COST *vs.* ACCURACY: SECOND-ORDER *vs.* HIGH-ORDER METHODS FOR EDDY-RESOLVING SIMULATIONS OF TURBULENT SEPARATED FLOWS

***F. Capuano*[1], *N. Beratlis*[2], *F. Zhang*[2], *Y. Peet*[2], *K. Squires*[2] *and E. Balaras*[3]**

[1] ***Universitat Politècnica de Catalunya · BarcelonaTech, Barcelona, Spain***
[2] ***Arizona State University, Tempe, AZ, USA***
[3] ***George Washington University Washington DC, USA***

francesco.capuano@upc.edu; balaras@gwu.edu

Abstract

We report a comparative study of three numerical solvers for the direct numerical simulation of the flow over a sphere at Re = 3700. A high-order spectral-element code (Nek5000), a general purpose, unstructured finite-volume solver (OpenFOAM) and an in-house Cartesian solver using the immersed-boundary method (IBM) are employed for the analysis; results are compared against previous numerical and experimental data. Numerical results show that Nek5000 and the IBM code operate within a similar computational performance range, in terms of cost-vs-accuracy analysis; on the other hand, OpenFOAM needed a significantly higher number of degrees of freedom (and, overall, a higher cost) to match some of the basic features of the flow. Overall, our results suggest that high-order methods and second-order, energy-conserving approaches based on the IBM may be both viable options for high-fidelity scale-resolving simulations of turbulent flows with separation.

1 Introduction

The computational fluid dynamics (CFD) community is currently devoting major research efforts towards enabling high-fidelity, eddy-resolving simulations of complex flows within turnaround times that are compatible with industrial design cycles, as outlined by, e.g. Slotnick et al (2014) and Lohner (2019). The key to achieving this goal is arguably the adoption of numerical discretization algorithms that are simultaneously accurate, robust, flexible and, at the same time, efficient (i.e., with a high accuracy/cost ratio). As a result, there is a growing interest in understanding and comparing the accuracy and efficiency of various approaches/solvers, particularly traditional low-order methods and emerging higher-order approaches.

A systematic platform to comparatively assess the cost-efficiency of several CFD approaches is provided by The International Workshops on High-Order CFD Methods, first introduced in 2012 and held alternately between USA and EU [Wang et al (2013)]. Multiple solvers utilizing formulations of varying order of accuracy are compared for a series of flow problems of increasing complexity. The comparison is done on the basis of *error vs cost*. From the vast array of problems considered, the ones with smooth solutions and simpler geometries have demonstrated that higher-order solvers performed better than standard second-order methods; on the other hand, the results for non-smooth solutions/geometries and turbulent flows are currently inconclusive and require further research. Comparative studies have been recently performed by, e.g., Vermeire et al (2017), Capuano et al (2019) and Saini et al (2020), showing that higher-order methods can be generally more efficient compared to lower-order ones, especially when fine-grained quantities are examined.

In all of the above-mentioned studies, the CFD methods were always based on a body-fitted approach. On the other hand, the immersed boundary method (IBM) has gained enormous popularity over the past twenty years but has not been as often included in cost/accuracy studies. In this study, it is hypothesized that careful IBM implementations might be as efficient as high-order accurate methodologies.

To test this hypothesis, we comparatively examined the cost-vs-accuracy behaviour of various methods for scale-resolving simulations of the flow around a sphere. This is a prototypical case representative of turbulent incompressible flows with separation, a family of problems for which current studies on cost-vs-accuracy indicate that further research is needed. The Reynolds number was set to the sub-critical value of $Re = 3700$ to allow for comparisons with previous work. The choice of the Reynolds number also allowed for moderately large grids and affordable computations, while at the same time retaining several challenging features, such as i) thin boundary layers on the front part of the sphere; ii) flow separation near the equator of the sphere, iii) shear layer detachment and instability, with associated characteristic frequency, and iv) transition to turbulence in the wake and macro-scale vortex shedding.

The paper is organized as follows. Section 2 describes the set-up of the test case and the three numerical approaches utilized in this work. Numerical

results are reported in Section 3, along with a critical discussion of the comparative performances of the three codes. Concluding remarks are finally given in Section 4.

2 Set-up of numerical simulations

We examine the performance of three representative numerical techniques: i) second-order unstructured finite-volume method; ii) second-order Cartesian method with IBM; and iii) spectral-element method. We selected two well-known, open-source implementations for methods i) and iii), i.e. OpenFOAM and Nek5000 respectively. Although there are open-source implementations for IBM, their userbase is not as extensive as the other two solvers and the adoption of one over another will be fairly arbitrary. For this reason we selected our in-house code as a characteristic example of a Cartesian IB solver, which has been extensively validated in a variety of complex turbulent flow problems [Yang and Balaras (2006)].

The dynamics of the flow problem under consideration is governed by the incompressible Navier-Stokes equations. A cylindrical domain was used in all simulations, with the sphere located at $r = 0, x = 0$. The inlet, outlet and freestream boundaries are located at $5D$, $20D$ and $7D$ from the center respectively. Three sets of meshes of increasing resolution are constructed for this study and tested for each solver, referred to as a coarse mesh, a medium mesh, and a fine mesh.

The computational grids for all different methods were carefully designed to meet the following criteria: i) place a minimum number of points in the laminar boundary layer along the upstream portion of the sphere, which is nonetheless sufficient to correctly capture the onset of separation; ii) resolve the energetic turbulent structures in the near wake, $x/D \leq 5$. The thickness of the boundary layer was determined by a dedicated study, in which the axisymmetric boundary layer equations were solved. Fig. 1 shows the evolution of the boundary layer thickness, $\delta(\theta)$, on the front part of the sphere for three different Reynolds numbers, $Re = 370$, $Re = 3700$ and $Re = 37000$; θ denotes the polar angle from the front stagnation point. For all Reynolds numbers the boundary layer thickness grows in a similar way and scales by the established $Re^{-0.5}$ law. The $\delta(\theta)/D$ above was used to guide the design of the computational grid near the surface of the sphere; care was taken to ensure that the grid resolution at the wall is comparable for the three solvers. In the next subsections we provide information on the setup for each solver.

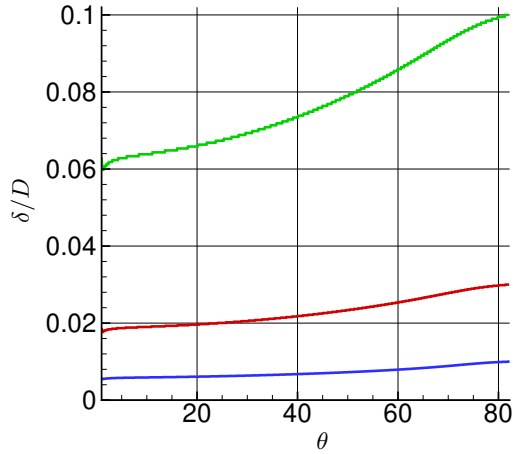

Figure 1: Boundary layer thickness on a surface of a sphere at various Reynolds numbers. Green: $Re = 370$; Red: $Re = 3700$; Blue: $Re = 37000$.

Spectral-element method: Nek5000

The open-source spectral-element code Nek5000 is employed in this work as a representative example of high-order methods. Nek5000 is a scalable solver that has been widely used for computations of fluid flows, heat transfer and magnetohydrodynamics, among others. The incompressible version is used for this work. With a spectral-element formulation, the computational domain is divided into N_e non-overlapping elements, which are related to the reference elements in the computational domain $[-1, 1]^{\text{dim}}$ by an appropriate transformation, where dim $= 3$ for the three-dimensional cases. In the version of Nek5000 used in the current study, velocity and pressure are approximated with a P_N/P_{N-2} formulation. In the current cases the polynomial order is chosen as $N = 5$. A semi-implicit time advancement scheme is used, where convection terms are marched explicitly with the second-order extrapolation in time and diffusion terms are integrated implicitly with the second-order backward difference scheme. The code is fully dealiased using an over-integration technique, the velocity is solved using the preconditioned conjugate gradient (PCG) method, and the pressure solver uses the iterative generalized mean residual (GMRES) method in the Krylov subspace. The tolerance was set to 10^{-6}. Further details of the current spectral-element methodology can be found in Zhang and Peet (2022).

An O-grid approach is utilized to produce the meshes, with a local refinement within the boundary layer around the sphere, with a minimum element height $\Delta r_{min} = 0.015D$ in a fine case. The mesh is entirely hexahedral and is generated in Ansys Icem. The total number of degrees of freedom (DOFs) for the coarse, medium and fine cases are approximately 2.8 millions, 6.5 millions and 9.8 millions, respectively. A view of the mesh around the sphere for the coarse grid is shown in Figure 2a. The no-slip wall boundary condition is employed for the surface of the sphere. A uniform Dirichlet boundary condition is used for the inflow boundary, while all the other boundaries of the cylindrical computational domain are set to stabilized outflow boundary condition. The time step was set to $2 \times 10^{-3} D/U$ for all the cases. This resulted in the

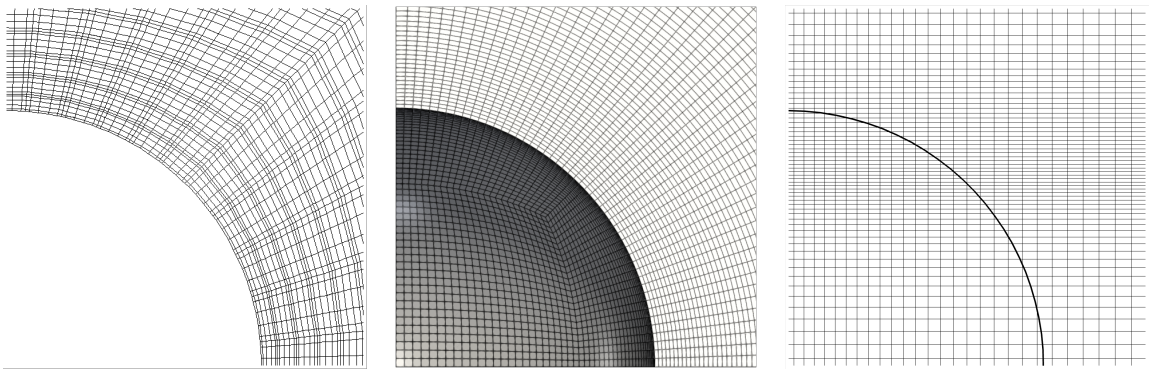

Figure 2: Comparison of the coarsest mesh used for each code. Left: Nek5000; Middle: OpenFOAM; Right: IBM.

CFL numbers being 0.65, 0.77 and 0.88 for the coarse, medium and fine grids, respectively.

Unstructured finite-volume method: OpenFOAM

OpenFOAM is a general purpose open-source platform with a wide range of possible applications. The native incompressible flow solver *pimpleFoam* from Version 7.0 of the software is used in this work. The discretization is based on a finite-volume approach with collocated arrangement, with both convective and diffusive terms approximated by second-order centered schemes. Time integration is achieved by a second-order Crank-Nicolson method. The overall solution methodology is based on the PIMPLE algorithm, which is a slightly modified version of the original PISO formulation. The reader is referred to Jasak (1996) for a detailed description of the spatial discretization methods employed by OpenFOAM. Here we highlight that the standard velocity interpolation procedure utilized in *pimpleFoam*, which is based on a cell center distance weighted average, is not discretely kinetic-energy preserving.In this work, the linear system arising from the pressure equation is solved by a generalized geometric-algebraic multi-grid (GAMG) method with a tolerance of 10^{-7}.

Three O-grid meshes, each with a successive 50% refinement, were created using the built-in OpenFOAM tool *blockMesh*. A cylindrical outer domain was used in all cases. The grid is qualitatively similar to the one used in the Nek5000 simulations (see Fig. 2a-b). The grid around the sphere was stretched in the radial direction to resolve the boundary layer while uniform elements were employed in the other two directions around the sphere. For the finest grid the minimum element height at the surface of the sphere is $\Delta r_{min} = 0.0036D$ and $0.0078D$ in the other two directions. The grid was kept uniform in a region in the near wake up to $x = 3.8D$ with an axial resolution $\Delta x = 0.009D$ and a resolution of $0.016D$ in the other two directions. The total number of elements for the coarse, medium and fine cases are approximately 3.7, 13.5 and 30 million nodes respectively.

No-slip wall boundary condition was specified at the sphere, while symmetry was used at the outer cylindrical boundaries and an outlet boundary condition at the outflow. The time step for the finest grid was set to $\Delta t = 2.5 \times 10^{-3} D/U$ ensuring the Courant number was slightly below 1.0. The simulation was integrated in time for $200D/U$ to get rid of transient effects and time-averaged statistics were sampled for another $300D/U$.

Immersed-boundary method

The Navier-Stokes equations for viscous incompressible flow are solved on a structured grid in cylindrical coordinates. The governing equations are advanced in time using a semi-implicit projection method, treating the explicit part with a 3^{rd} order Runge-Kutta scheme, and the implicit part with a 2^{nd} order Crank-Nicholson scheme. All spatial derivatives are approximated using second-order central-differences on a staggered grid; this method is discretely kinetic-energy-preserving (KEP) in the inviscid limit.The pressure Poisson equation is solved using a direct solver. The eptadiagonal matrix is initially reduced to a series of pentadiagonal problems via Fast Fourier Transforms (FFTs) along the azimuthal direction, which are then solved with a divide-and-concur strategy. The geometry of the sphere is represented by a Lagrangian grid consisting of triangular elements, and the no-slip boundary condition is imposed using an immersed-boundary formulation which is overall second-order accurate; details are provided in Yang and Balaras (2006) and a validation for a similar problem in, e.g. Pal et al (2017).

The design of the grid for this solver is not trivial since the grid does not conform to the surface. To control the distribution of the points inside the boundary layer the orientation of the outward normal with respect to the radial and axial directions was taken into account. In particular at $\theta \sim 84^{\circ}$, the normal pointing away from the sphere is almost aligned with the radial direction. The radial grid needs to be fine there while the axial grid can be relaxed. Since $\delta \sim 0.03D$ at $\theta \sim 84^{\circ}$ the radial resolution dr is $0.0045D$ resulting in approximately 7 points within the boundary layer.

Run / Reference	C_D	ΔC_D	St	ΔSt	L/D	ΔL	θ_s	$\Delta\theta$
N1	0.381	1.06%	0.23	4.54%	2.5	-5.4%	90.2°	0.33%
N2	0.376	-0.27%	0.22	0.00%	2.65	-0.6%	89.9°	0.00%
N3	0.377	–	0.22	–	2.67	–	89.9°	–
OF1	0.402	6.63%	0.21	-4.54%	2.22	-14.2%	88.5°	-1.6%
OF2	0.386	2.39%	0.21	-4.54%	2.48	-5.9%	89.2°	-0.7%
OF3	0.384	1.86%	0.21	-4.54%	2.36	-9.8%	89.8°	-0.1%
IBM1	0.356	-5.62%	0.24	9.09%	1.88	-24.9%	103.0°	14.5%
IBM2	0.370	-1.87%	0.23	2.22%	2.28	-12.3%	95.1°	5.8%
IBM3	0.371	-1.60%	0.22	0.00%	2.64	-0.9%	91.6°	1.9%
Rodriguez et al (2011)	0.394		0.215		2.28		89.4°	
Yun et al (2006)	0.355		0.21		2.62		90°	
Sakamoto et al (1990)	–		0.204		–		–	
Kim and Durbin (1998)	–		0.225		–		–	

Table 1: Main flow parameters provided by: i) the simulations carried out in this work (top part of the table) and ii) previous references (bottom part of the table). C_D is the drag coefficient, St the Strouhal number, L/D the non-dimensional length of the recirculation bubble, and θ_s the angle at which boundary layer separation occurs on the sphere. The percentage errors in the simulations are with respect to case N3.

At $\theta = 45°$ ($r/D \sim 0.35$, $z/D \sim 0.35$) the outward normal is not aligned with either axis. To maintain the same resolution normal to the surface of the sphere the grid resolution is increased to $\Delta r = \Delta z = 0.003D$, resulting in approximately 6 points inside the boundary layer. At the stagnation point, where the normal is aligned with the axial direction, the axial grid is refined while the radial grid is coarsened. There $dz = 0.003D$, which puts 5-6 points inside the boundary layer. Finally at the back of the sphere the axial grid resolution is slightly relaxed since the flow is separated. Three different grids each with 50% refinement were employed; a coarse with $275 \times 84 \times 150$ points, an intermediate with $375 \times 118 \times 225$ points and a fine with $550 \times 161 \times 300$ points in the axial, radial and azimuthal directions respectively. The CFL number for all simulations was set to $CFL = 1.2$.

3 Results and discussion

This section reports a selection of results. In all cases, flow statistics were collected for at least $200D/U$ units after a transient of $200D/U$.

The main flow parameters for each simulation are listed in Table 1, along with those provided by previous computational and experimental studies. Listed in the table are the predicted drag coefficient, $C_D = 2F_D/\rho U_\infty^2 A$ (where F_D is the drag force on the sphere, ρ is the fluid density, and A is the projected frontal area), the Strouhal number, $St = fD/U_\infty$ (where f is the shedding frequency), the recirculation bubble length, L/D, and the location where the laminar boundary layer separates indicated by the separation angle, θ_s. For the nine computations considered in this work the error for each variable with respect to the one predicted by finest grid in the Nek5000 simulation (N3) is also indicated. Available results from experiments and computations in the literature have also been added. For the case of Nek5000 all the above quantities converge and the difference between the two finest grids, N2 and N3, is less than 1%. The same applies to the IBM solver where all quantities converge as the grid is refined and the error on the finest grid (IBM3) is within 2% of N3. For the case of OpenFOAM the drag coefficient, C_D, also converges as the grid is refined and is within 2% of N3. The same applies to the separation angle, θ_s. The error on the recirculation bubble, however, does not converge monotonically and is relatively large when compared to N3 ($> 5\%$). In the process of designing the OpenFOAM grids we found that L/D is very sensitive to the adopted grid types and particularly to the way the mesh transitions from the boundary layer to the wake. The selected configuration was the one minimizing such errors. Note that St number predicted by OpenFOAM is not affected by the grid resolution and is 4.5% lower than N3. At the bottom part of Table 1 we also include the results from past experiments and eddy resolving simulations at the same Reynolds number range. Although not all the parameters listed above are provided, overall the present computations are in agreement within 5%. One notable difference is the recirculation length L/D in the DNS by Rodriguez et al (2011), which is underpredicted by 12% with respect to N3 and IBM3. We should note that given the lack of a comprehensive set of reference results we will use the grid-converged, spectral-element results for case N3 as the reference to compute all related errors involved in the assessment of cost-vs-accuracy below.

To shed light on the dynamics of the recirculation bubble, Figure 3 shows contours of the streamwise turbulent intensities $u'_x u'_x$ and cross Reynolds stress $u'_x u'_r$ for the three solvers at the highest grid resolutions. All solvers predict a similar evolution of the near wake velocity fluctuations. However, close examination of the streamwise component, $u'_x u'_x$, reveals that

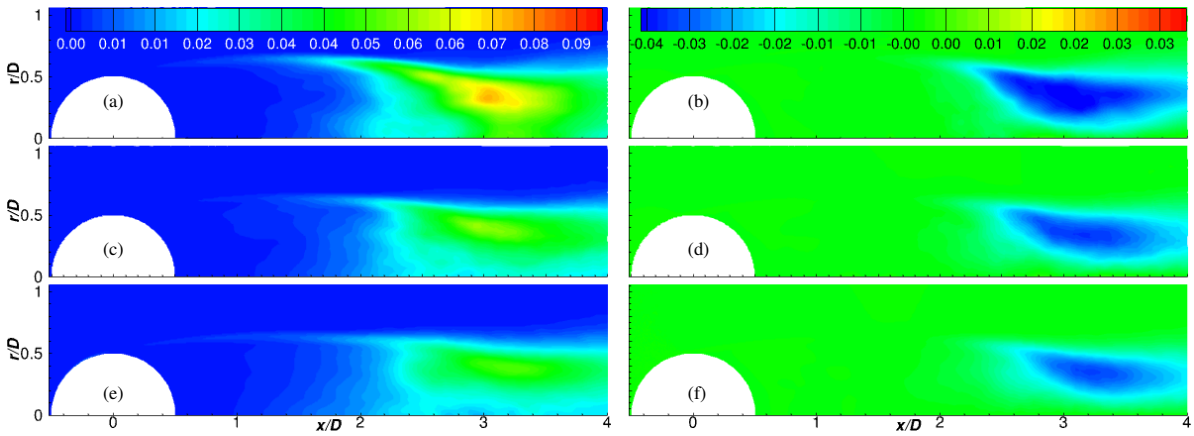

Figure 3: Contours of the streamwise turbulent intensity $u'_x u'_x$ (left) and turbulent Reynolds stresses $u'_x u'_r$ (right) (a) $u'_x u'_x$, OpenFOAM; (b) $u'_x u'_r$, OpenFOAM; (c) $u'_x u'_x$, Nek5000; (d) $u'_x u'_r$, Nek5000; (e) $u'_x u'_x$, IBM; (f) $u'_x u'_r$, IBM.

Run	#dof/core	$t^* \times N_p$	$\frac{t^* \times N_p}{\#\text{dof}}$	W
N1	20.6	1.0	0.36	0.14
N2	23.9	2.6	0.40	0.36
N3	24.0	3.9	0.40	0.54
OF1	92.5	5.96	1.61	0.33
OF2	167.5	56.4	4.21	4.47
OF3	148.5	117.9	3.97	13.10
IBM1	145	0.4	0.11	0.01
IBM2	410	1.2	0.11	0.06
IBM3	544	3.6	0.13	0.24

Table 2: Computational performances parameters. See text for details. Note: #dof/core to be multiplied by 10^3; $\frac{t^* \times N_p}{\#\text{dof}}$ by 10^{-6}.

the detached shear layer in OpenFOAM becomes unstable closer to the sphere, while the magnitude of both $u'_x u'_x$ and $u'_x u'_r$ is generally higher. The peak in both quantities is over-predicted by OpenFOAM, while the maximum discrepancy is seen at around $x = 2.5D$, where the peaks are roughly double in magnitude. This behaviour is probably due an early breakdown of the shear layer, leading to a shorter recirculation bubble. The finite-difference IBM solver slightly underpredicts $u'_x u'_x$ compared to Nek5000.

Finally, we examine the efficiency of the three codes in terms of cost-vs-accuracy behavior. All Nek5000 simulations were conducted on the KNL nodes of the Stampede-2 supercomputer at the Texas Advanced Computing Center located at The University of Texas at Austin. They utilize Intel Xeon Phi 7250 CPUs with a clock rate of 1.4GHz and 96GB DDR4 plus 16GB high-speed MCDRAM for RAM. For the IBM and OpenFOAM computations we used the PEGASUS HPC cluster at the George Washington University consisting of Dual 20-Core 3.70GHz Intel Xeon Gold 6148 processors with 96GB of 2666MHz DDR4 ECC Register DRAM. Given that the HPC architectures utilized in the computations in the present work are similar, we will use the elapsed CPU time normalized against the open-source tool TauBench[1] to obtain comparable cost estimates. The results are reported in Table 2. The normalized elapsed CPU time per time step is indicated as $t^* = t/\tau$, where τ is the TauBench execution time. Note that t was obtained by averaging the elapsed CPU time (i.e., the actual wall time) over the number of time-steps of each simulation. The selection of degrees-of-freedom per core ($\#dof$/core) was based on published results and prior experience with the solvers; it can be assumed that all solvers were used within a very similar (linear) parallel efficiency range. The total cost can be defined as $t^* \times N_p$, where N_p is the number of processors used for each run. A metric that is directly linked to the efficiency of each solver is the normalized cost per timestep per dof ($t^* \times N_p/dof$), which is also listed in Table 2. Based on this metric OpenFOAM was slower by a factor $\approx$ 30 with respect to the IBM code and $\approx 10\times$ slower with respect to Nek5000. The IBM code was the fastest one: roughly 3 times faster than Nek5000, on average. This can be attributed to a superior efficiency of the Poisson solver implemented in the IBM code. It is worth noting that the time step size is not involved in this comparison, which can be misleading as all three solvers use different time steps. To account for such differences we define the cost of each simulation as the CPU hours required to compute one flow-through time (D/U_∞). The resulting Work Unit (W) is thus defined as:

$$W = \frac{t^* \times N_p}{3600} \times \frac{D/U_\infty}{\Delta t}, \qquad (1)$$

where Δt is the time step size. Even though the codes employ different time integration schemes, the time step was not drastically different for the various runs, $\Delta t \approx O(10^{-3})$. The work unit W is shown in Figure 4 for all nine computations versus the error on the integral parameters listed in Table 1. Nek5000 rapidly

[1] https://github.com/slitvinov/taubench

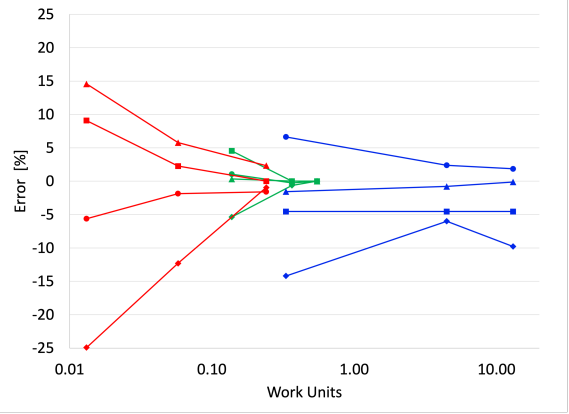

Figure 4: Cost (W) vs accuracy (% error) comparison for all cases. Symbols indicate errors on: ∘ C_D; □ St; ◊ L/D; △ θ_s. Green: Nek5000; Blue: OpenFOAM; Red: IBM.

converges towards negligible errors at the medium grid resolution for all quantities. For the IBM solver the errors are significant for the coarsest grid but also converge rapidly as the grid is refined. For OpenFOAM, grid convergence is not as rapid and some quantities such as the Strouhal number, St and the length of the recirculation bubble, L/D, still have errors of the order of 5% even on the finest grid. We did not attempt further grid refinement for OpenFOAM given that the unit-cost, W, for the finest grid is 13.1, which is already an order of magnitude higher than the other two solvers, which are clearly more efficient. The power metrics for Nek5000 could be further improved by utilizing larger time steps. However, the sensitivity of results to a temporal refinement was not performed in the current study. Overall, if one sets the threshold of the error to be $\leq 2\%$, the IBM solver delivers it at the lowest cost ($W = 0.24$).

4 Conclusions

This comparative study analyzed the efficiency of three radically different numerical methodologies for scale-resolving simulations of a prototype case of a turbulent separated flow. The results showed that Nek5000 and the IBM code operate in a similar computational performance range for the problem considered, while OpenFOAM (based on one of its standard incompressible solvers) turned out to be significantly less efficient. While caution should be taken in terms of generalizing the reported findings, our results suggest that high-order methods and second-order, kinetic-energy-preserving approaches based on the IBM may be both viable and efficient options for high-fidelity scale-resolving simulations of turbulent separated flows. It should be noted that grid generation costs, which can be an important factor, were not included in the current cost-vs-accuracy estimates, and can further tip the balance towards IBM approaches.

Acknowledgments

FZ and YP would like to acknowledge the support of NSF CBET 1707075 and 1944568 grants for this work. FC is a Serra Húnter fellow.

References

Capuano, F., Palumbo, A. and de Luca, L. (2019), Comparative study of spectral-element and finite-volume solvers for direct numerical simulation of synthetic jets, *Comput. Fluids*, Vol. 179, pp. 228–237.

Jasak, H. (1996), Error analysis and estimation for the finite volume method with applications to fluid flows.

Kim, H. J. and Durbin, P. A. (1988), Observations of the frequencies in a sphere wake and of drag increase by acoustic excitation, *Phys. Fluids*, Vol. 31(11), pp. 3260–3265.

Lohner, R. (2019), Towards overcoming the LES crisis, *Int. J. Comput. Fluid Dyn.*, Vol. 33(3), pp. 587–597.

Pal, A., Sarkar, S., Posa, A. and Balaras, E. (2017), Direct numerical simulation of stratified flow past a sphere at a subcritical Reynolds number of 3700 and moderate Froude number, *J. Fluid Mech.*, Vol. 826, pp. 5–31.

Rodriguez, I., Borell, R., Lehmkuhl, O., Segarra, C. D. P. and Oliva, A. (2011), Direct numerical simulation of the flow over a sphere at Re= 3700, *J. Fluid Mech.*, Vol. 679, pp. 263–287.

Sakamoto, H. and Haniu, H. (1990), A study on vortex shedding from spheres in a uniform flow, *J. Fluids Eng.*, Vol. 112(4), pp. 386–392.

Saini, V., Xia, H. and Page, G. J. (2022), Comparison of finite-volume and spectral/hp methods for large-eddy simulation of combustor port flow, *AIAA J.*, Vol. 60(7), pp. 4367–4383.

Slotnick, J. P., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E. and Mavriplis, D. J. (2014), CFD vision 2030 study: a path to revolutionary computational aerosciences. NASA Report No. NF1676L-18332.

Vermeire, B. C., Witherden, F. D. and Vincent, P. E. (2017), On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools, *J. Comput. Phys.*, Vol. 334, pp. 497–521.

Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., ... and Visbal, M. (2013), High-order CFD methods: current status and perspective. *Int. J. Numer. Methods Fluids*, Vol. 72(8), pp. 811–845.

Yang, J. and Balaras, E. (2006), An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *J. Comput. Phys.*, Vol. 215(1), pp. 12–40.

Yun, G., Kim, D. and Choi, H. (2006), Vortical structures behind a sphere at subcritical Reynolds numbers, *Phys. Fluids*, Vol. 18(1).

Zhang, F. and Peet, Y. T. (2022), The dynamics of coherent structures in a turbulent wake past a sphere at Re=3700, Proceedings of 12th Int. Symp. on Turbulence and Shear Flow Phenomena, Osaka, Japan.