Journal of Computational Physics 229 (2010) 2267–2286

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Coherent Vortex Simulation of weakly compressible turbulent mixing layers using adaptive multiresolution methods

Olivier Roussel [a], Kai Schneider [b,c,*]

[a] Institut für Technische Chemie und Polymerchemie (TCP), Universität Karlsruhe (TH), Kaiserstr. 12, 76128 Karlsruhe, Germany
[b] Laboratoire de Modélisation en Mécanique et de Procédés Propres (M2P2), CNRS et Universités d'Aix-Marseille, 38 rue Frédéric Joliot-Curie, 13451 Marseille cedex 20, France
[c] Centre de Mathématiques et d'Informatique (CMI), Université de Provence, 39 rue Frédéric Joliot-Curie, 13453 Marseille cedex 13, France

ARTICLE INFO

Article history:
Received 16 December 2008
Received in revised form 23 November 2009
Accepted 23 November 2009
Available online 27 November 2009

MSC:
65M50
80A32
76V05

Keywords:
Coherent vortex simulation
Turbulence
Finite volumes
Adaptivity
Multiresolution

ABSTRACT

An adaptive mulitresolution method based on a second-order finite volume discretization is presented for solving the three-dimensional compressible Navier–Stokes equations in Cartesian geometry. The explicit time discretization is of second-order and for flux evaluation a 2–4 Mac Cormack scheme is used. Coherent Vortex Simulations (CVS) are performed by decomposing the flow variables into coherent and incoherent contributions. The coherent part is computed deterministically on a locally refined grid using the adaptive multiresolution method while the influence of the incoherent part is neglected to model turbulent dissipation. The computational efficiency of this approach in terms of memory and CPU time compression is illustrated for turbulent mixing layers in the weakly compressible regime and for Reynolds numbers based on the mixing layer thickness between 50 and 200. Comparisons with direct numerical simulations allow to assess the precision and efficiency of CVS.

1. Introduction

The intrinsic multiscale character of turbulent flows in space and time is still a major challenge in computational fluid dynamics. The numerical simulation of fully developed turbulent flows requires turbulence models to reduce the computational complexity as direct approaches are still beyond the frontiers of the available computational resources and hence they will be limited to low Reynolds number flows in the near future.

In the above context the compressible turbulent regime is even more challenging than the incompressible one, as in addition to the large range of involved temporal and spatial scales, shocks appear and interact nonlinearly with coherent vortices. Their accurate and efficient numerical simulation becomes more complex.

Different directions for turbulence modeling have been entered upon. The large eddy simulation approach seems to be the most advanced, however reliable simulation of turbulent flow still requires the tuning of ad hoc parameters [15].

In the recent past, multiscale methods for the simulation of turbulent flows have become more and more fashionable, also for compressible flows. A review can be found in the book of Sagaut and coworkers [22]. Wavelet methods in computational fluid dynamics have been reviewed in [25].

[*] Corresponding author. Address: Centre de Mathématiques et d'Informatique (CMI), Université de Provence, 39 rue Frédéric Joliot-Curie, 13453 Marseille cedex 13, France. Tel.: +33 4 91 11 85 29; fax: +33 4 91 11 35 02.
E-mail addresses: roussel@ict.uni-karlsruhe.de (O. Roussel), kschneid@cmi.univ-mrs.fr (K. Schneider).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.11.034

The Coherent Vortex Simulation (CVS), which fits into the framework of multiscale methods, has been introduced by Farge, Schneider and coworkers for modeling incompressible turbulent flows [8,7]. The underlying idea is the decomposition of the flow into coherent and incoherent contributions by means of an orthogonal wavelet filtering of the vorticity field. The coherent flow is then computed deterministically, while the influence of the incoherent background flow is statistically modelled or neglected. In Okamoto et al. [19] it has been shown that for incompressible isotropic turbulence the number of degrees of freedom N required for CVS grows slower with the Reynolds number Re, i.e. $N \propto Re^{3.9}$, than for DNS where Kolmogorov type arguments imply $N \propto Re^{4.5}$. This motivates the development of CVS for computing fully developed turbulent flows. Adaptive space discretizations are hereby a key ingredient to be able to benefit from the efficient representation of the coherent flow to be computed in terms of memory and CPU time savings.

Adaptive multiresolution schemes have been developed in the past with the subject to speed up classical numerical discretization methods for PDEs, like finite differences or finite volumes, by locally adapting the grid to the solution which can then result in significant memory reduction. Therewith the costly evaluation of the nonlinear terms can also be reduced without degrading the precision of the solution. Three ingredients of multiresolution analysis are hereby essential: estimation of the local regularity of the solution, thresholding of weak coefficients in the multiresolution representation of the solution and fast interpolation of the solution from the locally refined grid to locally equidistant grids. A comprehensible overview on this rapidly growing topic is given in the book of Müller [17].

The aim of the present work is the extension of the CVS method to subsonic compressible turbulent flows using the adaptive multiresolution algorithm originally developed by Roussel and coworkers [21]. The formalism of CVS is adapted to the compressible Navier–Stokes equations which are written in primitive variables. The influence of the wavelet filtering of the conserved quantities is investigated and the choice of the filter tolerance and the normalization of the wavelets is studied.

The remainder of the paper is organized as follows. The governing equations and their numerical discretization are presented in Section 2. Section 3 recalls the multiresolution representation and introduces the CVS method for compressible flows. Section 4 shows various numerical results for weakly compressible, time-developing mixing layers. Finally, in Section 5 conclusion are drawn and some perspectives for future work are given.

2. Governing equations and numerical discretization

2.1. Navier–Stokes equations for compressible fluid flows

We consider a three-dimensional compressible flow of a Newtonian fluid in the Stokes hypothesis in a domain $\Omega \subset \mathbb{R}^3$. Using Einstein's summation convention, the balance equations in Cartesian coordinates can be written in the following dimensionless form, see e.g. [14],

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} = -\frac{\partial}{\partial x_j}(\rho u_j),\\
&\frac{\partial}{\partial t}(\rho u_i) = -\frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{i,j} - \tau_{i,j}),\\
&\frac{\partial}{\partial t}(\rho e) = -\frac{\partial}{\partial x_j}\left((\rho e + p)u_j - u_i\tau_{i,j} - \lambda\frac{\partial T}{\partial x_j}\right).
\end{aligned} \tag{1}$$

In the above equations, ρ, p, T and e denote the dimensionless density, pressure, temperature and specific total energy per unit of mass, respectively; $(u_1, u_2, u_3)^T$ is the dimensionless velocity vector. The components of the dimensionless viscous strain tensor $\tau_{i,j}$ are

$$\tau_{i,j} = \frac{\mu}{Re}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{i,j}\right), \tag{2}$$

where μ denotes the dimensionless molecular viscosity and Re the Reynolds number. The dimensionless conductivity λ is defined by

$$\lambda = \frac{\mu}{(\gamma - 1)Ma^2 Re Pr}, \tag{3}$$

where γ, Ma and Pr, respectively denote the specific heat ratio and the Mach and Prandtl numbers.

The system is completed by an equation of state for a calorically ideal gas

$$p = \frac{\rho T}{\gamma Ma^2}. \tag{4}$$

and suitable initial and boundary conditions.

Assuming the temperature to be larger than 120 K, the molecular viscosity varies with the temperature according to the dimensionless Sutherland law

$$\mu = T^{\frac{3}{2}}\left(\frac{1+T_s}{T+T_s}\right), \tag{5}$$

where $T_s \approx 0.404$.

Denoting by (x, y, z) the three Cartesian directions, this system of equations can be written in the following compact form

$$\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} - \frac{\partial H}{\partial z}, \tag{6}$$

where $U = (\rho, \rho u_1, \rho u_2, \rho u_3, \rho e)^T$ denotes the vector of the conservative quantities, and F, G, H are the flux vectors in the directions x, y, and z, respectively.

3. Multiresolution analysis and coherent vortex simulation

In the following we describe the cell-average multiresolution analysis used to accelerate the previously introduced finite volume scheme. Then the principle of the CVS approach applied to compressible flows is presented.

3.1. Cell-average multiresolution analysis

The principle of the multiresolution analysis is to represent a set of data on a set of nested dyadic grids. The data given on a fine grid is decomposed into values on a coarser grid plus a series of differences at different levels of dyadic grids. These differences contain the information of the solution when going from a coarse to a finer grid. In particular, these coefficients are small in regions where the solution is smooth [12] and yield hence high compression rates for functions with inhomogeneous regularity.

Depending on the choice of the numerical method, several multiresolution analyses can be used. For methods of approximation based on point-values, like e.g. finite differences, a point-value multiresolution analysis is well adapted [13]. In the present paper, to ensure the conservativity of mass, momentum and energy, the finite volume method has been chosen, together with a cell-average multiresolution analysis.

Let us consider a three-dimensional Cartesian domain $\Omega \subset \mathbb{R}^3$ and let us partition it into a set of nested dyadic grids $G_l, l \in \{0, \ldots, L\}$, each one containing 2^{3l} cells $\Omega_{l,i,j,k}, i, j, k \in \{0, \ldots, 2^{l-1}\}$.

The set of nested dyadic grids is usually represented as a *tree structure*. In the present article, we use the following terminology: a tree structure is composed of a *root* cell $\Omega_{0,0,0,0}$, which is the basis of the tree, the *nodes* which are elements of the tree, and the *leaves* which are the upper elements. In three dimensions, a *parent* cell at a level l has always $2^3 = 8$ *children* cells at the level $l+1$. On the left of Fig. 1, a tree structure is represented by its visible part, i.e. its leaves. For the incoming and outgoing flux computations, a leaf at the level l has sometimes no neighbour at the same level and needs to get information from a leaf at the level $l-1$. Therefore, *virtual* leaves are created. They only exist for the flux computation and no time evolution is made on them (Fig. 1, right). In order to be *graded*, the tree must verify that each leaf at a level l has always adjacent cells of level at least equal to $l-1$ in each direction, the diagonal being included (see Fig. 1).

Each node $\Omega_{l,i,j,k}$ of the tree contains the cell-average value of U, denoted by $\overline{U}_{l,i,j,k}$. To compute the average value of a cell at level l from the ones of cells at level $l+1$, we use the projection (or restriction) operator $P_{l+1\to l}$. It is *exact* and *unique*, given that the cell-average value of a *parent* cell is the weighted average value of its *children* cell-averages. Denoting $\overline{\mathbf{U}}_l$ the vector of the cell-average values at the level l, i.e.

$$\overline{\mathbf{U}}_l = (\overline{U}_{l,i,j,k})_{i,j,k},$$

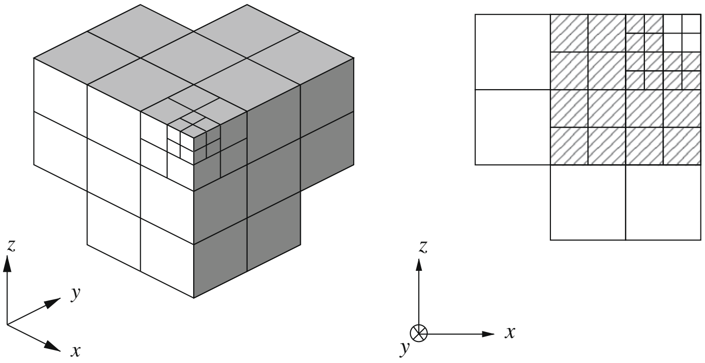

Fig. 1. Left: graded tree structure in 3D. Right: leaves (plain) and virtual leaves (striped) on the (x, z) plan.

we have

$$P_{l+1\to l}:\overline{\mathbf{U}}_{l+1}\mapsto\overline{\mathbf{U}}_{l}.$$

The prediction (or prolongation) operator $P_{l\to l+1}$ maps $\overline{\mathbf{U}}_l$ to an approximation $\widehat{\mathbf{U}}_{l+1}$ of $\overline{\mathbf{U}}_{l+1}$.

$$P_{l\to l+1}:\overline{\mathbf{U}}_{l}\mapsto\widehat{\mathbf{U}}_{l+1}$$

In contrast with the projection operator, there is an infinite number of choices for the definition of $P_{l\to l+1}$. Nevertheless, in order to be applicable in a graded tree structure, it needs to be *local* [2], i.e. based on an interpolation using the s nearest neighbours in each direction, and *consistent with the projection*, i.e. $P_{l+1\to l}\circ P_{l\to l+1}=Id$.

In the present paper, we chose a prediction operator based on an interpolation of the cell-averages in the *first* nearest neighbours in every direction, diagonal included. This operator in one, two and three dimensions is fully described in [21].

The *detail* is the difference between the exact and predicted values. The vector $\overline{\mathbf{D}}_l$ of the details at level l therefore verifies $\overline{\mathbf{D}}_l=\overline{\mathbf{U}}_l-\widehat{\mathbf{U}}_l$. Thanks to the consistency assumption, the sum of the details on all the children of a parent cell is equal to zero [12]. For a general curvilinear grid with non-equidistant points, the sum of the weighted details is equal to zero, the weights being the cell sizes: length in 1D, area in 2D, and volume in 3D.

Therefore, in three dimensions, the knowledge of the 8 *children* cell-averages of a given parent cell is equivalent to the knowledge of the *parent* cell-average and 7 *details*, i.e. $\overline{\mathbf{U}}_l\leftrightarrow(\overline{\mathbf{U}}_{l-1},\overline{\mathbf{D}}_l)$. Repeating the operation on L levels, one gets the so-called *multiresolution transform* [12]

$$\overline{\mathbf{M}}:\overline{\mathbf{U}}_L\mapsto(\overline{\mathbf{U}}_0,\overline{\mathbf{D}}_1,\ldots,\overline{\mathbf{D}}_L). \tag{7}$$

In other words, the ensemble of cell-average values on the level **L** is equivalent to the cell-average value on the root cell and the ensemble of details from the level 1 to the level **L**.

The threshold operator consists in removing leaves where details are smaller than a prescribed tolerance, without violating the graded tree data structure. After thresholding, one more level is also added to forecast the tree evolution at the next time step. The choice for the accurate value of the tolerance for each scale in fact depends on the normalization of the corresponding wavelet basis. This is the subject of the next subsections.

The cell-average multiresolution analysis is intimely linked to a biorthogonal wavelet basis [2,3]. Hence a quantity u can be expressed as follows,

$$u(x,y,z)=\sum_{i,j,k}\bar{u}_{0,i,j,k}\phi_{0,i,j,k}(x,y,z)+\sum_{l,i,j,k,\mu}\bar{d}^{\mu}_{l,i,j,k}\psi^{\mu}_{l,i,j,k}(x,y,z), \tag{8}$$

with the scaling coefficients $\bar{u}_{0,i,j,k}=\langle u,\tilde{\phi}_{0,i,j,k}\rangle$ corresponding to the cell-averages at level $l=0$ and the wavelet coefficients $\bar{d}^{\mu}_{l,i,j,k}=\left\langle u,\tilde{\psi}^{\mu}_{l,i,j,k}\right\rangle$ corresponding to the details at level l in one of the possible 7 directions μ, and where $\langle f,g\rangle=\int_\Omega f(x,y,z)g(x,y,z)dxdydz$ denotes the L^2-product. The couple $\phi,\tilde{\phi}$ denotes the biorthogonal scaling functions and $\psi,\tilde{\psi}$ are the corresponding biorthogonal wavelets [4].

3.2. Time evolution

The time evolution is performed on the leaves of the tree only. The cell-average values in the other nodes are obtained by projection from the leaves. In order to avoid numerical diffusion, we use an explicit 2–4 Mac Cormack scheme, which is second-order accurate in time, fourth-order in space for the convective terms, and second-order in space for the diffusive terms (see, Gottlieb and Turkel [11]). The time integration is decomposed into two stages, the first stage being decentered in one direction, the second one in the other direction. To avoid an accumulation of error in one direction, we alternate the directions at each time step. One gets the following scheme to compute the cell-averages $\overline{U}^{n+1}$ at time t^{n+1} from the given cell-averages $\overline{U}^n$ at time t^n

$$\begin{aligned}\overline{U}^{*}_{l,i,j,k}=\overline{U}^{n}_{l,i,j,k}&+\Delta t\left(\frac{-7\overline{F}^{n}_{l,i,j,k}+8\overline{F}^{n}_{l,i+1,j,k}-\overline{F}^{n}_{l,i+2,j,k}}{6\Delta x_l}\right)+\Delta t\left(\frac{-7\overline{G}^{n}_{l,i,j,k}+8\overline{G}^{n}_{l,1,j+1,k}-\overline{G}^{n}_{l,i,j+2,k}}{6\Delta y_l}\right)\\&+\Delta t\left(\frac{-7\overline{H}^{n}_{l,i,j,k}+8\overline{H}^{n}_{l,1,j,k+1}-\overline{H}^{n}_{l,i,j,k+2}}{6\Delta z_l}\right),\end{aligned} \tag{9}$$

$$\begin{aligned}\overline{U}^{n+1}_{l,i,j,k}=\frac{\overline{U}^{n}_{l,i,j,k}+\overline{U}^{*}_{l,i,j,k}}{2}&+\frac{\Delta t}{2}\left(\frac{-7\overline{F}^{n}_{l,i,j,k}+8\overline{F}^{n}_{l,i-1,j,k}-\overline{F}^{n}_{l,i-2,j,k}}{6\Delta x_l}\right)+\frac{\Delta t}{2}\left(\frac{-7\overline{G}^{n}_{l,i,j,k}+8\overline{G}^{n}_{l,1,j-1,k}-\overline{G}^{n}_{l,i,j-2,k}}{6\Delta y_l}\right)\\&+\frac{\Delta t}{2}\left(\frac{-7\overline{H}^{n}_{l,i,j,k}+8\overline{H}^{n}_{l,1,j,k-1}-\overline{H}^{n}_{l,i,j,k-2}}{6\Delta z_l}\right),\end{aligned} \tag{10}$$

where $\Delta x_l = 2^{-l}L_x,\ \Delta y_l = 2^{-l}L_y$, and $\Delta z_l = 2^{-l}L_z$ denote the space steps at a level l in the three directions. Here L_x, L_y, and L_z, denote the lengths of the computational domain in the x, y, and z directions, respectively.

For the computation of the diffusive terms, we do not need to use a decentered scheme. Here the diffusive terms are approximated the same way as if we were using a second-order Runge–Kutta–Heun method in time, together with a second-order centered scheme in space.

3.3. Principle of CVS

The CVS of incompressible turbulent flows is based on the decomposition of the vorticity $\omega = \nabla \times u$ into coherent and incoherent parts using thresholding of the wavelet coefficients. The corresponding coherent and incoherent velocity fields are then obtained by applying Biot–Savart's kernel. In the CVS method, the evolution of the coherent flow is then computed deterministically in a dynamically adapted wavelet basis and the influence of the incoherent components is statistically modelled or simply neglected [8,7,24].

For compressible flows, the situation is different, since both vortical and potential components are present. Here we decided to decompose the conservative variables $U = (\rho,\ \rho u_1,\ \rho u_2,\ \rho u_3,\ \rho e)$ into a biorthogonal wavelet series (8) by applying the cell-average multiresolution transform previously described. In [20], we applied the same biorthogonal wavelet decomposition to incompressible isotropic turbulence to decompose the flow into coherent and incoherent parts. We concluded that the biorthogonal decomposition yield reasonable results compared to the orthogonal wavelet decomposition typically used for CVS, which justifies the application of the former to compressible turbulence. A decomposition of the conservative variables into coherent and incoherent components is then obtained by decomposing the conservative variables into wavelet coefficients, applying a thresholding and reconstructing the coherent and incoherent contributions from the strong and weak coefficients, respectively.

First the dimensionless density and pressure are decomposed into

$$\begin{aligned} \rho &= \rho_C + \rho_I, \\ p &= p_C + p_I. \end{aligned} \tag{11}$$

where ρ_C and p_C, respectively denote the coherent part of the density and pressure fields, while ρ_I and p_I denote the corresponding incoherent parts.

Then the other remaining variables, i.e. the velocity components u_1, u_2, u_3, the temperature T and energy e, are decomposed using the Favre averaging technique [9], i.e. density weighted, as done in RANS and LES of compressible flows to simplify modeling [15]. This technique avoids the introduction of unresolved terms into the equation of mass conservation and trilinear terms into the momentum equation. For a quantity φ corresponding to one of these remaining variables, we obtain the following decomposition,

$$\varphi = \varphi_C + \varphi_I, \quad \text{where} \quad \varphi_C = \frac{(\rho\varphi)_C}{(\rho)_C}. \tag{12}$$

For sake of clarity, we use the notation $\varphi_C = \bar{\varphi}$. Retaining only the coherent contributions of the conservative variables we obtain the filtered compressible Navier–Stokes equations which describe the flow evolution of the coherent flow

$$\begin{aligned} &\frac{\partial \bar{\rho}}{\partial t} = -\frac{\partial}{\partial x_j}(\bar{\rho}\bar{u}_j), \\ &\frac{\partial}{\partial t}(\bar{\rho}\bar{u}_i) = -\frac{\partial}{\partial x_j}(\bar{\rho}\bar{u}_i\bar{u}_j + \bar{p}\delta_{i,j} - \bar{\tau}_{i,j} + A_{i,j}), \\ &\frac{\partial}{\partial t}(\bar{\rho}\bar{e}) = -\frac{\partial}{\partial x_j}\left((\bar{\rho}\bar{e} + \bar{p})\bar{u}_j - \bar{u}_i\bar{\tau}_{i,j} - \bar{\lambda}\frac{\partial \overline{T}}{\partial x_j} + B_j\right), \\ &\bar{p} = \frac{\bar{\rho}\overline{T}}{\gamma Ma^2}, \end{aligned} \tag{13}$$

where the terms

$$\begin{aligned} A_{i,j} &= \bar{\rho}\left(\overline{u_i u_j} - \bar{u}_i\bar{u}_j\right) - \overline{\tau_{i,j}} + \bar{\tau}_{i,j}, \\ B_j &= \bar{\rho}(\overline{u_j e} - \bar{u}_j\bar{e}) + \overline{u_j p} - \bar{u}_j\bar{p} - \overline{u_j\tau_{i,j}} + \bar{u}_j\bar{\tau}_{i,j} - \overline{\lambda\frac{\partial T}{\partial x_j}} + \bar{\lambda}\frac{\partial \overline{T}}{\partial x_j}, \end{aligned} \tag{14}$$

constitute the influence of the incoherent contributions. In the current paper, these terms are completely neglected. We also use the approximation $\bar{\mu} = \mu(\overline{T})$ and $\bar{\lambda} = \lambda(\overline{T})$. Note that modeling of the incoherent contributions, similar to LES approaches, has been proposed in [10], which allows for larger choices of the threshold and thus higher compression rates. A review of the different modeling strategies can be found in [25].

3.4. Choice of the thresholding rule

In the CVS of incompressible turbulence, the vorticity field is decomposed into orthogonal wavelets and filtered using a fixed threshold ϵ_0 based on the enstrophy, i.e. on the L^2-norm of vorticity.

In most papers on adaptive multiresolution methods [2,4,17,21], the conserved quantities are filtered using furthermore biorthogonal wavelets which are normalized with respect to the L^1-norm, which is well adapted to the finite volume discretization, i.e. $\left\|\tilde{\psi}^{\mu}_{j,i,j,k}\right\|_{L^1} = 1$. In the adaptive multiresolution method (see e.g. [21]) typically detail coefficients $\bar{d}^{\mu}_{j,i,j,k} = \left\langle u,\ \tilde{\psi}^{\mu}_{j,i,j,k}\right\rangle$ of a conserved quantity are removed if

$$\left|\bar{d}^{\mu}_{l,i,j,k}\right| < 2^{d(l-L)}\epsilon, \tag{15}$$

where d denotes the space dimension (here $d = 3$), ϵ the threshold, and $\mathbf{L}$ is the finest level present in the computations. The above choice of the threshold takes the L^1-normalization of the wavelets into account. The computations using this L^1 based thresholding rule are denoted by "Norm #1" results.

To investigate the influence of the thresholding rule and to compare with the classical CVS scheme which uses L^2-normalization of the wavelets and filters vorticity, which corresponds to the anti-symmetric part of the velocity gradient tensor, we consider two additional thresholding rules:

- one using the L^2-normalization of the wavelets for which we obtain the thresholding rule

  $$\left|\bar{d}^{\mu}_{l,i,j,k}\right| < 2^{d(l-L)/2}\epsilon, \tag{16}$$

  denoted in the following "Norm #2" results,
- and one based on the H^1-normalization of the wavelets for which we obtain the thresholding rule

  $$\left|\bar{d}^{\mu}_{l,i,j,k}\right| < 2^{(d-2)(l-L)/2}\epsilon, \tag{17}$$

  denoted in the following "Norm #3" results. The H^1 thresholding rule is motivated by the fact that here the filtering is adapted to the gradients, which mimics filtering of vorticity. However, in addition to vortical regions, strain-dominated regions are also retained in the coherent velocity field.

To apply the threshold to the vector valued function U of the dimensionless conservative quantities, we divide each component of the detail vector by the mean-value of the corresponding dimensionless quantity in the whole computational domain. Concerning the velocity vector (u_1, u_2, u_3), to avoid anisotropy in the threshold of the details, we compute only one detail for the velocity

$$\bar{d}^{\mu}_{l,i,j,k}(u) = \sqrt{\left(\bar{d}^{\mu}_{l,i,j,k}\right)^2(u_1) + \left(\bar{d}^{\mu}_{l,i,j,k}\right)^2(u_2) + \left(\bar{d}^{\mu}_{l,i,j,k}\right)^2(u_3)},$$

and we divide this component of the detail vector by the mean-value of the velocity in the whole computational domain.

Details of all components of U are removed only if all components are below the given threshold. The incoherent part U_I corresponds hence to the conservative quantities for which, for a given index $\{l, i, j, k, \mu\}$, all details are below the threshold. The coherent part U_C corresponds to the remainder. This choice implies that only one tree data structure is necessary to store all coherent components of the conserved variables. By construction we have $U = U_C + U_I$, however the sum is not orthogonal due to the use of biorthogonal wavelets instead of orthogonal ones (see e.g. [20] for a discussion on this topic).

3.5. Summary of the CVS algorithm

In the following, we briefly summarize the CVS algorithm. For more details on the general adaptive MR algorithm, we refer to [21]. First, depending on the initial condition, an *initial graded tree* is created. Then, given the graded tree structure, a *time evolution* is made on the *leaves*. Finally, details are computed by *multiresolution transform*, in order to *remesh* the tree.

Denoting by $\overline{\mathbf{E}}(\Delta t)$ the discrete *time evolution* operator, the global algorithm can schematically be summarized by

$$\overline{\mathbf{U}}^{n+1} = \overline{\mathbf{M}}^{-1} \cdot \mathbf{T}(\epsilon) \cdot \overline{\mathbf{M}} \cdot \overline{\mathbf{E}}(\Delta t) \cdot \overline{\mathbf{U}}^{n}, \tag{18}$$

where $\overline{\mathbf{M}}$ is the *multiresolution transform* operator, and $\mathbf{T}(\epsilon)$ is the *threshold* operator with tolerance ϵ. The threshold operator depends on the choice of the thresholding rule.

4. Numerical results

As test-case for the CVS method we consider in the following a three-dimensional weakly compressible temporally developing turbulent mixing layer. For a complete description of this test-case in the LES framework, we refer the reader to [26,27].

The CVS results are compared with a DNS reference computation, using a finite volume scheme on the finest regular grid. We study the impact of the different thresholding rules, the choice of the threshold, and the influence of the Reynolds number to assess the precision and efficiency of CVS.

4.1. Flow configuration of the mixing layer

We initialize the test-case by setting two layers of a fluid stacked one upon the other one, each of them with the same velocity norm but opposed directions (Fig. 2).

For every computation with $L = 7$ scales, the computational domain is a three-dimensional cube $\Omega = [-30, 30]^3$ with side-length $L_X = L_Y = L_Z = 60$, and the final time of all computations corresponds to $t = 80$. We set periodic boundary conditions for the x- and y-direction and Neumann conditions are imposed in the z-direction, i.e. on the top and bottom boundaries. The Prandtl and Mach numbers are set to 0.71 and 0.3 respectively, whereas the specific heat ratio γ equals 1.4. The CFL number is set to 0.4 and the maximal resolution is 128^3, which corresponds to $L = 7$ scales. Then, in Section 4.5, $L = 8$ scales are considered, i.e. a maximal resolution of 256^3.

In this test-case, the initial dimensionless velocity profile is

$$\mathbf{u}_0(\mathbf{x}) = \begin{pmatrix} \tanh(z) \\ 0 \\ 0 \end{pmatrix} \tag{19}$$

and the initial dimensionless density and temperature profiles are constant and set to 1.

A sinusoidal perturbation of the form

$$u'(\mathbf{x}) = A(z)\left(\frac{1}{2}\cos\frac{2\pi(x+y)}{L} + \frac{1}{2}\cos\frac{2\pi(x-y)}{L} + \sum_{k=1}^{b-1}\cos\frac{2^{k+1}\pi x}{L}\right), \tag{20}$$

is superimposed to the initial velocity in the streamwise direction, where

$$A(z) = \frac{1}{b\,\cosh^2(z)}.$$

is the amplitude of the perturbation, $L = 60$ is the length of the computational domain, and b denotes the maximal number of modes used for the computation, $b = 3$ in this case. The $\cosh^2(z)$ term is added to limit the perturbation to the junction of the two stacked-up fluid layers. This way, we spare unnecessary waste of CPU resources. Here, we have added only one oblique mode disturbance, so that we can distinguish more easily the structures. The same amount of intensity is given to each of the three modes.

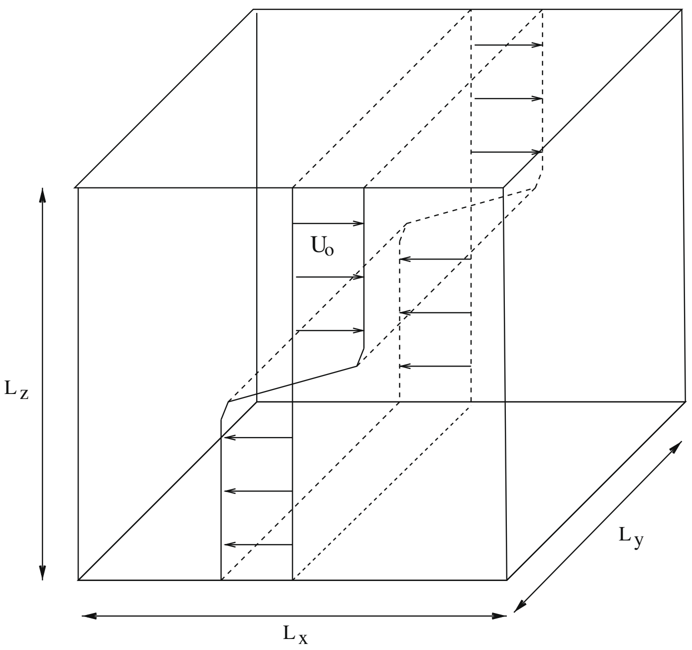

Fig. 2. Flow configuration: domain and initial basic flow $\mathbf{u}_0$ of the three-dimensional mixing layer.

4.2. Time evolution of a mixing layer

For the DNS reference computation, we use the finite volume scheme on the regular grid with resolution 128^3. The Reynolds number based on the initial velocity and half the initial layer thickness is set to $Re = 100$.

Fig. 3 shows the time evolution of the mixing layer.

At $t \approx 19$, the Kelvin–Helmholtz instability generates four rollers in the streamwise direction (Fig. 3, left side). The number of vortices at the initial stage depends on the number of modes b used in the initial perturbance and of the size of the

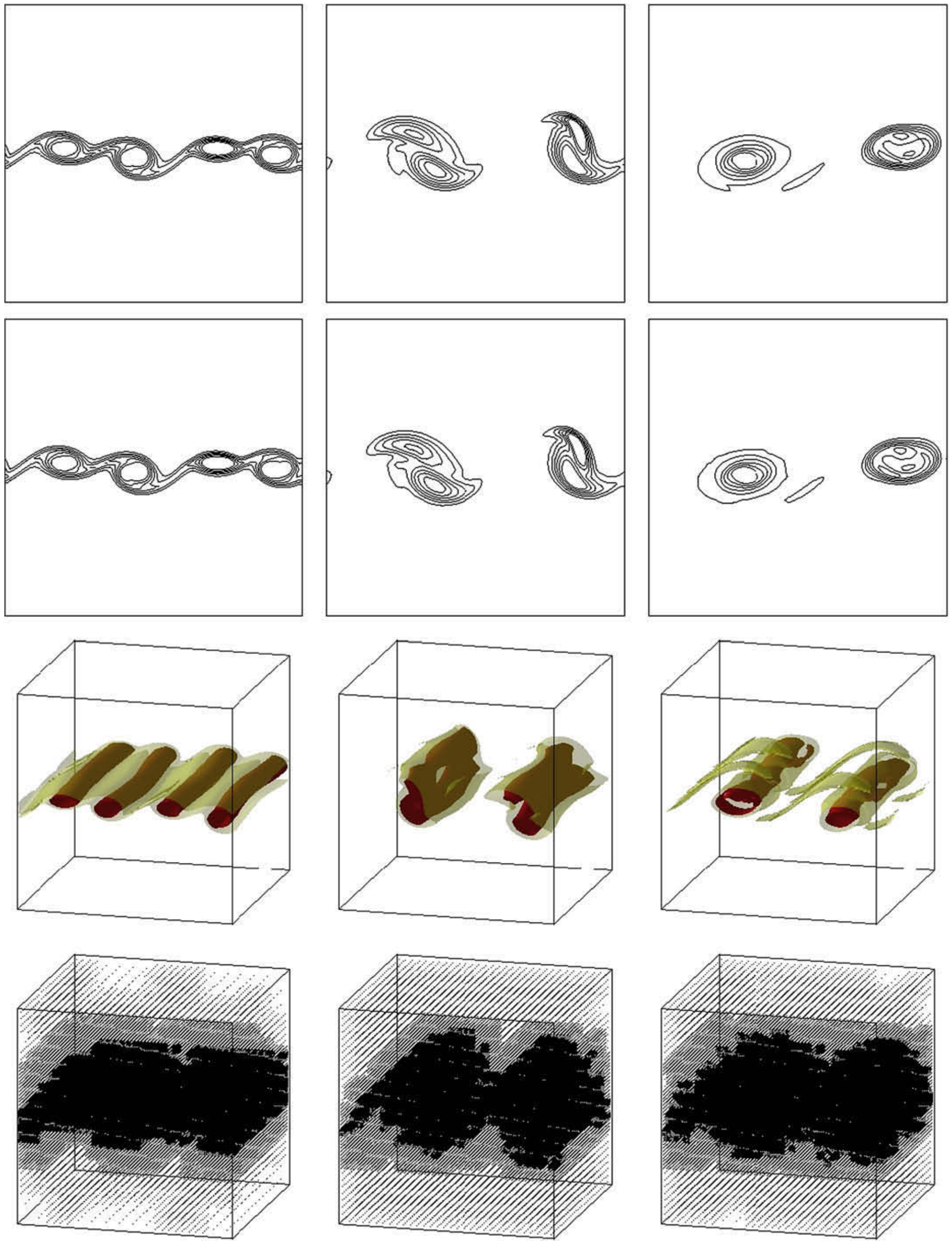

Fig. 3. Time evolution of a weakly compressible mixing layer for $Re = 100$. First row: isolines of vorticity in the plane $y = 0$ for the DNS solution. Second row: same isolines for the CVS computation with $\epsilon = 0.03$ and Norm #3. Third row: Corresponding isosurfaces of vorticity $\|\omega\| = 0.5$ (*black*) and $\|\omega\| = 0.25$ *(gray)*. Fourth row: Adaptive meshes for the CVS computation. The time instants are $t = 19$ (*left*), $t = 37$ (*center*) and $t = 78$ (*right*).

domain Ω. At the beginning, the mixing layer remains approximately two-dimensional, since we have introduced only one oblique mode. Later on, the vortices begin to pair so that, at $t \approx 37$, we observe two vortex pairings (Fig. 3, center). At $t \approx 78$, these two pairings are finished and three-dimensional structures appear, generated by the oblique mode between the two remaining vortices (Fig. 3, right side). For longer computational times, assuming that the domain is large enough, these two vortices would pair again, thus leading to only one vortex.

4.3. *Choice of the thresholding rule*

In this section, we perform a series of CVS computations for $Re = 100$ using the three different thresholding rules (Norms #1, #2 and #3) previously presented, and compare the results with the reference DNS computation on the finest grid, using the same numerical scheme. For each thresholding rule, we perform first several computations with different tolerances to determine its optimal value ϵ_{opt}. To assess which thresholding rule yields the best results, we compare then the energy spectra, the energy and enstrophy evolutions, as well as the corresponding computational resources (CPU time and memory) spent.

Once we have the optimal value for each norm, we compare those optimal computations and decide which norm shows better results.

4.3.1. *Results using the L^1-normalization (Norm #1)*

In Table 1, we show the results of the CVS computations for the Norm #1 together with the DNS reference run. We observe that the CVS computations require much less CPU time (between 26% and 40%) and memory (between 26% and 43%) than DNS, except for a very small value of $\epsilon = 0.01$. For the latter we see that the results of CVS are almost identical with the DNS run, as energy and enstrophy is almost perfectly retained. Memory is still reduced (<60%) but the CPU time is almost doubled with respect to DNS. The time evolution of energy shown in Fig. 4 (left side) illustrates that, for all considered values of ϵ, the CVS result matches well with the DNS reference solution. For the evolution of the enstrophy (Fig. 4, right side), we see however that the dynamics differs from the DNS reference, which shows that the numerical dissipation of the CVS is not sufficient.

The spectral distribution of energy in the streamwise direction is given in Fig. 5. For ϵ = 0.2 and 0.1, the CVS results agree well with the DNS reference in the large scales, but still differ in the small scales.

Table 1
Results obtained with the Norm #1. Percentages of CPU time (required on a Pentium IV, 2.5 GHz), memory compression, and errors in comparison with the DNS solution for kinetic energy E and enstrophy Z at $t = 80$ with different ϵ and for $N = 128^3$.

Method	ϵ	CPU time	% CPU	% *Mem*	% *E*	% *Z*
DNS		7 d 6 h	100	100	100	100
MR	0.3	2 d 2 h	28.74	25.96	99.83	83.19
MR	0.25	2 d 14 h	35.63	29.48	99.92	83.80
MR	0.2	2 d 15 h	36.21	33.44	99.92	84.85
MR	0.1	3 d 4 h	43.68	42.91	99.89	94.18
MR	0.01	13 d 20 h	190.80	75.93	99.98	99.99

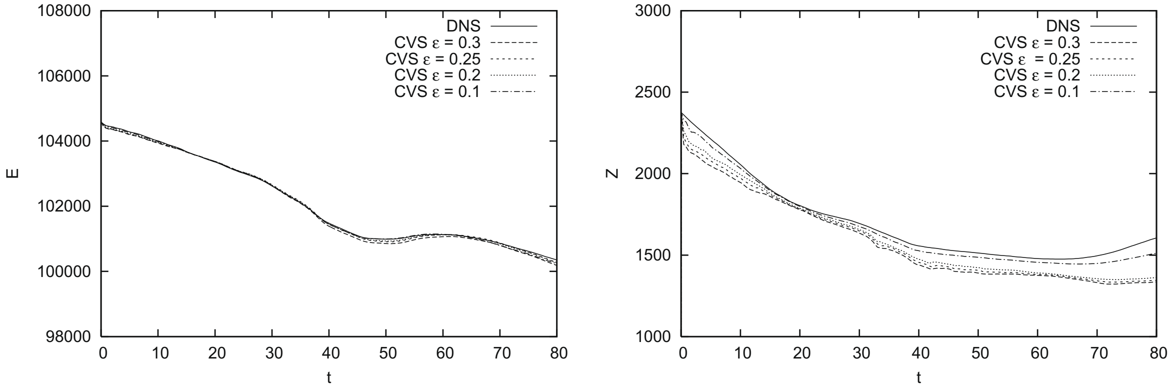

Fig. 4. Results obtained with the Norm #1. Kinetic energy vs. time (*left*) and enstrophy vs. time (*right*) for the reference DNS and the CVS with several tolerance values.

4.3.2. Results using the L^2-normalization (Norm #2)

Table 2 shows the results of the CVS computations for the Norm #2, together with the DNS reference computation. The CVS computations exhibit a significant reduction of CPU time and memory requirements with respect to the reference DNS computation, while the final kinetic energy and enstrophy are reasonably well predicted.

Fig. 6 shows the time evolution of the kinetic energy and the enstrophy for the Norm #2. The evolution of the enstrophy for the CVS computations (Fig. 6, right side) follows well the DNS reference computation until $t \approx 30$. For longer times, we observe a significant error growth. In particular for $\epsilon = 0.08$, the CVS underpredicts the enstrophy, the difference with respect to DNS becoming larger than 7% at $t = 80$. The evolution of the kinetic energy (Fig. 6, left side) presents, as for the Norm #1 results, a good agreement with the reference DNS computation, whatever the tolerance.

Concerning the final streamwise energy spectra (Fig. 7), it is shown that the agreement between the CVS computations and the reference DNS is significantly improved when compared to the results of the Norm #1. For the large scales, i.e. when $k < 6$, the CVS spectra almost coincide with the DNS. However, for $k > 6$, the differences becomes visible.

Still, the slight difference between CVS and DNS in the large scales ($k < 6$) involves a larger energy loss than the apparently larger energy difference in the smaller ones ($k > 6$), because of the logarithmically scaled plot. As we can observe in Table 2, the CVS with $\varepsilon = 0.08$ and the CVS with $\varepsilon = 0.07$ yield, respectively relative errors on energy e_E of 0.0004 and 0.0011. Norm #2 presents better results than the Norm #1, but the results on enstrophy are not accurate enough.

4.3.3. Results using the H^1-normalization (Norm #3)

Fig. 8 shows that CVS computations based on the H^1-normalization yield better results for the enstrophy and follow well the dynamics of the DNS computation. Hereby the difference between DNS and CVS decreases for decreasing threshold val-

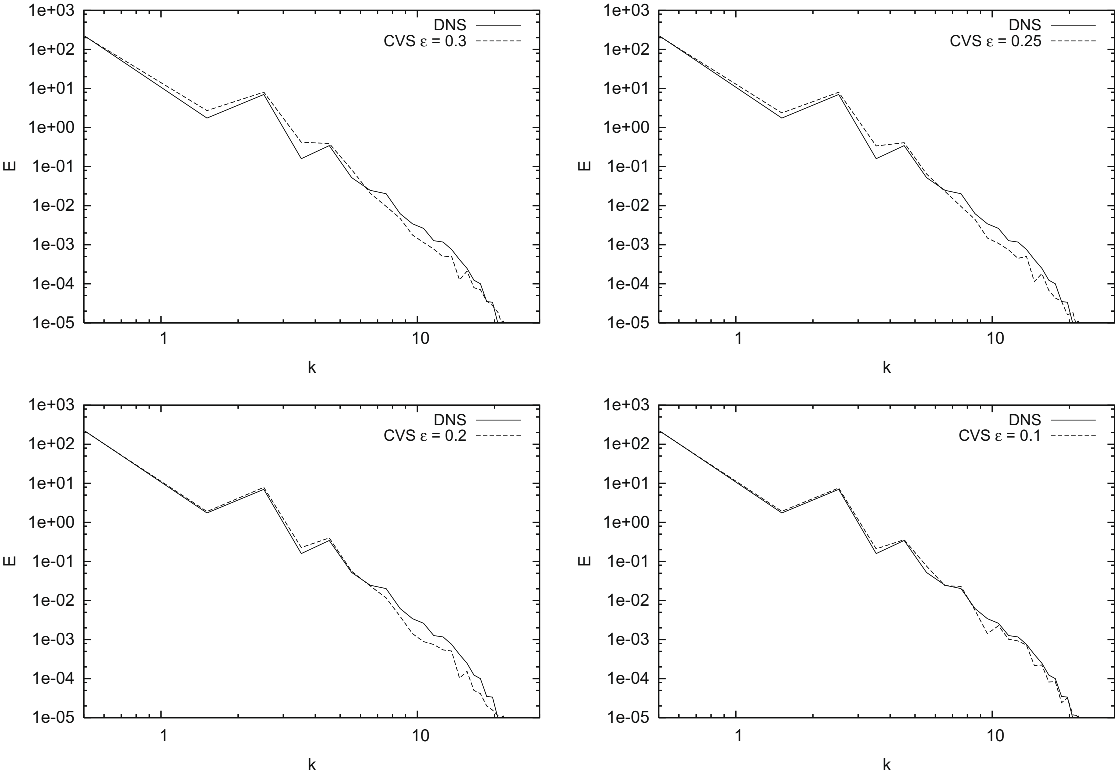

Fig. 5. Results obtained with the Norm #1. Energy spectra in the streamwise direction at $t = 80$ for $\epsilon = 0.3$ (*top left*), $\epsilon = 0.25$ (*top right*), $\epsilon = 0.2$ (*bottom left*), and $\epsilon = 0.1$ (*bottom right*).

Table 2
Results obtained with the Norm #2. Percentages of CPU time (required on a Pentium IV, 2.5 GHz), memory compression, and errors in comparison with the DNS solution for the kinetic energy E and the enstrophy Z at $t = 80$ with different ϵ and for $N = 128^3$.

Method	ε	CPU time	% CPU	% *Mem*	% *E*	% *Z*
DNS		7 d 6 h	100	100	100	100
MR	0.5	7 h 57 min	4.58	5.44	97.84	89.35
MR	0.08	2 d 18 h	37.93	30.55	99.96	88.17
MR	0.07	3 d 5 h	44.25	32.73	99.90	92.46

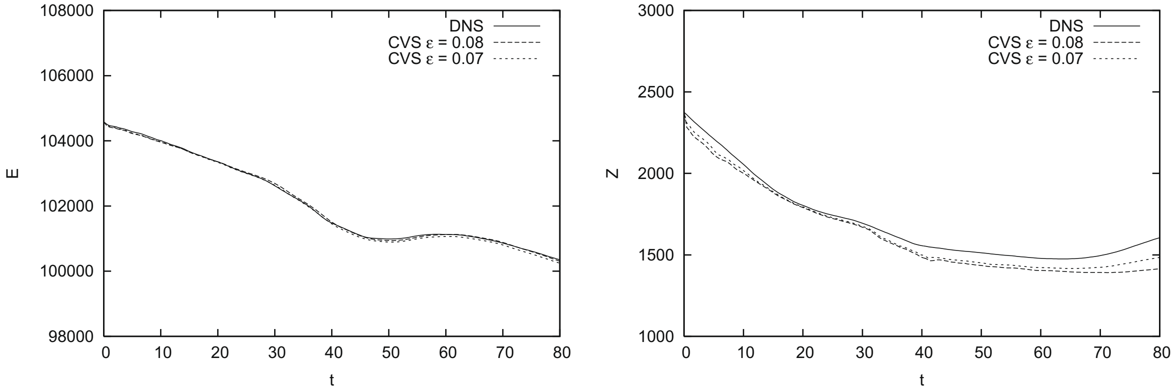

Fig. 6. Results obtained with the Norm #2. Kinetic energy (*left*), and enstrophy (*right*) of the DNS and several tolerance values for the CVS computations.

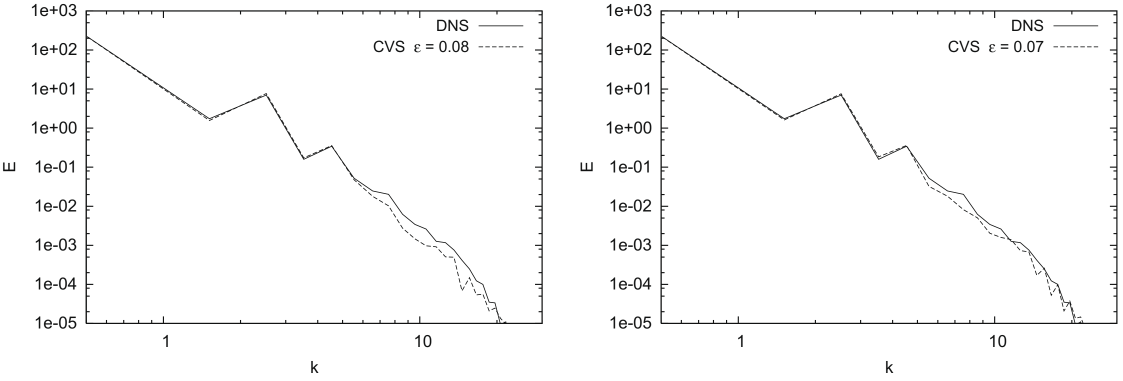

Fig. 7. Results obtained with the Norm #2. Energy spectra in the streamwise direction at $t = 80$ for $\epsilon = 0.08$ (*left*) and $\epsilon = 0.07$ (*right*).

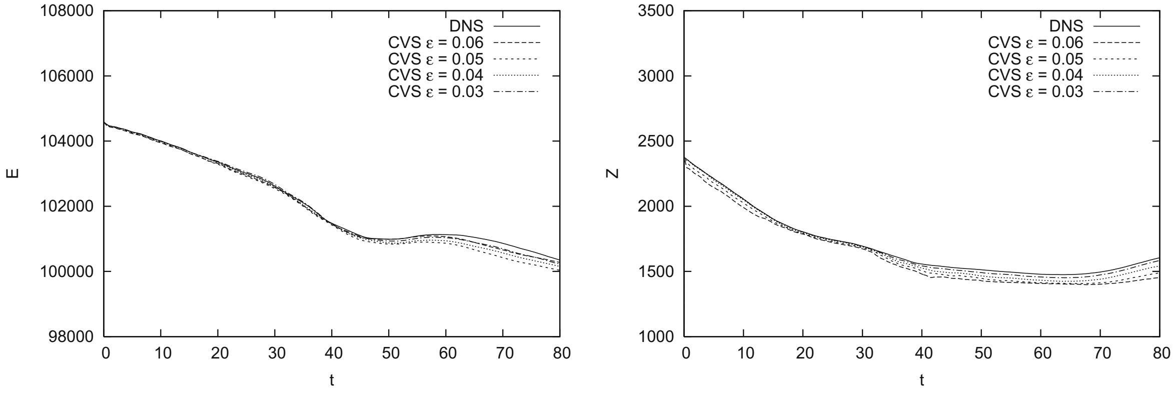

Fig. 8. Results obtained with the Norm #3. Time evolution of the kinetic energy (*left*) and the enstrophy (*right*) for the DNS and CVS computations using different tolerance values.

ues and for $\epsilon = 0.03$ we find the optimal agreement for the largest compression rate. These observations are confirmed in Table 3.

Fig. 9 shows the final streamwise energy spectra for DNS and CVS computations with different tolerance values. Again the agreement between CVS and DNS improves for decreasing tolerances. The CVS computation with $\epsilon = 0.03$ seems to be the optimal choice, since both DNS and CVS energy spectra almost coincide everywhere, whatever the wavenumber.

Table 3 shows the efficiency and accuracy of the CVS computations with the Norm #3. We observe that only 34.54% of the memory requirements of the DNS are used to reproduce about 99.88% of the energy and 98.66% of the enstrophy of the DNS

Table 3
Results obtained with the Norm #3. Percentages of CPU time (required on a Pentium IV, 2.5 GHz), memory compression, and errors in comparison with the DNS solution for the kinetic energy E and enstrophy Z at $t = 80$ with different ϵ, $N = 128^3$.

Method	ε	CPU time	% CPU	% *Mem*	% *E*	% *Z*
DNS		7 d 6 h	100	100	100	100
MR	0.06	2 d 0 h	27.59	24.23	99.93	90.60
MR	0.05	2 d 2 h	28.74	27.46	99.69	92.81
MR	0.04	2 d 10 h	29.89	30.80	99.80	96.12
MR	0.03	2 d 8 h	32.18	34.54	99.88	98.66
MR	0.01	4 d 6 h	58.62	48.20	99.95	99.91

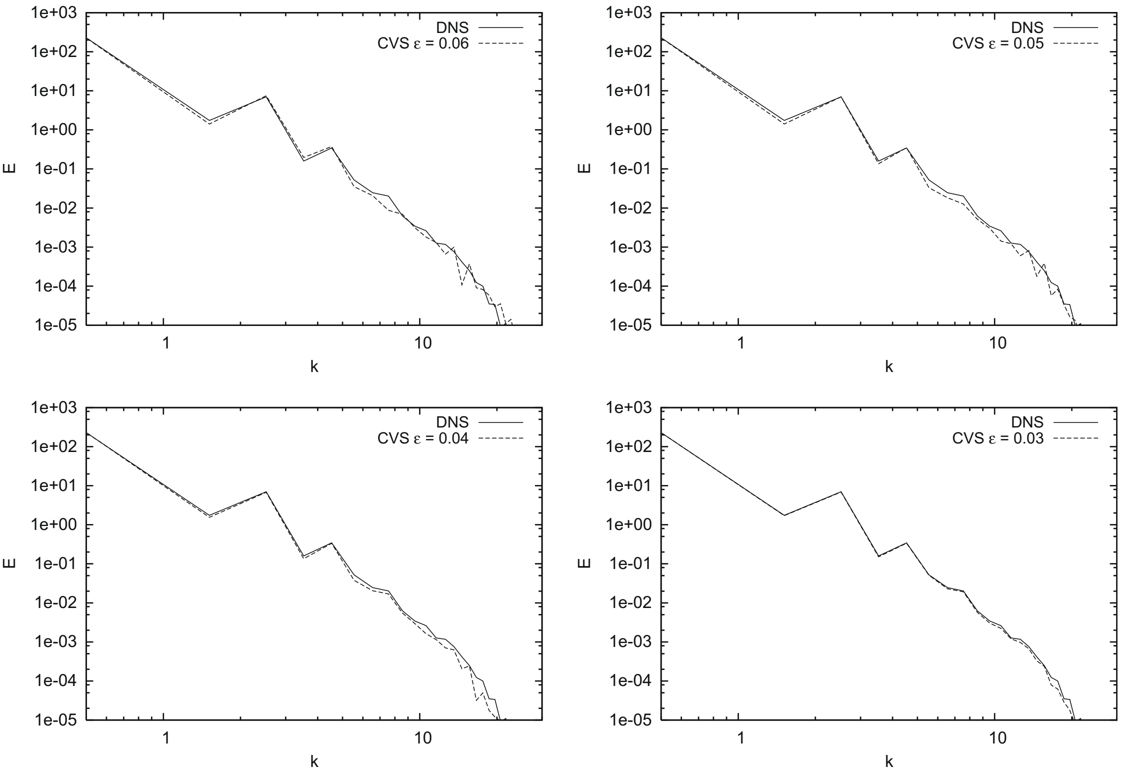

Fig. 9. Results obtained with the Norm #3. Energy spectra in the streamwise direction at $t = 80$ for different tolerance values: $\varepsilon = 0.06$ (*top left*), $\varepsilon = 0.05$ (*top right*), $\varepsilon = 0.04$ (*bottom left*), $\varepsilon = 0.03$ (*bottom right*).

computation. For this computation, less than one third of the CPU time of the DNS is required, which means that the CVS is more than three times faster than the DNS.

The vorticity fields in Fig. 3 confirm that we have a very good agreement between DNS and CVS computation with Norm #3 and $\epsilon = 0.03$. The last row of the figure presents the adaptive mesh at three different times and illustrates that the mesh dynamically adapts to the flow evolution and concentrates points in the shear layer, without any heuristical refinement criterion.

Concerning the DNS computation, it is shown that all the scales of the turbulent flow are well resolved. In fact, the energy 10^7 times lower than the largest energy is reached at the wavenumber $k \approx 20$. This value is much lower than the largest admissible wavenumber $k = 64$, when using a resolution 128^3.

4.3.4. Influence of the incoherent part of the CVS computation (Norm #3)

In this part, we compute the energy spectrum of the discarded part of the CVS computation at the final time. We take the difference between the DNS and CVS computations at $t = 80$. We compare this result with the DNS data where we applied the coherent vortex extraction (CVE) of the final solution using the same wavelet filtering (Fig. 10).

We observe a good agreement between the filtered DNS data (Fig. 10, right) and the CVS computation (Fig. 10, left). The incoherent contributions are strongly reduced at all scales and they exhibit a k^{-2} decay behavior which corresponds to an equipartition of enstrophy, i.e. the incoherent vorticity is decorrelated in physical space. The probability distribution func-

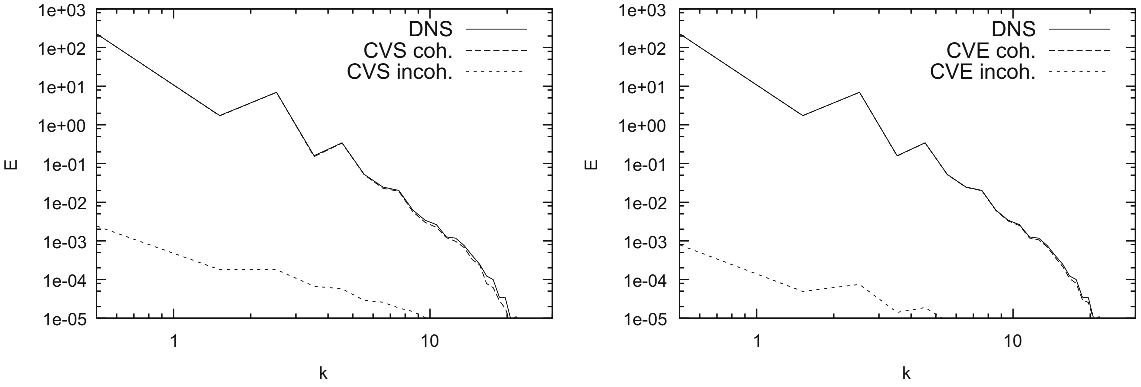

Fig. 10. Results obtained with the Norm #3. Energy spectra in the streamwise direction at $t = 80$ for a tolerance $\varepsilon = 0.03$, $Re = 100$. Left side: incoherent part obtained from the difference between the DNS and CVS computations. Right side: incoherent part obtained from the difference between the DNS and the CVE of the final DNS solution.

tions of the total, coherent and incoherent velocities, the latter obtained either from the velocity difference between the DNS and CVS computations or the velocity difference between the DNS and the CVS applied to the DNS solution, are shown in Fig. 11. The results confirm the strongly reduced variance of the incoherent contributions with respect to both, the total and coherent velocities and show that the high order statistics of the CVS computation are in good agreement with the DNS. Furthermore we find that the incoherent velocity exhibits a Gaussian-like distribution. Thus neglecting incoherent contributions during the flow evolution could justify the modeling of turbulent dissipation.

4.4. *Influence of the Reynolds number*

In this section, we study the influence of the Reynolds number. In addition to the previous case with $Re = 100$, we perform computations for $Re = 50$ and 200. For every CVS computation, we use the Norm #3 and the optimal tolerance $\epsilon_{opt} = 0.03$.

4.4.1. *Results with $Re = 50$*

The plots in Fig. 12 show the more dissipative behaviour of the flow compared to the case with $Re = 100$ (see Fig. 3). The two first rows of Fig. 12 are two-dimensional isolines of vorticity in the plane $y = 0$ for both DNS and CVS computations. They confirm that the solutions agree as well as in the case $Re = 100$. At the bottom of Fig. 12, the evolution of the adaptive mesh for the CVS computation is plotted, which confirms the efficient self-adaptive behaviour of the multiresolution algorithm.

Table 4 gives the performance of the CVS computation in comparison with the DNS one. The percentages are very similar to the ones otained for $Re = 100$ using the optimal tolerance (see Table 3), which shows the robustness of the choice of the

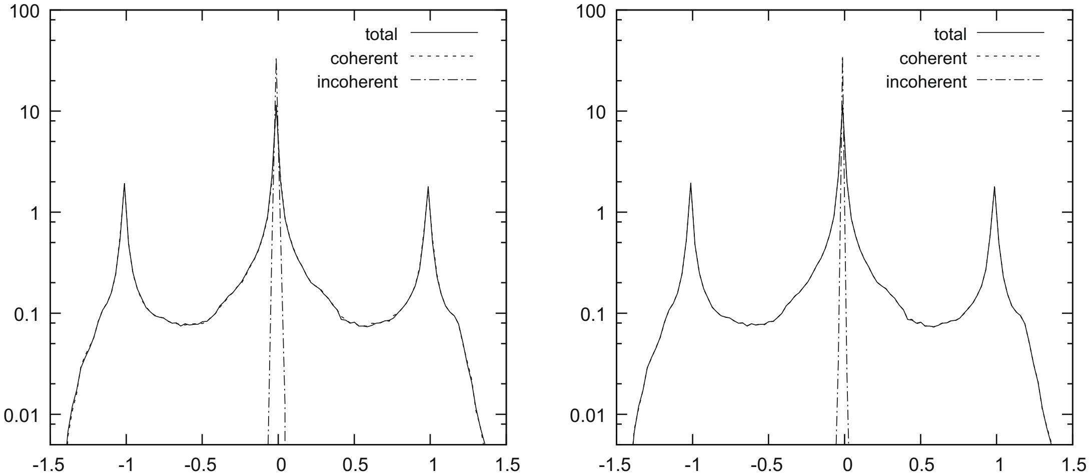

Fig. 11. Results obtained with the Norm #3. Probability distribution functions of the velocity at $t = 80$ for a tolerance $\varepsilon = 0.03$, $Re = 100$. Left side: total, coherent and incoherent part, the latter obtained from the difference between the DNS and CVS computations. Right side: total, coherent and incoherent part, the latter obtained from the difference between the DNS and the CVE of the final DNS solution.

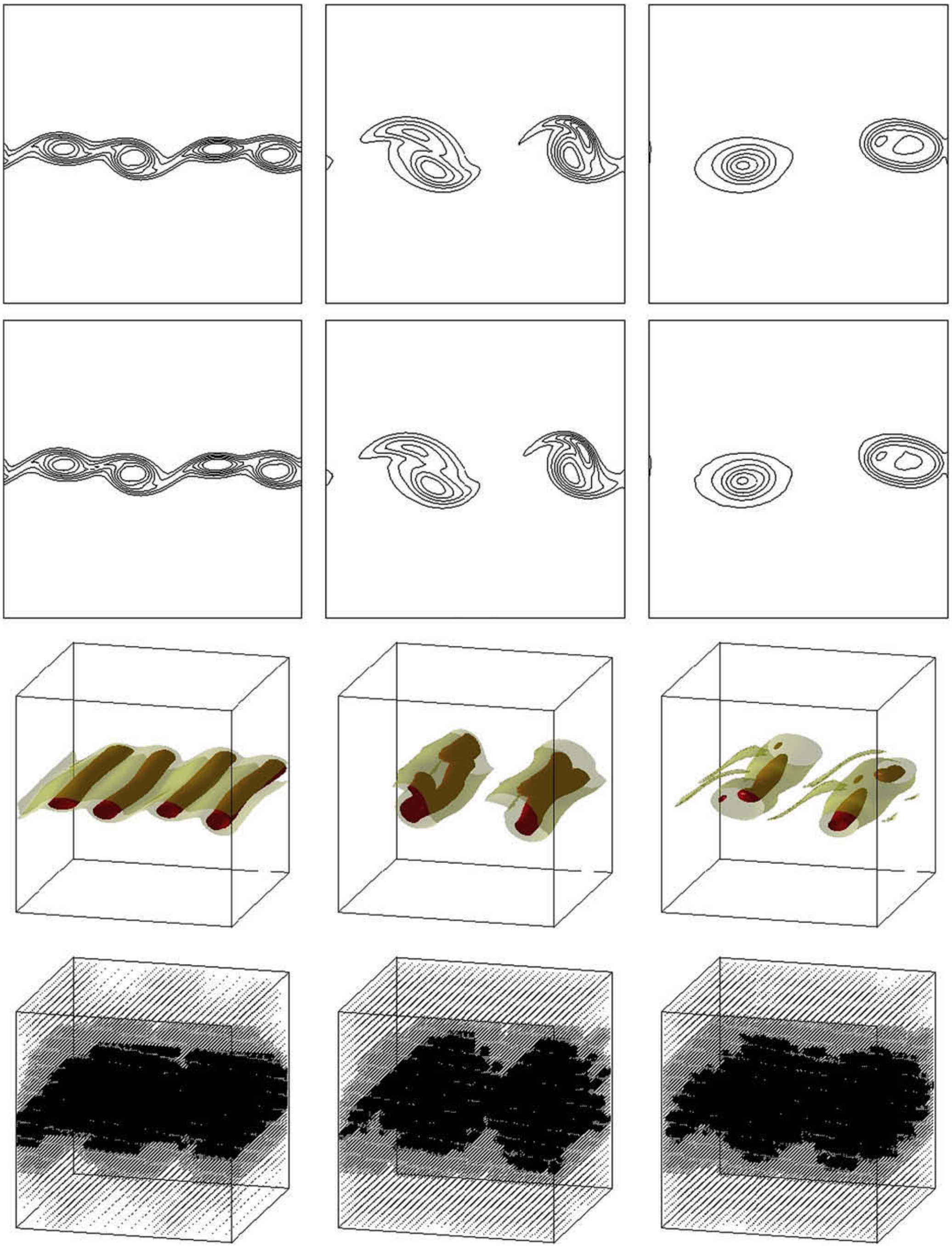

Fig. 12. Time evolution of a weakly compressible mixing layer for $Re = 50$. First row: isolines of vorticity in the plane $y = 0$ for the DNS solution. Second row: same isolines for the CVS computation with $\epsilon = 0.03$ and Norm #3. Third row: Corresponding isosurfaces of vorticity of the CVS $\|\omega\| = 0.5$ (black) and $\|\omega\| = 0.25$ (gray). Fourth row: Adaptive mesh of the CVS computation. The time instants are $t = 19$ (*left*), $t = 37$ (*center*) and $t = 78$ (*right*).

Table 4
Percentages of CPU time, memory compression, and errors in comparison with the reference DNS solution for the kinetic energy E and the enstrophy Z at $t = 80, N^3 = 128^3, Re = 50$.

Method	CPU time	% CPU	% *Mem*	% *E*	% *Z*
DNS	7 d	100	100	100	100
CVS	2 d 9 h	33.93	33.73	99.82	98.60

tolerance. The energy spectra obtained with CVS agrees well for every scale with the reference DNS (Fig. 13, left side). The time evolution of the kinetic energy and the enstrophy (Fig. 13, center and right side) shows likewise a good agreement between CVS and DNS.

4.4.2. $Re = 200$

Compared with the case $Re = 100$ (Fig. 3), we observe that the structures are less dissipated for $Re = 200$ and that three-dimensional effects are, as expected, more intense (Fig. 14). The isolines of the two-dimensional vorticity in the plane $y = 0$ (Fig. 14) look identical for CVS and DNS computations, except a slight difference for the left vortex at $t = 78$.

The results given in Table 5 for the percentages of CPU time, memory, energy and enstrophy are also very close to the ones computed with $Re = 100$ (see Table 3) using the same tolerance $\epsilon = 0.03$.

The final energy spectra in the streamwise direction obtained by CVS and DNS methods are shown in Fig. 15. They are in good accordance for almost every wavenumber. However, in the smallest scales ($k > 10$), the energy spectra loses accuracy. Despite this, the CVS computation conserves 99.82% of the energy and 98.24% of the enstrophy of the DNS, which satisfies our expectations.

The time evolution of energy and enstrophy (Fig. 15, center and right side) confirm the good agreement between the CVS and DNS computations, keeping in mind that CVS requires around one third of the CPU time and the memory required by the DNS (Table 5).

4.4.3. Comparison of energy spectra for different Reynolds numbers

In order to assess the influence of the Reynolds number on the streamwise energy spectra, we compare computations for three Reynolds numbers (Fig. 16).

In every case we find a good agreement between the CVS and DNS computations for the whole range of wavenumbers. For increasing Reynolds, we also see that the range of active wavenumbers increases and that the CVS is able to reproduce well the small scale behaviour.

4.4.4. Dependance of the relative error on final energy and enstrophy with the Reynolds number

Additional computations for $Re = 75,\ Re = 125,\ Re = 150$ and $Re = 175$ have been performed, to check if the relative error of energy and enstrophy for CVS compared to DNS shows a Reynolds number dependence or remains constant. The energy and enstrophy values at $t = 80$ vs. the Reynolds number are plotted in Fig. 17, for both DNS and CVS computations. First, we observe that both energy and enstrophy increase with Re and that the gap between CVS and DNS computations slightly increases.

However, we find that, for the kinetic energy, the relative error yields $e_E = 1.797 \times 10^{-3}$ for $Re = 50$, whereas, for $Re = 200$, we have $e_E = 1.722 \times 10^{-3}$. This means that the relative error remains almost constant. For the enstrophy, similar observations can be made, i.e. for $Re = 50$ the relative error of enstrophy is $e_Z = 1.397 \times 10^{-2}$ and for $Re = 200$ we have $e_Z = 1.764 \times 10^{-2}$. This shows that the relative error on enstrophy slightly increases, but is still of the same order of magnitude. We conclude that the relative errors on energy and enstrophy remain stable for the investigated range of Reynolds numbers.

4.5. Time evolution of a mixing layer computed at resolution 256^3

Finally we present a CVS computation with norm #3 and $\epsilon = 0.03$ at resolution $N = 256^3$. The domain is now set to $[-60, 60]^3$, so that the smallest space step is the same as in the case 128^3. As before, the Reynolds number is set to $Re = 200$, and the final time corresponds to $t = 80$.

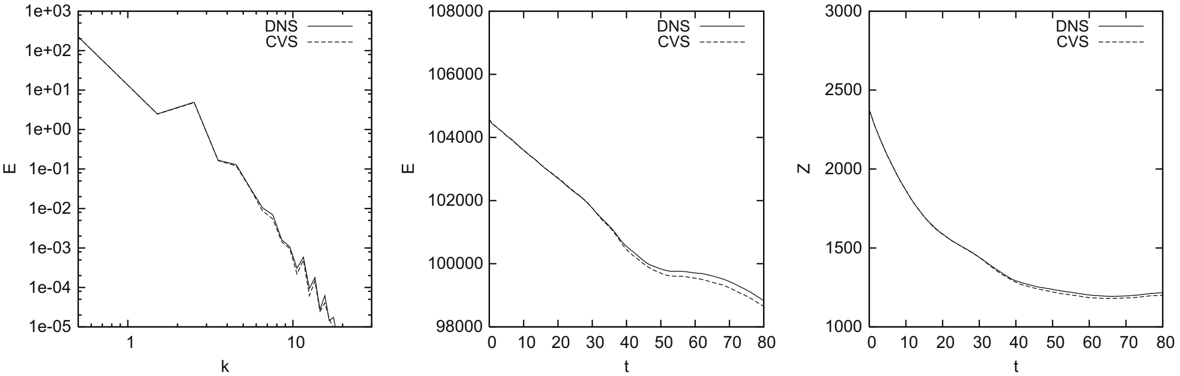

Fig. 13. Energy spectra in the streamwise direction at $t = 80$ (*left*), time evolution of the kinetic energy (*center*) and the enstrophy (*right*) for both DNS and CVS computations, $N^3 = 128^3, Re = 50$.

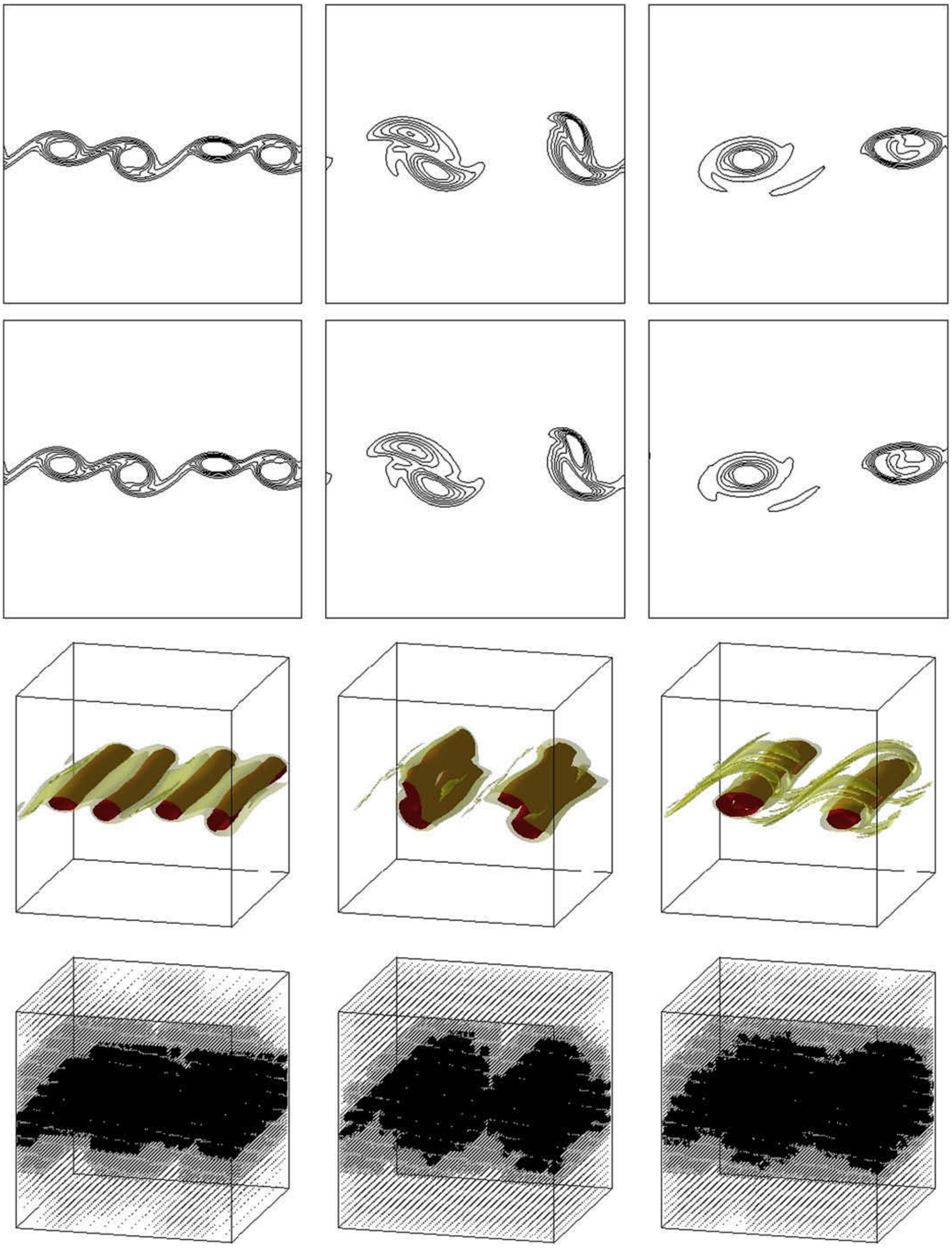

Fig. 14. Time evolution of a weakly compressible mixing layer for $Re = 200$. First row: Isolines of vorticity in the plane $y = 0$ for the reference DNS solution. Second row: same isolines for the CVS computation with $\epsilon = 0.03$ and norm #3. Third row: Corresponding isosurfaces of vorticity $\|\omega\| = 0.5$ (black) and $\|\omega\| = 0.25$ (gray) for the CVS computation. Fourth row: Adaptive mesh for the CVS computation. The time instants are t = 19 (*left*), *t* = 37 (*center*) and *t* = 78 (*right*).

Table 5

Percentages of CPU time, memory compression, and errors for *E* and *Z*, in comparison with the DNS solution, $N^3 = 128^3$, $Re = 200$.

Method	CPU time	*% CPU*	*% Mem*	*% E*	*% Z*
DNS	8 d 14 h	100	100	100	100
CVS	2 d 13 h	29.81	34.93	99.83	98.24

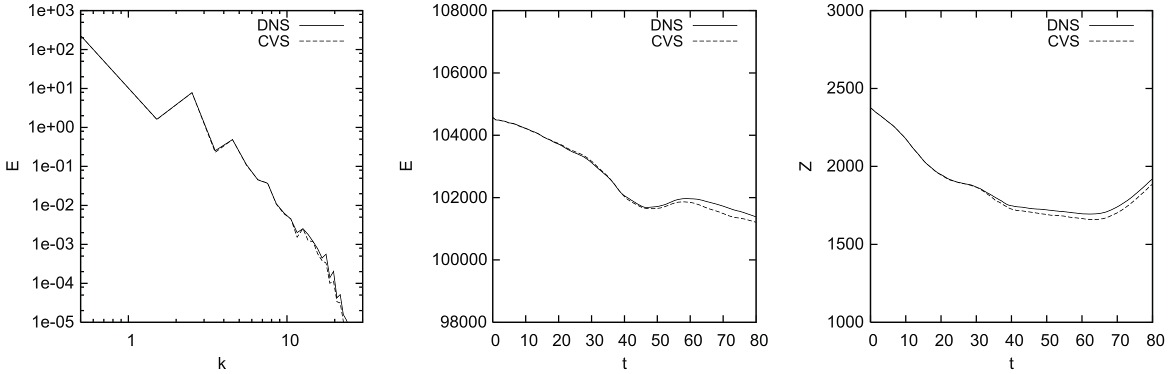

Fig. 15. Energy spectra in the streamwise direction at $t = 80$ (*left*). Time evolution of the kinetic energy (*center*) and enstrophy (*right*) for the CVS and DNS computations, $N^3 = 128^3, Re = 200$.

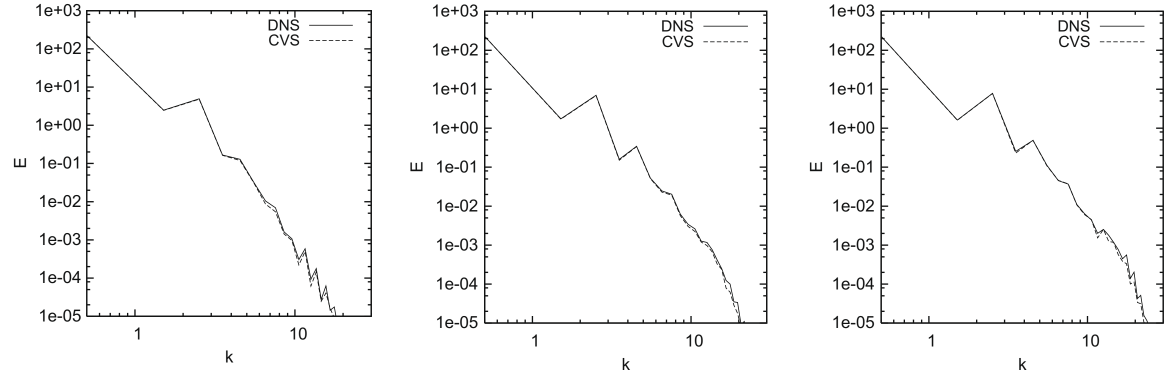

Fig. 16. DNS and CVS energy spectra at $t = 80$ for $Re = 50$ (*left*), $Re = 100$ (*center*) and $Re = 200$ (*right*).

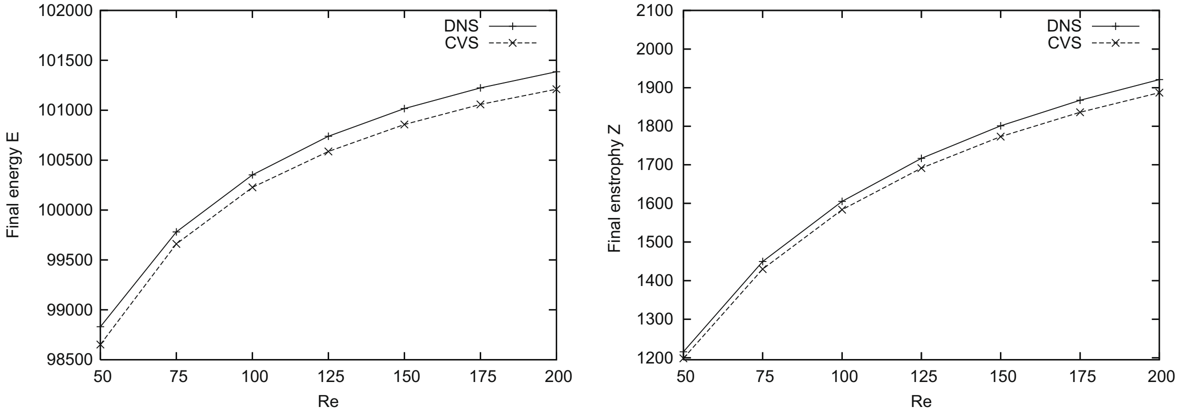

Fig. 17. Final energy E (left) and enstrophy Z (right) vs. the Reynolds number for the DNS and CVS computations.

Since the method has previously been validated for the same smallest space step, and since the DNS computation would require more than two months, we only perform the CVS computation here. The goal of this test-case is not to validate the method, but to show that larger CPU time and memory compressions can be reached with this method when using a larger maximal resolution. To estimate the CPU time that such a DNS computation would require, we performed the DNS computation on a few iterations only. We found that the DNS computation would approximately require 71 days of CPU time on a single PC.

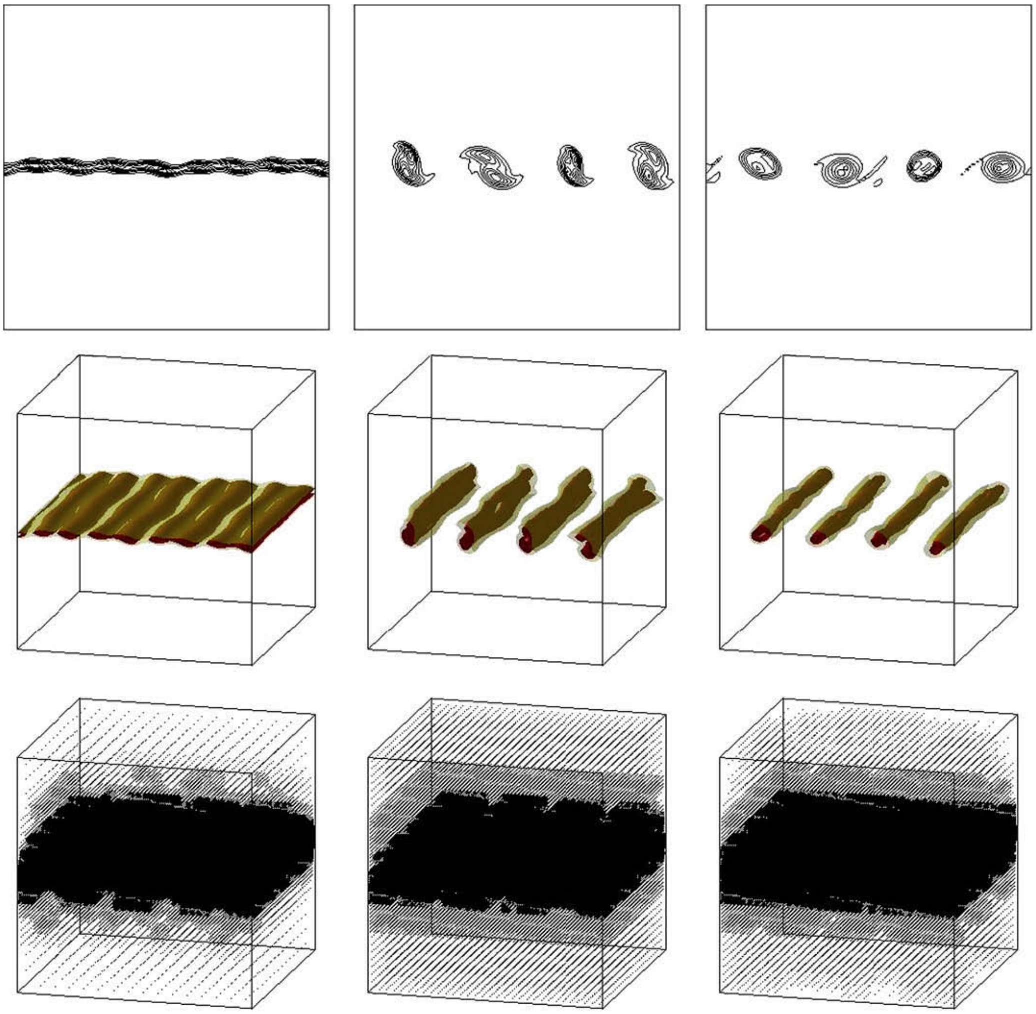

Fig. 18. Time evolution of a weakly compressible mixing layer at resolution $N = 256^3$. CVS computation with $\epsilon = 0.03$ and norm #3. First row: Two-dimensional isolines of vorticity in the plane $y = 0$, 10 isolines between 0.1 and 1. Second row: Corresponding isosurfaces of vorticity $\|\omega\| = 0.5$ (*black*) and $\|\omega\| = 0.25$ (*gray*). Third Row: Adaptive mesh of the CVS computation. The corresponding time instants are $t = 19$ (*left*), $t = 37$ (*center*) and $t = 78$ (*right*).

Table 6
Percentages of CPU time, and CPU and memory compression for the CVS computation. Results for $Re = 200$ and $N^3 = 256^3$.

Method	CPU time	% CPU	% Mem
CVS	16 d 17 h	23.40	17.06

The time evolution of the mixing layer is shown in Fig. 18. As before, we observe the efficient self-adaptive behavior of the algorithm, which exhibits more points in the shear zone, and less points far from this zone. The performance of the CVS computation is shown in Table 6. As expected, the performance increases with the number of scales, and the CVS computation is here more than four times faster than the DNS one (see Fig. 19).

5. Conclusion and perspectives

In the present paper, an adaptive multiresolution method is presented to solve the three-dimensional compressible Navier–Stokes equations in a Cartesian geometry. The Coherent Vortex Simulation approach previously developed for incompressible turbulence has been extended to compressible flows. It was shown that the time evolution of the coherent flow contributions can be computed efficiently using the adaptive multiresolution method. As generic test-case for free shear flows, we considered weakly compressible turbulent mixing layers at low Reynolds numbers ($50 < Re < 200$). Different

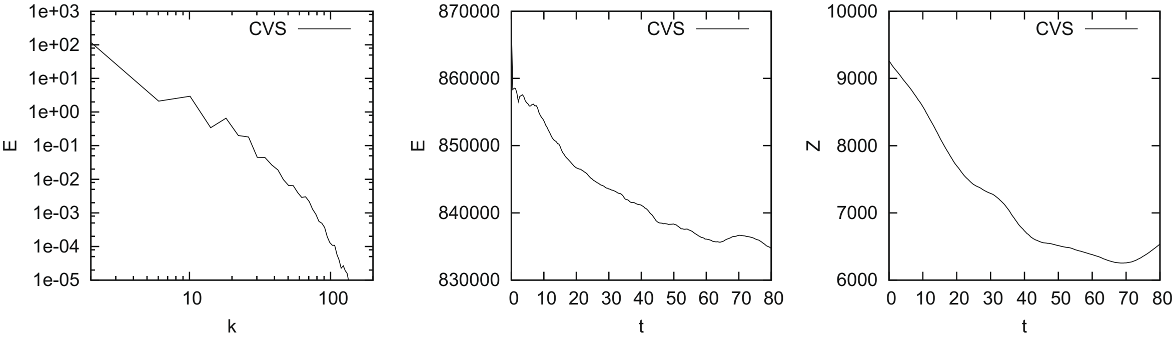

Fig. 19. Energy spectra in the streamwise direction at $t = 80$ (*left*). Time evolution of the kinetic energy (*center*) and enstrophy (*right*) for the CVS computations at $Re = 200$, $N = 256^3$.

thresholding rules based on different norms, i.e. L^1, L^2 and H^1 norms, have been investigated. We found that the H^1 based threshold yield the best results in terms of accuracy and efficiency. Concerning the computational efficiency, we showed that the CVS computation is approximatively three time faster than the DNS for a maximal resolution of 128^3, and more than four times faster for a maximal resolution 256^3. The CVS computation also requires less than one third of the memory that the DNS requires for a maximal resolution of 128^3. In the case 256^3, this factor is reduced to almost one sixth. Additionally, all the dynamically active scales of the flow are well resolved. The above CVS results are of course problem dependent, i.e. on the considered flow and its dimensionless parameters. This work can be seen as a first feasability study of CVS for weakly compressible shear flows and further investigations for higher Reynolds and Mach numbers and other flow types are necessary and will be reported in future work. Applications of the adaptive multiresolution method to supersonic flows considering the compressible Euler equations can be found in [6], which illustrates the potential of the adaptive multiresolution method for higher Mach numbers.

To improve further the compression of the CVS computations the threshold value ϵ has to be increased and thus more wavelet coefficients are discarded. However in this case the corresponding incoherent flow contributions cannot be simply neglected and some subgrid scale modeling is required to account for turbulent dissipation, similar to LES models. This approach called SCALES has been introduced in [10] and could be one possible direction for future work.

From numerical point of view one drawback of the current approach is the explicit time discretization, which imposes a time step limitation due to stability reasons, i.e. the smallest spatial scale dictates the actual size of the time step. A possible solution to circumvent this obstacle is the use of local time stepping strategies proposed in [5,18]. Using local time stepping, one is able to increase the time step on larger scales without violating the stability criterion of the explicit time integration and hence to further speed up the computation.

The restriction to Cartesian geometries can be surmounted using a volume penalization approach, introduced originally by Angot et al. [1] for incompressible flows and successfully applied with adaptive wavelet methods in [23]. An extension of the volume penalization to compressible flows has been published recently in [16] where, in addition to the permeability parameter, an impedance parameter is introduced to get rid of the sound waves inside the obstacle or solid wall regions. Current work is going in this direction to develop efficient CVS methods for compressible flows in complex geometries.

Acknowledgments

The authors acknowledge financial support from the CNRS–DFG program ‘‘LES of complex flows’’. KS thanks the Agence Nationale pour la Recherche for financial support under the grant ‘‘M2TFP’’. We would also like to thank Igor Molina for his help in computations, and Siegfried Müller for fruitful discussions.

References

[1] P. Angot, C.-H. Bruneau, P. Fabrie, A penalisation method to take into account obstacles in viscous flows, Numer. Math. 81 (1999) 497–520.
[2] A. Cohen, Wavelet methods in numerical analysis, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, vol. 7, Elsevier, Amsterdam, 2000.
[3] A. Cohen, I. Daubechies, J.C. Feauveau, Biorthogonal bases of compactly supported wavelets, Commun. Pure Appl. Math. 45 (1992) 485–560.
[4] A. Cohen, S.M. Kaber, S. Müller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, Math. Comp. 72 (2003) 183–225.
[5] M. Domingues, S. Gomes, O. Roussel, K. Schneider, An adaptive multiresolution scheme with local time – stepping for evolutionary PDEs, J. Comput. Phys. 227 (2008) 3758–3780.
[6] M. Domingues, S. Gomes, O. Roussel, K. Schneider, Space-time adaptive multiresolution methods for hyperbolic conservation laws: applications to compressible Euler equations, Appl. Numer. Math. 59 (2009) 2303–2321.
[7] M. Farge, K. Schneider, Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets, Flow, Turbulence Combust. 66 (4) (2001) 393–426.
[8] M. Farge, K. Schneider, N. Kevlahan, Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthonormal wavelet basis, Phys. Fluids 11 (8) (1999) 2187.
[9] A. Favre, Equations des gaz turbulents compressibles, J. Mécanique 4 (1965) 361–421.

[10] D. Goldstein, O. Vasilyev, Stochastic coherent adaptive large eddy simulation method, Phys. Fluids 16 (2004) 2497–2513.
[11] D. Gottlieb, E. Turkel, Dissipative two-four methods for time-dependent problems, J. Comput. Phys. 30 (1976) 703–723.
[12] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, Commun. Pure Appl. Math. 48 (1995) 1305–1342.
[13] A. Harten, Multiresolution representation of data: a general framework, SIAM J. Numer. Anal. 33 (3) (1996) 1205–1256.
[14] C. Hirsch, Numerical Computation of Internal and External Flows, vol. 2, John Wiley & Sons, 1990.
[15] M. Lesieur, O. Metais, P. Comte, LES of Turbulence, Cambridge University Press, 2005.
[16] Q. Liu, O. Vasilyev, Brinkman penalization method for compressible flows in complex geometries, J. Comput. Phys. 227 (2) (2007) 946–966.
[17] S. Müller, Adaptive multiscale schemes for conservation laws, Lecture Notes in Computational Science and Engineering, vol. 27, Springer-Verlag, Heidelberg, 2003.
[18] S. Müller, Y. Stiriba, Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, J. Sci. Comput. 30 (2007) 493–531.
[19] N. Okamoto, K. Yoshimatsu, K. Schneider, M. Farge, Y. Kaneda, Coherent vortices in high resolution DNS of homogeneous isotropic turbulence: a wavelet viewpoint, Phys. Fluids 19 (115109) (2007).
[20] O. Roussel, K. Schneider, M. Farge, Coherent vortex extraction in 3d homogeneous turbulence: comparison between orthogonal and biorthogonal wavelet decompositions, J. Turbulence 6 (11) (2005) 1–15.
[21] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, J. Comput. Phys. 188 (2) (2003) 493–523.
[22] P. Sagaut, S. Deck, M. Terracol, Multiscale and Multiresolution Approaches in Turbulence, Imperial College Press, 2006.
[23] K. Schneider, M. Farge, Adaptive wavelet simulation of a flow around an impulsively started cylinder using penalisation, Appl. Comput. Harm. Anal. 12 (2002) 374–380.
[24] K. Schneider, M. Farge, G. Pellegrino, M. Rogers, Coherent vortex simulation of 3D turbulent mixing layers using orthogonal wavelets, J. Fluid Mech. 534 (2005) 39–66.
[25] K. Schneider, O. Vasilyev, Wavelet methods in computational fluid dynamics, Ann. Rev. Fluid Mech. 42 (2010) 473–503.
[26] B. Vreman, Direct and large-Eddy simulation of the compressible turbulent mixing layer, Ph.D Thesis, University of Twente, Netherlands, 1995.
[27] B. Vreman, B. Geurts, H. Kuerten, Large-eddy simulation of the turbulent mixing layer, J. Fluid Mech. 339 (1997) 357–390.