Journal of Computational Physics 301 (2015) 308–321

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A multiple-resolution strategy for Direct Numerical Simulation of scalar turbulence

R. Ostilla-Monico [a,*,1], Yantao Yang [a,1], E.P. van der Poel [a], D. Lohse [a], R. Verzicco [a,b]

[a] Physics of Fluids, MESA+ Institute, University of Twente, Enschede, The Netherlands
[b] Department of Industrial Engineering, Università di Roma "Tor Vergata", Roma, Italy

ARTICLE INFO

Article history:
Received 7 February 2015
Received in revised form 7 July 2015
Accepted 14 August 2015
Available online 28 August 2015

Keywords:
Multiple resolution
Direct Numerical Simulation
Scalar turbulence

ABSTRACT

In this paper a numerical procedure to simulate low diffusivity scalar turbulence is presented. The method consists of using a grid for the advected scalar with a higher spatial resolution than that of the momentum. The latter usually requires a less refined mesh and integrating both fields on a single grid tailored to the most demanding variable produces an unnecessary computational overhead. A multiple resolution approach is used also in the time integration in order to maintain the stability of the scalars on the finer grid. The method is the more advantageous the less diffusive the scalar is with respect to momentum, therefore it is particularly well suited for large Prandtl or Schmidt number flows. However, even in the case of equal diffusivities the present procedure gives CPU time and memory occupation savings, due to the increased gradients and more intermittent behaviour of the scalars when compared to momentum.

1. Introduction

Countless phenomena in Nature and technology involve one or more scalar fields that are advected and diffused by a turbulent flow. The dilution of pollution in the atmosphere [1], the transport of nutrients in oceans [2], the cooling or heating of devices [3] and the buoyancy-driven currents generated by natural- [4,5] and double-diffusive [6,7] convection are just few examples among many. Scalars have also be used to capture interfaces in multiphase flows [8] or to determine the dynamical properties of turbulent flows [9].

Scalars can be classified as either passive or active, depending on whether they couple back to the velocity field or not. Many studies of simulations of passive scalars in incompressible homogeneous isotropic turbulence (HIT) have been performed during the years [10–15]. For a comprehensive overview, we refer the reader to the review by Warhaft [16] and the references therein. For active scalars, especially in the context of natural convection, numerical simulations have also been proven to be very helpful in unravelling the complex physics behind these phenomena [17] even if the calculations have shown to be more demanding than expected, taking up to millions of CPU hours in recent studies [18,19].

The common understanding of the problem is that in three-dimensional turbulent flows, there is a cascade from the largest towards the small spatial scales up to a lower limit that is determined by the diffusivity. As each field has its own diffusivity, these scales can have different magnitudes. In direct numerical simulation (DNS) the mesh size must be smaller

* Corresponding author.
E-mail address: r.ostillamonico@utwente.nl (R. Ostilla-Monico).
[1] Both these authors have made an equal contribution to the manuscript.

http://dx.doi.org/10.1016/j.jcp.2015.08.031
0021-9991/

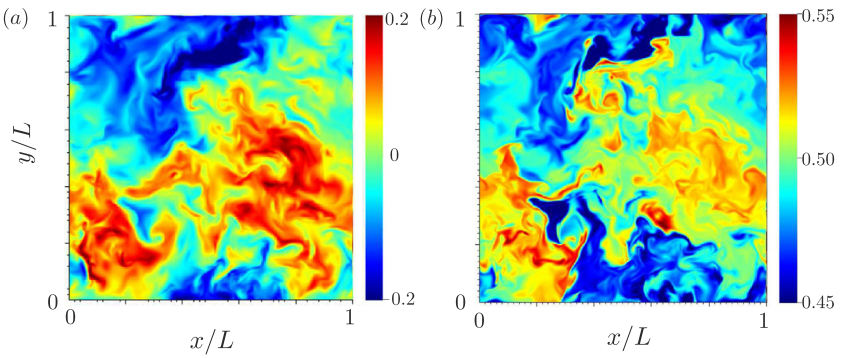

Fig. 1. A horizontal plane halfway between the plates for a Rayleigh–Bénard simulation in a Cartesian geometry at $Ra = 10^{10}$ and Prandtl number $Pr = 1$. (a) Vertical velocity, red indicating rising fluid while blue indicates falling fluid, (b) temperature, red indicating hot fluid and blue indicating cold fluid. Even though the Prandtl number is one, much sharper gradients can be seen in the right panel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

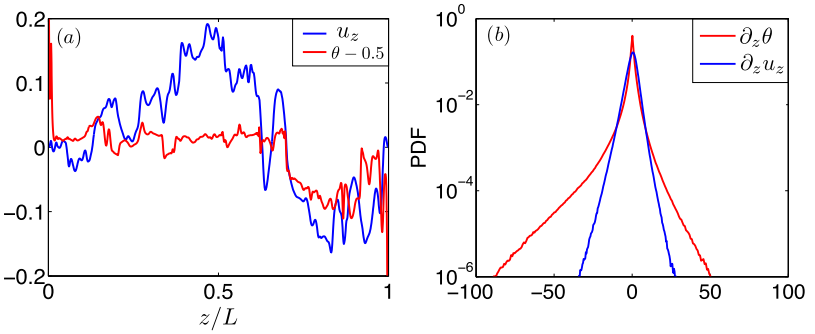

Fig. 2. (a) Instantaneous θ and u_z profiles as a function of the vertical coordinate z/L for $Ra = 10^{10}$, $Pr = 1$ RB simulation. (b) PDF of $\partial_z\theta$ and $\partial_z u_z$ for the same simulation in the bulk.

than the smallest among them: This requirement quickly renders DNS infeasible. Denoting as η_K the smallest (Kolmogorov) scale of the momentum field, we can calculate the analogous quantity for a scalar field S as $\eta_B = \eta_K/Sc^{1/2}$, also called the Batchelor scale, with $Sc = \nu/\kappa_S$ the Schmidt number defined as the ratio of the kinematic viscosity ν and the scalar diffusivity κ_S, respectively. In some cases, like sugar in water, the Schmidt number exceeds 10^3 resulting in a Batchelor scale of $\eta_B \simeq \eta_K/30$. With equal grid resolutions for the scalar and the momentum fields, this entails that the momentum field is overresolved by a factor of approximately 30 in each spatial direction. The problem is exacerbated by the fact that a scalar is described by only a single quantity, while momentum is a vector field satisfying the incompressibility condition or other related constraints. This implies that the solution of the momentum alone generally takes an order of 90% of the total CPU time of a simulation and therefore resolving it on an unnecessary fine mesh is not desirable.

The above scenario, essentially derived from dimensional analysis, does not give the complete picture since it does not account for the structure of the equations. In fact, the naïve comparison between the Kolmogorov and Batchelor scales suggests that for $Sc \approx 1$, $\eta_K \simeq \eta_B$ although in practice the resolution requirements for the momentum and the scalar fields are not the same. Visual evidence of the latter statement can be obtained from the instantaneous snapshots of Fig. 1 showing horizontal cross-sections of temperature and vertical velocity in a thermally driven turbulent flow, the Rayleigh–Bénard (RB) problem, i.e. the flow between two parallel plates heated from above and cooled from below. In RB flow, the fluid hotter than the average temperature (0.5 in nondimensional units) generates upward buoyancy and therefore positive vertical velocity (and vice versa). Although the two fields are very well correlated on the large scales, the sharp fronts of the scalar field do not have an analogous counterpart in the momentum distribution and this results in a different resolution requirement for scalar and momentum fields.

For a fully resolved DNS, the momentum gradients must be adequately captured so that the dissipative scale (and thus vorticity) is adequately resolved. Analogously, the scalar (temperature) gradients must be correctly captured so that the diffusive scale (and thus the scalar variance) is adequately resolved. We quantify the difference in gradients between scalars and momentum in Fig. 2(a) by showing instantaneous temperature θ and vertical velocity u_z profiles across a vertical line from a doubly periodic Rayleigh–Bénard simulation: much steeper gradients can be seen in the temperature (scalar) field. These steep gradients are smoother in the vertical velocity and this lowers the resolution requirements of momentum with respect to those of scalars. This observation is further corroborated by Fig. 2(b) showing the probability density functions of $\partial_z\theta$ and $\partial_z u_z$ computed in the bulk of the flow without the boundary layers. Extreme gradients can be seen to be more likely for θ thus evidencing a more intermittent behaviour. This behaviour has been extensively studied in experiments

(cf. Fig. 2 in Ref. [16]). It is a well established fact that the intermittency corrections to the structure function exponents are much larger for scalars than for velocity [20,21]. Indeed, these fronts become much sharper with increasing Reynolds number, and thus increasing small-scale intermittency [22,10,12,13,16,23].

This consideration motivates the present paper that describes a strategy for efficiently simulating scalar driven turbulent flows with different spatial resolutions for momentum and scalar fields. A problem related to the non-local equations for momentum is their high computational cost and the detrimental implications on the parallel performance. Therefore, the possibility of using different meshes for momentum and scalars opens the door to very large gains in performance, not only by reducing the amount of operations, but also the communication of data among processors and the total memory usage. In this context, there already exist similar approaches, which however, have focused on passive scalars in HIT. Gotoh et al. [24] already used an approach using a multiple resolution strategy to simulate a scalar in decaying HIT. They used a pseudospectral method to solve the incompressible Navier–Stokes equations on a coarse mesh, and combined compact finite differences for the scalar advection–diffusion equation on a finer mesh for decaying HIT turbulence. Reductions of computational time of 25% for $Sc = 1$ and 74% for $Sc = 50$ were reported. Recently, Lagaert et al. [25] used a hybrid spectral-particle approach to simulate high Sc flows, where the incompressible Navier–Stokes equations are solved pseudospectrally, while a particle method is used to solve for the scalar field on the same test cases as Ref. [24], providing even greater speedups. Another approach worth mentioning by Verma and Blanquart [26] uses a Large Eddy Simulation (LES)-type filter in the "viscous-convective" scale range, i.e. the scale range λ_{vc} for which $\eta_K << \lambda_{vc} << \eta_B$, to simulate high Schmidt number flows, resolved in velocity but filtered for the scalar. However, it is unclear how this approach will work with active scalars.

All previous methods focus on the simulation of passive scalars. In natural convection however, the flow is driven by temperature, so in this case we must simulate an *active* scalar. This motivates the present study where we will detail the application of a multiple resolution strategy for active scalars on non-uniform grids needed to properly capture the dynamics in the near-wall region. This method will be mainly applied to Rayleigh–Bénard (RB) convection, the flow of a fluid vertically confined by a top cold and a bottom hot plate. We will simulate a "rectangular" domain, periodic in both horizontal directions, so only the vertical (wall-bounded) direction will be non-homogeneous. The horizontal periodicity length will be taken as a half of the height, so the aspect ratio Γ is fixed $\Gamma = 1/2$. In the RB problem, the flow is driven by the temperature (scalar) field whose diffusivity can be changed by the Prandtl number Pr (the temperature Schmidt number). RB is a particularly suitable example for the present application since analytical exact relations are available for this problem that can be used to check the correctness of the numerical results. It is worth mentioning that in RB convection the forcing comes from the heated and cooled surfaces where viscous and thermal boundary layers develop. Since they become thinner as the forcing strengthens, the resolution requirements in these boundary layers become more stringent than in the bulk. This implies a strong non-uniformity in the spatial discretization, not accounted for in the previously mentioned multiple-resolution methods. An extensive analysis of the problem can be found in Ref. [27] where all the details, estimates and guidelines for numerical simulations are given. Here, it suffices to mention that a non-uniform mesh is required in the wall normal direction such to cluster the nodes within the boundary layers. Nevertheless, even if the grid spacing at the wall is much finer than that in the bulk, the volume of fluid within these layers is at most a few percent of the total and the nodes allocated there are only a small fraction of the whole mesh.

We will additionally show another numerical example. Namely, double diffusive convection (DDC), in which the flow is driven by two scalars with very different diffusivities and with opposite, stabilizing and destabilizing, effects on the flow. In this case the multiple resolution strategy is even more advantageous and it allows for the simulation of flow regimes that otherwise would be out of reach.

The organization of the paper is the following. In the next section we describe the governing equations and the numerical method. Section 3 quantifies the differences of momentum and scalar gradients and presents some analytical exact relations for RB. The section closes with the results of reference simulations obtained on a standard single grid. In Section 4 the multiple resolution strategy and numerical details are explained. Finally, Section 5 discusses the results and the computational performance of the method for RB flow and double diffusive convection. Closing remarks are reported in Section 6.

2. Governing equations and numerical method

The incompressible Navier–Stokes equations with the Boussinesq approximation for thermal convection, in nondimensional form, read:

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\partial_i p + \sqrt{\frac{Pr}{Ra}} \nabla^2 \mathbf{u} + \theta \mathbf{e}_z, \tag{2}$$

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = \sqrt{\frac{1}{Ra\,Pr}} \nabla^2 \theta, \tag{3}$$

where $\mathbf{u}$ is the velocity, p is the pressure, θ is the temperature (rescaled such that it is one at the hot plate and zero at the cold plate) and $\mathbf{e}_z$ is the unitary vector in the anti-parallel direction to gravity, which is also the plate-normal direction. The

Rayleigh number Ra is the non-dimensional temperature difference defined as $Ra = g\beta_T(T_h - T_c)L^3/\kappa_T\nu$, where ν is the kinematic viscosity, β_T is the isobaric thermal expansion coefficient and κ_T is the thermal diffusivity of the fluid, g is the gravity, L is the distance and T_h and T_c are the temperature at the hot and cold plates, respectively. The Prandtl number Pr, which is the temperature analogue of the Schmidt number, is defined as $Pr = \nu/\kappa_T$.

The integration of the above system is performed by a fractional timestepping [28] with the modifications proposed in Refs. [29,30]. In short, a provisional velocity $\hat{u}_i$ is computed from the previous field u_i^n using the old pressure p^n

$$\frac{\hat{u}_i - u_i^n}{\Delta t} = -\partial_i p^n - N_i^{n+1/2} + D_i^{n+1/2}, \tag{4}$$

$N_i^{n+1/2}$ contains the non-linear terms and the scalar forcing while $D_i^{n+1/2}$ has the viscous terms: The former are computed explicitly in time, the latter implicitly. The flow incompressibility is then enforced by a pressure correction that takes the form of a Poisson equation $\nabla^2\phi = \partial_i\hat{u}_i$ whose solution is the most computationally demanding step, especially on non-uniform grids. In addition, the Poisson equation is non-local and this has consequences on code parallelization, requiring the largest amount of communication. Once the scalar ϕ is obtained, the velocity $\hat{u}_i$ is projected onto the solenoidal field u_i^{n+1} and the new pressure p^{n+1} can be computed. The advancement of the scalar is performed directly through

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = -M^{n+1/2} + V^{n+1/2}, \tag{5}$$

where, again, $M^{n+1/2}$ contains the explicit non-linear terms and $V^{n+1/2}$ the implicit diffusive terms.

All the variables are discretized by central second-order finite-differences on a staggered grid and the time advancement of the solution is obtained by a low-storage third-order Runge–Kutta scheme. Further details can be found in Ref. [29].

3. Heat transfer in RB

In RB flow, one of the interesting quantities is the heat flux Q transferred from one plate to the other. In non-dimensional form this is the Nusselt number, $Nu = Q/\kappa_T(T_h - T_c)L^{-1}$. This is not only interesting from a physical point of view, but also as a monitoring variable since it has been observed [31] that when the separately calculated Nusselt numbers converge and are grid independent, at least all quantities up to second order statistics are properly resolved.

There are several ways to calculate Nu, either by measuring the convective heat transport in the system

$$Nu(z) = \sqrt{Ra\,Pr}\langle u_z\theta\rangle_{A,t} + 1, \tag{6}$$

or by using the exact relationships derived from global balances [32] of kinetic energy,

$$\epsilon_\nu = \nu U_f^2 L^{-2}\langle[\partial_i u_j]^2\rangle_{V,t} = \nu^3 L^{-4}(Nu - 1)Ra\,Pr^{-2}, \tag{7}$$

and thermal energy,

$$\epsilon_\theta = \kappa_T(T_h - T_c)^2 L^{-2}\langle[\partial_i\theta]^2\rangle_{V,t} = \kappa_T(T_h - T_c)^2 L^{-2}Nu. \tag{8}$$

Here the subscripts t, A and V denote, respectively, averages in time, horizontal homogeneous planes and the whole fluid volume, and U_f the free-fall velocity $U_f = \sqrt{\beta g(T_h - T_c)L}$. Although equation (6) depends on z, once the equilibrium is reached its value becomes constant. This condition is requested to assess the statistical convergence of the results.

From here on, we denote Nu with a subscript, either $Nu_{u_z\theta}$, Nu_{ϵ_ν} or Nu_{ϵ_θ} which specifies the particular equation, i.e. (6)–(8) respectively, with which Nu is calculated. In addition, we also denote the Nusselt number calculated by the temperature gradient at the wall as Nu_{θ_w}. All definitions are equivalent analytically, but they involve gradients of the variables that, when calculated numerically, can deviate from each other if the simulations do not have enough spatial resolution to capture the smallest flow scales: the diffusive/scalar variance scale in Nu_{ϵ_θ} and the dissipative/vorticity scale Nu_{ϵ_ν}: Their comparison can thus be used as a test for the adequacy of the mesh [31]. We note that these relationships are not only valid in the Cartesian geometry considered here, with horizontally periodic boundary conditions, but in any geometry in which the horizontal sidewalls are impermeable and adiabatic (or periodic). For example, this criterium was adopted by Stevens et al. [31], where the geometry is a cylinder with impermeable adiabatic sidewalls. According to their results, summarized in Fig. 3, it is evident that equation (8) shows the slowest grid convergence, and this is the reason for adopting the above mentioned criterium.

This criterium will be now applied for the Cartesian geometry considered here. Figs. 4(a) and 4(b) show the ratios $Nu_{\epsilon_\theta}/Nu_{u_z\theta}$ and $Nu_{\epsilon_\nu}/Nu_{u_z\theta}$ for RB simulations performed on the same grid for momentum and temperature (scalar) at $Ra = 10^9$ and $Pr = 1$ or $Pr = 10$. Resolutions between $96^2 \times 192$ and $384^2 \times 768$ were used, the larger number corresponding to the vertical (wall-bounded) direction. An aspect ratio of $\Gamma = 1/2$ was used in both directions. Points were clustered near the wall using a Chebychev-like distribution according to the prescriptions of [27]. See Table A.1 for the raw numerical results. The Kolmogorov scale is computed from the kinetic energy dissipation rate using $\eta_K/L = (\nu^3/\epsilon_\nu)^{1/4}$ and the Batchelor scale $\eta_B/L = \eta_K/LPr^{-1/2}$. As mentioned above, the various expressions for Nu should be equivalent, their ratios however,

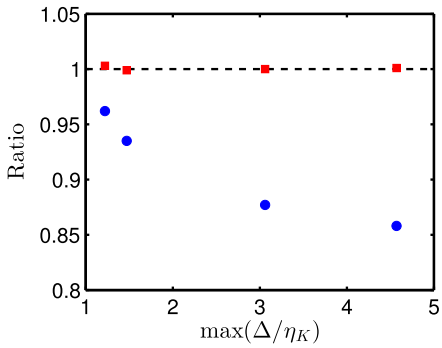

Fig. 3. Ratio between different ways of calculating Nu against grid size for a cylindrical RB simulation with $Ra = 10^9$ and $Pr = 0.7$, taken from Ref. [31]. Red squares are $Nu_{\epsilon_v}/Nu_{u_z\theta}$, blue circles are $Nu_{\epsilon_\theta}/Nu_{u_z\theta}$.

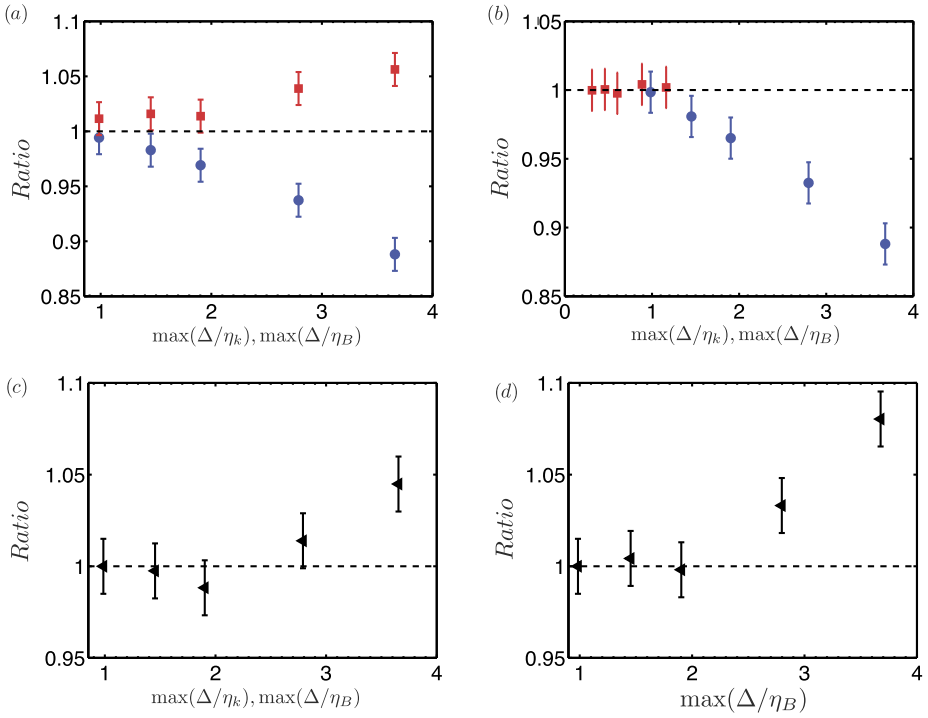

Fig. 4. (a) Ratio between different ways of calculating Nu against grid size for a rectangular RB simulation with $Ra = 10^9$ and $Pr = 1$. Red squares are $Nu_{\epsilon_v}/Nu_{u_z\theta}$, blue circles are $Nu_{\epsilon_\theta}/Nu_{u_z\theta}$. (b) The same as (a) for $Pr = 10$. Nu_{ϵ_θ} is plotted against $\max(\Delta/\eta_B)$ while Nu_{ϵ_v} is plotted against $\max(\Delta/\eta_B)$. For $Pr = 1$, $\eta_B \equiv \eta_K$. (c) Convergence to an asymptotic value of $Nu_{u_z\theta}$ for $Ra = 10^9$ and $Pr = 1$. (d) The same as (c) for $Pr = 10$. (See Table A.1 for the raw numerical results.)

approach the unity limit only when the normalized mesh sizes Δ/η_K and Δ/η_B decrease and they do not converge at the same rate. In particular it can be noted that $Nu_{\epsilon_v}/Nu_{u_z\theta}$ tends to unity for larger grid spacings than $Nu_{\epsilon_\theta}/Nu_{u_z\theta}$ even for $Pr = 1$ and this corroborates our hypothesis that a finer resolution is needed for the scalar than for momentum. Using an identical mesh to spatially discretize both momentum and the scalars therefore produces an overhead in computational resources that is redundant.

At the highest resolution, all the definitions converge to the same value (within an uncertainty of 2–3%), therefore we will refer to it as Nu_{ref} without specifying the particular expression and Figs. 4(c) and 4(d) show the convergence of $Nu_{u_z\theta}$ to this asymptotic value.

4. The multiple resolutions strategy

4.1. Multiple resolution strategy in space

In this subsection we present a method to decouple the spatial discretization of the scalars and the momentum, which allows for large computational savings. This is achieved by refining every cell of a base mesh $\mathcal{M}^i$ times in each i-th

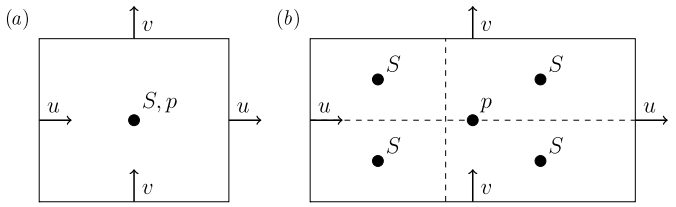

Fig. 5. Location of pressure, active scalar and velocities of a 2D simulation cell. The third dimension (z) is omitted for clarity. As on an ordinary staggered scheme, the velocity vectors are placed on the borders of the cell and pressure and scalars are placed in the cell center.

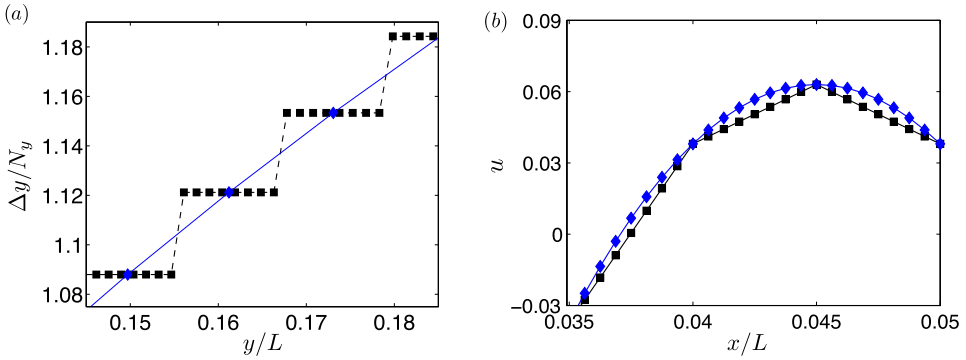

Fig. 6. (a) Normalized metric $\Delta x/N_x$, where N_x is the amount of grid points for a Chebychev-type grid clustering for a refined-mesh generated ($\mathcal{M}=8$) from a base mesh by splitting the base cells (squares), and the base mesh generated from the refined mesh (diamonds). The first method causes a staircase-like metric, which leads to spurious oscillations. (b) Comparison on interpolated velocity v from a base mesh to the refined mesh ($\mathcal{M}=8$) using linear interpolation (squares) and cubic Hermite splines (diamonds). Both interpolations coincide at the base mesh points. The direction of interpolation is uniformly discretized, showing that this problem with linear interpolation is present in all directions.

direction. A simplified two-dimensional sketch of this procedure is shown in Fig. 5. On the left, the location of the scalars and velocities in the standard single mesh is shown for a staggered arrangement. There, the velocity components are at the centers of the cell faces, while the pressure and the scalar are discretized at the center of the cell volume. The right panel shows a case with velocity and pressure on the base grid, and a doubly refined ($\mathcal{M}^x=\mathcal{M}^y=2$) mesh for the scalar, which is temperature in the RB case.

The method works as follows. We first generate the refined mesh over which the scalar field is discretized and then a coarser mesh is obtained by taking only one out of $\mathcal{M}^i$ nodes in the i-th direction. Note that when the mesh is uniform in space this is equivalent to start from the coarse cells and split them into $\mathcal{M}^i$ identical parts. For a non-uniform mesh, however additional care must be taken. This naive splitting would result in a staircase distribution of the metrics for the fine grid, which would have constant coefficients within each coarse element and with jumps across its boundaries. These discontinuities in the metric would locally decrease at the border elements the accuracy of the discretization to first order and also introduce spurious oscillations in the resolved fields. The difference between the resulting metric from the naïve method and the used method is highlighted in Fig. 6(a) for a mesh obtained by a Chebychev-like distribution with 96 base nodes and a refinement factor of eight. We note that in this section, most numerical examples are obtained using the same $\mathcal{M}^i$ in every direction therefore, unless otherwise specified, from here on we will use only $\mathcal{M}$ to indicate the isotropic refinement factor without specifying the direction. However, this is not required for the method to work and the same procedure can be applied to refinement levels different in each direction depending on the particular flow physics. An example of anisotropic refinement factors will be shown in Section 5.2.

In order to advect the scalar, a velocity field has to be projected from the base mesh onto the refined grid in a staggered arrangement with respect to the scalar. A first straightforward possibility consists of using a tri-cubic Hermite spline interpolation, with a stencil of four points in every direction, for a total of sixty-four points in three dimensions. This is, according to our numerical tests, the minimum required. At the top and bottom boundaries, one-sided interpolation is used, which in principle is less accurate, but as it is performed on a much finer grid, the amplitude of the error is much smaller than in the bulk. The accuracy of Hermitian interpolation has proven to be sufficient in our turbulent flows, and it is comparable to that of B-splines [33]. Hermitian interpolation, however, is preferred in this method as B-splines are much more computationally expensive.

Preliminary simulations have shown that a linear interpolation using a two point stencil, though computationally cheaper, is not adequate since it results in a spatially interpolated velocity field which has equal gradients for all the refined points inside every base cell, and discontinuities at the base cell boundaries (Fig. 6(b)). This lack of homogeneity results in spurious oscillations in the scalar field, in particular around local maxima and minima of velocity. An example of these oscillations is

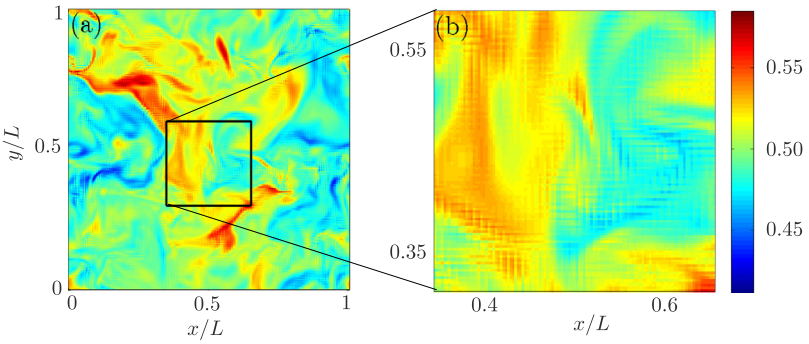

Fig. 7. (a) Pseudocolour plot of temperature at the mid-height when using a refinement of $\mathcal{M} = 3$ and linear interpolation for the velocities. Spurious oscillations every three points can be seen all over the domain. (b) Zoom-in of the region inside the black square of (a). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

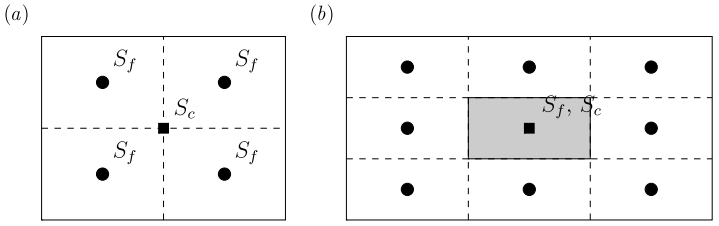

Fig. 8. When the refinement factors are even, as in (a), the finely-discretized scalar does not directly correspond to the coarsely-discretized scalar. When the refinement factors are all odd as in (b), one of the finely-discretized scalars coincides with the coarsely-discretized scalar. However, the value of S at this point only represents the value of the scalar in the sub-cell (shaded area) and not the entire coarse cell.

shown in Fig. 7 with the checkboard pattern given by the footprint of the base mesh. We note that this problem is not due to the non-uniformity of the mesh, and it would be present for uniform meshes if linear interpolation were to be used.

The spatially interpolated velocity can then be used to advance equation (3), and compute the values of the scalars at the new time. If a finely-discretized scalar couples back to the momentum field, like in the case of RB flow, a spatial filter must be applied to calculate a "coarsened" scalar, as information from the multiple sub-cells must be mapped on to the coarser momentum mesh. In this case, an averaging using equal weights within each refined cell is used. This averaged scalar is then used in equation (2) to advance momentum and pressure.

This procedure is needed mainly due to two reasons. First, the position of the scalar in the fine grid only coincides with the position of the scalar on a coarse grid if odd refinement factors are used in all directions. This is shown graphically in Fig. 8. Therefore, some sort of interpolation is needed to map the scalars on the fine grid onto a value for the scalar on the coarse grid. Second, even if all refinement factors are odd, and there exists a coincident position for the scalar in the coarse and fine mesh, the scalar in the fine mesh only represents a fraction of the total volume of the coarse cell, and thus averaging the value across the cells of the fine mesh contained in the coarse cell gives a conserved scalar. In addition, for non-uniform meshes, even if the refinement factors are all odd, there is not a coincident position and an interpolation or filtering is always needed.

Indeed, by using a multiple resolutions strategy for the scalar, we are missing the forcing of the scalar at sub-grid wavenumbers. However, if the velocity field is resolved correctly, this sub-grid forcing which we are missing is at sub-Kolmogorov wavenumbers, which are damped by viscosity. This again highlights the need to have a sufficient mesh for the velocity field, otherwise this missing forcing will result in a different (and wrong) physics for the velocity.

We stress that the interpolation of a velocity field between two different grids is a very dangerous operation since its effect is equivalent to that of a low-pass filter, which usually leads to loss of energy and generation or destruction of information. This is catastrophic in the DNS of turbulent flows where the dynamics are based on the energy cascade through the scales. If the velocity is underresolved, then the unresolved sub-grid scales are a relevant part of the physics and contain a significant energy. This is lost, and thus the energy cascade is seriously disturbed. It has even more serious consequences in a RB flow where the balance between thermal (potential) and kinetic energy determines the heat transfer. Nevertheless, if the base mesh already captures the smallest momentum structures, the field is smooth already at the scale of the base cell, and the subgrid scales have very little energy. Thus, an interpolation kernel that is continuous enough does not alter the energy content or the velocity gradients of the field. This has been directly verified with the DNS data by computing the relation (7) and the various meshes and comparing the results.

As mentioned above, interpolating the velocity field from the base to the refined mesh is the most immediate although it must be noted that it results in a *non*-solenoidal field. We wish to stress that this is not due to the tricubic Hermitian

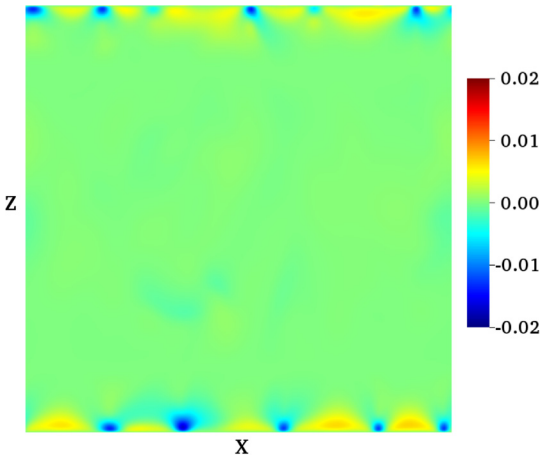

Fig. 9. Pseudocolour plot of the local divergence of the tricubically-interpolated velocity field $\nabla \cdot \tilde{\mathbf{u}}$. The local residual can be seen to be very large near the walls, where the grid is highly non-uniform. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

interpolation since, owing to the staggered arrangement of the velocity components, even a linearly interpolated velocity would be non-solenoidal on the fine grid.

Indeed one may also just use the tricubic interpolation "as-is". Our simulations in this section indicate that the global responses and turbulent statistics show great agreement with those by using a single refined grid, and this locally non-solenoidal field has not resulted in apparent problems. The residual divergence is in fact very small ($\sim \mathcal{O}(10^{-3})$) for most part of the flow domain as long as the base mesh is in the DNS range for the momentum. Higher residual divergence may be observed in some very localized regions which are usually close to the wall boundary, such as the places where the scalar plumes develop and the grid is highly non-uniform. This is shown in Fig. 9, which shows a pseudocolour of the local divergence of the tricubically-interpolated field in a vertical cut. Notice that at every time step the refined velocity field is interpolated from a solenoidal velocity field on the coarse grid, and the tricubic interpolation preserves the property of being globally divergence-free in our problems. Any interpolation, even non-physical ones preserve a globally solenoidal field, as the domain is bounded by two impermeable walls in one direction, and is periodic in the other two. Thus the error induced by the residual divergence should be small and does not accumulate during the time advance. Nevertheless, the effect may get worse as the grid becomes more non-uniform which could become a problem at higher drivings.

A drastic way of addressing this issue could be a "minimum energy" correction to fully remove the residual divergence of the interpolated velocity; this would require the solution of a Poisson equation for a velocity correction, with the local divergence as source term. While the multiple resolutions strategy would remain favourable, solving a Poisson equation on the fine grid would negate one of the main advantages of the method.

Alternatively, we propose here a locally mass-conserving interpolation which produces a divergence-free velocity field without solving the Poisson equation on the fine grid. Notice that the velocity field on the coarse grid is solenoidal in the sense of $\partial_x u + \partial_y v + \partial_z w = 0$ at the cell center. Thus we first compute $(\partial_x u, \partial_y v, \partial_z w)$ at the cell center on the coarse grid, then interpolate them to the cell center on the fine grid by using the tricubic interpolation. Since the three quantities are defined at the same locations on both the coarse and fine grids, the interpolation involves the same stencil points and therefore the relationship $\partial_x u + \partial_y v + \partial_z w = 0$ is always satisfied at the cell centers on the fine grid, as it is satisfied to machine precision at every cell center of base mesh. Therefore, continuity is automatically preserved during the interpolation.

With the interpolated $(\partial_x u, \partial_y v, \partial_z w)$ the velocity field can be reconstructed. For example, one can conduct a two-dimensional tricubic interpolation for u on an arbitrary y–z plane, then integrate along the x-direction according the interpolated $\partial_x u$. v can be treated similarly. For wall normal velocity w, one can just integrate $\partial_z w$ from the two plates where $w \equiv 0$.

Of course some price has to be paid by using this new interpolation. It requires more CPU time than the tricubic interpolation due to extra integration and data communication between processes. Another issue is the accumulated error of the integration. For instance, the integral of the interpolated $\partial_z w$ from one plate to the opposite one may not equal to zero, which will violate the impermeability condition at wall boundary. In practice, one can reduce the error by constructing w from both plates and taking the average. In our simulation we found out that the error of w at two plates is of order 10^{-6}, and the total flux caused by this error is basically machine zero ($\mathcal{O}(10^{-16})$ as we use double precision) at both plates. We note that the difference between the total kinetic energy of the coarse grid and fine grid is usually about 0.1%. Lastly, this interpolation technique is anisotropic. From the previous discussion, we have seen that the divergence in the interpolated field is the highest near the walls. We perform the integration in the non-uniform direction to keep errors as low as possible because there is a clear starting point for the interpolation (the walls), unlike the two homogeneous directions.

The previous discussion suggests that the proposed multiple resolution procedure can only work if the coarse mesh is fine enough to fully resolve the momentum field. In Section 5 numerical results will confirm this statement showing that when the base mesh adequately resolves the momentum field good results and CPU time saving can be obtained by refining only the grid for the scalar. On the other hand, if the coarse grid does not fully resolve the momentum field even very large values of $\mathcal{M}$ do not lead to correct results.

4.2. Multiple resolution strategy in time

The multiple resolution in space entails that the node spacing for the scalars (Δ_S) is smaller than that for the momentum (Δ_U) and this has immediate consequences on the stability of the time integration because of the explicit terms. Due to the Courant–Friedichs–Lewy condition [34], $\Delta t \cdot \max[U/\Delta_S] \leq \mathcal{C}_{CFL}$, in fact, the time step must decrease by a factor equal to the refinement $\mathcal{M}$ because $\min[\Delta_S] = \min[\Delta_U]/\mathcal{M}$. As this small Δt is not needed by the base mesh for momentum and pressure, the usage of the current approach becomes disadvantageous very rapidly, especially in high *Sc* flows requiring high values of $\mathcal{M}$. However, a multiple resolution strategy can be applied also in time, by advancing the more expensive momentum and pressure with a larger time-step and the scalar with a smaller one, using a temporally interpolated velocity. In this way, the stability of the explicit terms in the scalar equation is retained without the penalization of an unnecessary small integration step for all the other equations.

The integration of the scalar equation is therefore performed in $\mathcal{L}$ sub-steps, and at each intermediate l time level the velocity is linearly interpolated through

$$\bar{q}^l = \frac{\mathcal{L}-l}{\mathcal{L}}\bar{q}^n + \frac{l}{\mathcal{L}}\bar{q}^{n+1}, \tag{9}$$

where $\bar{q}^n$ is the spatially interpolated velocity at time step n. A simple linear interpolation is used, which is enough to ensure correctness as is demonstrated in Section 5. This velocity is then used to advect the scalar(s) for every subtimestep using equation (3) until the scalars have been advanced to the time t^{n+1}. Then, the velocity can be advanced a further time-step and the procedure repeated. If the maximum possible CFL is used for the velocity equations, which is usually the case, then $\mathcal{L} \geq \mathcal{M}$ must be satisfied to ensure stability. We also note that no filtering in time is applied for the scalar when coupling back to the momentum equation. This filtering appears to be unnecessary, as the time-step is set not due to the physical time but instead due to the stability constraints of the advective term and thus the error incurred is small.

5. Results and computational performance

In this section we will present the results of the multiple resolution method, and the associated saving in computational time and memory usage. A more extensive analysis will be performed in the first part for RB flow while, in the second part, additional results with anisotropic refinements will be shown for DDC flow.

5.1. Rayleigh–Bénard flow

A series of rectangular RB simulations with aspect ratio $\Gamma = 1$ were run to validate and to demonstrate the benefits of the described method. Meshes of $96^2 \times 192$, $128^2 \times 256$ and $192^2 \times 384$ (only for $Pr = 1$) were used for momentum, the *Ra* was kept constant at 10^9 while *Pr* was taken $Pr = 1$ or $Pr = 10$. In order to minimize the computational costs $\mathcal{L} = \mathcal{M}$ was always used except for a specific set of runs in which the effects of $\mathcal{L}$ were isolated (see Table A.3).

Fig. 10(a), (b) shows the ratio between the different definitions of *Nu* and the maximum refinement level $\mathcal{M}$ while the bottom panels report the convergence of $Nu_{u_z\theta}$ to the asymptotic reference value, calculated from a wiggle-free simulation such as the one seen in Fig. 7(b). The raw numerical values can be found in Table A.2 at the end of Appendix A.

As mentioned before, the multiple resolution strategy only works if the base mesh is sufficiently fine to fully resolve the momentum field. Looking at Fig. 4(a) we can see that at $Pr = 1$ the $96^2 \times 192$ mesh is not sufficiently fine, and the different *Nu* definitions do not show a monotonic convergence of the Nusselt ratios to the asymptote. It is interesting to note that for $\mathcal{M} = 4$ the *Nu* ratios get close to one, although the absolute values of *Nu* are wrong and do not indicate a convergence towards the reference grid independent value. On the other hand, the $192^2 \times 384$ grid yields a converged value of Nu_{ϵ_v} even though that resolution is not enough for the computation of Nu_{ϵ_θ} that involves squared temperature gradients. For this case a refinement factor $\mathcal{M} = \mathcal{L} = 2$ for the temperature gives an appropriate resolution as shown by the Nusselt numbers that converge to the reference value. Although a factor two in space and time might seem to produce only limited benefits, we should consider that it implies a grid for the momentum and pressure with 2^3 less elements than for the temperature. In addition, the momentum equations are solved only once every other time sub-step therefore, even if there is an overhead introduced by the interpolation of the fields, the CPU time savings are substantial. In our case the standard simulation on the single grid $384^2 \times 768$ required for the integration of dimensionless time unit a wall-time of 36.4 minutes on 96 processors, for a total of 58.2 CPU hours using 13 GB of RAM memory. Using $\mathcal{L} = \mathcal{M} = 2$, one dimensionless time unit required a wall-time of 36.7 minutes on 48 processors for a total of 29.4 CPU hours using 4.9 GB of RAM memory.

The method becomes even more advantageous as the Prandtl number increases. In fact, from equations (2)–(3) we see that the nondimensional diffusivities of momentum and temperature, read $Re = \sqrt{Ra/Pr}$ and $Pe = \sqrt{Ra\,Pr}$, respectively.

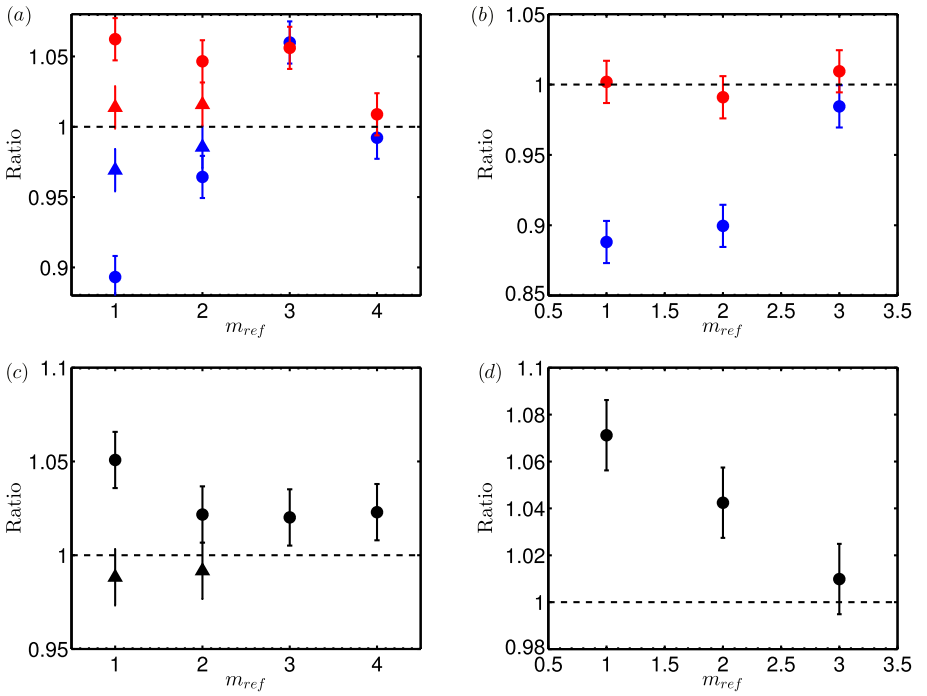

Fig. 10. (a) Ratio between different ways of calculating Nu against grid size for the two base meshes and increasing spatial refinement for $Pr = 1$. (b) The same as (a) for $Pr = 10$. (c) Convergence to an asymptotic value of $Nu_{u_z\theta}$ for increasing refinement and $Pr = 1$. (d) The same as (c) for $Pr = 10$. Circles are for base meshes of $96^2 \times 192$, while triangles are for $192^2 \times 384$ base meshes. On the top panels, blue indicates the ratio $Nu_{\epsilon_\theta}/Nu_{u_z\theta}$ and red the ratio $Nu_{\epsilon_v}/Nu_{u_z\theta}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Therefore for $Pr > 1$ the momentum field smoothens while the temperature (scalar) field develops sharper gradients. This results in larger Kolmogorov η_K and smaller Batchelor η_B scales that need different meshes to be properly resolved. Fig. 4(b) shows that at $Pr = 10$ the increased momentum diffusivity makes even the relatively coarse grid $128^2 \times 256$ adequate for the description of the momentum field. On the other hand, the same mesh is clearly too coarse for the scalar field as the ratio $Nu_{u_z\theta}/Nu_{\epsilon_\theta}$ strongly deviates from one. This grid, however, can be used to advance the momentum and to generate a refined mesh to advect the scalar. For this case the convergence of the Nusselt numbers to the reference value is obtained for $\mathcal{M} = \mathcal{L} = 3$ that yields a computational gain around a factor 7 and a reduction of RAM memory by a factor 3.5 when compared to the reference cases using a single mesh.

Before concluding this section we point out that for all simulations we have used a refinement factor for the time step $\mathcal{L} = \mathcal{M}$. Values of $\mathcal{L}$ smaller than $\mathcal{M}$ can be used provided the CFL number for momentum is reduced so that the scalar integration remains stable; this increases the number of time steps needed to advance the simulation over the same physical time resulting in an increased computational cost. On the other hand, further increasing $\mathcal{L}$ beyond $\mathcal{M}$ does not modify the results within statistical error and empirical evidence supporting this statement can be found in Table A.3 at the end of Appendix A.

5.2. Double diffusive convection

In this section we apply the multiple-resolution technique to convection flows with two active scalar components, namely, the double-diffusive convection (DDC). In this system two scalars with very different diffusivities are coupled to the flow field, one of which is stabilizing and the other destabilizing. For instance, in the ocean the two scalars are temperature ($Pr_T \approx 7$) and salinity ($Pr_S \approx 700$). We will simulate the situation often found in subtropic gyres, where the top water are warmer and saltier. Thus the temperature field stabilizes the flow while salinity field has the opposite effect. We utilize a geometry similar to the RB flow which is vertically bounded by two non-slip plates. A snapshot of the flow can be seen in Fig. 11. Very sharp gradients of salinity can be observed as salt fingers grow from both plates, while the temperature field is in a quasi-diffusive state.

Even for the flow shown in Fig. 11 with $Ra_S = 5 \times 10^7$, which is not so turbulent, the salt fingers have rather small length scale and require very fine mesh due to the very high Prandtl number $Pr_S = 700$. If one uses a single-resolution strategy and solves the salinity field properly, then momentum and temperature fields must be heavily over-resolved. By using our multiple-resolution method, such as the meshes shown in Fig. 12, the unnecessary computation cost can be reduced significantly. In this case, the temperature and momentum are discretized on base mesh of resolution $144 \times 144 \times 144$, and salinity on refined mesh of resolution $432 \times 432 \times 432$ with factor $\mathcal{M} = 3$, respectively. We did not use the maximum possible CFL for the velocity equations, and $\mathcal{L} = 1$ for the time step.

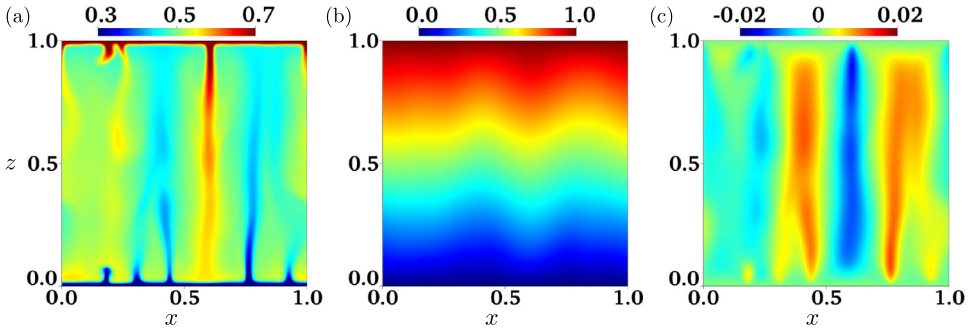

Fig. 11. Instantaneous snapshot of (a) salinity ($Sc_s = 700$), (b) temperature ($Pr = 7$) and (c) vertical velocity in a DDC simulation at drivings of $Ra_s = 5 \times 10^7$ and $Ra_T = 5 \times 10^5$. This results in $Le = 100$ and $R_\rho = 1$.

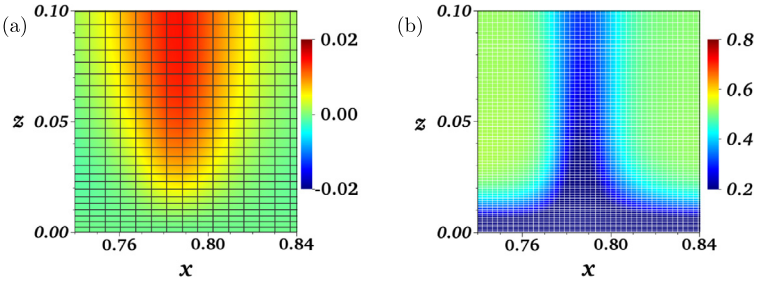

Fig. 12. Zoom-in plots show the base mesh for velocity and refined mesh for salinity near one salt finger root. The control parameters are the same as in Fig. 11.

Similar to the RB case, the results in terms of heat transfer, salinity fluxes and turbulence statistics agree within the uncertainty of 2–3% with those obtained using a single refined grid for all the variables. By using two 12-core 2.6 GHz Intel Xeon E5-2690v3 (Haswell) CPUs with 12 MPI process and 2 OpenMP threads, the CPU time for one time step is about 12.9 seconds for using single refined grid, 2.69 seconds for multi-grids with tricubic interpolation, and 2.98 seconds for multi-grids with our mass conserving interpolation, yielding a saving factor that exceeds 4. A decrease of RAM memory usage by more than 50% is also achieved by using the multi resolution strategy.

Actually, the multiple-resolution method has been used to conduct a systematic investigation on the DDC flow, e.g. see Ref. [35], in which the simulation was validated by making a one-to-one comparison with experiment and agreement is within the measurement uncertainty. Also the huge saving of CPU time and memory allows us to achieve very high Rayleigh number even for Pr_S as large as 700.

Before concluding this section it is worthwhile to discuss briefly why in DDC it is possible to simulate a flow with a Schmidt number of 700 without using a refinement factor $\sqrt{Sc} \simeq 27$. The DDC equations in nondimensional form read

$$\partial_t u_i + u_j \partial_j u_i = -\partial_i p + \sqrt{\frac{Pr_S}{Ra_S}}\, \partial_{jj} u_i + (R_\rho \theta - S)\delta_{iz}, \tag{10}$$

$$\partial_t \theta + u_j \partial_j \theta = Le \sqrt{\frac{1}{Pr_S Ra_S}}\, \partial_{jj} \theta, \tag{11}$$

$$\partial_t S + u_j \partial_j S = \sqrt{\frac{1}{Pr_S Ra_S}}\, \partial_{jj} S, \tag{12}$$

where the flow parameters are defined as $Ra_T = g\beta_T \Delta T L/(\nu\kappa_T)$, $Pr_T = \nu/\kappa_T$, $Ra_S = g\beta_S \Delta S L/(\nu\kappa_S)$ and $Pr_S = \nu/\kappa_S$, $Le = Pr_S/Pr_T$ and $R_\rho = Ra_T Le/Ra_S$.

It can be noted that the Reynolds number $Re = \sqrt{Pr_S/Ra_S}$ decreases with Pr_S. Therefore for large enough values the flow does not fully transition to turbulence. Furthermore, when dealing with an active scalar with a very large Prandtl number, the inertial terms in the Navier–Stokes equations become negligible and the velocity becomes effectively "linearly enslaved" to the scalar field [36]. The equations for both scalars are linear and they cannot sustain the cascade without a turbulent velocity field. This is indeed the case for the flow parameters of the present numerical example (see also Ref. [35]) where the fully turbulent three dimensional cascade cannot be achieved and a factor $\mathcal{M} = \sqrt{Sc} \simeq 27$ is not required for salinity. Even so, the multiple resolutions strategy results in a substantial gain factor in computational time and RAM memory occupation when compared against the single grid strategy.

6. Summary and conclusions

In this paper we have presented a numerical strategy for the direct numerical simulation of turbulent flows with active and/or passive scalar fields without over-resolving the momentum equation and its pressure correction. This is certainly the case of flows with scalar diffusivity smaller than the kinematic viscosity (Pr or $Sc > 1$). Substantial computational time and memory occupation savings are even obtained for equally diffusive fields with Schmidt numbers of order unity. The different requirements for spatial discretization of scalars with respect to momentum originate not only from the diffusivity but also from the higher intermitency of scalars. This scenario modifies the picture obtained from dimensional analysis that compares only the Kolmogorov and the Batchelor scales for the resolution requirements.

To reduce computational costs, a multiple resolutions strategy was developed in which momentum is discretized on a base mesh while scalars are discretized on a refined mesh. To solve the scalar diffusion–advection equation, momentum is spatially interpolated onto the refined grid and either tricubic Hermitian splines or a more sophisticated procedure, based on the interpolation of velocity divergence, are proposed. The scalar is advanced in time, and if necessary, coarsened to couple it back to the momentum equations. Due to stability constraints on the non-linear terms, the scalar is advanced in time using a refined timestep. Velocity is linearly interpolated in time for all the intermediate timesteps. The optimal amount of substeps $\mathcal{L}$ coincides with the grid refinement factor $\mathcal{M}$, when it is isotropic, or with $\max[\mathcal{M}^i]$ when it is anisotropic.

The method was applied to Rayleigh–Bénard convection, and decoupling the grid resolutions was found to result in computational speedups around two for Prandtl number unity, and seven for $Pr = 10$. This strategy was also applied to high Sc flows, also resulting in computational advantages of approximately a factor four in CPU time reduction. In large Sc flows with an active scalar, the velocity becomes effectively enslaved to the active scalars, and very large grid refinements are not needed. Due to the large costs, both in operations, memory usage and in communication associated to solving the Poisson equation, we expect the gains to increase for larger grids and larger drivings. This is because the Poisson solver is the most expensive part of advancing the Navier–Stokes equations in time, and this does not scale linearly with the amount of points, while the scalar diffusion–advection equations do. We expect gains of about three to four times for RB simulations at $Pr = 1$ and $Ra = 10^{12}$ with production grids of about 1 billion points and $\mathcal{M} = 2$, planned for the future. Also the memory consumption is heavily reduced, by a factor $2.6\times$ with a refinement of two, and this makes some simulations possible on supercomputers with a lower memory per core and decreases the dependence on high CPU-memory bandwidth.

Once again, it is crucial that the base mesh is fine enough to correctly resolve the momentum field. Adding more refinement to the scalar mesh when the velocity grid is insufficient does not give an improvement of the quality of the results, and it might even lead to the suppression of small scales that violate energy conservation. This method could in principle be additionally applied to flows with $Sc < 1$. Obviously, in this case the velocity field should be solved on a mesh finer than that of the scalar. Although explicit numerical tests have not been attempted, we expect that the computational overhead introduced by the interpolation and coarsening of the fields overcomes the advantages produced by solving the scalar equation on a coarser mesh.

Acknowledgements

We acknowledge FOM, an ERC Advanced Grant and computing time from SurfSARA (granted through NWO) and PRACE grant 2013091966.

Appendix A. Numerical details

Table A.1
Details of grid resolution used for standard single grid runs. Simulations were run until $Nu_{u_z\theta}$ achieved 1% temporal convergence. All the simulations are performed at $Ra = 10^9$ and $\Gamma = 1$. The first column shows resolution, the second shows Pr, while the other four show the results of the different definitions of Nu.

$N_x \times N_y \times N_z$	Pr	$Nu_{u_z\theta}$	Nu_{θ_w}	Nu_{ϵ_v}	Nu_{ϵ_θ}
$96 \times 96 \times 192$	1	66.8	67.2	71.0	59.7
$128 \times 128 \times 256$	1	64.9	64.6	67.1	60.5
$192 \times 192 \times 384$	1	63.2	63.2	64.1	61.3
$256 \times 256 \times 512$	1	63.8	63.6	64.6	62.5
$384 \times 384 \times 768$	1	64.0	63.6	64.3	63.2
$96 \times 96 \times 192$	10	68.4	67.9	68.6	60.8
$128 \times 128 \times 256$	10	65.5	65.2	65.7	61.0
$192 \times 192 \times 384$	10	63.2	63.0	63.1	61.0
$256 \times 256 \times 512$	10	63.6	63.9	63.6	62.4
$384 \times 384 \times 768$	10	63.4	63.9	63.4	63.3

Table A.2
Details of grid resolution used for multiple resolutions runs. Simulations were run until $Nu_{u_z\theta}$ achieved 1% temporal convergence. All the simulations are performed at $Ra = 10^9$ and $\Gamma = 1$. The first column shows resolution, the second shows the refinement of the scalar grid in all directions, the third shows Pr, while the other four show the results for the different definitions of Nu. For all simulations $\mathcal{L} = \mathcal{M}$.

$N_x \times N_y \times N_z$	$\mathcal{M}$	Pr	$Nu_{u_z\theta}$	Nu_{θ_w}	Nu_{ϵ_v}	Nu_{ϵ_θ}
$96 \times 96 \times 192$	2	1	65.5	65.4	68.4	63.0
$96 \times 96 \times 192$	3	1	65.6	65.3	69.3	69.5
$96 \times 96 \times 192$	4	1	65.4	65.4	66.0	64.9
$128 \times 128 \times 256$	2	1	64.4	64.4	66.7	66.7
$128 \times 128 \times 256$	3	1	64.4	64.4	66.9	66.9
$192 \times 192 \times 384$	2	1	63.5	63.4	62.6	64.5
$96 \times 96 \times 192$	2	10	66.6	64.8	60.0	66.0
$96 \times 96 \times 192$	3	10	64.5	64.5	63.5	63.5
$96 \times 96 \times 192$	4	10	65.4	65.0	64.2	64.2
$128 \times 128 \times 256$	2	10	64.3	64.2	64.7	63.0
$128 \times 128 \times 256$	3	10	64.6	64.1	62.2	64.8

Table A.3
Details of the testing for the temporal multiple resolutions. Simulations were run until $Nu_{u_z\theta}$ achieved 1% temporal convergence. All the simulations are performed at $Ra = 10^9$ and $\Gamma = 1$ on a grid $128 \times 128 \times 256$ with $\mathcal{M} = 2$. The first column shows the time refinement level $\mathcal{L}$, the second shows the maximum CFL computed on the momentum grid, while the last four show the results of the different definitions of Nu.

$\mathcal{L}$	$\mathcal{C}_{CFL}$	$Nu_{u_z\theta}$	Nu_{θ_w}	Nu_{ϵ_v}	Nu_{ϵ_θ}
1	0.6	64.5	64.6	67.3	64.0
2	1.2	64.4	64.4	66.8	63.5
3	1.2	64.6	64.3	67.2	63.5

References

[1] S. Arya, Air Pollution Meteorology and Dispersion, Oxford University Press, 1990.
[2] W.R. Peltier, C.P. Caulfield, Mixing efficiency in stratified shear flows, Annu. Rev. Fluid Mech. 35 (2003) 135–167.
[3] F.P. Incropera, Convection heat transfer in electronic equipment cooling, J. Heat Transf. 110 (1988) 1097–1111.
[4] E.D. Siggia, High Rayleigh number convection, Annu. Rev. Fluid Mech. 26 (1994) 137–168.
[5] P.F. Linden, The fluid mechanics of natural ventilation, Annu. Rev. Fluid Mech. 31 (1999) 201–238.
[6] J.S. Turner, Multicomponent convection, Annu. Rev. Fluid Mech. 17 (1985) 11–44.
[7] R.W. Schmitt, Double diffusion in oceanography, Annu. Rev. Fluid Mech. 26 (1994) 255–285.
[8] J. Sethain, P. Smerenka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 35 (2010) 341–372.
[9] C. Conti, D. Rossinelli, P. Koumoutsakos, GPU and APU computations of finite time Lyapunov exponent fields, Annu. Rev. Fluid Mech. 231 (2012) 2229–2244D.
[10] A. Pumir, A numerical study of the mixing of a passive scalar in three dimensions in the presence of a mean gradient, Phys. Fluids 6 (1994).
[11] D.J. Bogucki, J.A. Domaradzki, P.K. Yeung, Direct numerical simulations of passive scalars with Pr > 1 advected by turbulent flow, J. Fluid Mech. 343 (1997) 111–130.
[12] A. Pumir, Structure of the three-point correlation function of a passive scalar in the presence of a mean gradient, Phys. Rev. E 57 (1998).
[13] A. Celani, A. Lanotte, A. Mazzino, M. Vergassola, Universality and saturation of intermittency in passive scalar turbulence, Phys. Rev. Lett. 84 (2000) 2385.
[14] J. Schumacher, K.R. Sreenivasan, P.K. Yeung, Very fine structures in scalar mixing, J. Fluid Mech. 531 (2005) 113–122.
[15] D. Donzis, P.K. Yeung, Resolution effects and scaling in numerical simulations of passive scalar mixing in turbulence, Physica D 239 (2010) 1278–1287.
[16] Z. Warhaft, Passive scalars in turbulent flows, Annu. Rev. Fluid Mech. 32 (2000) 203–240.
[17] G. Ahlers, S. Grossmann, D. Lohse, Heat transfer and large scale dynamics in turbulent Rayleigh–Bénard convection, Rev. Mod. Phys. 81 (2009) 503.
[18] R.J.A.M. Stevens, D. Lohse, R. Verzicco, Prandtl and Rayleigh number dependence of heat transport in high Rayleigh number thermal convection, J. Fluid Mech. 688 (2011) 31–43.
[19] E.P. van der Poel, R. Verzicco, D. Lohse, Plume emission statistics in turbulent Rayleigh–Bénard convection, J. Fluid Mech. 772 (2015) 5–15.
[20] B.I. Shraiman, E.D. Siggia, Scalar turbulence, Nature 405 (2000) 639–646.
[21] D. Lohse, K.-Q. Xia, Small-scale properties of turbulent Rayleigh–Bénard convection, Annu. Rev. Fluid Mech. 42 (2010) 335–364.
[22] K.R. Sreenivasan, On local isotropy of passive scalars in turbulent shear flows, Proc. R. Soc. Lond. A 434 (1991) 165.
[23] F. Moisy, H. Willaime, J. Andersen, P. Tabeling, Passive scalar intermittency in low temperature helium flows, Phys. Rev. Lett. 86 (2001) 4827–4830.
[24] T. Gotoh, S. Hatanaka, H. Miura, Spectral compact difference hybrid computation of passive scalar in isotropic turbulence, J. Comput. Phys. 231 (2012) 7398–7414.
[25] J.-B. Lagaert, G. Balarac, G.-H. Cottet, Hybrid spectral–particle method for the turbulent transport of a passive scalar, J. Comput. Phys. 260 (2014) 127–142.
[26] S. Verma, G. Blanquart, On filtering in the viscous-convective subrange for turbulent mixing of high Schmidt number passive scalars, Phys. Fluids 25 (2013).
[27] O. Shishkina, R.J.A.M. Stevens, S. Grossmann, D. Lohse, Boundary layer structure in turbulent thermal convection and its consequences for the required numerical resolution, New J. Phys. 12 (2010) 075022.
[28] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[29] R. Verzicco, P. Orlandi, A finite-difference scheme for three-dimensional incompressible flow in cylindrical coordinates, J. Comput. Phys. 123 (1996) 402–413.
[30] E.P. van der Poel, R. Ostilla-Monico, J. Donners, R. Verzicco, A pencil distributed finite difference code for strongly turbulent wall-bounded flows, Comput. Fluids 116 (2015) 10–16.
[31] R.J.A.M. Stevens, R. Verzicco, D. Lohse, Radial boundary layer structure and Nusselt number in Rayleigh–Bénard convection, J. Fluid Mech. 643 (2010) 495–507.

[32] B.I. Shraiman, E.D. Siggia, Heat transport in high-Rayleigh number convection, Phys. Rev. A 42 (1990) 3650–3653.
[33] M.A.T. van Hinsberg, J.H.M. ten Thije Boonkkamp, F. Toschi, H.J.H. Clercx, Optimal interpolation schemes for particle tracking in turbulence, Phys. Rev. E 87 (2013) 043307.
[34] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, Math. Ann. 100 (1928) 32–74.
[35] Y. Yang, E.P. van der Poel, R. Ostilla-Monico, C. Sun, R. Verzicco, S. Grossmann, D. Lohse, Salinity transfer in bounded double diffusive convection, J. Fluid Mech. 768 (2015) 476–491.
[36] C.R. Doering, P. Constantin, On upper bounds for infinite-Prandtl-number convection with or without rotation, J. Math. Phys. 42 (2001) 784–795.