6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11-15 June 2018, Glasgow, UK

# PERFORMANCES OF TWO OPEN-SOURCE SOLVERS IN THE NUMERICAL SIMULATION OF SYNTHETIC JETS

**ANDREA PALUMBO[1], FRANCESCO CAPUANO[1] and LUIGI DE LUCA[1]**

[1] Department of Industrial Engineering, Università di Napoli Federico II, Napoli, 80125, Italy,
{andrea.palumbo, francesco.capuano, deluca}@unina.it

**Key words:** Synthetic Jets, DNS, Solver Efficiency, Nek5000, OpenFOAM

**Abstract.** We report direct numerical simulations of a synthetic jet issuing in a quiescent environment performed by two widespread open-source computational fluid dynamics (CFD) codes: Nek5000 and the OpenFOAM solver *pimpleFoam*. While the former employs a high-order spectral-element method, the latter is based on finite-volume, lower-order schemes. The flow parameters are based on the experimental data of Yao *et al.* (2006). The performances of the two codes are compared thoroughly, with regard to accuracy as well as computational cost. Our computations show that Nek5000 is globally more efficient than *pimpleFoam* for the problem under study.

## 1 INTRODUCTION

Synthetic jets are fluid mechanical devices employed in several technological applications, such as flow control, electronic cooling, jet vectoring, and many others [1–4].

The realization of a synthetic jet requires the design of an actuator which is capable of generating a periodic alternation of fluid ejection and suction from a cavity. The resulting flow field is rather complex, being characterized by vortex formation and instability, transitional behaviour, and fully-developed turbulent flow [5]. A thorough understanding of the flow physics of the synthetic jet issuing in an external environment is essential to properly calibrate the device for the relevant application. In recent years, massive experimental [6–8] and computational [9–11] efforts have been made to this end. Overall, a rather wide variation in the results has been observed, and it is not clear which specific technique and modelling approach are the best suited in predicting the flow behaviour. From a computational perspective, direct numerical simulation (DNS) is in principle the method of choice to gather reliable insights into the flow physics; on the other hand, the achievement of high-fidelity results at high computational efficiency is a key issue towards the use of CFD for design and engineering purposes.

In this work, we report a comparison between direct numerical simulations of a synthetic jet field, issuing in a quiescent ambient, performed by two popular open-source CFD codes, Nek5000 and the OpenFOAM native solver *pimpleFoam*. The former employs a high-order spectral-element method, whereas the latter is based on finite-volume, lower-order schemes. The computational setup targets the experimental campaign described in Yao *et al.* [12]. Following the results by Kotapati *et al.* [13], a reasonably simplified geometry of the device is adopted, whereas the flow non-dimensional parameters of the simulations match the experiments. At first, the streamwise evolution of the flow field is considered, with particular focus on the trajectory of the vortices expelled from the cavity orifice. Then, time- and phase-averaged statistics in the streamwise and crosswise directions are analyzed. The code-to-code comparison and the agreement with experimental data are discussed. Finally, the computational efficiency of the two codes is assessed for different grid resolutions.

## 2 SETUP OF NUMERICAL SIMULATIONS

### 2.1 Computational model and boundary conditions

The computational domain used in the simulations is shown in Fig. 1, along with the boundary conditions enforced on each boundary face. The domain comprises the cavity, the orifice neck (having a slotted base), and the external field. The actuator geometry is a simple parallelepiped which can be fully characterized by four non-dimensional parameters: aspect ratio $l_c/h_c$, slot height-to-width ratio $l/h$, cavity-to-slot width ratio $h_c/h$ and slot spanwise aspect ratio $L_z/h$. In the present simulations of course this last is a mere computational parameter. The current simulations employ $l_c/h_c = 2.0$, $l/h = 2.6$, $h_c/h = 2.45$ and $L_z/h = 9$. The other external field dimensions are chosen to be $L_x/h = L_y/h = 60$.

On the bottom side of the cavity, $x_c$, the diaphragm motion is mimicked by an oscillatory boundary condition for the wall-normal velocity

$$U(x = x_c, y, z, t) = U_w \sin(2\pi f t)\ , \tag{1}$$

with $f$ being the actuation natural frequency and $t$ the time. It should be noted that the enforcement of an autonomous oscillatory diaphragm motion corresponds to decoupling the structural oscillator (diaphragm) and the acoustic (Helmholtz) one [14]. A no-slip condition is used for all the other solid surfaces of the cavity and the external flat plate, while the external field faces are modeled as outflow boundaries. The flow is assumed to be homogeneous along the spanwise direction and thus periodic boundary conditions are applied for the faces normal to $z$.

The computational parameters are chosen to match the experimental configuration reported in Yao *et al.* [12], so that a reference solution is available. To allow for a fair comparison, the modeling approach suggested in [13] is employed: while the cavity geometry is simplified (based on the results in [15]), the relevant nondimensional numbers of the jet are matched between computation and experiments. The relatively high aspect ratio of the slot orifice used in the experimental campaign ($AR = 28$) justifies the assumption of homogeneous spanwise flow. Furthermore, in experiments the diaphragm was driven at a frequency close to its natural one, and sufficiently lower than the cavity Helmholtz frequency, leading to a nearly incompressible operating regime [16]. Therefore, the synthetic jet is assumed to be completely determined by the following parameters: a properly defined jet velocity $U_j$, the orifice width $h$, the actuation frequency $f$ and the kinematic viscosity $\nu$. Following [13], the velocity $U_j$ is defined as

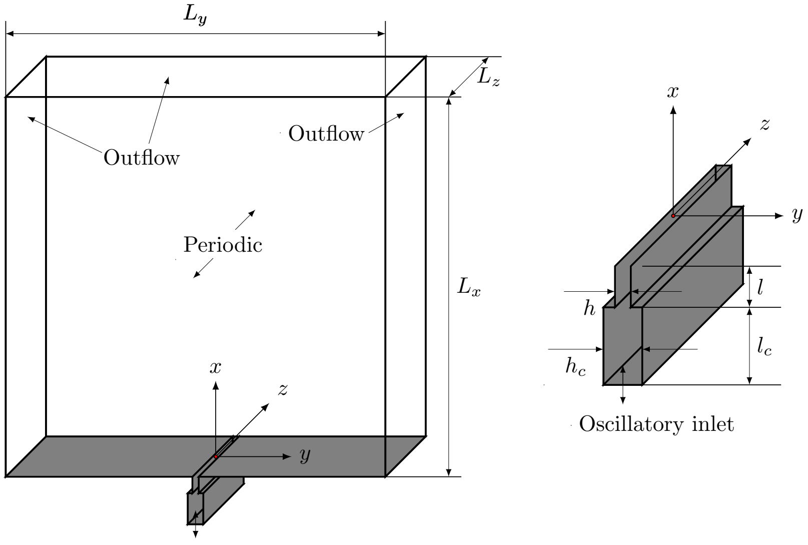

**Figure 1**: Computational domain and boundary conditions used in the numerical simulations. The gray patches are solid walls. An enlarged view of the cavity is shown in the right part of the figure.

$$U_j = \frac{1}{A}\frac{2}{T}\int_0^{T/2}\int_A U(x=0,y,z,t)\,\mathrm{d}A\,\mathrm{d}t\ , \tag{2}$$

where $A$ is the cross-sectional area of the slot and $T = 1/f$ is the time period of the jet cycle. Note that a simple relation between the velocity amplitude $U_w$ and $U_j$ can be derived from the application of mass conservation, $U_j = (2/\pi)U_w(h_c/h)$. In the present simulations, $U_j$ is set to 1 for simplicity and $U_w$ is obtained by the previous relation. The remaining parameters are chosen to yield a Reynolds number $\mathrm{Re} = U_j h/\nu = 1150$ and a Strouhal number $\mathrm{St} = fh/U_j = 0.04$, in accordance with the experimental data [12].

## 2.2 Numerical methods

Two codes have been used to solve the incompressible Navier-Stokes equations, equipped with boundary conditions, within the above mentioned computational domain. The solvers are briefly described in the following.

### 2.2.1 OpenFOAM

OpenFOAM is a general purpose open-source platform with a wide range of possible applications. Written in C++, it comes with several native *applications* (i.e., solvers) and also offers the possibility to create custom codes. The version 5.0 is used in this work [18].

The selected application is the native incompressible flow solver *pimpleFoam*. The discretization is based on a finite-volume approach, with both convective and diffusive terms approximated by second-order centred schemes. Time integration is achieved by a second-order Crank-Nicolson method. The solution algorithm is based on the PIMPLE algorithm, a slightly modified version of the PISO. The pressure linear system is solved by a generalized geometric-algebraic multi-grid (GAMG) method with a tolerance of $10^{-7}$.

Boundary conditions are implemented using the standard capabilities provided by OpenFOAM. The oscillating inlet boundary condition is assigned as follows

```
type            uniformFixedValue;
uniformValue    sine;
uniformValueCoeffs
{
    frequency 0.04;
    amplitude 0.641;
    scale (0 1 0);
    level (0 0 0);
}
value uniform (0 0 0);
```

while the outflow boundaries make use of the `inletOutlet` boundary type.

### 2.2.2 Nek5000

Nek5000 is an open-source, high-order and highly-scalable code, which is capable of simulating incompressible or low-Mach-number flows, even in presence of forced or natural convection, for stationary or moving geometries. It was written in Fortran77 and C languages, and uses Message Passing Interface for the parallelization. The version `r1115` of the SVN repository is used in this work [19].

The spatial discretization is based on the spectral element method (SEM). This method was first introduced by Patera [17] to combine the main advantages of finite element (geometric flexibility) and spectral methods (high-order resolution). For these reasons the SEM is used for the simulation of transitional and turbulent flows, even in complex geometries, and has also been employed for pulsatile flows in the context of vascular applications [20]. The domain is partitioned in a finite number of non-overlapping hexahedral elements: the $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation is used, thus the velocity and pressure fields are obtained as the combination of Lagrange polynomials (order $N$ for the velocity, $N-2$ for the pressure) based on Gauss-Lobatto-Legendre (GLL) and Gauss-Legendre (GL) quadrature points. In this work, $N = 9$ is chosen; in preliminary computations, this choice proved to be a trade-off between high-order resolution and a reasonable computational cost.

Time integration is based on a high-order splitting method (BDF3-EXT3). A third-order extrapolation is used for the convective terms (EXT3), while the viscous terms is treated implicitly. Two linear systems of equations arise: the pressure system is solved using a generalized minimal residual method (GMRES), while the linear system for the velocity field is solved using the conjugate gradient method with Jacobi preconditioning. For both solvers, the tolerance is set to $10^{-7}$.

A standard outflow condition (“O”) is used for the environment lateral boundary; it is worth to note that this option did not work at the top of the domain, due to the strong

| Solver | Acronym | $L_x/h$ | $L_y/h$ | $L_z/h$ | $N_x$ | $N_y$ | $N_z$ | #dof |
|---|---|---|---|---|---|---|---|---|
| Nek5000 | N1 | 60 | 60 | 9 | 13 | 12 | 4 | 592182 |
| Nek5000 | N2 | 60 | 60 | 9 | 19 | 18 | 6 | 1709514 |
| Nek5000 | N3 | 60 | 60 | 9 | 26 | 24 | 8 | 3953630 |
| OpenFOAM | OF1 | 60 | 60 | 9 | 120 | 120 | 36 | 655200 |
| OpenFOAM | OF2 | 60 | 60 | 9 | 160 | 170 | 54 | 1674000 |
| OpenFOAM | OF3 | 60 | 60 | 9 | 220 | 240 | 72 | 4075200 |

**Table 1**: Test matrix of the computational runs. The number of elements (Nek5000) and cells (OpenFOAM) along each direction refer to the external environment only. The final column reports the total number of degrees of freedom (dof) for each case.

backflow induced by the vortices, that led to a blowout of the computation. Hence, a stabilized outflow condition is used for this boundary [21].

### 2.3 Test matrix

The computational meshes have been obtained using the meshing capabilities provided by OpenFOAM (`blockMesh`) and Nek5000. After a preliminary sensitivity study, three grids have been selected and used. The resulting test matrix is reported in Table 1.

The mesh within the cavity and the orifice neck has not been changed, while the number of elements (in each direction) in the external environment has been gradually increased. In every calculation, the grid is clustered in the slot area, along both the $x$ and $y$ directions, in order to obtain a good resolution of the turbulent motion in that region, even for the coarsest grid. The intermediate resolution (case 2) is similar to the one used for the DNS performed in [13]. To allow for a fair comparison, the number of cells for the OpenFOAM runs was obtained by multiplying the number of elements used in Nek5000 by the polynomial order ($N = 9$ in the present case), so to obtain a comparable number of total degrees of freedom between the two approaches.

## 3 RESULTS AND DISCUSSION

Results of the simulations are presented and discussed in this section. As usual in periodic flows, both time- and phase-averaged velocity fields are taken into consideration. The former is indicated as $\langle U \rangle$, whereas the latter as $\overline{U}$. Both have been obtained by accumulating statistics over 10 cycles and by further spanwise averaging.

### 3.1 Streamwise evolution

An effective ensemble picture of the streamwise evolution of the flow field can be obtained by constructing a contour map of the phase-averaged $x$-velocity as a function of streamwise distance and phase. The result is shown in Fig. 2 for the experiment (left) as well as for cases N3 and OF3 (right). The experimental map was reconstructed by means of the hot-wire dataset, that consisted of phase-averaged data of the $x$-velocity every 1.6° at 42 locations along the centerline, of which 28 for $x/h < 10$. Similarly, the CFD maps were constructed by probing the simulations every 0.72°; the finest grids have roughly

100 computational points within $x/h < 10$. Superposed on the maps are also plotted the trajectories of the vortex pair, that have been obtained by tracking the peak of the velocity profile at each streamwise station, as suggested in [6].

The streamwise evolution is basically characterized by the formation and roll-up of a vortex pair, that remains attached to the orifice exit for $t/T < 0.25$. At the phase of maximum ejection, the vortices detach and start to travel downstream with an approximately constant celerity, roughly in the interval $0.35 < t/T < 0.5$. As the suction phase starts (intersection of the contour line $\overline{U}/U_j = 0$ with $x/h = 0$), the vortex pair undergoes turbulent transition and thus its celerity decreases, although the vortical structures retain a high degree of coherence also farther downstream.

At the finest grid resolution, the agreement between the two codes is remarkably good. In the first tract ($0.35 < t/T < 0.5$), the vortex celerity non-dimensionalized by $U_j$ is 0.631 for OF3 and 0.642 for N3, as computed by a least-square linear regression. For the intermediate and coarse cases, the celerity is respectively 0.636 and 0.656 for OpenFOAM, while amounts to 0.640 and 0.627 for Nek5000. Both codes are thus converging monotonically as the grid is refined. A similar behaviour is observed (not shown herein) for other quantities of engineering interest, such as the position of the saddle point $x_{\text{sp}}$, i.e., the streamwise distance above which the phase-averaged $x$-velocity is entirely positive, and the characteristic length, commonly referred to as stroke length, $L_s$,

$$L_{\text{s}} = \int_0^{T/2} \overline{U}(x=0,0,t)\,\mathrm{d}t\ . \tag{3}$$

The comparison with the experimental dataset, while showing the same qualitative trends, presents some noticeable quantitative differences, which are attributable to the inherent modeling assumptions as well as to the experimental uncertainty. There was significant disagreement between PIV and hot-wire results for this campaign, presumably due to actuator ageing effects [12]. The map in Fig. 2 was also re-built (not shown here) from the PIV dataset, which consists of phase-averaged velocity statistics at 72 evenly-spaced phases and up to $x/d \approx 6$. The cross-comparison demonstrated that the CFD results actually fall within the experimental uncertainty.

### 3.2 Crosswise profiles

The crosswise distribution of the time-averaged velocity field is reported in Fig. 3 at four streamwise stations in the near field of the synthetic jet. Except for a slight overestimation of the peak velocity at the nearest location from the two coarsest OpenFOAM simulations, there are virtually no significant differences in the computational results. The experimental data acquired from PIV are also reported for comparison, showing overall a good agreement. The streamwise jet velocity is consistently under-predicted by computations, although the discrepancy falls within the experimental uncertainty taking into account results from other techniques (e.g. hot-wire).

The crosswise profiles of phase-averaged streamwise velocity at the maximum ejection ($\phi \approx 90°$) and ingestion ($\phi \approx 270°$) phases are shown in Fig. 4. At the maximum expulsion phase (left graph), results from all the computational cases practically coincide. The

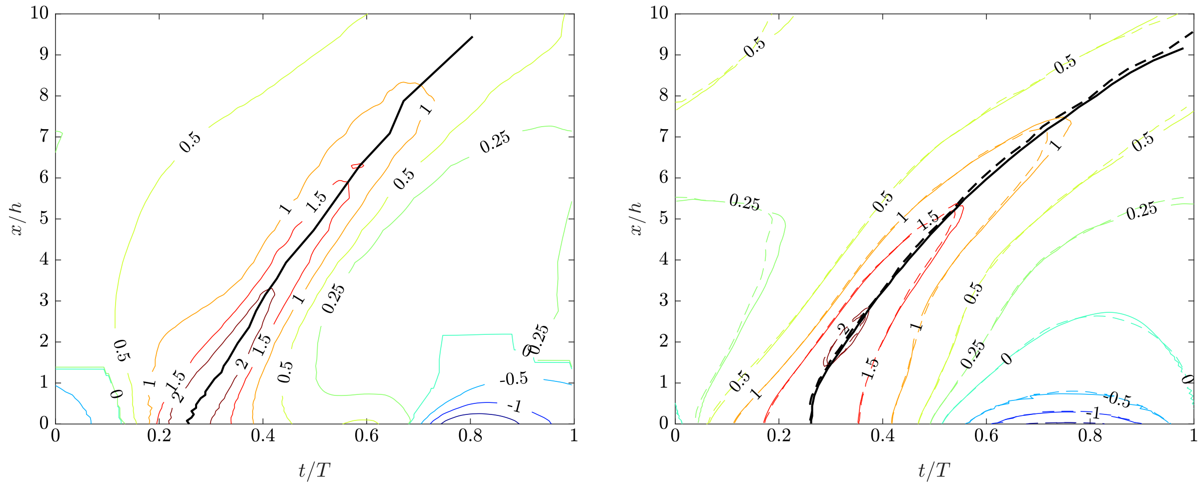

**Figure 2**: Contour map of $\overline{U}/U_j$ in a phase-space plot. (left) Results of the experiment by Yao [12] reconstructed by hot-wire data; (right) result of the runs OF3 (dashed lines) and N3 (solid lines).

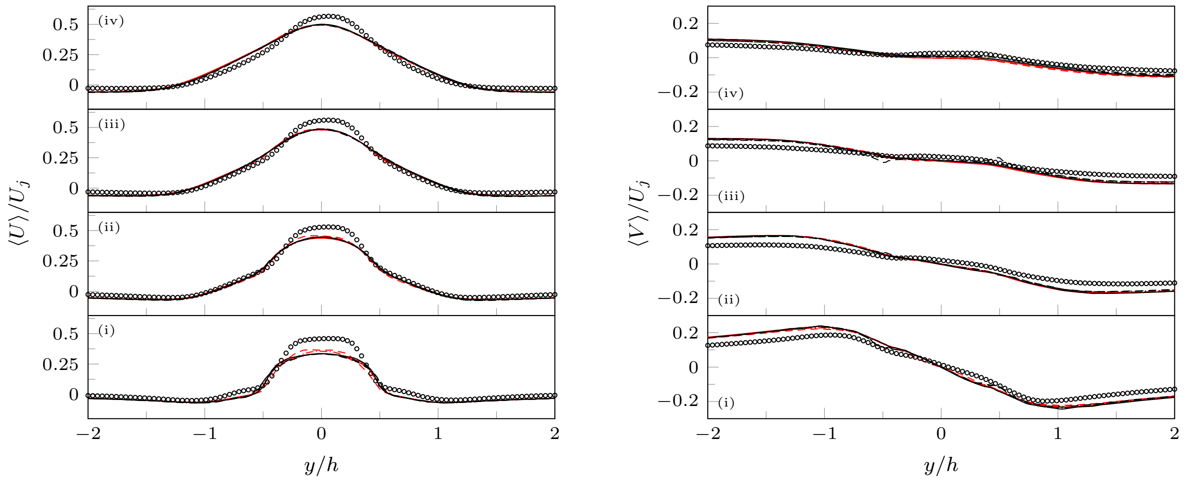

**Figure 3**: Crosswise distribution of time-averaged (a) streamwise and (b) cross-stream velocity at streamwise stations (i) $x/h = 0.5$, (ii) 1.0, (iii) 1.5, (iv) 2.0. Lines refer to the Nek5000 (black) and OpenFOAM (red) computations on the coarse (dashed), medium (dashdotted) and fine (solid) grid levels; ○ PIV.

agreement with the experiments is also very good, as both codes (for all the resolutions) are able to capture the jet-core region as well as the reverse-flow area induced by the jet entrainment. Differences clearly emerge at the maximum suction phase, reported in Fig. 4 (right), especially for what concerns the OpenFOAM simulations. The profiles for the cases OF1 and OF2 are abnormally noisy, particularly at the first two streamwise stations. In contrast, results from Nek5000 are smoother, and converge finely as the grid is refined. At stations (iii) and (iv), the code-to-code agreement and the comparison to experimental data improve satisfactorily.

Further insights can be drawn by analyzing the turbulent behaviour of the flow field. As is well known, the relevant turbulent fluctuations in periodic flows are constituted by deviations of the instantaneous velocity field with respect to phase-averaged statistics, e.g., $u = U - \overline{U}$. The root mean square (r.m.s.) streamwise velocity was thus computed as $u' = \sqrt{\langle uu \rangle}$. The crosswise distribution of $u'$ at the usual four streamwise stations is displayed in Fig. 5 again at the maximum expulsion (left) and ingestion (right) phases. At $\phi \approx 90°$, the simulation results are close to each other, although non-negligible differences are visible, with particular reference to the coarsest cases N1 and OF1. As compared to experimental data at station (i), both codes greatly over-predict the two peaks at $|y/h| \approx 0.5$, corresponding to the location of the two vortex cores. In general, the jet-core turbulence level, which settles around 0.2, is correctly captured. Conversely, major discrepancies occur at the maximum ingestion phase. During suction, experimental data show that the flow experiences partial re-laminarization, with relatively low levels of turbulent kinetic energy. The streamwise component decreases to around 5% of $U_j$. However, the OpenFOAM computations fail to capture the reduction of turbulent fluctuations and, although the residual noise decreases as the grid is refined, a significant amount of spurious r.m.s. velocity is still present for the intermediate resolution (case OF2) at the first two streamwise stations. This behaviour does not take place for the Nek5000 results, which predict the correct turbulent levels even with the coarsest grid.

### 3.3 Computational efficiency

The computational performances of the two codes have been evaluated in terms of elapsed CPU time per time step for the three grids (on the same number of processors) as well as in the context of a strong scaling test on the CINECA supercomputer Marconi [22].

Figure 6 (left) reports the CPU time per time step as a function of the CPU load, i.e., for the three cases of Table 1 (results were obtained on 68 processors). As expected, the computational effort scales linearly with the total number of degrees of freedom for both Nek5000 and OpenFOAM.

The strong scaling test was carried out for the finest grid resolution (cases N3 and OF3, see Table 1); results are shown in Fig. 6 (right). At the lowest number of processors, $N_p = 34$, OpenFOAM takes roughly 2/3 of the time with respect to Nek5000 to perform a single time step. On the other hand, the parallel performances of OpenFOAM decline quickly, leading to speedup flattening at $N_p = 272$, where the computational load per processor is of roughly 15000 cells/processor. In contrast, Nek5000 scales quite well up to $N_p = 544$. The OpenFOAM domain decomposition was handled by the native `scotch`

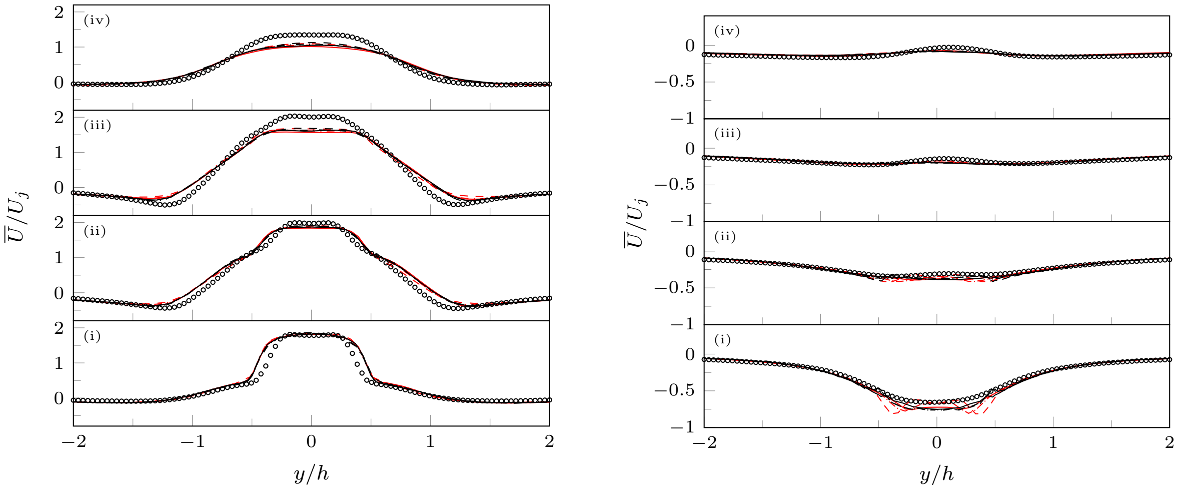

**Figure 4**: Crosswise distribution of phase-averaged streamwise velocity at (a) maximum ejection and (b) maximum ingestion phases at streamwise stations (i) $x/h = 0.5$, (ii) 1.0, (iii) 1.5, (iv) 2.0. Lines refer to the Nek5000 (black) and OpenFOAM (red) computations on the coarse (dashed), medium (dashdotted) and fine (solid) grid levels; ○ PIV.

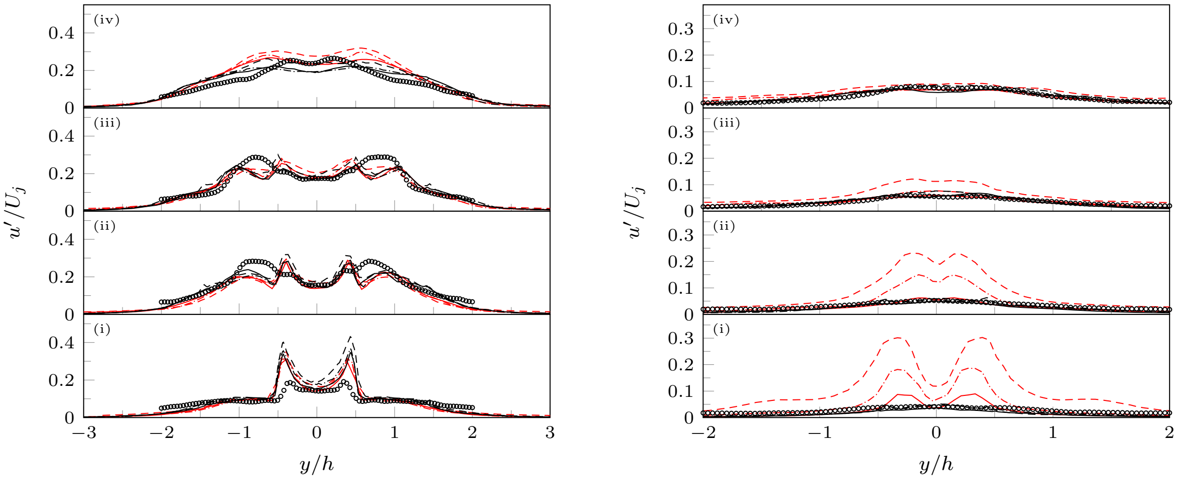

**Figure 5**: Crosswise distribution of stremwise r.m.s. velocity at (a) maximum ejection and (b) maximum ingestion phases at streamwise stations (i) $x/h = 0.5$, (ii) 1.0, (iii) 1.5, (iv) 2.0. Lines refer to the Nek5000 (black) and OpenFOAM (red) computations on the coarse (dashed), medium (dashdotted) and fine (solid) grid levels; ○ PIV.

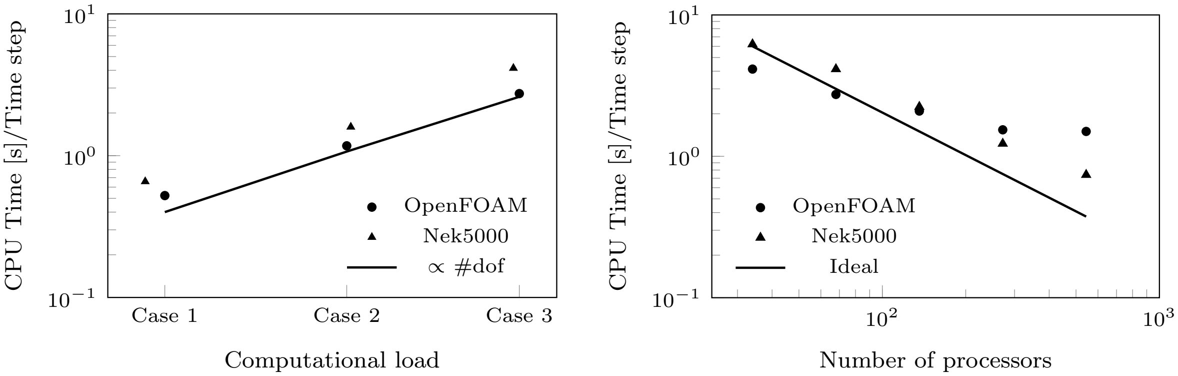

**Figure 6**: Computational performances of OpenFOAM and Nek5000. (left) Elapsed CPU time per time step for the three cases of Table 1 on 68 processors; (right) elapsed CPU time per time step for the cases OF3 and N3 as a function of the number of processors.

option, which attempts to minimize the number of processor boundaries.

Importantly, Nek5000 required an average number of time steps per cycle, on equal maximum CFL number, about 2.5 times higher than OpenFOAM. This difference is mainly due to the non-uniform distribution of the GLL points within the elements, that requires smaller time steps to satisfy the linear stability condition. By taking into account the parallel scaling results, Nek5000 was slower than OpenFOAM by a factor ranging from 1.25 to 4, depending on the number of processors employed.

For the sake of determining the global efficiency of the two codes, accuracy considerations have to be taken into account. The results reported in the previous sections have shown that, while both codes are fairly robust in predicting macroscopic quantities, higher-order statistics converge much slower for the OpenFOAM computations and only the results at the finest grid level could be considered satisfactory. On the other hand, even the coarsest grid provided reliable flow fields for Nek5000. Therefore, OpenFOAM requires roughly 8 times the number of total gridpoints to achieve similar accuracy, a result in line with a previous comparison carried out in a canonical channel flow [23]. It can thus be concluded that Nek5000 was globally more efficient than OpenFOAM.

## 4 CONCLUSIONS

This work reported a comparison between direct numerical simulations of a synthetic jet issuing in a quiescent external environment performed by means of two open-source, general-purpose CFD codes, Nek5000 and OpenFOAM. The two codes are built upon radically different numerical algorithms, and the scientific community is becoming increasingly interested in exploring and comparing their efficiency for the solution of problems of engineering interest [24].

Numerical simulations were conducted for three grids of increasing resolutions, targeting the parameters of a well-documented experimental campaign [12]. As far as macroscopic quantities of engineering interest (e.g., stroke length) are concerned, the two codes proved to be fairly robust at all the tested resolutions. On the other hand, a closer inspection of the crosswise velocity profiles in the near field revealed an anomalous behaviour of the lower-order solutions, especially for the ingestion phase. Specifically, the OpenFOAM runs (even for the intermediate grid) reported largely over-predicted residual turbulent fluctuations, and therefore failed to capture the partial re-laminarization of the near-field flow occurring within the suction phase. At the finest grid resolution, the code-to-code agreement was fairly good and the comparison with experimental data was satisfactory, within the limits due to inherent modeling assumptions (previously discussed in [13]) and experimental uncertainty.

In terms of computational efficiency, the present results reported that, on equal number of total degrees of freedom, Nek5000 is generally slower than OpenFOAM by a factor ranging from 1.5 to 4, mainly depending on the parallel speed-up performances achieved. In contrast, OpenFOAM was found to require roughly 8 times the number of gridpoints to compare with the accuracy provided by Nek5000. It can thus be concluded that Nek5000 is potentially more efficient than OpenFOAM for the investigated problem.

## ACKNOWLEDGMENTS

The authors kindly acknowledge Philipp Schlatter for providing support with Nek5000. Part of this work was supported by a grant of HPC time from CINECA under the ISCRA project DENSITY.

## REFERENCES

[1] Glezer, A. and Amitay, M. Synthetic jets. *Annu. Rev. Fluid Mech.* (2002) **34**:503–529.

[2] Pavlova, A. and Amitay, M. Electronic cooling using synthetic jet impingement. *J. Heat Transfer* (2006) **128**:897–907.

[3] Smith, B.L. and Glezer, A. Jet vectoring using synthetic jets. *J. Fluid Mech.* (2002) **458**:1–34.

[4] Chiatto, M., Marchitto, L., Valentino, G. and de Luca, L. Influence of piezo-driven synthetic jet on water spray behavior. *Atomization Sprays* (2017) **27**:691–706.

[5] Mohseni, K. and Mittal, R. *Synthetic Jets: Fundamentals and Applications.* CRC Press, 2014.

[6] Smith, B.L. and Glezer, A. The formation and evolution of synthetic jets. *Phys. Fluids* (1998) **10**:2281–2297.

[7] Smith, B.L. and Swift, G.W. A comparison between synthetic jets and continuous jets. *Exp. Fluids* (2003) **34**:467–472.

[8] Van Buren, T., Whalen, E. and Amitay, M. Vortex formation of a finite-span synthetic jet: effect of rectangular orifice geometry. *J. Fluid Mech.* (2014) **745**:180–207.

[9] Kral, L.D., Donovan, J.F., Cain, A.B. and Cary, A.W. Numerical simulation of synthetic jet actuators. In *4th Shear Flow Control Conference* (1997) 1824.

[10] Rizzetta, D.P., Visbal, M.R. and Stanek, M.J. Numerical investigation of synthetic-jet flow fields. *AIAA J.* (1999) **37**:919–927.

[11] Rumsey, C.L., Gatski, T.B., Sellers, W.L., Vatsa, V.N. and Viken, S.A. Summary of the 2004 computational fluid dynamics validation workshop on synthetic jets. (2006) **44**:194–207.

[12] Yao, C.S., Chen, F.J. and Neuhart, D. Synthetic jet flowfield database for computational fluid dynamics validation. *AIAA J.* (2006) **44**:3153–3157.

[13] Kotapati, R.B., Mittal, R. and Cattafesta III, L.N. Numerical study of a transitional synthetic jet in quiescent external flow. *J. Fluid Mech.* (2007) **581**:287–321.

[14] de Luca, L., Girfoglio, M., Chiatto, M., Coppola, G. Scaling properties of resonant cavities driven by piezo-electric actuators. *Sensors and Actuators A: Physical.* (2016) **247**:465–474.

[15] Utturkar, Y. and Mittal, R. Sensitivity of synthetic jets to the design of the jet cavity. *AIAA paper* (2002) 2002-0124.

[16] Chiatto, M., Capuano, F., Coppola, G. and de Luca, L. LEM characterization of synthetic jet actuators driven by piezoelectric element: A review. *Sensors* (2017) **17**:1216.

[17] Patera, A.T. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.* (1984) **54**:468–488.

[18] `https://openfoam.org`

[19] `http://nek5000.mcs.anl.gov`

[20] Varghese, S.S., Frankel, S.H. and Fischer, P.F. Direct numerical simulation of stenotic flows. Part 2. Pulsatile flow. *J. Fluid Mech.* (2007) **582**:281–318.

[21] Dong, S., Karniadakis, G.E. and Chryssostomidis, C. A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *J. Comput. Phys.* (2014) **261**:83–105.

[22] `http://www.hpc.cineca.it/hardware/marconi`

[23] Sprague, M.A. A comparison of Nek5000 and OpenFOAM for the DNS of Turbulent Channel Flow. Presentation at Nek5000 Users Meeting (2010) Argonne National Laboratory.

[24] Kooij, G.L., Botchev, M.A., Frederix, E.M.A., Geurts, B.J., Horn, S., Lohse, D., van der Poel, E.P., Shishkina, O., Stevens, R.J. and Verzicco, R. Comparison of computational codes for direct numerical simulations of turbulent Rayleigh–Bénard convection. *Comput. Fluids* (2018) **166**:1–8.