In view of accelerating CFD simulations through coupling with vortex particle approximations

Giorgos Papadakis, Spyros G. Voutsinas
National Technical University of Athens, 9, Heroon Polytechniou Str, 15780, Athens, Greece
E-mail: papis@fluid.mech.ntua.gr, spyros@fluid.mech.ntua.gr

Abstract. In order to exploit the capabilities of Computational Fluid Dynamics in aerodynamic design, the cost should be reduced without compromising accuracy and consistency. In this direction a hybrid methodology is formulated within the context of domain decomposition. The strategy is to choose in each sub-domain the best performing method. Close to solid boundaries a grid-based Eulerian flow solver is used while in the far field the flow is described in Lagrangian coordinates using particle approximations. Aiming at consistently including compressible effects, particles carry mass, dilatation, vorticity and energy and the complete set of conservation laws is solved in Lagrangian coordinates. At software level, the URANS solver MaPFlow is coupled to the vortex code GENUVP. In the present paper the two dimensional formulation is given alongside with validation tests around airfoils in steady and inherently unsteady conditions. It is verified that: purely Eulerian and hybrid simulations are equivalent; the Eulerian domain in the hybrid solver can be effectively restricted to a layer 1.5 chord lengths wide; significant cost reduction reaching up to 1:3 ratio is achieved.

1. Introduction

Motivation & context: Computational Fluid Dynamics (CFD) is a well-established and highly valued tool. Following the amazing progress in computer technology, large scale simulations of the flow around complete wind turbine configurations are now feasible. Key drivers to that, are the ability of CFD to model the flow mechanisms in detail and to accurately predict the flow development in space and time [1]. In principle both are achievable but the computational cost remains excessively high for conducting productive design simulations. This explains why CFD is always in search of methods that reduce cost without compromising accuracy. In this respect, domain decomposition is a widely used technique [2] which has significantly reduced the user time of CFD simulations. In standard domain decomposition, the same formulation is used in all sub-domains. However, in certain methodologies, named hybrid, different formulations are used. The strategy in this case is to choose for every sub-domain the best performing methodology.

Eulerian CFD is by far the most popular simulation method in aerodynamics. Wall boundary conditions are accurately introduced while the flow equations can be solved with high accuracy, provided that the grid is sufficiently dense. However, when applied to external flows, the flow field must be truncated at some finite distance. Over the distant boundary, far-field conditions are needed which in most cases approximate a fully developed flow. Depending on the approximation chosen, far-field conditions could introduce errors that are sensed as reflections in the simulations [3]. Furthermore, domain truncation is usually combined with gradual grid...
coarsening which increases numerical diffusion and adds errors. Finally, when considering flows around independently moving bodies, sliding or over set grids are needed in order to properly handle the involved interactions. Typical examples in this respect are the blade-tower interaction problem for Horizontal Axis Wind Turbines and the flow around Vertical Axis Wind Turbines.

Alternatively, one may solve the flow equations in Lagrangian formulation as particle methods do. They are grid-free and self-adaptive, having (in theory) zero numerical diffusion; domain truncation is not needed and true velocity conditions at infinity are exactly embedded in the formulation [4]. However, in particle methods wall boundary conditions constitute a major challenge, involving costly convolution operations and large number of particles [5, 6]. Therefore, since particle methods appear so complementary to Eulerian CFD, it is natural to combine the two in one single package following a domain decomposition approach.

The option of combining domain decomposition with essentially different formulations has taken several forms. From a topological point of view, the sub-domains can be either overlapping or not. Strong viscous-inviscid interaction models [7, 8] and RANS-Vortex coupled ones [9, 10] are examples of overlapping hybrid methodologies. They combine an inviscid Lagrangian formulation for the entire flow with a viscous layer close to the solid boundaries and define coupling conditions for the pressure and the velocity on the outer boundary of the viscous layer. The coupling in this case is approximate, not only because the two formulations are not equivalent, but also because the sub-domains overlap. Exact coupling consists of imposing continuity of all flow quantities which in principle refers to non-overlapping sub-domains. In this case, the coupling conditions take the form of integral equations (also known as Neumann-to-Dirichlet map) defined on the boundary interface [4, 11, 12]. In 2D problems the extra cost is manageable but in 3D the penalty can be substantial. This explains why certain hybrid methods prefer to have at least some degree of overlapping, in the form of a buffer area as in [13], while others directly approximate the coupling conditions. In [14, 15] compressible RANS solvers were coupled with viscous vortex methods. They both partially implement non-overlapping decomposition. Vortex particles are defined outside the Eulerian domain based on the flux of vorticity across the interface while the calculation of the velocity on the interface includes the vorticity within the Eulerian domain. Another particularity, is that density and pressure are not explicitly matched at the boundary.

**Present contribution:** The present work formulates a hybrid method for two dimensional external flows by combining a compressible Eulerian solver with a particle based Lagrangian one. This research was originally motivated from helicopter applications which explains the choice of solving the compressible flow equations. For the Eulerian part MaPFlow is used; a multi-block, MPI enabled, unstructured finite volume compressible URANS solver with preconditioning for low Mach numbers, Ma, [16]. For the Lagrangian part, a modified version of GENUVP is used, which is also MPI enabled [17]. Originally GENUVP was developed for incompressible flows and particles were carrying mass and vorticity. In order to comply with the compressible context of MaPFlow, particles are also carrying dilatation and energy. This extension was first proposed by Eldregde [18], who however did not include solid boundaries. Relevant to the present work is also the formulation by Oxley [19] who included solid boundaries but assumed isentropic flow and therefore particles did not carry energy. Based on the above, the essence of the present formulation is to define a consistent coupling between Eulerian and Lagrangian CFD solvers. A variety of methods and tools are brought together in one single package, mainly targeting cost reduction while maintaining accuracy.

**Cost and available reduction options:** Parallel programming has certainly revolutionized the perception on cost which is now measured as the elapsed user time. All CFD methods can be parallelized by means of domain decomposition [2]. In Eulerian methods, the grid is split into blocks while in Lagrangian methods, particles are split into sub-groups. In many applications
this is not sufficient and further reduction is needed. In this respect, multi-grid convergence schemes [20] aim at accelerating Eulerian solvers while Multi-pole expansions [21] and Particle-Mesh (PM) methods [22] try to achieve the same with particle methods. In the present work, the PM method is used.

2. The HoPFlow hybrid approach

2.1. The Eulerian formulation of the flow equations

In external aerodynamics, Eulerian CFD will solve the flow equations in $U_E = \{\rho, \rho \vec{u}, \rho E\}^T$ formulation over a bounded domain $D_E$ of the two dimensional space $\mathbb{R}^2$. In integral form,

$$\frac{\partial}{\partial t} \int_{D_E} U_E \, dD + \oint_{\partial D_E} \left( \vec{F}_c(U_E) - \vec{F}_v(U_E) \right) \, ds = \int_{D_E} Q_E \, dD$$

(1)

where $F_{c,i} = \{ \rho u_n, pu_i + n_i \rho, \rho H u_n \}^T$ and $F_{v,i} = \{ 0, n_j \tau_{ij}, n_j u_i \}^T$ are the convective and viscous fluxes respectively, $\rho$ is the fluid density, $\vec{u}$ is the flow velocity, $p$ is the pressure, $E$ is the total internal energy, $H$ is the total enthalpy, $\tau$ denotes the viscous stress tensor, $u_n = \vec{u} \cdot \vec{n}$ and $Q_E$ is the source term. The above system is supplemented with the equations of the turbulence model. The $k-\omega$ SST [23] and the Spalart-Allmaras [24] turbulence models are used herein.

$D_E$ is internally bounded by the solid surface $S_B$ and externally by $S_E$. On $S_B$ wall boundary conditions are applied while $S_E$ is defined as the matching boundary with the Lagrangian solver. It is important to note that $S_E$ is not necessarily identified as an interface or as the surface shedding particles into $\mathbb{R}^2 - D_E$. In the present formulation the Lagrangian flow equations are solved over the entire $\mathbb{R}^2$ and continuity of all flow quantities is imposed across $S_E$, in a similar fashion to an immersed boundary approach [25]. In the present formulation, the boundary conditions on $S_E$ are defined based on $1-$D Riemann invariants and therefore the necessary input to the Euler solver is the flow state on the outer side of $S_E$ [26].

2.2. Lagrangian formulation of the two dimensional flow equations

The flow properties are assigned to a collection of fluid particles indexed by subscript “p”. Each particle is centered at $\vec{Z}_p$, occupies a volume $V_p$ and carries volume integrals of density $m_p$ (mass), vorticity $\Omega_p$, dilatation $\Theta_p$ and energy $\Pi_p$:

$$q_p(\vec{x},t) = \sum_p Q_p(t) \zeta(\vec{x} - \vec{Z}_p(t)), \quad Q_p(t) = \int_{V_p(t)} q(\vec{x},t) \, dV = q_p(t) \, V_p(t)$$

(2)

where $q_p$ is any of $(\rho_p, \omega_p, \theta_p, (\rho E)_p)$ and respectively $Q_p$ is any of $(m_p, \Omega_p, \Theta_p, \Pi_p)$ (e.g. $\Omega_p = \omega_p V_p$). Also $\zeta$ denotes a smooth approximation of the Dirac function [27].

In Lagrangian description, the 2D inviscid flow equations take the form:

$$D_t U_p = F_p(U_p, t), \quad D_t (\cdot) = \partial_t + (\vec{u} \cdot \nabla)(\cdot), \quad U_p = \{ \vec{Z}_p, m_p, V_p, \Theta_p, \Omega_p, \Pi_p \}^T$$

$$F_p = \left\{ \vec{u}_p, 0, \Theta_p, V_p \left( \nabla \rho \times \nabla \rho \right)_p, V_p \left( 2 ||\nabla \vec{u}|| - \nabla \left( \frac{\nabla \rho}{\rho} \right)_p \right), -V_p \left( \nabla \left( \vec{u} \cdot \vec{n} \right) \right)_p \right\}$$

(3)

where $(\cdot)_p$ denotes evaluation at the particle position $\vec{Z}_p$. Extension to viscous flows involves extra terms in $\Theta, \Omega, \Pi$ equations. For the first two, they derive from the momentum equation by taking the div and rot of $\nabla \sigma / \rho$ while for the energy equation the extra term is $\nabla \left( \vec{u} \cdot \sigma \right) / \rho$. In these terms $\sigma$ denotes the viscous stress tensor.

The above system is supplemented with the Helmholtz velocity decomposition:

$$\vec{u} = \vec{U}_\infty + \nabla \phi + \vec{w}, \quad \vec{w} = \nabla \times \vec{K}, \quad \nabla^2 \phi = \nabla \cdot \vec{u} = \theta, \quad \nabla^2 \psi = -\omega$$

(4)
where \( \vec{k} \) denotes the normal to \( D \) unit vector. Using Green’s theorem, integral representations for \( \phi \) and \( \psi \) are obtained. Let \( \vec{n}, \vec{\tau} \) denote the normal and tangent unit vectors to \( S \), and \( \partial_n(\cdot) = \vec{n} \cdot \nabla, \partial_\tau(\cdot) = \vec{\tau} \cdot \nabla \) the corresponding directional derivatives. Then:

\[
\begin{align*}
\phi(\vec{x}) &= \int_D \nabla^2 \phi(\vec{y}) G(\vec{r}) \, dD(\vec{y}) + \int_S (\partial_n \phi(\vec{y}) G(\vec{r}) - \phi(\vec{y}) \partial_n G(\vec{r})) \, dS(\vec{y}) \\
\psi(\vec{x}) &= \int_D \nabla^2 \psi(\vec{y}) G(\vec{r}) \, dD(\vec{y}) + \int_S (\partial_n \psi(\vec{y}) G(\vec{r}) - \psi(\vec{y}) \partial_n G(\vec{r})) \, dS(\vec{y})
\end{align*}
\]  

(5)

where \( \vec{r} = \vec{x} - \vec{y} \) and \( G = (2\pi)^{-1}\ln(r) \) is the 2D Green’s function for the Laplace operator. Taking the \( \nabla \) of the first and the \( \nabla \times \) of the second, the integral representation of \( \vec{u} \) is obtained:

\[
\begin{align*}
\vec{u}(\vec{x}) &= \vec{U}_\infty + \int_D (\theta(\vec{y}) \vec{K}(\vec{r}) + \omega(\vec{y}) \vec{K} \times \vec{K}(\vec{r})) \, dD(\vec{y}) + \\
&\quad \int_S (u_n(\vec{y}) \vec{K}(\vec{r}) + u_\tau(\vec{y}) \vec{K} \times \vec{K}(\vec{r})) \, dS(\vec{y})
\end{align*}
\]  

(6)

In the above expression, \( u_n, u_\tau \) denote the normal and tangential disturbance velocity components on \( S \) and \( \vec{K}(\vec{r}) = \nabla G(\vec{r}) \). The volume term in (6) is directly obtained from the volume integrals in (5). Also the surface terms that are associated to \( \partial_n \) in (5) directly contribute their shares in \( u_n, u_\tau \) respectively. Finally the terms involving \( \partial_\tau \) for \( \phi, \psi \) are obtained by applying Stokes’ theorem to the second surface terms in (5) [28].

If there are no boundaries, the way to proceed is rather straightforward. For given initial conditions, the transport equations given in (3) are integrated in time and the particles are transported using the velocity field defined in (6). Cost is dominated by the convolution integral in (6). For \( N \) particles, the associated cost is proportional to \( N^2 \), which can easily explode as \( N \) becomes large and the intended duration of the simulation is long. The situation becomes even worse, if boundaries are present and the intended duration of the simulation is long. The situation becomes difficult when transforming a surface quantity \( \theta dD, u_n dS \) as well as \( \omega dD, u_\tau dS \) are associated to the same kernel and hence they can be treated as particles without distinction. However, it is important to respect conservation and therefore when transforming a surface quantity \( q_s \) into a volume quantity \( Q_p \), the following should be used: \( Q_p = q_s \Delta S \), where \( \Delta S \) denotes the support of \( q_s \). Then, as regards accuracy, the fact that the Eulerian solver is used close to the boundaries considerably reduces our requirements. In fact the quick decay of \( \vec{K}(r) \) for increasing \( r \) assures that the error will also decay quickly.
2.3. Solution strategy using particles

For $\psi$ and similarly for $\phi$, the PM solver takes the following form (Fig 1):

1. Project $\omega$ on the grid: $\omega_{i,j} \equiv \text{Proj}_{PM}(\omega_p; V_p) = \frac{\sum_p \omega_p V_p W(\bar{x}_{i,j} - \bar{Z}_p)}{\sum_p V_p W(\bar{x}_{i,j} - \bar{Z}_p)}$

2. Calculate $\psi_S$ as Dirichlet boundary conditions on $S_{PM}$.

3. Solve $\nabla^2 \psi_{i,j} = -\omega_{i,j}$ and obtain: $\psi_{i,j}, \bar{w}_{i,j} = \nabla \times \psi_{i,j} \bar{k}$

4. Interpolate $\bar{w}_{i,j}$ at the particle positions: $\bar{w}_p \equiv \text{Interp}(\bar{w}_{i,j}) = \sum_{i,j} \bar{w}_{i,j} W(\bar{x}_{i,j} - \bar{Z}_p)$

where $W(r) = W_1(x/h)W_1(y/h)$, $W_1$ is an 1D interpolation function and $h$ denotes the grid spacing. The $M_1'$ interpolation function was chosen for $W_1$ which conserves the moments up to $3^{rd}$ order (for other options see [4]). The calculation of the Dirichlet boundary conditions (step (2)) is the most time consuming step which becomes critical in 3D simulations [30]. Earlier cost reduction techniques involved merging distant particles into groups as in [31] or using tree-algorithms. Out of the various options that were tested, the best results were obtained with the James Lackner algorithm [32]. Based on the above PM solver, the following particle algorithm has been formulated:

For a given collection of particles: $\{\bar{Z}_p^n, m_p^n, V_p^n, \Omega_p^n, \Theta_p^n, \Pi_p^n\}$ at $t = n\Delta t$:

**Step 1:** Project $\{m_p^n, \Omega_p^n, \Omega_p^n, \Pi_p^n\}$ on the PM grid and get: $\rho^n_{i,j}, \theta^n_{i,j}, \omega^n_{i,j}, E^n_{i,j}$

**Step 2:** Solve $\nabla^2 \phi = \theta, \nabla^2 \psi = -\omega$ and obtain: $\phi^n_{i,j}, \psi^n_{i,j}, \bar{w}^n_{i,j}, \bar{w}^n_{i,j}$

**Step 3:** Calculate on the PM grid the terms in the RHS of (3), e.g. $\nabla \rho_{i,j}, \nabla p_{i,j}, \nabla \bar{w}_{i,j}$

**Step 4:** Interpolate all grid based data $q^n_{i,j}$ at the particle positions:

$$q^n_p = \sum_{i,j} q^n_{i,j} W(\bar{x}_{i,j} - \bar{Z}_p)$$

**Step 5:** Update all particle properties (integrate (3) in time)

In particle methods, re-meshing is added after Step 5. Once the particles are placed in their updated positions, re-meshing consists of projecting the information they carry at pre-selected positions of the PM grid. In the present work, the nodes of the PM grid have been selected. The projection operator assumes that there is a complete coverage of particles in $D_{PM}$. In order to prevent loss of smoothness in the particle approximation, re-meshing [4] is applied every $N$ time-steps, where $N$ depends on the free-stream velocity and the size of the PM grid.

Another point concerns the extent of the PM grid. As indicated in Fig 1 (Left), $\mathbb{R}^2$ is divided in two parts: $D_{PM}$ which contains $D_E$, and $D_p$ which extends to infinity. In the present solver $D_p$ acts like a far-field in which particles are passively convected with the free-stream velocity, without changing $\{m_p, V_p, \Omega_p, \Theta_p, \Pi_p\}$ any longer. However, their induced effect on $D_{PM}$ is taken into account by solving the $\psi, \phi$ equations on a coarse PM grid.

2.4. Coupling procedures

The Eulerian and PM solutions are coupled in two ways: a) the Eulerian flow information $U_E$ corrects the PM solution in $D_E$, and b) the PM solution defines the boundary conditions on $S_E$. The key idea of the coupling is to ensure that at the end of each time step, the PM solution smoothly extends the Eulerian one outside $D_E$. To this end, a procedure is formulated that replaces the PM particles with those defined by the Eulerian solution in the course of the PM-CFD iterations. In this sense, particles are not emitted nor absorbed at the boundaries but the flow information is corrected throughout $D_E$. An important feature of the coupling procedure is that in every PM-CFD iteration, the boundary conditions set of $S_E$ are using the latest update of the flow information which means that a strong and conservative coupling is defined.
Let \( \hat{q}_{PM} \) denote the PM solution obtained at the beginning of each time step. In \( \hat{q}_{PM} \) there are two errors: the projection error and the solution error. The first is related to the transformation of the discrete particle information into a continuous space distribution, while the second is associated to the fact that the PM solver does not take into account the presence of \( S_E \). Depending on the PM resolution and the size of the time step, particles could even cross \( S_E \). Correction of the solution error is based on the Eulerian solution. Since the positions \( P_E \) at which the Eulerian solver will define particles are fixed relative to the Eulerian grid, it is possible to determine the amount of \( q \) we should subtract from \( \hat{q}_{PM} \). First \( \hat{q}_{PM} \) is interpolated at \( P_E \) and then projected back to the PM grid. In this way the error in the PM solution and the correction of \( q_{PM} \) are obtained (Fig 1, Right):

\[
\text{error}\{q_{PM}\} = \text{Proj}_{PM}\{\text{Interp}_{PE}(\hat{q}_{PM})\}
\]

\[
\text{correct}\{q_{PM}\} = \hat{q}_{PM} - \text{error}\{q_{PM}\} + q_E
\]

The contribution of the Eulerian solution, \( q_E \), is obtained by defining \((\rho, \omega, \theta, \rho E)\) at the \( P_E \) positions and then projecting the information to the PM grid. Four particles per Eulerian cell are created. Their positions \( P_E \) and their associated volumes are defined using iso-parametric finite element approximations. The required \( \theta, \omega \) are obtained by applying Green-Gauss evaluations. The above correction conserves the moments up to 3\(^{rd}\) order for all quantities and is applied to all particles within \( S_E \). Based on \( \text{correct}\{q_{PM}\} \), the PM flow information is interpolated at the centers of the boundary faces on \( S_E \) and then used as outer state for the Riemann invariants. Having set the boundary conditions, the solution at the current state is obtained using the Eulerian solutions at the two previous steps (the time scheme of the Eulerian solver was 2\(^{nd}\) order).

Finally, as regards viscosity, for most of the flow a constant value would be sufficient, which largely simplifies the Lagrangian part of the solver. This is valid outside of the boundary layer where the eddy viscosity is small, and so the physical viscosity can be used. Within \( D_E \), the Eulerian solver includes turbulence modeling and therefore the variation of eddy viscosity within the boundary layer is taken into account. By properly choosing the extent of \( D_E \), on \( S_E \) the physical viscosity can be used as boundary condition. Coming to the Lagrangian part, in the present formulation the assumption of constant viscosity is extended over the entire \( D_{PM} \). Using wrong viscosity within \( D_E \) and in particular close to \( S_E \) when solving (3), is not critical because at each step the particles within \( D_E \) are renewed and so errors do not propagate. It is noted that diffusion in the Lagrangian solver is only carried out at the end of every time step and that viscosity is not transferred to the Eulerian solver. Ultimately, particle quantities are used in order to define the outer state over the boundary of the Eulerian domain and it is there, that high accuracy is essential.
3. Verification tests

The following test cases are considered. The NACA0012 and the flat back FB35001750 airfoils were included in the validation of MaPFlow [16] while the FFAW3241 is one of the thick airfoils considered in [33]. The aim is to check the HoPFlow in comparison to MaPFlow in terms of consistency, accuracy and performance. In order to minimize any grid density issues, HoPFlow used an exact subset of the MaPFlow grid. Since both models are using the same modeling close to the airfoil, improvement in accuracy in this region is not expected for the same grid density. Definitions and basic numerical parameters are given in Table 1 along with the time averaged lift and drag data. In all simulations the flow was assumed fully turbulent. The Eulerian grid in HoPFlow extended up to 1.5c, while the MaPFlow grid extended up to 50c, where c denotes the airfoil chord. The HoPFlow grid had \( \approx 130000 \) cells while that of MaPFlow had \( \approx 220000 \) in all cases.

For the NACA0012, up to an angle of attack, \( \alpha = 15^\circ \), the differences in Cl, Cd, are less than 1%. This is further supported in Fig 2 showing the distribution of the (Hop-MaP)Flow difference in pressure at \( \alpha = 15^\circ \). However at 20\(^\circ\), stall in HoPFlow is stronger (lower lift and higher drag Table 1) which is also seen in the comparison of pressure distributions shown in Fig 3. Furthermore, at this angle, a von Karman street wake is generated by HoPFlow (Fig 4) that renders the flow unsteady contrary to MaPFlow (Fig 5).

In the FFAW3241 case, a high Reynolds number \( Re = 12 \cdot 10^6 \) is considered. Both solvers predict inherent unsteadiness in the flow and agree well in terms of mean Cl, Cd. At \( \alpha = 24^\circ \) (and similarly at 20\(^\circ\), 28\(^\circ\)) the Cl time histories (Fig 6) correspond to the same Strouhal number while a small difference in amplitude is found. However at \( \alpha = 32^\circ \) there is difference in the Cl response pattern (Fig 7). The basic frequency in the MapFlow signal corresponds to a Strouhal number of 0.38, while in that of HoPFlow half of that frequency is excited also. By comparing the flow fields of the two methods (Fig 8, Fig 9), different wake dynamics are seen which could explain the difference in the response. Vortices in the HoPFlow flow are not diffused as in the MaPFlow results while their mutual interaction changes their spacing.

Finally in the flat back airfoil case, the flow is expected to develop inherent unsteadiness due to strong vortex shedding at the trailing edge corner points. Now the Re number is much smaller (\( Re = 0.66 \cdot 10^6 \)) compared to the previous case and wake dynamics are expected to be different (Fig 12, Fig 13). Because of low diffusion in HoPFlow, the differences in Cl and Cd (Table 1), are now more pronounced. Also the time response is different as indicated in Fig 10 at \( \alpha = 5^\circ \) and in Fig 11 at \( \alpha = 10^\circ \).

### Table 1. Definition of test cases & Lift, Drag predictions (Difference=HoPFlow-MaPFlow)

| Case          | NACA0012 | FFAW3241 | FB35001750 |
|---------------|----------|----------|------------|
| Conditions    | Re=6 \cdot 10^6, Ma = 0.15 | Re=12 \cdot 10^6, Ma = 0.26 | Re=0.66 \cdot 10^6, Ma = 0.15 |
| PM details    | \( \Delta t = 0.004, \Delta x_{PM} = 0.02 \) | \( \Delta t = 0.004, \Delta x_{PM} = 0.02 \) | \( \Delta t = 0.002, \Delta x_{PM} = 0.02 \) |
| CFD grid      | \( y^+ \approx 0.5, S_B : 650 \) nodes | \( y^+ \approx 1, S_B : 650 \) nodes | \( y^+ \approx 0.7, S_B : 740 \) nodes |
| \( \alpha \)  | 5° 10° 15° 20° | 20° 24° 28° 32° | 0° 5° 10° 15° |
| HoPFlow       |          |          |            |
| Cl            | 0.553    | 1.07     | 1.52       | 0.86       | 1.73    | 1.87     | 1.67 | 1.61 | 0.37 | 0.97 | 1.56 | 2.25 |
| Cd            | 0.009    | 0.015    | 0.026      | 0.27       | 0.10    | 0.24     | 0.40 | 0.70 | 0.21 | 0.21 | 0.21 | 0.17 |
| Difference    |          |          |            |
| \( \Delta Cl \times 100 \) | -0.003  | -0.45    | -0.26      | -0.27      | 0.27   | 0.47     | 3.35 | 5.38 | -4.56 | -3.22 | -6.23 | -0.28 |
| \( \Delta Cd \times 100 \) | -0.04   | 0.08     | -0.01      | 5.15       | -0.75  | -1.50    | 0.39 | 2.96 | -0.06 | 0.33 | 0.20 | 2.11 |
4. Discussion and conclusions

A consistent way of coupling Eulerian CFD with particle methods for external flows has been formulated and its equivalence to full Eulerian CFD has been indicated in three verification tests. Key elements of the whole approach are: the adaptation of the immersed boundary concept in the PM part that allows using fast Poisson solvers; the introduction of compressible vortex particles that allows defining a consistent coupling with compressible Eulerian CFD solvers; the definition of the PM-CFD error field that allows smooth passage in between the two solvers.

In purely numerical terms, there are 5 important parameters:

1. The extent of $D_E$ and $D_{PM}$. A series of preliminary tests was carried out in which the extent of the Eulerian domain $D_E$ varied. It was found that in the range of 1-1.5c, HoPFlow and MaPFlow converged to the same solution. Based on that, a more or less uniform extent 1.5c wide, was finally selected. This choice was made in order to include in $D_E$ the “difficult” parts of the flow, such as the separation bubble. Using non-uniform or smaller Eulerian domains is certainly of interest in view of further reducing the cost and increasing the near wall resolution. This would require a systematic investigation, which is pending.
With regard to $D_{PM}$, its extent depends on the nature of the problem and the requirements of the Poisson solver. In the present simulations $D_{PM}$ extended up to 7c and the PM grid spacing was kept constant. The latter is mainly due to requirement of uniform grid set by the multi-block Poisson solver we have used [32]. The option of using non-uniform grids although possible and eventually advantageous with respect to cost, was not considered at this stage as it would substantially complicate the projection and interpolation operations. Concerning the extent of $D_{PM}$, it is noted that the computational domain does not end there; particles are kept after they cross the $D_{PM}$ boundary. Although in $D_{P}$ the evolution of particles reduces to simple convection, the effect of flow history is still taken into account on $D_{PM}$. For example in steady conditions the starting vortex will be always present in $D_{P}$.

(2) The PM resolution. The PM resolution depends on the grid spacing $\Delta x_{PM}$ and the number of particles. In particle methods, the overlapping of particles is an important error condition that also applies to PM methods. In order to achieve sufficient overlapping the difference between $\Delta x_{PM}$ and the mean transport distance $\Delta l$ per time step should be positive, which is equivalent to the CFL condition in CFD. By choosing to create 4 particles per Eulerian
Figure 10. FB35001750: Cl response at $\alpha = 5^o$ (Time is dimensionless).

Figure 11. FB35001750: Cl response at $\alpha = 10^o$ (Time is dimensionless).

Figure 12. FB35001750: Vorticity contours at $\alpha = 5^o$ (MaPFlow)

Figure 13. FB35001750: Vorticity contours at $\alpha = 5^o$ (HoPFlow)

cell, $\Delta x_{PM}$ approximately matched the size of the Eulerian grid in the vicinity of $S_E$ and so in this region the above requirement was satisfied. Tests with up to 64 particles per cell did not show any appreciable difference indicating that the number of particles per cell is not as important as the size of $\Delta x_{PM}$. Loss of resolution in particle methods is also related to distortions that gradually build up as the flow evolves in time. They manifest as isolated “holes” (cells without any particle), which violate the overlapping condition and therefore re-meshing is necessary. In the present formulation there are particles everywhere. Non-zero Volume, mass and energy is defined for all particles. So the issue concerning “holes” formation refers to vorticity and dilatation only. It is noted that by weighting the projection with the volume of the particles, re-meshing can be automatically applied to all particles.

(3) Time integration. In the present simulations, time integration in Lagrangian equations was performed using a 1st order scheme in order to keep the implementation simple. Higher order time schemes can be applied but because of re-meshing the information from preceding steps must be carefully recorded since re-meshing is redistributing particle quantities.

(4) Viscous effects. In particle methods diffusion has been successfully treated by the Particle Strength Exchange (PSE) method [34]. In PM methods the implementation of PSE is
straightforward as long as viscosity remains constant. However in turbulent flows the situation is different. Fortunately, far from solid boundaries \( \nu_T \) can be considered constant and equal to the physical viscosity. Viscosity will also vary across regions of high shear which in flows with massive separation corresponds to the shed vortices. In this case Lagrangian modeling could be adapted by solving a transport equation for \( \nu_T \) and adding the extra terms in (3) that account for space variations of viscosity.

(5) Cost reduction. The cost of HoPFlow depends on: a) the size and extent of the Eulerian grid covering \( D_E \), b) the resolution of the PM grid and c) the convergence rate of the method.

The usual grid generation rules apply to the Eulerian grid: \( y^+ < 1 \) and the cell size should gradually increase as we move away from solid boundaries. For high Re flows (\( Re \gg 10^6 \)), the majority of the cells is placed close to the walls. In the cases considered, the 1.5c extent of \( D_E \) meant that 60% the MaPFlow grid was within \( D_E \). Although further reduction of the \( D_E \) extent would lead to additional saving, at least in 2D flows, the change in number of cells will not be that important. The saving in this respect concerns only the cost per Eulerian iteration, which is proportional to the grid reduction and will partially counter balance the extra cost of the PM solver.

This extra cost depends on \( D_{xPM} \) and the size of \( D_{PM} \). In the present simulations, \( D_{xPM} \) was chosen so that for a given Eulerian grid, the solution remained unchanged. Based on tests the value of 0.02 was finally set leading to a grid 350x350 PM grid. In this case the combined cost per iteration, HoPFlow was found 10% slower than MaPFlow. Although the parallelization of the PM part in HoPFlow has not been yet optimized, it is not expected that, at least in 2D flows, the benefit will be important.

On the contrary, drastic cost reduction were found with respect to convergence rate. In Eulerian solvers, errors are generated at solid boundaries and subsequently propagate towards the far field boundary of the computational domain. By reducing the extent of the Eulerian grid, errors will propagate much faster and therefore accelerate convergence. The amount of reduction was found to be case dependent.

In steady conditions, HoPFlow converged after 30% of the steps needed by MaPFlow, giving an 1:3 reduction rate. In the unsteady cases, the reduction rate ranged from 1:2 up to 1:3 depending on the wake characteristics. Convergence to a periodic state is achieved once the wake is fully formed. In MaPFlow due to numerical diffusion the starting vortex fades out over a finite distance from the airfoil while in HoPFlow vortex structures remain intact and therefore the starting vortex must cover a longer distance before convergence is reached. A rigorous analysis on convergence is certainly needed before concluding on the exact cost reduction that can be achieved.

The particular tests were selected in order to check the accuracy and the effectiveness of the method. In attached and light stall conditions, HoPFlow is equivalent to MaPFlow while at higher angles of attack HoPFlow predicts the onset of unsteadiness earlier. This is attributed to the lower numerical dissipation in the wake. However further investigation is needed in order to understand the origin and the significance of these fluctuations.

The methodology as described is extendable to 3D flows. With respect to (3) the only additional term needed is the deformation term \((\vec{\Omega} \cdot \nabla)\vec{u}\) in the vorticity equation. Otherwise, the stream function and vorticity become vector fields; the solution of the Poisson equations becomes more complicated; while the zero divergence condition for \( \vec{\omega} \) should be added. The latter requires special attention as noted in [4].

Acknowledgments
This research has been co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of
the National Strategic Reference Framework (NSRF) Research Funding Program: THALES. Investing in knowledge society through the European Social Fund.

References

[1] Jawahar P and Kemath H 2000 J. Computational Physics 164 165–203
[2] Quarteroni A and Valli A 1999 Domain decomposition methods for partial differential equations (Oxford Science Publications)
[3] Poinset T and Lele S 1992 J. Computational Physics 101 104–139
[4] Cottet G H and Koumoutsakos P 2000 Vortex methods: Theory and Practice (Cambridge University Press)
[5] Barba L, Leonard A and Allen C 2005 Int. J. Numerical Methods in Fluids 47 387–421
[6] Koumoutsakos P and Leonard A 1995 Fluid Mechanics 296 1–38
[7] Drela M 1989 Conference on Low Reynolds Number Aerodynamics, University Notre Dame
[8] Riziotis V and Voutsinas S 2008 Int. J. Num. Methods in Fluids 56 185–208
[9] Chaviaropoulos P K, Nikolao I G, Aggelis K A, N N Soerensen and J Johansen M, Hansen O L, Gaunaa M, Hambraus T, von Geyr H F, Hirsch C, Shun K, Voutsinas S G, Tzabiras G, Perivolaris Y and Dyrmose S Z 2003 Wind Energy 6 365–385
[10] Schmitz S and Chattot J J 2006 Computers & Fluids 35 742–745
[11] Guermond J, Huberson S and Shen W 1993 J. Computational Physics 108 343–352
[12] Plouhmans P, Winckelmans G, Salmon J, Leonard A and MSWarren 2002 J. Computational Physics 178 427–436
[13] Cottet G H 1991 Lectures in Applied Mathematics Series, 28 100–118
[14] Anusonti-Inthra, Phuriwat and Floros M 2008 38th AIAA Fluid Dynamics Conference and Exhibit.
[15] Stone C P, Duque E P, Hennes C C and Gharamkhani A 2010 AIAA Paper 312
[16] Papadakis G, Voutsinas S, Sieros G and Chaviaropoulos T 2012 CFD Aerodynamic Analysis of Non-Conventional Airfoil Sections for Very Large Rotor Blades The Science of Making Torque from Wind
[17] Voutsinas S 2006 Int. Journal of Computational Fluid Dynamics 20
[18] Elderidge J, Colonius T and Leonard A 2002 J. Computational Physics 179 371–399
[19] Oxley G S 2009 A 2-D Hybrid Euler-Compressible Vortex Particle Method For Transonic Rotorcraft Flows Ph.D. thesis Dept. Mech. & Aero. Engineering, Carleton University, Ottawa, Canada
[20] Madrlepis D 1990 AIAA Journal 28 213–221
[21] Greengard L and Rokhlin V 1987 J. Computational Physics 73
[22] Hockney R W and Eastwood J W 1981 Computer Simulation Using Particles (McGraw-Hill)
[23] Menter F 1994 AIAA Journal 32 1598–1605.
[24] Spalart P and Allmaras S 1994 Recherche Aerospatiale 5–21
[25] Colonius T and Taira K 2008 Comput. Methods Appl. Mech. Engrg. 197 2131–2146
[26] Hirsch C 1990 Numerical Computation of Internal and External Flows vol 2 (Wiley)
[27] Beale J T and Majda A 1985 J. Computational Physics 58
[28] Richardson S and Cornish A 1977 J. of Fluid Mechanics 82 309–319
[29] Boisvert R 1984 Academic Press 35–44
[30] Cocle R, Winckelmans G and Daeninck G 2008 J. Computational Physics 227 9091–9120
[31] Christiansen J 1973 J. Computational Physics 13 363–379
[32] Balls G T and Colella P 2002 J. Computational Physics 180 25–53
[33] Prospathopoulos J, Papadakis G, Sieros G, Voutsinas S, Chaviaropoulos P and Diakakis K 2014 Assessment of the aerodynamic characteristics of thick airfoils in high Reynolds and moderate Ma numbers using CFD modeling submitted to The Science of Making Torque from Wind
[34] Degond P and Mas-Gallic S 1989 Math. Comput. 53 485–526