical number for the transition to

unsteadiness is $Ra_2 = 306192 \pm 10$, while the two-dimensional to three-dimensional transition is observed at a higher Ra_2 number, at least $\approx 3.84 \times 10^5$.

Concerning the turbulent regime, a three-dimensional DNS of a differentially heated cavity with adiabatic horizontal walls and periodic vertical boundary conditions with $A_3 = 4$, $Pr = 0.71$ and $Ra_3 = 6.4 \times 10^8$ performed by Soria *et al.* [4] provided worthwhile data to investigate three-dimensional effects, finding that the general features of the averaged flow do not change significantly if the flow is treated as two-dimensional. The main differences were found near the downstream corners where stronger recirculations occur for the two-dimensional simulation. On the other hand, turbulent statistics are substantially different, specially at the vertical boundary layers where two-dimensional simulation incorrectly predicts very low turbulence values. However, the Rayleigh number studied by Soria *et al.* [4], which is only ≈ 6 times above the critical value, corresponds to a very weak turbulent flow. Recently, a set of two-dimensional and three-dimensional DNS simulations with $Ra_3 = 6.4 \times 10^8$, 2×10^9 and 10^{10} have been presented and analyzed by Trias *et al.* [3]. When comparing two-dimensional and three-dimensional results significant differences are observed in the flow dynamics. Large unsteady eddies observed in the two-dimensional simulations do not persist in the three-dimensional counterparts. Their energy is rapidly passed down to smaller scales of motion causing an evident reduction of the large-scale mixing effect at the hot upper and cold lower regions and consequently a still motionless stratified cavity core is displayed. This also causes that the transition point at the vertical boundary layers clearly moves downstream for the two-dimensional simulations. These differences become more marked when the Rayleigh number is increased. In conclusion, in the turbulent regime, three-dimensional simulations are necessary for an accurate description of the flow, specially for turbulent statistics.

2.1.1 Motivation and summary of the present work

The main goal of the present work is to improve our understanding of the dynamics of turbulent convection in a differentially heated cavity up to Ra -number 10^{11} (*i.e.* three orders of magnitude higher than the critical Ra -number). To do so, a set of five direct numerical simulations (DNS) of a DHC of aspect ratio 4 ($Ra_3 = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} , $Pr = 0.71$) are presented and analyzed. They cover a relatively wide range of Ra -numbers from weak to fully developed turbulence. These configurations have been selected as an extension of our previous work [3] where two-dimensional and three-dimensional results for the three lowest Ra -numbers were presented and compared. Here, results for the two lowest Rayleigh numbers are reported again while simulation at $Ra_3 = 10^{10}$ has been carried out again for a finer mesh. Periodic boundary conditions in the x -direction are used because they allow the study of three-dimensional effects due to intrinsic instability of the main flow and

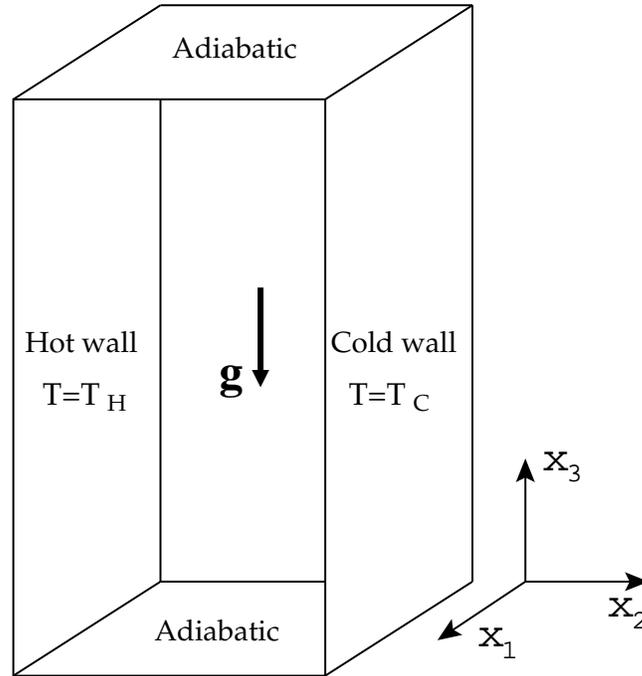


Figure 2.1: Differentially heated cavity problem.

not to the boundary conditions. Hence, a uniform mesh in such direction is suitable. This is an important computational advantage because Fourier-based methods can be used to solve the Poisson equation in one direction.

The present chapter is arranged as follows. In the next section the governing equations and the numerical method are briefly described. In section 2.3, the main features of the flows, including the time-averaged flow structure, the turbulent statistics, the global kinetic energy balances and the internal waves motion phenomena are presented and discussed. Finally, relevant results are summarized and conclusions are given in section 2.4.

2.2 Governing equations and numerical method

2.2.1 Governing equations

The cavity under consideration is of height L_z , width L_y and depth L_x (height and depth aspect ratios are $A_3 = L_z/L_y$ and $A_1 = L_x/L_y$, respectively) filled with an incompressible Newtonian viscous fluid of kinematic viscosity ν and thermal diffusivity α . To account for the density variations, the Boussinesq approximation is used. Thermal radiation is neglected. Under these assumptions, the dimensionless governing equations in primitive variables are

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{Pr}{Ra^{0.5}} \nabla^2 \mathbf{u} - \nabla p + \mathbf{f} \quad (2.2)$$

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \frac{1}{Ra^{0.5}} \nabla^2 T \quad (2.3)$$

where Ra^1 is the Rayleigh number based on the cavity height $(g\beta\Delta TL_z^3)/(\nu\alpha)$ and $Pr = \nu/\alpha$ and the body force vector is $\mathbf{f} = (0, 0, PrT)$. The reference length, time, velocity, temperature and dynamic pressure used for the dimensionless form are respectively L_z , $(L_z^2/\alpha) Ra^{-0.5}$, $(\alpha/L_z) Ra^{0.5}$, ΔT and $\rho(\alpha^2/L_z^2) Ra$. With the above reference quantities the vertical buoyant velocity, $Pr^{0.5}$, and the characteristic dimensionless Brunt-Väisälä frequency, $f_{BV} = (CPr)^{0.5}/(2\pi)$, where C is the dimensionless stratification of the time-averaged temperature field, are independent of the Rayleigh number.

The cavity is subjected to a temperature difference ΔT across the vertical isothermal walls ($T(x, 0, z) = 1$, $T(x, 1/A_3, z) = 0$) while the top and bottom walls are adiabatic. At the four planes $y = 0, y = 1/A_3, z = 0, z = 1$, non-slip boundary condition are imposed for velocity. Periodic boundary conditions in the x -direction are used,

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x} + L_x \mathbf{e}_x, t) \quad (2.4)$$

$$T(\mathbf{x}, t) = T(\mathbf{x} + L_x \mathbf{e}_x, t) \quad (2.5)$$

The initial conditions are not relevant because the statistics of the flow are recorded after a long enough time-integration period to reach a statistically steady-state behaviour. Periodic vertical boundary conditions are used because they allow to study the three-dimensional effects due to intrinsic instability of the main flow and not to

¹Here, for simplicity, the subindex 3 is dropped.

the boundary conditions. If we furthermore consider that the cavity is filled with air ($Pr = 0.71$) and that its height aspect ratio A_3 is equal to 4, then the configuration only depends on the Rayleigh number Ra and the depth aspect ratio A_1 .

2.2.2 Numerical method

Time-integration method

In order to simplify the notation, momentum equation (2.2) can be rewritten as

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{R}(\mathbf{u}) - \nabla p \quad (2.6)$$

where $\mathbf{R}(\mathbf{u})$ represents the right-hand-side terms of the momentum equation except for the pressure gradient,

$$\mathbf{R}(\mathbf{u}) \equiv Pr \nabla^2 \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{u} + \mathbf{f} \quad (2.7)$$

For the temporal discretization, a central difference scheme is used for the time derivative term, a fully explicit second-order one-leg scheme [12] for $\mathbf{R}(\mathbf{u})$ and a first-order backward Euler scheme for the pressure-gradient term. Incompressibility constraint is treated implicit. Thus, we obtain the semi-discretized NS equations

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (2.8)$$

$$\frac{(\beta + 1/2) \mathbf{u}^{n+1} - 2\beta \mathbf{u}^n + (\beta - 1/2) \mathbf{u}^{n-1}}{\Delta t} = \mathbf{R}((1 + \beta) \mathbf{u}^n - \beta \mathbf{u}^{n-1}) - \nabla p^{n+1} \quad (2.9)$$

where the parameter β is computed each time-step to adapt the linear stability domain of the time-integration scheme to the instantaneous flow conditions in order to use the maximum Δt possible. For further details about the time-integration method the reader is referred to [3].

To solve the velocity-pressure coupling a classical fractional step projection method [17, 18] is used. In the projection methods, solutions of the unsteady NS equations are obtained by first time-advancing the velocity field \mathbf{u} without regard for its solenoidality constraint (2.8), then recovering the proper solenoidal velocity field, \mathbf{u}^{n+1} ($\nabla \cdot \mathbf{u}^{n+1} = 0$). This projection is derived from the Helmholtz-Hodge vector decomposition theorem [19], whereby the velocity \mathbf{u}^{n+1} can be uniquely decomposed into a solenoidal vector, \mathbf{u}^p , and a curl-free vector, expressed as the gradient of a scalar field, $\nabla \tilde{p}$. This decomposition is written as

$$\mathbf{u}^p = \mathbf{u}^{n+1} + \nabla \tilde{p} \quad (2.10)$$

where the predictor velocity \mathbf{u}^p is

$$\mathbf{u}^p = \frac{2\beta\mathbf{u}^n - (\beta - 1/2)\mathbf{u}^{n-1}}{\beta + 1/2} + \frac{\Delta t}{\beta + 1/2} \mathbf{R}((1 + \beta)\mathbf{u}^n - \beta\mathbf{u}^{n-1}) \quad (2.11)$$

and the pseudo-pressure is $\tilde{p} = \Delta t / (\beta + 1/2) p^{n+1}$. Taking the divergence of (2.10) yields a Poisson equation for \tilde{p}

$$\nabla \cdot \mathbf{u}^p = \nabla \cdot \mathbf{u}^{n+1} + \nabla \cdot (\nabla \tilde{p}) \longrightarrow \nabla^2 \tilde{p} = \nabla \cdot \mathbf{u}^p \quad (2.12)$$

The question of what boundary condition to use for the pressure equation (2.12) in the non-periodic directions has led to much discussion. The main ideas were shortly reviewed in [20]. The use of the normal component of the momentum equation is commonly accepted as the most appropriate boundary condition, see for example [21]. However, at the discrete level on staggered grids with prescribed velocity boundary conditions, as in our case, the incompressibility condition occurs naturally and no specific boundary condition for the pressure needs to be specified as pointed out in [22].

Finally, once the solution is obtained, \mathbf{u}^{n+1} results from the correction

$$\mathbf{u}^{n+1} = \mathbf{u}^p - \nabla \tilde{p} \quad (2.13)$$

Therefore, the algorithm for the integration of each time-step is

1. Evaluate $\mathbf{R}((1 + \beta)\mathbf{u}^n - \beta\mathbf{u}^{n-1})$.
2. Evaluate the predictor velocity \mathbf{u}^p from Eq.(2.11).
3. Evaluate $\nabla \cdot \mathbf{u}^p$ and solve the discrete Poisson (2.12) equation.
4. Obtain the new velocity field with Eq.(2.13).

It is well-known that due to stability reasons explicit temporal schemes introduce severe restrictions on the time-step, while implicit discretization would improve the overall stability. However, for the use of implicit methods in DNS of turbulent flows the computational costs are rather high compared to those of explicit methods. This is because of the underlying restrictions to time-step that are required to fully resolve all temporal scales in the NS equations. Therefore, only explicit methods are considered in the view of the lower costs.

Spatial discretization

Governing equations (2.1-2.3) are discretized on a staggered grid in space by second- or fourth-order symmetry-preserving discretizations [12]. Following the same notation, the symmetry-preserving discretization of the Navier-Stokes equations becomes

$$\mathbf{M}\mathbf{u}_h = \mathbf{0} \quad (2.14)$$

$$\Omega \frac{d\mathbf{u}_h}{dt} = -\mathbf{C}(\mathbf{u}_h)\mathbf{u}_h + \mathbf{D}\mathbf{u}_h + \mathbf{f}_h - \mathbf{M}^t \mathbf{p}_h \quad (2.15)$$

where \mathbf{u}_h stands for the discrete velocity vector, Ω is a positive-definite diagonal matrix representing the sizes of the control volumes, the convective coefficient matrix $\mathbf{C}(\mathbf{u}_h)$ is skew-symmetric, the discrete diffusive operator \mathbf{D} is a symmetric negative-definite matrix and \mathbf{M} is the discrete divergence operator. The discrete gradient operator is the transpose of the discrete divergence multiplied by a diagonal scaling $\mathbf{G} = -\Omega^{-1}\mathbf{M}^t$.

The main feature of such discretization is that it preserves the underlying symmetry properties of the continuous differential operators. These global discrete operator properties ensure both stability and that the global kinetic-energy balance is exactly satisfied even for coarse meshes if incompressibility constraint is accomplished [12]. Energy transport equation is also discretized using a symmetry-preserving discretization.

It must be noted that periodic boundary conditions are prescribed in the x -direction because it allows to study the 3D effects due to intrinsic instability of the main flow and not due to the boundary conditions [4, 3]. This case is also more convenient from a computational point of view because the resulting flows are free from boundary layers on the x -direction and therefore, the mesh can be coarser and *uniform* in this direction. This allows to apply Fourier-based methods to solve the Poisson equation.

Poisson solver

The discrete Laplacian operator of the Poisson equation can be viewed as the product of the discrete divergence operator \mathbf{M} by the discrete gradient operator, which is the transpose of the discrete divergence multiplied by a diagonal scaling $\mathbf{G} = -\Omega^{-1}\mathbf{M}$. So, the Laplacian operator is approximated by the matrix product $\mathbf{L} = -\mathbf{M}\Omega^{-1}\mathbf{M}^t$. Therefore the discrete Poisson equation to be solved at each time step is of the form

$$\mathbf{L}\mathbf{p}_h = \mathbf{M}\mathbf{u}_h^p \quad (2.16)$$

Based on a domain decomposition strategy, the parallelisation of the explicit parts of the code is straightforward. However, the efficient solution of the discrete Poisson equations is a critical aspect. Here, a new parallel Poisson solver, named Krylov-Schur-Fourier Decomposition (KSFD) [53], that provides a fairly good scalability for a wide range of problem sizes and computer architectures has been used. It basically consist on combining a preconditioned Conjugate Gradient (CG) method in the two wall-normal directions and an FFT in the periodic direction. Fourier decomposition transform the original three-dimensional system into a set of two-dimensional system that are solved using the CG method preconditioned by the DSD solver [1]. Each plane is decomposed into blocks that are solved with the DSD solver. The number of blocks governs the convergence properties of the algorithm. Therefore, different strategies are used for different two-dimensional problems of the set depending on how well-posed is the system to be solved.

The scalability and flexibility of the KSFD algorithm have allowed us to perform efficiently DNS simulations on both a low-cost PC cluster (for the two lowest Re -number here presented) and the MareNostrum supercomputer. For further details about the KSFD solver the reader is referred to the chapter 1.

2.2.3 Code and simulation verifications

The code was verified using the Method of Manufactured Solutions (MMS), described by Roache [54] and tested for several benchmark reference results. Figure 2.2 displays convergence tests for convective and divergence operators for orders of accuracy from second- up to to eighth-order. In addition, as a symmetry-preserving discretization is being used, the exact fulfilment of the global kinetic energy balance has been used as an additional verification. For more details about the code verification the reader is referred to our previous works [4, 3].

As no subgrid-scale model is used in the computation, the grid resolution and the time step have to be sufficiently fine to solve all the relevant turbulence scales. Moreover, a sufficient length in the periodic direction is required to ensure that turbulence fluctuations are uncorrelated at a separation of one half-period. The time to begin the averaging period t_0 and the time integration period Δt_a have also to be long enough to evaluate the flow statistics properly. Of course, for all these parameters, a compromise between accuracy and computing time must be accepted. The physical and numerical parameters for all cases are given in table 2.1. Grid spacing in the periodic x -direction is uniform and the wall-normal points are distributed using a hyperbolic-tangent function

$$(y)_j = \frac{L_y}{2} \left(1 + \frac{\tanh \{ \gamma_y (2(j-1)/N_y - 1) \}}{\tanh \gamma_y} \right) \quad (2.17)$$

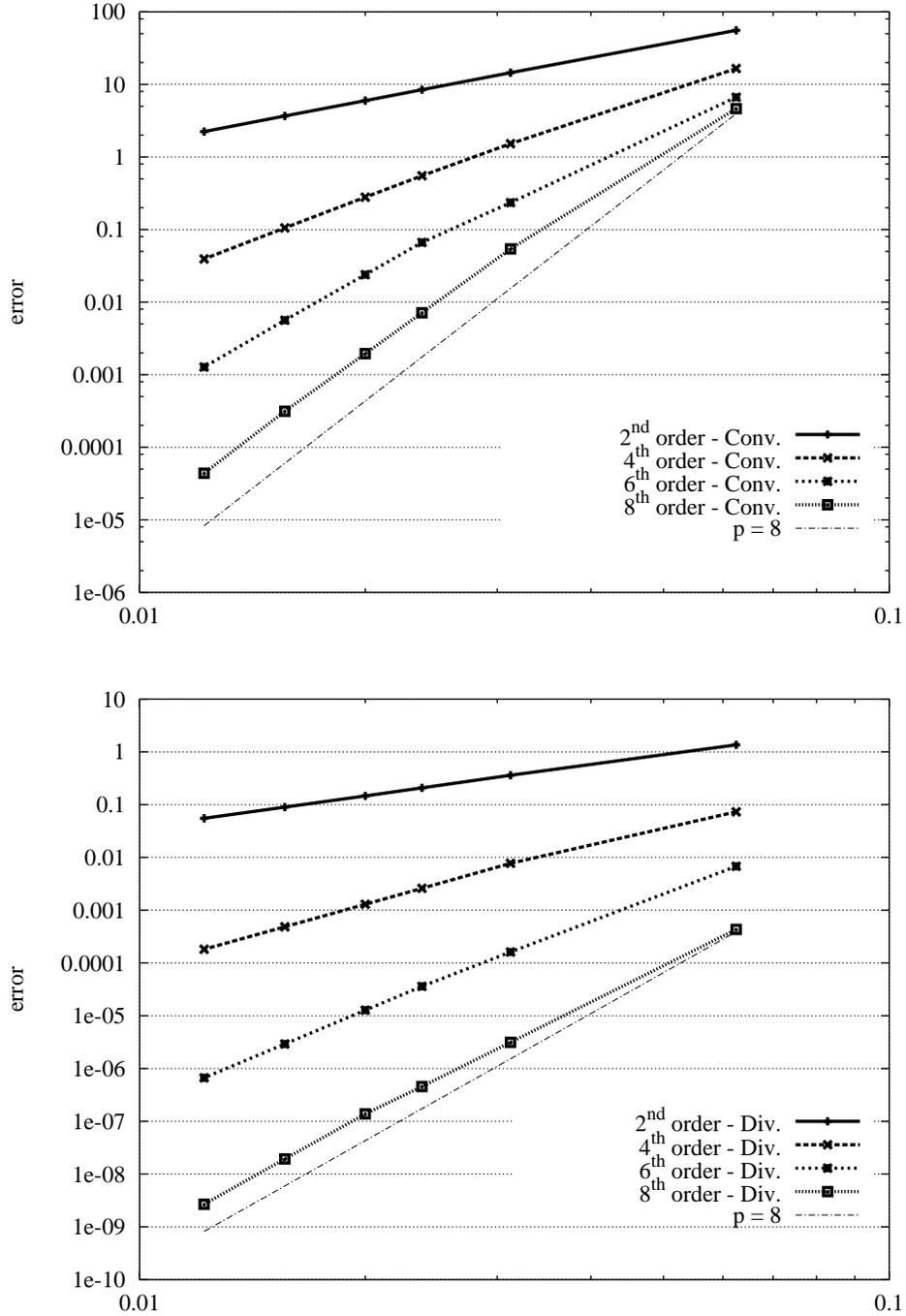


Figure 2.2: Numerical errors versus mesh size for convective (top) and divergence (bottom) operators for second-, fourth-, sixth- and eighth-order discretizations.

Spatial resolution in the two wall-normal directions was determined by means of a systematic procedure based on successive mesh refinements explained by Trias *et al.* [3]. The mesh concentration factors γ_y and γ_z are computed to minimise the flow gradients on the computational space for a set of representative instantaneous maps. Note that grid resolution near the isothermal vertical walls (see table 2.1) is in quite good agreement with the $Ra^{-1/3}$ turbulent scaling².

| Case | Ra | N_x | N_y | N_z | L_x | γ_y | γ_z |
|------|--------------------|-------|-------|-------|-------|------------|------------|
| A | 6.4×10^8 | 128 | 156 | 312 | 2.0 | 1.5 | 1.5 |
| B | 2×10^9 | 64 | 144 | 318 | 1.0 | 1.75 | 0.0 |
| C | 10^{10} | 128 | 190 | 462 | 1.0 | 2.0 | 0.0 |
| D | 3×10^{10} | 128 | 338 | 778 | 0.5 | 2.0 | 0.0 |
| E | 10^{11} | 128 | 682 | 1278 | 0.25 | 2.0 | 0.0 |

| Case | $(\Delta y)_{min}$ | Δt | Total time | Average time | order |
|------|-----------------------|-----------------------|------------|--------------|-----------------|
| A | 2.44×10^{-4} | 1.26×10^{-3} | 1000 | 800 | 2 nd |
| B | 1.88×10^{-4} | 1.27×10^{-3} | 800 | 550 | 4 th |
| C | 9.63×10^{-5} | 6.25×10^{-4} | 440 | 300 | 4 th |
| D | 5.45×10^{-5} | 3.95×10^{-4} | 280 | 180 | 4 th |
| E | 2.71×10^{-5} | 1.76×10^{-4} | 240 | 160 | 4 th |

Table 2.1: Physical and numerical simulation parameters.

Regarding the domain size and grid resolution in the homogeneous direction they need to be adjusted to ensure that the turbulence fluctuations are uncorrelated at a separation of one half-period and the smallest relevant scales are also well-resolved. To do so, spanwise two-point correlations and one-dimensional energy spectra at different (y, z) -locations have been used to check the suitability of the numerical simulation parameters (see table 2.1) in the x -direction. Further details can be found in [3].

2.3 Results and discussion

2.3.1 Time-averaged flow

Averages over the three statistically invariant transformations (time, x_1 -direction and central point symmetry around the centre of the cavity) are carried out for all fields.

²The product $(\Delta y)_{min} Ra^{1/3}$ takes values 0.210, 0.237, 0.207, 0.170 and 0.123 for meshes corresponding to cases A, B, C, D and E, respectively.

| | A | B | C | D | E |
|--------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| \overline{Nu} | 49.24 | 66.63 | 101.94 | 137.22 | 194.48 |
| \overline{Nu}_{max} | 171.89 | 260.49 | 459.50 | 651.12 | 910.10 |
| z | 3.63×10^{-3} | 0 | 0 | 0 | 0 |
| $\sigma (Nu)_{max}$ | 4.16 | 10.92 | 25.35 | 50.71 | 110.54 |
| z | 0.914 | 0.875 | 0.813 | 0.704 | 0.613 |
| $\overline{u}_{y_{max}}$ | 9.02×10^{-4} | 6.36×10^{-4} | 4.08×10^{-4} | 6.11×10^{-4} | 6.81×10^{-4} |
| y | 2.22×10^{-1} | 2.29×10^{-1} | 4.70×10^{-3} | 4.23×10^{-3} | 3.52×10^{-3} |
| $\overline{u}_{z_{max}}$ | 2.22×10^{-1} | 2.22×10^{-1} | 2.23×10^{-1} | 2.29×10^{-1} | 2.24×10^{-1} |
| y | 7.26×10^{-3} | 5.76×10^{-3} | 3.79×10^{-3} | 3.00×10^{-3} | 2.19×10^{-3} |
| $4yRa^{1/4}$ | 4.62 | 4.87 | 4.79 | 4.99 | 4.93 |
| $\overline{u}_{y_{max}}$ | 2.72×10^{-2} | 1.76×10^{-2} | 8.93×10^{-3} | 1.05×10^{-2} | 1.15×10^{-2} |
| z | 9.52×10^{-1} | 9.57×10^{-1} | 9.84×10^{-1} | 9.76×10^{-1} | 9.78×10^{-1} |
| $\overline{u}_{z_{max}}$ | 4.47×10^{-2} | 1.67×10^{-2} | 1.94×10^{-3} | 3.64×10^{-3} | 1.95×10^{-3} |
| z | 6.91×10^{-2} | 6.01×10^{-2} | 9.63×10^{-1} | 2.34×10^{-1} | 3.68×10^{-1} |
| \overline{T}_{max} | 8.91×10^{-1} | 8.93×10^{-1} | 9.09×10^{-1} | 9.03×10^{-1} | 8.67×10^{-1} |
| z | 9.85×10^{-1} | 1 | 1 | 9.98×10^{-1} | 9.98×10^{-1} |

Table 2.2: Summary of the averaged flow results. By rows, from top to bottom, the magnitudes are: the overall averaged Nusselt number, the maxima of the averaged local Nusselt and the standard deviation of local Nusselt and their respective z positions at the vertical hot wall, the maxima of \overline{u}_z and \overline{T} at the horizontal mid-height plane and their respective y positions, the maxima of \overline{u}_y and \overline{T} at the vertical mid-width plane and their respective z positions.

The time-averaged temperature fields and the streamlines of the averaged flow have been represented in the figure 2.3. A summary of several first-order statistics is presented in table 2.2 for direct comparison.

Despite the relatively large range of Rayleigh numbers all them exhibit similar flow characteristics: thin vertical boundary layers and a large core area with very low time-averaged velocity and a stratified temperature distribution.

The waves travelling downstream grow up to a point where they disrupt the boundary layers ejecting large unsteady eddies to the core of the cavity. The mixing effect of these eddies, that throw hot and cold fluid respectively, tends to result in almost isothermal hot upper and cold lower regions. The point where this phenomenon occurs moves upstream of the boundary layer when the Rayleigh number is increased. This mixing effect at the top and bottom areas of the cavity, clearly displayed in the time-averaged solutions (figure 2.3 and 2.4), force the temperature drop in the core

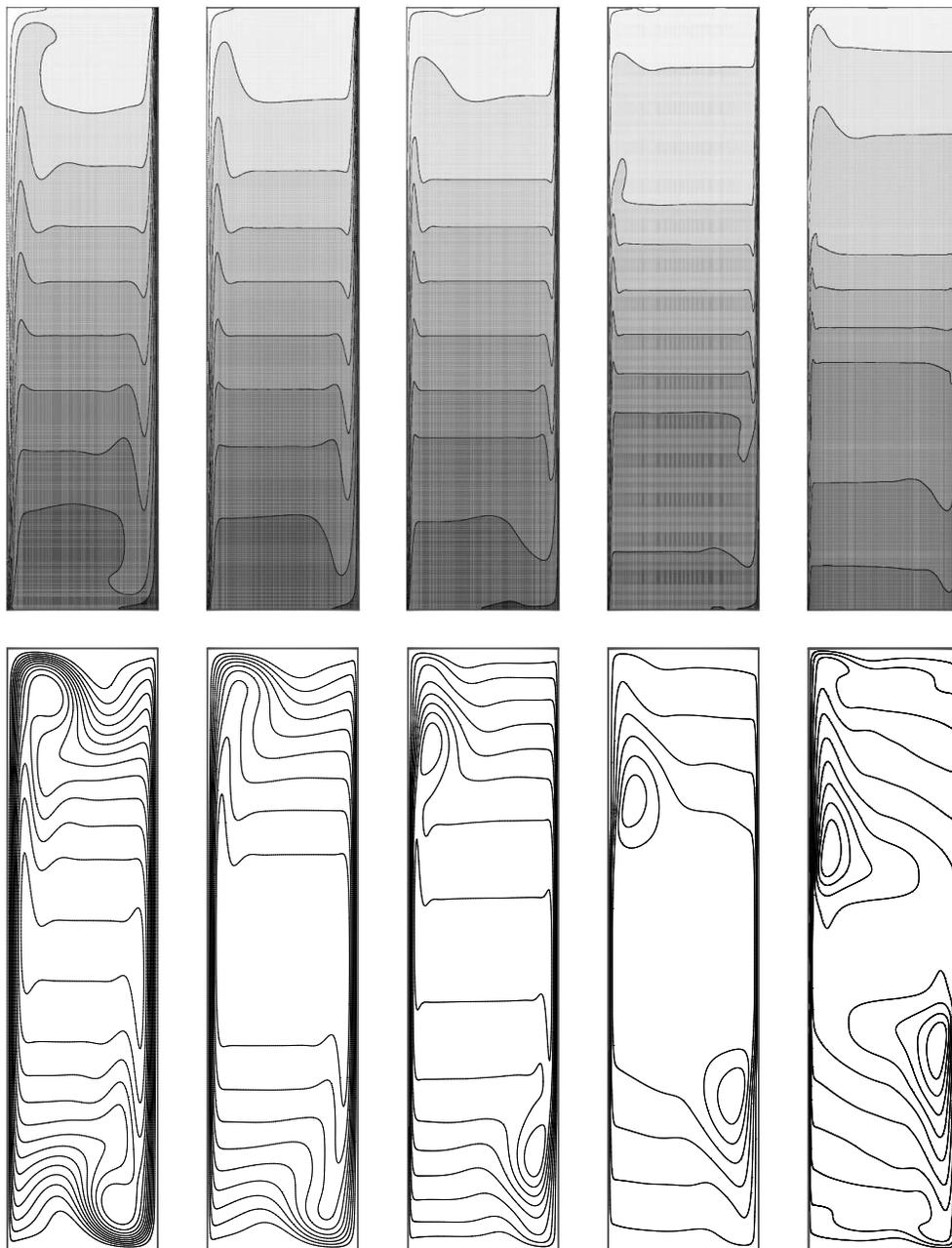


Figure 2.3: Averaged solutions. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} . Top: averaged temperature field. Bottom: streamlines of the averaged flow. For temperature fields, the isotherms are uniformly distributed from 0 to 1.

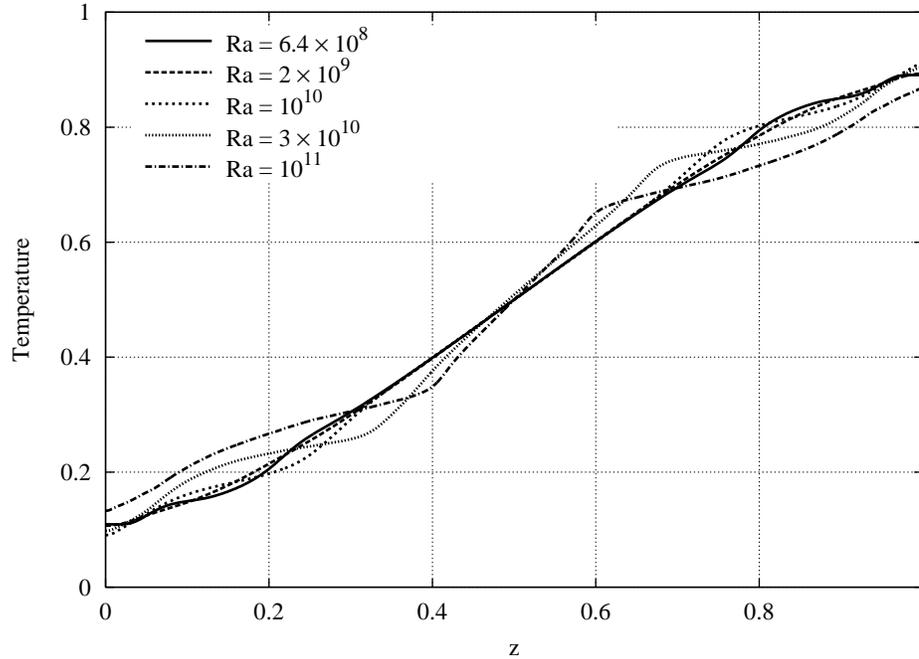


Figure 2.4: Averaged vertical temperature profiles at mid-width.

of the cavity occurs in a smaller region. The strengthening of the horizontal motion by means of large unsteady eddies also results into a remarkable thickening of the boundary layer in the downstream part and consequently a sudden decrease of the vertical velocity (figure 2.6) and a reduction of the wall-shear stress (figure 2.5).

The averaged temperature and vertical velocity component profiles displayed in the figure 2.6, show that velocity maxima remain constant (see also table 2.2) and that identical profiles are obtained for more than half vertical boundary layer when the lengths are scaled by the laminar $Ra^{1/4}$ factor. This laminar scaling is also observed in the table 2.2 for the position of its maxima. Discrepancies only occur from the point where waves travelling downstream grow large enough to totally disrupt the boundary layer. This results confirms that in the range of Rayleigh numbers investigated at least the downstream half vertical boundary layer, where most of the heat transfer occurs, is still laminar or quasi-laminar.

Thermal stratification in the core of the cavity is one of the basic questions that still remains open. Comparison between numerical and experimental results (see Salat

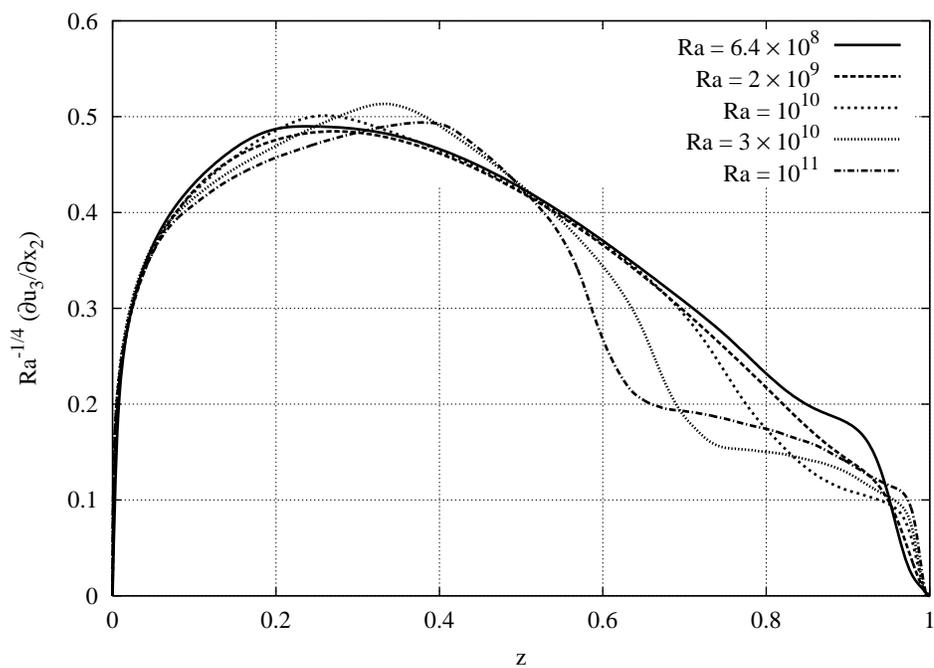


Figure 2.5: Dimensionless time-averaged vertical wall shear stress scaled by $Ra^{-1/4}$.

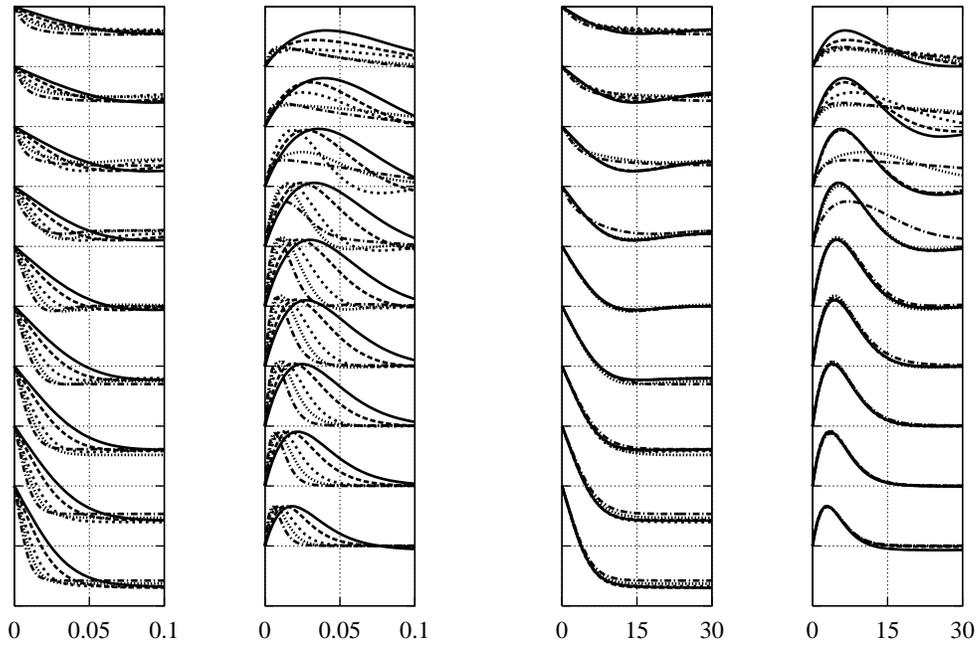


Figure 2.6: Averaged temperature (left-hand side of pair) and vertical velocity (right-hand side of pair) profiles at $z = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$. Each vertical subdivision represents 0.5 units for temperature and 0.2 units for vertical velocity. For the pair of plots on the left, the abscissa scale factor is $4y$ and for the pair on the right, is $4yRa^{1/4}$.

| Case | Ra | \overline{Nu} | $\overline{Nu}/Ra^{1/4}$ | $\overline{Nu}/Ra^{1/3}$ | \overline{Nu}_{down} | $\overline{Nu}_{down}/Ra^{1/3}$ |
|------|--------------------|-----------------|--------------------------|--------------------------|------------------------|---------------------------------|
| A | 6.4×10^8 | 49.24 | 0.3096 | 0.05713 | 2.98 | 3.45×10^{-3} |
| B | 2×10^9 | 66.63 | 0.3151 | 0.05288 | 4.54 | 3.60×10^{-3} |
| C | 10^{10} | 101.94 | 0.3224 | 0.04732 | 7.91 | 3.67×10^{-3} |
| D | 3×10^{10} | 137.22 | 0.3297 | 0.04416 | 10.77 | 3.47×10^{-3} |
| E | 10^{11} | 194.48 | 0.3458 | 0.04190 | 18.70 | 4.03×10^{-3} |

Table 2.3: Nusselt number and correlations

et al. [55] for a detailed overview) for a wide range of width/height aspect ratios give completely different results. Experimental studies yield a dimensionless stratification of about 0.5 while numerical simulations predict values about 1. According to the results obtained by Salat *et al.* [55], Soria *et al.* [4] and Trias *et al.* [3] the two-dimensional assumption is not a critical issue to explain these differences. Salat *et al.* [55] also concluded that introducing experimental temperature measurements in the top and bottom wall instead of assuming adiabaticity hypothesis do not improve the discrepancies in the thermal stratification.

Our DNS results for the two highest Rayleigh numbers shows that thermal stratification clearly tends to increase with the Rayleigh number: 1.25 and 1.41 for $Ra = 3 \times 10^{10}$ and 10^{11} , respectively. This phenomenon seems to be directly related with the displacement of the transition point of the vertical boundary layer to more downstream positions.

2.3.2 Heat transfer

Results for the time-averaged and spatial mean Nusselt numbers³ are given in table 2.3. It is shown that Nusselt number correlation is much closer to the $Ra^{1/4}$ correlation for laminar flow than the $Ra^{1/3}$ correlation for turbulent flow. The reason for such behaviour is that most of the heat transfer occurs in the upstream part of the boundary layer where it is almost laminar. To confirm this point we have computed the mean Nusselt number at the most downstream part where the boundary layer becomes turbulent. In the last column of table 2.3, we see that the \overline{Nu}_{down} , that has been integrated from $z = 0.8$ to $z = 1$ over the hot sidewall, is very close to the classical $Ra^{1/3}$ turbulent scaling. This confirms that at the most downstream part boundary layers become turbulent.

The averaged local Nusselt distribution and their standard deviations are displayed

³The reference heat flux is given by $\lambda\Delta T/L_z$, where λ is the thermal conductivity. Thus, the dimensionless local Nusselt number at the vertical hot wall is given by $Nu = -\left.\frac{\partial T}{\partial y}\right|_{y=0}$

in figure 2.7. Large fluctuations are observed in the downstream part of the boundary layer whereas upstream remains almost laminar. Trias *et al.* [3] observed that this region of large fluctuations seems to remain almost invariant slightly above $z = 0.6$ position while the peak moves upstream for Rayleigh number up to 10^{10} . Here, for the two highest Rayleigh numbers, it is observed that region of large fluctuations is extended to more downstream position.

Nusselt number correlation

An approximation of the Nusselt number can be presented in the following form

$$Nu = c_1 (1 - z^{Tr}) Ra^{1/3} + c_2 z^{Tr} Ra^{1/4} \quad (2.18)$$

where z^{Tr} is the point where transition occurs at the vertical boundary layer. Doing so, we are splitting vertical boundary layer into a downstream laminar part that follows the $Ra^{1/4}$ correlation and an upstream turbulent part that follows the $1/4$ power law. However, equation (2.18) needs to know the value of the transition point z^{Tr} . In order to circumvent this, we can replace z^{Tr} by the following power law form,

$$z^{Tr} = 1 - aRa^b \quad (2.19)$$

Using least squares, constants a and b (with the data obtained in the present work) were determined to be 2.2636×10^{-4} and 2.9462×10^{-1} , respectively. Now, using least square criteria again, constants c_1 and c_2 are found out to be 4.6580×10^{-2} and 3.2148×10^{-1} , respectively with a maximum error of around 1.8% in the investigated range. Hence, proposed correlation (2.18) finally results into

$$Nu = c_1 a Ra^{(1/3+b)} + c_2 Ra^{1/4} - c_2 a Ra^{(1/4+b)}, \quad 6.4 \times 10^8 \leq Ra \leq 10^{11} \quad (2.20)$$

where,

$$\begin{aligned} c_1 &= 4.6580 \times 10^{-2} & c_2 &= 3.2148 \times 10^{-1} \\ a &= 2.2636 \times 10^{-4} & b &= 2.9462 \times 10^{-1} \end{aligned} \quad (2.21)$$

2.3.3 Flow dynamics

A general view of the several instantaneous temperature fields is displayed in the figures 2.8, 2.9. At first sight, we can see that when the Ra number is increased important differences are observed. With the aim to investigate the origin of such differences and the physical mechanisms involved, several flow dynamics aspects are analyzed and discussed in the next sections.

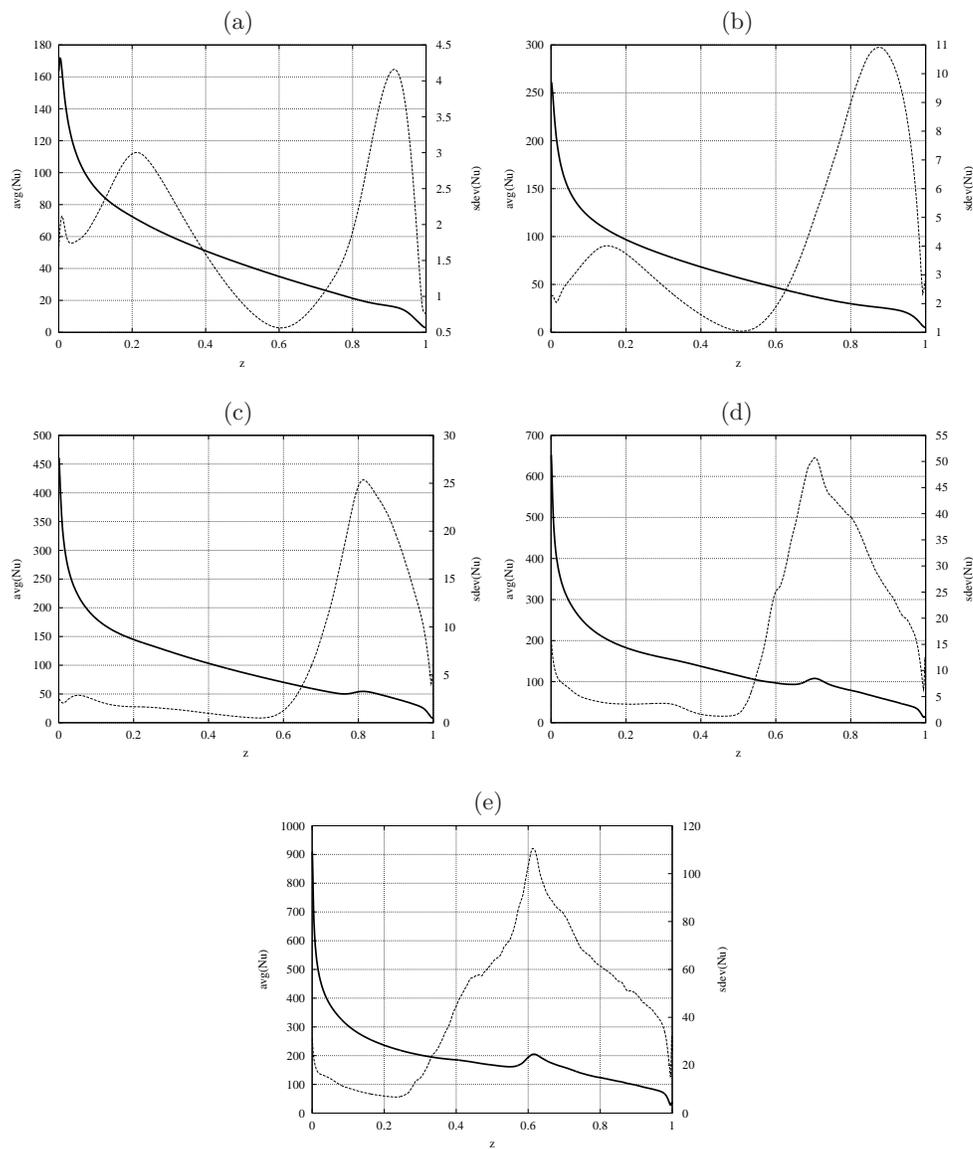


Figure 2.7: Local Nusselt number distribution (solid lines) and their standard deviation (dashed lines): (a) $Ra = 6.4 \times 10^8$, (b) $Ra = 2 \times 10^9$, (c) $Ra = 10^{10}$, (d) $Ra = 3 \times 10^{10}$ and (e) $Ra = 10^{11}$.

Instantaneous fields

Flow dynamics of the three lowest Rayleigh numbers considered here were analyzed in detail by Trias *et al.* [3]. They focussed not only on the Ra -number dependence but also on the influence of the two-dimensional assumption. As expected, significant differences were observed between two- and three-dimensional results. For two-dimensional simulations the oscillations at the downstream part of the vertical boundary layer are clearly stronger, ejecting large eddies to the cavity core. In the three-dimensional simulations these large eddies do not persist and their energy is rapidly passed down to smaller scales of motion. It yielded on a reduction of the large-scale mixing effect at the hot upper and cold lower regions and consequently the cavity core still remains almost motionless. Regarding the boundary layers it was observed that the transition point clearly moves upstream for the three-dimensional simulations. It was observed that all these differences become more marked for higher Rayleigh numbers. For further details about the discrepancies between two- and three-dimensional results and the physical mechanisms involved the reader is referred to our previous works [4, 3].

Here we focus our attention on the dependence of the results respect to the Ra -number. In figure 2.8 it is clearly observed that boundary layers remain laminar in their upstream parts up to the point where they become totally disrupted and large-eddies are ejected. Although the position of this point tends to move to upstream position

Flow dynamics if the lowest Rayleigh number considered here was analyzed in detail by Soria *et al.* [4] concluding that both two- and three-dimensional configurations have a motionless and stratified cavity core and concentrate important fluctuations in the two downstream corners of the cavity. The main differences occur in the vertical boundary layers. For the two-dimensional simulation it is almost totally stable and only periodic oscillations can be observed in the most downstream part of the boundary layer while in the three-dimensional simulation instabilities generated in the upstream corner moves downstream resulting on considerable values for turbulent statistics at the vertical boundary layer, as discussed in next sections. Also, in the two-dimensional results, the vortices at the end of the vertical boundary layers are more vigorous and stable, as can also be appreciated in the streamline maps displayed in the figure 2.3 (bottom left). Occasionally, in the three-dimensional simulation, there are large instability episodes where the three-dimensional structures generated at the top right and bottom left areas of the cavity propagate across all the vertical boundary layers. However, this phenomenon (that has not been observed in the two-dimensional results) is too infrequent to generate significant values of $\overline{u'_x u'_x}$ and $\overline{u'_y u'_y}$ at the vertical boundary layers.

Finally figure 2.9 shows 3D structures of the flow with highest Rayleigh number.

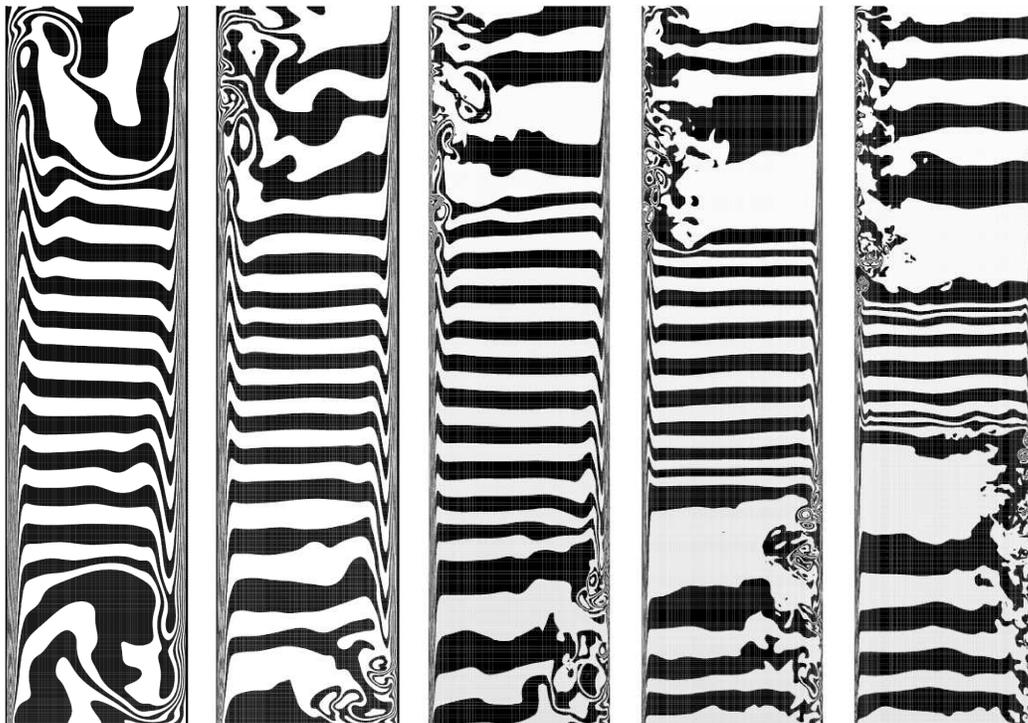


Figure 2.8: Representative instantaneous isotherms. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} . The isotherms are uniformly distributed from 0 to 1.

| Case | Ra | C | | f_{BV} |
|------|--------------------|-------|------|----------|
| A | 6.4×10^8 | 0.131 | 1.02 | 0.135 |
| B | 2×10^9 | - | 1.01 | 0.135 |
| C | 10^{10} | 0.133 | 1.01 | 0.135 |
| D | 3×10^{10} | 0.141 | 1.25 | 0.150 |
| E | 10^{11} | 0.147 | 1.41 | 0.159 |

Table 2.4: From left to right: fundamental frequencies of the Nu_c , dimensionless stratification in the core of the cavity and Brunt-Väisälä frequency

Top of the figure is a time evolution of the flow as it develops to the statistically stationary state. Bottom of the figure is zoom to turbulent transition area.

Internal waves

The velocities in the cavity core become much smaller compared to those in the vertical boundary layers for increasing Rayleigh number. However, simulations show that the cavity core is in motion and isotherms in this region oscillate around the mean horizontal profile. Since the cavity core remains well stratified (see figure 2.3 and 2.4), this phenomenon can be attributed to internal waves. All the normalized density power spectra of the mean Nusselt number through the vertical midplane, Nu_c , displayed in figure 2.10, are quite similar. In table 2.4, these fundamental oscillation frequencies are compared with the dimensionless Brunt-Väisälä frequencies: $f_{BV} = (CPr)^{0.5}/(2\pi)$ (see Lighthill [56], for example), where C is the dimensionless stratification of the time-averaged temperature. A fairly good agreement is observed for the range of Rayleigh numbers investigated. Following the same arguments exposed in [3], we conclude that eddies ejected from the vertical boundary layers at the hot upper and cold lower regions are large enough, even for the highest Rayleigh number here considered, to be characterized by excitation frequencies ω smaller than the Brunt-Väisälä frequency and maintain an internal wave of motion.

Turbulent statistics

The distributions of turbulent kinetic energy $k = \overline{u'_i u'_i}$, its total viscous dissipation rate $\epsilon_\nu = \overline{(\nabla \mathbf{u}' + (\nabla \mathbf{u}')^t) : \nabla \mathbf{u}'}$, the temperature variance $\overline{T'T'}$, the turbulent heat flux $\overline{u'_3 T'}$ and two of the four non-zero components of the Reynolds stress have been represented in figures 2.11- 2.16. It should be noted that, as the domain is homogeneous in the x_1 -direction, all the statistics are independent of x_1 , $\overline{u_x} = 0$, and the fluctuations of u_x are uncorrelated with the fluctuations of other variables, i.e.,

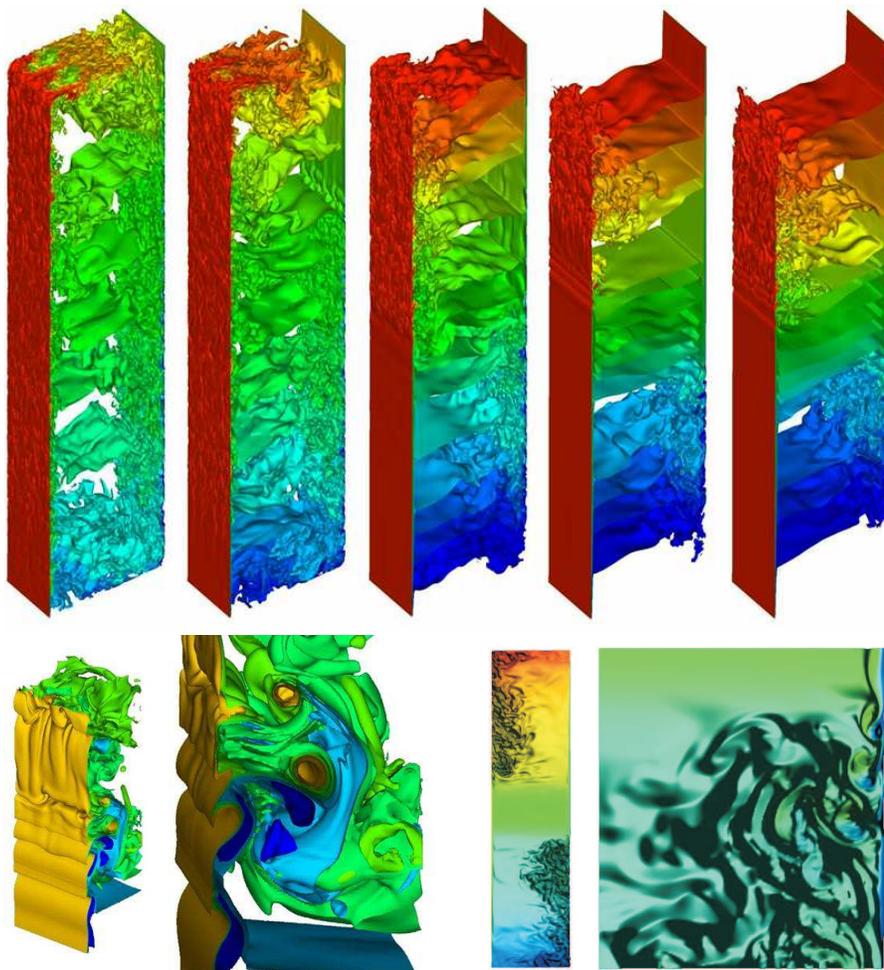


Figure 2.9: Representative instantaneous temperature iso-surfaces for highest Ra number 10^{11} . Time evolution of the flow to statistically stationary state (top), Zoom view of the turbulent transition area (bottom left), 2D section with zoom to transition area (bottom right)

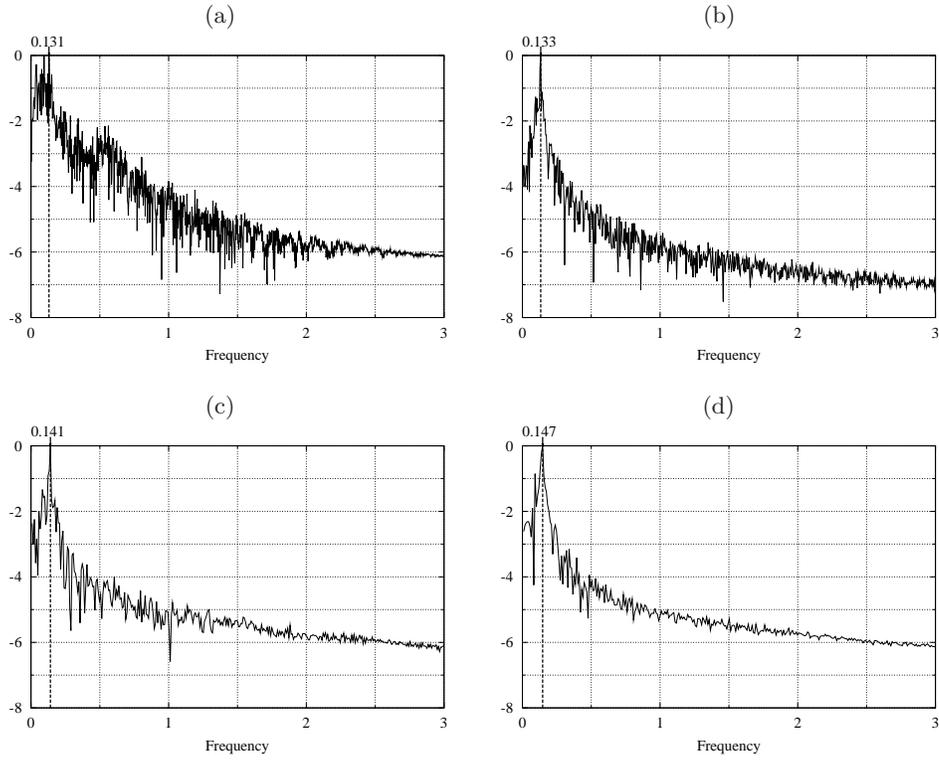


Figure 2.10: Normalized density power spectra of Nusselt number at the vertical mid-plane: (a) $Ra = 6.4 \times 10^8$, (b) $Ra = 10^{10}$, (c) $Ra = 3 \times 10^{10}$ and (d) $Ra = 10^{11}$.

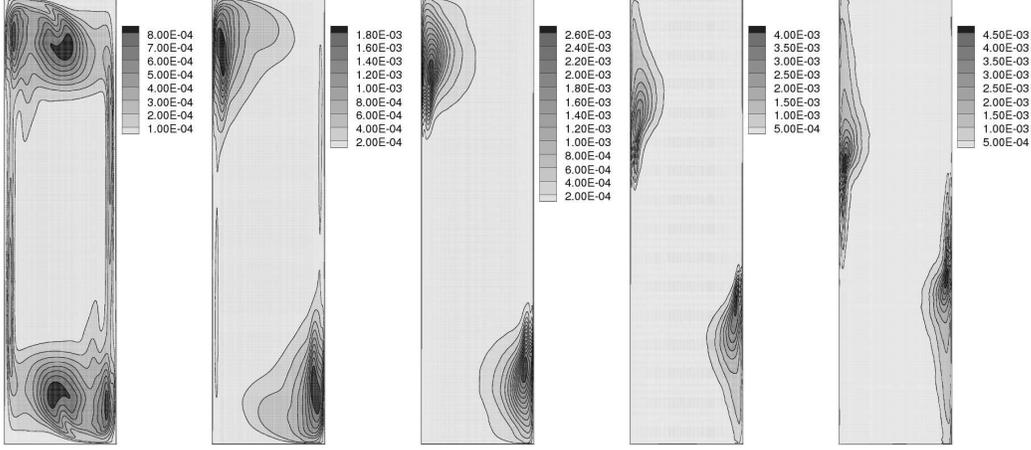


Figure 2.11: Distribution of turbulent kinetic energy k . From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} .

$\overline{u'_x u'_y} = \overline{u'_x u'_z} = \overline{u'_x T'} = 0$, except statistical noise that decreases with the integration period.

For the range of Rayleigh numbers here investigated we observe a bad correlation between two important quantities for turbulence modelling such as k and ϵ_ν . Near the vertical walls, the non-slip boundary conditions for velocity make $\partial u'_z / \partial x_2$ be the leading term of the viscous dissipation rate ϵ_ν and its maxima be located very close to the wall while k is essentially contributed by $\overline{u'_y u'_y}$ whose maxima is located outside the boundary layer. The horizontal profiles at $z = 0.8$, a region with high values of turbulent statistics, displayed in the figure 2.3.3 show more clearly that these two quantities are uncorrelated near the vertical wall where turbulence is more intense.

Energy budgets

The global kinetic energy balance per volume unit is given by (see [3], for details)

$$\underbrace{\frac{Pr}{V} \int_{\Omega} (\overline{u_z \overline{T}} + \overline{u'_z T'}) d\Omega}_{\overline{E_g}} = \underbrace{\frac{Pr}{V Ra^{0.5}} \int_{\Omega} (\phi(\overline{\mathbf{u}}) + \overline{\phi(\mathbf{u}')}) d\Omega}_{\overline{E_d}} \quad (2.22)$$

where V is the cavity volume and $\phi(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^t) : \nabla \mathbf{u}$. That is, for a statis-

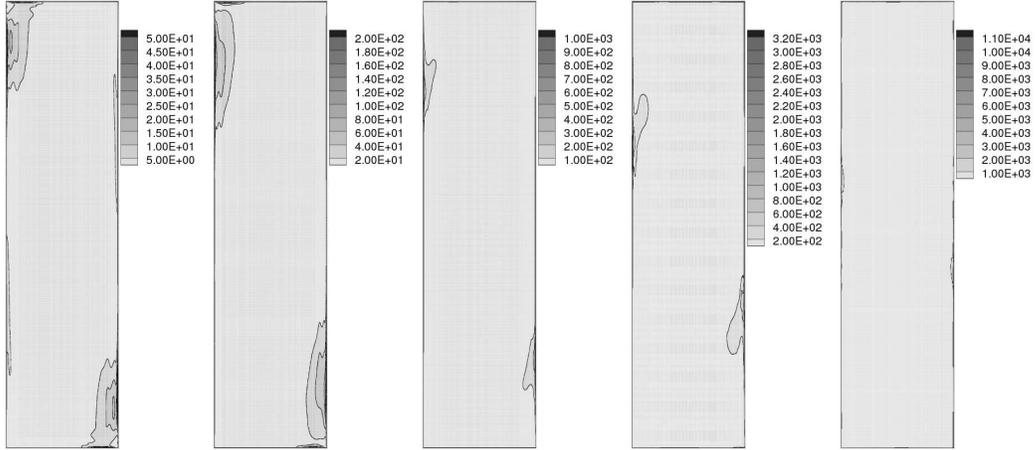


Figure 2.12: Distribution of turbulent viscous dissipation ϵ_ν . From left to right: $Ra = 6.4 \times 10^8, 2 \times 10^9, 10^{10}, 3 \times 10^{10}$ and 10^{11} .

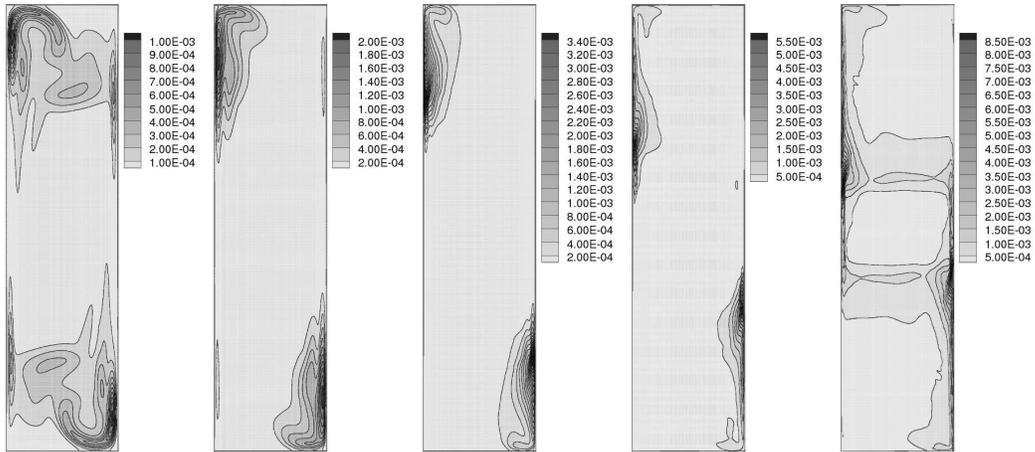


Figure 2.13: Distribution of temperature variance $\overline{T'T'}$. From left to right: $Ra = 6.4 \times 10^8, 2 \times 10^9, 10^{10}, 3 \times 10^{10}$ and 10^{11} .

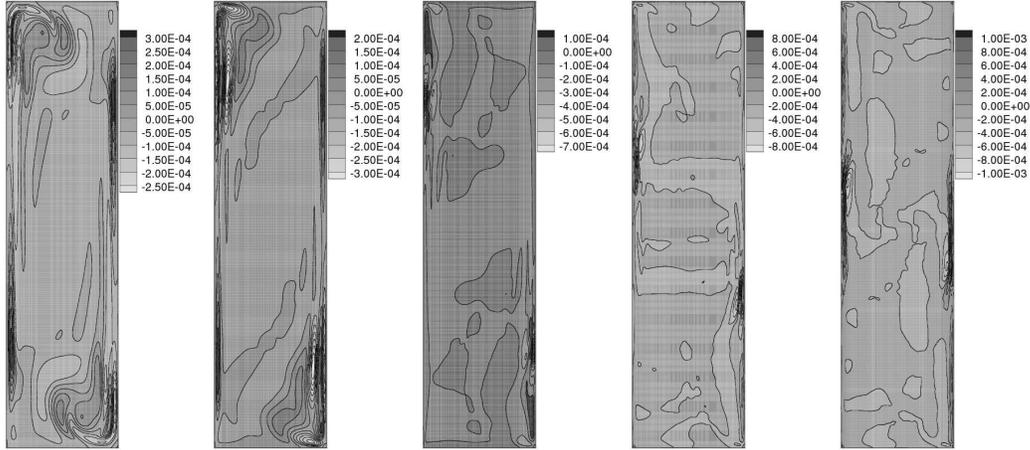


Figure 2.14: Distribution of the turbulent heat flux $\overline{u'_z T'}$. From left to right: $Ra = 6.4 \times 10^8, 2 \times 10^9, 10^{10}, 3 \times 10^{10}$ and 10^{11} .

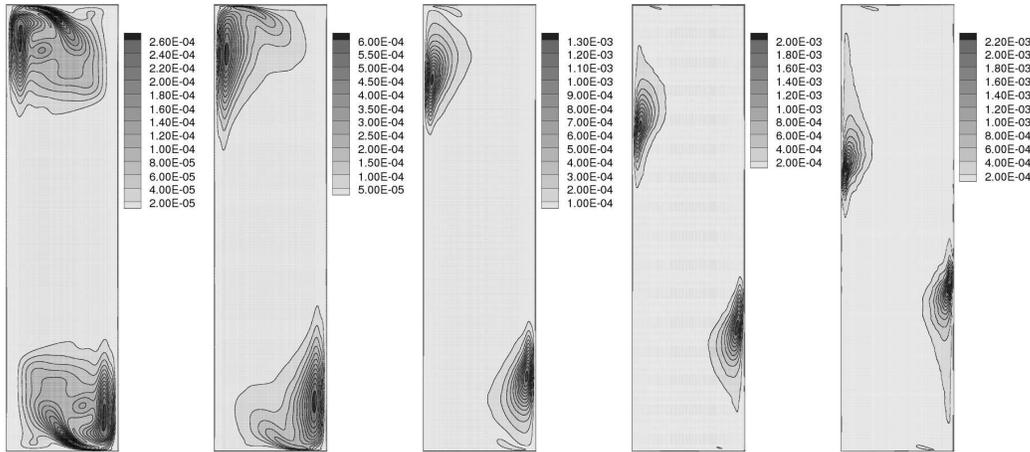


Figure 2.15: Distribution of the Reynolds stress component $\overline{u'_y u'_y}$. From left to right: $Ra = 6.4 \times 10^8, 2 \times 10^9, 10^{10}, 3 \times 10^{10}$ and 10^{11} .

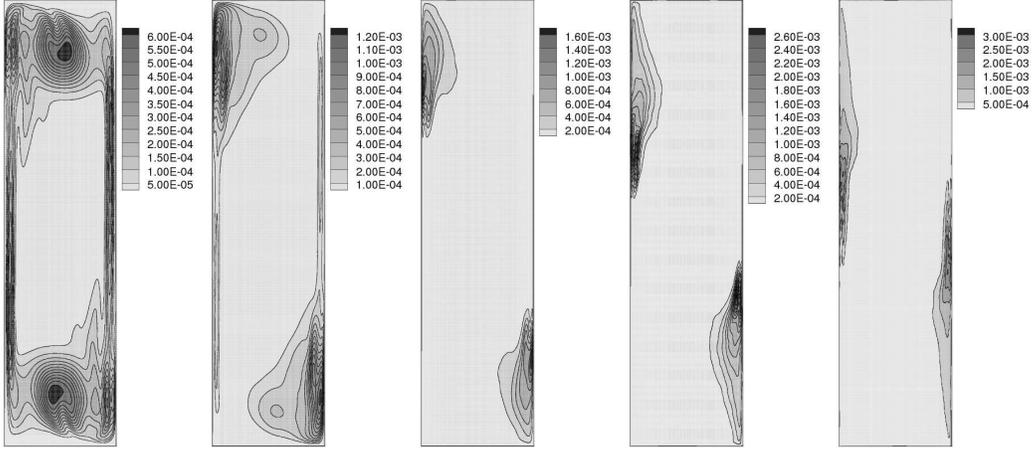


Figure 2.16: Distribution of the Reynolds stress component $\overline{u'_z u'_z}$. From left to right: $Ra = 6.4 \times 10^8$, 2×10^9 , 10^{10} , 3×10^{10} and 10^{11} .

tically stationary flow, \overline{E}_g , the averaged kinetic energy generation rate (only due to the buoyancy forces in our case) must be equal to \overline{E}_d , the averaged kinetic energy dissipation rate due to viscous forces. Since the instantaneous kinetic energy balances are exactly satisfied, the energy imbalance expression $|\overline{E}_g - \overline{E}_d|/\overline{E}_g$ can be used to control if averaging time is long enough. The values of \overline{E}_g obtained in the different simulations can be found in table 2.5. At first sight, we see that the dimensionless overall kinetic energy generation rate \overline{E}_g tends to decrease with the Rayleigh number following a correlation closer to $Ra^{-1/4}$, at least for the three lowest Ra -numbers. At section 2.3.1, we saw that time-averaged temperature and vertical velocity profiles for different Rayleigh numbers collapse when the laminar $Ra^{1/4}$ scaling is used for lengths (see figure 2.6). We also saw that such laminar behaviour is exhibit in the upstream part of the boundary layers, where most of the kinetic energy is generated, and extends for more than half of the cavity. Thus, it is not surprising the $Ra^{1/4}$ scaling shown by the overall kinetic energy generation rate \overline{E}_g . Regarding to the two highest Rayleigh number such correlation seems to be lost probably due to the fact, as we saw in previous sections, that the transition point clearly moves upstream resulting on a smaller region of laminar or quasi-laminar behaviour.

With respect the turbulent dissipation term, $PrRa^{-0.5}/V \int_{\Omega} \overline{\phi(\mathbf{u}')} d\Omega$, there is not a clear tendency. Its peak tends to grow with the Ra -number but the area of large values shrinks to the two vertical boundary layers (see figure 2.12). Finally, turbulent generation term, $Pr/V \int_{\Omega} \overline{u'_z T'} d\Omega$ displays completely different tendencies.

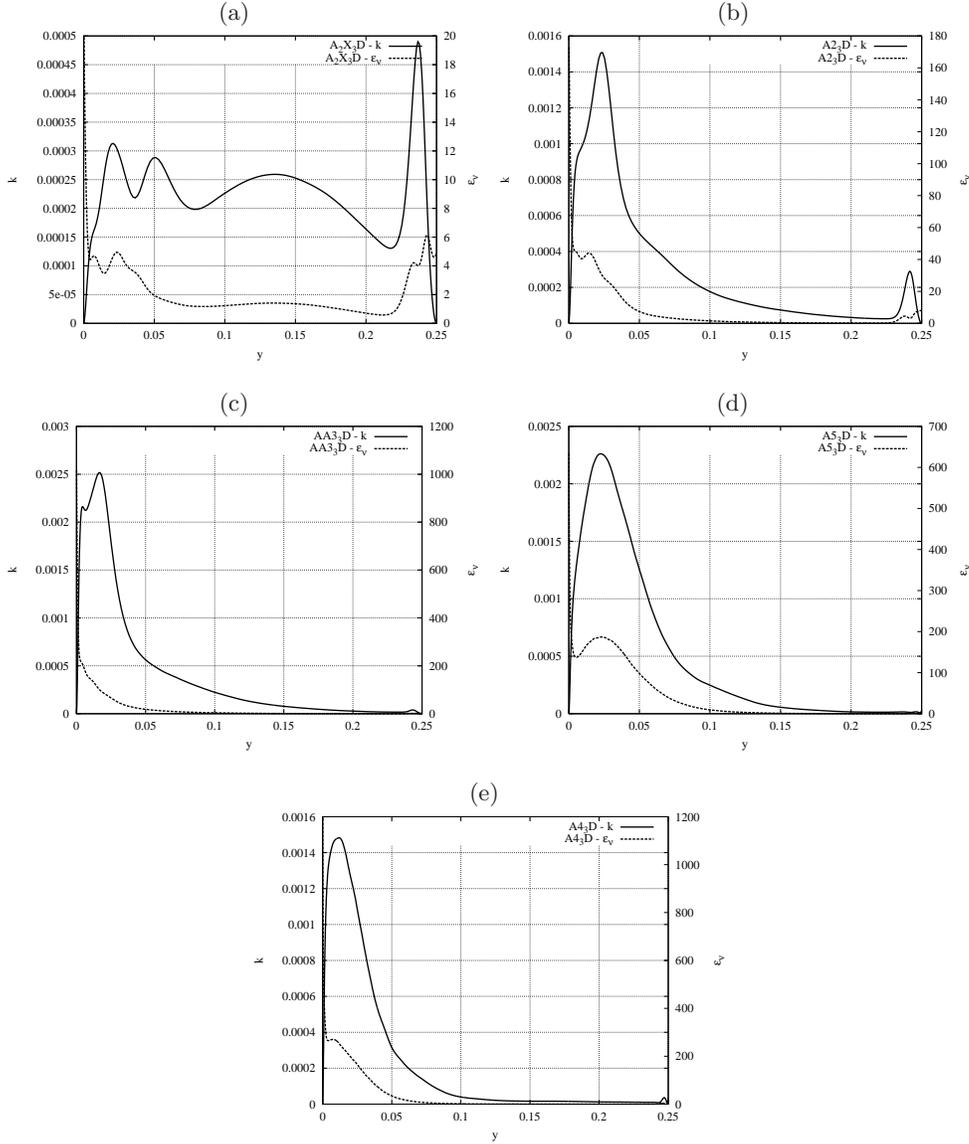


Figure 2.17: Horizontal profiles at $z = 0.8$ of the turbulent kinetic energy k (solid line) and its viscous dissipation rate ϵ_ν (dashed line): (a) $Ra = 6.4 \times 10^8$, (b) $Ra = 2 \times 10^9$, (c) $Ra = 10^{10}$, (d) $Ra = 3 \times 10^{10}$ and (e) $Ra = 10^{11}$.

| Case | Ra | $\overline{E}_g = \overline{E}_d$ | $\frac{Pr}{V Ra^{0.5}} \int_{\Omega} \overline{\phi(\mathbf{u}')} d\Omega$ | $\frac{Pr}{V} \int_{\Omega} \overline{u'_z T'} d\Omega$ | $\overline{E}_g Ra^{1/4}$ |
|------|--------------------|-----------------------------------|--|---|---------------------------|
| A | 6.4×10^8 | 1.996×10^{-3} | 5.33×10^{-5} | 4.71×10^{-6} | 0.317 |
| B | 2×10^9 | 1.524×10^{-3} | 8.08×10^{-5} | 2.36×10^{-6} | 0.322 |
| C | 10^{10} | 1.023×10^{-3} | 6.65×10^{-5} | -5.48×10^{-7} | 0.324 |
| D | 3×10^{10} | 8.205×10^{-4} | 9.58×10^{-5} | 2.76×10^{-6} | 0.341 |
| E | 10^{11} | 6.647×10^{-4} | 1.29×10^{-4} | 1.13×10^{-5} | 0.374 |

Table 2.5: Global kinetic energy balances

2.4 Conclusions

A set of complete direct numerical simulations of a buoyancy driven flow in a differentially heated air-filled ($Pr = 0.71$) cavity of aspect ratio 4 and Rayleigh numbers up to 10^{11} has been presented. The correctness of the code has been verified using the method of manufactured solutions, that ensures that the order of accuracy is in good agreement with the theoretical expectation in the whole domain. Both a conventional PC cluster and the MareNostrum supercomputer have been used to carry out the simulations. An explicit scheme has been used for temporal integration and second and fourth-order symmetry-preserving spatial discretization. These schemes preserve the global kinetic energy balances even for very coarse meshes. The parallel algorithm is based on spatial domain decomposition. A new parallel solver (KSFD) has been used for the Poisson equations. The main features of the flow, including the time-averaged flow structure, the turbulent statistics, the global kinetic energy balances and the internal waves motion phenomena, have been presented and described.

All simulations share some basic flow features: a stratified cavity core, recirculating structures near the downstream corners and thin vertical boundary layers that remain laminar at their upstream part up to a point above the mid-height where transition occurs. Periodic oscillations are amplified in the boundary layer and trigger non-linear effects provoking the transition. With respect to the centreline dimensionless thermal stratification, all the numerical simulations performed in this work give values close to 1 for the three lowest Ra -numbers and larger values (1.25 and 1.41 for the two highest Ra -numbers, respectively).

Acknowledgements

Calculations have been performed on the JFF cluster at the CTTC and on the IBM MareNostrum supercomputer at the Barcelona Supercomputing Center. The author thankfully acknowledge these institutions.

Chapter 3

Extension of the KSFD Poisson solver for fully-3D DNS of turbulent flows on supercomputers

Main contents of this chapter have been published in:

A. Gorobets, F. X. Trias, M. Soria, C. D. Perez-Segarra and A. Oliva On the extension of the Krylov-Schur-Fourier Decomposition Poisson solver for fully-3D DNS of turbulent flows on supercomputers, *Parallel CFD 2008 Conference*.

Abstract.

Many important applications in the computational fluid dynamics (CFD) field demand huge computing power and need parallel computers to be feasible. The Poisson equation, which arises from the incompressibility constraint and has to be solved at least once at each time step, is usually the main bottleneck from a parallel point of view. In this context, efficient and scalable algorithms for the solution of the Poisson equation on a wide range of parallel systems are of high interest.

In the chapter 1 a scalable algorithm for Poisson equation was proposed. It performs well on both small clusters with poor network performance and supercomputers using efficiently up to at least a thousand of CPUs. This algorithm named Krylov-Schur-Fourier Decomposition (KSFD) can be used for problems in parallelepipedic 3D domains with structured meshes and several obstacles can be placed inside the flow. However, since a FFT decomposition is applied in one direction mesh is restricted to be uniform in this direction an only 2D extruded obstacles can be placed.

This chapter is devoted to extend the previous algorithm to eliminate these limitations. The most promising results are obtained when combining a two-level multigrid (MG) and the KSFD method. Finally, illustrative DNS results of a turbulent channel flow with a mounted cube are presented.

3.1 Introduction

Direct numerical simulation (DNS) has become an important area of contemporary fluid dynamics, because its interest for improving the understanding of the physics of turbulence and because it is an essential tool for the development of better turbulence models. Recently, relevant improvements on turbulence modelling based on symmetry-preserving regularization models for the convective (non-linear) term have been developed [57, 58, 59]. Since now, they have been successfully tested for a differentially heated cavity at high Rayleigh numbers [58, 59]. At this stage, high resolution DNS results at of relatively complex geometries and configurations are of extreme importance for further progress. The main idea behind this is to assess the validity of turbulence models in more realistic configurations, understand their limitations and finally improve them. Therefore, this is really a crucial issue since turbulence modelling ultimately becomes an essential tool for engineering applications. In this context the availability of efficient and scalable Poisson solvers for fully-3D geometries are of extreme importance.

The DNS can be classified by the flow complexity represented by the number of wall-bounded directions. Typical DNS examples are represented in table 3.1

| Wall-orthogonal directions | DNS example | Mesh size |
|----------------------------|---------------------------|--------------------------|
| 0 | Isotropic turbulence | $6.8 * 10^{10}$ [60] |
| 1 | Channel flow | $2.0 * 10^{10}$ [61] |
| 2 | Natural convection in DHC | $1.1 * 10^8$ (chapter 2) |
| 3 | Surface mounted cube | about 10^7 [62] |

Table 3.1: Typical DNS cases

The isotropic turbulence and channel flow are the most well studied cases solved with extremely fine meshes. While cases with 2 and more wall-orthogonal directions requires much more computational cost per node due to the limitations and inapplicability of the fast numerical methods.

In this context, efficient and scalable Poisson solvers applicable to the DNS of fully-3D flows with 3 wall-orthogonal directions are of high interest. Verification of the turbulence models on this class of problems especially for high Reynolds numbers could be an important step in development of LES methods.

In the chapter 1 the Poisson solver is proposed. The KSFD solver is based on combination of FFT and Conjugate Gradient (CG) method [16] preconditioned with a direct Schur Decomposition (DSD) method [1]. The Fourier decomposition is used to uncouple the original 3D Poisson equation into a set of 2D planes. Then, each 2D problem in solved using a CG method preconditioned by a DSD algorithm. To

do that, each plane is divided into *blocks* and each of them is solved with the DSD solver. However, the use of the FFT has the following restrictions:

- Mesh must be uniform in the direction where FFT is applied.
- Obstacles geometry is restricted to be 2D extruded on the FFT direction.

This kind of geometry with restrictions will be denoted as periodic-3D geometry as it is in fact an extrusion of 2D geometry with a constant mesh step. An example of extruded geometry is shown on figure 3.1.

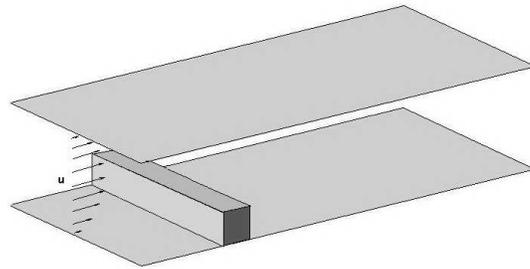


Figure 3.1: Surface mounted square cylinder in a channel - example of extruded geometry

For example a square infinite cylinder in a channel with periodic conditions could be easily solved with the KSFD solver. But it cannot be applied if there is a cube instead of cylinder because then there are boundary layers that have to be resolved by mesh concentration.

The present chapter is devoted to the extension of the KSFD solver to be able to solve fully 3D cases with non-uniform mesh in all three spatial directions and with fully 3D obstacles. The fairly good scalability of the original method should be preserved with a reasonable efficiency. The resulting extended method is combining a two-level multigrid (MG) with the KSFD method.

3.2 Numerical methods for DNS

The non-dimensional incompressible Navier-Stokes equations in a parallelepipedic domain $\Omega = (0, L_x) \times (0, L_y) \times (0, L_z) \subset \mathbb{R}^3$ in primitive variables are considered

$$\nabla \cdot \mathbf{u} = 0 \tag{3.1a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{Re} \nabla^2 \mathbf{u} - \nabla p \tag{3.1b}$$

where Re is the non-dimensional Reynolds number.

Equations (3.1a-3.1b) are discretized on a staggered grid in space by fourth-order symmetry-preserving schemes [12]. For the temporal discretization, a fully explicit dynamic second-order one-leg scheme [12] is used for both convective and diffusive terms. Finally, to solve the pressure-velocity coupling a classical fractional step projection method is used. Further details about the time-integration method can be found in [2, 3].

3.3 Extension of the KSFD algorithm for fully-3D problems

Extension is based on approximation of fully-3D case by means of some periodic-3D case. This periodic-3D case can be solved by KSFD method while solution of the original fully-3D case is provided by an overlying iterative method. It can be a multigrid method (MG) which have periodic-3D case as a second level or for example a conjugate gradient method (CG) preconditioned with the periodic-3D case.

In present work only the one approach using MG is considered as it substantially outperformed the CG option in the preliminary convergence tests. So called MG-KSFD method is a two-level Multigrid (MG) method that uses KSFD as a second-level solver. CG method with local preconditioner is used as a smoother.

In general the idea of the MG-KSFD method is that lower frequencies of the error (most difficult for an iterative solver) are eliminated effectively by the second level solver while higher frequencies are effectively killed by a smoother. The original fully-3D system to be solved is denoted

$$\mathbf{A}^{3D} \mathbf{x}^{3D} = \mathbf{b}^{3D} \tag{3.2}$$

and the periodic-3D system is

$$\tilde{\mathbf{A}} \mathbf{x} = \mathbf{b} \tag{3.3}$$

where \mathbf{A}^{3D} and $\tilde{\mathbf{A}}$ are symmetric positive definite matrices. Multigrid iteration has following algorithm

Algorithm on i -th iteration:

1. Smoother: Obtain approximate solution \mathbf{x}_i^{3D} of (3.2) using CG with local preconditioner. It does not require any data exchange.
2. Calculate residual \mathbf{r}_i^{3D} of system (3.2).
3. Transform residual to second level $\mathbf{r}_i = \mathbf{Q}\mathbf{r}_i^{3D}$
4. Solve error equation $\tilde{\mathbf{A}}\mathbf{z}_i = \mathbf{r}_i$ on second level using KSFD algorithm.
5. Transform error from second level periodic-3D to fully-3D: $\mathbf{z}_i^{3D} = \mathbf{P}\mathbf{z}_i$
6. Correct $\mathbf{x}_{i+1}^{3D} = \mathbf{x}_i^{3D} + \mathbf{z}_i^{3D}$

Here matrixes \mathbf{Q} and \mathbf{P} represent remapping between the levels - restriction and prolongation operators.

Further several aspects will be considered in details.

The idea behind of the MG-KSFD method is the following: KSFD algorithm solving periodic-3D case efficiently eliminates low-frequencies of error and smoother provides fast convergence on higher frequencies. This way, the number of iterations is substantially reduced compared with the CG-KSFD approach.

3.3.1 Solution of the second level

Solution of the second level is provided by the KSFD solver described in the chapter 1. Both levels have equal number of nodes in each direction. Remapping is not required as the both levels have equal mesh size. Also no additional data allocation is needed. This way simplifies implementation, saves memory and computing time. It provides stable convergence of the multigrid method. But to be sure in efficiency of this straightforward approach a comparative convergence test was performed. It is described in the further.

3.3.2 Remapping between the two levels

A first order conservative interpolation was used as an example of remapping to compare with simplified approach without remapping (or in other words when restriction and prolongation operators are identity matrixes). This interpolation preserves the integral of the quantity:

$$\sum_{i=1}^N \Omega_i \mathbf{x}_i = \sum_{i=1}^N \tilde{\Omega}_i \tilde{\mathbf{x}}_i \quad (3.4)$$

where Ω_i and $\tilde{\Omega}_i$ are control volumes of first and second level respectively, \mathbf{x}_i and $\tilde{\mathbf{x}}_i$ are values from first and second level. This conservative inter-level transfer provides stability and convergence of the iterative process. Example of simple first order conservative interpolation is shown on figure 3.2.

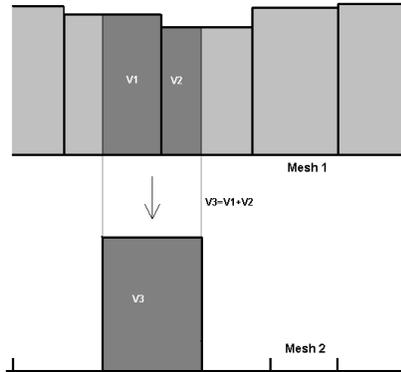


Figure 3.2: 1D example of conservative interpolation. The bars represent values in the control volumes.

The comparative tests showed that this interpolation provides a stable convergence but it gives no advantage in convergence in comparison with the case when restriction and prolongation operators are just identity matrixes. This result confirms efficiency of the simplified approach.

Probably the use of some more complicated interpolation may provide better effect on convergence. But this question is left for the future actions and is beyond the scope of the present work.

3.3.3 Modification of the stencil for solid wall BC

For the periodic boundary conditions on FFT direction the matrix $\tilde{\mathbf{A}}$ of the second level (3.3) is constructed the same way as matrix \mathbf{A}^{3D} of system (3.2) (but on a different grid and ignoring presence of the obstacle). The KSFD method in this case is used without any modifications.

But in case of solid wall boundaries on the FFT direction some modifications are required. Modification of the scheme stencil near the boundaries is required to make matrix compatible with FFT. The matrix $\tilde{\mathbf{A}}$ has a block structure as it shown in chapter 1. Originally the blocks to be diagonalized with FFT (in case of 2-nd order approximation) have form (3.5).

$$\mathbf{A}_{j,k}^p = \begin{pmatrix} a_p + a_{ew} & a_{ew} & & & \\ & a_{ew} & a_p & a_{ew} & \\ & & \ddots & \ddots & \ddots \\ & & & a_{ew} & a_p & a_{ew} \\ & & & & a_{ew} & a_p + a_{ew} \end{pmatrix} \in \mathbb{R}^{N_x \times N_x} \quad (3.5)$$

While to be compatible with FFT blocks must have form for example like (3.6).

$$\tilde{\mathbf{A}}_{j,k}^p = \begin{pmatrix} a_p & a_{ew} & & & \\ a_{ew} & a_p & a_{ew} & & \\ & \ddots & \ddots & \ddots & \\ & & a_{ew} & a_p & a_{ew} \\ & & & a_{ew} & a_p \end{pmatrix} \in \mathbb{R}^{N_x \times N_x} \quad (3.6)$$

The boundary nodes in the system for Pressure equation are uncoupled and can have arbitrary values. They are not taken into account as they are disconnected from the stencil. It is done by modification of the stencil near boundaries - a leg that connects boundary node with its inner neighbor is removed (corresponding coefficient is set to zero). In the original way 3.5 the value of this coefficient is added to the diagonal element (which has the opposite sign). This way contribution of the leg is eliminated. For the second level changing of diagonal element is not allowed by FFT, so diagonal element is just left unchanged providing blocks of FFT compatible form 3.6. Figure 3.3 shows difference in stencil construction near the boundary.

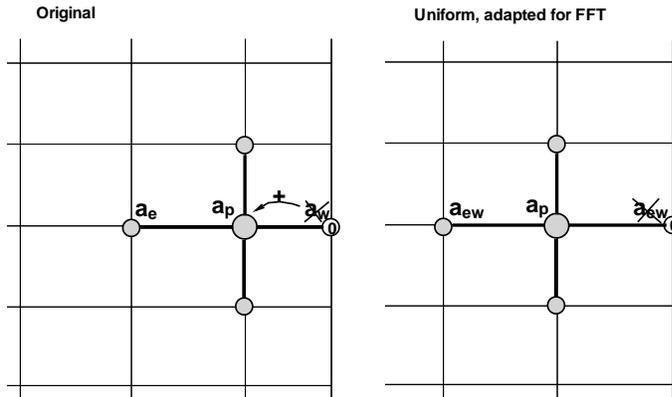


Figure 3.3: Stencil modification near the boundary

Also FFT must use different transformations and eigenvalues than in a periodic case. The Fourier transform is

$$y_j = \frac{1}{n} \left(y_0 + 2 \sum_{i=1}^{n-1} y_i \cos(ji \frac{\pi}{n}) + (-1)^j y_n \right), \quad j = 0, \dots, n \quad (3.7)$$

The inverse Fourier transform is

$$y_j = \frac{1}{2} y_0 + \sum_{i=1}^{n-1} y_i \cos(ji \frac{\pi}{n}) + \frac{1}{2} (-1)^j y_n, \quad j = 0, \dots, n \quad (3.8)$$

Where $n = N_x - 1$ and N_x is the number of nodes on FFT direction. The eigenvalues are following

$$\lambda_j = a_p + \sum_{i=1}^m a_{ew}^i \cos(\frac{\pi}{n} i), \quad j = 0, \dots, n \quad (3.9)$$

Where m is the size of the stencil leg.

Matrixes of the first and second levels \mathbf{A}^{3D} and $\tilde{\mathbf{A}}$ are singular. It should be noted that in case of solid wall BC the image of the original matrix $im(\mathbf{A}^{3D})$ is different from the image of second level matrix $\tilde{\mathbf{A}}$. This leads to incompatibility of the second level system while original system is compatible. This affects negatively convergence and may lead multigrid to fail. To avoid this situation matrix $\tilde{\mathbf{A}}$ must be "patched" to remove singularity. This can be done by changing one diagonal element of the first plane after FFT transformation $\tilde{a}_p^i = 1.1 * a_p^i$. This of course will violate residual of original system in this point but it doesn't spoil the overall convergence. This way provides stable convergence.

3.4 Convergence and performance issues

Performance of the MG-KSFD solver depends on many factors. Es an example a brief list of factors is following.

1. Mesh size - number of iterations growth with number of nodes
2. Mesh geometry. In particular, mesh concentration factors on FFT direction can substantially affect convergence
3. Reynolds number
4. Number of smoother iterations per multigrid iteration

5. Preconditioner of the smoother
6. Accuracy of second level solution

This list can be continued. All of the this factors should be taken into account to achieve optimal performance of the solver. Throughout optimization is beyond of the scope of present work and is left for future work. Only several first priority issues are considered.

Scalability tests were performed to estimate performance on a large DNS. Average number of iterations was obtained for meshes of different size varying from 2×10^5 to 1.2×10^7 nodes (surface mounted cube in a channel, $Re = 5000$ based on cube size). Mesh is described in section 3.5.1. Residual criteria for the iterative solver is chosen to provide on each time step initial divergence norm decrease 10^5 times. Multigrid solver has following configuration.

Smoother precondition: Jacobi (diagonal scaling)

Smoother iterations: 15

Restriction operator: Identity

Precision of second level: tolerance 10^{-2}

Prolongation operator: Identity

Up to 200 CPU were used in the tests. Results are represented in the figure 3.4.

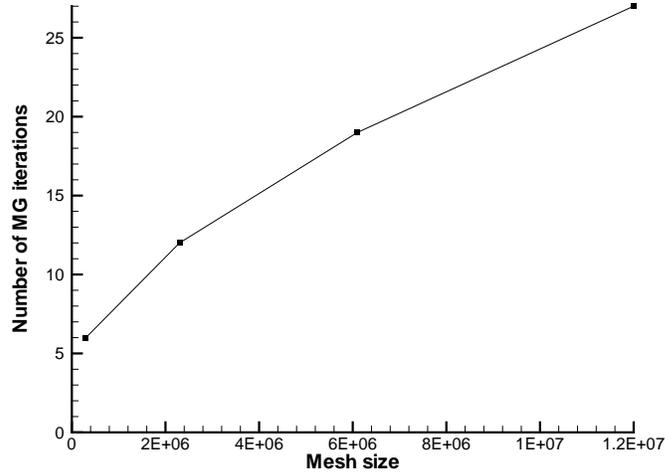


Figure 3.4: Scalability test: number of iterations with mesh growth

The increase of the mesh size 60 times leads to growth of the number of iterations about 4.5 times. As an example of the sensitivity to the smoother iterations number, following demonstration was performed: on 1.2×10^7 nodes mesh the number of smoother iterations was changed from 15 to 25. Number of overall multigrid iterations reduced from 27 to 13 providing substantial increase of performance.

Finally the solver demonstrates moderate scalability and it gives an estimation of performance on larger meshes. According to preliminary estimations the current configuration of the solver can be used to perform DNS with meshes up to 3×10^7 5×10^7 nodes.

3.5 Problem under consideration: surface mounted cube

Surface mounted cube in a channel is a typical DNS problem of rather high interest that cannot be solved with KSFD. It requires a non uniform mesh in all directions to resolve the obstacle region well. The obstacle itself makes a Poisson matrix incompatible with an FFT-based method. Hence it is a good case for MG-KSFD application.

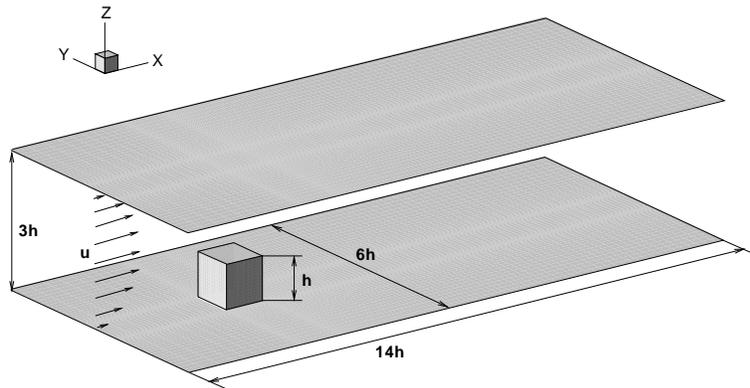


Figure 3.5: Surface mounted cube in a channel

3.5.1 Boundary conditions

The choice of the inflow boundary conditions (BC) may lead to a discussion. It can be a dynamic profile obtained from separate channel flow DNS, it can be a periodic

buffer area that let flow develop to fully turbulent, it can be a stationary profile, etc.

In this case the choice was a compromise between two requirements. From one point of view definition of BC must be clear and easy to reproduce in order to provide a convenient case for the turbulence models development. From another point of view BC must be as close to reality as possible. A comparison described further shows that chosen inflow BC provide adequate results.

- X direction

There are two BC on this direction - inflow and outflow. A stationary analytical profile for turbulent flow is given on the input. The profile specifies velocity components as follows:

$$u = \min(Y_p, k \log(\max(1, Y_p)) + B), \quad v = 0, \quad w = 0 \quad (3.10)$$

where $k = \frac{1}{4}$, $B = 5$, $Y_p = \frac{1}{H} Re_\tau \min(y, H - y)$.

outflow BC are null-derivative with restriction $u \geq 0$. For stability the outflow mass flux is forced (by scaling) to be exactly equal to the inflow flux.

- Y direction

Periodic boundary conditions (PBC)

- Z direction

Solid non-slip walls

3.5.2 Mesh construction

The mesh is not uniform in all directions and it is based on the following mesh concentration function:

$$x_i = \frac{L}{2} \left[1 + \frac{\tanh(\gamma [2(i-1)/N - 1])}{\tanh \gamma} \right] \quad (3.11)$$

where L is the length of the mesh fragment, N is the number of nodes in the fragment and γ is the concentration factor. $\gamma = 0$ corresponds to uniform mesh, bigger γ gives more concentration at the fragment boundaries. Each direction of the mesh is decomposed into several fragments as it is shown on figure 3.6. Each fragment can have its own concentration factor to provide better discretization. The number of nodes in each fragment is selected automatically to satisfy the condition of equal mesh step at the fragments joint. The proportion of nodes for each fragment is governed by the relation of concentration factors. Fragment with smaller γ gets more nodes.

In the case considered the mesh is specified by the 11 parameters which are numbers of nodes N_x , N_y , N_z and following mesh concentration factors:

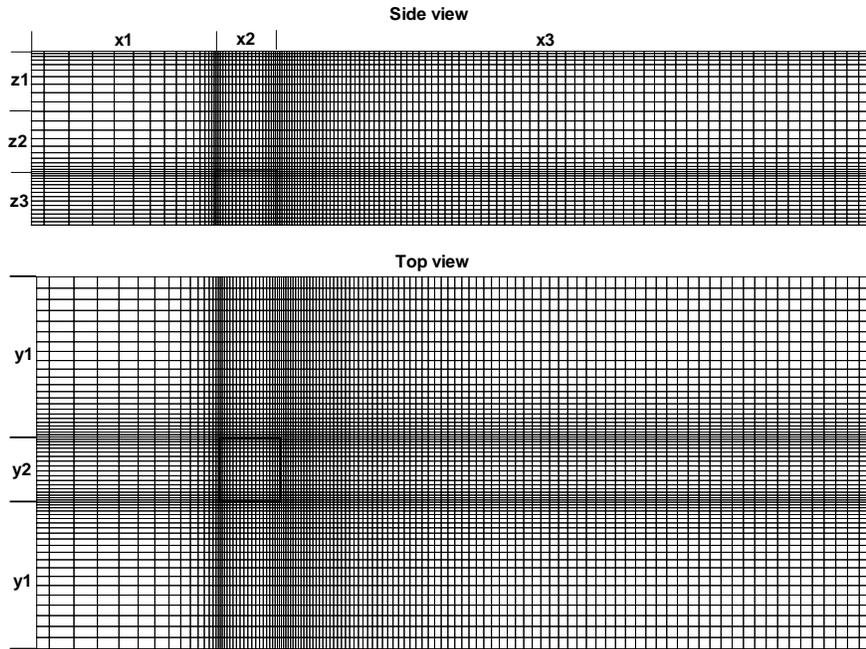


Figure 3.6: Mesh overview

1. X direction
Mesh has 3 fragments on X direction which are specified by concentration factors $\gamma_{x1}, \gamma_{x2}, \gamma_{x3}$.
2. Y direction
On Y direction mesh is specified by 2 concentration factors γ_{y1}, γ_{y2} .
3. On Z direction mesh concentration factors are $\gamma_{z1}, \gamma_{z2}, \gamma_{z3}$.

3.6 Set of stages performed to complete the DNS

1. Verification of BC

Series of preliminary DNS of the surface mounted cube were carried out to chose the inflow boundary conditions. Several inflow conditions were under consideration:

- Constant velocity profile
- Analytical average turbulent profile
- Averaged turbulent profile obtained from channel flow DNS
- Dynamic profile obtained from channel flow DNS
- Periodic buffer zone¹

The analytical average turbulent profile was chosen as it provides clear and simple definition needed for turbulence models development. At the same time it provides adequate physically realistic results. It was verified by a comparison of results with a similar DNS [63] that have dynamic inflow profile obtained from a channel flow DNS. When performing DNS for the comparison it was noticed that flow was very sensitive to the distance from the inflow BC to the cube. To avoid this in the following primary DNS, domain geometry verification was performed to ensure that the input is far enough.

2. Verification of the domain geometry
Preliminary DNS was performed using a small mesh. Distance from the cube to the inflow was chosen minimal where average vertical velocity still is zero (see fig. 3.7). Distance to the outflow was defined the same way.
3. Optimization of the mesh parameters
The mesh parameters were computed to minimize the flow gradients on the computational space for a set of representative instantaneous maps.
4. Mesh refinement at lower $Re=1870$ (based on the cube height)
5 preliminary DNS were performed for mesh refinement at lower Re (corresponds to Re_τ) that requires smaller meshes. After the mesh size for this Re was verified, the mesh size for the primary DNS was obtained by increasing the number of nodes on each direction proportionally to Re_τ .
5. Primary DNS
The primary DNS was performed on MareNostrum supercomputer of BSC using 300 CPU. The case parameters are following:
 $Re = 7220$ ($Re_\tau=590$)
 Cube height = 1,
 Channel height = 3,
 Channel width = 6,
 Channel length = 14,

¹Velocity profile from the flow-orthogonal plane somewhere between input BC and the cube, far enough from both, is picked up and set to the inflow BC each time step. Mass flux is forced to be preserved.

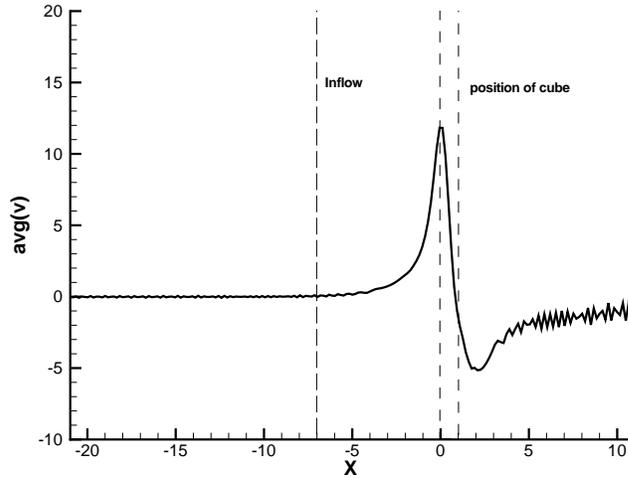


Figure 3.7: Choice of the cube position. Average vertical velocity profile in the middle of the channel. Inflow is placed where vertical velocity fades to zero

Distance from inflow BC to cube = 7,
Mesh size 1.6×10^7
See the figure 3.5 for the domain overview.

The results of the DNS are to be postprocessed and published. The primary DNS consumed about 100000 CPU hours on the MareNostrum supercomputer. All the preliminary DNS and solver tests were performed on the MVS10000 supercomputer of the Russian academy of science. Illustrative snapshots of instantaneous fields are represented on figure 3.8.

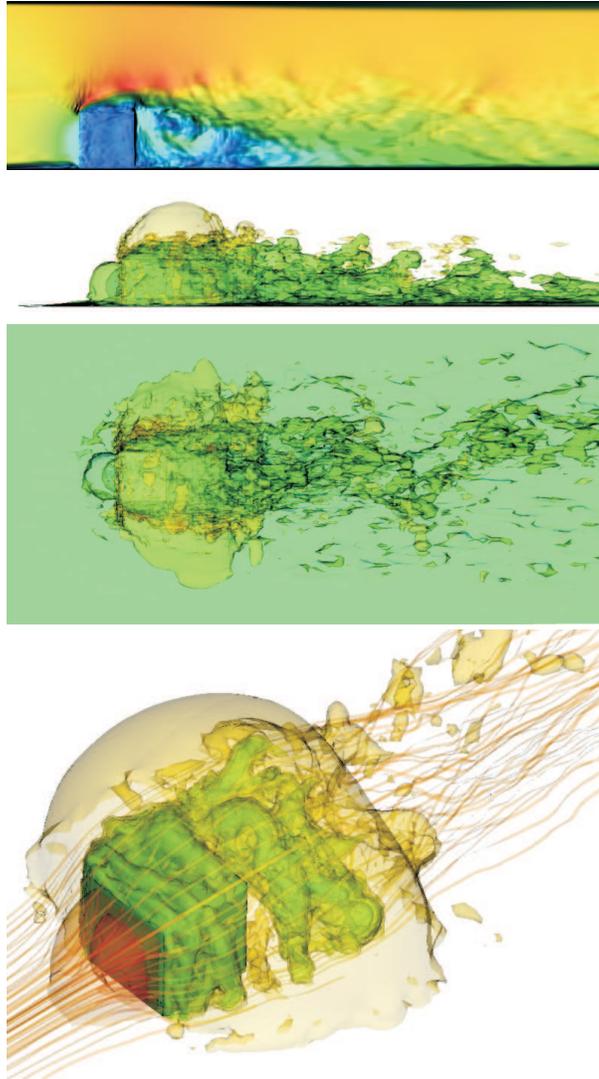


Figure 3.8: Surface mounted cube in a channel. Instantaneous maps of kinetic energy averaged on periodic direction (top) and iso-surfaces (middle), pressure iso-surfaces with streamlines (bottom)

Chapter 4

Conclusions and future research

4.1 Concluding remarks

A brief list of main results of the thesis work

- **KSFD solver**
It was developed on a base of Schur-Fourier Decomposition method [2] and Krylov conjugate gradient method. The solver can be efficiently used on both small cluster and supercomputers using up to at least 1024 CPU. The solver is applicable to DNS or LES with mesh size more than 10^8 nodes using 4-th order scheme.
- **DNS of a differentially heated cavity (DHC) with $Ra = 10^{11}$**
The challenging-size DNS using 512CPU and mesh of 1.1×10^8 using 4th order scheme was performed on MareNostrum supercomputer. The DNS demonstrated scalability and performance of the KSFD solver.
- **Contribution to the LES validation basis**
The set of three DNS of differentially heated cavity was performed for Rayleigh numbers 10^{10} , 3×10^{10} , 10^{11} . Together with previous results [3] [4] it gives a relatively wide range of Ra -numbers from weak to fully developed turbulence.
- **MG-KSFD solver**
An extension of the KSFD solver with multigrid overlay was developed to solve fully 3D problems with 3 wall-bounded directions using non-uniform mesh in all directions and 3D obstacles in the flow.
- **DNS of a surface mounted cube**
The DNS (mesh size 1.6×10^7 nodes) was performed using 300CPU on MareNostrum supercomputer.

4.2 Future research

Future research can be considered in several aspects.

4.2.1 Implementation aspects

Further development of the DNS/LES code includes several directions

- **Algebraic kernel replacement**
In-house algebraic operation implementations and local solvers can be replaced with one of the widely used linear algebra libraries (BLAS, etc). This may substantially improve overall performance of the code.
- **Memory access optimization**
Explicit part of the code that includes operations with complex high-order numerical scheme coefficients takes significant part of the overall CPU time. Additional optimization of data structures and memory access may improve performance. Also it may result on reduction of memory consumption.
- **Hybrid MPI+OpenMP parallelization**
The use of the hybrid parallelization may improve performance on parallel systems with multi-core nodes. It also provides much more memory for each MPI process.

4.2.2 Numerical methods aspects

This includes further development of MG+KSFD solver for fully 3D geometries. There is a lot of work to be done, in particular there are several issues:

- **Preconditioner for the smoother**
Current implementation has conjugate gradient method with a straightforward Jacobi preconditioner as a smoother. The search for more efficient smoother is of high priority.
- **Prolongation/resrtiction operators**
Currently prolongation and restriction operators are reduced to identity matrix. Implementation of high-order conservative interpolations and search for better ways to perform interlevel transfer are to be done.
- **FFT based methods**
Development of the methods based on FFT transformation on two directions and applicable for boundary conditions different from periodic.

Appendix A: Parallel CFD terminology

Don't take it seriously!!!

ShitFD (SFD) - under-resolved, obsolete, poorly defined CFD simulations and algorithms

Differentially Shited Cavity (DSC) - DNS of natural convection in a differentially heated cavity

Shit Mounted Cube (SMC) - DNS of surface mounted cube in a channel

Open Shame Cavity (OSC) - DNS of natural convection in an open cavity

Open Shit - open source software

Numerical Shit Transfer - Journal of Numerical Heat Transfer

Cucumbers and fluids - Journal Computers and Fluids

Fuckingly huge - common property of Poisson matrix in DNS

Especially fucking - common property of Poisson equation in DNS

Friki Cluster - small cluster without a queue system

Friki DNS - DNS performed on a Friki cluster

Fire fuckin wall - firewall configured with a paranoid inclination

Incredible shit - poorly optimized CFD code

NightmareNostrum - MareNostrum supercomputer in its early period

Network fuckup - decrease of performance due to slowdown of network operations

Shit paper - any journal paper currently being prepared for publication

Brutal Monster (brumon) - high level CFD specialist

Brutal solver - surprisingly fast solver implementation

Brutal DNS - DNS with huge mesh size and computing power demands

Cucumber - any problem to be solved in CFD context

Make love not Krylov - common slogan in a context of solver development

Bibliography

References

- [1] M. Soria, C. D. Pérez-Segarra, and A. Oliva. A Direct Parallel Algorithm for the Efficient Solution of the Pressure-Correction Equation of Incompressible Flow Problems Using Loosely Coupled Computers. *Numerical Heat Transfer, Part B*, 41:117–138, 2002.
- [2] F. X. Trias, M. Soria, C. D. Pérez-Segarra, and A. Oliva. A Direct Schur-Fourier Decomposition for the Efficient Solution of High-Order Poisson Equations on Loosely Coupled Parallel Computers. *Numerical Linear Algebra with Applications*, page (published online), 2005.
- [3] F. X. Trias, M. Soria, A. Oliva, and C. D. Pérez-Segarra. Direct numerical simulations of two- and three-dimensional turbulent natural convection flows in a differentially heated cavity of aspect ratio 4. *Journal of Fluid Mechanics*, submitted, 2006.
- [4] M. Soria, F. X. Trias, C. D. Pérez-Segarra, and A. Oliva. Direct numerical simulation of a three-dimensional natural-convection flow in a differentially heated cavity of aspect ratio 4. *Numerical Heat Transfer, part A*, 45:649–673, April 2004.
- [5] M. Soria, C. D. Pérez-Segarra, and A. Oliva. A Direct Schur-Fourier Decomposition for the Solution of the Three-Dimensional Poisson Equation of Incompressible Flow Problems Using Loosely Parallel Computers. *Numerical Heat Transfer, Part B*, 43:467–488, 2003.
- [6] S. Kocak and H. U. Akay. Parallel Schur complement method for large-scale systems on distributed memory computers. *Applied Mathematical Modelling*, 25:873–886, 2001.
- [7] Natalja Rakowsky. The Schur Complement Method as a Fast Parallel Solver for Elliptic Partial Differential Equations in Oceanography. *Numerical Linear Algebra with Applications*, 6:497–510, 1999.
- [8] D. G. E. Grigoriadis, J. G. Bartzis, and A. Goulas. Efficient treatment of complex geometries for large eddy simulations of turbulent flows. *Computers and Fluids*, 33:201–222, 2004.

- [9] M. Soria. Parallel Multigrid Algorithms for Computational Fluid Dynamics and Heat Transfer, 2000.
- [10] P.N.Swarztrauber. The Methods of Cyclic Reduction, Fourier Analysis and the FACR Algorithm for the Discrete Solution of Poisson's Equation on a Rectangle. *SIAM Review*, 19:490–501, 1977.
- [11] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Inc., 2001.
- [12] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-Preserving Discretization of Turbulent Flow. *Journal of Computational Physics*, 187:343–368, May 2003.
- [13] Shihe Xin and Patrick Le Quéré. Direct numerical simulations of two-dimensional chaotic natural convection in a differentially heated cavity of aspect ratio 4. *Journal of Fluid Mechanics*, 304:87–118, 1995.
- [14] D. H. Kim, K. S. Yang, and M. Senda. Large eddy simulation of turbulent flow past a square cylinder confined in a channel. *Computers and Fluids*, 33:81–96, 2004.
- [15] R. W. Hockney. A Fast Direct Solution of Poisson's Equation Using Fourier Analysis. *Journal of the Association for Computing Machinery*, 12:95–113, 1965.
- [16] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. PWS, 1996.
- [17] Alexandre Joel Chorin. Numerical Solution of the Navier-Stokes Equations. *Journal of Computational Physics*, 22:745–762, 1968.
- [18] N. N. Yanenko. *The Method of Fractional Steps*. Springer-Verlag, 1971.
- [19] A. J. Chorin. *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1993.
- [20] William D. Henshaw. A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids. *Journal of Computational Physics*, 113(1):13–25, 1994.
- [21] N. Anders Petersson. Stability of Pressure Boundary Conditions for Stokes and Navier-Stokes Equations. *Journal of Computational Physics*, 172:40–70, 2001.
- [22] J. Kim and P. Moin. Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 123:308–323, 1985.
- [23] P.J.Davis. *Circulant Matrices*. Chelsea Publishing, New York, 1994.

- [24] W.H.Press, S.A.Teukolsky, W.T.Vetterling, and B.P.Flannery. *Numerical Recipes in C. The art of Scientific Computing*. Cambridge University Press, Cambridge, 1992.
- [25] Michele Benzi. Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics*, 182:418–477, 2002.
- [26] G. de Vahl Davis and I. P. Jones. Natural Convection in a Square Cavity: a Comparison Exercise. *International Journal for Numerical Methods in FLuids*, 3:227–248, 1983.
- [27] M. Hortmann, M. Peric, and G. Scheuerer. Finite Volume Multigrid Prediction of Laminar Natural Convection: Bench-Mark Solutions. *International Journal for Numerical Methods in Fluids*, 11:189–207, 1990.
- [28] P. Le Quéré. Accurate Solutions to the Square Thermally Driven Cavity at High Rayleigh Number. *Computers and Fluids*, 20:29–41, 1991.
- [29] M.R. Ravi, R.A.W.M. Henkes, and C.J. Hoogendoorn. On the high-Rayleigh-number structure of steady laminar natural-convection flow in a square enclosure. *Journal of Fluid Mechanics*, 277:325–351, 1994.
- [30] D.C. Wan, B.S.V. Patnaik, and G.W. Wei. A New Benchmark Quality Solution for the Buoyancy-Driven Cavity by Discrete Singular Convolution. *Numerical Heat Transfer, part B*, 40:199–228, 2001.
- [31] T. Fusegi, J.M. Hyun, K. Kuwahara, and B. Farouk. A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure. *International Journal of Heat and Mass Transfer*, 34:1543–1557, 1991.
- [32] E. Tric, G. Labrosse, and M. Betrouni. A first incursion into the 3D structure of natural convection of air in a differentially heated cubic cavity. *International Journal of Heat and Mass Transfer*, 43:4043–4056, 2000.
- [33] Bérengère Lartigue, Sylvie Lorente, and Bernard Bourret. Multicellular natural convection in a high aspect ratio cavity: experimental and numerical results. *International Journal of Heat and Mass Transfer*, 43:3157–3170, 2000.
- [34] P. Le Quéré. A Note on Multiple and Unsteady Solutions in Two-Dimensional Convection in a Tall Cavity. *Journal of Heat Transfer*, 112:965–974, 1990.
- [35] H. Schweiger, A. Oliva, M. Costa, and C.D.Pérez-Segarra. Numerical experiments on laminar natural convection in rectangular cavities with and without honeycomb structures. *International Journal of Numerical Methods for Heat and Fluid Flow*, 5:243–443, 1995.

- [36] K. H. Winters. Hopf Bifurcation in the Double-Glazing Problem with Conducting Boundaries. *Journal of Heat Transfer*, 109:894–898, 1987.
- [37] R.A.W.M. Henkes. *Natural-Convection Boundary Layers*. PhD thesis, Faculty of Applied Physics at the Delft University of Technology, 1995.
- [38] Patrick Le Quéré and Masud Behnia. From onset of unsteadiness to chaos in a differentially heated square cavity. *Journal of Fluid Mechanics*, 359:81–107, 1998.
- [39] P. Le Quéré. Transition to unsteadiness routes to chaos and simulation of turbulent flows in cavities heated from the side: a review of present status. In *Heat Transfer: Proceedings of the 10th International Heat Transfer Conference*, volume 1, pages 281–296. Hewitt GF, 1994.
- [40] M. Farhangnia, S. Biringen, and L.J. Peltiery. Numerical Simulation of Two-Dimensional Bouyancy-Driven Turbulence in a Tall Rectangular Cavity. *International Journal for Numerical Methods in Fluids*, 23:1311–1326, 1996.
- [41] S. Xin and P. Le Quéré. Direct numerical simulations of two-dimensional chaotic natural convection in a differentially heated cavity of aspect ratio 4. *Journal of Fluid Mechanics*, 304:87–118, 1995.
- [42] M. A. Christon, P. M. Gresho, and S. B. Sutton. Computational Predictability of Time-Dependent Natural Convection Flows in Enclosures (Including a Benchmark Solution). *International Journal of Numerical Methods in Fluids*, 201:953–980, 2002.
- [43] S. Paolucci and D. R. Chenoweth. Transition to chaos in a differentially heated vertical cavity. *Journal of Fluid Mechanics*, 201:379–410, 1989.
- [44] S. Paolucci. Direct Numerical Simulation of Two-Dimensional Turbulent Natural Convection in an Enclosed Cavity. *Journal of Fluid Mechanics*, 215:229–262, 1990.
- [45] R. J. A. Jansen, R. A. W. M. Henkes, and C. J. Hoogendoorn. Transition to Time-Periodicity of a Natural-Convection Flow in a 3-D Differentially Heated Cavity. *International Journal of Heat and Mass Transfer*, 36(11):2927–2940, 1993.
- [46] T. Fusegi, J. M. Hyun, and K. Kuwahara. Three-Dimensional Numerical Simulation of Periodic Natural Convection in Differentially Heated Cubical Enclosure. *Appl. Sci. Res.*, 49:271–282, 1992.

- [47] R. J. A. Jansen and R. A. W. M. Henkes. Instabilities in Three-Dimensional Differentially-Heated Cavities with Adiabatic Horizontal Walls. *Physics of Fluids*, 8(1):62–74, 1996.
- [48] G. Labrosse, E. Tric, H. Khallouf, and M. Betrouni. A Direct (Pseudo-Spectral) Solver of the 2-D/3-D Stokes Problem: Transition to Unsteadiness of Natural-Convection Flow in a Differentially Heated Cubical Cavity. *Numerical Heat Transfer B*, 31:261–276, 1997.
- [49] F. Fusegi, B. Farouk, J. M. Hyun, and K. Kuwahara. A direct numerical simulation of periodic natural convection in differentially heated cubical enclosure. In *Proceedings of the International Symposium on Engineering Turbulence Modeling and Measurements*, pages 261–268, 1990.
- [50] R. A. W. M. Henkes and P. Le Quéré. Three-dimensional transition of natural-convection flows. *Journal of Fluid Mechanics*, 319:281–303, 1996.
- [51] F. Penot, A. N’Dame, and P. Le Quéré. Investigation of the Route to Turbulence in a Differentially Heated Cavity. In *9th International Heat Transfer Conference*, pages 417–422, Jerusalem, 1990.
- [52] S. Xin and P. Le Quéré. An Extended Chebyshev Pseudo-Spectral Benchmark for the 8:1 Differentially Heated Cavity. *International Journal of Numerical Methods in Fluids*, 8:2182–2189, 2002.
- [53] A. Gorobets, F. X. Trias, M. Soria, and A. Oliva. A scalable Krylov-Schur-Fourier Decomposition for the efficient solution of high-order Poisson equation on parallel systems from small clusters to supercomputers. *Computers and Fluids*, (submitted).
- [54] Patrick J. Roache. Code Verification by the Method of Manufactured Solutions. *Journal of Fluids Engineering, Transactions of ASME*, 124:4–10, 2002.
- [55] J. Salat, S. Xin, P. Joubert, A. Sergent, F. Penot, and P. Le Quéré. Experimental and numerical investigation of turbulent natural convection in large air-filled cavity. *International Journal of Heat and Fluid Flow*, 25:824–832, 2004.
- [56] James Lighthill. *Waves in Fluids*. Cambridge University Press, Cambridge, 2002.
- [57] Roel Verstappen. On smooth, symmetry-preserving approximations of turbulent convection in a plane channels. *Physics of Fluids*, 2006.
- [58] F. X. Trias, M. Soria, A. Oliva, and R. Verstappen. Regularization models for the simulation of turbulence in a differentially heated cavity. September 2006.

- [59] F. X. Trias, R. W. C. P. Verstappen, M. Soria, A. Gorobets, and A. Oliva. Regularization modelling of a turbulent differentially heated cavity at $Ra = 10^{11}$. In *5th European Thermal-Sciences Conference, EURO THERM 2008*, Eindhoven, The Netherlands, May 2008.
- [60] Yukio Kaneda and Mitsuo Yokokawa. DNS of Canonical Turbulence with up to 4096^3 Grid Points. In *Parallel Computational Fluid Dynamics*, pages 23–32. Elsevier, May 2004.
- [61] Sergio Hoyas and Javier Jimenez. Channel 2000, Computing wall turbulence at experimental Reynolds numbers. *School of Aeornautics, Universidad Politecnica de Madrid, Spain*, 2006.
- [62] Alexander Yakhot, Tomer Anor, Heping Liu and Nikolay Nikitin. Direct numerical simulation of turbulent flow around a wall-mounted cube: spatio-temporal evolution of large-scale vortices. *J. Fluid Mech.*, 556:1–9, 2006.
- [63] Heping Liu Nikolay Nikitin Alexander Yakhot, Tomer Anor. Direct numerical simulation of turbulent flow around a wall-mounted cube: spatio-temporal evolution of large-scale vortices. *Journal of Fluid Mechanics*, 566:1–9, 2006.