Comparison of a spectral method with volume penalization and a finite volume method with body fitted grids for turbulent flows

Benjamin Kadoch, Thorsten Reimann, Kai Schneider, Michael Schäfer

Comparison of a spectral method with volume penalization and a finite volume method with body fitted grids for turbulent flows

B. Kadoch[a], T. Reimann[b], K. Schneider[c], M. Schäfer[b]

[a]*Aix Marseille Université, CNRS, IUSTI UMR 7343, 13453, Marseille, France*
[b]*Fachgebiet Numerische Berechnungsverfahren im Maschinenbau, Technische Universität Darmstadt, Darmstadt, Germany*
[c]*Aix Marseille Universit, CNRS, Centrale Marseille, M2P2 UMR 7340, 13451, Marseille, France*

Abstract

We consider a turbulent flow past periodic hills at Reynolds number 1400 and compare two numerical methods: A Fourier pseudo–spectral scheme with volume penalization to model the no-slip boundary conditions and a finite volume method with body fitted grids. A detailed comparison of the results is presented for mean velocity profiles and Reynolds stress and confronted with those obtained by Breuer et al. [1]. In addition higher order statistics are performed and their scale-dependence is analyzed using orthogonal wavelets. Moreover, for the Fourier pseudo-spectral scheme, the influence of the Reynolds number is investigated.

Keywords: computational fluid dynamics, periodic hill, volume penalization, Fourier spectral method, finite volumes, wavelets

1. Introduction

Computational fluid dynamics in complex geometries is still a challenge especially for high Reynolds numbers. The thin boundary layers which destabilize, subsequently detach and thus generate coherent vorticies are a key feature to understand turbulence generated by walls. Two main approaches

Email address: benjamin.kadoch@univ-amu.fr (B. Kadoch)

have been developed so far, either body fitted grids using, *e.g.*, finite volume or finite element discretizations, or immersed boundary methods (IBM) which are becoming more and more attractive due to their high flexibility. Indeed, for the latter the complex geometry is typically embedded into a larger computational domain of simple shape, *e.g.*, a rectangle. Hence classical discretizations, using Cartesian grids, for which efficient solvers are available, can be employed. The influence of the geometry and the no-slip boundary conditions is modeled by modifying the underlying Navier–Stokes equations. The origin of IBM can be traced back to Courant (1943) [2] in the context of constraint optimization. A large range of immersed boundary methods can be found since then, for example Lagrangian multipliers [4], level-set methods [5], fictitious domain approaches and surface [6] and volume penalization approaches [7]. For reviews we refer the reader to [8] and [9].

Validation and benchmarking are an essential step in code development. Comparing the results of different numerical methods enables us to know the advantages and drawbacks. Their domain of validity can thus be checked and their precision can be assessed.

In this study, we have chosen the volume penalization method [7] coupled with a Fourier pseudo-spectral method [10]. Investigations on the Gibbs ocsillations which appear in Fourier based schemes can be found in [11], as well as filtering techniques for removing this Gibbs phenomenon. The code has been benchmarked in two space dimensions [12] and applied to moving obstacles [13]. To evaluate the performance of this new method in three-dimensional, turbulent fluid flow, it is compared here to that of a well-established second-order finite-volume method in terms of accuracy and efficiency. The latter, called "Fastest" [14] is based on a finite volume discretization and uses body fitted grids. The flow configuration is a well documented benchmark "flow over periodic hills" [1]. This benchmark is of interest, since it is a geometrically simple test case and there is no difficulty to specify inflow/outflow boundary conditions. Moreover, the physical mechanisms of separation on curved surfaces and the streamwise decorrelation enhanced, by choosing a sufficiently large distance between the two hills, are an attractive features of this test case. An other class of IBM was investigated recently using this benchmark in [15]. The three-dimensional Fourier pseudo-spectral code including volume penalization (denoted by "Pen4Flow") to impose non-periodic boundary conditions is thus applied for the first time to compute

fully developed turbulent flows. The aim of the paper is to show a detailed benchmark of the volume penalization method for three-dimensional turbulent flows and to compare the efficiency with respect to a classical finite volume code Fastest.

For the two solvers several grid levels are employed. As the considered flow is turbulent, statistical analysis are performed and compared with each other. In addition we also use orthogonal wavelets to decompose the flow into different scales of motion. Statistical quantities can thus be defined as a function of scale and direction.

The manuscript is organized as follows. In section 2 the "Flow over periodic hills" configuration is explained and the two numerical methods are shortly presented. Then in section 3, the results for the different codes are described and the flow statistics are compared. Scale dependent analysis of the results are also performed. Finally, the influence of the Reynolds number is investigated for the pseudo-spectral code with the volume penalization method. Some conclusions and perspectives are exposed at the end of the manuscript.

2. Flow configuration and Numerical methods

2.1. Flow configuration

The classical benchmark "Flow over periodic hills" [1], is considered in detail at a Reynolds number of $Re = 1400$ which is based on the bulk velocity u_b at $x = 0$ and the bump height H. To check the influence of the Reynolds number, we also considered $Re = 700$ and $Re = 2800$. The boundary conditions are periodic along the streamwise x-direction and the spanwise z-direction. The flow is confined by a hill at the bottom and by a wall plane at the top (y-direction). The flow configuration is illustrated in Fig. 1.

2.2. Pen4Flow: Fourier pseudo-spectral code with "volume penalization" method

In this subsection, we present the volume penalization method and the methodology to include the periodic hill configuration. More details on the numerical and mathematical validation of this method can be found [13, 16].

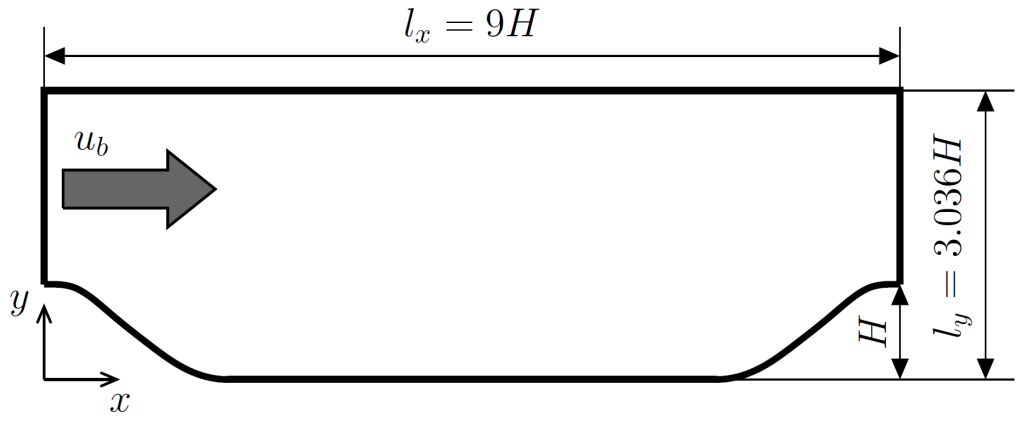

Figure 1: Sketch of the 2d hill geometry. The dimension in spanwise z-direction is $l_z = 4.5H$.

The Fourier pseudo-spectral code solves the incompressible Navier-Stokes equations in velocity-vorticity formulation. The volume penalization method modifies the Navier-Stokes equations by adding a penalization term to the right hand side of the momentum equation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\omega} \times \nabla \boldsymbol{u} + \nabla \Pi - \nu \nabla^2 \boldsymbol{u} - \boldsymbol{F} = -\frac{1}{\eta}\chi(\boldsymbol{u}), \qquad \nabla \cdot \boldsymbol{u} = 0, \qquad \mathbf{x} \in \boldsymbol{\Omega} \quad (1)$$

with

$$-\nabla^2 \Pi = \nabla \cdot \left(\boldsymbol{\omega} \times \boldsymbol{u} + \boldsymbol{F} + \frac{1}{\eta}\chi(\boldsymbol{u}) \right) \quad (2)$$

where $\boldsymbol{u} = (u_x, u_y, u_z)$ is the velocity, ν the kinematic viscosity, Π is the modified pressure, $\boldsymbol{\omega}$ the vorticity, $\boldsymbol{F}$ an external force, η the permeability (or penalization parameter) and χ the mask function which equals to 1 in Ω_s (solid domain) and 0 in Ω_f (fluid domain). Figure 2 (a) shows the computational domain $\Omega = \Omega_f \cup \Omega_s$ where Ω_f is the fluid domain (in white) and Ω_s is the solid domain (in black). The solution of the penalized Navier–Stokes equations does converge towards the solution of the Navier–Stokes equations with no-slip boundary conditions [7]. The difference between the exact solution of the penalized equation and the exact solution of the Navier–Stokes equations, called modeling error is proportional to $\sqrt{\eta}$ and for sufficiently small values of η the solution is precise enough.

In space all variables are represented as truncated Fourier series, product and nonlinear terms are evaluated in physical space while derivatives are evaluated in spectral space. For time integration a second order Adams-Bashforth scheme is used for the nonlinear and the penalization term. The viscous term is integrated exactly using the semi-group formulation [13], which means that the semi-group of the heat kernel is used for time integration. This corresponds to the exact integration of the linear viscous term. The numerical study was carried out on three different grids that differed in the number of nodes employed, see Table 1. As the volume penalization method takes care of generating the boundary layer, the numerical grid has to be chosen sufficiently fine to be able to resolve these thin layers. For each grid, the size l_y in the y-direction of the computational domain is chosen such that $y = 3.036$ coincides with a collocation point and that the solid domain on the top contains 10 points along the y-direction. In all simulations, the penalization parameter is $\eta = 10^{-3}$ and the CFL constant is equal to 0.1. The choice of the penalization parameter is a compromise which works well for the resolutions considered here, more details on the influence of the penalization parameter can be found in [11]. For Pen4Flow, we observe that the time step limitation comes from the CFL condition and not for the stability limit ($\Delta t < \eta$) imposed by the penalization term.

The flow is driven by an external force $\boldsymbol{F} = f\boldsymbol{e_x} \in \Omega$. In order to keep the desired mass flow rate constant throughout the whole simulation, a control function similar to [17], updates the forcing term on every time step:

$$f_{n+1} = f_n + (u_b - 2\tilde{u}_n + \tilde{u}_{n-1})/(\alpha_f \Delta t) \tag{3}$$

where $\tilde{u}_n$ is the measured bulk velocity at time level n, α_f a damping factor ($\alpha_f = 10$ for Pen4Flow and $\alpha_f = 1$ for Fastest). In all computations the time-step for Pen4Flow is $\Delta t = 10^{-3}$ and for Fastest $\Delta t = 1.5\ 10^{-3}$.

The flow is initialized with $\boldsymbol{u}(t = 0) = 0$. To trigger the instability, an impulsion is imposed during the first time interval $[0, 2]$, *i.e.*, the forcing term is extended to $\boldsymbol{F} = f\boldsymbol{e_x} + f_{\text{imp}}\boldsymbol{e_z}$ with $f_{\text{imp}} = A\sin(2x2\pi/l_x) * \exp(1 - 2(y - l_y/2)^2) * \sin(2z2\pi/l_z)$ and $A = 1$ being a suitable amplitude.

2.3. Fastest: Finite volume method

To solve the transient, incompressible Navier-Stokes equations with the finite volume method on body-fitted grids, we employ the solver Fastest

[14]. This code is based on a cell-centered finite volume method on block-structured grids. Discretization of the convective term is carried out using a multi-dimensional Taylor series expansion which preserves second-order accuracy on strongly distorted grids [18], while the time is advanced by the Crank-Nicolson scheme. This numerical procedure is of second order in space and in time. The SIMPLE procedure is employed for the solution of the pressure. Velocity and pressure fields are coupled via the Rhie-Chow interpolation. The discretized system of equations is implicitly solved by Stone's method.

Two grids were generated for the finite volume computations, the first of which, subsequently referred to as fine grid, was constructed to fulfill the requirements of a DNS, while a coarse grid was derived from that by omitting every second node in each direction. A precursor RANS simulation employing the ζ-f model of Hanjalić [19] gave an estimate of the Kolmogorov length scale η. Following Pope's recommendations [20], a maximum ratio of $\Delta x/\eta = 2.1$ was ensured over the entire domain for the construction of the fine grid. For the boundary layer resolution, the normalized wall distance of the first cell center y_1^+ was kept below 0.1 everywhere. A coarse grid was derived from the fine one by omitting every second node in every index direction.

The time step size was chosen such that CFL < 1 for all time steps. During the computation, the mass flux was kept constant utilizing (3) as external forcing. However, $u(t = 0)/u_b = 1.0$ was chosen as initial condition, and during the initial simulation period no additional forcing was necessary to excite the flow instabilities.

3. Results

First we consider the periodic hill configuration at Reynolds number $Re = 1400$ and compare Fastest and Pen4Flow with the results obtained by Breuer et al. [1]. Second, we focus on Pen4Flow and study the influence of the Reynolds number.

3.1. Case $Re = 1400$

In the following we present the results at $Re = 1400$ and study the influence of the resolution.

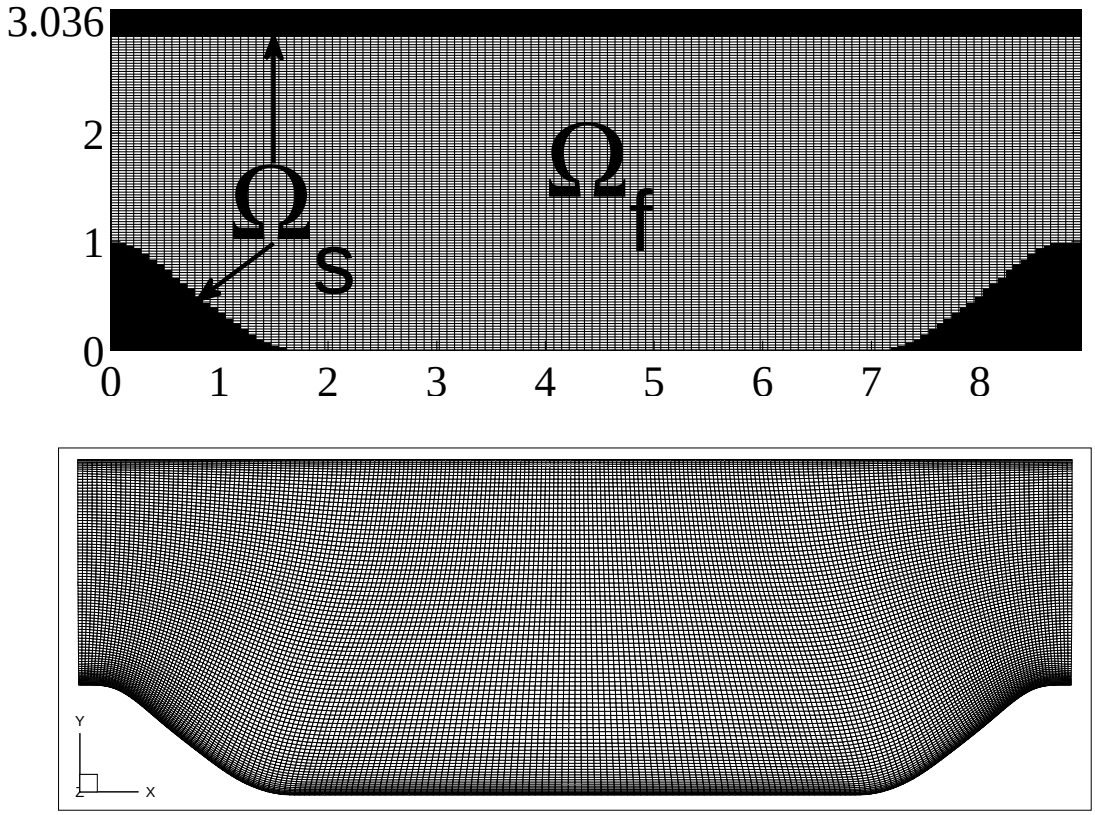

Figure 2: Numerical grids: (top) Pen4Flow (128^3 for visualization), (bottom) Fastest for the coarsest grid ($442 \times 184 \times 200$).

Table 1 summarizes the numerical and physical parameters for both codes. Scaling of the CPU time with respect to the number of processors is given in Fig. 3 for both codes using coarse resolution, obtained on the HHLR supercomputer ("Hessischer Hochleistungsrechner"). Note that for Pen4Flow, the fluid domain height is constant and equal to $y_f = 3.036$ and the solid

domain height is chosen such that the solid contains 10 grid points and the boundary of the fluid coincides with a grid point.

	N_x	N_y	N_z	y_l/H	FTT
Pen4Flow 256^3	256	256	256	3.15941	48.88
Pen4Flow 512^2256	512	512	256	3.09647	14.69
Pen4Flow 512^3	512	512	512	3.09647	29.19
Fastest coarse	221	92	100	3.05	215
Fastest fine	442	184	200	3.05	213

Table 1: Numerical parameters, flow domain height y_l/H, flow-through times (FTT = L/u_b) used for statistical evaluation for $Re = 1400$.

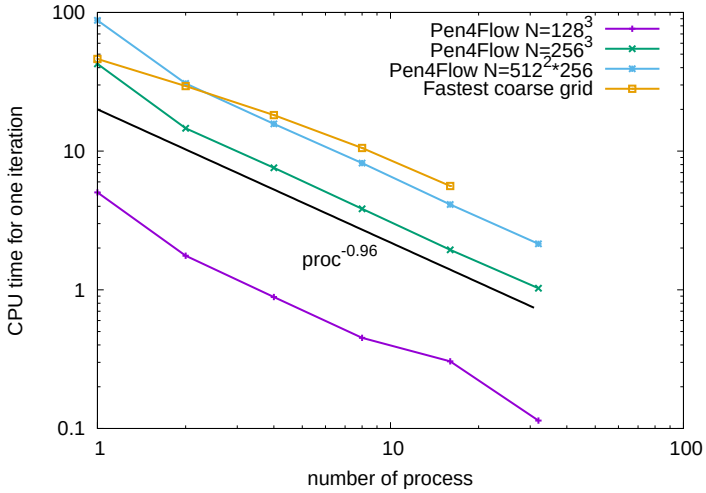

Figure 3: CPU time(s) for one iteration as function of the number of processors and for different resolutions using Pen4Flow and Fastest codes.

Figure 4 shows a three-dimensional representation of isosurfaces of the vorticity norm colored with the streamwise velocity. The computation has been obtained with the volume penalization method at resolution 512^3. We observe that the flow is turbulent, characterized by the presence of vortices of different size. We can also identify a recirculation zone corresponding to an upstream (negative) velocity. In the following we will consider statistical quantities only. Table 2 shows that both codes yield similar values for

the separation and reattachment point. A slight difference can however be observed for the reattachment point which might be due to the penalization method.

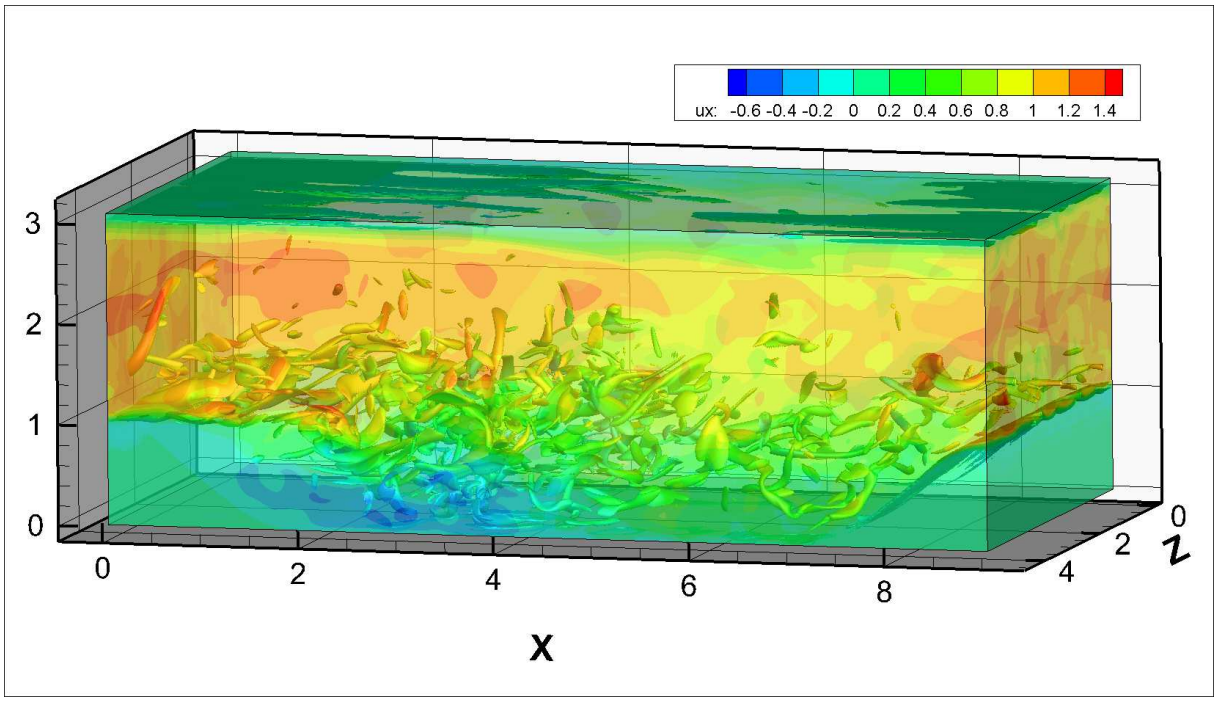

Figure 4: Isosurface of the vorticity norm $||\boldsymbol{\omega}|| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$ colored with the horizontal velocity u_x computed with the Pen4Flow code at resolution 512^3 for $Re = 1400$.

	N_x	N_y	N_z	x_{sep}/H	x_{reatt}/H
Pen4Flow 512^3	512	512	512	0.26	5.26
Fastest coarse	221	92	100	0.26	5.20
Fastest fine	442	184	200	0.26	5.21
BPRM09 LESOCC	N_{tot}	=	13.6 10^6	0.26	5.19

Table 2: Numerical parameters, separation point x_{sep}/H and reattachment point x_{reatt}/H.

3.1.1. Moment statistics

Table 3 summarizes the first and second order moments together with the skewness and flatness values of the velocity fields for both codes at a given time instant in the statistically steady state. We can note that averaging the results over time does not change the results and slight differences ($\leq 2\%$) appear only for high order moment statistics. The flatness reflects the intermittency in a turbulent flow while the skewness factor of velocity is related to the energy transfer, according to the Kolmogorov 4/5 law. The results show that all values are in good agreement, even for the higher order statistics. In all cases the difference is less than 5%.

	M_1	M_2	Flatness: M_4/M_2^2	Skewness: $M_3/M_2^{3/2}$
Pen4Flow 512^3	0.218	0.228	4.57	2.025
Fastest fine grid 512^3	0.221	0.233	4.36	1.966

Table 3: Moments of the velocity field (averaged over the three components) for the Pen4Flow code (512^3) and for the Fastest code, at a given time instant in the statistically steady state. The data of the finite volume computed on the fine grid are interpolated onto an uniform grid with 512^3 grid points.

Figure 5 (top) shows the probability density functions (PDF), estimated with a histogram using 50 bins, of the velocity at a given time instant in the statistically steady state. We find that the PDFs do almost collapse except for weak negative values where we can observe a small deviation. Note that the PDFs are plotted in log-lin representation, hence the difference is negligible. Spectra of turbulent kinetic energy along the spanwise direction are shown in figure 5 (bottom). A small inertial range with a $k^{-5/3}$ behavior can be identified for $k = 3$ to 10, followed by a faster decay. The spectra of both simulations collapse for low wavenumbers up to $k = 60$. For the Fastest code we then find a saturation, while for the spectral code energy further decays with increasing wavenumber. The saturation might be due to the linear interpolation of the finite volume computation.

The statistical confidence of the second order moments can be quantified by the fourth order moments which are of the same order as the second order moments. Statistical convergence is also reflected by the velocity PDF shown

in Fig. 5 (top). The tails do not show any fluctuations at the extreme values and seem thus well converged.

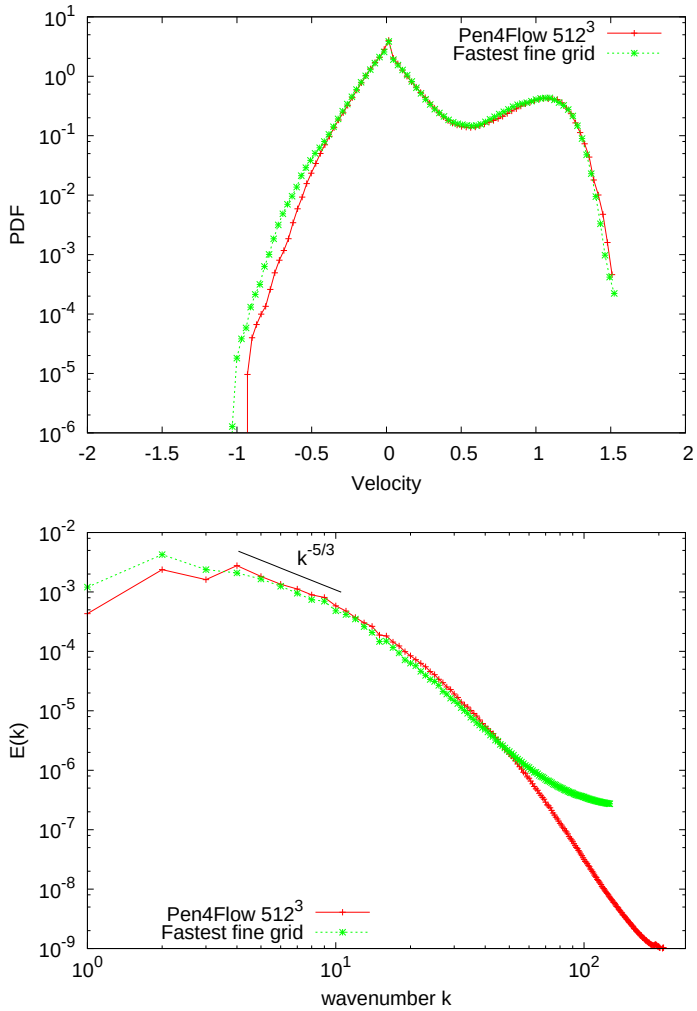

Figure 5: Top: PDF of the velocity (averaged over the three components) at a given time instant in the statistically steady state. Bottom: Turbulent kinetic energy spectrum versus wavenumber k in the spanwise direction. The codes used are Pen4Flow and Fastest.

3.1.2. Mean velocity and Reynolds stress profiles

First we consider the volume penalization code and we study the influence of the resolution on the statistics. Figure 6 shows mean velocity and Reynolds stress profiles at four different streamwise locations, $x = 0.5, 2, 4$ and 6 for three different resolutions, $256^3, 512^2 \times 256$ and 512^3. Note that the profiles are all averaged over time (between 30 and 40 flow through times) and over the spanwise direction. We observe that the influence of the resolution is weak for all mean velocity profiles (Fig. 6 only shows the x-component $\langle u_x \rangle$) while it is much more sensitive for the second order moments, illustrated in Fig. 6 (right) for the cross Reynolds stress $\langle u'_x u'_y \rangle$. Hence in the following we will consider only the two higher resolutions for Pen4Flow.

In Fig. 7 and 8, the coarse grid computations using Fastest show a good agreement for mean velocities with the fine grid computations, similar to what has been observed for Pen4Flow. Concerning the Reynolds stresses the differences are less pronounced than for Pen4Flow, but are still slightly more visible, in particular for the $\langle u'_x u'_y \rangle$ (see Fig. 7, bottom, right).

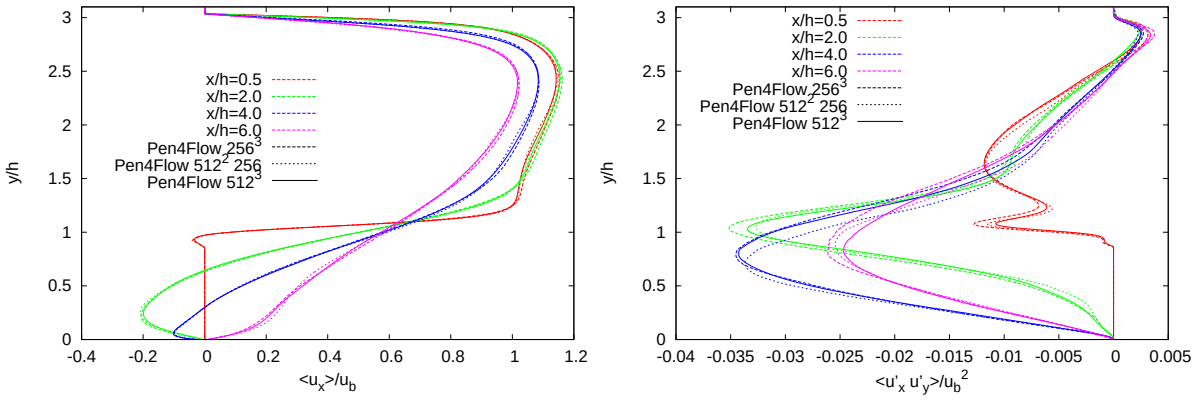

Figure 6: Mean velocity (left) and Reynolds stress (right) profiles at four different locations $x = 0.5, 2, 4$ and 6 for Pen4Flow using different resolutions.

Now, we compare the statistics for the two codes for different resolutions and different streamwise positions. We also plot the results from [1] and we use the notation: BPRM09 for LES (adapted grid) and DNS (MGLET: non uniform spatial grid).

The comparison of the statistics for the two codes is shown for the mean velocity and Reynolds stress profiles at location $x = 0.5$ and $x = 4$ in figure 7 and figure 8, respectively. The profiles are averaged over time and over the spanwise direction. We observe that, the mean velocities $\langle u_x \rangle$ are the same, the mean velocities $\langle u_y \rangle$, the Reynolds stress $\langle u'_x u'_x \rangle$ and $\langle u'_y u'_y \rangle$ almost coincide, except at $x = 0.5$ where we observe a small deviation for $\langle u'_y u'_y \rangle$. The profiles of the Reynolds stress $\langle u'_z u'_z \rangle$ and $\langle u'_x u'_y \rangle$ exhibit slight differences for $x = 2$ and $x = 4$. Significant differences in $\langle u'_z u'_z \rangle$ are observed if we compare with the results from [1], however Pen4Flow and Fastest do still match. The difference is probably due to the fact that, in [1], Breuer et al. plotted the turbulent kinetic energy k instead of the Reynolds stress $\langle u'_z u'_z \rangle$. We checked the turbulent kinetic energy computed from our data and found a good agreement with the $\langle u'_z u'_z \rangle$ obtained in [1].
We can note also that for the mean vertical velocity $\langle u_y \rangle$, at $x = 6$ some fluctuations and an overestimation around $y = 1$ are observed for Pen4Flow. However, the fluctuations tend to disappear by increasing the resolution. The higher moments are more sensitive when we change the methods, but we can conclude that the two codes yield similar results which are in reasonable agreement.

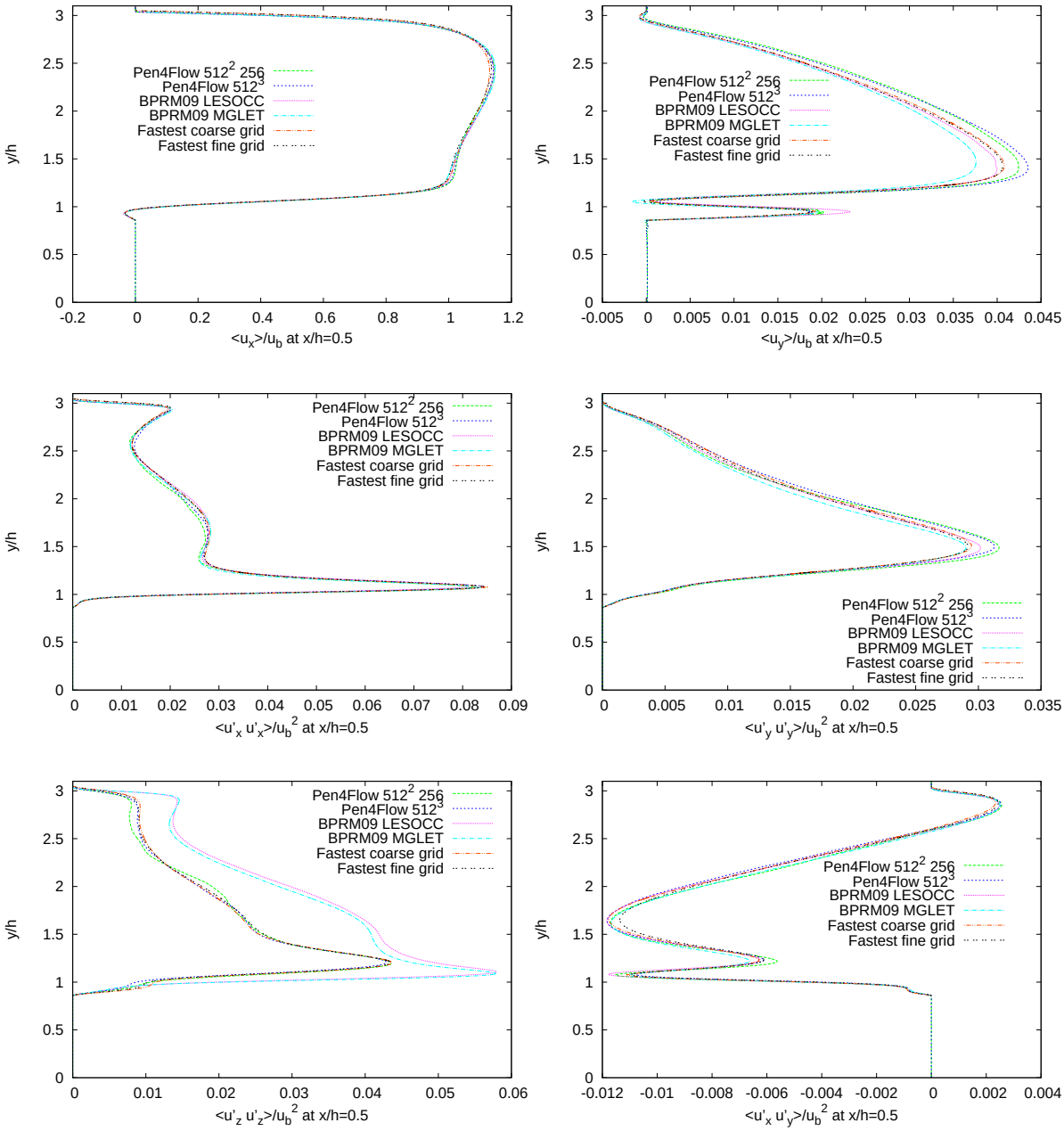

Figure 7: Mean velocity (top) and Reynolds stress profiles (middle and bottom) at $x = 0.5$. The profiles are averaged over time and over the spanwise direction. The codes used are Pen4Flow and Fastest with fine and coarse grids. The results from [1] are also plotted and denoted by BPRM09 for LES (adapted grid) and DNS (MGLET: non uniform spatial grid).

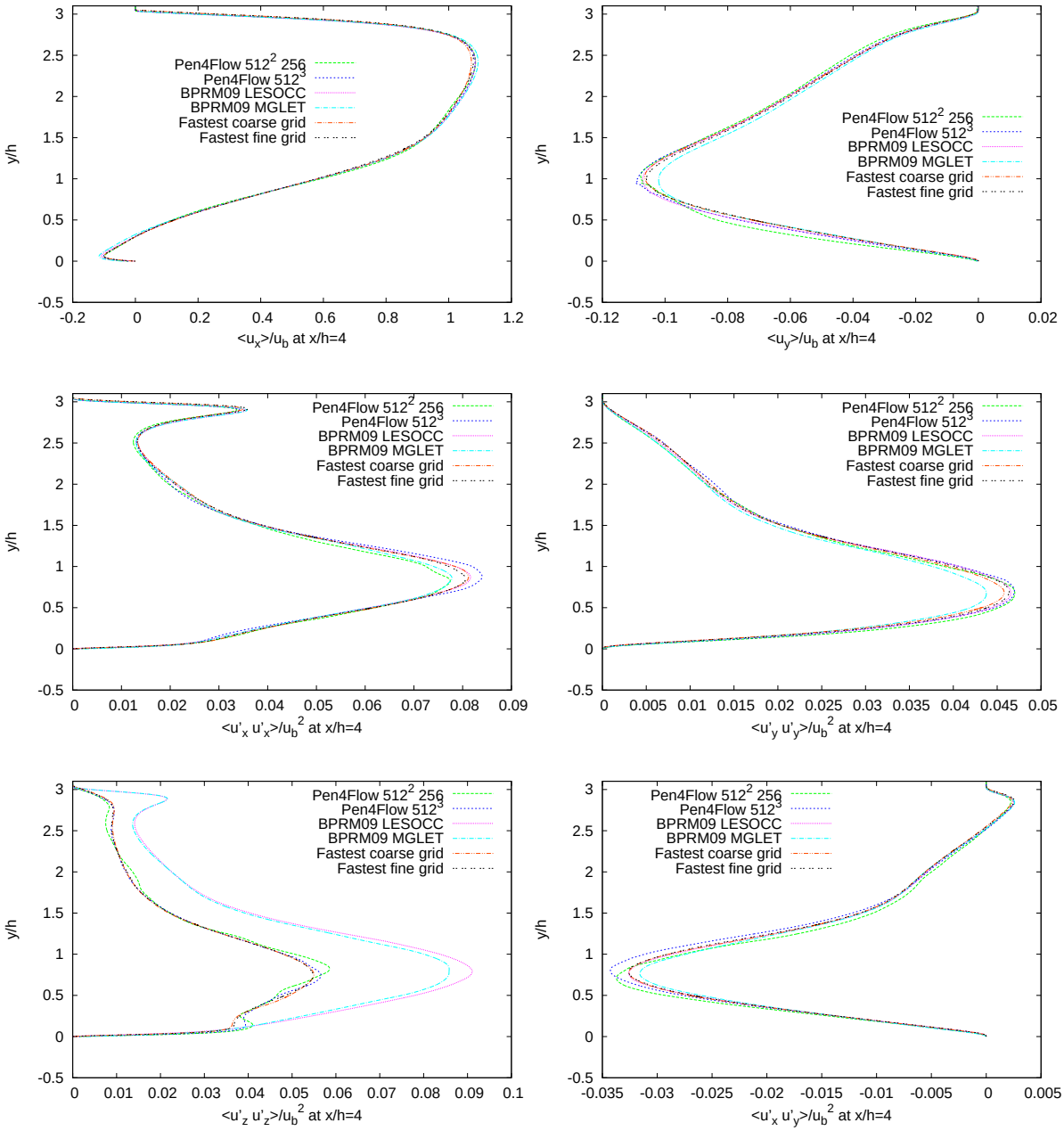

Figure 8: Mean velocity (top) and Reynolds stress profiles (middle and bottom) at $x = 4$. The profiles are averaged over time and over the spanwise direction. The codes used are Pen4Flow and Fastest with fine and coarse grids. The results from [1] are also plotted and denoted by BPRM09 for LES (adapted grid) and DNS (MGLET: non uniform spatial grid).

3.1.3. Wavelet analysis for the velocity fields

Wavelet decomposition of flow fields yields information on scale, position and direction. Scale-dependent statistical analysis of turbulent flows can thus be performed by considering the wavelet coefficients for example of velocity or vorticity, see, *e.g.*, [21]. Wavelet-based directional energies can be defined which capture the properties of velocity gradients. The intermittency of the flow in different directions can be quantified with scale-dependent directional flatness. The scale-dependent velocity flatness was shown to quantify the spatial variability of the energy spectrum [22]. Here we will present applications of scale dependent statistics of the velocity field for the periodic hill flow computed with both methods presented above.

The velocity field, $\boldsymbol{u}(\boldsymbol{x}) = (u_x, u_y, u_z)$, is decomposed into an orthogonal wavelet series: with $\boldsymbol{x} = (x, y, z) \quad \in \quad [0, x_l] \times [0, y_l] \times [0, z_l]$ rescaled onto $[0, 2\pi]$ and given at resolution $N = 2^{3J}$, where J is the number of octaves in each spatial direction. The decomposition of $\boldsymbol{u}$ into an orthogonal wavelet series yields:

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{\boldsymbol{\lambda} \epsilon \boldsymbol{\Lambda}} \tilde{\boldsymbol{u}}_{\boldsymbol{\lambda}} \psi_{\boldsymbol{\lambda}}(\boldsymbol{x}), \tag{4}$$

where the multi-index $\boldsymbol{\lambda} = (j, i_x, i_y, i_z, d)$ denotes the scale j, the position $\boldsymbol{i} = (i_x, i_y, i_z)$, and the directions $d = 1, 2, ..., 7$ of the wavelets. Indeed, the wavelet decomposition in three dimensions yields seven directions due to the tensor product construction [23]. The corresponding index set $\boldsymbol{\Lambda}$ is

$$\boldsymbol{\Lambda} = \{\boldsymbol{\lambda} = (j, i_x, i_y, i_z, d), \quad j = 0, ..., J-1; \quad i_x, i_y, i_z = 0, ..., 2^J - 1 \quad \text{and } d = 1, ..., 7\}. \tag{5}$$

Due to the orthogonality the wavelet coefficients are given by $\tilde{\boldsymbol{u}}_{\boldsymbol{\lambda}} = \langle \boldsymbol{u}, \psi_{\boldsymbol{\lambda}} \rangle$ where $\langle \cdot, \cdot \rangle$ denotes the L^2-inner product, defined by $\langle f, g \rangle = \int_{[0,2\pi]^3} f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x}$, for each velocity component. The coefficients measure fluctuations of $\boldsymbol{u}$ around scale 2^{-j} and around position $2\pi i / 2^j$ in one of the seven possible directions. The N wavelet coefficients $\tilde{\boldsymbol{u}}_{\boldsymbol{\lambda}}$ are efficiently computed from the N grid point values of $\boldsymbol{\omega}$ using the fast wavelet transform, which has linear complexity [24]. In this study, we have chosen the Coiflet 30 wavelet, which has 10 vanishing moments ($\int \boldsymbol{x}^p \psi(\boldsymbol{x}) d\boldsymbol{x} = 0, p = 0, ..., 9$) and which is well adapted to represent the current flow simulations.

The a-order moments dependent on the scale index j and the direction d are defined as:

$$M^a_{(j,d)}(u_i) = \langle \tilde{u_i}^a \rangle_{(j,d)}, \qquad \text{where } \boldsymbol{i} = i_x, i_y, i_z \text{ and } a \in \mathbb{N}. \tag{6}$$

$\langle \rangle_{(j,d)}$ is the average over all positions $\boldsymbol{i}$ of the wavelets for a given direction d and a given scale j.

At a given time instant in the statistically steady state, we applied the wavelet based multi-scale analysis. Scale-dependent moments are shown in figure 9 for Pen4Flow with 512^3 collocation grid points and for the finite volume code with a fine grid interpolated on an uniform Cartesian grid 512^3. Large scale corresponds to $J = 0$ and small scale to $J = 8$. The second order moments show the distribution of turbulent kinetic energy at different scales and yield similar results for both codes. The scale-dependent flatness, which is a measure for intermittency of the flow, obeys to the same behavior, but the difference between the two curves is small. The skewness (not shown here) gives for both cases similar values. Both codes show thus the same behavior for the scale dependent statistics from the largest to the smallest scale.

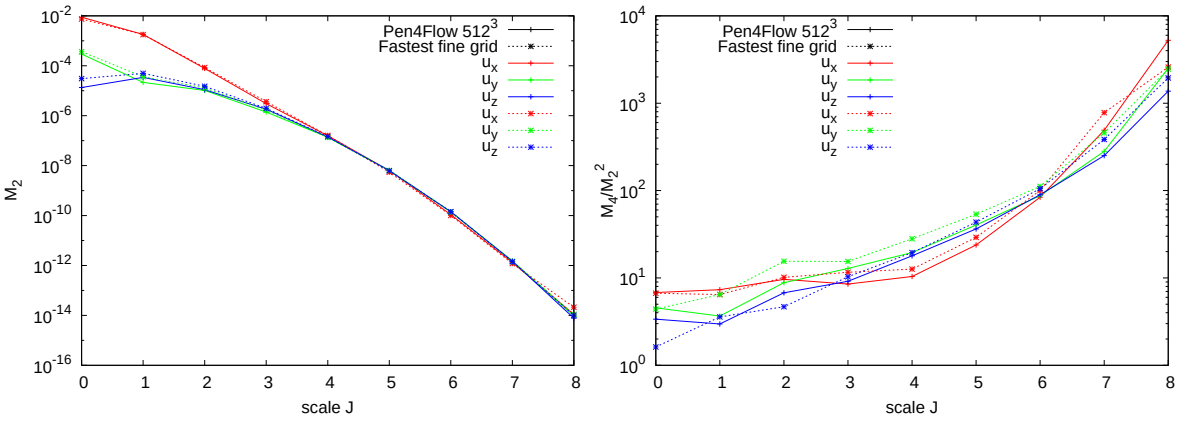

Figure 9: Scale-dependent second order moments (left) and scale dependent flatness (right) for the three different velocity components for Pen4Flow with 512^3 collocation grid points and for Fastest with a fine grid interpolated on an uniform Cartesian grid 512^3.

To get further insight into the scale dependence of the Pen4Flow and the Fastest computations, we consider in Fig. 10 scale dependent moments of the

velocity fields in different spatial directions, defined in eq. (6) using an orthogonal wavelet decomposition. In Fig. 10 (top) the second order moment of the three velocity components in the longitudinal and transverse directions, *i.e.*, for u_x the longitudinal direction corresponds to x while y and z are the transverse directions, is shown. Large scale corresponds to $J = 0$ and small scale to $J = 8$. For clarity, we decided to omit the mixed directions xy, xz, yz, xyz. The scale dependent first (not shown here) and second order moments, are almost the same for both codes. The behaviors of the scale-dependent flatness are also similar; indeed the difference between the two curves is small for all scales, except for the smallest scale $J = 8$ where the deviation is more pronounced.

3.2. Influence of the Reynolds number

In the following, we investigate the influence of the Reynolds number using the Pen4Flow code and compare the results again with those obtained by Breuer et al. [1]. In Table 4, the separation and reattachment points are compared for three Reynolds numbers, $Re = 700, 1400$ and 2800. Slight differences are observed for the separation point, while the discrepancies are larger for the reattachment by comparison with those obtained with BPRM09 LESOCC code.

	x_{sep}/H	x_{reatt}/H
$Re = 700$		
Pen4Flow $N = 256^3$	0.30	5.30
BPRM09 LESOCC	0.29	5.24
$Re = 1400$		
Pen4Flow $N = 512^3$	0.26	5.26
BPRM09 LESOCC	0.26	5.19
$Re = 2800$		
Pen4Flow $N = 512^3$	0.24	5.44
BPRM09 LESOCC	0.21	5.41

Table 4: Reynolds numbers, separation point x_{sep}/H and reattachment point x_{reatt}/H. The results from [1] are also plotted and denoted by BPRM09 for LES (adapted grid).

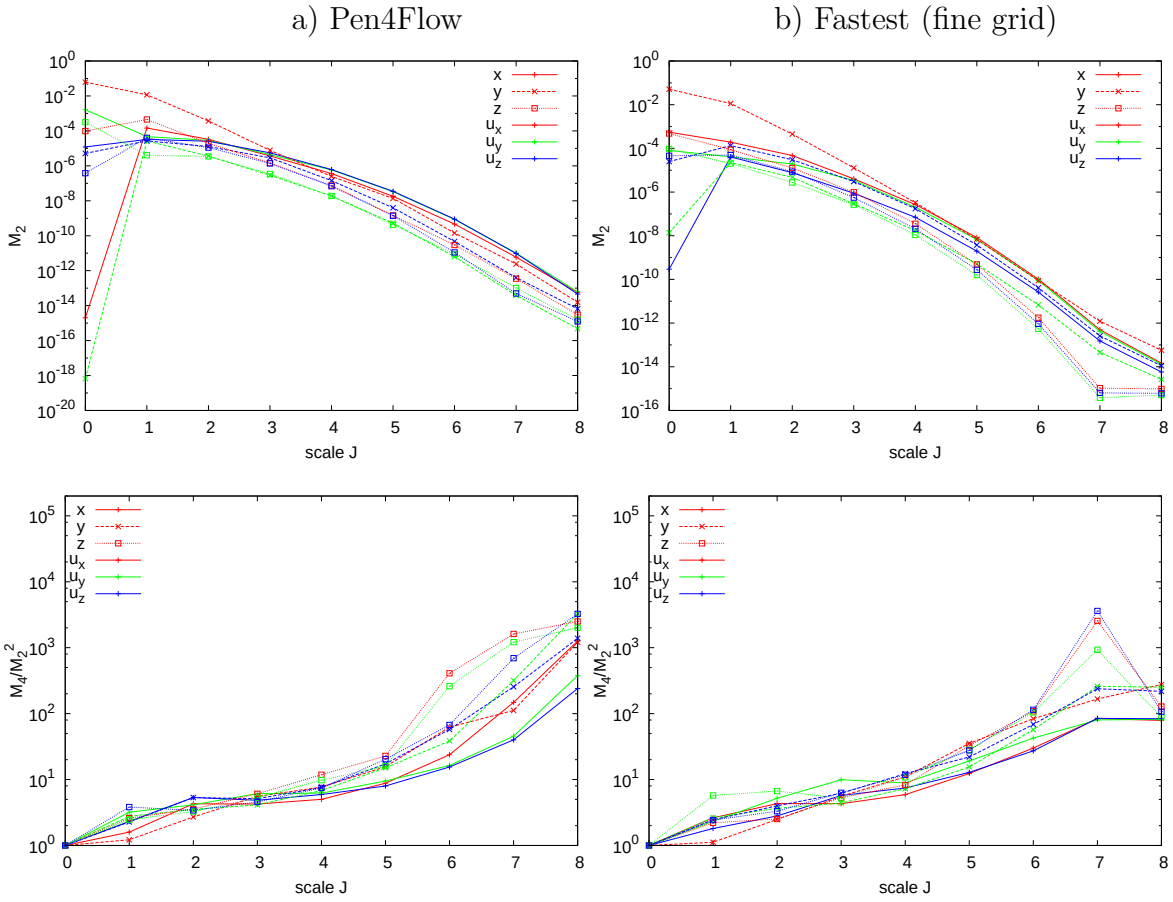

Figure 10: Scale-dependent second order moments (top) and scale dependent flatness (bottom) for the three velocity components in the three main directions. The codes used are Pen4Flow with 512^3 collocation grid points and Fastest with a fine grid interpolated on a Cartesian uniform grid 512^3.

As example in Fig. 11, we selected the position $x/h = 4$ as the comparison of the results for the other positions shows the similar tendencies. Again we start with the mean velocity profile for $\langle u_x \rangle$ and $\langle u_y \rangle$, we observe that for the three Reynolds numbers all curves agree reasonably well. The Reynolds stress profiles, Fig. 11 middle and bottom, reflect the same tendency, however for $\langle u'_x u'_x \rangle$ we observe larger differences compared to the mean velocity. Nevertheless, for the worst case, $Re = 2800$, the differences are below 10%.

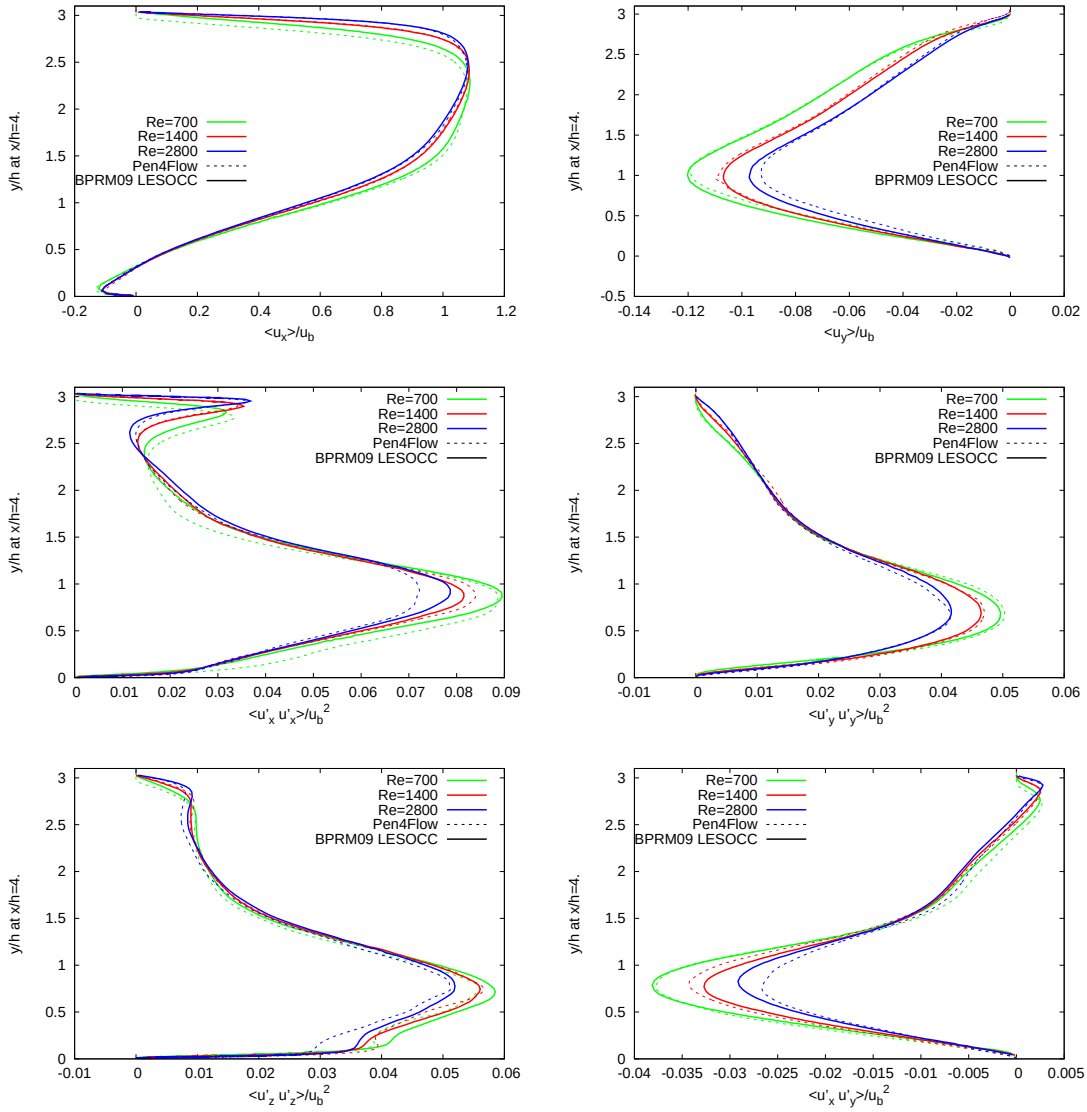

Figure 11: Mean velocity (top) and Reynolds stress profiles (middle and bottom) at $x = 4$ for Reynolds number $Re = 700, 1400, 2800$. The profiles are averaged over time and over the spanwise direction. The code used is Pen4Flow with fine and coarse grids. The results from [1] are also plotted and denoted by BPRM09 for LES (adapted grid).

4. Conclusion

The flow over a periodic hill configuration has been revisited using a Fourier pseudo-spectral code with volume penalization and a finite volume

code with body fitted grids. This classical benchmark involves complex, non-Cartesian geometries and turbulent flow conditions. The presence of a recirculation zone caused by the presence of the hill constitutes an additional difficulty. This detailed study showed that the volume penalization code and the finite volume code reproduce well the physics of this flow and produce similar results. Wavelet analysis enabled us to show that the multi-scale physics is almost the same for both codes. The volume penalization method is a powerful technique to impose Dirichlet boundary conditions in complex geometries using still a Fourier pseudo-spectral method, which has been confronted here for three different Reynolds numbers.

The pros and the cons for Pen4Flow and Fastest can be summarized as follows. Due to the volume penalization method, Pen4Flow is very flexible for changing the geometry and the convergence of the method is mathematically justified and proved in [7]. The penalization method becomes more efficient for higher Re numbers [25] because the penalization boundary layer is proportional to $\sqrt{\eta\nu}$. Moreover for this code, no linear system has to be solved (differential operators are diagonal) and most computational cost is due to the FFT, for which highly efficient parallel implementations are available. We can note also that the Gibbs oscillations are not amplified [11] and the method does not show numerical diffusion and dispersion. However, the penalization term models boundary conditions with first order accuracy only which is also the case for the geometrical errors due to the mask function (staircase effect). Equidistant grids are required in Pen4Flow, which is less efficient as fine grid size is required close to the boundary. Furthermore, the volume penalization technique requires more grid points as the governing equations have to be solved in the whole domain, including both solid and fluid domains. Finally, the time step is limited due to stability restrictions imposed by the penalization term in addition to the classical CFL limit. In conclusion, considering these latter points, the volume penalization method is an attractive alternative to simulate turbulent flows in complex geometries. Fastest takes advantage of body fitted grids which are well adapted to the geometry and involve no interpolation error. As a consequence less grid points are required compared to Pen4Flow. The curvilinear grid with block-structured grids enables to have boundary conditions fulfilled exactly. Nevertheless, Fastest adopts the SIMPLE projection scheme for imposing the incompressibility, which is not exactly fullfilled. The finite volume scheme, used in this code, is second order accurate in space and time, and an alge-

braic system has to be solved in each time step. The parallelization is also less efficient, as shown in Fig. 3. Finally, Fastest is less flexible compared to Pen4Flow since the grid has to be generated for a given geometry.

Interesting perspectives of Pen4Flow are the application to compute turbulent flows in complex geometries which are simply defined by the mask function and which may even vary in time, as it is the case in fluid-structure interaction problems. The implementation of additional transport equations, e.g. for passive scalars, or the Boussinesq approximation is another promising direction. The necessary Neumann boundary conditions can be implemented using the technique proposed in [16].

Acknowledgements

B.K. thankfully acknowledges financial support and hospitality from the Graduate School of Excellence Computational Engineering at TU Darmstadt. B.K. and K.S. thank D. Kolomenskiy for providing the spectral code Pen4Flow. K.S. thanks the international affairs office of Aix Marseille Université for financial support.

[1] M. Breuer, N. Peller, Ch. Rapp and M. Manhart. Flow over periodic hills - Numerical and experimental study in a wide range of Reynolds number. *Computers & Fluids*, **38**:433–457, 2009.

[2] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, **49**:1–23, 1943.

[3] J. Fröhlich, C.P. Mellen, W. Rodi, L.Temmerman and M.A. Leschziner. Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. *Journal of Fluid Mechanics*, **526**, 19-66, 2005.

[4] R. Glowinski and Y. Kuznetsov. On the solution of the Dirichlet problem for linear elliptic operators by a distributed Lagrange multiplier method. *C. R. Acad. Sci. Paris, Sér. I*, **327**:693-698, 1999.

[5] Y. Cheny and O. Botella. The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. *J. Comput. Phys.* **229**:1043-1076, 2010.

[6] C. S. Peskin, . (1972). Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, **10**(2):252–271, 1972.

[7] P. Angot, C.-H. Bruneau and P. Fabrie. A penalization method to take into account obstacles in viscous flows. *Numer. Math.*, **81**:497, 1999.

[8] C.S. Peskin. The immersed boundary method. *Acta Numer.*, **11**:479517, 2002.

[9] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, **37**:23961, 2005.

[10] C. Canuto, M.Y. Hussaini, A. Quarteroni and T.A. Zang. Spectral Methods in Fluid Dynamics. *Springer-Verlag*, 1988.

[11] G. H. Keetels, U. D'Ortona, W. Kramer, H.J.H. Clercx, K. Schneider and G.J.F. Van Heijst. Fourier spectral and wavelet solvers for the incompressible Navier–Stokes equations with volume-penalization: Convergence of a dipolewall collision. *J. Comput. Phys.*, **227**(2) 919–945, 2007.

[12] K. Schneider. Numerical simulation of the transient flow behaviour in chemical reactors using a penalization method. *Comput. Fluids*, **34**:1223–1238, 2005.

[13] D. Kolomenskiy and K. Schneider. A Fourier spectral method for the Navier-Stokes equations with volume penalization for moving solid obstacles. *J. Comput. Phys.*, **228**:5687–5709, 2009.

[14] Fastest - User Manual, Institute of Numerical Methods in Mechanical Engineering. Technische Universität Darmstadt, 2004.

[15] P. H. Chang, C.C. Liao, H.W. Hsu, S.H. Liu and C.A. Lin. Simulations of laminar and turbulent flows over periodic hills with immersed boundary method. *Comput. Fluids*, **92**:233–243, 2014.

[16] B. Kadoch, D. Kolomenskiy, P. Angot and K. Schneider. A volume penalization method for Navier-Stokes flows and scalar advection-diffusion with moving obstacles. *J. Comput. Phys.*, **231**:4365–4383, 2012.

[17] Y. Benarafa, O. Cioni, F. Ducros and P. Sagaut. RANS/LES coupling for unsteady turbulent flow simulation at high Reynolds number on coarse meshes. *Comput. Methods Appl. Mech. Engrg.*, **195**(23–24):2939–2960, 2006.

[18] T. Lehnhäuser and M. Schäfer, Improved linear interpolation practice for finite-volume schemes on complex grids. *Int. J. Numer. Meth. Fluids*, **38**(7):625–645, 2002.

[19] K. Hanjalić, M. Popovac and M. Hadziabdić. A robust near-wall elliptic-relaxation eddy-viscosity turbulence model for CFD. *Int. J. Heat and Fluid Flow*, **25**(6):1047–1051, 2004.

[20] S. B. Pope. Turbulent Flows. *Cambridge University Press*, 2000.

[21] K. Schneider and O. Vasilyev. Wavelet methods in computational fluid dynamics. *Annu. Rev. Fluid Mech.*, **42**:473–503, 2010.

[22] W. J. T. Bos, L. Liechtenstein and K. Schneider. Small scale intermittency in anisotropic turbulence. *Phys. Rev. E*, **76**:046310, 2007.

[23] I. Daubechies. Ten lectures on wavelets. *Philadelphia: Society for industrial and applied mathematics*, **61**:198–202, 1992.

[24] S. Mallat. A Wavelet Tour of Signal Processing. *Academic Press*, 2nd ed. 1999.

[25] R. Nguyen van yen, D. Kolomenskiy and K. Schneider. Approximation of the Laplace and Stokes operators with Dirichlet boundary conditions through volume penalization: a spectral viewpoint. *Numer. Math.*, **128**:301–338, 2014.