INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2016; **82**:261–285
Published online 5 February 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.4217

A local mesh refinement approach for large-eddy simulations of turbulent flows

M. Cevheri *,† R. McSherry and T. Stoesser

School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK

SUMMARY

In this paper, a local mesh refinement (LMR) scheme on Cartesian grids for large-eddy simulations is presented. The approach improves the calculation of ghost cell pressures and velocities and combines LMR with high-order interpolation schemes at the LMR interface and throughout the rest of the computational domain to ensure smooth and accurate transition of variables between grids of different resolution. The approach is validated for turbulent channel flow and flow over a matrix of wall-mounted cubes for which reliable numerical and experimental data are available. Comparisons of predicted first-order and second-order turbulence statistics with the validation data demonstrated a convincing agreement. Importantly, it is shown that mean streamwise velocities and fluctuating turbulence quantities transition smoothly across coarse-to-fine and fine-to-coarse interfaces. © 2016 The Authors International Journal for Numerical Methods in Fluids Published by John Wiley & Sons Ltd

Received 27 May 2015; Revised 28 October 2015; Accepted 19 December 2015

KEY WORDS: local mesh refinement; large-eddy simulation; incompressible flow; turbulent channel flow; LMR; LES

1. INTRODUCTION

In recent decades, considerable progress has been made in the development of high fidelity numerical methods for the solution of the Navier–Stokes equations to simulate turbulent flows of practical interest. Computational techniques such as large-eddy simulation (LES) and direct numerical simulation (DNS) are applied increasingly to such flows, but their use continues to be restricted by grid resolution requirements, and therefore also by available computing resources. In general, sophisticated numerical methods on sufficiently fine grids enable accurate solutions of these simulations to be obtained.

It is generally difficult to achieve the desired level of accuracy in turbulent flow simulations without the use of spectral methods, which are presently limited to simple geometries with periodic boundary conditions. High-order methods such as the compact scheme and high-order accurate finite difference discretisation [1, 2], which have been investigated principally for LES of turbulent flow [3], therefore offer more convenient, easier ways to handle the nontrivial geometries that are involved in most modern scientific and engineering applications. Inherently, high-order methods achieve better accuracy on coarser meshes compared with low-order schemes, which means lower computational cost and faster convergence with lower dissipation and dispersion errors. When applied to eddy-resolving simulations such as LES, however, high-order methods are nonetheless subject to the restriction that the grid must be fine enough to accurately resolve the flow features of interest. Furthermore,

*Correspondence to: Mehtap Cevheri, School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK.
†E-mail: mehtap4@gmail.com

The copyright line for this article was changed on 23 February 2016 after original online publication.

the accuracy of subgrid scale (SGS) modelling, which forms the basis of LES, is normally achieved using closure models that are a function of grid resolution and are generally less complex than the analogue of a Reynolds-averaged Navier–Stokes turbulence model.

Grid-based (as opposed to meshless) numerical solution of the governing equations of fluid flow can be achieved on structured or unstructured grids. Although the recent trend in LES has been for the application of finite volume [4] and finite element methods [5, 6] on unstructured grids, the present study focuses on the use of structured grids for two principal reasons: firstly, the accuracy of the computation on unstructured grids often suffers when extreme gradients are experienced along the dominant flow direction, as is the case in boundary layer flow, for example [7], and secondly, unstructured grids are often not suitable for use with multigrid methods, which have been used by the authors in the past to achieve very efficient computations. In addition, the immersed boundary method (IBM) [8, 9] permits the accurate and efficient representation of complex solid bodies on fixed Cartesian grids, thereby circumventing one of the main perceived limitations of structured grids, which is their inability to represent complex geometries.

For computation of flows on structured grids in which only certain regions of the physical domain are critical, two approaches have received significant recent attention: grid stretching, in which cells are stretched in one direction as distance from the critical region(s) increases, and local mesh refinement (LMR), in which sequences of nested finer grids are applied in the critical region(s). Both approaches aim at optimising the use of available computing power by concentrating grid resolution where it is most needed, and reducing it where it is not. Stretched grids inherently include cells with high aspect ratios in some areas of the flow (and particularly in complex domains) prolonging or hindering the convergence of the Poisson equation in projection methods, particularly when immersed boundaries are also used [10]. The present study therefore focuses on the application of LMR in structured grids.

The development of a mesh refinement algorithm for the solution of hyperbolic partial differential equations in one and two dimensions was initially discussed by [11, 12], who applied specially derived difference equations at the interface between nested grids of differing resolution. Since then, a number of researchers have investigated the application of mesh refinement to incompressible flows, but this is significantly complicated by the elliptic nature of the governing equations and the difficulty associated with the enforcement of the divergence-free constraint. The violation of mass and momentum conservation properties at the coarse-fine interfaces affects the global solution: several techniques such as idempotent [13] and non-idempotent (i.e. approximate) [14, 15] second-order projection methods using different convection schemes have therefore been developed.

Minion [14] developed an adaptive projection method for the incompressible Euler equations on locally refined grids to enforce the divergence-free constraint at each time step, establishing second-order accuracy in time and space. Almgren *et al.* [16] described a conservative adaptive projection method for the variable-density incompressible Navier–Stokes equations in two and three dimensions on adaptive grids, which refines in time as well. An algorithm for the incompressible Euler equations on block-structured local refinement grids using a multilevel projection algorithm was proposed by Martin and Colella [17]; the method has subsequently been extended to incompressible viscous flows by Martin *et al.* [18]. Popinet [19] proposed the adaptive quad/octree Gerris solver, which uses a combination of an approximate projection approach and the multigrid method to numerically solve the time-dependent incompressible Euler equations. A nested finite volume-based Cartesian grid was presented by Peng *et al.* [20] to simulate unsteady viscous incompressible flows with complex immersed boundaries. The equations resulting from discretisation on structured grids were solved using an iterative method known as line-successive-over-relaxation with a fast convergence rate.

Of critical importance in LMR is the establishment of procedures for communication of data across the non-matching interfaces to accommodate the requirements of discretisation stencils. This involves two processes: prolongation (i.e. extrapolation or interpolation) and restriction (i.e. averaging). Generally, linear interpolations to transfer data from coarse to fine grids are used for the prolongation process, and cell averaging procedures to reduce fine grid data onto coarse grids are used for the restriction process, and many fluid flow problems can be treated effectively using this technique. However, while the aforementioned studies [11–20] focused on the simulation of laminar flows, the application of nested grids for turbulent flows, and in particular for LES/DNS, requires different, more sophisticated strategies to minimise (or ideally remove) the negative by-products of grid discontinuities (in particular the inherent violation of conservativeness) on the accuracy and stability of the solution. Special interpolation algorithms, advanced numerical methods, higher-order discretisations and variable filters have been applied in LES computations to circumvent these issues [21–24].

Kravchenko *et al.* [21] computed wall-bounded turbulent flows using a pseudo-spectral/B-spline approach with a Fourier spectral method in homogeneous directions and a B-spline method in the wall-normal direction on multi-level grids, with the finest zones located near the walls. Although the results were in good agreement with previously published experimental results, the applicability of spectral methods to turbulent flows in complex geometries is limited. Piomelli *et al.* [22] studied LES of turbulent channel flow with different SGS models and variable filters to examine the combined effect of truncation, modelling and commutation errors on discontinuous grids. It was observed that interfaces parallel to the direction of mean advection did not result in significant discontinuities, but those normal to it were characterised by large jumps in the resolved shear stresses, especially close to the wall. Application of a variable filter across the interface reduced commutation errors but did not reduce the discontinuity, and the authors therefore postulated that commutation errors were not the main cause. Pantano *et al.* [23] presented a dynamically adaptive, three-dimensional hybrid finite difference method for LES of compressible flows. The method employed a finite difference weighted essentially non-oscillatory scheme close to the grid resolution discontinuities, while a tuned centred finite difference scheme was applied in regions of uniform resolution. Smooth flow and conserved properties were observed around the resolution discontinuities, and the turbulent structures passing through the interface were shown to be well resolved. Simulations of spatially developing homogeneous isotropic turbulent were performed by [24] on fine-to-coarse and coarse-to-fine grids, and the development of the turbulence downstream of the interface was investigated. It was observed that a significant amount of energy accumulation occurs when the grid is suddenly coarsened in the streamwise direction. Although filter widths that changed discontinuously across the interface gave more accurate results for coarse-to-fine grids than filter widths that varied gradually between the values of coarse and fine grids, commutation errors were shown to be larger for the discontinuous filter widths. Furthermore, gradually changing filter width across the interface gives better results for a fine-to-coarse interface. All of the observations in [24] were obtained for relatively simple turbulent flows, and their accuracy for complex flows remains unquantified.

The present study introduces and validates an LMR method for LESs of turbulent incompressible flows. The most important feature of the proposed approach derives from the combination of high-order spatial discretisation and LMR, which aims at achieving improvements in terms of the predictions of second-order statistics in complex turbulent flows. Two validation cases have been chosen to illustrate the robustness of the code: turbulent channel flow at friction Reynolds number $Re_\tau = 590$ and flow over a matrix of cubes at bulk Reynolds number $Re = 1.3 \times 10^4$. The paper is organised as follows: Section 2 introduces the numerical framework that has been developed; the computational setup of the turbulent channel and cube matrix validation simulations are presented in Section 3; and the results, which are compared with DNS and experimental data, are presented and discussed in Section 4. Finally, some conclusions are drawn in Section 5.

2. NUMERICAL FRAMEWORK

2.1. *Large-eddy simulation method*

In this study, the in-house LES code Hydro3D, which has been validated thoroughly for many internal and external flows (e.g. [25–30]), is modified with an LMR algorithm. In LES, large-scale turbulent motions are simulated directly, and only the less-energetic, dissipative part of the turbulence spectrum is modelled [31, 32]. Hydro3D solves the incompressible, filtered Navier–Stokes equations which read

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_i \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{2}$$

where x_i and x_j are the spatial location vectors (i.e. x_i and $x_j = x, y, z$ for i and $j = 1, 2, 3$), u_i, u_j ($i, j = 1, 2, 3$) are the resolved velocity components in $x-, y-$ and $z-$directions, normalised on the reference velocity U and p is the resolved pressure divided by density. $Re = UL/\nu$ is Reynolds number, where ν is the kinematic viscosity and L is the reference length scale.

The SGS stress, τ_{ij}, arises from unresolved small scale fluctuations and is approximated using the wall-adapting local eddy viscosity model [33]. Unlike the classic Smagorinsky model, the wall-adapting local eddy viscosity model does not require damping near solid walls. This makes it very well-suited to simulations in which solid boundaries are not sharply defined, such as those performed using the IBM, as is the case in the present study. In addition, the model is based on the strain and rotation rates of resolved velocities, and it therefore automatically calculates zero eddy viscosity in laminar shear flows. The eddy viscosity, ν_T, is calculated as follows:

$$\nu_T = (C_w \Delta)^2 \frac{\left(s_{ij}^d s_{ij}^d\right)^{\frac{3}{2}}}{(\overline{S_{ij}}\overline{S_{ij}})^{\frac{5}{2}} + \left(s_{ij}^d s_{ij}^d\right)^{\frac{5}{4}}} \tag{3}$$

where C_w is a constant with a value 0.46. The traceless symmetric part of the square of the velocity gradient tensor is computed in the following form:

$$s_{ij}^d = \frac{1}{2}\left(\overline{g_{ik}g_{kj}} + \overline{g_{jk}g_{ki}}\right) - \frac{1}{3}\delta_{ij}\overline{g_{kk}}^2 \quad \text{with} \quad \frac{\partial \bar{u}_i}{\partial x_{ij}} \tag{4}$$

Implicit coupling of pressure and velocity terms is achieved by the fractional step method, which comprises two steps [34]. In the first step, the intermediate velocities, $u_i{}^*$, are initially predicted without enforcing the incompressibility constraint using a two-stage explicit Runge–Kutta method as follows:

$$u_i{}^{*,n+\frac{1}{2}} = u_i{}^n + \frac{\Delta t}{2}\left[\mathbf{R}^n - \frac{\partial}{\partial x_i}p^{n-\frac{1}{2}}\right] \tag{5}$$

$$u_i{}^{*,n+1} = u_i{}^n + \Delta t\left[\mathbf{R}^{*,n+\frac{1}{2}} - \frac{\partial}{\partial x_i}p^{n-\frac{1}{2}}\right] \tag{6}$$

where

$$R = \left[-\frac{\partial (u_i u_j)}{\partial x_j} + \left(\frac{1}{Re} + \nu_T\right)\frac{\partial^2 u_i}{\partial x_i \partial x_j}\right]. \tag{7}$$

In the second step, the intermediate velocities are projected onto the divergence-free vector fields through a Poisson equation, which calculates an increment to the pressure field, δp:

$$\nabla^2(\delta p) = \frac{1}{\Delta t}\frac{\partial}{\partial x_i}u_i{}^{*,n+1} \tag{8}$$

where ∇^2 is the Laplacian operator. Equation (8) is solved iteratively using the multigrid method [[35]], and the velocity field is updated as follows:

$$u_i{}^{n+1} = u_i{}^{*,n+1} - \Delta t\frac{\partial(\delta p)}{\partial x_i} \tag{9}$$

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \delta p \tag{10}$$

The Poisson equation is then solved with this new pressure, and an iterative process is established. The iteration continues until the divergence-free condition of the computed velocity field within some tolerance is satisfied. It is observed that the overall algorithm is second-order accurate and stable on either uniform or locally refined grids when the ghost cell pressures at coarse-fine grid interfaces are treated with a specific interpolation algorithm, which is described in Section 2.2.

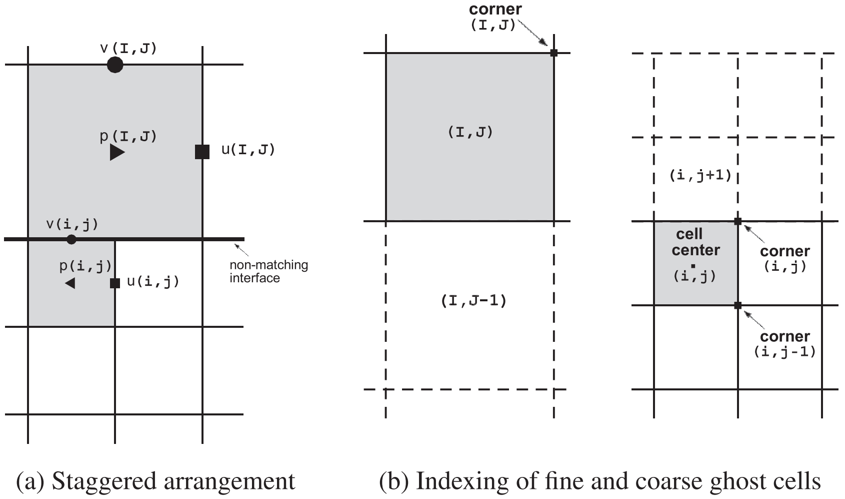
Hydro3D employs a Cartesian grid with a staggered arrangement of pressure and velocity variables, which has the advantage of avoiding the odd-even coupling between the pressure and velocity without requiring special interpolation algorithms. On a staggered grid, scalar quantities such as pressure and divergence are stored at the cell centres, and vector quantities such as velocity and pressure gradient are stored at the middle of the cell faces. Figure 1(a) presents a schema showing the storage of pressures and velocities on a 2D staggered grid, either side of an LMR interface: the indices (I, J) and (i, j) correspond to coarse and fine grid points, respectively. The indices of u velocity on the east face and v velocity on the north face of the shaded-fine cell are (i, j), and u velocity on the west face and v velocity on the south face of the same cell are $(i-1, j)$ and $(i, j-1)$, respectively. The indexing of coarse cells is handled analogously. One or more layers of ghost cells, denoted in Figure 1(b) by dashed lines, are appended around each side of the physical domain in order to provide numerical boundaries and to facilitate spatial approximations of cells at the fine-coarse interfaces without requiring complicated discretisation stencils of non-uniform grids.

Equation (7) contains partial derivatives in space of first and second order, and in LES, the accurate approximation of these is key to the success of the simulation [31]. In Hydro3D, explicit second-order and fourth-order central differences for continuity, convective and viscous terms are available and for the sake of simplicity are presented here in two dimensions.

Second-order accurate approximations of the cell-centre partial derivatives in the continuity equation read

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + O\left(h^2\right) \qquad \text{and} \qquad \left(\frac{\partial v}{\partial y}\right)_{i,j} = \frac{v_{i,j} - v_{i,j-1}}{\Delta y} + O\left(h^2\right) \tag{11}$$

Because of the staggered arrangement, the $x-$, $y-$ and $z-$momentum equations are solved at the cell faces. The second-order accurate discretisations of convective and diffusive terms in the $x-$momentum equation are as follows:



(a) Staggered arrangement (b) Indexing of fine and coarse ghost cells

Figure 1. The staggered variable arrangement and index location of coarse and fine ghost cells.

$$\left(\frac{\partial u^2}{\partial x}\right)_{i,j} = \frac{(u_{i+1,j}+u_{i,j})^2-(u_{i,j}+u_{i-1,j})^2}{4\Delta x} + O\left(h^2\right) \tag{12}$$

$$\left(\frac{\partial uv}{\partial y}\right)_{i,j} = \frac{(v_{i+1,j}+v_{i,j})(u_{i,j+1}+u_{i,j})-(v_{i+1,j-1}+v_{i,j-1})(u_{i,j}+u_{i,j-1})}{4\Delta y} + O\left(h^2\right) \tag{13}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} + O\left(h^2\right) \tag{14}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j} = \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\Delta y)^2} + O\left(h^2\right) \tag{15}$$

and the $y-$momentum convective and diffusive terms are evaluated as

$$\left(\frac{\partial uv}{\partial x}\right)_{i,j} = \frac{\left(u_{i,j+1}+u_{i,j}\right)\left(v_{i+1,j}+v_{i,j}\right)-\left(u_{i-1,j+1}+u_{i-1,j}\right)\left(v_{i,j}+v_{i-1,j}\right)}{4\Delta x} + O\left(h^2\right) \tag{16}$$

$$\left(\frac{\partial v^2}{\partial y}\right)_{i,j} = \frac{\left(v_{i,j+1}+v_{i,j}\right)^2-\left(v_{i,j}+v_{i,j-1}\right)^2}{4\Delta y} + O\left(h^2\right) \tag{17}$$

$$\left(\frac{\partial^2 v}{\partial x^2}\right)_{i,j} = \frac{v_{i+1,j}-2v_{i,j}+v_{i-1,j}}{(\Delta x)^2} + O\left(h^2\right) \tag{18}$$

$$\left(\frac{\partial^2 v}{\partial y^2}\right)_{i,j} = \frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{(\Delta y)^2} + O\left(h^2\right) \tag{19}$$

Analogous formulae are used to predict convective and diffusive terms for the $z-$momentum equation.

Fourth-order accurate approximations of the cell-centre partial derivatives in the continuity equation are [2]

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{-u_{i+1,j}+27u_{i,j}-27u_{i-1,j}+u_{i-2,j}}{24\Delta x} + O\left(h^4\right) \tag{20}$$

and

$$\left(\frac{\partial v}{\partial y}\right)_{i,j} = \frac{-v_{i,j+1}+27v_{i,j}-27v_{i,j-1}+v_{i,j-2}}{24\Delta y} + O\left(h^4\right) \tag{21}$$

Fourth-order accurate discretisations of convective and diffusive terms in the $x-$momentum equation are as follows:

$$\left(\frac{\partial u^2}{\partial x}\right)_{i,j} = \frac{-\widetilde{u_{i+2,j}}^2+27\widetilde{u_{i+1,j}}^2-27\widetilde{u_{i,j}}^2+\widetilde{u_{i-1,j}}^2}{24\Delta x} + O\left(h^4\right) \tag{22}$$

$$\left(\frac{\partial uv}{\partial y}\right)_{i,j} = \frac{-\widetilde{u^c_{i,j+1}}\widetilde{v^c_{i,j+1}} + 27\widetilde{u^c_{i,j}}\widetilde{v^c_{i,j}} - 27\widetilde{u^c_{i,j-1}}\widetilde{v^c_{i,j-1}} + \widetilde{u^c_{i,j-2}}\widetilde{v^c_{i,j-2}}}{24\Delta y} + O\left(h^4\right) \tag{23}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{-u_{i+2,j} + 16u_{i+1,j} - 30u_{i,j} + 16u_{i-1,j} - u_{i-2,j}}{12(\Delta x)^2} + O\left(h^4\right) \tag{24}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j} = \frac{-u_{i,j+2} + 16u_{i,j+1} - 30u_{i,j} + 16u_{i,j-1} - u_{i,j-2}}{12(\Delta y)^2} + O\left(h^4\right) \tag{25}$$

The $y-$momentum convective and diffusive terms, at the north face of the cell indexed by (i,j), are ultimately evaluated as

$$\left(\frac{\partial uv}{\partial x}\right)_{i,j} = \frac{-\widetilde{u^c_{i+1,j}}\widetilde{v^c_{i+1,j}} + 27\widetilde{u^c_{i,j}}\widetilde{v^c_{i,j}} - 27\widetilde{u^c_{i-1,j}}\widetilde{v^c_{i-1,j}} + \widetilde{u^c_{i-2,j}}\widetilde{v^c_{i-2,j}}}{24\Delta x} + O\left(h^4\right) \tag{26}$$

$$\left(\frac{\partial v^2}{\partial y}\right)_{i,j} = \frac{-\widetilde{v_{i,j+2}}^2 + 27\widetilde{v_{i,j+1}}^2 - 27\widetilde{v_{i,j}}^2 + \widetilde{v_{i,j-1}}^2}{24\Delta y} + O\left(h^4\right) \tag{27}$$

$$\left(\frac{\partial^2 v}{\partial x^2}\right)_{i,j} = \frac{-v_{i+2,j} + 16v_{i+1,j} - 30v_{i,j} + 16v_{i-1,j} - v_{i-2,j}}{12(\Delta x)^2} + O\left(h^4\right) \tag{28}$$

$$\left(\frac{\partial^2 v}{\partial y^2}\right)_{i,j} = \frac{-v_{i,j+2} + 16v_{i,j+1} - 30v_{i,j} + 16v_{i,j-1} - v_{i,j-2}}{12(\Delta y)^2} + O\left(h^4\right) \tag{29}$$

where $\widetilde{u_{i,j}}$ and $\widetilde{v_{i,j}}$ are the interpolated velocities at the cell centre (i,j):

$$\widetilde{u_{i,j}} = \frac{-u_{i+1,j} + 9u_{i,j} + 9u_{i-1,j} - u_{i-2,j}}{16} + O\left(h^4\right) \tag{30}$$

$$\widetilde{v_{i,j}} = \frac{-v_{i,j+1} + 9v_{i,j} + 9v_{i,j-1} - v_{i,j-2}}{16} + O\left(h^4\right) \tag{31}$$

and $\widetilde{u^c_{i,j}}$ and $\widetilde{v^c_{i,j}}$ are the interpolated velocities at the corner (i,j):

$$\widetilde{u^c_{i,j}} = \frac{-u_{i,j+2} + 9u_{i,j+1} + 9u_{i,j} - u_{i,j-1}}{16} + O\left(h^4\right) \tag{32}$$

$$\widetilde{v^c_{i,j}} = \frac{-v_{i+2,j} + 9v_{i+1,j} + 9v_{i,j} - v_{i-1,j}}{16} + O\left(h^4\right) \tag{33}$$

The partial derivatives of the $z-$momentum equation are defined similarly.

Finally, it should be noted that Hydro3D is parallelized using domain decomposition, that is, the computational domain is divided into a number of smaller sub-domains to allow parrallelisation. Communication across internal boundaries between neighbouring sub-domains is achieved using layers of ghost cells that are located around the peripheries of the domains. The standard message passing interface accomplishes communication between sub-domains.

2.2. *Local mesh refinement*

In the present study, a 2:1 reduction in grid element size between neighbouring sub-domains is imposed on the staggered computational grid to achieve LMR in critical areas. In the following, the arrangement of ghost cell velocities and pressures is mostly depicted in two dimensions for simplicity.

2.2.1. Calculation of pressure at the non-matching interface. The calculation of ghost cell pressures is achieved by adjusting the coarse pressure gradients at the coarse-fine interface to those computed for neighbouring fine cells, thereby coupling the pressure fields. In the prolongation process, pressure values, the location of which are indicated by '**x**' symbols in Figure 2, are first quadratically interpolated using pressure values from the coarse grid computational cells (indicated by solid right triangles, ▶, in the same figure) as follows:

$$\begin{aligned} p^a_{I,J} &= \frac{5p_{I-1,J} + 30p_{I,J} - 3p_{I+1,J}}{32} + O\left(h^2\right) \\ p^b_{I,J} &= \frac{5p_{I+1,J} + 30p_{I,J} - 3p_{I-1,J}}{32} + O\left(h^2\right) \end{aligned} \tag{34}$$

Another quadratic interpolation is then applied to calculate fine ghost pressures using the computational cells indicated by solid left triangles (◀) and the interpolated pressures:

$$\begin{aligned} p_{i,j+1} &= \frac{8p^a_{I,J} + 10p_{i,j} - 3p_{i,j-1}}{15} + O\left(h^2\right) \\ p_{i+1,j+1} &= \frac{8p^b_{I,J} + 10p_{i+1,j} - 3p_{i+1,j-1}}{15} + O\left(h^2\right) \end{aligned} \tag{35}$$

In the restriction process, the edge-centred gradients of the fine grids, F^a_{fine} and F^b_{fine} in Figure 2, are calculated as follows:

$$F^a_{fine} = \frac{p_{i,j+1} - p_{i,j}}{\Delta y_{fine}} + O(h) \quad \text{and} \quad F^b_{fine} = \frac{p_{i+1,j+1} - p_{i+1,j}}{\Delta y_{fine}} + O(h) \tag{36}$$

The coarse grid pressure gradients are then taken as the arithmetic average of the fine-grid pressure gradients to enforce the continuity of the gradient across the interface. Ghost cell pressures from the coarse grid $(I, J-1)$ are calculated using the following formula:

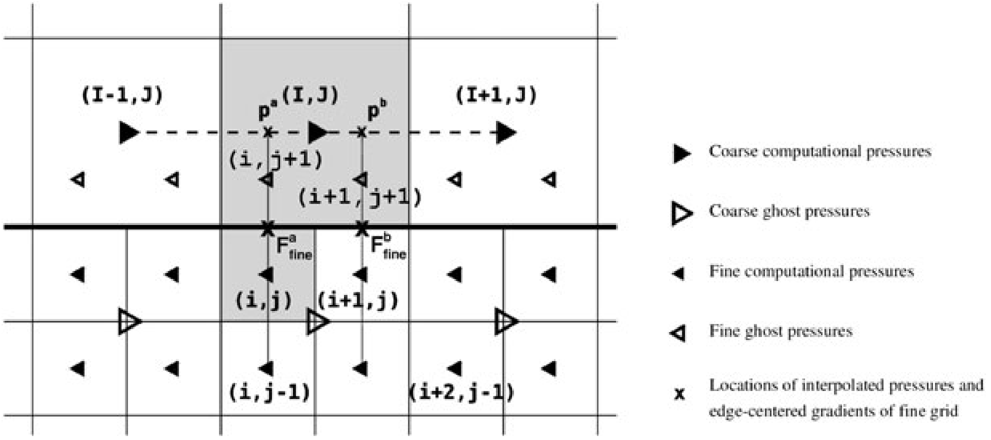


Figure 2. Pressure locations around the interface between a fine and coarse grid.

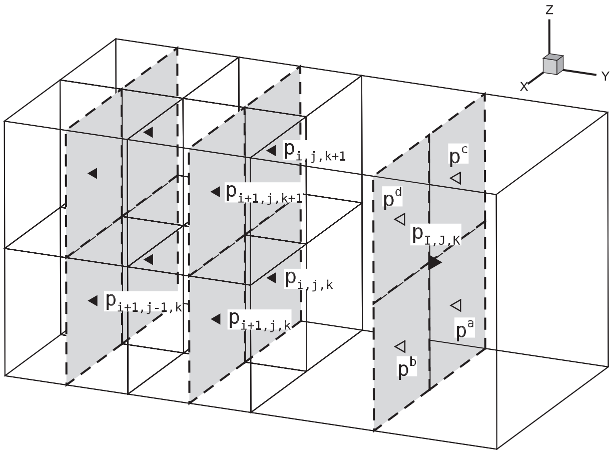


Figure 3. Pressure locations between fine and coarse grids for three-dimensional problems when the refinement is considered in the $y-$ direction.

$$\frac{p_{I,J} - p_{I,J-1}}{\Delta y_{coarse}} = \left(\frac{F^a_{fine} + F^b_{fine}}{2} \right) + O(h) \tag{37}$$

Although quadratic interpolations give first-order accurate approximations of the Laplacian operator along the interface, it is proposed in [36] that this is sufficient to ensure global second-order accuracy of the projected velocity field.

Figure 3 presents the indexing of neighbouring cells around a non-matching interface to illustrate the means by which pressure values are calculated in the three-dimensional domain. Considering refinement in the $y-$direction, a coarse cell, which is located on the north faces of four fine cells, is indexed as (I, J, K), and the south neighbours associated with it are indexed as follows: (i, j, k), $(i, j, k+1)$, $(i+1, j, k)$ and $(i+1, j, k+1)$. In the prolongation process, quadratic interpolations are performed to obtain the four ghost fine pressure values on the coarse side of the interface, which are denoted p^a, p^b, p^c and p^d in Figure 3:

$$\begin{aligned} p^a_{I,J,K} = \frac{1}{1024} \big(& 25p_{I-1,J,K-1} + 150p_{I,J,K-1} - 15p_{I+1,J,K-1} + 150p_{I-1,J,K} + 900p_{I,J,K} \\ & -90p_{I+1,J,K} - 15p_{I-1,J,K+1} - 90p_{I,J,K+1} + 9p_{I+1,J,K+1} \big) + O\left(h^2\right) \end{aligned} \tag{38}$$

$$\begin{aligned} p^b_{I,J,K} = \frac{1}{1024} \big(& -15p_{I-1,J,K-1} + 150p_{I,J,K-1} + 25p_{I+1,J,K-1} - 90p_{I-1,J,K} + 900p_{I,J,K} \\ & +150p_{I+1,J,K} + 9p_{I-1,J,K+1} - 90p_{I,J,K+1} - 15p_{I+1,J,K+1} \big) + O\left(h^2\right) \end{aligned} \tag{39}$$

$$\begin{aligned} p^c_{I,J,K} = \frac{1}{1024} \big(& -15p_{I-1,J,K-1} - 90p_{I,J,K-1} + 9p_{I+1,J,K-1} + 150p_{I-1,J,K} + 900p_{I,J,K} \\ & -90p_{I+1,J,K} + 25p_{I-1,J,K+1} + 150p_{I,J,K+1} - 15p_{I+1,J,K+1} \big) + O\left(h^2\right) \end{aligned} \tag{40}$$

$$\begin{aligned} p^d_{I,J,K} = \frac{1}{1024} \big(& 9p_{I-1,J,K-1} - 90p_{I,J,K-1} - 15p_{I+1,J,K-1} - 90p_{I-1,J,K} + 900p_{I,J,K} \\ & +150p_{I+1,J,K} - 15p_{I-1,J,K+1} + 150p_{I,J,K+1} + 25p_{I+1,J,K+1} \big) + O\left(h^2\right) \end{aligned} \tag{41}$$

Additional information pertaining to the derivation of Equations (38)–(41) can be found in [37]. Another quadratic interpolation is then applied to calculate fine ghost pressures using the computational cells indicated by solid left triangles (◀) and the interpolated pressures:

$$p_{i,j+1,k} = \frac{1}{15}\left(8p^{a}_{I,J,K} + 10p_{i,j,k} - 3p_{i,j-1,k}\right) + O\left(h^2\right) \tag{42}$$

$$p_{i+1,j+1,k} = \frac{1}{15}\left(8p^{b}_{I,J,K} + 10p_{i+1,j,k} - 3p_{i+1,j-1,k}\right) + O\left(h^2\right) \tag{43}$$

$$p_{i,j+1,k+1} = \frac{1}{15}\left(8p^{c}_{I,J,K} + 10p_{i,j,k+1} - 3p_{i,j-1,k+1}\right) + O\left(h^2\right) \tag{44}$$

$$p_{i+1,j+1,k+1} = \frac{1}{15}\left(8p^{d}_{I,J,K} + 10p_{i+1,j,k+1} - 3p_{i+1,j-1,k+1}\right) + O\left(h^2\right) \tag{45}$$

In the restriction process, the coarse ghost pressure value, $p_{I,J-1,K}$, is calculated by adjusting the coarse pressure gradients at the coarse-fine interface to those computed for neighbouring fine cells in the following form:

$$\begin{aligned}\frac{p_{I,J,K} - p_{I,J-1,K}}{\Delta y_{coarse}} = \frac{1}{4}\Bigg(&\frac{p_{i,j+1,k} - p_{i,j,k}}{\Delta y_{fine}} \\ &+\frac{p_{i+1,j+1,k} - p_{i+1,j,k}}{\Delta y_{fine}} + \frac{p_{i,j+1,k+1} - p_{i,j,k+1}}{\Delta y_{fine}} + \frac{p_{i+1,j+1,k+1} - p_{i+1,j,k+1}}{\Delta y_{fine}}\Bigg) + O(h)\end{aligned} \tag{46}$$

2.2.2. Calculation of ghost velocities tangential to the non-matching interface. Prolongation and restriction of ghost velocities, which are tangential to the non-matching interface, are required in the discrete formulations of derivatives. The procedures are different for velocity components that are tangential and normal to the non-matching LMR interfaces; evaluation of tangential velocities is not as challenging as evaluation of normal velocities. Figure 4 is a schema illustrating the arrangement of ghost cell tangential velocities at a fine-coarse grid interface.

In the prolongation procedure, fine grid tangential ghost velocities are calculated using quadratic interpolations with mass conservation in the following forms for two-dimensional and three-dimensional problems, respectively:

$$u_{i+1,j+1} = \frac{u_{I,J-1} + 8u_{I,J} - u_{I,J+1}}{8} + O\left(h^2\right) \tag{47}$$

$$\begin{aligned}u_{i+1,j+1,k} = \frac{1}{128}\big(&3u_{I,J-1,K-1} + 15u_{I,J,K-1} - 2u_{I,J+1,K-1} + 15u_{I,J-1,K} \\ &+128u_{I,J,K} - 15u_{I,J+1,K} - 2u_{I,J-1,K+1} - 15u_{I,J,K+1} + u_{I,J+1,K+1}\big) + O\left(h^2\right)\end{aligned} \tag{48}$$

$$\begin{aligned}u_{i+1,j+1,k+1} = \frac{1}{128}\big(&-2u_{I,J-1,K-1} - 15u_{I,J,K-1} + u_{I,J+1,K-1} + 15u_{I,J-1,K} \\ &+128u_{I,J,K} - 15u_{I,J+1,K} + 3u_{I,J-1,K+1} + 15u_{I,J,K+1} - 2u_{I,J+1,K+1}\big) + O\left(h^2\right)\end{aligned} \tag{49}$$

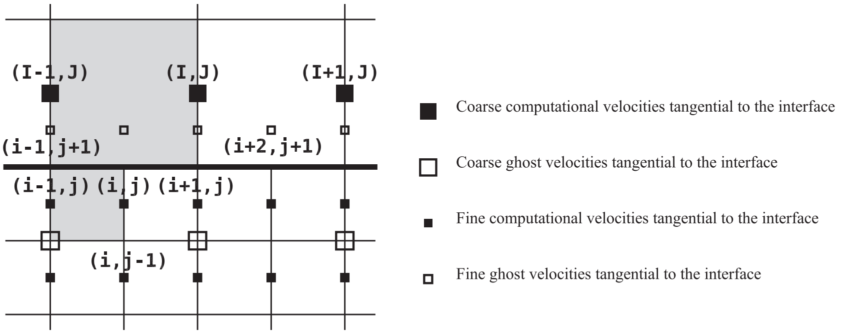

(I-1,J)
(I,J)
(I+1,J)
(i-1,j+1)
(i+2,j+1)
(i-1,j)
(i,j)
(i+1,j)
(i,j-1)
■ Coarse computational velocities tangential to the interface
□ Coarse ghost velocities tangential to the interface
▪ Fine computational velocities tangential to the interface
▫ Fine ghost velocities tangential to the interface

Figure 4. Locations of velocities tangential to the LMR interface between fine and coarse grids.

The second-order average of these edge values is then taken to obtain the value at the location in between: $u_{i+2,j+1} = (u_{i+1,j+1} + u_{i+3,j+1})/2$ in 2D and $u_{i,j+1,k+1} = (u_{i-1,j+1,k+1} + u_{i+1,j+1,k+1})/2$ in 3D.

In the restriction procedure, the coarse tangential velocities are obtained by averaging the computed fine values, thus enforcing mass conservation at the interfaces. Mass conservation is also enforced for velocities normal to the interface (Section 2.2.3). The coarse grid ghost velocity, indexed by $(I, J-1)$, is calculated as follows:

$$u_{I,J-1} = \frac{u_{i+1,j} + u_{i+1,j-1}}{2} + O\left(h^2\right) \tag{50}$$

and the three-dimensional procedure for the coarse grid ghost velocity indexed by $(I, J-1, K)$, which is prescribed for the three-dimensional pressure formulation, is achieved for both approaches as follows:

$$u_{I,J-1,K} = \frac{u_{i+1,j,k} + u_{i+1,j-1,k} + u_{i+1,j,k+1} + u_{i+1,j-1,k+1}}{4} + O\left(h^2\right) \tag{51}$$

2.2.3. Calculation of ghost velocities normal to the non-matching interface. Interpolation and restriction procedures for the velocity component normal to the interface differ slightly from those used for the tangential velocities. Figure 5 presents the arrangement of ghost cells.

In the prolongation process, the values marked with '**x**' symbols in Figure 5 are first calculated using coarse grid computational values as follows:

$$\begin{aligned} v_{i,j+2} &= \frac{v_{I-1,J} + 8v_{I,J} - v_{I+1,J}}{8} + O\left(h^2\right) \\ v_{i+1,j+2} &= \frac{v_{I+1,J} + 8v_{I,J} - v_{I-1,J}}{8} + O\left(h^2\right) \end{aligned} \tag{52}$$

One additional quadratic interpolation between computed coarse and fine grid values is then employed using $v_{i,j+2}$ and $v_{i+1,j+2}$, which are interpolated quadratically with mass conservation enforced. This interpolation is defined as follows:

$$\begin{aligned} v_{i,j+1} &= \frac{1}{3}\left(v_{i,j+2} + 3v_{i,j} - v_{i,j-1}\right) + O\left(h^2\right) \\ v_{i+1,j+1} &= \frac{1}{3}\left(v_{i+1,j+2} + 3v_{i+1,j} - v_{i+1,j-1}\right) + O\left(h^2\right) \end{aligned} \tag{53}$$

The formulation for three-dimensional problems is similar. First, the normal velocities on the coarse edge are interpolated as follows:

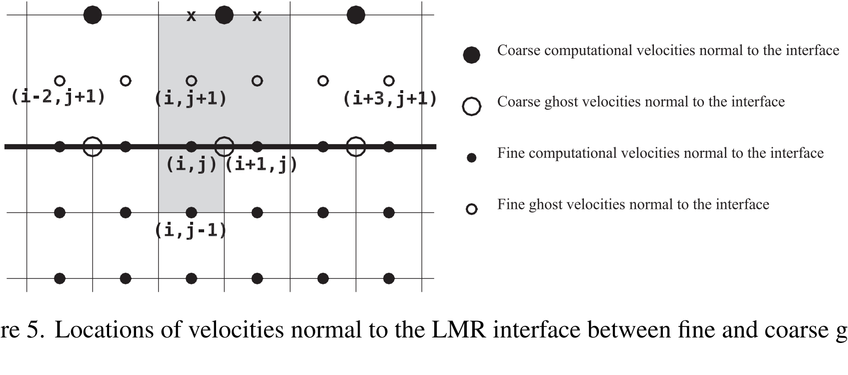

(I-1,J)
(I,J)
(I+1,J)
x
x
(i-2,j+1)
(i,j+1)
(i+3,j+1)
(i,j)
(i+1,j)
(i,j-1)
● Coarse computational velocities normal to the interface
○ Coarse ghost velocities normal to the interface
• Fine computational velocities normal to the interface
∘ Fine ghost velocities normal to the interface

Figure 5. Locations of velocities normal to the LMR interface between fine and coarse grids.

$$\begin{aligned} v_{i,j+2,k} = \frac{1}{128}\big(&3v_{I-1,J,K-1} + 15v_{I,J,K-1} - 2v_{I+1,J,K-1} + 15v_{I-1,J,K} \\ &+128v_{I,J,K} - 15v_{I+1,J,K} - 2v_{I-1,J,K+1} - 15v_{I,J,K+1} + v_{I+1,J,K+1}\big) + O\left(h^2\right) \end{aligned} \tag{54}$$

$$\begin{aligned} v_{i+1,j+2,k} = \frac{1}{128}\big(&-2v_{I-1,J,K-1} + 15v_{I,J,K-1} + 3v_{I+1,J,K-1} - 15v_{I-1,J,K} \\ &+128v_{I,J,K} + 15v_{I+1,J,K} + v_{I-1,J,K+1} - 15v_{I,J,K+1} - 2v_{I+1,J,K+1}\big) + O\left(h^2\right) \end{aligned} \tag{55}$$

$$\begin{aligned} v_{i,j+2,k+1} = \frac{1}{128}\big(&-2v_{I-1,J,K-1} - 15v_{I,J,K-1} + v_{I+1,J,K-1} + 15v_{I-1,J,K} \\ &+128v_{I,J,K} - 15v_{I+1,J,K} + 3v_{I-1,J,K+1} + 15v_{I,J,K+1} - 2v_{I+1,J,K+1}\big) + O\left(h^2\right) \end{aligned} \tag{56}$$

$$\begin{aligned} v_{i+1,j+2,k+1} = \frac{1}{128}\big(&v_{I-1,J,K-1} - 15v_{I,J,K-1} - 2v_{I+1,J,K-1} - 15v_{I-1,J,K} \\ &+128v_{I,J,K} + 15v_{I+1,J,K} - 2v_{I-1,J,K+1} + 15v_{I,J,K+1} + 3v_{I+1,J,K+1}\big) + O\left(h^2\right) \end{aligned} \tag{57}$$

before the quadratic interpolation is performed according to Equation (53) to compute the values of $v_{i,j+1,k}$, $v_{i+1,j+1,k}$, $v_{i,j+1,k+1}$ and $v_{i+1,j+1,k+1}$.

In the restriction process, the fourth-order averages of computed fine grid values are used to calculate the coarse grid ghost velocities instead of second-order averages as was the case for the tangential velocities (Equations 50 and 51). The fourth-order formulation is as follows:

$$v_{I,J-1} = \frac{-v_{i-1,j} + 9v_{i,j} + 9v_{i+1,j} - v_{i+2,j}}{16} + O\left(h^4\right) \tag{58}$$

In 3D, the formulation of the fourth-order average is slightly more complicated than in 2D. Figure 6 presents a view of a 3D interface in the $x-z$ plane, with the locations of fine and coarse grid velocities denoted by hollow and solid circles, respectively. In the first step, the values marked with '**x**' symbols are calculated as follows, using the first '**x**' on the left side of Figure 6 as an example:

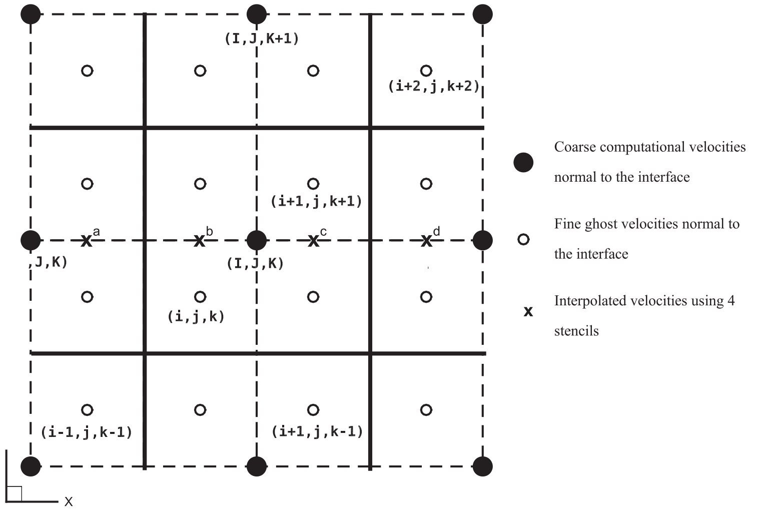

(I,J,K+1)
(i+2,j,k+2)
(i+1,j,k+1)
a
b
c
d
,J,K)
(I,J,K)
(i,j,k)
(i-1,j,k-1)
(i+1,j,k-1)
x
Coarse computational velocities normal to the interface
Fine ghost velocities normal to the interface
Interpolated velocities using 4 stencils

Figure 6. Locations of velocities normal to the LMR interface between fine and coarse grids for three-dimensional problems when the refinement is considered in the $y-$ direction.

$$v^a = \frac{-v_{i-1,j,k-1} + 9v_{i-1,j,k} + 9v_{i-1,j,k+1} - v_{i-1,j,k+2}}{16} + O\left(h^4\right) \qquad (59)$$

The coarse grid ghost velocity is then evaluated using the calculated values at '**x**' locations as follows:

$$v_{I,J-1,K} = \frac{-v^a + 9v^b + 9v^c - v^d}{16} + O\left(h^4\right) \qquad (60)$$

Finally, it should be noted that although the generated grid consists of domains at different resolutions, refinement in time is not employed in the present study. Instead, all grids in all levels are advanced with a uniform time step to simplify the implementation and to make for more straightforward parallelisation and CPU load distribution.

3. PERFORMANCE ASSESSMENT: COMPUTATIONAL SETUP AND BOUNDARY CONDITIONS FOR TWO VALIDATION CASES

In order to assess the performance of the LES-LMR code, two flows are simulated. The first is fully developed turbulent flow in a straight channel, probably the most prominent validation case for LES. The challenge lies in the accurate prediction of wall-shear stress and profiles of first-order and second-order statistics. The second case is the flow over a matrix of cubes. Here, the goal is to reproduce numerically the well-known experiment of Meinders *et al.* [38], which is included in the European Research Community On Flow, Turbulence and Combustion (ERCOFTAC) data base of standard test cases for Computational Fluid Dynamics (CFD) developers. The computational setup and boundary conditions for the LES of the two cases are presented in the following subsections, and Table I summarises all of the simulations that have been undertaken.

3.1. Fully developed turbulent channel flow

The Reynolds number based on half channel height, δ, and bulk velocity of this flow is $Re = 10,935$, which is exactly the same as the DNS of Moser *et al.* [39], data from which are used for comparison. In the streamwise direction (along the x-axis), the computational domain spans $L_x \approx 2\pi\delta$. The domain has a cross-section of $L_y = 2\delta$ and $L_z \approx \pi\delta$ in wall-normal (y-axis) and spanwise (z-axis) directions, respectively. No-slip boundary conditions are stipulated at the walls, and periodic boundary conditions are used in streamwise and spanwise directions. The streamwise pressure gradient, $\delta p/\delta x$, that drives the flow is adjusted at each time step to ensure a constant mass flow rate through the channel.

Three different grids are tested, and these are sketched in Figure 7. LES that employed second-order and fourth-order spatial discretisations were performed on each grid. The finest grid, that is, the

Table I. Summary of test cases.

Name	Case	Grid	Spatial discretisation	Normal ghost cell interpolation
CH-FIN-O2	Channel	Uniform fine	Second-order	Equation (60)
CH-CRS-O2	Channel	Uniform coarse	Second-order	Equation (60)
CH-LMR-O2	Channel	LMR	Second-order	Equation (60)
CH-FIN-O4	Channel	Uniform fine	Fourth-order	Equation (60)
CH-CRS-O4	Channel	Uniform coarse	Fourth-order	Equation (60)
CH-LMR-O4	Channel	LMR	Fourth-order	Equation (60)
CH-LMR-O4b	Channel	LMR	Fourth-order	Equation (51)
CB-LMR-500	Cube, $Re_b = 500$	LMR	Fourth-order	Equation (60)
CB-LMR	Cube, $Re_b = 13000$	LMR	Fourth-order	Equation (60)
CB-UNI	Cube, $Re_b = 13000$	Uniform	Fourth-order	Equation (60)

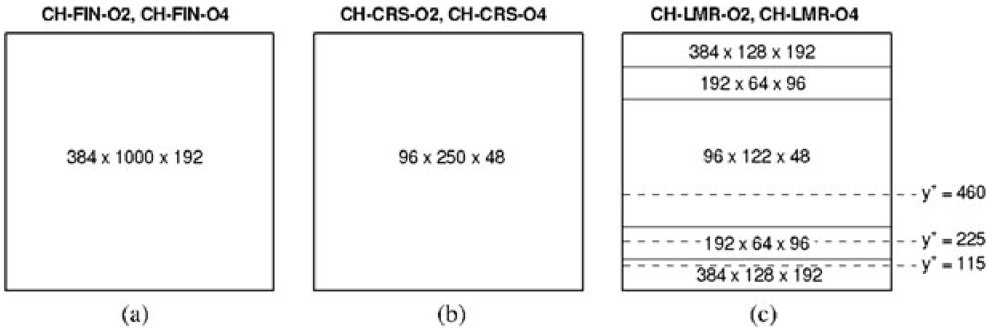


Figure 7. Grid resolutions used in the channel flow test case: (a) fine uniform, (b) coarse uniform and (c) LMR. Solid lines in plot (c) represent LMR interfaces, and dashed lines denote spectra locations.

one used for CH-FIN-O2 and CH-FIN-O4, comprises approximately 74 million grid points, while that one used for CH-CRS-O2 and CH-CRS-O4 is much coarser and comprises approximately 1 million grid points. The grid for CH-LMR-O2 and CH-LMR-O4 uses LMR and incorporates three resolution levels; the entire computational domain contains approximately 22 million grid points. The refinement factor between two neighbouring resolutions is 2:1 in all spatial directions. The resolution of the uniform grids in wall units is as follows: the uniform fine grid (CH-FIN-O2 and CH-FIN-O4) has grid spacings of $\Delta x^+ = \Delta z^+ = 9.8$ and $\Delta y^+ = 1.2$, the grid spacings of the uniform coarse grid (CH-CRS-O2 and CH-CRS-O4) are $\Delta x^+ = \Delta z^+ = 39.7$ and $\Delta y^+ = 4.8$. The spacings of the LMR grid (CH-LMR-O2 and CH-LMR-O4) in wall units in the fine-grid near-wall region are $\Delta x^+ = \Delta z^+ = 9.5$ and $\Delta y^+ = 1.2$. For comparison, the DNS grid of [39] featured grid spacings of $\Delta x^+ = 9.7$, $\Delta z^+ = 4.8$ and $\Delta y^+ = 7.2$ plus 13 or more Chebyshev grid points in the near-wall ($\Delta y^+ \leqslant 10$) region.

The computations are first run for approximately 10 convective time units (tu_τ/δ, where u_τ is friction velocity) to reach fully developed turbulent flow, and then statistics are collected over approximately 12 additional convective time units. All flow profiles are averaged in the streamwise and spanwise directions.

As the main motivation for using LMR is to reduce overall computational expense by reducing grid resolution in areas where it is not needed, it is pertinent to compare the relative savings that were achieved in the present investigation when the LMR grid was used. To facilitate such a comparison, the uniform fine and LMR cases with fourth-order interpolations, CH-FIN-O4 and CH-LMR-O4, are considered. In both cases, the grids were decomposed into 256 sub-domains of equal dimensions, which were distributed between a number of parallel processors in such a manner as to divide the computational load equally between the processors. The uniform fine grid simulation used 256 processors (one sub-domain per processor), while the LMR simulation used only 112 (between one and eight sub-domains per processor). The average wall time required for the fine uniform grid was 1.344 seconds per time step, while for the LMR grid, it was 1.355 seconds per time step, equating to an average increase of 0.8%. This small increase in the per-time step computational expense is primarily due to the extra interpolation operations that are required to restrict and prolong variables between meshes of different resolutions in the LMR algorithm. In fact, in the LMR simulation, the total wall time required to perform one exchange of pressures and velocities between two adjacent sub-domains with non-matching resolutions (one in the finest region and one in the intermediate region) was observed to require 2.5 times as much wall time as the exchange between the same sub-domains in the uniform case. However, the proportion of the overall computation time taken up by the exchange operations is sufficiently small to ensure that the overall increase is only 0.8%. The overall expense of the uniform fine grid simulation was $256 \times 1.344 \approx 344$ CPU seconds, whereas for the LMR simulation, it was $112 \times 1.344 \approx 152$ CPU seconds, a saving of almost 56%.

3.2. Turbulent flow over a matrix of cubes

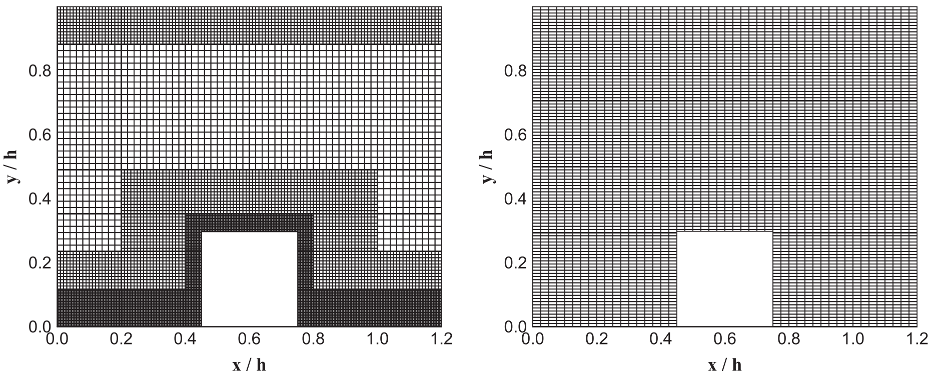
In the experiment of Meinders *et al.* [38], 250 cubes of height k were placed on one wall of a wide wind tunnel of height $h = 3.4k$. The cubes were arranged in a regular fashion, with equal streamwise and spanwise spacings ($S_x = S_z = 1.2h$). The Reynolds number of the flow based on tunnel height, h, was $Re_h = 13000$.

Meinders *et al.* proposed that the matrix of cubes was wide and long enough to ensure that the flow around individual cubes that were positioned sufficiently far from the edge of the matrix could be considered to be fully developed and spatially periodic. A number of researchers (see, for example, [40] and [41]) have therefore chosen to simulate this case using a domain of dimensions $S_x \times S_z \times h$, with periodic conditions applied to the spanwise and streamwise boundaries, no-slip conditions applied to the top and bottom walls, and the cube located at the centre of the bottom wall; this is the approach that is adopted here.

The LESs are performed on two computational grids: one in which the grid resolution is increased in the vicinity of the cube and walls using LMR to ensure proper resolution of the near-wall regions and one ‘affordable’ coarse grid in which the grid spacing is uniform across the whole domain. Figure 8 presents longitudinal sections of the two grids in the spanwise mid plane (i.e. $z/h = 0.6$). The grid with LMR comprises approximately 18.3 million grid points; if the finest resolution of this grid was applied uniformly throughout the domain, the grid would comprise nearly 100 million grid points. The minimum grid resolution in wall units in the finest grid zone, determined using the peak local friction velocity, which occurs at the bottom corners of the leading face of the cube, is as follows: $\Delta x^+ = \Delta y^+ = \Delta z^+ \approx 3.32$. In the intermediate zone, the mesh resolution is $\Delta x^+ = \Delta y^+ = \Delta z^+ \approx 6.65$, and in the coarse zone, it is $\Delta x^+ = \Delta y^+ = \Delta z^+ \approx 13.3$. The uniform grid comprises $97 \times 97 \times 205$ (≈ 1.93 million) points, and the cell sizes yield minimum resolutions in wall units of $\Delta x^+ = \Delta z^+ \approx 16.6$ and $\Delta y^+ \approx 6.65$. Unlike in the LMR grid, the cells of which have sides of equal length in all directions, the cells in the uniform grid are anisotropic. It should be noted that, as flow variables are stored in a staggered arrangement, the normalised wall distance of the first grid point is always half of the resolution quoted previously. This means that the first grid point for streamwise velocity, for example, is located at $y^+ \approx 1.66$ for the LMR case and $y^+ \approx 3.32$ for the uniform case. On the sides of the cube, the first grid point for streamwise velocity is located at $z^+ \approx 1.66$ for the LMR case and $z^+ \approx 8.31$ for the uniform case.

Before replicating the exact experimental conditions, a lower Reynolds number laminar flow ($Re_h = 500$) is simulated to provide preliminary confirmation that the LMR algorithm is capable of computing flow variables accurately and stably near transitions between zones of differing resolution and in areas of steep gradients. The computational domain for this laminar case is the same as that described earlier, but the computational grid is much coarser.

The IBM [8, 9] is employed to represent the cubes in the Cartesian grid. In the present investigation, the direct forcing approach that was first introduced by [42] and subsequently improved by [43]



(a) (b)

Figure 8. Numerical grids for the turbulent cube flow case, shown here in the $x - y$ plane at $z/h = 0.6$: (a) CB-LMR and (b) CB-UNI.

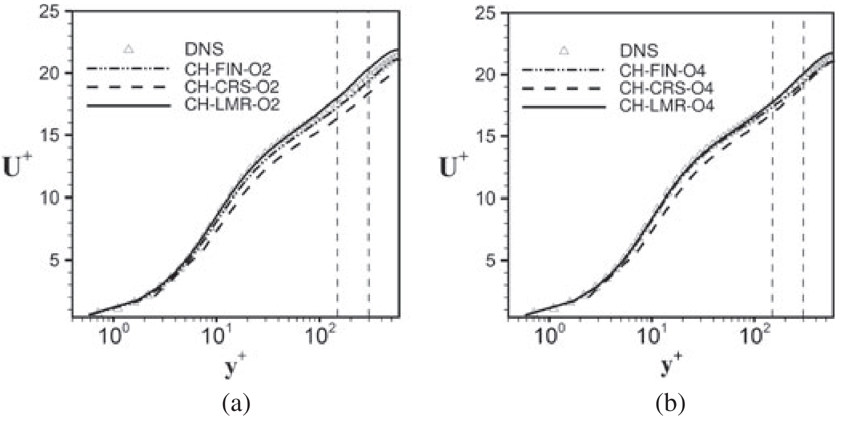
is used. This approach relies on the transmission of Eulerian velocities to Lagrangian markers: a reaction force is calculated and then added as a source/sink to the momentum equations. The accuracy of the IBM depends on the fineness of the Lagrangian representation: the more Lagrangian points used, the more accurately the body is represented; therefore, the finer the Cartesian grid, the more accurate the representation of the body. In the turbulent LMR case (CB-LMR), the cube is defined by approximately 1.8 million Lagrangian points, while in the uniform turbulent case (CB-UNI) it is defined by approximately 38 000 points. In the laminar flow case, 15 600 Lagrangian points define the cube.

4. RESULTS AND DISCUSSIONS

4.1. Fully developed turbulent channel flow

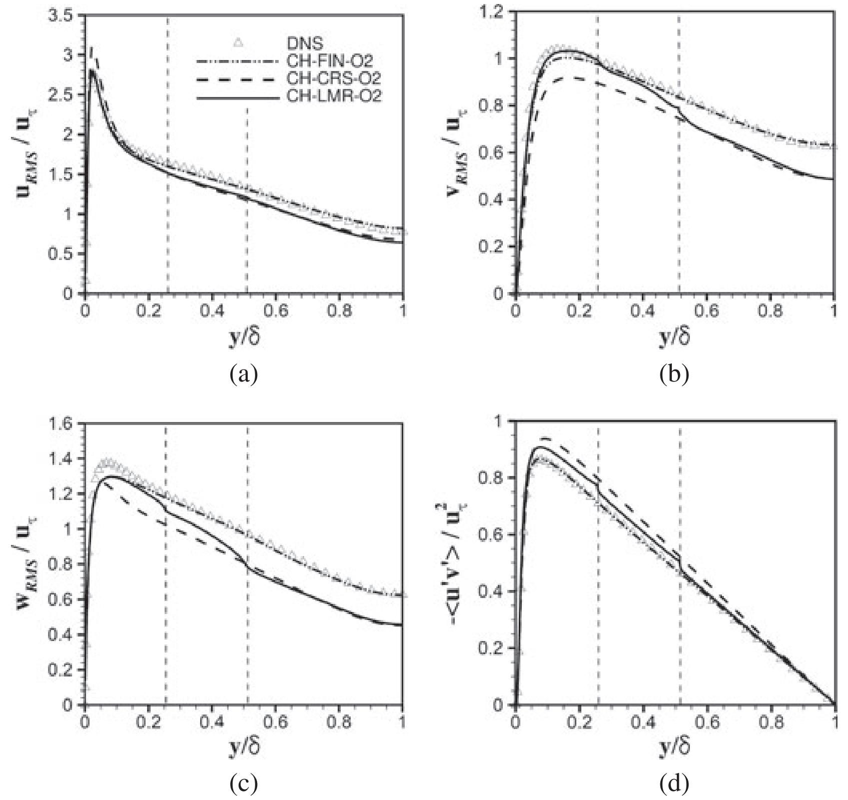
This section presents the results of a series of turbulent channel flow simulations that have been performed on grids with different resolutions, as summarised in Figure 7, as well as with different (i.e. second or fourth order) spatial discretisations. Cases CH-FIN-O2, CH-CRS-O2 and CH-LMR-O2 refer to LES in which second-order central spatial discretisation schemes are employed, and cases CH-FIN-O4, CH-CRS-O4 and CH-LMR-O4 refer to LES in which fourth-order spatial discretisation schemes are used. First of all, the effect of grid resolution on the friction velocity is established. The DNS featured a Reynolds number based on friction velocity of $Re_\tau = 590$, for comparison, Re_τ is 593 and 592 for CH-FIN-O2 and CH-FIN-O4, 621 and 602 for CH-CRS-O2 and CH-CRS-O4, and 572 and 578 for CH-LMR-O2 and CH-LMR-O4.

Profiles of the time-averaged streamwise velocity normalised by the friction velocity are presented in Figure 9, with the DNS result of [39] included for comparison. The dash-double dot lines in Figure 9(a) and (b) show the results of the computation on the fine grid (i.e. CH-FIN-O2 and CH-FIN-O4), and they are in good agreement with the DNS data, regardless of whether second-order or fourth-order spatial discretisations are employed: this indicates that this level of resolution is adequate to produce an accurate solution with the present LES. The velocities in CH-CRS-O2 and CH-CRS-O4 are consistently less than those of the DNS, which is the result of the overestimation of the shear on the coarse grid, which is in turn due to the coarseness of the grid near the wall. Interestingly, higher-order spatial discretisation lessens this underestimation in the middle of the channel, as can be seen when comparing the dashed lines of Figure 9(a) and (b) relative to the DNS data. The time-averaged velocity profile of the LES on the locally refined grid exhibits slight overestimation of the velocity around the mid-channel, but this overestimation disappears when higher-order discretisation is employed. It is apparent that fourth-order spatial discretisation results in slightly better accuracy in areas where the grid is coarse. Also noteworthy is the fact that the LMR velocity profiles do not exhibit any discontinuity at the coarse-fine grid interfaces (denoted in the figure by dashed vertical lines).



(a) (b)

Figure 9. Mean streamwise velocity profile of fully developed turbulent channel flow at $Re = 10935$.

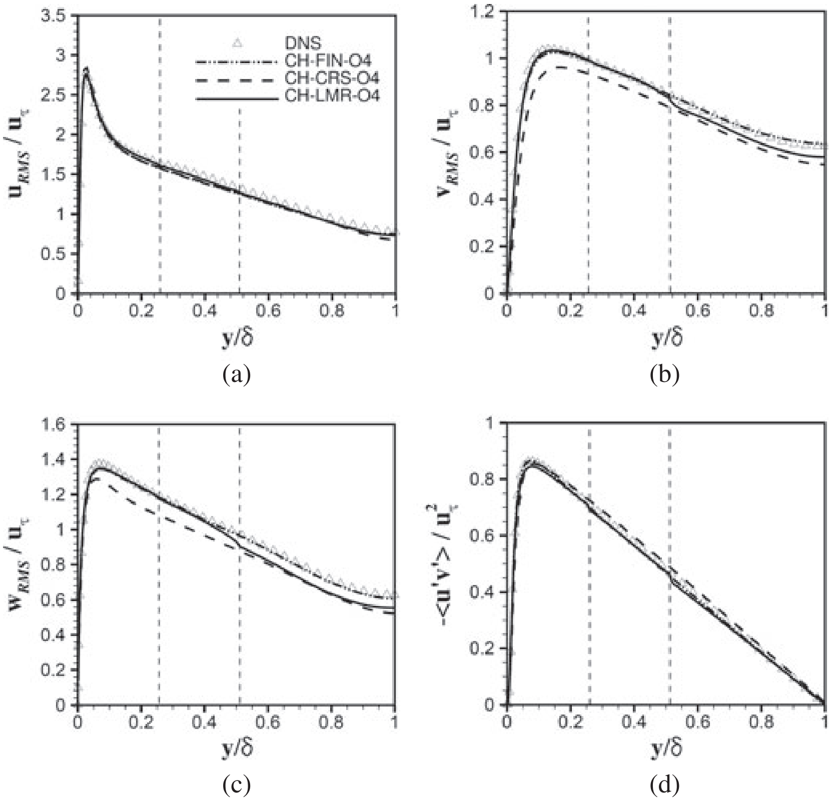


(a) (b) (c) (d)

Figure 10. Comparison of RMS velocity fluctuations normalised by the friction velocity for three different grids, obtained using second-order spatial discretisation.

Profiles of turbulence intensities normalised by the friction velocity are presented in Figure 10 for three different grid resolutions using second-order spatial discretisation together with DNS data. The simulated profiles from the uniform fine grid simulation are in good agreement with the DNS data for all four fluctuating quantities. Although the general shape of the profiles is in good agreement, there are some discrepancies for the uniform coarse and locally refined grid cases. The coarse grid case, represented by the dashed line, overestimates the normalised root-mean-square (RMS) streamwise velocity fluctuation in the near-wall region and underestimates it in the middle of the channel (Figure 10(a)). Furthermore, the coarse grid simulation underestimates the normalised spanwise and normal velocity fluctuations, v_{RMS} and $\underline{w_{RMS},}$ throughout the channel (Figure 10(b) and (c)) and overestimates the normalised shear stress, $-u'v'$, most significantly near the wall (Figure 10(d)). This suggests that energetic features of the flow are not properly captured, and simple eddy viscosity-based SGS models are not sufficient to model them. The locally refined grid, which is represented by the solid line, offers improved predictions near the wall, but similarly to the coarse grid, some underestimation of turbulent fluctuations in the mid-channel area is evident for u_{RMS}, v_{RMS} and w_{RMS}, while the shear stress is overestimated until $y/\delta = 0.5$.

Figure 10 reveals an important feature of the velocity fluctuations computed using the LMR method: significant drops can be observed across the LMR interfaces, particularly for v_{RMS}, w_{RMS} and $-\overline{u'v'}$. For CH-LMR-O2, there exist two LMR interfaces, which are located at $y = 0.256\delta$ and 0.512δ, and drops in the normal and spanwise fluctuations at these locations reduce the accuracy and stability of the simulation. In the middle of the channel, all four fluctuations are similar to the profile of CH-CRS-O2, which suggests that introducing finer resolution in the near-wall region does not improve the accuracy of the solution in the coarser grid regions. In addition, the profiles of the normal and spanwise velocity fluctuations bulge upwards between $y = 0.256\delta$ and 0.512δ, which provides an indication that LMR has a negative effect on the computation of the turbulence physics.



(a) (b)

(c) (d)

Figure 11. Comparison of RMS velocity fluctuations for three different grids, obtained using fourth-order spatial discretisation.

Figure 11 displays turbulence intensities normalised by the friction velocity for three different cases using fourth-order spatial discretisation. The order of the spatial discretisation does not affect the results for the uniform fine grid case, as can be seen by comparing Figures 10 and 11. The positive impact of higher-order spatial discretisation can however be observed for the uniform coarse and LMR cases. Profiles of u_{RMS} and $-\overline{u'v'}$ exhibit good agreement with the DNS computation for the coarse and LMR grid simulations. When the profiles of v_{RMS} and w_{RMS} are examined (Figure 11(b) and (c)), the underprediction of quantities in the middle of the channel for the uniform coarse and LMR cases is not as severe as it was with second-order discretisation, suggesting improved accuracy of the results with higher-order spatial discretisation. Noteworthy for the LMR case is the fact that the drops in the normal and spanwise velocity fluctuations at the LMR interfaces are less severe than before and that discontinuities between the fine and medium mesh, that is, at $y = 0.256\delta$, are almost imperceptible. Although the uniform coarse grid and LMR cases have the same grid resolution in the middle of the channel, improved accuracy is obtained with fourth-order spatial discretisation in the LMR simulation in this region. The bulging of the profiles of normal and spanwise velocity fluctuations between $y = 0.256\delta$ and 0.512δ disappears for CH-LMR-O4.

In addition to the spatial discretisation, it is worthwhile to consider the effect of the choice of interpolation scheme for the calculation of coarse grid ghost velocities normal to the interface on the normal turbulence profile. Figure 12 presents profiles of the normalised wall-normal turbulent fluctuation, v_{RMS}, calculated using two different interpolation schemes, alongside the DNS profile. In CH-LMR-O4, the normal ghost velocities are calculated using the approach outlined in Section 2.2.3, which is also the approach that has been adopted for normal ghost velocities in all simulations that have been discussed up to this point. In CH-LMR-O4b, the normal ghost velocities are calculated using the approach that was proposed for velocities *parallel* to the LMR interface, as outlined in Section 2.2.2. An important difference between the two approaches relates to the restriction

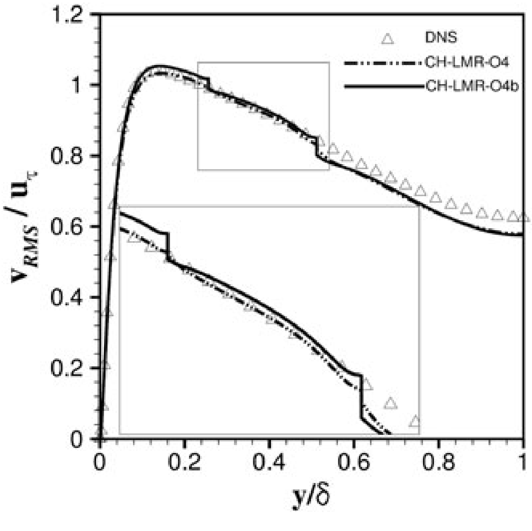


Figure 12. Comparison of RMS normal velocity fluctuation using two different averaging schemes for ghost coarse velocities normal to the LMR interface.

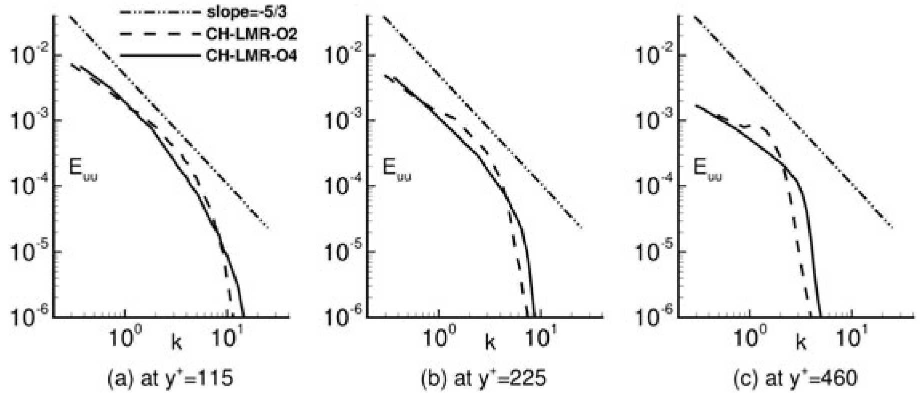


Figure 13. One-dimensional streamwise energy spectra at different y^+ locations for turbulent channel flow with LMR. The locations at which the spectra are taken are denoted in Figure 7.

of fine grid values onto the coarse grid: in CH-LMR-O4, a fourth-order interpolation is used, whereas in CH-LMR-O4b, the interpolation is second order. The figure shows that v_{RMS} profiles are generally very similar in both cases, but notable differences occur at the near-wall peak and in the vicinity of interfaces between different resolution levels. The peak of the fluctuation is overpredicted for CH-LMR-O4b, but perhaps more importantly CH-LMR-O4b produces larger drops across the interfaces than CH-LMR-O4. It should be noted that the number of multigrid iterations required for CH-LMR-O4 is higher than for CH-LMR-O4b: CH-LMR-O4 required approximately 3.1 iterations per time step compared with approximately 2.3 for CH-LMR-O4b, and CH-LMR-O4b therefore required approximately 17% less wall time per time step on average. However, the significant improvement in the continuity of the profile across the interfaces that was obtained with CH-LMR-O4 justifies the additional computational expense.

One-dimensional streamwise power density spectra are shown in Figure 13 to illustrate the effect of the grid resolution on the turbulent energy transfer. Here, k is the wave number in the streamwise direction, and E_{uu} is the power density obtained from fast Fourier transforms of streamwise velocity time signals taken at different y^+ locations. These locations are in areas of different grid resolution in the LMR grid (refer to Figure 7). The spectra of the LES using second-order and fourth-order discretisations are compared with the dash-double dot line, which has a slope of $-5/3$ and indicates homogeneous turbulence. At the first location, $y^+ = 115$, which is in the near-wall region with the finest grid level, the spectra of both cases are very similar and follow the $-5/3$rd line for a large range of wavenumbers (nearly two decades). However, the second-order discretisation simulation is not quite parallel to the $-5/3$rd line and it appears to fall-off a bit earlier. In the medium mesh region, this trend is a bit more pronounced, and it appears that the resolved energy is not transferred properly among the various scales in the second-order discretisation simulation. This is clearly visible in the coarsest grid energy spectrum, where a pile-up of energy is observed just before the cut-off wavenumber (i.e. beyond which the SGS model quickly dissipates all turbulent energy from the flow). The LES with the fourth-order spatial discretisation appears to transfer the energy across the grid interfaces more effectively, but the slope is no longer parallel to the $-5/3$rd line.

4.2. *Turbulent flow over a matrix of wall-mounted cubes*

Figure 14(a) presents contours of instantaneous streamwise velocity in the $x - y$ plane at $z/h = 0.6$ for the laminar ($Re_h = 500$) flow case. The plot reveals that velocities below the top of the cube in this plane are very low, with some local acceleration noticeable immediately above it, especially near the leading edge. The transitions between the high and low resolution zones are denoted by the thick red line, and the contours are seen to behave continuously across these transitions. This is also evidenced in Figure 14(c), which presents profiles of u at the two locations indicated by dashed lines in Figure 14(a).

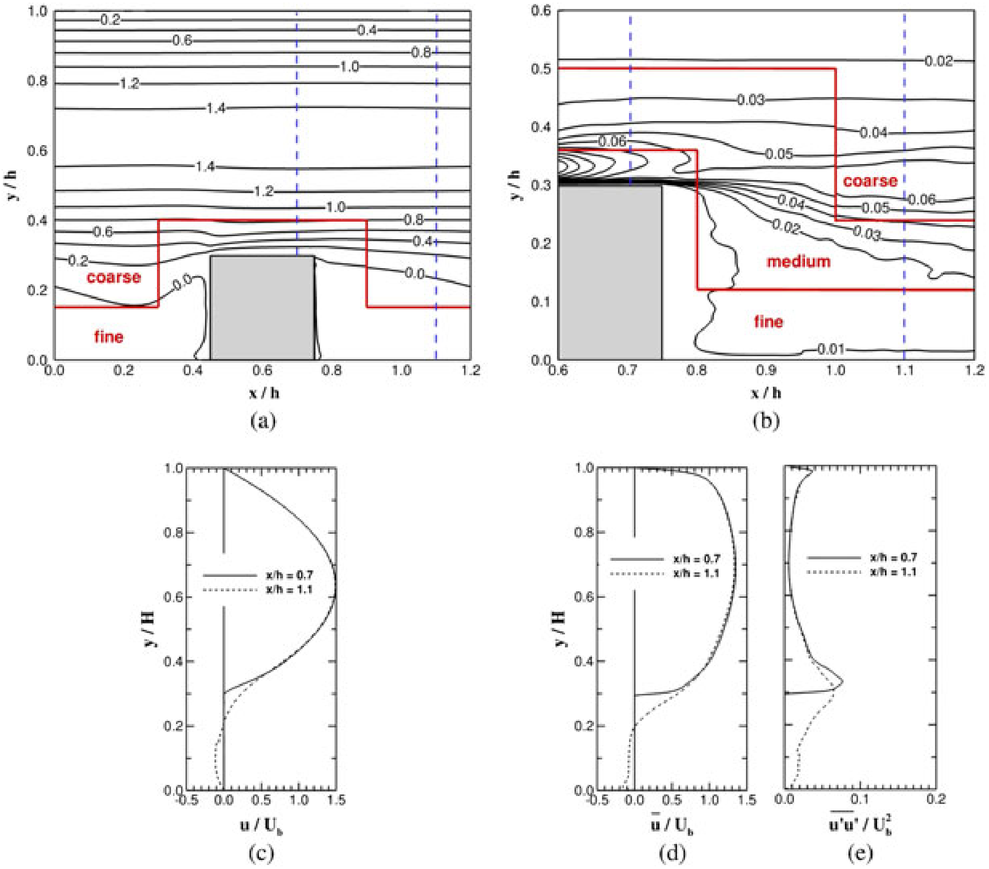
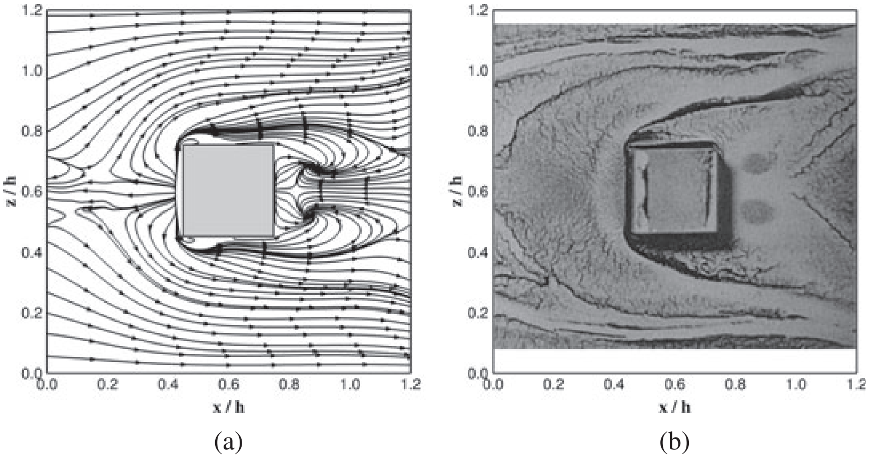


Figure 14. Contour and vertical profile plots displaying continuity of flow variables across grid resolution transitions: (a) contours of streamwise velocity, u, in the laminar flow case, $Re_h = 500$; (b) contours of normal stress, $\overline{u'u'}$, in the turbulent flow case, $Re_h = 13000$; (c) vertical profiles of u in the laminar flow case at two streamwise locations; and (d) and (e) vertical profiles of $\overline{u}$ and $\overline{u'u'}$ in the turbulent flow case at two streamwise locations.

Figure 14(b) presents contours of time-averaged streamwise normal stress, $\overline{u'u'}$, also in the $x-y$ plane at $z/h = 0.6$, for the turbulent ($Re_h = 13,000$) flow case. The plot indicates that the normal stress variation is largely continuous across the transitions between fine grid and medium grid resolutions, and that only minor jumps at the medium to coarse grid interface can be observed. Figure 14(e) and (f) presents vertical profiles of $\overline{u}$ and $\overline{u'u'}$ at the two streamwise locations indicated by dashed lines in Figure 14(c) and also suggests that the transitions between regions of differing resolutions are largely continuous. This is an important result, bearing in mind the drops that were observed in the turbulent intensities in the channel flow case, and provides a strong indication that the interpolation of ghost cell velocities is achieved satisfactorily in flows that are much more complex than the previous test case.

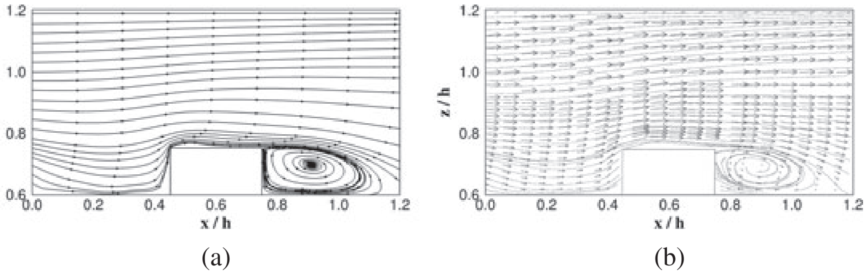
Simulated streamlines of the time-averaged flow very close to the bottom wall ($y/h = 0.0025$) are plotted for the LMR LES in Figure 15, alongside an oil film flow visualisation from the experiments of Meinders *et al.* Although direct quantitative comparison is difficult, there are some very encouraging similarities such as the position of the two recirculation nodes in the cube's mid-wake region and the general shape of the streamlines around the cube.

Figure 16 presents time-averaged flow streamlines in the $x-z$ plane a quarter of the way up the cube ($y/k = 0.25$, $y/h = 0.075$). The simulated streamlines, from the LMR LES (CB-LMR), are plotted in Figure 16(a), and the corresponding plot from the experiments of Meinders *et al.* is shown in Figure 16(b). The agreement is very encouraging: the computed mean length of the circulation bubble is very close to the experimentally observed length, indicating that the separating shear layers on the sides of the cube are accurately computed in the LES. Noteworthy is the fact that streamlines are continuous, evidence of the absence of discontinuities near grid interfaces.



(a) (b)

Figure 15. (a) Time-averaged streamlines in the $x-y$ plane close to the bottom wall from CB-LMR and (b) oil-film visualisation on the bottom wall, from the experiment of [38].



(a) (b)

Figure 16. Time-averaged streamlines in the $x-z$ plane at $y/H = 0.25$. (a) CB-LMR and (b) experiment of [38].

DOI: 10.1002/fld

Figure 17(a) presents the friction coefficient, c_f, that was computed in the LMR simulation, on the bottom wall and surface of the cube along the centreline of the domain ($z/h = 0.6$). DNS computed data points from [40] are included for comparison. The variable s signifies the length of the wall and cube surfaces in direct contact with the fluid along the centreline of the channel, as illustrated in Figure 17(c). Figure 17(a) reveals that the friction coefficient computed by the LMR LES generally agrees very well with the DNS, although some discrepancies are in evidence, most noticeably on the top wall of the body (i.e. $1 \leqslant s/k \leqslant 2$). There the peak friction velocity at the leading edge is overestimated, while over the rest of the cube surface, the friction velocity is underestimated by approximately 10%. This is to be expected, as in the LES, the cube is represented using the IBM, and the no-slip condition on its surface is only approximately prescribed; the friction coefficient on an immersed boundary would generally be expected to have a lower magnitude than that computed using a boundary-fitted grid, as was the case in the DNS.

A more quantitative evaluation of the LMR code's performance is possible with the help of Figures 18–20. Profiles of simulated time-averaged streamwise velocity at selected locations along the centreline of the computational domain ($z/h = 0.6$) are plotted against the experimental data (open circles) of Meinders *et al.* [38] in Figure 18. The five locations, labelled (a) to (e), are shown in Figure 17(b). Data from the LMR simulation (solid line) and the coarse uniform grid simulation (dashed line) are both plotted, while DNS data (small filled circles) from [40] are plotted in the lower half of the domain for reference. Note that the velocities are normalised with the bulk streamwise velocity, U_b. The agreement between the LES and the experiment and DNS is generally good. The comparison of profiles indicates that the uniform grid is capable of reproducing fairly well the flow

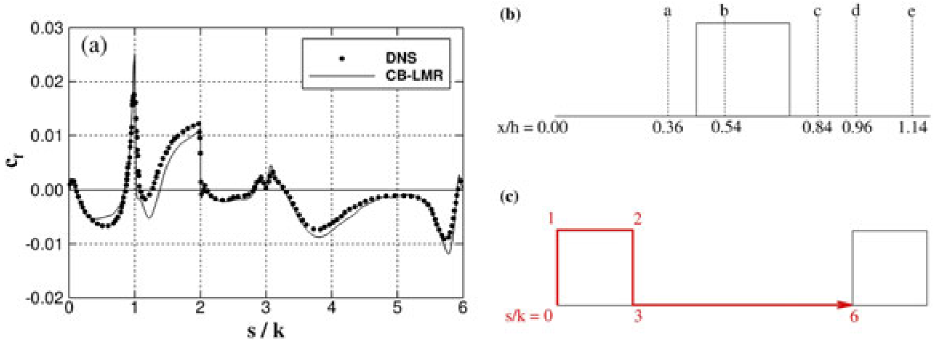


Figure 17. (a) Friction coefficient as a function of distance along the duct and cube surface ($z/h = 0.6$); (b) profile locations used in Figures 18–20; and (c) schema introducing the distance s, as used in plot (a).

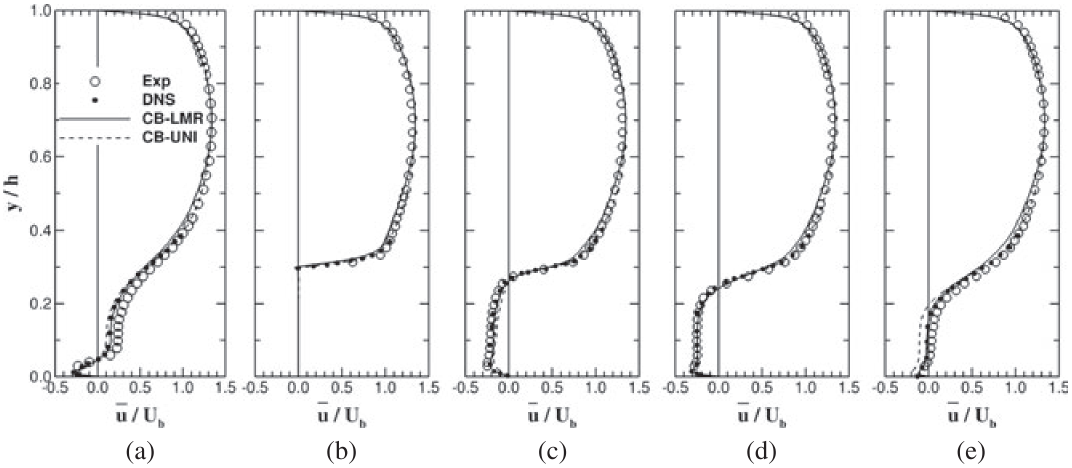


Figure 18. Profiles of normalised mean streamwise velocity along the centreline of the domain ($z/h = 0.6$). (a) $x/h = 0.36$; (b) $x/h = 0.54$; (c) $x/h = 0.84$; (d) $x/h = 0.96$; and (e) $x/h = 1.14$.

and good agreement with the experimental data is observed, although a clear improvement is found when employing LMR at locations $x/h = 0.36$ and $x/h = 1.14$ (i.e. Figures 18(a) and (e)) close to the bottom wall. This is because the resolution in the $x - z$ plane in this region is much finer in the LMR grid than in the uniform grid, resulting in more accurate computation of the shear layers around the cube. This in turn produces a more accurate computation of the recirculation length, with inherent improvements in the prediction of mean and fluctuating velocities.

Figures 19 and 20 present profiles of normalised normal stresses in the streamwise and spanwise directions, $\overline{u'u'}/U_b^2$ and $\overline{w'w'}/U_b^2$, respectively. The profile locations are once again those shown in Figure 17(b). As would be expected, significant peaks are observed near the top of the cube (see plot (b)), where the fluid experiences steep velocity gradients, separation and recirculation, while a smaller peak is observed at the top wall due to the presence of a classic smooth wall turbulent boundary layer there. Stresses are also significantly non-zero in the cube wake ($y/h \leqslant 0.3$), where the flow is characterised by unsteady recirculation between spatially periodic cubes. The LMR and coarse uniform grid LES both produce good agreement with the experimental data, but the LMR simulation achieves some notable improvements, particularly near the bottom wall and in the region immediately above the top of the cube. There are some rather significant discrepancies in the profiles of the spanwise Reynolds stress, $\overline{w'w'}/U_b^2$, in particular at locations $x/h = 0.36$ and $x/h = 1.14$

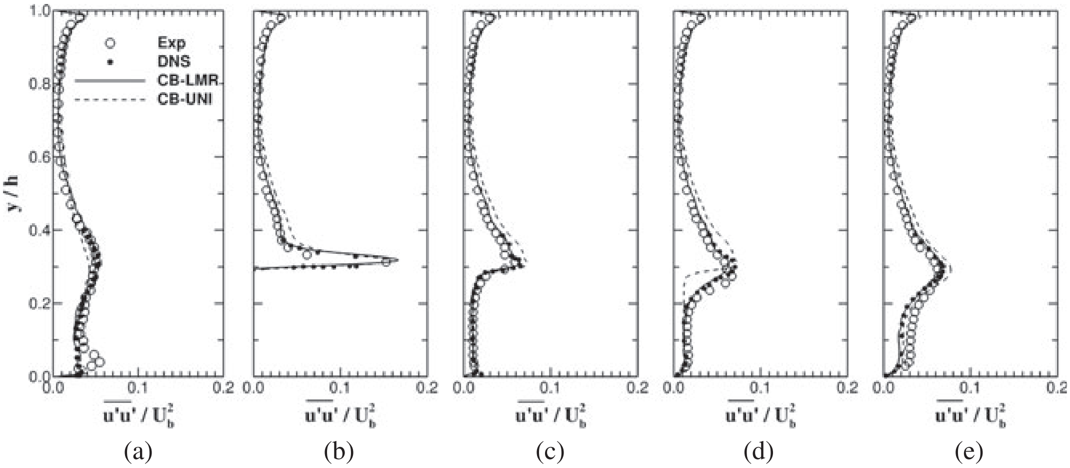


Figure 19. Profiles of normalised streamwise normal stress along the centreline of the domain ($z/h = 0.6$). (a) $x/h = 0.36$; (b) $x/h = 0.54$; (c) $x/h = 0.84$; (d) $x/h = 0.96$; and (e) $x/h = 1.14$.

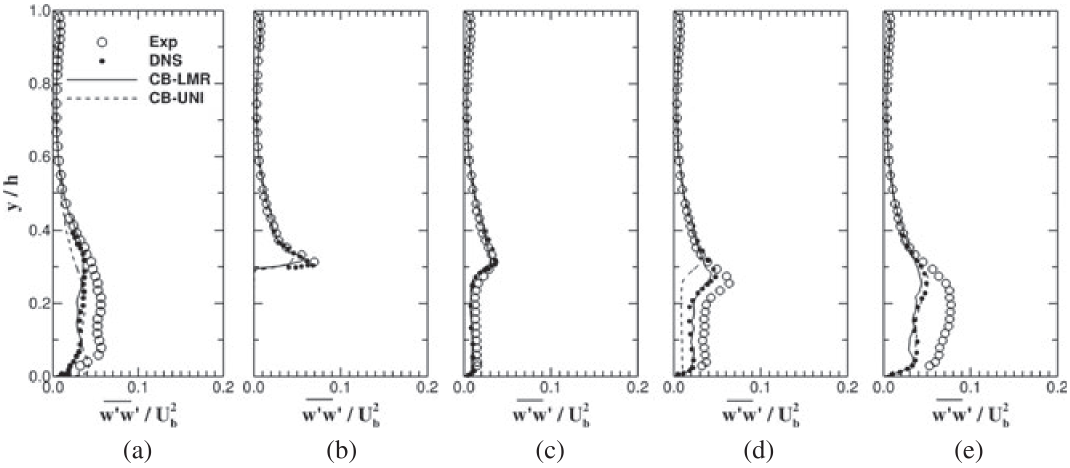


Figure 20. Profiles of normalised spanwise normal stress along the centreline of the domain ($z/h = 0.6$). (a) $x/h = 0.36$; (b) $x/h = 0.54$; (c) $x/h = 0.84$; (d) $x/h = 0.96$; (e) $x/h = 1.14$.

(i.e. Figure 20(a) and (e)); however, the same discrepancies were observed by [40] in their DNS (DNS data points are the small filled circles), and as can be seen, the LMR LES matches the DNS almost perfectly. Noteworthy is the fact that no discontinuities in the profiles are observed across transitions between zones of differing grid resolutions, providing further indication that the interpolation of ghost cell velocities is dealt with satisfactorily in the LMR approach.

5. CONCLUSION

A refined LMR approach on Cartesian grids for LESs has been presented and validated with data from well-known experiments and DNSs. LMR is promising to become an important ingredient for LESs of practical relevance because it can provide large computational savings when applied to high-Re flows with important localised features. The LMR approach calculates ghost cell pressures and velocities using higher-order interpolation schemes at the interface and throughout the domain to remedy mass and momentum conservation issues that occur at interfaces between nested grids of different resolutions. The LMR approach also adopts a more robust method to calculate fine grid ghost velocities normal to the grid, which are generally more challenging than tangential velocities. It is shown that the refined method is able to minimise numerical artefacts that have been shown to arise at the interfaces when second-order spatial discretisation is used. The study has incorporated careful validation, making comparisons with DNS and experimental data for two canonical test cases that together incorporate the most challenging features of turbulent flows, including wall-bounded turbulence, separating shear layers and unsteady vortex shedding. The agreement between the present method and DNS and physical experiments is convincing. Importantly, first-order and second-order turbulence statistical quantities are shown to transition smoothly across coarse-to-fine and fine-to coarse interfaces, giving continuous and accurate profiles in wall-normal and streamwise directions.

ACKNOWLEDGEMENTS

This work was supported by the BP/The Gulf of Mexico Research Initiative via the Gulf Integrated Spill Response Consortium (GISR) and by the UK Engineering and Physical Sciences Research Council (EPSRC; grant number EP/k041088/1). All simulations were performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University. All data created during this research are openly available from the Cardiff University data archive at http://dx.doi.org/10.17035/d.2016.0008219534.[‡]

REFERENCES

1. Lele SK. Compact Finite Difference Schemes with Spectral-like Resolution. *Journal of Computational Physics*. 1992; **103**(1): 16–42.
2. Kampanis NA, Ekaterinaris JA. A Staggered Grid, High-Order Accurate Method for the Incompressible Navier–Stokes Equations. *Journal of Computational Physics*. 2006; **215**(2): 589–613.
3. Morinishi Y, Lund TS, Vasilyev OV, Moin P. Fully Conservative Higher Order Finite Difference Schemes for Incompressible Flow. *Journal of Computational Physics*. 1998; **143**(1): 90–124.
4. Verma A, Mahes K. A Lagrangian subgrid-scale model with dynamic estimation of Lagrangian time scale for LES of complex flows. *Physics of Fluids*. 2010; **24**: 085101.
5. Kopera MA, Giraldo FX. Mass conservation of the unified continuous and discontinuous element-based Galerkin methods on dynamically adaptive grids with application to atmospheric simulations. *Journal of Computational Physics*. 2015; **297**: 90–103.
6. Panourgias K, Ekaterinaris JA. Three-Dimensional Discontinuous Galerkin h/p Adaptive Numerical Solutions for Compressible Flows. In *53rd AIAA Aerospace Sciences Meeting*, AIAA, January 2015; 2015–0576.
7. Thompson JF, Soni SK, Weatherill NP. *Handbook of Grid Generation*. CRC Press: London, 1998.
8. Peskin CS. Flow Patterns around Heart Valves - Numerical Method. *Journal of Computational Physics*. 1972; **10**(2): 252–271.
9. Peskin CS. Numerical-Analysis of Blood-Flow in Heart. *Journal of Computational Physics*. 1977; **25**(3): 220–252.
10. Tseng YH, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*. 2003; **192**(2): 593–623.

[‡] Correction added on 23 February 2016, after first online publication: information about the research data archive was added to the Acknowledgements.

11. Berger MJ, Oliger J. Adaptive Mesh Refinement for Hyperbolic Partial Differential Problems. *Journal of Computational Physics*. 1984; **53**(3): 484–512.
12. Berger MJ, Colella P. Local Adaptive Mesh Refinement for Shock Hydrodynamics. *Journal of Computational Physics*. 1989; **82**(1): 64–84.
13. Howell LH, Bell JB. An Adaptive Mesh Projection Method for Viscous Incompressible Flow. *SIAM Journal of Scientific Computing*. 1997; **18**(4): 996–1013.
14. Minion ML. A Projection Method for Locally Refined Grids. *Journal of Computational Physics*. 1996; **127**(1): 158–178.
15. Zuzio D, Estivalezes JL. An Efficient Block Parallel AMR Method for Two Phase Interfacial Flow Simulations. *Computers and Fluids*. 2011; **44**(1): 339–357.
16. Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML. A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier–Stokes Equations. *Journal of Computational Physics*. 1998; **142**(1): 1–46.
17. Martin DF, Colella P. A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations. *Journal of Computational Physics*. 2000; **163**(2): 271–312.
18. Martin DF, Colella P, Graves D. A Cell-Centered Adaptive Projection Method for the Incompressible Navier–Stokes Equations in Three Dimensions. *Journal of Computational Physics*. 2008; **227**(3): 1863–1886.
19. Popinet S. Gerris: A Tree-Based Adaptive Solver for the Incompressible Euler Equations in Complex Geometries. *Journal of Computational Physics*. 2003; **190**(2): 572–600.
20. Peng Y, Mittal R, Sau A, Hwang RR. Nested Cartesian Grid Method in Incompressible Viscous Fluid Flow. *Journal of Computational Physics*. 2010; **229**(19): 7072–7101.
21. Kravchenko AG, Moin P, Moser R. Zonal Embedded Grids for Numerical Simulations of Wall-Bounded Turbulent Flows. *Journal of Computational Physics*. 1996; **127**(2): 412–423.
22. Piomelli U, Kang S, Ham F, Iaccarino G. Effect of discontinuous filter width in Large-Eddy Simulations of Plane Channel Flow. In *Studying turbulence using numerical databases XI*. Stanford University, 2006; 151–162.
23. Pantano C, Deiterding R, Hill DJ, Pullin DI. A Low Numerical Dissipation Patch-Based Adaptive Mesh Refinement Method for Large-Eddy Simulation of Compressible Flows. *Journal of Computational Physics*. 2007; **221**(1): 63–87.
24. Vanella M, Piomelli U, Balaras E. Effect of Grid Discontinuities on Large-Eddy Simulation Statistics and Flow Fields. *Journal of Turbulence*. 2008; **9**(32): 1–23.
25. Bomminayuni S, Stoesser T. Turbulence Statistics in an Open-Channel Flow over a Rough Bed. *Journal of Hydraulic Engineering*. 2011; **137**(11): 1347–1358.
26. Kim SJ, Stoesser T. Closure Modeling and Direct Simulation of Vegetation Drag in Flow through Emergent Vegetation. *Water Resources Research*. 2011; **10**: W10511.
27. Kara S, Stoesser T, Sturm TW. Turbulence statistics in compound channels with deep and shallow overbank flows. *Journal of Hydraulic Research*. 2012; **50**(5): 482–493.
28. Papanicolaou AN, Kramer CM, Tsakiris AG, Stoesser T, Bomminayuni S, Chen Z. Effects of a fully submerged boulder within a boulder array on the mean and turbulent flow fields: Implications to bedload transport. *Acta Geophysica*. 2012; **60**(6): 1502–1546.
29. Kim D, Kim JH, Stoesser T. The Effect of Baffle Spacing on Hydrodynamics and Solute Transport in Serpentine Contact Tanks. *Journal of Hydraulic Research*. 2013; **51**(5): 558–568.
30. Kara S, Stoesser T, Sturm TW, Mulahasan S. Flow dynamics through a submerged bridge opening with overtopping. *Journal of Hydraulic Research*. 2015; **53**(2): 186–195.
31. Rodi W, Constantinescu G, Stoesser T. *Large-Eddy Simulation in Hydraulics*. CRC Press: London, 2013.
32. Stoesser T. Large-eddy simulation in hydraulics: Quo Vadis? *Journal of Hydraulic Research*. 2014; **52**(4): 441–452.
33. Nicoud F, Ducros F. Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor. *Flow, Turbulence and Combustion*. 1999; **62**(3): 183–200.
34. Kim J, Moin P. Application of a Fractional-Step Method to Incompressible Navier–Stokes Equations. *Journal of Computational Physics*. 1985; **59**(2): 308–323.
35. Ferziger JH, Peri M. *Computational Methods for Fluid Dynamics*. Springer Verlag: Berlin Heidelberg, New York, 2002.
36. Cevheri M, Stoesser T. Large-eddy simulation of a jet in cross-flow using local mesh refinement. *Progress in Computational Fluid Dynamics*. 2016. (revised).
37. Thorne DT. Multigrid with Cache Optimizations on Adaptive Mesh Refinement Hierarchies. *Ph.D. Thesis*, Kentucky University, Lexington, Kentucky, 2003.
38. Meinders ER, Hanjalic K. Vortex structure and heat transfer in turbulent flow over a wall-mounted matrix of cubes. *International Journal of Heat and Fluid Flow*. 1999; **20**(3): 255–267.
39. Moser RD, Kim J, Mansour NN. Direct Numerical Simulation of Turbulent Channel Flow up to $Re_\tau = 590$. *Physics of Fluids*. 1999; **11**(4): 943–945.
40. Hellsten A, Rautaheimo P. *Proceedings of the 8th ERCOFTAC/IAHR/COST Workshop on Refined Turbulence Modelling*. Helsinki University of Technology: Espoo, Finland, 1999.
41. Stoesser T, Mathey F, Frohlich J, Rodi W. LES of flow over multiple cubes. In *ERCOFTAC Bulletin*, ERCOFTAC, London, UK: 2003, 15–19.
42. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*. 2000; **161**(1): 35–60.
43. Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*. 2005; **209**(2): 448–476.