Available online at www.sciencedirect.com
ScienceDirect
Computers & Fluids 36 (2007) 1187–1198
computers & fluids
www.elsevier.com/locate/compfluid

Simple and accurate scheme for fluid velocity interpolation for Eulerian–Lagrangian computation of dispersed flows in 3D curvilinear grids

Cristian Marchioli [a,*], Vincenzo Armenio [b], Alfredo Soldati [a,1]

[a] Dipartimento di Energetica e Macchine and Centro Interdipartimentale di Fluidodinamica e Idraulica, Università di Udine, Via delle Scienze 208, 33100 Udine, Italy
[b] Dipartimento di Ingegneria Civile e Ambientale, Università di Trieste, Piazzale Europa 1, 34127 Trieste, Italy

Received 27 January 2006; received in revised form 4 August 2006; accepted 2 November 2006
Available online 29 March 2007

Abstract

Particle tracking in turbulent flows in complex domains requires accurate interpolation of the fluid velocity field. If grids are non-orthogonal and curvilinear, the most accurate available interpolation methods fail. We propose an accurate interpolation scheme based on Taylor series expansion of the local fluid velocity about the grid point nearest to the desired location. The scheme is best suited for curvilinear grids with non-orthogonal computational cells. We present the scheme with second-order accuracy, yet the order of accuracy of the method can be adapted to that of the Navier–Stokes solver.

An application to particle dispersion in a turbulent wavy channel is presented, for which the scheme is tested against standard linear interpolation. Results show that significant discrepancies can arise in the particle displacement produced by the two schemes, particularly in the near-wall region which is often discretized with highly-distorted computational cells.

1. Introduction

The interpolation of a field computed over a discrete grid is required to know the field values in physical points not coinciding with the grid nodes. In the specific case of applications which involve flows with dispersed species, solid particles, droplets or bubbles, best-suited simulations are those based on the Eulerian–Lagrangian approach which, at the ever-decreasing price of a larger computational effort, grants easy coding and reliable representation of the physics. In this approach, accurate Lagrangian tracking of the dispersed phase requires accurate interpolation of the Eulerian carrier flow field.

In previous papers focused on dispersed flows, several interpolation methods (among others: hybrid Lagrange/Chebyshev polynomials; cubic splines and quadratic interpolation) were used by many researchers [1–6] in simple geometries such as channels, pipes and boundary layers, in which the Eulerian grid was Cartesian and uniform in two directions (for further details see the review by Soldati [7]).

However, if the physical problem is such to require non-uniform and non-Cartesian grids for discretization of complex and irregular flow domains, interpolation may not be straightforward and may lead to inaccuracies. Recently, we had to face this issue in the frame of our work on particle/droplet deposition and entrainment over a wavy interface [8], which mimics the dynamics of the droplets sprayed over capillary waves at the ocean/atmosphere interface.

While literature interpolation techniques (viz. spectral, partial Hermite, Lagrangian, shape function methods, cubic splines, etc.) have been extensively studied and

* Corresponding author. Tel.: +39 0432 558006.
E-mail address: marchioli@uniud.it (C. Marchioli).
[1] Also affiliated with Department of Fluid Mechanics, CISM, 33100 Udine, Italy.

0045-7930/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2006.11.004

applied on Cartesian grids [1,9,10], less effort has been devoted to development of interpolation methods for non-Cartesian grids with boundary-fitted curvilinear coordinates. Typically, these methods are linear and use different geometrical weighting procedures [11–14] and produce similar interpolation functions in terms of performance and accuracy. Some of these procedures make use of geometry-based algorithms which may be applicable only in two-dimensional meshes or may become impractical in Eulerian–Lagrangian simulations due to significant loss in accuracy when highly-distorted computational grids are used [13]. Other procedures are based on iterative methods which require properly defined residuals to check the convergence of the solution [13].

To overcome these restrictions, in this work we propose an interpolation scheme based on Taylor series expansion of fluid velocities in computational space, best suited for structured curvilinear grids characterized by cells of arbitrary shape and regular topology. The main aspects considered are accuracy, performance and simplicity of the scheme. Specifically, we will focus on the second-order version of the method since current state-of-the-art fully conservative numerical codes working with curvilinear coordinates are second-order accurate: see the Finite Difference (FD) multigrid method by Zang et al. [15,16] or the Finite Volume (FV) method by Patankar and Joseph [17] among others. However, one of the most interesting features of the present interpolation scheme is that its order of accuracy can be easily adapted to that of the Navier–Stokes solver. For instance, the order of accuracy can be increased by including higher-order derivatives in the Taylor expansion, thus producing schemes beyond linear interpolation.

The choice of an interpolation scheme depends also on computational cost, which is of crucial importance when tracking large swarms of particles in Eulerian flow fields. Calculations involved in particle tracking may become cost effective if the interpolation scheme requires reduced computational overhead with respect to standard techniques (for instance, trilinear interpolation). This is one of the reasons that led us to develop the scheme proposed in this paper, which we have used to study the abovementioned dispersed turbulent flow over wavy interfaces [8]. As an example of possible application of the interpolation scheme, some results from this study are briefly outlined in Section 3. However, we believe that the proposed scheme could be used to investigate a large variety of problems in the context of Eulerian–Lagrangian simulations of dispersed flows in three-dimensional curvilinear grids.

2. Velocity evaluation method

Most smooth functions $f(x)$ can be approximated in the form of a Taylor series expansion about a point x_0 where exact values of $f(x)$ and of its derivatives are known. Following this approach, the interpolation problem can be addressed within an Eulerian carrier scheme in which the fluid velocity components, U_i, are available at spatially discrete grid nodes.

First, the host cell has to be identified. This inclusion problem is solved (i) by orienting counter-clockwise the cell nodes belonging to the same face and (ii) by computing the cross product between the vector connecting one of the face nodes to the particle and the vector connecting the same node to its counter-clockwise neighbor. If the product is negative for all faces the point is inside the cell. Further details can be found in works by Chen [18] and by Zhou and Leschnizer [19], which focus on the inclusion problem and provide efficient particle-locating algorithms.

Once the host cell is known, the algorithm uses the fluid velocity at the cell node nearest to the particle, $N(x_j) = N(x,y,z)$, to estimate the local fluid velocity at the instantaneous particle position, $P(X_j) = P(X,Y,Z)$. Henceforth x_j and X_j represent the coordinates of the Eulerian grid points and those of the Lagrangian particle, respectively. In scalar form, we obtain:

$$U_i|_P = \sum_{n_1=0}^{s_1}\sum_{n_2=0}^{s_2}\sum_{n_3=0}^{s_3}\left(\frac{\partial^{n_1}}{\partial x^{n_1}}\frac{\partial^{n_2}}{\partial y^{n_2}}\frac{\partial^{n_3}}{\partial z^{n_3}}U_i\right)\Bigg|_N \cdot\frac{(X-x)^{n_1}(Y-y)^{n_2}(Z-z)^{n_3}}{n_1!n_2!n_3!}, \quad i=1,2,3. \tag{1}$$

With specific reference to a second-order accurate FD/FV code, the above equation can be rewritten in terms of Cartesian coordinates as follows:

$$U_i|_P \approx U_i|_N + \left.\frac{\partial U_i}{\partial x}\right|_N (X-x) + \left.\frac{\partial U_i}{\partial y}\right|_N (Y-y) + \left.\frac{\partial U_i}{\partial z}\right|_N (Z-z), \quad i=1,2,3. \tag{2}$$

Only first-order terms are considered, so that the truncation error is $O[(X_j - x_j)^2]$, consistent with that of the numerical scheme. Recall that, in an mth-order Taylor series scheme, the interpolation error decreases as $(\Delta x)^m$ as the grid spacing Δx tends to zero.

In curvilinear structured grids, the Navier–Stokes equations are usually solved in a computational space, whose coordinates we denote with ξ, η and ζ. Derivatives are thus transformed as follows:

$$\frac{\partial U_i}{\partial x_j} = \frac{\partial U_i}{\partial \xi_k}\frac{\partial \xi_k}{\partial x_j} = \frac{\partial U_i}{\partial \xi}\frac{\partial \xi}{\partial x_j} + \frac{\partial U_i}{\partial \eta}\frac{\partial \eta}{\partial x_j} + \frac{\partial U_i}{\partial \zeta}\frac{\partial \zeta}{\partial x_j}, \quad i,j,k=1,2,3. \tag{3}$$

No additional metric calculation is required since the terms $\partial\xi_k/\partial x_j$ are readily available once the computational grid has been generated and only the terms $\partial U_i/\partial\xi_k$ have to be computed. In principle, these terms could be discretized by a 2-point finite-difference formula. This choice would be self-consistent with the second-order truncation error and should allow to obtain an interpolation scheme with C^0 continuity, i.e. the interpolated function is continuous across the interfaces of neighboring interpolation cells [9], without sacrificing the order of accuracy. However, discon-

tinuous interpolation on conservative schemes has important implications especially for particulate solvers. To address the issue of discontinuous interpolation, we decided to use the 3-point symmetric formula centered around N for the first-order derivatives. In this case, the explicit equation for the interpolated fluid velocities reads:

$$\begin{aligned} U_i|_P \approx U_i|_N &+ \left[\frac{U_i^{l+1}-U_i^{l-1}}{2\Delta\xi}\left.\frac{\partial\xi}{\partial x}\right|_N + \frac{U_i^{m+1}-U_i^{m-1}}{2\Delta\eta}\left.\frac{\partial\eta}{\partial x}\right|_N \right. \\ &\left. + \frac{U_i^{n+1}-U_i^{n-1}}{2\Delta\zeta}\left.\frac{\partial\zeta}{\partial x}\right|_N\right](X-x) \\ &+ \left[\frac{U_i^{l+1}-U_i^{l-1}}{2\Delta\xi}\left.\frac{\partial\xi}{\partial y}\right|_N + \frac{U_i^{m+1}-U_i^{m-1}}{2\Delta\eta}\left.\frac{\partial\eta}{\partial y}\right|_N \right. \\ &\left. + \frac{U_i^{n+1}-U_i^{n-1}}{2\Delta\zeta}\left.\frac{\partial\zeta}{\partial y}\right|_N\right](Y-y) \\ &+ \left[\frac{U_i^{l+1}-U_i^{l-1}}{2\Delta\xi}\left.\frac{\partial\xi}{\partial z}\right|_N + \frac{U_i^{m+1}-U_i^{m-1}}{2\Delta\eta}\left.\frac{\partial\eta}{\partial z}\right|_N \right. \\ &\left. + \frac{U_i^{n+1}-U_i^{n-1}}{2\Delta\zeta}\left.\frac{\partial\zeta}{\partial z}\right|_N\right](Z-z). \end{aligned} \tag{4}$$

Index space $L(l,m,n)$ identifies the cell node N in the curvilinear coordinate system, whereas the 3D index space $I(i,j,k)$ is used in the Cartesian coordinate system (see Fig. 1).

Eq. (4) requires the values of the three components of velocity and their first-order spatial derivatives at N, 12 scalars in total. For M particles $12M$ scalars are thus required for the evaluation of $U_i|_P$. Trilinear interpolation would require $24M$ scalars [10].

Eq. (4) has been applied not only to the cell node N nearest to the particle but also to each node N_r of the host cell, to check whether the use of more grid points would improve the accuracy of the interpolation. For the three-dimensional cases, eight estimates $(U_i|_P)_{N_r}$ have been obtained and then averaged to calculate the final fluid velocity. We tested both simple arithmetic average of $(U_i|_P)_{N_r}$ and weighted average of $(U_i|_P)_{N_r}$, which yields:

$$U_i|_P = \sum_{r=1}^{8} w_r \cdot (U_i|_P)_{N_r}, \quad w_r = \frac{1/d_r^2}{\sum_{r=1}^{8} 1/d_r^2}, \tag{5}$$

where d_r is the Eulerian distance between particle position and node N_r [12]. Upon substitution of Eq. (2) into Eq. (5), one obtains:

$$\begin{aligned} U_i|_P \approx &\sum_{r=1}^{8} w_r \cdot (U_i|_{N_r}) + \sum_{r=1}^{8} w_r \left[\left.\frac{\partial U_i}{\partial x}\right|_{N_r}(X-x)_r\right] \\ &+ \sum_{r=1}^{8} w_r \left[\left.\frac{\partial U_i}{\partial y}\right|_{N_r}(Y-y)_r\right] \\ &+ \sum_{r=1}^{8} w_r \left[\left.\frac{\partial U_i}{\partial z}\right|_{N_r}(Z-z)_r\right], \end{aligned} \tag{6}$$

where $i = 1,2,3$. The terms on the right-hand side of Eq. (6) can be rewritten as:

$$\begin{aligned} U_i|_P \approx &\overline{U_i|_N} + \overline{\left.\frac{\partial U_i}{\partial x}\right|_N \cdot (X-x)} + \overline{\left.\frac{\partial U_i}{\partial y}\right|_N \cdot (Y-y)} \\ &+ \overline{\left.\frac{\partial U_i}{\partial z}\right|_N \cdot (Z-z)}, \end{aligned} \tag{7}$$

where $i = 1,2,3$ and the overbar denotes averaging over the eight corners of the host cell, N_r. Second-order terms have the following general form:

$$\overline{\frac{\partial}{\partial x_j}\left.\frac{\partial U_i}{\partial x_k}\right|_N (X_j - x_j)(X_k - x_k)},$$

where $j,k = 1,2,3$. Neglecting these terms, the truncation error is $O[(X_j - x_j)(X_k - x_k)]$, consistent with that of the numerical scheme for the flow solver.

For sake of completeness, the overall second-order accuracy Eqs. (2) and (5) will be also proved through benchmark numerical simulations of a particle tracker in the following sections. For such benchmarks, results (not shown) obtained using either arithmetic or weighted average are slightly less accurate than those obtained using interpolation about the nearest node, at the cost of a computational effort increased by more than a factor of eight. Considering also that the weighted-average procedure requires approximately $96M$ operations to evaluate $U_i|_P$ [10], the most suitable choice is to apply Eq. (4) to the nearest Eulerian grid point only.

Once the interpolated fluid velocity at particle location is obtained, the trajectory of a fluid particle is calculated by numerical integration of the particle equation of motion, $\partial\mathbf{X}(\mathbf{x},t)/\partial t = \mathbf{U}(\mathbf{x},t)$, subject to the initial condition $\mathbf{X}(\mathbf{x}_0,t_0) = \mathbf{X}_0$. In this study, an explicit second-order Adams–Bashforth time advancement scheme is used for the integration. With this choice, time-differencing errors can be assumed negligible even for lower temporal resolution [1]. However, the largest tracking time step permissible

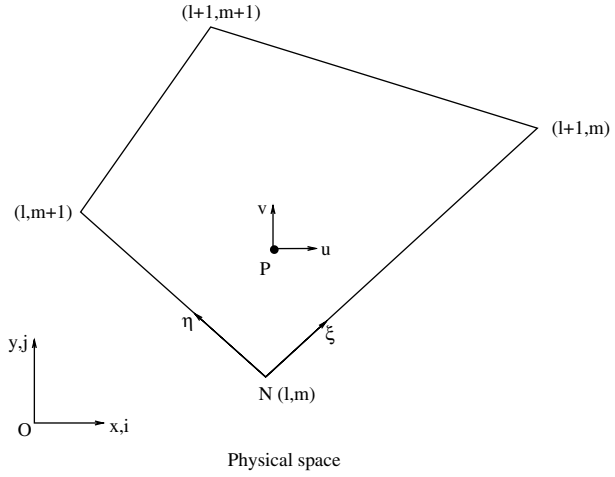

Fig. 1. Control volume of the curvilinear grid and the coordinate mapping in two dimensions.

is bound by stability requirements, which are fulfilled by monitoring the Courant number, Co, over all grid points in physical space. In our work, we computed the Courant number at Eulerian grid points as $Co = u_i \frac{\Delta t}{\Delta x_i}$, where u_i is the fluid velocity component in the ith direction, Δt is the tracking time step size and Δx_i is the grid spacing.

For finite-difference methods, numerical stability requires $Co < 0.5$ to have negligible time step dependence of the results [9]. This requirement becomes more severe when the grid spacing Δx_i is small, i.e. when refined grids are used.

Particle velocities $\mathbf{U}(\mathbf{x}, t)$ and positions $\mathbf{X}(\mathbf{x}, t)$ at selected sampling times are stored on Cartesian grid points and on curvilinear grid points for statistical post-processing and comparison purposes. The curvilinear grid is generated from the Cartesian grid (of the same resolution) by taking a wave shape in each direction as $X_{j,\mathrm{curv}} = X_{j,\mathrm{cart}} + 0.25 L_{X_j} \cdot \sin(\pi \cdot X_{j,\mathrm{cart}})$ where L_{X_j} is the dimension of the domain along the jth direction. This simple analytical formula allowed us to evaluate the performance of the interpolation scheme under severe cell deformation. In this condition, the off-diagonal elements of the transformation tensor $\partial \xi_k / \partial x_j$ are comparable to the diagonal ones and the interpolation procedure is much more complicated with respect to the case of Cartesian grids, in which the off-diagonal elements of $\partial \xi_k / \partial x_j$ are absent. Fig. 2 shows an example of Cartesian and curvilinear grids used for tests.

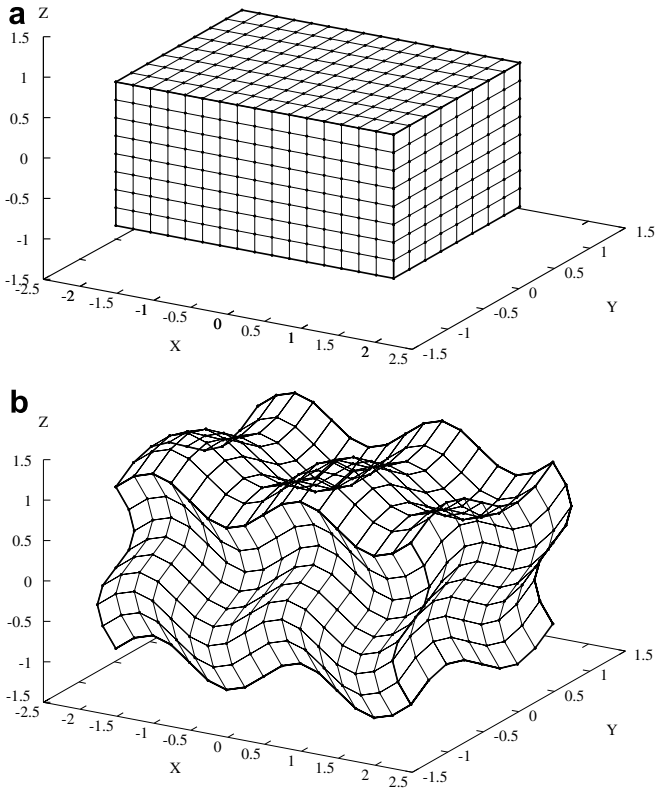

Fig. 2. Example of grids used for test simulations: (a) Cartesian, (b) curvilinear.

2.1. Test 1: sinusoidal flow

The performance of the scheme is assessed by comparing the error made when interpolating a prescribed velocity field for which particle paths are known analytically. Such a simple 2D test case is provided by the following velocity field:

$$\begin{aligned} U(x,y,z,t) &= \mathrm{d}X/\mathrm{d}t = U_0, \\ V(x,y,z,t) &= \mathrm{d}Y/\mathrm{d}t = 0, \\ W(x,y,z,t) &= \mathrm{d}Z/\mathrm{d}t = A \cdot \sin(\pi \cdot X). \end{aligned} \tag{8}$$

One fluid particle, representing a mathematical point moving with the local velocity of the fluid continuum, is introduced in this periodic, time-frozen (i.e. steady) flow: it is uniformly translated in the x-direction and moves sinusoidally in the horizontal plane x–z. The solution in terms of particle coordinates (X, Y, Z) at time t is given by:

$$\begin{aligned} X(t) &= X_0 + U_0 \cdot t, \\ Y(t) &= Y_0, \\ Z(t) &= Z_0 + \frac{A}{\pi \cdot U_0} [\cos(\pi \cdot X_0) - \cos(\pi \cdot X_0 + \pi \cdot U_0 \cdot t)], \end{aligned} \tag{9}$$

(X_0, Y_0, Z_0) being the initial particle position. The parameters of the prescribed flow are chosen as $U_0 = A = 0.025$ and the particle is initially placed at point $(-0.8, 0.0, 0.75)$. The computational box has dimensions $L_{X_j} = 2$ in each direction, discretized on both Cartesian and curvilinear grids made of 9^3, 17^3, 33^3, 65^3 and 129^3 nodes, respectively. The total tracking time is $t_{\mathrm{tr}} = 100$, long enough for the particle to cover a distance equal to one wavelength ($\lambda = 2$ length units). The time step used is $\Delta t = 0.1$ and the corresponding Courant number ranges from $Co_{\max} = 0.181$ for the 129^3 grid to $Co_{\max} = 0.01165$ for the 9^3 grid. We point out here that, due to the low computational cost of the test case simulations, the Courant number is not a concern. Rather, Δt is the controlling parameter. Thus, we chose values of Δt small enough to ensure (i) numerical stability at all times even for the smallest Δx_i considered and (ii) negligible time-stepping error with respect to the interpolation error. Once the *optimum* value of Δt had been chosen, it was maintained for all simulations. This explains why Co decreases to very small values[2] when coarser grids are used.

A similar test was run by Kontomaris et al. [1] to test a mixed polynomial-spectral scheme for fluid velocity interpolation in a spectral simulation of turbulent channel flow.

[2] The behavior of Adams–Bashforth methods, like the one used here to integrate Eq. (8), can become weakly unstable for periodic convection-dominated problems. We expect that present results for sinusoidal flow, obtained for rather small values of the Courant number, may be slightly affected by numerical diffusion introduced by the interpolation scheme. However, the error caused by instability should be relatively unimportant due to the small time step size adopted in the simulations.

The interpolation error affecting particle velocity is calculated by comparing the exact values of the velocity field against the interpolated values, both evaluated at the instantaneous position of the fluid particle. Thus, at each time step, the absolute local interpolation error is computed as $e_{p,U_i} = |U_{i,\mathrm{int}} - U_{i,\mathrm{exact}}|$ [1]. As a measure of the overall interpolation error the Root Mean Square (RMS) value of e_{p,U_i} is also computed:

$$\epsilon_{p,U_i} = \sqrt{\sum_{s=1}^{N_{\mathrm{ts}}} (e_{p,U_i})_s^2}, \tag{10}$$

where N_{ts} is the total number of time steps covered by the simulation. The interpolation error on particle displacement, ϵ_{p,X_i}, is calculated in a similar way.

Our aims are (i) to prove that the interpolation scheme is second-order accurate on curvilinear grids; and (ii) to isolate and characterize the effect of interpolation errors (i.e. the effects of grid spacing) on overall errors. We will not analyze the time-stepping error in detail, since it is much less significant than the interpolation error [1,9].

In Fig. 3, the interpolation RMS errors ϵ_{p,U_i} and ϵ_{p,X_i} (averaged over time) are plotted versus the grid size M. Error profiles for Cartesian grids (black squares) and curvilinear grids (empty squares) are considered. The behavior of the overall RMS error on particle velocity, $\epsilon_{p,U}$, provides evidence of the second-order accuracy of the Taylor-based velocity interpolation procedure. Similarly, the behavior of the overall RMS error on particle displacement, $\epsilon_{p,X}$, confirms the expected second-order accuracy of the particle tracking scheme.

In Fig. 4, we compare numerical and analytical prediction of the fluid particle trajectory. For clarity, a close-up in the range $-0.2 < X < 0.2$ is also shown (Fig. 4c and d). All different grid sizes are considered. With Cartesian grids (Fig. 4a and c), the scheme fails to follow the sinusoidal variation of the particle trajectory if the grid is coarse

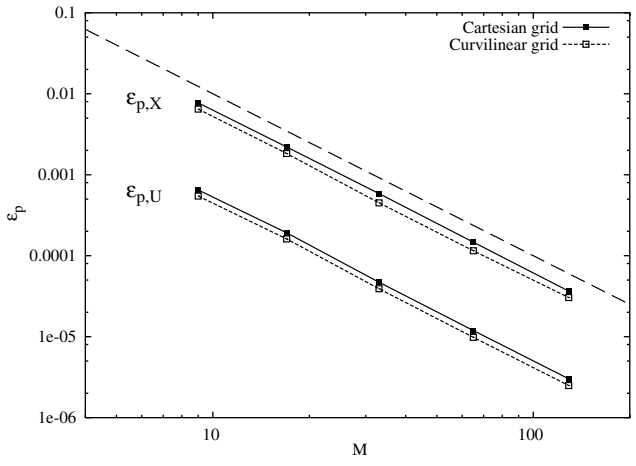

Fig. 3. Overall interpolation error (ϵ_p) versus grid size (M) for interpolation of sinusoidal velocity field. Symbols: (■) Cartesian grid, (□) curvilinear grid. Line: ––– slope −2.

(9^3 and 17^3 grids, dashed lines) while the analytic solution, represented by the solid line, is reproduced almost exactly with more refined grids (33^3 and up, dash-dotted lines). Similar considerations apply for the curvilinear grid (Fig. 4b and d).

Local interpolation errors in particle displacements have been quantified as $e^{\%}_{p,X_j} = 100 \cdot e_{p,X_j}/L_{X_j}$. For sinusoidal flow, we find errors for the Z-component of particle trajectory (i.e. $e^{\%}_{p,X_j} \equiv e^{\%}_{p,Z}$) only. Fig. 5 shows the effect of different grid resolutions on the time behavior of $e^{\%}_{p,Z}$. As expected, the largest errors appear in the low resolution simulation (9^3 grid) and the accuracy of the interpolation is significantly improved by increasing the resolution of the simulation: $e^{\%}_{p,Z}$, which is always less than 1% for the 9^3 grid, decreases by roughly three-orders of magnitude when the 129^3 grid is used. For Cartesian grids (lines in Fig. 5), $e^{\%}_{p,Z}$ exhibits a periodic behavior which repeats itself over time intervals of 64 time units, required for the particle to move horizontally from the initial $X = -0.8$ location to the $X = 0.8$ location with constant advection velocity U_0 (see Fig. 4a and b). The percentage error $e^{\%}_{p,Z}$ increases as the particle moves towards the $X = 0$ location (i.e. as the slope of particle trajectory decreases), where a maximum is reached. Conversely, $e^{\%}_{p,Z}$ decreases as the particle moves away from the $X = 0$ location (i.e. as the slope of particle trajectory increases). With curvilinear grids (symbols in Fig. 5), overall $e^{\%}_{p,Z}$ is slightly smaller even though, with adequate spatial resolution, $e^{\%}_{p,Z}$ profiles for Cartesian and curvilinear grid deviate only negligibly from each other even after a long integration time. Note also that error profiles are characterized by strong oscillations which are damped when the grid resolution is increased.

A careful examination of Eq. (4) reveals that the resulting approximating velocity function $U_i|_P$ does not possess the C^0 continuity property, i.e. the function is not continuous across the interfaces of neighboring interpolation cells [9]. This is confirmed by Fig. 6, in which the time behavior of particle velocity along the Z-direction, W, is shown for different resolutions of the curvilinear grid. When a coarse grid is used (9^3 grid, solid line), the calculated particle velocity does not change smoothly as the particle crosses an interface. Small jumps occur which contribute to numerical noise. Yet, they become significant only for poorly resolved velocity field (e.g. when a coarse grid is used) and can be damped if more than one corner node of the host cell is used for interpolation. For such reason, we consider this a tolerable, minor drawback. Similar conclusions on the C^0 continuity property of $U_i|_P$ apply to the analysis of local velocity interpolation errors (not shown).

2.2. Test 2: helical flow

As done by Yeung and Pope [9] to test the third-order Taylor series 13-point scheme on Cartesian grids, we benchmark our interpolation scheme on the steady 3D helical flow. In this flow, a given particle moves on a cylinder of radius r (measured from the axis of the cylinder), with

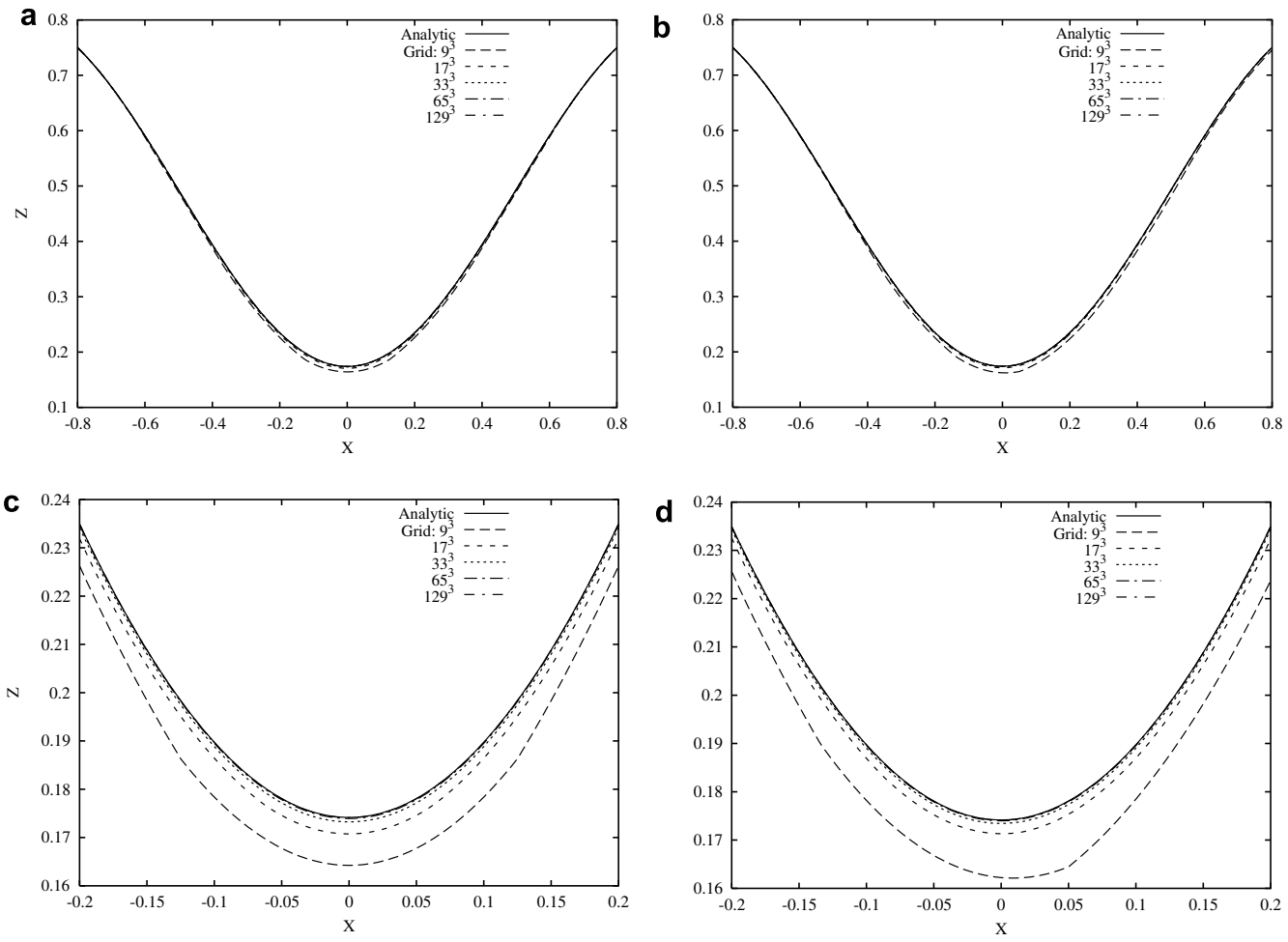

Fig. 4. Sinusoidal flow: effect of low grid resolution on fluid particle trajectory. (a) and (c): Cartesian grid, (b) and (d): curvilinear grid. The solid line represents the analytical (exact) solution for particle trajectory, all other lines (dashed, dotted and dash-dotted) represent the numerical (approximated) solution obtained using different grid resolutions.

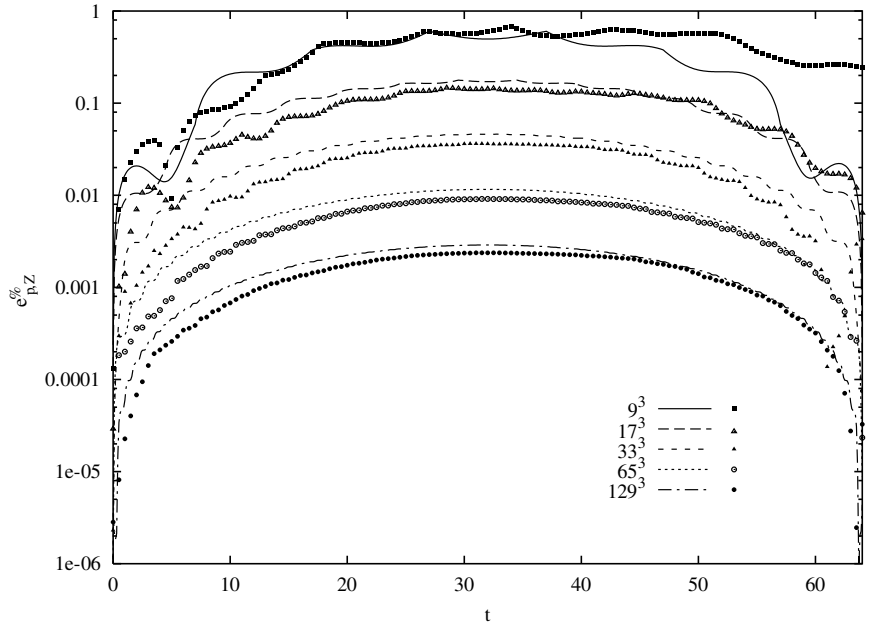

Fig. 5. Sinusoidal flow: time behavior of local interpolation error $e^{\%}_{p,Z}$ on particle trajectory for different grid resolutions. Lines refer to Cartesian grid, symbols refer to curvilinear grid.

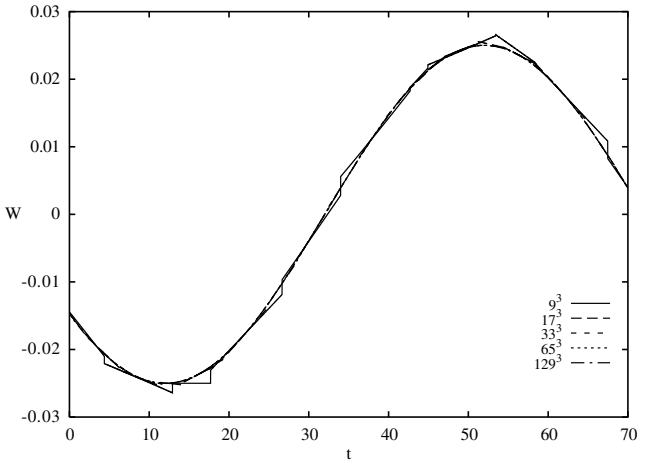

Fig. 6. Sinusoidal flow: effect of grid resolution on particle velocity (W component) as a function of time, t.

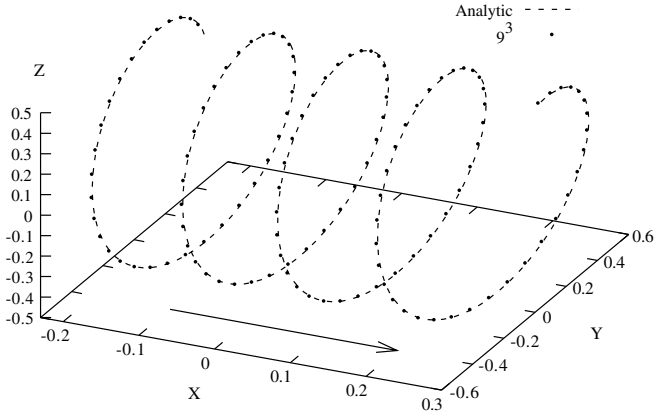

Fig. 7. Helical flow: perspective view of particle trajectory (9^3 curvilinear grid).

angular velocity $\omega(r)$. Particles lying within the cylinder move in axisymmetric circular motion in the y–z plane and in sinusoidal translational motion in the x-direction. The equations of motion are:

$$\begin{aligned} U(x,y,z,t) &= \mathrm{d}X/\mathrm{d}t = U_0 + 0.5\cdot A\cdot \sin\left(\frac{\pi}{4}X\right),\\ V(x,y,z,t) &= \mathrm{d}Y/\mathrm{d}t = -(Z-Z_0)\cdot\omega,\\ W(x,y,z,t) &= \mathrm{d}Z/\mathrm{d}t = (Y-Y_0)\cdot\omega. \end{aligned} \tag{11}$$

The solution trajectory in terms of particle coordinates (X, Y, Z) at time t is given by

$$\begin{aligned} X(t) &= X_0 + \frac{8}{\pi}\arctan\left[\frac{\sqrt{U_0^2-(0.5\cdot A)^2}\cdot\tan(\alpha\cdot t+\beta)-0.5\cdot A}{U_0}\right],\\ Y(t) &= Y_0 + \frac{r}{\omega}\cos(\omega\cdot t),\\ Z(t) &= Z_0 + \frac{r}{\omega}\sin(\omega\cdot t) \end{aligned} \tag{12}$$

with $(X_0, Y_0, Z_0) = (0, 0, 0)$ in the present case. Constants α and β are defined as follows:

$$\begin{aligned} \alpha &= \frac{8}{\pi}\sqrt{U_0^2-(0.5\cdot A)^2},\\ \beta &= \arctan\left[\frac{U_0\tan\left(\frac{\pi}{8}X_0\right)+0.5\cdot A}{\sqrt{U_0^2-(0.5\cdot A)^2}}\right]. \end{aligned} \tag{13}$$

Flow parameters are chosen as $U_0 = A = 0.02$ and the particle is initially placed at point $(-0.8, 0.0, 0.75)$. Grid sizes are the same as those for sinusoidal flow whereas the time step size is $\Delta t = 0.01$, with the corresponding Courant number ranging from $Co = 0.02$ for the 9^3 grid to $Co = 0.32$ for the 129^3 grid. The total tracking time is $t_{\mathrm{tr}} = 150$, long enough to cover more than 20 particle revolutions (time interval for one revolution: $\Delta t_{rev} = 2\pi$). This choice brings out all evolutionary features of particle motion and allows for the RMS particle displacements to reach a steady behavior.

Overall interpolation RMS errors ϵ_{p,U_i} and ϵ_{p,X_i} for helical flow, not shown, are roughly one-order of magnitude smaller than those for sinusoidal flow (see Fig. 3) and confirm the second-order accuracy of the velocity interpolation procedure and of the particle tracking scheme, respectively.

In Fig. 7, the numerically computed trajectory of the fluid particle on the coarse curvilinear grid (black circles) is compared to the exact trajectory predicted analytically (dashed line). For clarity, only few particle revolutions are shown. It is apparent that the interpolation scheme is able to reproduce the spiraling motion of the fluid particle accurately, even for low resolution simulations with very distorted cells.

To quantify further the accuracy of the scheme, in Fig. 8a we show a log–log plot of the time behavior of the local interpolation error $e^{\%}_{p,X}$ for the axial component of particle trajectory. Also shown is a close-up view (Fig. 8b) relative to a smaller time window ($0 < t < 150$). Lines refer to simulations with Cartesian grids, symbols refer to simulations with curvilinear grids. Several observations can be drawn from this figure, where the effect of different grid resolutions is considered.

First, percentage errors show a significant accumulation over time. In particular, the error grows roughly with the cube of time for the lower resolutions (9^3 and 17^3 grid points) when a Cartesian grid is used. However, this growth is limited to the initial stage of the simulation; at later times $e^{\%}_{p,X}$ is seen to increase linearly, which is to be expected. The increase of $e^{\%}_{p,X}$ with time appears to be linear for Cartesian grids with higher resolution and for all curvilinear grids tested in this study. As shown by the close-up of Fig. 8b, error profiles start to level off at the final stage of the simulation and seem to reach a saturation value (represented by the straight solid line). The rate of accumulation is higher for lower grid resolution, this being more evident for curvilinear grids.

Second, errors incurred in simulations with Cartesian grids are initially smaller than those incurred in simulations

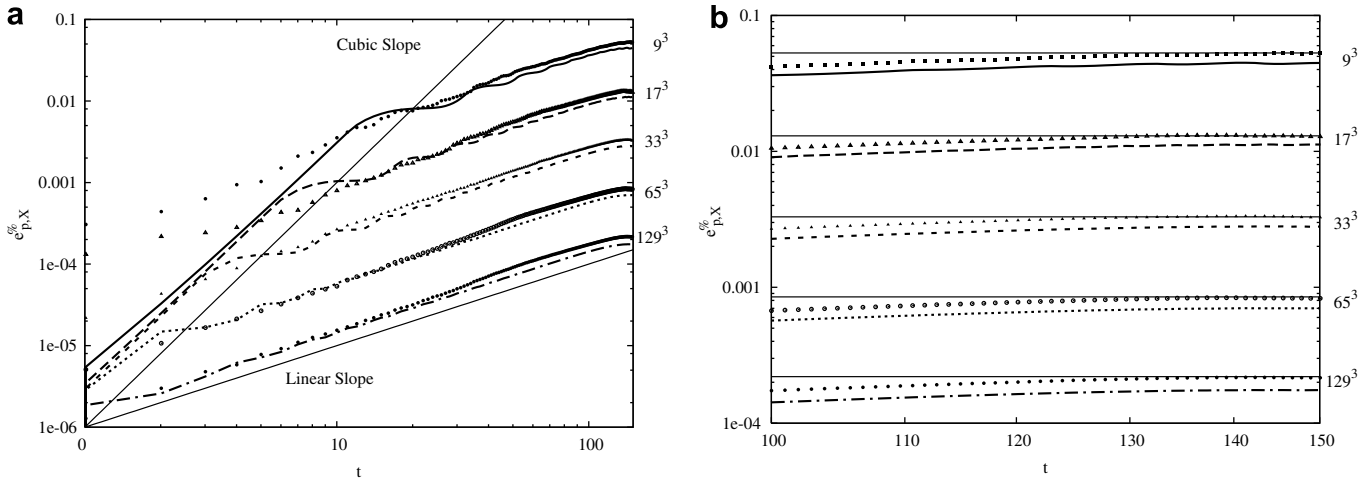

Fig. 8. Helical flow: local interpolation error $e^{\%}_{p,X}$ on particle trajectory versus time, t, for different grid resolutions. Lines refer to Cartesian grid, symbols refer to curvilinear grid.

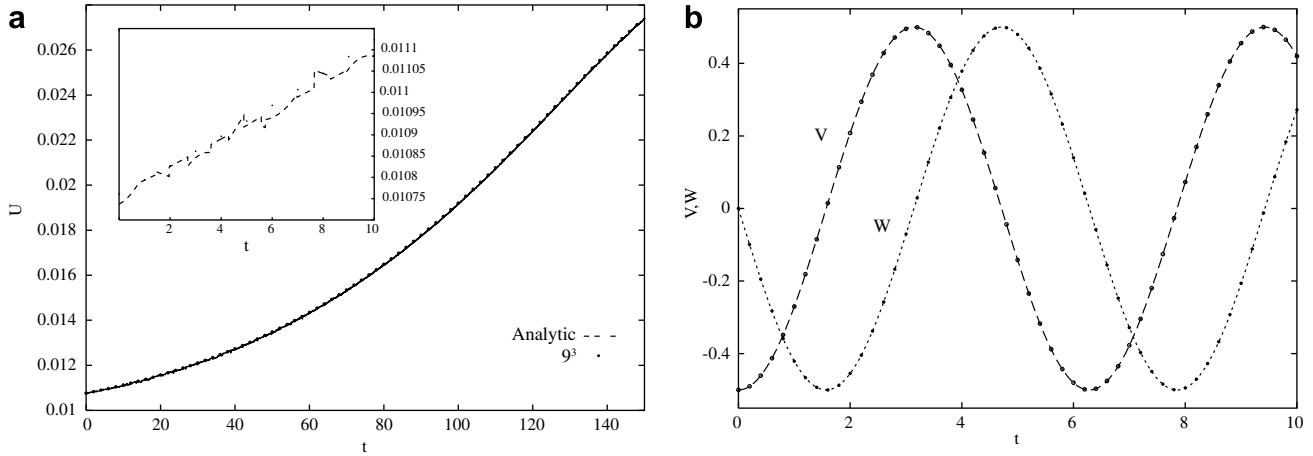

Fig. 9. Helical flow: (a) translational velocity, U, as a function of time, t, (b) components of particle tangential velocity, V and W, as a function of time, t. Symbols refer to analytic velocity, lines refer to interpolated velocity on the coarse grid (9^3).

with curvilinear grids, which appear to be more sensitive to the initial condition for integration of the equation of particle motion. For larger integration times and fixed number of grid points, error profiles deviate only negligibly from each other.

Third, it is confirmed that higher grid resolution is required to improve the interpolation accuracy. The local interpolation errors, $e^{\%}_{p,Y}$ and $e^{\%}_{p,Z}$ (not shown), also accumulate over time but remain lower than 0.1% at the end of the simulation.

Fig. 9 shows the time behavior of particle translational velocity, U, along the X-direction (Fig. 9a) and of particle tangential velocity components, V along the Y-direction, and W along the Z-direction (Fig. 9b) in the case of 9^3 curvilinear grid (from a visual viewpoint, negligible difference exists with profiles obtained using the corresponding Cartesian grid). Components V and W vary sinusoidally: for clarity of presentation, we show their behavior during the first 10 time units of the simulation only. In Fig. 9, lines and symbols refer to analytic and interpolated velocity, respectively. The three velocities are reproduced with good accuracy even for low grid resolution. Yet, an inadequate number of grid points yields an approximated velocity which is not C^0 continuous (see close-up in Fig. 9a). The local interpolation errors on particle velocities are $e^{\%}_{p,U} = 100 \cdot e_{p,U}/U_0$, $e^{\%}_{p,V} = 100 \cdot e_{p,V}/V_{\tan}$ and $e^{\%}_{p,W} = 100 \cdot e_{p,W}/V_{\tan}$. During the simulations, $e^{\%}_{p,U}$ oscillates reaching a maximum value approximately equal to 0.2% for the 9^3 grid and to 0.0009% for the 129^3 grid. The approximated particle tangential velocity, $V_{\tan} = (V^2 + W^2)^{0.5}$, coincides with the exact analytic value and the test particle follows exactly the analytical trajectory. Yet, a small, but finite, error is incurred in the calculation of its components, which accumulates over time: at the end of the simulation, $e^{\%}_{p,V}$ and $e^{\%}_{p,W}$ are lower than 0.6%.

3. Application to particle dispersion in turbulence over waves

To the best of our knowledge, interpolation of fluid velocity was not performed on curvilinear grids in previous

studies dealing with Lagrangian tracking of particles dispersed in turbulent flows over wavy walls [20,21]. Boersma [20] tracked particles in the computational rectangular space using quadratic interpolation of 27 neighboring velocity points. More recently, Chang and Scotti [21] discretized the flow domain on a rectangular grid exploiting the immersed boundary technique and used sixth-order Lagrangian polynomials.

Problem formulation and numerical method are briefly described in the following. Attention is paid only to aspects relevant to the present work: more details can be found elsewhere [8,15,16]. A statistically stationary and fully developed turbulent flow over fixed 2D sinusoidal waves laden with large swarms of heavy particles is considered. The fluid is air, entering the domain with bulk velocity (namely, the average velocity computed along the height of the channel) $U_b = 1.22\ \mathrm{m\ s^{-1}}$. The corresponding bulk Reynolds number is $Re_b = U_b H/\nu = 3108$ where $H = 0.04$ m is the channel height and ν is the fluid kinematic viscosity. The flow is driven by a constant mean streamwise pressure gradient. The domain size is $2\lambda \times \frac{4}{3}\pi H \times H$ along the streamwise, x, the spanwise, y, and the vertical direction, z, respectively. The wave shape is taken as $z(x) = a\cos(kx)$ where $a = 0.005$ m is the wave amplitude, $k = 2\pi/\lambda$ is the wave number and $\lambda = 0.06$ m is the wave length. The corresponding slope is $ak \simeq 0.5236$, high enough to ensure flow separation in the trough of the wave.

For the incompressible flow considered here, the governing equations (Navier–Stokes and continuity) are solved using the finite-difference solver by Zang et al. [15], imposing free-slip stress-free conditions at the upper boundary, periodic boundary condition at the box sides, no-slip condition at the lower wavy wall. Equations are transformed into a general curvilinear coordinate system and discretized on a co-located grid using a finite volume approach [16]. The Navier–Stokes solver uses two sets of variables, defining velocities and pressure at cell centers and contravariant volume fluxes at the cell faces. In the present study, fluid velocities at cell centers are used directly to track particles: no additional interpolation is required and no further smoothing of the velocity signal is introduced. Overall, the algorithm is second-order accurate both in space and time.

The computational grid has $64 \times 64 \times 48$ nodes in x, y and z, respectively. It is uniform in x and y while, in the vertical direction, nodes are clustered near the wavy wall. The grid discretization corresponds to a resolution $\Delta x^+ = 7.9$, $\Delta y^+ = 11.0$ and Δz^+ ranging from 1.6 to 7.9 in non-dimensional wall units (represented by the superscript +). Note that the non-dimensional length of the wavy channel is $L_x^+ = 510$: this length is longer than a minimal channel flow [22] and adequate to capture the dynamical trends of the flow; yet it is short for simulating a developed flow. There will be several levels of sub-harmonic components to the flow relative to the wavelength which allow investigation of particle dynamics in connection with coherent flow structures of different temporal and spatial scales. This is precisely the purpose of our study. In particular, we focused on the geometry-dependent long-lived flow structures located in the wave trough, and on the quasi-streamwise vortices, which grow up on the upslope side of the wave and affect particle dynamics on smaller scales [8].

Calculation of the Eulerian flow field is coupled with Lagrangian tracking of individual particle trajectories. We simulated heavy particles characterized by different Stokes numbers ($St = 0.5$, 1 and 2). In this work, St is equal to the non-dimensional particle response time, τ_p^+, defined as the ratio of the dimensional particle response time, $\tau_p = \rho_p d_p^2/18\mu$, to the fluid time scale, $\tau_f = \nu/u_\tau^2$. Here, ρ_p and d_p are particle density and diameter, ν is the fluid kinematic viscosity, $u_\tau = (\tau_w/\rho)^{0.5}$ is the shear fluid velocity, based on the shear stress at the wall, τ_w, and on fluid density, ρ.

The trajectory of each particle (treated as a non-interacting, non-deformable solid sphere) is computed under the assumption of dilute flow system conditions by integrating over time the following equation of particle motion in vector form:

$$\frac{d\mathbf{v}_p}{dt} = \frac{3}{4}\frac{\hat{\rho}}{d_p} C_D(\mathbf{v}-\mathbf{v}_p)|\mathbf{v}-\mathbf{v}_p| + (1-\hat{\rho})\mathbf{g} - \xi(\epsilon)\frac{6.46}{12\pi}\frac{d_p}{\tau_p}\left|\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right|^{0.5} \mathrm{sign}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)(\mathbf{v}-\mathbf{v}_p)\cdot\mathbf{n}, \tag{14}$$

where $\mathbf{v}_p$ is particle velocity, $\hat{\rho}$ is the ratio of fluid density to particle density, $\mathbf{v}$ is fluid velocity, $\mathbf{g}$ is gravity, $\mathbf{n}$ is the unit vector in wall-normal direction. $C_D = f(Re_p)$ is the Stokes coefficient for drag, which depends on the particle Reynolds number, $Re_p = \rho d_p|\mathbf{v}-\mathbf{v}_p|/\mu$. We use the following nonlinear correction for C_D [23] when $Re_p > 1$:

$$C_D(Re_p) = \frac{24}{Re_p}\left(1 + 0.15\cdot Re_p^{0.687}\right). \tag{15}$$

For the simulation parameters considered in the present study, only particle inertia, Stokes drag, buoyancy and lift have been taken into account. The lift force, represented by the third term on the right-hand side of Eq. (14), is formulated here according to the model proposed by Saffman [24] and corrected by McLaughlin [25,26] for a small spherical particle in linear shear flows including an additional correction factor that becomes important when the relative velocity between the particle and the fluid is large. This expression requires less restrictive assumptions with respect to the original Saffman formulation but still neglects wall effects: when the distance of the particle from the bottom wall becomes small compared to particle radius, the actual mechanism of deposition is complicated by the possible rise of surface related phenomena and the formulation without wall effects is not very accurate [26].

Eq. (14) is integrated with an explicit second-order modified Euler method and the integration time step for all particles is one tenth of the characteristic time of the smallest particles ($St = 0.5$). Particles are elastically

reflected away from the wall when their center is less than a distance $d_p/2$ from the boundary. The elastic collision of particles with the lower wall will produce physical effects of its own and discontinuities to the motion. During the particle tracking, however, we recorded particle position and velocity after each impact with the lower wall: this way, a sub-set of data relative to the condition of perfectly-absorbing wall can be extracted from the data base relative to the condition of perfectly-reflecting wall.

Here, we will discuss some relevant results on the dispersion of heavy particles to highlight how this phenomenon is affected by the choice of the interpolation scheme, in terms of individual particle trajectories and Lagrangian statistical quantities. Comparison will be made between results obtained using the proposed interpolation scheme and results obtained using trilinear interpolation to highlight the most relevant performance and sensitivity issues. Trilinear interpolation requires the cell indices (l,m,n) and the offsets (ϵ,η,γ) obtained through point location. Let the basis function Ψ be defined as $\Psi_0(\epsilon)=(1-\epsilon)$ and $\Psi_1(\epsilon)=\epsilon$. Then, for a point in a cell (l,m,n) with offsets (ϵ,η,γ), as point P shown in Fig. 1 for the 2D case, the function Φ determines the interpolated value from the eight corner velocities $\mathbf{v}_{l,m,n},\ldots,\mathbf{v}_{l+1,m+1,n+1}$:

$$\mathbf{v}=\Phi(\mathbf{v},\epsilon,\eta,\gamma)=\sum_{L,M,N=0}^{1}\mathbf{v}_{l+L,m+M,n+N}\cdot\Psi_N(\epsilon)\Psi_M(\eta)\Psi_N(\gamma), \tag{16}$$

where indices $L,M,N\in\{0,1\}$.

Fig. 10 shows the trajectory of three sample $St=1.0$ particles (labeled A, B and C) originating from different locations of the channel. These trajectories represent worst case results and are indicative of an upper limit for the particle displacement produced by the two schemes we are comparing. Particles were tracked for a time span equal to $510t^+$ (superscript $+$ indicates non-dimensional wall units), assuming they are absorbed at the wall upon impact. The solid lines in Fig. 10 represent trajectories calculated using the present Taylor-based interpolation scheme; the dash-dotted lines represent trajectories calculated using standard trilinear interpolation. It is apparent that the choice of a specific interpolation scheme may have an important effect on individual particle trajectories: for all three particles, trajectories are identical initially and virtually collapse onto each other as long as the particle is relatively far from the bottom wall. As particles approach the wall, where the grid is more distorted, differences in particle displacements become non-negligible and accumulate over time, producing different trajectories. For instance, particle A will deposit on the upslope part of the wave before the fifth crest, roughly at a non-dimensional streamwise location $X^+\simeq 1000$, if the present scheme is used (solid line). When trilinear interpolation is used, the same particle will not deposit at $X^+\simeq 1000$: it will be re-entrained by the boundary layer which develops on the upslope part of the wave and then lifts away as a free-shear layer, eventually re-ejecting the particle towards the outer flow (dash-dotted trajectory). Particles B and C follow a similar destiny. Trilinear interpolation is more diffusive and can produce less accurate estimates of particle trajectories. Yet, large discrepancies in particle displacements are found only for a relatively small number of tracked particles: individual particle trajectories remain reasonably close in most of the cases. Also, the location of particle deposition computed with either scheme is comparable in the majority of the cases.

In a dispersion process, however, the details of the Lagrangian trajectories of single particles are not usually of major significance; the evaluation of the statistical properties of the process is generally more important. Lagrangian statistical quantities averaged over many particles will be less sensitive than individual particle trajectories to the choice of the interpolation scheme. Here, we will restrict

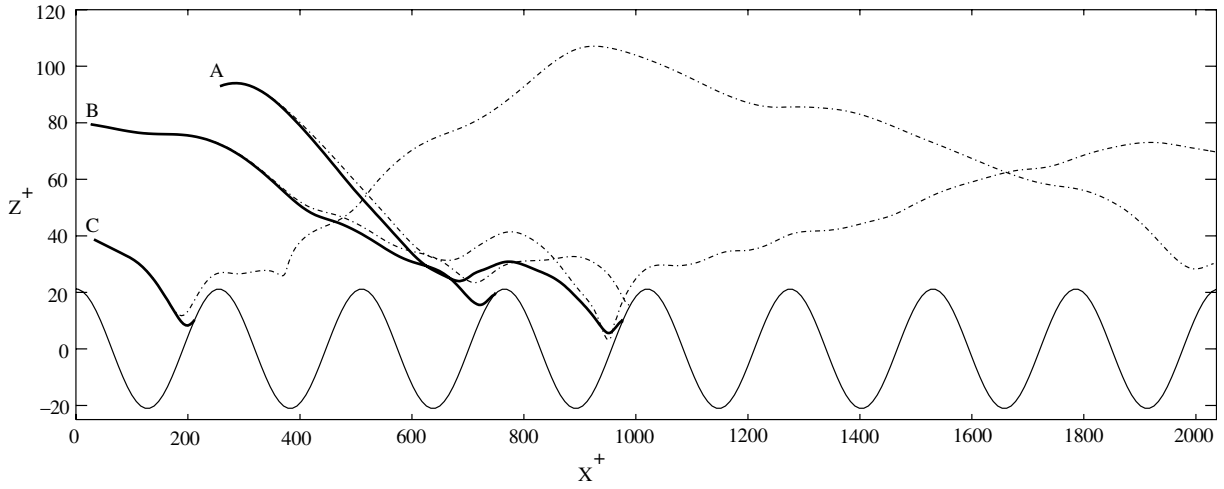

Fig. 10. Displacement errors on the trajectory of three sample particles A, B and C released in turbulent wavy channel flow for different interpolation schemes: —— present, –·–·– linear interpolation.

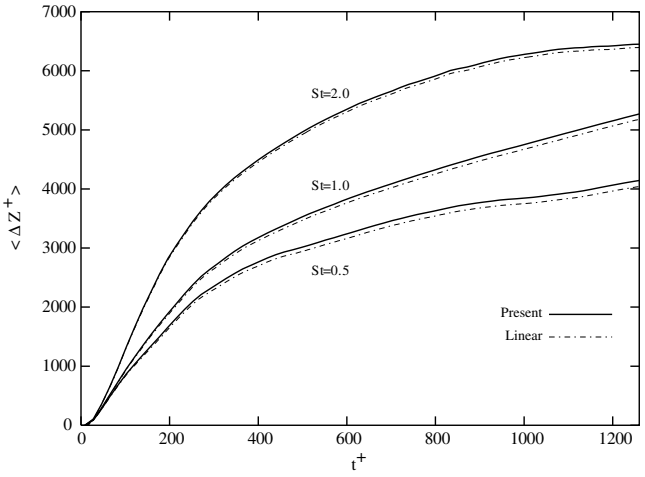

Fig. 11. Time evolution of the mean square dispersion, $\langle \Delta Z^+ \rangle$, of $O(10^5)$ particles released in turbulent wavy channel flow and traced using different interpolation schemes: —— present, –·–·– linear interpolation. Three different particle sets (characterized by Stokes number, St, equal to 0.5, 1.0 and 2.0, respectively) are considered. Time t^+ is made non-dimensional with u_τ^2/ν where u_τ is the friction velocity.

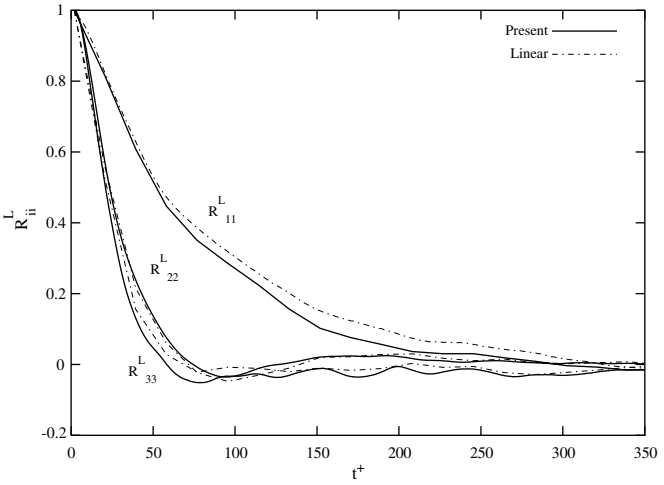

Fig. 12. Time evolution of Lagrangian velocity autocorrelations, R_{ii}^{L}, for $St = 1.0$ particles using different interpolation schemes: —— present, –·–·– linear interpolation. Time t^+ is made non-dimensional with u_τ^2/ν where u_τ is the friction velocity.

the comparison to Mean Square Dispersion (MSD) and Lagrangian particle-velocity autocorrelations, which are among the one-particle statistics that quantify the characteristics of a particle-laden flow.

In Fig. 11, the time evolution of the MSD in the vertical direction for particles tracked using the present interpolation scheme (solid line) is compared to that of the MSD obtained using standard trilinear interpolation (dash-dotted line). All particle sets have been considered here. Following Kontomaris et al. [1], the dispersion tensor is defined as:

$$D_{ij}(t - t_0) = \langle X_i'(\mathbf{X}_0, t) \cdot X_j'(\mathbf{X}_0, t) \rangle, \tag{17}$$

where brackets are used to indicate averaging over all tracked particles and $X_i'(\mathbf{X}_0, t) = X_i(\mathbf{X}_0, t) - \langle X_i(\mathbf{X}_0, t) \rangle$ is the ith component of the displacement fluctuation vector. MSD profiles shown in Fig. 11 are identical initially and remain reasonably close even after a long simulation time. Small discrepancies arise which accumulate over time: linear interpolation seems to overestimate particle dispersion with respect to the proposed scheme, especially in the case of smaller particles.

Fig. 12 compares Lagrangian velocity autocorrelations for $St = 1.0$ particles, computed with either scheme. The Lagrangian velocity autocorrelation is a statistical measure of the coherent structures encountered by an ensemble of particles during their motion through the fluid. Here, it is defined as in Kontomaris et al. [1]:

$$R_{ij}^{\mathrm{L}}(t_0, s) = \frac{\langle U_i'(\mathbf{X}_0, t_0) \cdot U_j'(\mathbf{X}_0, t_0 + s) \rangle}{U_{i,\mathrm{RMS}}(t_0) \cdot U_{j,\mathrm{RMS}}(t_0 + s)}. \tag{18}$$

Fig. 12 suggests that trilinear interpolation (see dash-dotted lines) overestimates the velocity autocorrelation with respect to the proposed scheme (see solid lines). Being based on Taylor series expansion, our scheme uses a larger number of grid points for interpolation of fluid velocity: thus, it appears more effective in capturing the decorrelating effect of the small scales of motion [1], particularly in the streamwise and vertical directions (compare R_{11}^{L} and R_{33}^{L} profiles, respectively).

4. Summary and conclusions

A second-order accurate interpolation scheme based on Taylor series expansion of the local fluid velocity about the grid point nearest to the desired location has been developed. The scheme is best suited for 3D curvilinear grids (either of the finite-volume or finite-element type) with non-orthogonal computational cells. In the finite element terminology, this scheme can be viewed as similar to finite element interpolation within a piece-wise linear element using shape functions: in principle, such procedure should work for any geometry, not limited to the curvilinear systems. The interpolation scheme attains adequate numerical accuracy if applied within the framework of highly resolved Eulerian numerical simulations (DNS or LES) of particle-laden turbulent flows. The interpolation error, which appears to be the major source of discrepancy with respect to the analytic values, can be reduced significantly by increasing the grid spatial resolution. Higher spatial resolution is also important when C^0 continuous approximations are a must: as the particle moves across interpolation cell boundaries, jumps in the approximated velocity occur which can be dramatically damped using a reasonably adequate number of grid points.

In conclusion, the reasons why we believe that the proposed scheme adds to the state-of-the-art of Eulerian–Lagrangian computations of turbulent dispersed flows can be summarized as follows:

- the present interpolation scheme is formally accurate at least at the order of accuracy of the Navier–Stokes solver;
- the scheme is conceptually simple and easy to implement;
- the scheme is specifically developed for boundary-fitted curvilinear coordinates and highly-deformed, non-orthogonal grid cells;
- the scheme does not use any geometrical algorithm based on the shape of the computational cell in the physical domain;
- the use of this scheme in Eulerian–Lagrangian computations of turbulent dispersed flows over complex geometries, typical of engineering applications, provides sufficiently accurate results (more accurate than those obtained using linear interpolation, in our opinion);
- the scheme is not CPU-time expensive (computational overhead required to evaluate the fluid velocity derivatives and storage allocation are small even when a large number of particles – $O(10^5)$ or more – is considered): this avoids overincreased computational cost of already expensive simulations, such as DNS or LES.

To test the performance of the proposed scheme, an application to particle deposition and resuspension in Large Eddy Simulation of the turbulent wavy channel flow [8] has been presented. This type of flow boundary imposes the use of highly-distorted cells to discretize the near-wall region and, thus, has been chosen to represent benchmark in the context of Eulerian–Lagrangian simulations of dispersed flows in three-dimensional curvilinear grids.

Acknowledgements

Financial support from MIUR under Grants no. RBAU012FRS (FIRB program) and 2003099224_002 (PRIN program) and from Regione Friuli Venezia-Giulia under Grant *Fluidodinamica e Analisi delle Dispersioni nella Bassissima Atmosfera del Friuli Venezia-Giulia*. C.M. thankfully acknowledges Regione Friuli Venezia-Giulia for financial support under Grant PORO3 FVG 2000–2006 – Asse D – Misura D4 and Dipartimento di Ingegneria Civile e Ambientale, Universita' di Trieste, for the support received during this study.

References

[1] Kontomaris K, Hanratty TJ, McLaughlin JB. An algorithm for tracking fluid particles in a spectral simulation of turbulent channel flow. J Comput Phys 1992;103:231–42.
[2] Pan Y, Banerjee S. Numerical simulation of particle interactions with wall turbulence. Phys Fluids 1996;8:2733–55.
[3] Uijttewaal WSJ, Oliemans RVA. Particle dispersion and deposition in direct numerical and large eddy simulations of vertical pipe flows. Phys Fluids 1996;8:2590–604.
[4] Armenio V, Piomelli U, Fiorotto V. Effect of the subgrid scales on particle motion. Phys Fluids 1999;11:3030–42.
[5] Marchioli C, Soldati A. Mechanisms for particle transfer and segregation in a turbulent boundary layer. J Fluid Mech 2002;468:283–315.
[6] Giusti A, Lucci F, Soldati A. Influence of the lift force in direct numerical simulation of upward/downward turbulent channel flow laden with surfactant contaminated microbubbles. Chem Eng Sci 2005;60:3742–52.
[7] Soldati A. Particles turbulence interactions in boundary layers. ZAMM – J Appl Math Mech 2005;85:683–99.
[8] Marchioli C, Armenio V, Salvetti MV, Soldati A. Mechanisms for deposition and resuspension of heavy particles in turbulent flow over wavy interfaces. Phys Fluids 2006;18:025102.
[9] Yeung PK, Pope SB. An algorithm for tracking fluid particles in numerical simulations of homogeneous turbulence. J Comput Phys 1988;79:373–416.
[10] Balachandar S, Maxey MR. Methods for evaluating fluid velocities in spectral simulations of turbulence. J Comput Phys 1989;83:96–125.
[11] Kenwright DN, Lane DA. Interacting time-dependent particle tracing using tetrahedral decomposition. IEEE Trans Visual Comput Graph 1996;2:120–9.
[12] Sadarjoen IA, van Walsum T, Hin AJS, Post FH. Particle tracing algorithms for 3D curvilinear grids. In: Nielson GM, Muller H, Hagen H, editors. Scientific visualization: overviews, methodologies, and techniques. IEEE CS Press; 1997, ISBN 0-8186-7777-5.
[13] Oliveira PJ, Gosman AD, Issa RI. A method for particle location and field interpolation on complex, three-dimensional computational meshes. Adv Eng Softw 1997;28:607–14.
[14] Lehnhauser T, Schafer M. Improved linear interpolation practice for finite-volume schemes on complex grids. Int J Numer Methods Fluids 2002;38:625–45.
[15] Zang Y, Street RL, Koseff JR. A non-staggered grid fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. J Comput Phys 1994;114:18–33.
[16] Zang Y, Street RL. A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. Int J Numer Methods Fluids 1995;20:341–61.
[17] Patankar NA, Joseph DD. Modeling and numerical simulation of particulate flows by the Eulerian–Lagrangian approach. Int J Multiphase Flow 2001;27:1659–84.
[18] Chen XQ. Efficient particle tracking algorithm for two-phase flows in geometries using curvilinear coordinates. Numer Heat Transfer 1997;32:387–405.
[19] Zhou Z, Leschnizer M. An improved particle-locating algorithm for Eulerian–Lagrangian computations of two-phase flows in general coordinates. Int J Multiphase Flow 1999;25:813–25.
[20] Boersma BJ. Particle distribution in the flow over a wavy wall. Proc Summer School, Center for Turbulence Research (CTR), 2001, Stanford University, 2000. p. 109–17.
[21] Chang YS, Scotti A. Entrainment and suspension of sediments into a turbulent flow over ripples. J Turbulence 2003;4:1–22.
[22] Jimenez J, Moin P. The minimal flow unit in near-wall turbulence. J Fluid Mech 1991;225:213–40.
[23] Rowe PN, Enwood GA. Drag forces in hydraulic model of a fluidized bed – Part I. Trans Inst Chem Eng 1962;39:43–7.
[24] Saffman PG. The lift on a small sphere in a slow shear flow. J Fluid Mech 1965;22:385–400. Corrigendum, 31, 1968. p. 624.
[25] McLaughlin JB. Inertial migration of a small sphere in linear shear flows. J Fluid Mech 1991;224:261–74.
[26] McLaughlin JB. The lift on a small sphere in wall-bounded linear shear flows. J Fluid Mech 1993;246:249–56.