mathematics

MDPI

Article

Evaluation of Interpolation Scheme Alternatives and Variation of the Number of Slave Cells in a Cut-Cell Methodology

Luis Henríquez-Vargas *, Francisco Angel, Lawrence Lackey and Pablo Donoso-García

Departamento de Ingeniería Química, Universidad de Santiago de Chile, Santiago 9170022, Chile; francisco.angel@usach.cl (F.A.); lawrence.lackey@usach.cl (L.L.); pablo.donosoga@usach.cl (P.D.-G.)

* Correspondence: luis.henriquez@usach.cl

Abstract: The present work determines numerical solutions applied to flow problems in a cut-cell framework, introducing and evaluating two interpolation alternatives for the treatment of the convective terms and the effect of the variation of the number of slave cells generated near the solid interfaces. Using the upwind, QUICK and WAHYD (TVD) schemes, three benchmark cases were studied in the laminar regime, namely, flow between concentric cylindrical walls, flow in an inclined channel and flow around a cylinder. The numerical results obtained were favorable for the proposed interpolation methodology that prevents velocity over/under-estimations on the finite control volume faces, observing a tendency to produce smaller errors and mid-to-high computational efficiencies when coupled with a smaller number of slave cells generated at the boundaries. Although the magnitude of the errors found were small, improvements are of more significance for quantities that depend on gradient estimations at surfaces.

Keywords: interpolation scheme; cut cell; immersed solid; Navier–Stokes; numerical solution

MSC: 65M08

check for updates

Citation: Henríquez-Vargas, L.; Angel, F.; Lackey, L.; Donoso-García, P. Evaluation of Interpolation Scheme Alternatives and Variation of the Number of Slave Cells in a Cut-Cell Methodology. *Mathematics* **2022**, *10*, 895. https://doi.org/10.3390/math10060895

Academic Editor: James M. Buick

Received: 10 February 2022
Accepted: 9 March 2022
Published: 11 March 2022

1. Introduction

Within the framework of CFD, the use of structured grids to represent complex geometries is a convenient approach to overcome computational accuracy problems and complex grid-generation processes presented in unstructured and/or boundary-conforming grids [1–3]. Berger et al. [4] cite among the strengths of Cartesian embedded-boundary grid schemes their accuracy, rapid turnaround time and level of automation.

Cartesian grid schemes can be classified based on their geometrical approximation quality. For this, the lowest order method is the Voxel method, in which the boundary is represented in a staircase fashion [3]. Higher order methods comprise the immersed boundary and cut-cell methods, among others. In the former, the influence of solid boundaries is modeled by the introduction of additional terms and/or velocity interpolations to force the no-slip internal boundary condition. Consequently, the conforming grid cells maintain their integrity, as special treatments make the presence of the interfaces felt at the boundary cells [5–10]. For a discussion on the classification of immersed boundary methods, the reader is referred to Mittal [1].

On the other hand, in the cut-cell approach, boundary conforming grid cells are truncated to better approximate the immersed solid's geometry. Special treatment is only required for the boundary cells while the rest of the grid is treated with the usual Cartesian methodology, achieving strict global and local conservation of transported properties. As for weaknesses, these schemes present numerical issues of discretization, stiffness and convergence for the small and irregular cells [4,11]. There has been extensive development in cut-cell methods and their applications, ranging from the 2D Euler equations to viscous compressible flow with conjugate heat transfer, using sophisticated adaptive and multigrid schemes for fixed and moving boundaries [2,4,12,13].

Three approaches can be found in the literature regarding the computation of the resulting small cells: (i) merging small onto larger cells [11,13–15]; (ii) flux redistribution [16]; (iii) cell linking in a so-called master–slave relation [17–19].

As for the cell linking procedure proposed by Kirkpatrick et al [17], it addressed the problem of numerical stiffness derived from the creation of small cells by introducing the concept of master-slave cell linking. A slave cell can be identified by having only one pressure node; this criteria limits the minimum cell length in the flow direction (must be greater than half the uncut cell width), thus dealing with the problem of having abrupt changes of the Courant number between regular cells and small (cut) cells. Moreover, main grid cells that have less than 1% of the original face are discarded from computation and the associated velocity cells are treated as slave cells. Fluid cells which are nearest to the normal vector erected from the boundary are designated as master cells. Conveniently, this cut-cell methodology largely unifies the computation process for all classes of cells and avoids the problems generated by the merging of small cells onto larger cells that had been the standard procedure. Among the problems related to the cell-merging procedure, the authors cited the need to compute additional fluxes and the difficulties presented in 3D geometry to formulate the algorithm in a systematic way. Results included 2D cavity-driven flow with an immersed cylinder, channel-skewed flow, flow past cylinder (Re = 40) and 3D flow past a hemispherical body. With regard to the 2D flow past a cylinder results, the authors informed the drag coefficient, separation angle and the pressure distribution represented by the pressure coefficient as a function of the angle and the number of cells alongside the cylinder diameter (grid resolution). Interpolation of variables on the cylinder surface such as pressure and vorticity was carried out using fourth-order tension splines. Comparison was favorable with experimental data of [20], and surpassed the Voxel method, especially at lower grid resolutions.

The cut-cell methodology implementation of the open-source software MFIX, follows closely the one of Kirkpatrick et al. [17], and is discussed in [21–23]. Treatment for small pressure cells translates in their removal, considering slight alterations of the original geometry by snapping the intersection point of the curve and underlying grid. The snapping procedure is applied when the ratio of the intersection to vertex length to cell edge is 1–5% of the cell edge. A cell is considered small when its volume over the uncut volume is lower than a user-defined tolerance of about 1%. Small pressure cells remaining after the snapping procedure are removed from computation. Since MFIX is a multiphase simulation software, extension of the cut-cell methodology included wall boundary conditions in the form of no-slip, partial-slip and free-slip. Applications presented included 2D/3D simulations of circulating fluidized beds, bubbling fluidized beds with horizontal tube bundle and a cyclone.

Recently, Xie and Stoesser [24] used an implicit time integration procedure to numerically solve an unsteady, turbulent, incompressible, two-phase flow configuration with immersed moving bodies on a three-dimensional Cartesian grid. Their time integration procedure allowed them to avoid the common instability problems found in small cut cells. In other words, small cells found in the cut-cell procedure were not treated, which reported supra linear convergences for the benchmark case of a 3D dam-break flow when using a high-resolution scheme for the advection flow.

In the present work, we revisited the cut-cell approach that leaves the small cells generated out of the computation, evaluating and presenting now two alternatives for the face interpolations of the velocity components in the fluid cells near the solid boundary within an implicit scheme. Moreover, we studied the numerical solution dependency on the solid body-resolved geometry representation with the number of master and slave cells generated for a given mesh size. Three benchmark cases for laminar incompressible flow were studied: flow between concentric walls, flow on a inclined channel and flow around a circular cylinder. We expected to empirically find the error and computational efficiency trends associated with the number and distribution of untreated slave cells generated and how these results may vary with the interpolation approaches studied for the upwind, QUICK and WAHYD numerical schemes. The paper is summarized as follows: Section 2 presents the mathematical model; Section 3 provides insight into the numerical technique employed and illustrates the problematic cell cases and the interpolation schemes, and criteria for defining small cells is given with the aid of a parameter definition; Section 4 presents results and their discussions. The paper is concluded in Section 5.

2. Mathematical Model

Total mass and momentum differential balances for a constant density and viscosity fluid in the laminar regime give the Navier–Stokes equations, which can be written as

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

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \cdot \rho \boldsymbol{u}\boldsymbol{u} = -\nabla \cdot \boldsymbol{\tau} - \nabla p + \rho \boldsymbol{g}, \tag{2}$$

$$\boldsymbol{\tau} = -\mu\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\dagger}\right), \tag{3}$$

with t time, $\boldsymbol{u} = (u, v, w)$ velocity vector, $\boldsymbol{\tau}$ viscous stress tensor, p pressure, ρ density, $\boldsymbol{g}$ external force per unit of mass, μ dynamic molecular viscosity, $(\nabla \boldsymbol{u})^{\dagger}$ transpose of the velocity gradient tensor $\nabla \boldsymbol{u}$.

In Cartesian rectangular 2D coordinates, the convective and molecular contributions of momentum take the form

$$\rho \boldsymbol{u}\boldsymbol{u} = \begin{Bmatrix} \rho uu & \rho uv \\ \rho vu & \rho vv \end{Bmatrix}, \quad \boldsymbol{\tau} = \begin{Bmatrix} -2\mu\dfrac{\partial u}{\partial x} & -\mu\left(\dfrac{\partial v}{\partial x} + \dfrac{\partial u}{\partial y}\right) \\ -\mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) & -2\mu\dfrac{\partial v}{\partial y} \end{Bmatrix}. \tag{4}$$

The velocity at the solid–fluid interfaces is given by the no-slip boundary condition

$$\boldsymbol{u} = \boldsymbol{u}_b, \tag{5}$$

where $\boldsymbol{u}_b$ is the wall velocity.

3. Numerical Method

3.1. Preliminaries

Figure 1 shows the fluid–solid 2D discretized space comprising the main and the staggered grids on which conservation equations are solved. Continuity and other scalar equations are solved on the main grid while momentum equations are discretized on a backward-staggered grid for each component following the finite volume usual practice [25]. In what follows, 2D treatment of u-cells are considered. According to their proximity to the solid body, cells can be classified as interior cells or cut cells, see u-cells (A) and (B) or (C) and (D), respectively. As the name suggests, a cut cell has its area (volume) reduced or cut when compared to a interior cell. For an interior u-cell, the discretized form of the momentum balance in 2D considering steady state and no sources other than the pressure term can be derived as

$$\begin{aligned} \int_s^n\int_w^e \frac{\partial}{\partial x}(\rho uu)dxdy + \int_s^n\int_w^e \frac{\partial}{\partial y}(\rho vu)dxdy &= -\int_s^n\int_w^e \frac{\partial p}{\partial x}dxdy + \int_s^n\int_w^e \frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)dxdy \\ &+ \int_s^n\int_w^e \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right)dxdy, \end{aligned} \tag{6}$$

$$\begin{aligned} F_e u_e - F_w u_w + F_n u_n - F_s u_s &= D_e u_E + D_w u_W + D_n u_N + D_s u_S + A_w^u p_w - A_e^u p_e \\ &- u_P(D_e + D_w + D_n + D_s), \end{aligned} \tag{7}$$

with u_P center velocity component, u_E, u_W, u_N, u_S velocity neighbors, u_e, u_w, u_n, u_s advected velocities at cell faces, p_e, p_w, p_n, p_s pressure at cell faces, A_e^u, A_w^u, A_n^u, A_s^u east, west, north and south faces of the backward-staggered u-cell.

The strength of convection F, and diffusion conductance D are given by Equations (8) and (9)

$$F_e = \rho u_e A_e^u, \quad F_w = \rho u_w A_w^u, \quad F_n = \rho v_n A_n^u, \quad F_s = \rho v_s A_s^u, \tag{8}$$

$$D_e = \mu \frac{A_e^u}{\Delta_P^E}, \quad D_w = \mu \frac{A_w^u}{\Delta_W^P}, \quad D_n = \mu \frac{A_n^u}{\Delta_P^N}, \quad D_s = \mu \frac{A_s^u}{\Delta_S^P}, \tag{9}$$

where u_e, u_w, v_n, v_s are the advecting velocities, $\Delta_P^E, \Delta_W^P, \Delta_P^N, \Delta_S^P$ are the distances to east, west, north and south neighbors.

Figure 2 presents a more general case involving a cut cell where the areas and distances are truncated. Special procedures are needed in this case to evaluate the convection and diffusion contributions. Refer to Kirkpatrick et al. [17] for more details on the methodology.

3.2. Advected Velocity Component Treatment for Interior Cells

This section develops three different methods that will be utilized to determine the advected velocity component on the faces: upwind, QUICK and WAHYD (TVD). The above methods consider all neighbors as interior cells but allow unequal distances with the center P.

The general form corresponds to an upwinded deferred-correction scheme, which for the east face results in

$$u_e = \chi_e u_P + (1 - \chi_e) u_E + S_e^*, \tag{10}$$

$$S_e^* = \chi_e S_e^{*+} + (1 - \chi_e) S_e^{*-}, \tag{11}$$

$$\chi_e = \begin{cases} 1, & F_e > 0, \\ 0, & F_e < 0. \end{cases} \tag{12}$$

Resulting source terms have the form $S_e = F_e S_e^*, S_w = F_w S_w^*$, etc. Note that for the upwind scheme one has $S_e = S_w = S_n = S_s = 0$.

As in [17], the QUICK scheme can be derived considering flow direction and that the second partial derivatives are constant $\partial^2 u / \partial x^2 = \text{const}$, $\partial^2 u / \partial y^2 = \text{const}$. When put in the form of Equation (10), resulting source terms for the east face are

$$S_e^{*+} = u_E \frac{\Delta_P^e \Delta_W^e}{\Delta_P^E \Delta_W^E} + u_P \left(\frac{\Delta_e^E \Delta_W^e}{\Delta_P^E \Delta_W^P} - 1 \right) - u_W \frac{\Delta_P^e \Delta_e^E}{\Delta_W^P \Delta_W^E}, \tag{13}$$

$$S_e^{*-} = u_P \frac{\Delta_e^E \Delta_e^{EE}}{\Delta_P^E \Delta_P^{EE}} + u_E \left(\frac{\Delta_P^e \Delta_e^{EE}}{\Delta_P^E \Delta_E^{EE}} - 1 \right) - u_{EE} \frac{\Delta_P^e \Delta_e^E}{\Delta_P^{EE} \Delta_E^{EE}}, \tag{14}$$

where as before, symbols Δ_1^2 indicate distances between 2 and 1.

For the WAHYD scheme of [26], the source terms are computed in the following manner

$$\psi(r, R) = \begin{cases} 0, & r < 0, \\ \dfrac{R}{2} \dfrac{r + |r|}{R - 1 + r}, & 0 \leq r \leq 1, \\ \min\left(\dfrac{r + Rr|r|}{R + r^2}, R \right), & r > 1, \end{cases} \tag{15}$$

$$S_e^{*+} = \frac{\psi(r_e^+, R_e^+)}{R_e^+}, \quad R_e^+ = \frac{\Delta_P^E}{\Delta_P^e}, \quad r_e^+ = \frac{u_P - u_W}{\Delta_W^P} \frac{\Delta_P^E}{u_E - u_P}, \tag{16}$$

$$S_e^{*-} = \frac{\psi(r_e^-, R_e^-)}{R_e^-}, \quad R_e^- = \frac{\Delta_P^E}{\Delta_e^E}, \quad r_e^- = \frac{u_{EE} - u_E}{\Delta_E^{EE}} \frac{\Delta_P^E}{u_E - u_P}. \tag{17}$$

3.3. Interpolation Details for the Convective Term

Following Kirkpatrick et al. [17], the velocity nodes on the main grid faces which are cut by the solid boundary are displaced to the center of the face segment that remains in the fluid, as shown in Figure 1. For an interior u-cell such as case (A), see Figure 1, there are no displacements for its velocity components. However, for interior u-cell (B) and cut u-cells (C) and (D), velocity components are displaced to the center of their cut faces and special interpolation procedures are needed. Consider the cut u-cell Case (D), which is set apart in Figure 3. Here, v_{ne} is displaced since the vertex lies within the solid body. In order to estimate the advecting velocity component v_n, a simple linear interpolation can be used

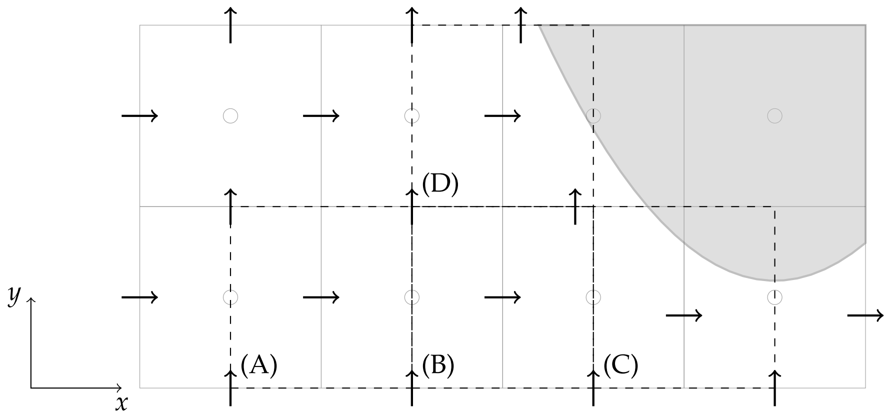


Figure 1. Main grid (solid gray lines) with immersed solid (gray area) showing displaced velocity components and four u-cell types (dashed black lines): (A) and (B) interior cells, (C) and (D) cut cells.

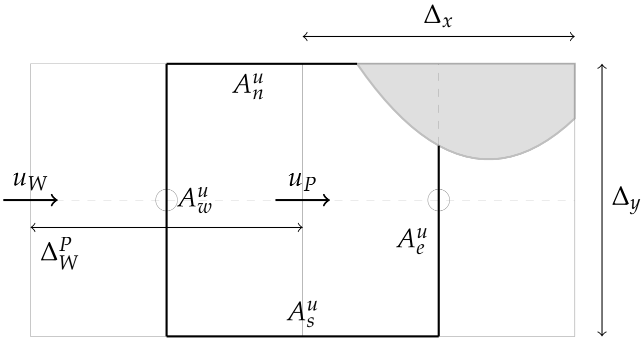


Figure 2. Cut u-cell showing areas and lengths used in the discretization.

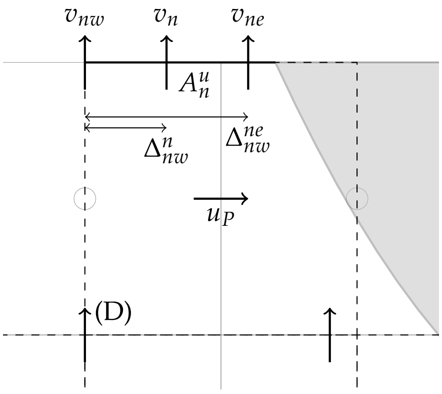


Figure 3. Case (D) cut u-cell interpolation for v_n at cut face A_n^u, v_{nw}, v_{ne} on fluid.

$$v_n = v_{nw}\left(1 - \frac{\Delta^n_{nw}}{\Delta^{ne}_{nw}}\right) + v_{ne}\frac{\Delta^n_{nw}}{\Delta^{ne}_{nw}}, \tag{18}$$

where Δ^n_{nw} and Δ^{ne}_{nw} are the distances among velocities in the n face shown in Figure 3. Note that this procedure can also be used in case of interior u-cells (A) and (B), and will subsequently be referred as A-scheme.

Depending on the interface position, v_{nw} or v_{ne} can end up being located inside the solid, see Figure 4. In this case, the A-scheme would over/under-estimate the velocity. Instead, an interpolation alternative that uses a point on the interface (also referred as an auxiliary point), can be considered. This interpolation scheme will be labeled B-scheme and it is presented in Equation (19)

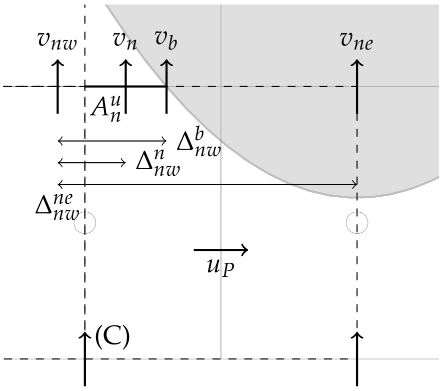


Figure 4. Case (C) cut u-cell interpolation for v_n at cut face A^u_n, v_{nw} on fluid, v_{ne} on solid and v_b on interface.

$$v_n = v_{nw}\left(1 - \frac{\Delta^n_{nw}}{\Delta^b_{nw}}\right) + v_b\frac{\Delta^n_{nw}}{\Delta^b_{nw}}, \tag{19}$$

with v_b the y velocity component of the boundary.

The vicinity of the interface may result in a situation as depicted in Figure 5 in which the (east) neighbor node is displaced due to the presence of the solid body. In this case, Kirkpatrick calculates the advecting velocity u_e using Equation (20). The correction is due to the fact that the linear estimate u'_e is not necessarily located at the center of the east face, see the dashed line joining u_P and u_E in Figure 5.

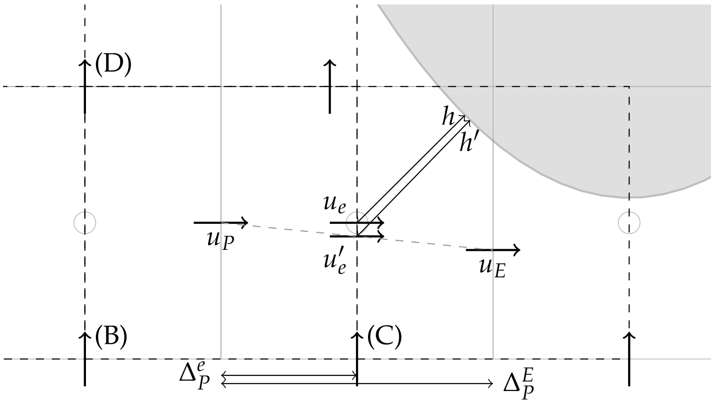


Figure 5. Interpolation for u_e near solid boundary. u'_e is the resulting interpolation between u_P and u_E, which is corrected to the face center with the distances to the solid boundary h and h'.

$$u_e = (u'_e - u_b)\frac{h}{h'} + u_b = \left(u_P\left(1 - \frac{\Delta_P^e}{\Delta_P^E}\right) + u_E\frac{\Delta_P^e}{\Delta_P^E} - u_b\right)\frac{h}{h'} + u_b, \tag{20}$$

where u_b is the x velocity component of the boundary, h and h' are the normal distances to the solid boundary of u_e and u'_e respectively. In case u_E ends up located inside the solid, an auxiliary point on the solid surface is used for the interpolation. If the east face is cut, the auxiliary point is located in the north or south face of the u_E cell, depending on the solid boundary orientation, as shown in Figure 6, and u_e is calculated using Equation (20). If the east face is uncut, the auxiliary point is located moving horizontally from u_E, as shown in Figure 7.

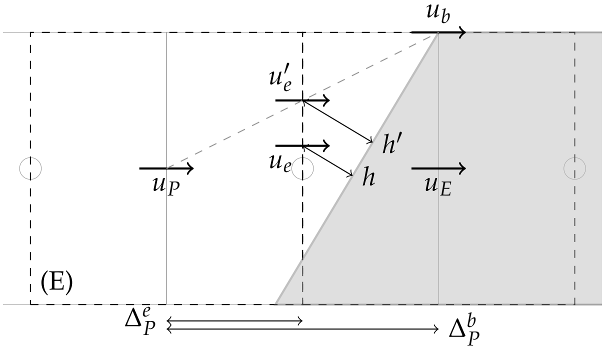


Figure 6. Limit case of interpolation for u_e near solid boundary in which A_e^u is cut, u_E is within the solid and $A_e^p = 0$.

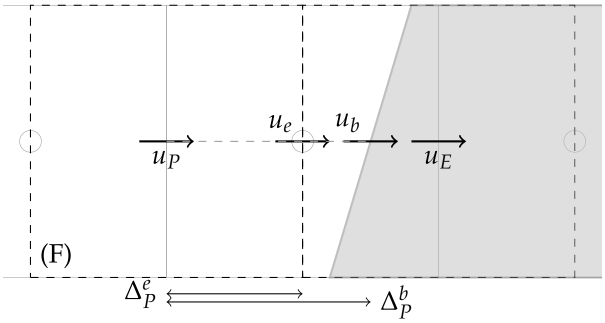


Figure 7. Interpolation for u_e near solid boundary in which A_e^u is uncut, u_E is within the solid and $A_e^p = 0$.

3.4. Interpolation Details for the Diffusive Term

Near a solid interface, as shown in Figure 8, the vector that goes from u_P to u_E is not perpendicular to the east face, Kirkpatrick suggest the diffusive term to be calculated using Equation (21). In case u_E is located inside the solid, auxiliary points are used for calculation, as explained previously for the determination of u_e

$$\left(\frac{\partial u}{\partial x}\right)_e = \frac{u_E - u_P}{\Delta_P^E} - \left(\left(1 - \frac{\Delta_P^e}{\Delta_P^E}\right)u_P + \frac{\Delta_P^e}{\Delta_P^E}u_E\right)\frac{S_y}{S_x}n_y, \tag{21}$$

with vector $\boldsymbol{S}$ defined from u_P to u_E, n_y is the y component of the unit normal vector $\boldsymbol{n} = \boldsymbol{N}/|\boldsymbol{N}|$, and $\boldsymbol{N}$ is the normal vector from surface that intersects $\boldsymbol{S}$ in the e face.

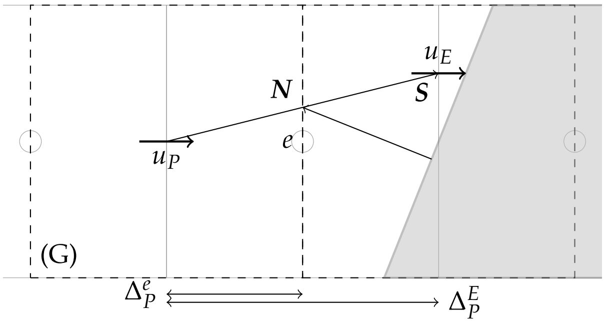


Figure 8. Computation for diffusive contribution requires vector $\boldsymbol{S}$ defined from u_P to u_E and normal vector from surface $\boldsymbol{N}$ which intersects with $\boldsymbol{S}$ in the e face.

3.5. Interpolation Details for the Source Term

Source terms that take into account the effect of pressure and shear due to the presence of the solid must be added to the momentum equation [27]. With reference to Figure 9 and for the cut-cell scheme, this gives Equation (22)

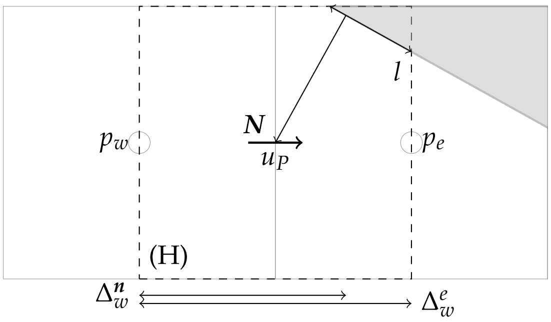


Figure 9. Computation for force requires normal vector from surface $\boldsymbol{N}$ which intersects with u_P and area l of solid body.

$$F_x = -\mu\left(\frac{u - u_b}{|\boldsymbol{N}|}\right)l + \left(p_w\left(1 - \frac{\Delta_w^n}{\Delta_w^e}\right) + p_e\frac{\Delta_w^n}{\Delta_w^e}\right)ln_x, \tag{22}$$

where l is the (wet) area of the solid body within the cell, n_x is the x component of the unit normal vector $\boldsymbol{n} = \boldsymbol{N}/|\boldsymbol{N}|$.

3.6. Classification of Small Cells

With reference to Figure 2 consider the following relations

$$A^u < \kappa\Delta_y, \tag{23}$$
$$A^v < \kappa\Delta_x, \tag{24}$$
$$(A_w^p < \kappa\Delta_y) \wedge (A_e^p < \kappa\Delta_y), \tag{25}$$
$$(A_s^p < \kappa\Delta_x) \wedge (A_n^p < \kappa\Delta_x), \tag{26}$$

where $0 < \kappa < 1$ is an adjustable parameter to obtain different levels of refinement in the (cut cell) geometrical representation. For a given grid and $\kappa \to 0$ we obtain the most refined geometric representation, while $\kappa \to 1$ will give a more staircase shape similar to the Voxel method.

If a determined main cell fulfills: (1) Equations (23) and (24) or; (2) Equations (23) and (26) or; (3) Equations (24) and (25) or; (4) Equations (25) and (26); it is classified as a small cell and removed from the computational domain.

3.7. Numerical Solution

The SIMPLE algorithm of Patankar [25] was used to treat the pressure-velocity coupling. Fully implicit discretized equations were solved using a parallel implementation of the ADI/TDMA algorithm [28] in an in-house C/OMP code. All simulations were conducted until the normalized residues were below 10^{-8} for the momentum and continuity equations. For comparisons between results, the definition of computational efficiency given in [29] was used

$$\eta = \frac{K}{et_c}, \tag{27}$$

where K is a constant arbitrarily fixed that depends on the machine used, e is some measurement of the numerical prediction error, t_c is the computation time.

4. Results

4.1. Flow between Concentric Cylindrical Walls

Figure 10 presents the schematic of a laminar flow driven by a momentum source between concentric cylindrical walls. Similar to Sato et al. [3], we compared the numerical results with the analytical solutions for the velocity field. Additionally, we give the pressure field analytic solution in the present work.

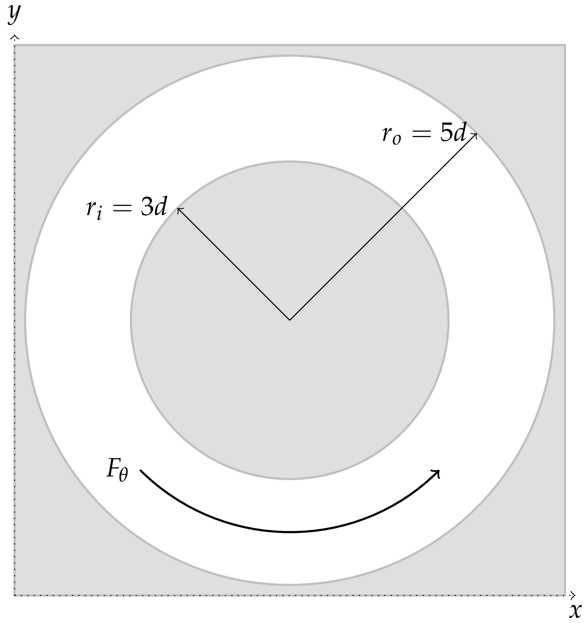


Figure 10. Geometry considered for the case of flow between concentric cylindrical walls.

Considering the r and θ components of Equation (2), and adding a volumetric force F_θ/r as a source term for the θ equation results in

$$-\frac{\rho v_\theta^2}{r} = -\frac{\partial p}{\partial r}, \tag{28}$$

$$0 = \mu \frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}(r v_\theta)\right) + \frac{F_\theta}{r}, \tag{29}$$

which upon integration using the no-slip requirement at the walls, provides the analytical solutions for the velocity and stress

$$v_\theta^+ = -\frac{\mathrm{Re}_\tau F_\theta^+ r^+}{2(r_o^2 - r_i^2)}\left(r_o^2 \ln{(r/r_o)} + r_i^2 \ln{(r_i/r)} - \frac{r_o^2 r_i^2}{r^2}\ln{(r_i/r_o)}\right), \tag{30}$$

$$\tau_{r\theta} = \frac{F_\theta}{2}\left(1 + \frac{2r_o^2 r_i^2 \ln{(r_i/r_o)}}{(r_o^2 - r_i^2)r^2}\right), \tag{31}$$

$$u_\tau = \sqrt{\frac{(u_{\tau i} r_i)^2 + (u_{\tau o} r_o)^2}{2r_c^2}}, \quad u_{\tau i} = \sqrt{\frac{\tau_{r\theta}|_{r_i}}{\rho}}, \quad u_{\tau o} = \sqrt{\frac{\tau_{r\theta}|_{r_o}}{\rho}},$$

$$r^+ = \frac{r}{r_c}, \quad r_c = \frac{r_i + r_o}{2}, \quad v_\theta^+ = \frac{v_\theta}{u_\tau}, \quad F_\theta^+ = \frac{F_\theta}{u_\tau^2 \rho}, \quad \mathrm{Re}_\tau = \frac{\rho u_\tau r_c}{\mu},$$

Solving for the pressure requires the determination of a constant at a position where the pressure is known. Choosing $p = 0$ for $r = r_c$, and defining $\Delta p^+ = \Delta p/(\rho u_\tau^2)$, the analytical dimensionless pressure profile is

$$\begin{aligned}\Delta p^+ = \left(\frac{\alpha}{u_\tau}\right)^2 \Bigg[&\frac{\beta^2}{2}\left((r\ln r)^2 - (r_c \ln r_c)^2 - r^2 \ln r + r_c^2 \ln r_c + \frac{r^2 - r_c^2}{2}\right) + \\ &\beta\gamma\left(r^2 \ln r - r_c^2 \ln r_c + \frac{r_c^2 - r^2}{2}\right) + \beta\delta\left((\ln r)^2 - (\ln r_c)^2\right) + \\ &2\gamma\delta \ln\left(\frac{r}{r_c}\right) + \gamma^2\left(\frac{r^2 - r_c^2}{2}\right) + \frac{\delta^2}{2}\left(\frac{1}{r_c^2} - \frac{1}{r^2}\right)\Bigg], \\ \alpha = \frac{F_\theta}{2\mu\beta}, \quad &\beta = \left(r_o^2 - r_i^2\right), \quad \gamma = r_i^2 \ln r_i - r_o^2 \ln r_o, \quad \delta = (r_i r_o)^2 \ln\left(\frac{r_o}{r_i}\right),\end{aligned} \tag{32}$$

Consider the L_2 error norm for the velocity as

$$e_{L2}^{\boldsymbol{u}} = \frac{1}{N}\sum_i^N \left(v_{\theta i}^{+n} - v_{\theta i}^+\right)^2, \tag{33}$$

where $v_{\theta i}^{+n}$ is the numerical value of the theta component dimensionless velocity at i-cell obtained with the $n = A \vee B$ interpolation scheme, N is the number of cells within the region of interest $r_i \leq r \leq r_o$.

Numerical results were obtained using the three methods described in Section 3.2, $\kappa = 0.02, 0.20, 0.40$ and grids 22×22, 42×42, 62×62, 82×82, 102×102, 122×122 for physical parameters equivalent to a $Re = 2.0$. With these kappa values chosen, and as stated in Section 3.6, small $u \vee v$-cells will be designated as those which areas of their cut flow faces $(A_e^{\boldsymbol{u}}, A_w^{\boldsymbol{u}}, A_n^{\boldsymbol{u}}, A_s^{\boldsymbol{u}})$ and/or the corresponding areas of the main cells cut flow faces $(A_e^p, A_w^p, A_n^p, A_s^p)$ are under 2, 20 and 40% of their respective uncut areas (Δ_x, Δ_y).

Table 1 presents the number and percentage of slave cells generated as function of κ and the grid size. These numbers correspond to either the u or v cut cells since, given the geometry studied, their numbers turned out the same. As expected, the proportion of slave cells increases with the value of κ, being more cut cells classified as slave cells with this rise in tolerance. Moreover, the percentage of slave cells increases when the grid is coarser due to the resulting lower geometry resolution for the immersed solid body representation. Figure 11 presents the distribution of fluid, slave and solid cells near the solid boundaries. Notice that for $\kappa = 0.40$ the distribution of slave cells mimics an enlargement of the solid body's dimensions. The amount of slave cells generated with $\kappa = 0.20$ reduces the apparent dimension of the solid body but produces less smoothness on the body surface. Lastly, with $\kappa = 0.02$ the solid body is represented more accurately with a smoother surface since less slave cells are produced.

Table 1. Number of u or v slave cells generated and their percentage (in parenthesis) over total u or v cut cells for the flow between concentric cylindrical walls case as function of κ and the grid size.

κ	20 × 20	42 × 42	62 × 62	82 × 82	102 × 102	122 × 122
0.02	12 (20.0)	8 (6.8)	32 (16.7)	32 (13.3)	52 (17.1)	36 (9.8)
0.20	18 (30.0)	32 (27.1)	40 (20.8)	64 (26.7)	60 (19.7)	84 (22.8)
0.40	26 (43.3)	54 (45.8)	68 (35.4)	104 (43.3)	92 (30.3)	120 (32.5)

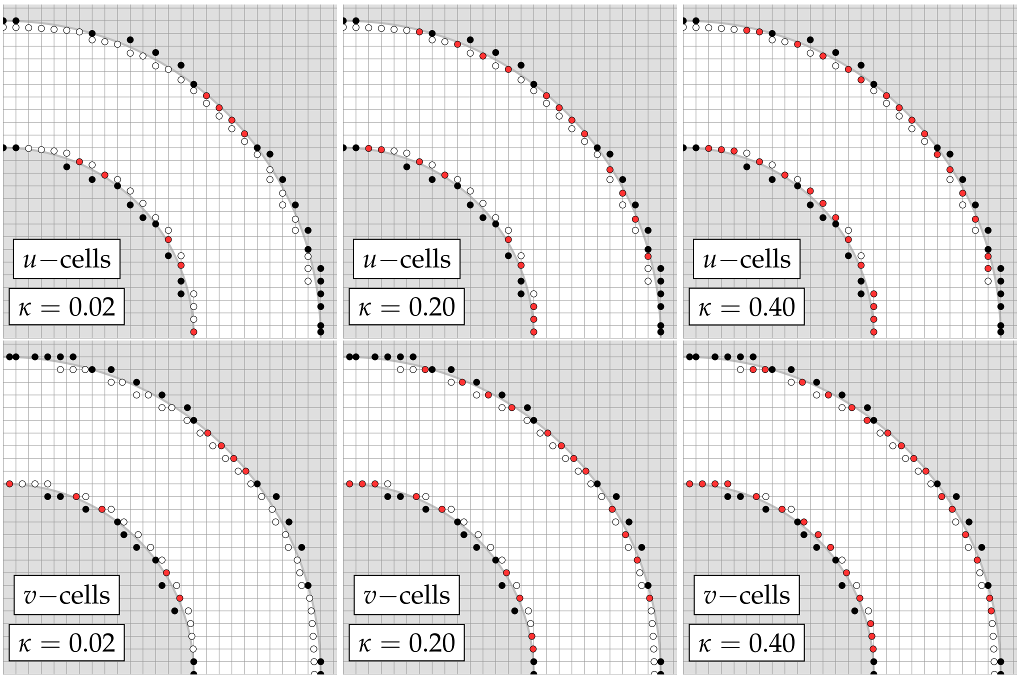


Figure 11. Fluid (white), solid (black) and slave (red) cells distribution around the immersed solids for the flow between cylindrical walls case with the 122 × 122 grid.

Figures 12–14 present the L_2 errors plotted against the grid spacing where it can be seen that, for the upwind method, a less than first order tendency was obtained, whereas the QUICK and WAHYD schemes, supra linear behavior was achieved. These degradations in the order of accuracy for cut-cell methodologies had been observed in the literature [17], and explained as the possible result of irregularities of the mesh near the solid boundaries affecting the local error. In the three methods studied, a big difference in magnitude in the errors was observed when using $\kappa = 0.40$ compared to $\kappa = 0.02, 0.20$, where no major differences were detected.

Figures 15 and 16 illustrate the dependency observed for the computational efficiency with the κ values. In the present case, the solid bodies are represented by circumferences and the results indicate that, regardless of the grid used, the lower values of κ favor the computational efficiency. The QUICK and WAHYD schemes gave higher efficiency values than the upwind scheme. Although small, generally the B interpolation scheme produced a more efficient algorithm than the A alternative for each method at the smallest values of κ.

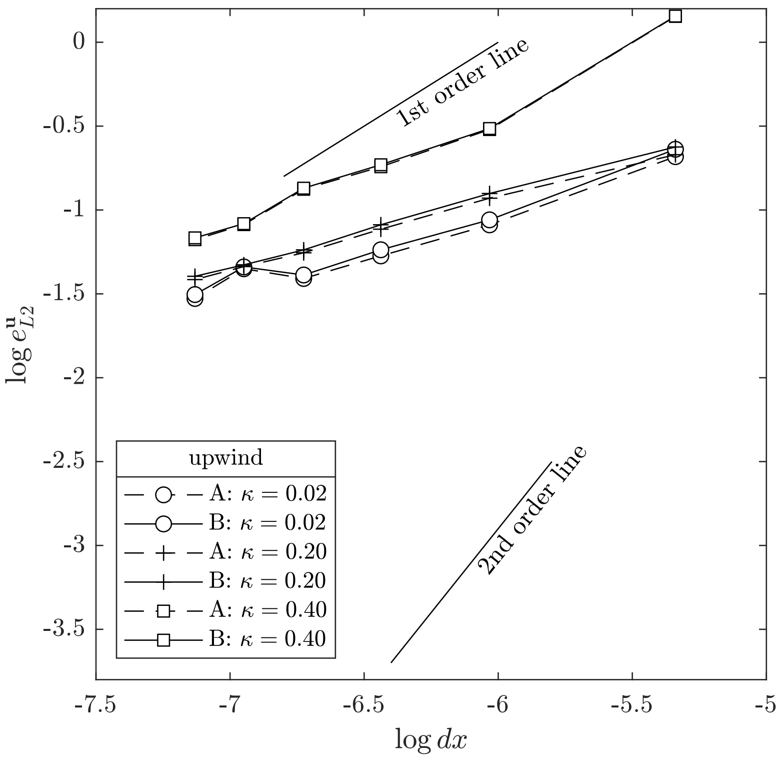


Figure 12. Velocity L_2 error norm as function of the interpolation schemes and κ parameter for the flow between concentric cylinders case using the upwind method.

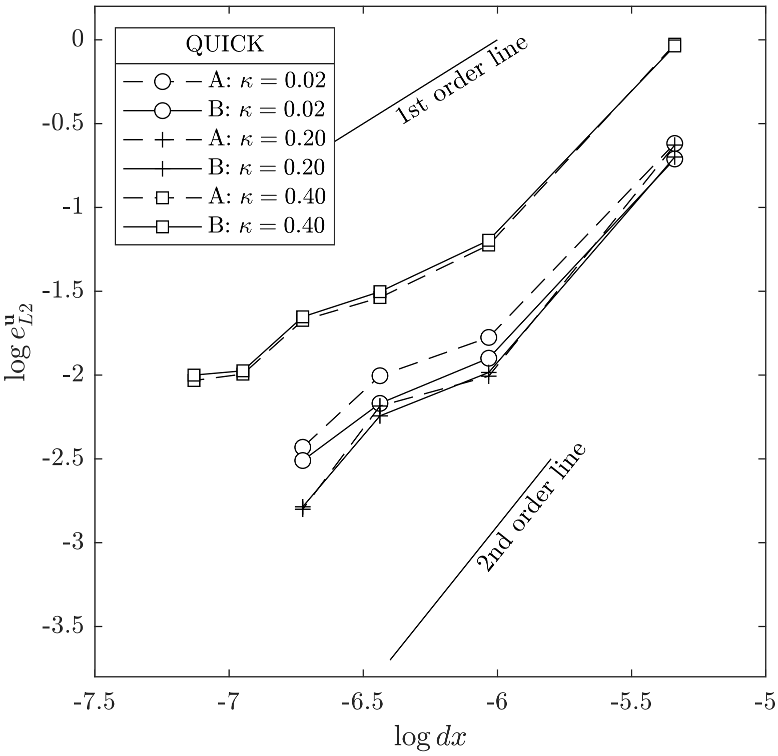


Figure 13. Velocity L_2 error norm as function of the interpolation schemes and κ parameter for the flow between concentric cylinders case using the QUICK method.

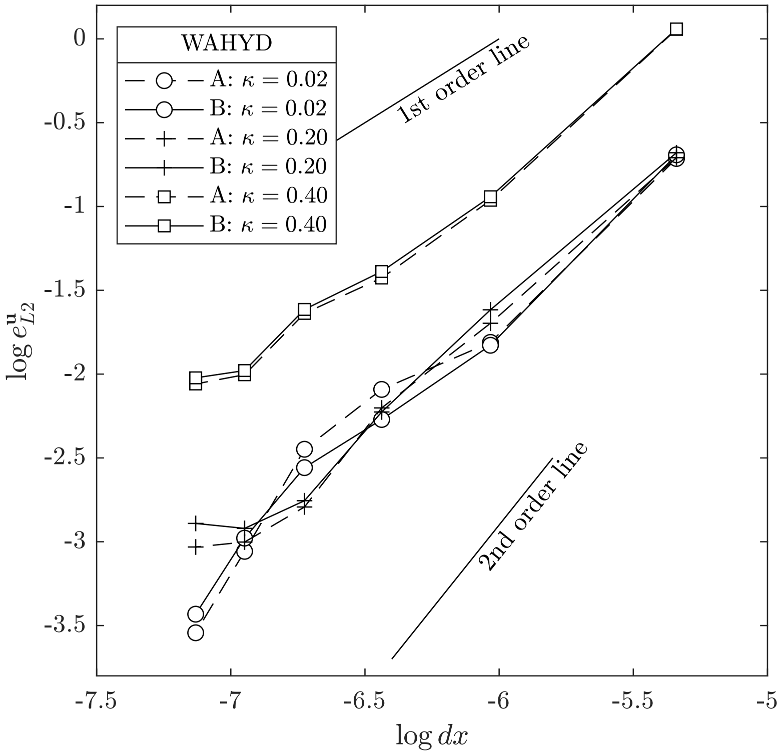
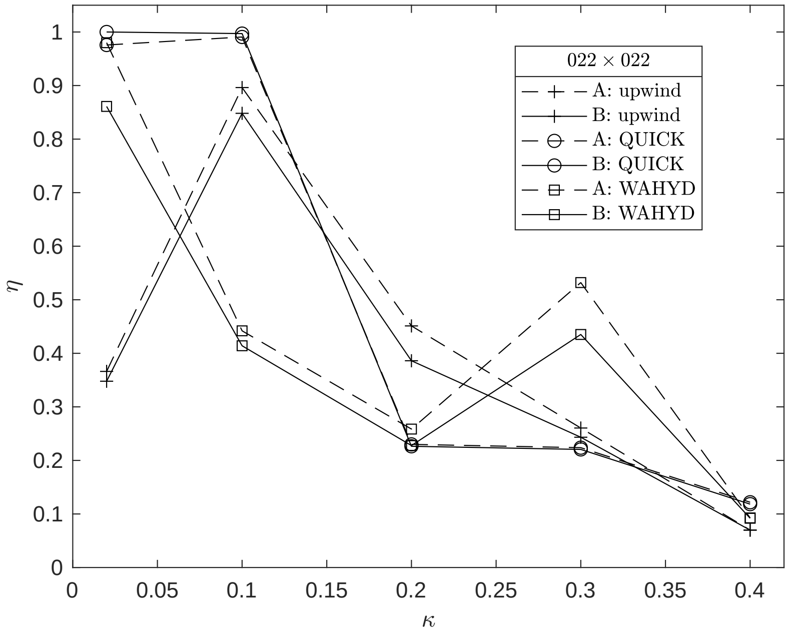
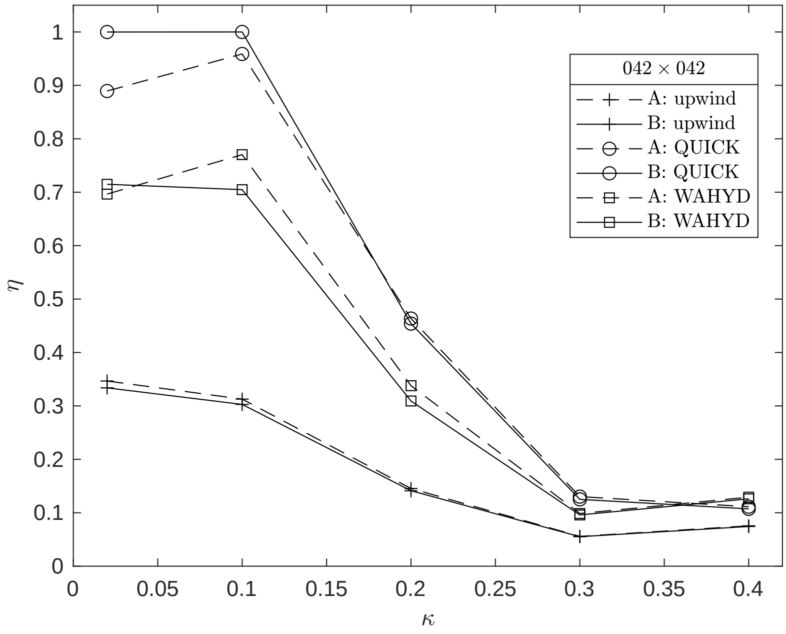


Figure 14. Velocity L_2 error norm as function of the interpolation schemes and κ parameter for the flow between concentric cylinders case using the WAHYD method.



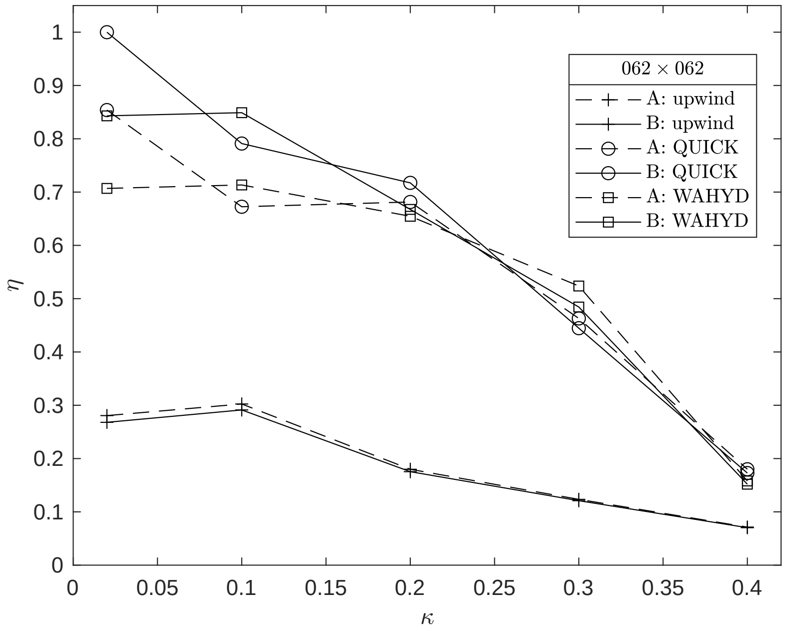
(**a**)

Figure 15. *Cont*.



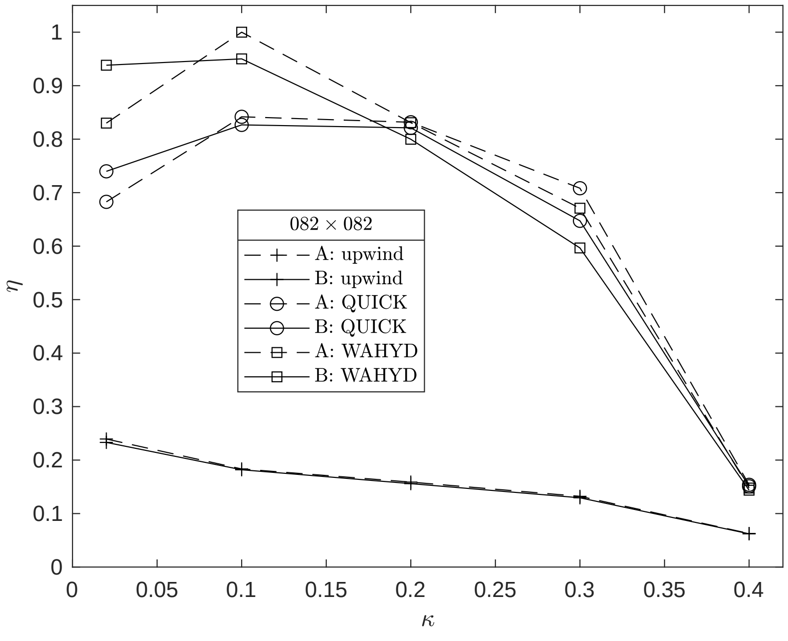
(**b**)

Figure 15. Computational efficiency as function of κ for the flow between concentric cylinders case and grids: (**a**) 22 × 22, (**b**) 42 × 42.



(**a**)

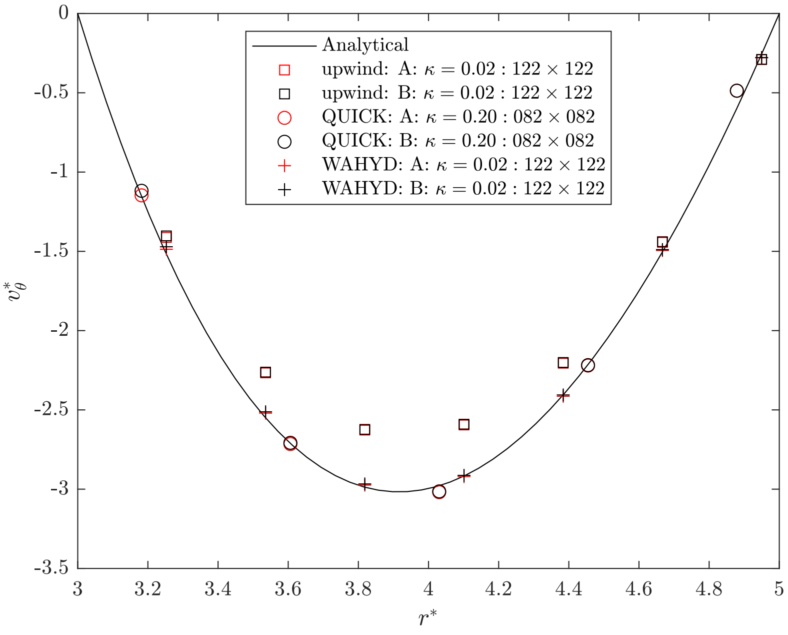
Figure 16. *Cont*.



(**b**)

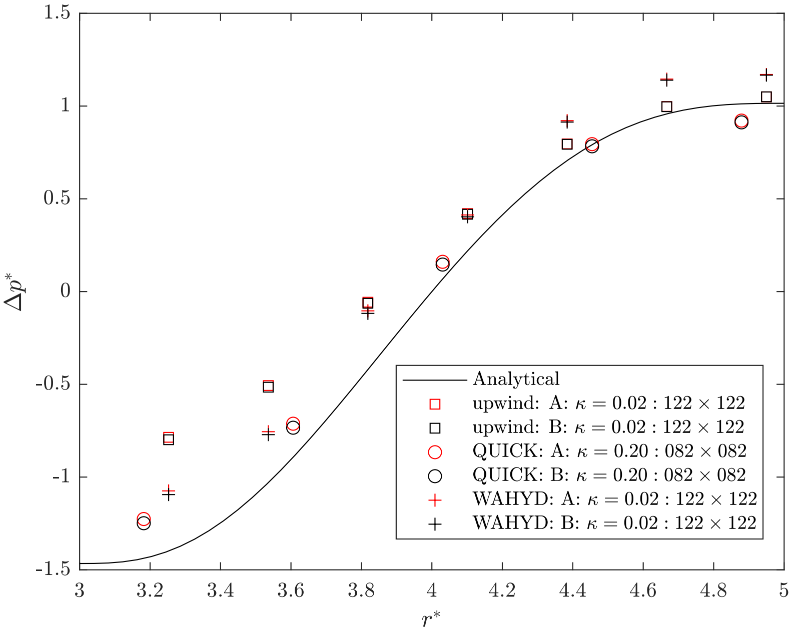
Figure 16. Computational efficiency as function of κ for the flow between concentric cylinders case and grids: (**a**) 62 × 62, (**b**) 82 × 82.

Figure 17 presents the dimensionless velocity and pressure profiles for the cases with the lowest numerical error obtained. A good agreement of the higher-order methods in the dimensionless velocity profile can be observed, whereas the upwind scheme underpredicts the velocity peak due to its diffusive nature. Following the same argument, the sigmoidal-like dimensionless pressure profile is better represented by the QUICK and WAHYD numerical solutions than the upwind scheme, at least in a shape-preserving manner for the relatively coarse grids used.



(**a**)

Figure 17. *Cont.*



(**b**)

Figure 17. Analytical and best numerical solutions found for the flow between concentric cylindrical walls case, $(\theta = 1.5\pi)$. (**a**) dimensionless velocity; (**b**) dimensionless pressure.

4.2. Flow on a Inclined Channel

Figure 18 depicts the flow on an inclined channel situation. For the $Ox'y'$ coordinate system, making a y' momentum balance gives

$$0 = -\frac{\partial p}{\partial y'} + \mu \frac{\partial^2 v}{\partial x'^2}. \tag{34}$$

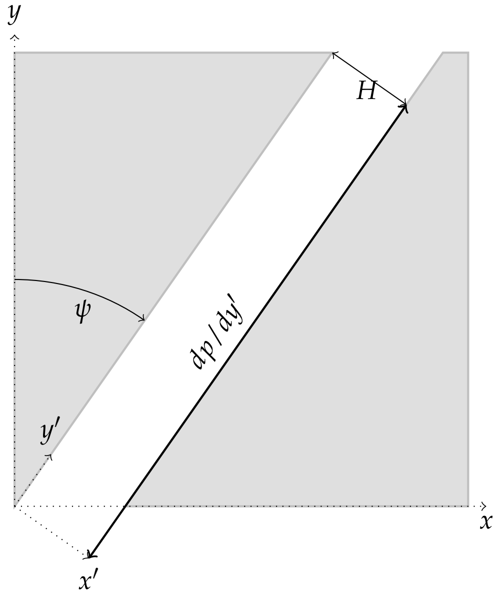


Figure 18. Geometry considered for the case of flow on a inclined channel.

Considering $\partial p/\partial y' = (p_L - p_0)/L$, one has as analytical solution for the y' velocity component

$$v^* = 4x^*(1 - x^*), \tag{35}$$

$$x^* = \frac{x'}{H}, \quad v^* = \frac{v}{v_{\max}}, \quad v_{\max} = \left(\frac{p_0 - p_L}{L}\right)\left(\frac{H^2}{8\mu}\right), \quad \bar{v} = \frac{2}{3}v_{\max}, \quad \mathrm{Re}_H = \frac{\rho\bar{v}H}{2\mu}.$$

A velocity L_2 error difference between the A and B interpolation schemes can be defined as

$$\Delta e^{\boldsymbol{u}}_{L2} = e^{*B}_{L2} - e^{*A}_{L2}, \quad e^{*n}_{L2} = \frac{1}{N}\sum_{i}^{N}(v^{*n}_i - v^*_i)^2, \quad n = A, B. \tag{36}$$

Simulations were performed for $\kappa \in [0.05, 0.40]$, angles $\psi = 10, 20$, grid sizes 22×22 and 62×62. Parameters were adjusted to give $\mathrm{Re}_H = 0.1$.

Tables 2 and 3 display cut-cell details obtained as function of κ and Figures 19 and 20 present their distribution for the 22×22 grid and $\psi = 10, 20$ and three values of κ. Similar tendencies to those of Table 1 are observed, but in the present situation there is some asymmetry in the origination of cut cells and slave cells which increases with ψ, since no symmetry plane can be found parallel to x or y coordinates. Figure 21 presents the errors $\Delta e^{\boldsymbol{u}}_{L2}$ for all cases considered. Similar error profiles are found for both grids, with the coarser one having its magnitudes larger for the angles studied. For values of $\kappa \leq 0.25$, and for both grids, the B interpolation scheme performed better than A (i.e., $\Delta e^*_{L2} < 0$). For other values of the geometrical refinement parameter κ, positive values of the error difference can be seen. Nevertheless, the magnitudes of the negative errors are bigger than the positive ones, and these bigger negative values are found in the region of more geometrical refinement. These trends are more pronounced in the cases with greater slope. Figure 22 presents the dimensionless velocity profiles, where it can be seen that the three methods employed, i.e., upwind, QUICK and WAHYD, produced similar solutions.

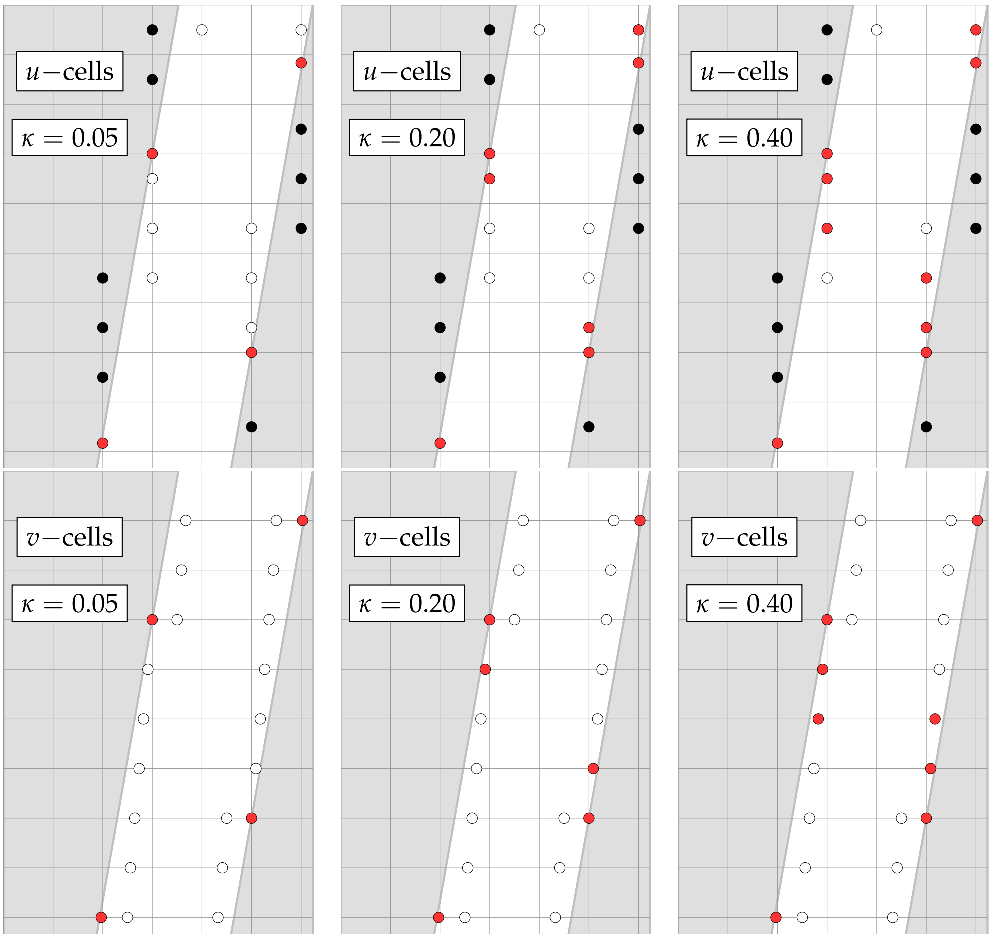


Figure 19. Fluid (white), solid (black) and slave (red) cells distribution around the immersed solids for the inclined channel flow case with the 22×22 grid and $\psi = 10°$.

Table 2. Number of u slave cells generated and their percentage (in parenthesis) over total u cut cells for the inclined channel flow case as function of κ, ψ and grid size.

	22 × 22		**62 × 62**	
κ	$\psi = 10°$	$\psi = 20°$	$\psi = 10°$	$\psi = 20°$
0.05	7 (15.9)	10 (19.2)	19 (13.7)	20 (12.4)
0.10	9 (20.5)	12 (23.1)	26 (18.7)	32 (19.9)
0.20	12 (27.3)	15 (28.9)	32 (23.0)	43 (26.7)
0.30	16 (36.4)	20 (38.5)	47 (33.8)	53 (32.9)
0.40	20 (45.5)	24 (46.2)	61 (43.9)	64 (39.8)

Table 3. Number of v slave cells generated and their percentage (in parenthesis) over total v cut cells for the inclined channel flow case as function of κ, ψ and grid size.

	22 × 22		**62 × 62**	
κ	$\psi = 10°$	$\psi = 20°$	$\psi = 10°$	$\psi = 20°$
0.05	7 (15.6)	9 (17.7)	19 (13.8)	20 (12.4)
0.10	9 (20.0)	11 (21.6)	26 (18.8)	32 (19.9)
0.20	11 (24.4)	14 (27.5)	32 (23.2)	42 (26.1)
0.30	15 (33.3)	19 (37.3)	47 (34.1)	52 (32.3)
0.40	19 (42.2)	23 (45.1)	61 (44.2)	63 (39.1)

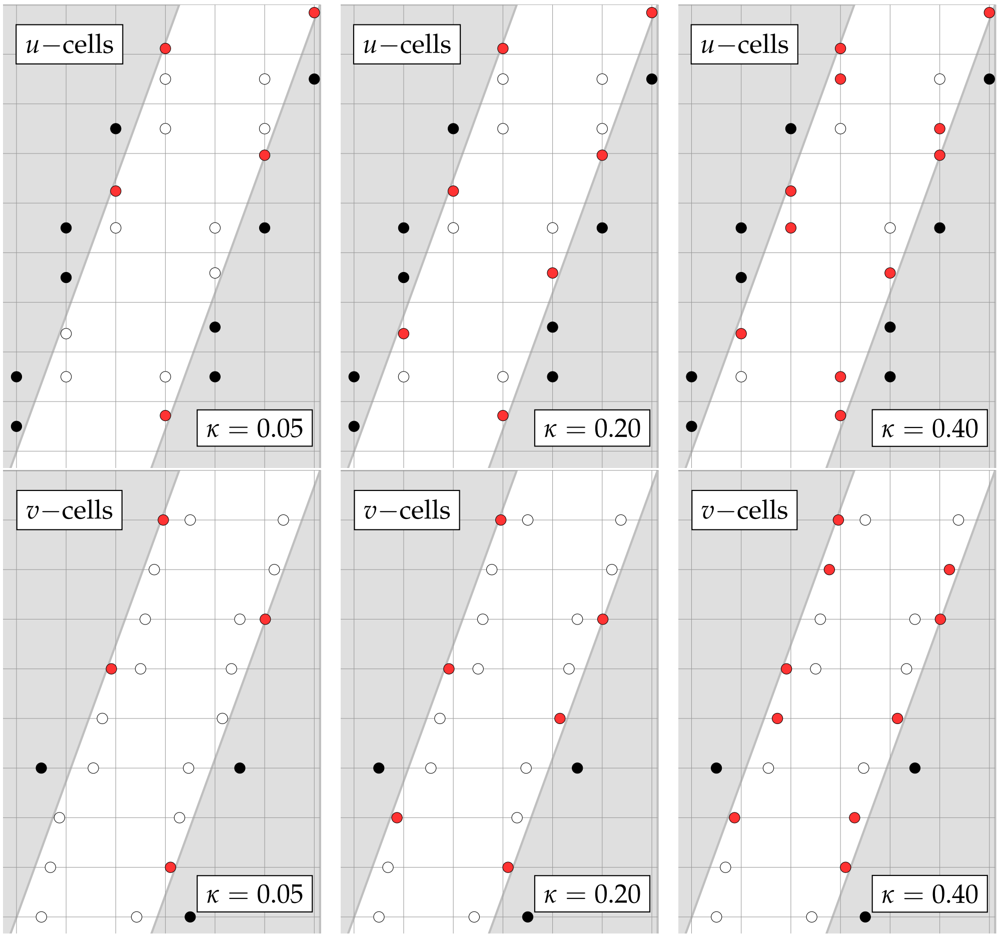
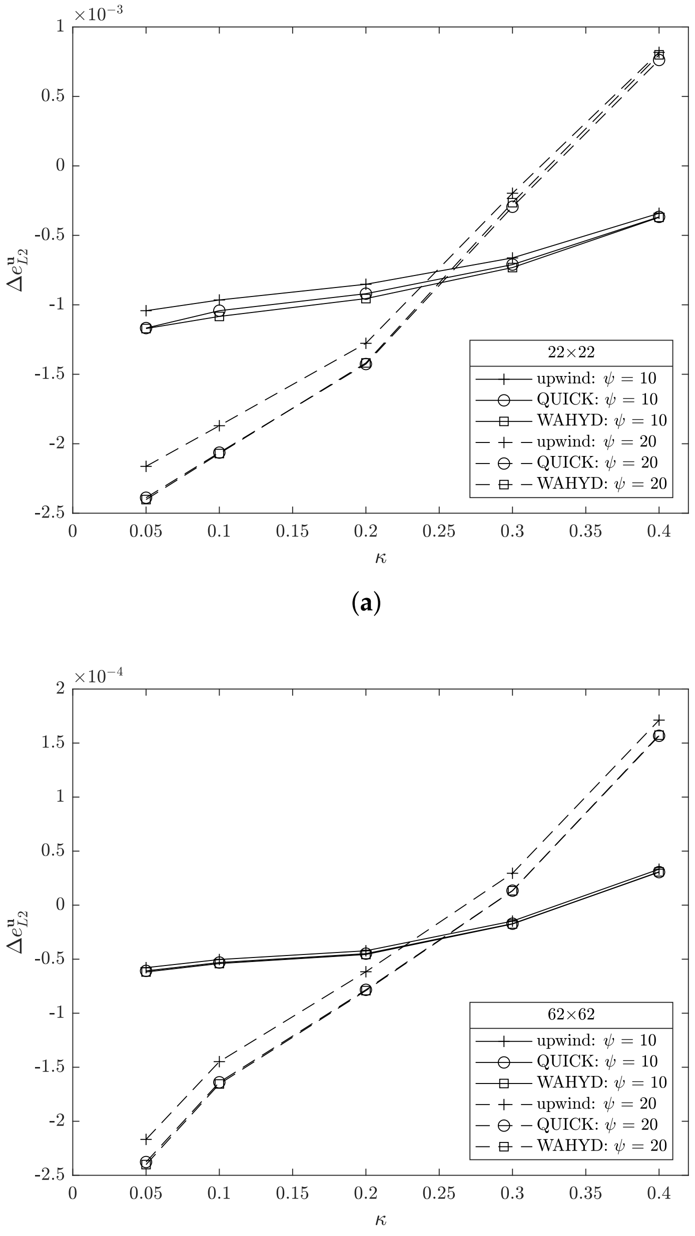


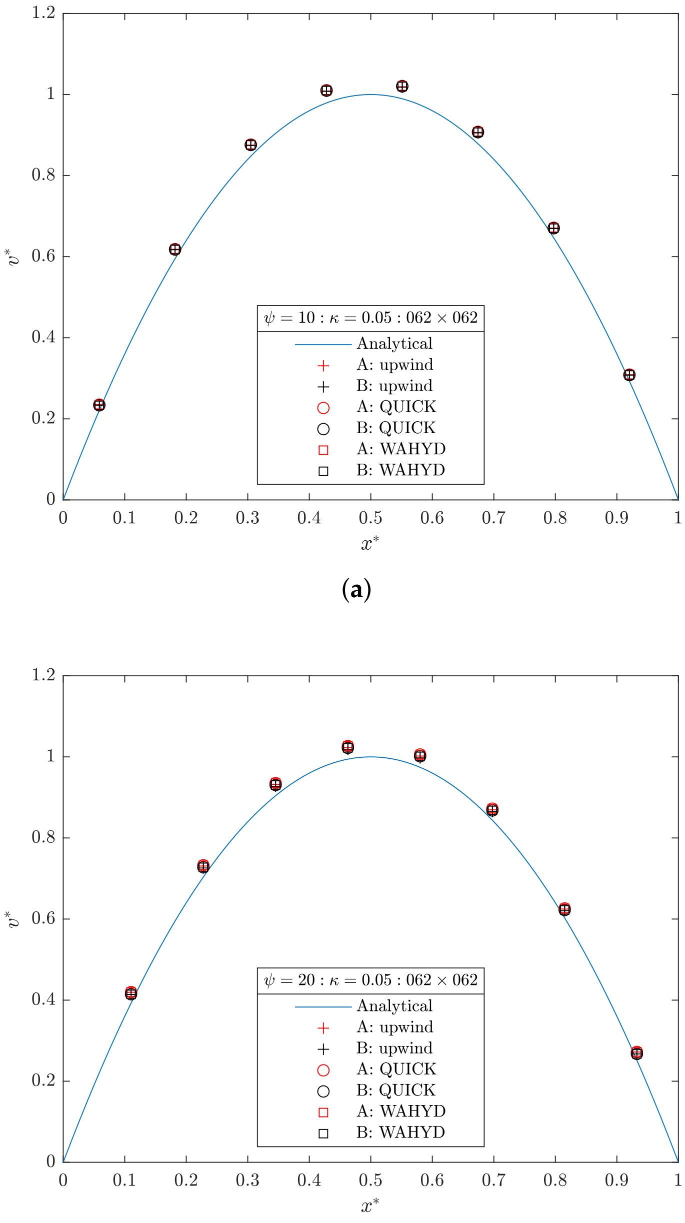
Figure 20. Fluid (white), solid (black) and slave (red) cells distribution around the immersed solids for the inclined channel flow case with the 22 × 22 grid and $\psi = 20°$.



(**a**)

(**b**)

Figure 21. Velocity L_2 error norm difference as function of κ parameter and ψ, for the flow on inclined channel case. (**a**) 22 × 22 grid; (**b**) 62 × 62.

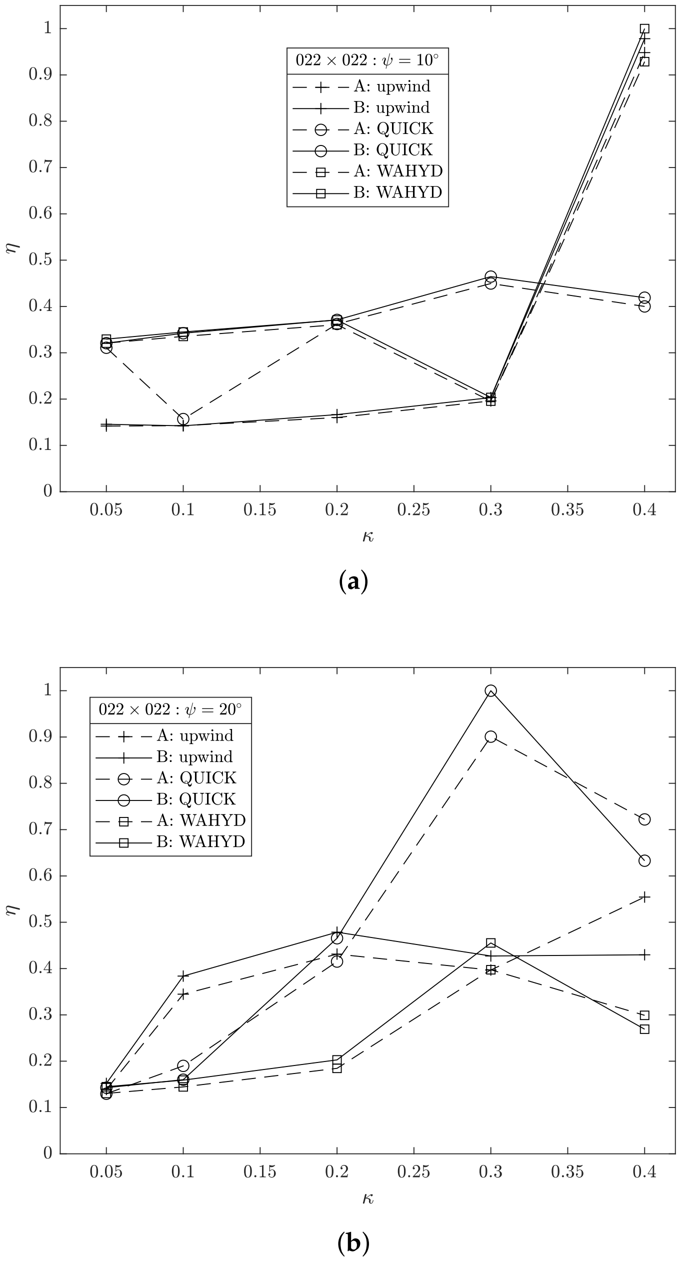


(**a**)

(**b**)

Figure 22. Dimensionless velocity profile for the flow on inclined channel case. (**a**) $\psi = 10$; (**b**) $\psi = 20$.

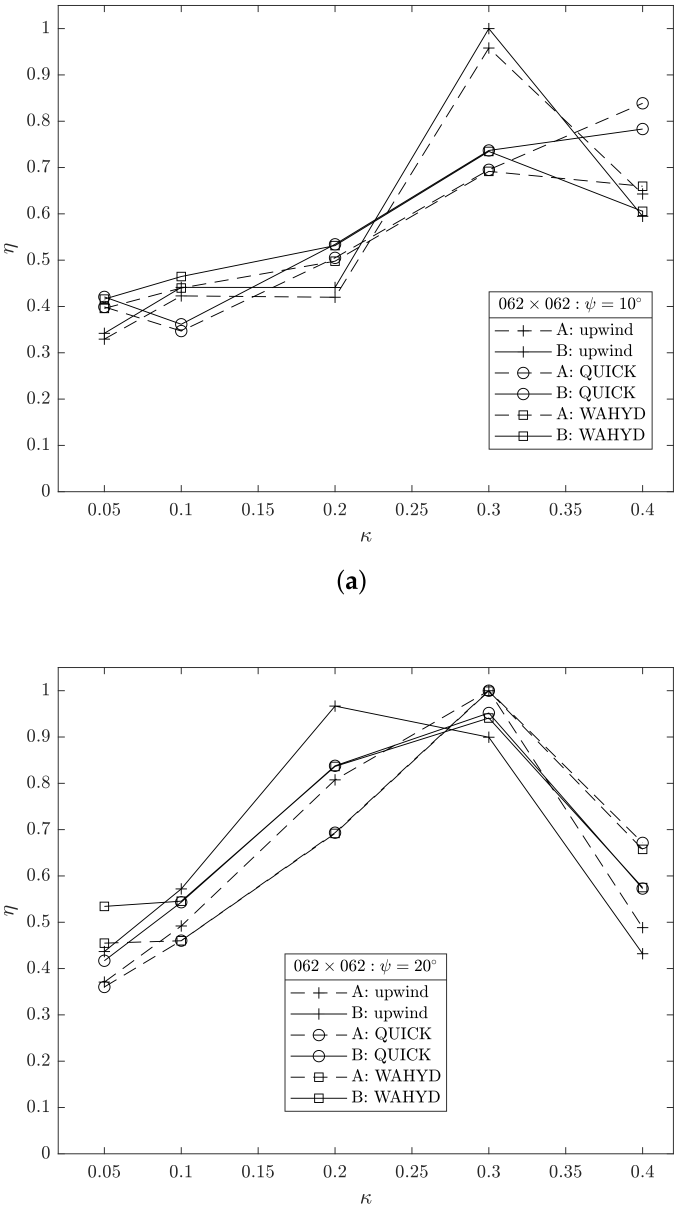
Figures 23 and 24 present the computational efficiency obtained for the cases studied as a function of κ. Different from the tendency shown in the flow between concentric cylinders case, the higher efficiency values are obtained with the mid-to-high values of κ. This difference may be due to the type of curve that represents the immersed solid body, which in the present case are two straight lines, whereas in the former case they were circumferences. Nevertheless, the maximum-to-minimum computational efficiency ratio is higher for the flow between concentric cylinders case than in the present one, suggesting a greater effect of κ on the computational efficiency when the streamlines are curved.



(**a**)

(**b**)

Figure 23. Computational efficiency as function of κ for the flow on a skewed-channel case using grid 22 × 22 and ψ: (**a**) 10º, (**b**) 20º.



(**b**)

Figure 24. Computational efficiency as function of κ for the flow on a skewed-channel case using grid 62 × 62 and ψ: (**a**) 10º, (**b**) 20º.

4.3. Flow around a Circular Cylinder

Two-dimensional flow around a single cylinder was simulated for Reynolds numbers 7, 10 and 20. Domain size was 40D in the transverse direction and 60D in the streamwise direction. The cylinder center was located at the coordinates (20D, 20D), see Figure 25. Boundary conditions of the complete domain were established as uniform velocity profile at the left boundary, outlet at right boundary. Bottom and top boundaries were set as symmetry. No-slip at the cylinder surface. Results of the contributions to the drag coefficient and the pressure coefficient are considered as in [30]

$$C_f = \frac{1}{0.5\rho u_\infty^2 D}\int_0^{2\pi} \boldsymbol{\tau}\cdot\boldsymbol{n} r d\theta, \quad C_p = \frac{-1}{0.5\rho u_\infty^2 D}\int_0^{2\pi} p\cos\theta r d\theta, \tag{37}$$

and the dimensionless pressure

$$\hat{p}(\theta) = \frac{p(\theta) - p_\infty}{0.5\rho u_\infty^2}, \tag{38}$$

with $\boldsymbol{n}$ normal vector at cylinder surface, $\boldsymbol{\tau}$ viscous stress tensor, D cylinder diameter, r cylinder radius, u_∞, p_∞ upstream velocity and pressure.

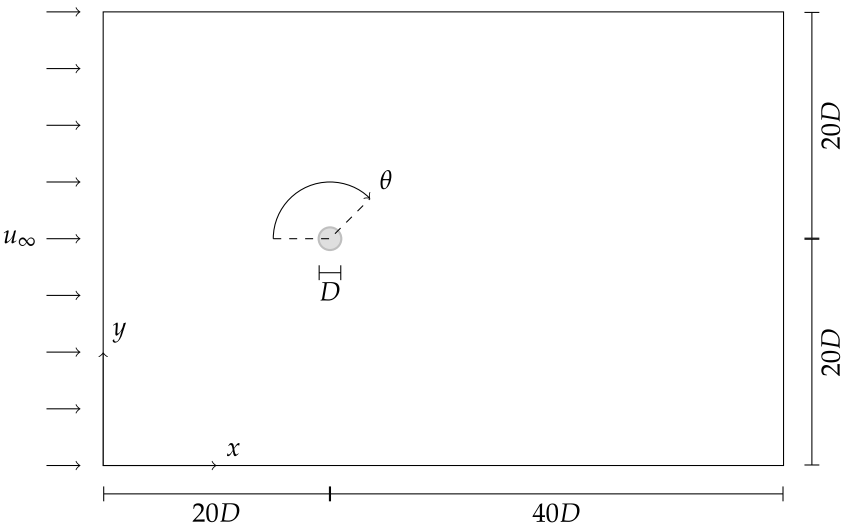


Figure 25. Geometry considered for the case of flow around a cylinder.

Three values of κ and five nonuniform structured grids were used with a finer section of size $5D \times 5D$ centered around the cylinder. Table 4 presents the cases considered and the percentage of slave cells generated. Figure 26 presents part of their distribution on the vicinity of the solid body. To evaluate surface derived quantities, interpolation for missing surface points was performed using piecewise cubic Hermite interpolating polynomials. Considering the treatment of slave cells, the higher value of κ value (0.40) gives the impression of a cylinder with an effective diameter greater than the original body. The intermediate value of κ (0.20) produces a less smooth surface but with better approximation of the true solid dimension. Finally, the lower value of κ better approximates the solid body shape both on smoothness and its real dimension.

Table 4. Grid sizes used for the flow around a cylinder case.

Grid	Cells Along Cylinder Diameter	Total Cells	Slave Cells		
			$\kappa = 0.05$	$\kappa = 0.20$	$\kappa = 0.40$
A	8	44,772	10 (33.3)	13 (43.3)	21 (70.0)
B	12	100,096	6 (12.5)	21 (43.8)	25 (52.1)
C	15	154,923	10 (16.7)	18 (30.0)	20 (33.3)
D	20	276,624	14 (17.5)	25 (31.1)	37 (46.3)
E	30	395,064	18 (15.0)	37 (30.8)	53 (44.2)

Numerical computation of magnitudes $C_f, C_p, \hat{p}(0), \hat{p}(\theta)$ was performed for each case of Table 4 and their deviation with respect to reference values taken from [30] was computed as

$$e^\phi = \left| \frac{\phi - \phi^r}{\phi^r} \right|, \quad \phi = C_f, C_p, \hat{p}(0), \hat{p}(\theta), \tag{39}$$

$$e_T = \frac{e^{C_f} + e^{C_p} + e^{\hat{p}(0)} + e^{\hat{p}(\pi)}}{4}, \tag{40}$$

where ϕ^r denotes the references, e^ϕ absolute error of numerical estimation of ϕ, e_T combined error of numerical predictions.

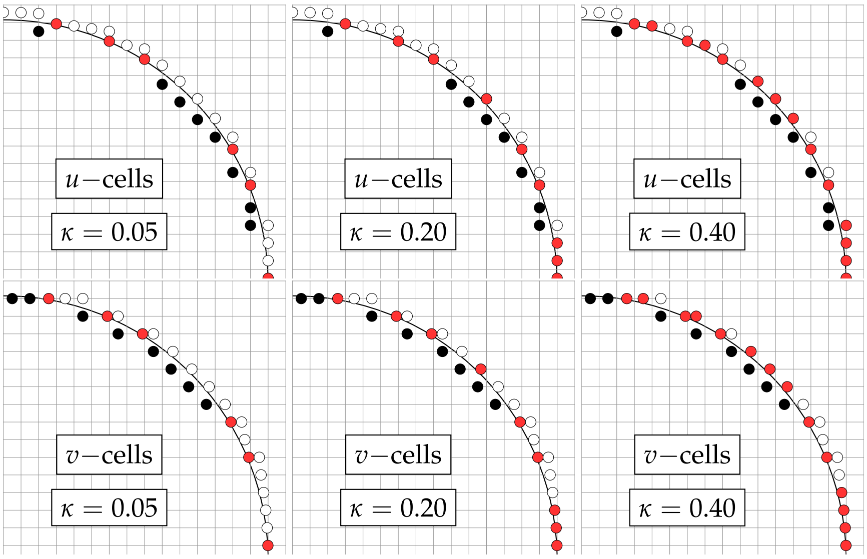


Figure 26. Fluid (white), solid (black) and slave (red) cells distribution around the immersed solid for the flow around a cylinder case with the E grid.

Figures 27–30 present the absolute error of $C_f, C_p, \hat{p}(0), \hat{p}(\theta)$ as function of the grid size and the extremes values of κ used for Re = 20 and both interpolation approaches. It can be seen that smaller errors are obtained with $\kappa = 0.05$ and the B interpolation scheme, although influence of the interpolation is more modest on the profiles. Results for g20 and g30 grids are almost the same, implying that convergence of the numerical procedure is achieved with these grid sizes. Similar profiles were obtained for Re = 7, 10 but their figures are not presented to avoid extending the manuscript. Figure 31 presents the combined errors of the four previously studied quantities as function of the Reynolds number for each interpolation scheme and method used. As expected, on account of its greater numerical diffusion, the upwind method presented larger combined error compared to the higher-order alternatives. Regardless of the method and the Reynolds number, the difference observed in the combined error is about 1% in favor of the B interpolation scheme.

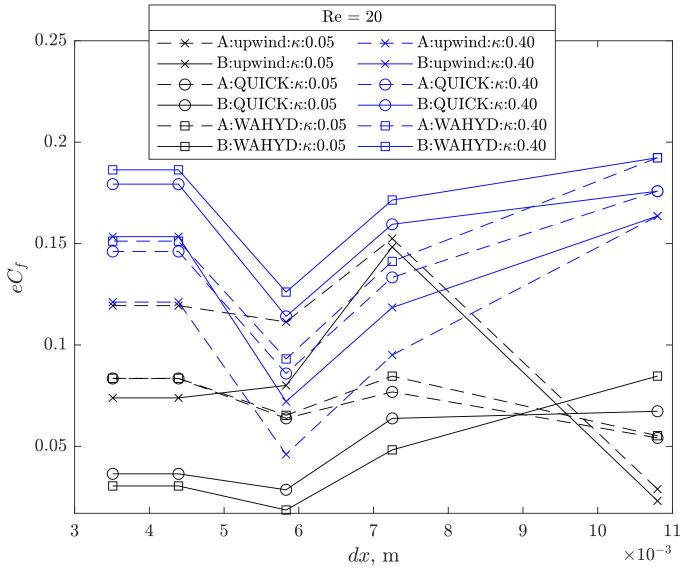


Figure 27. Drag coefficient error as function of grid size for the flow around a cylinder case with Re = 20.

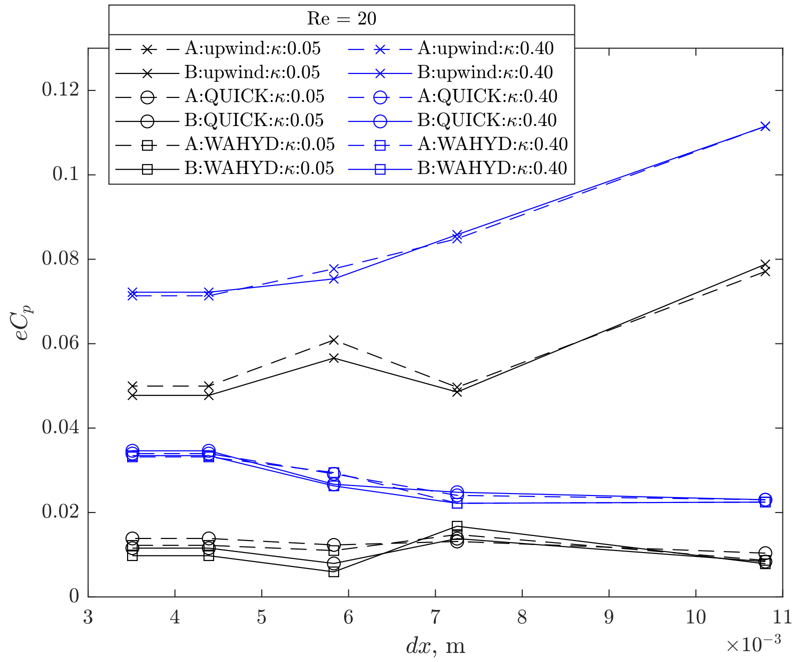


Figure 28. Pressure coefficient error as function of grid size for the flow around a cylinder case with Re = 20.

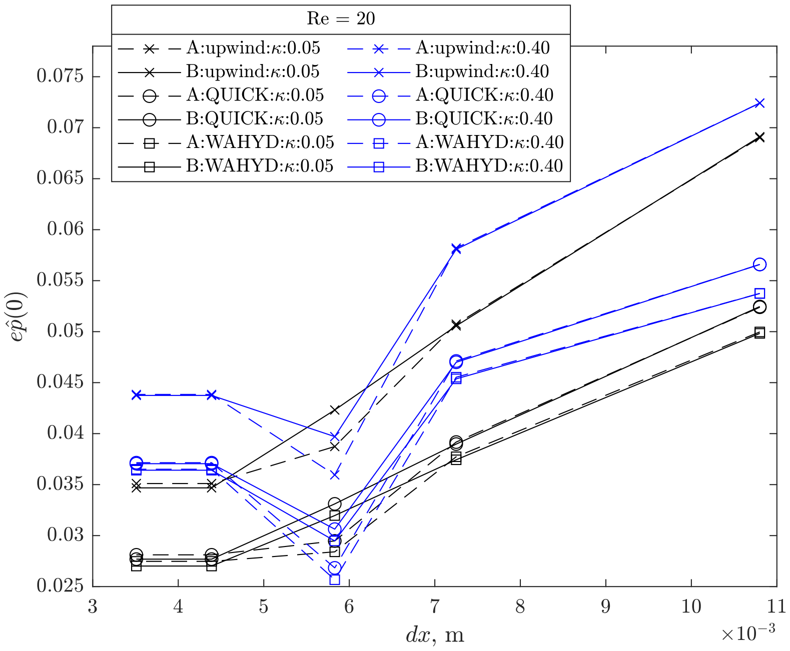


Figure 29. Dimensionless pressure error at $\theta = 0$ as function of grid size for the flow around a cylinder case with Re = 20.

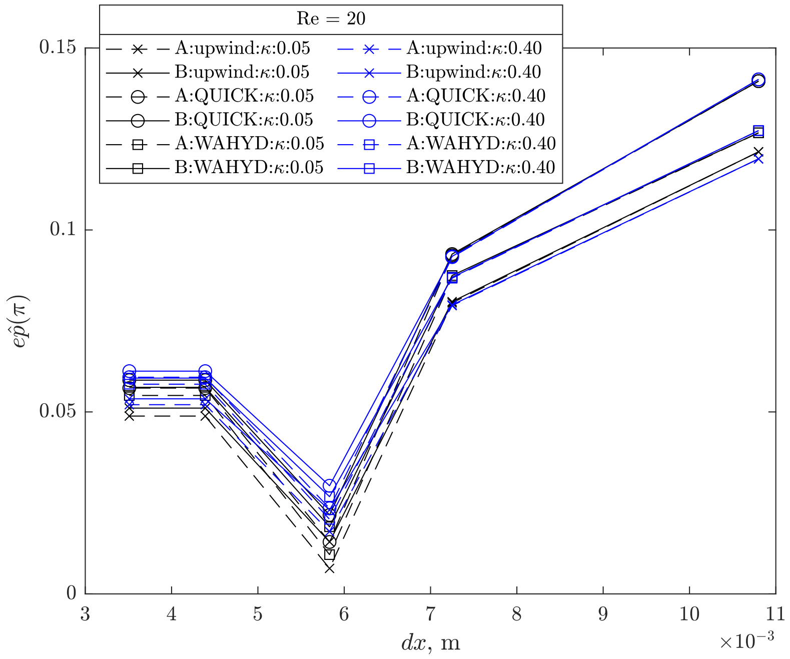


Figure 30. Dimensionless pressure error at $\theta = \pi$ as function of grid size for the flow around a cylinder case with Re = 20.

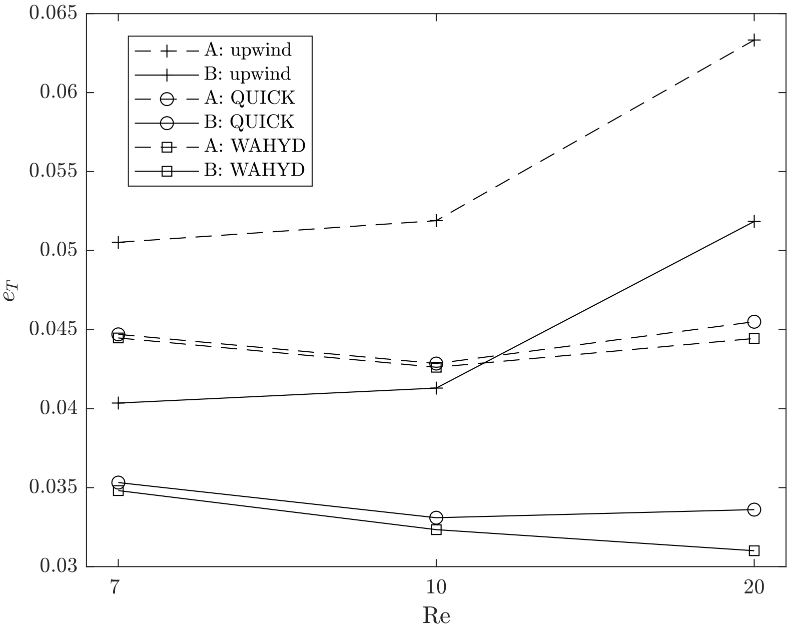


Figure 31. Total error as function of Re for the flow around a cylinder case using $\kappa = 0.05$ and the E grid.

Figures 32–34 show the computational efficiency as function of κ for each Reynolds number used. Similar to the flow between concentric cylindrical walls case, which used curves instead of lines to represent the solid body, the bigger values of η are shifted towards the smaller values of κ.

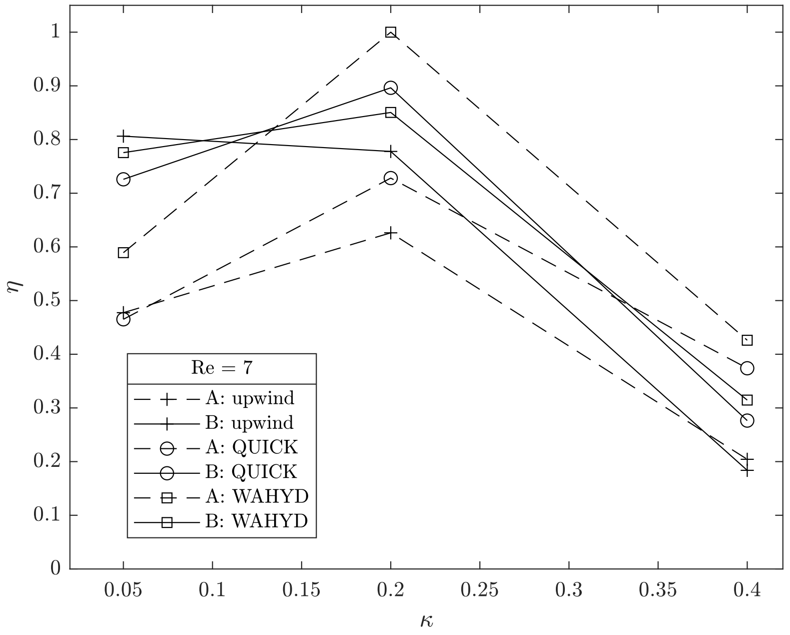


Figure 32. Computational efficiency as function of κ for the flow around a cylinder case with Re = 7 using the E grid.

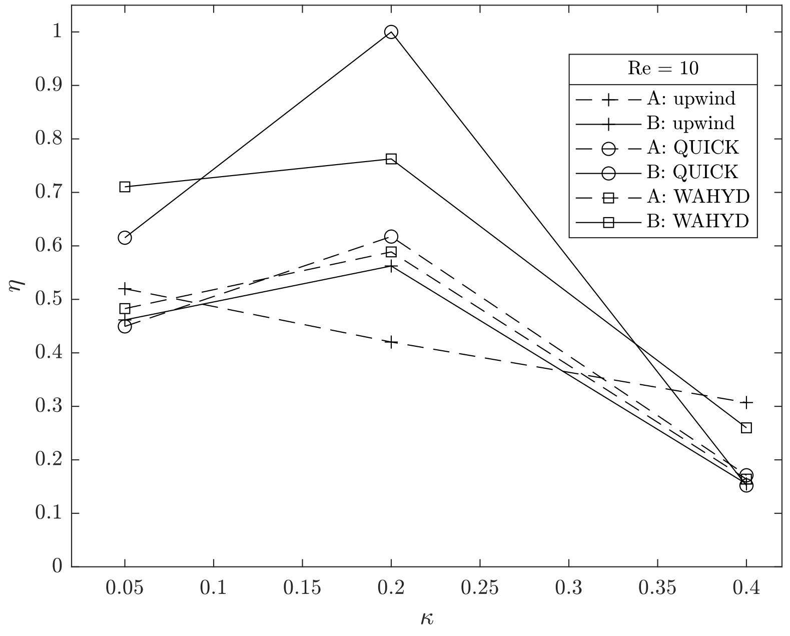


Figure 33. Computational efficiency as function of κ for the flow around a cylinder case with Re = 10 using the E grid.

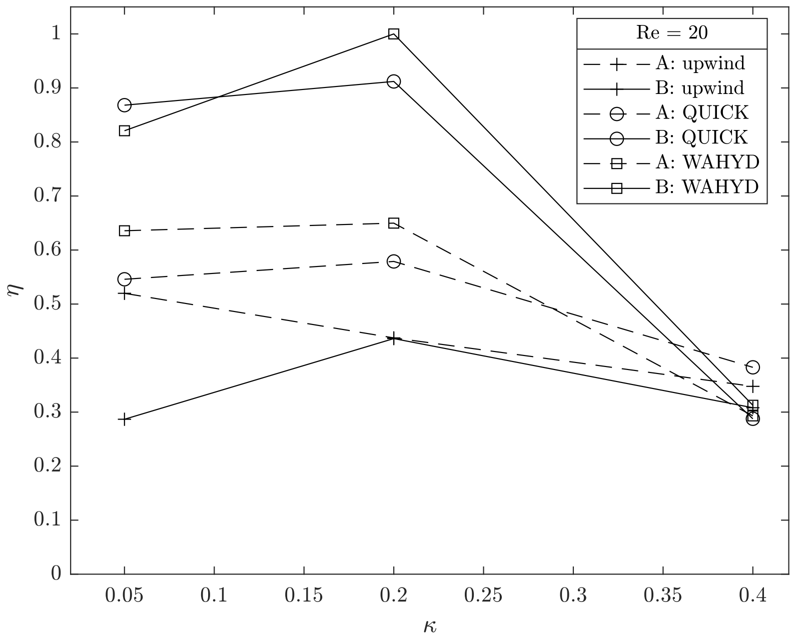


Figure 34. Computational efficiency as function of κ for the flow around a cylinder case with Re = 20 using the E grid.

5. Conclusions

The number of slave cells generated increases with the κ value resulting in an apparent enlargement of the solid body dimensions accompanied by an erosion of the smoothness of its interfacial surface. This interpretation is bounded to the treatment given to the generated slave cells but, nonetheless, a higher number of slave cells around the body representation are likely to introduce more errors to the numerical solution.

It was empirically found that smaller values of κ favor the computational efficiency of the numerical solution of problems that include an immersed solid represented by circumferences. Conversely, larger values of κ are paired with larger computational efficiency for problems characterized by solids represented by straight lines.

The B interpolation scheme produces less error and higher computational efficiencies than the A scheme, regardless of the quadric shape used to represent the solid body, when it is paired with low values of κ. The effect is generally small in the error comparison but higher in the computational efficiency. The improvement of the B scheme over the A scheme is of more significance when evaluating magnitudes which depend on gradient estimations at surfaces.

The effects described for choices of κ and interpolation scheme are valid for all the advected velocity component treatments tested, namely, upwind, QUICK and WAHYD (TVD).

Author Contributions: Conceptualization, L.H.-V., F.A. and L.L.; methodology, L.H.-V. and F.A.; software, L.H.-V., F.A. and L.L.; validation, L.H.-V., F.A. and P.D.-G.; formal analysis, L.H.-V. and F.A.; investigation, L.H.-V. and F.A.; resources, L.H.-V. and L.L.; data curation, L.H.-V., F.A. and P.D.-G.; writing—original draft preparation, L.H.-V. and F.A.; visualization, L.H.-V. and P.D.-G.; supervision, L.H.-V.; funding acquisition, L.H.-V. and P.D.-G. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by DICYT-USACH Universidad de Santiago de Chile.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the support of PFCHA CONICYT NACIONAL/2017 22170013. Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02) and by the high-performance computing system of PIDi-UTEM (SCC-PIDi-UTEM CONICYT-FONDEQUIP-EQM180180).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

QUICK	Quadratic Upstream Interpolation for Convective Kinematics
WAHYD	Water Resources Management and Modeling of Hydrosystems
TVD	Total Variation Diminished
ADI	Alternating Direction Implicit
TDMA	Tri-Diagonal Matrix Algorithm

References

1. Mittal, R.; Iaccarino, G. Immerse Boundary Methods. *Annu. Rev. Fluid Mech.* **2005**, *37*, 239–261. [CrossRef]
2. Ingram, D.; Causon, D.; Mingham, C. Developments in Cartesian cut cell methods. *Math. Comput. Simul.* **2003**, *61*, 561–572. [CrossRef]
3. Sato, N.; Takeuchi, S.; Kajishima, T.; Inagaki, M.; Horinouchi, N. A consistent direct discretization scheme on Cartesian grids for convective and conjugate heat transfer. *J. Comput. Phys.* **2016**, *321*, 76–104. [CrossRef]
4. Berger, M.; Aftosmis, M. Progress Towards a Cartesian Cut-Cell Method for Viscous Compressible Flow. In Proceedings of the 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Nashville, TN, USA, 9–12 January 2012. [CrossRef]
5. Peskin, C.S. Flow patterns around heart valves: A numerical method. *J. Comput. Phys.* **1972**, *10*, 252–271. [CrossRef]
6. Fadlun, E.; Verzicco, R.; Orlandi, P.; Mohd-Yusof, J. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *J. Comput. Phys.* **2000**, *161*, 35–60. [CrossRef]
7. Kim, J.; Kim, D.; Choi, H. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *J. Comput. Phys.* **2001**, *171*, 132–150. [CrossRef]
8. Kajishima, T.; Takiguchi, S. Interaction between particle clusters and particle-induced turbulence. *Int. J. Heat Fluid Flow* **2002**, *23*, 639–646. [CrossRef]
9. Mittal, R.; Dong, H.; Bozkurttas, M.; Najjar, F.; Vargas, A.; von Loebbecke, A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys.* **2008**, *227*, 4825–4852. [CrossRef]
10. Liu, C.; Hu, C. An immersed boundary solver for inviscid compressible flows. *Int. J. Numer. Methods Fluids* **2017**, *85*, 619–640. [CrossRef]
11. Berger, M. A Note on the Stability of Cut Cells and Cell Merging. *Appl. Numer. Math.* **2015**, *96*, 180–186. [CrossRef]
12. Dechristé, G.; Mieussens, L. A Cartesian cut cell method for rarefied flow simulations around moving obstacles. *J. Comput. Phys.* **2016**, *314*, 465–488. [CrossRef]
13. Chung, M.H. An adaptive Cartesian cut-cell method for conjugate heat transfer on arbitrarily moving fluid-solid interfaces. *Comput. Fluids* **2019**, *178*, 56–72. [CrossRef]
14. Ye, T.; Mittal, R.; Udaykumar, H.; Shyy, W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J. Comput. Phys.* **1999**, *156*, 209–240. [CrossRef]
15. Chung, M.H. Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape. *Comput. Fluids* **2006**, *35*, 607–623. [CrossRef]
16. Colella, P.; Graves, D.T.; Keen, B.J.; Modiano, D. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *J. Comput. Phys.* **2006**, *211*, 347–366. [CrossRef]
17. Kirkpatrick, M.; Armfield, S.; Kent, J. A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid. *J. Comput. Phys.* **2003**, *184*, 1–36. [CrossRef]
18. Hartmann, D.; Meinke, M.; Schröder, W. An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods. *Comput. Fluids* **2008**, *37*, 1103–1125. [CrossRef]
19. Hartmann, D.; Meinke, M.; Schröder, W. A Cartesian Cut-Cell Solver for Compressible Flows. In *Computational Science and High Performance Computing IV*; Krause, E., Shokin, Y., Resch, M., Kröner, D., Shokina, N., Eds.; Russian-German Advanced Research Workshop on Computational Science and High Performance Computing (4th:2009:Freiburg, Germany); Springer: Berlin/Heidelberg, Germany, 2011; pp. 363–376.
20. Grove, A.S.; Shair, F.H.; Petersen, E.E. An experimental investigation of the steady separated flow past a circular cylinder. *J. Fluid Mech.* **1964**, *19*, 60–80. [CrossRef]

21. Dietiker, J.F.; Guenther, C.; Syamlal, M. A Cartesian cut cell method for gas/solids flows. In Proceedings of the AIChE Annual Meeting, Nashville, TN, USA, 8–13 November 2009.
22. Li, T.; Dietiker, J.F.; Shahnam, M. MFIX simulation of NETL/PSRI challenge problem of circulating fluidized bed. *Chem. Eng. Sci.* **2012**, *84*, 746–760. [CrossRef]
23. Dietiker, J.F.; Li, T.; Garg, R.; Shahnam, M. Cartesian grid simulations of gas-solids flow systems with complex geometry. *Powder Technol.* **2013**, *235*, 696–705. [CrossRef]
24. Xie, Z.; Stoesser, T. A three-dimensional Cartesian cut-cell/volume-of-fluid method for two-phase flows with moving bodies. *J. Comput. Phys.* **2020**, *416*, 109536. [CrossRef]
25. Patankar, S. *Numerical Heat Transfer and Fluid Flow*, 1st ed.; Hemisphere Publishing Corporation: Philadelphia, PA, USA, 1980.
26. Hou, J.; Simons, F.; Hinkelmann, R. Improved total variation diminishing schemes for advection simulation on arbitrary grids. *Int. J. Numer. Methods Fluids* **2011**, *70*, 359–382. [CrossRef]
27. Ferziger, J.; Peric, M. *Computational Methods For Fluid Dynamics*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2002.
28. Henríquez-Vargas, L.; Villaroel, E.; Gutierrez, J.; Donoso-García, P. Implementation of a parallel ADI algorithm on a finite volume GPU-based elementary porous media flow computation. *J. Braz. Soc. Mech. Sci. Eng.* **2017**, *39*, 3965–3979. [CrossRef]
29. Fletcher, C. *Computational Techniques For Fluid Dynamics 1*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1990.
30. Dennis, S.C.R.; Chang, G.Z. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *J. Fluid Mech.* **1970**, *42*, 471–489. [CrossRef]