Supplementary Material for Existence of a sharp transition in the peak propulsive efficiency of a low-*Re* pitching foil

Anil Das[1] and Ratnesh K. Shukla[1]† and Raghuraman N. Govardhan[1]‡

[1]Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 560012, India

(Received xx; revised xx; accepted xx)

1. Simulation methodology

All our simulation runs were performed using a GPU (graphics processing unit) accelerated extension of the original two-dimensional viscous vortex particle method (VVPM) developed in Eldredge (2007). Like the original vortex particle methods, the VVPM relies on a Lagrangian description of the vorticity field through vortex particles and employs appropriate techniques for their advection and diffusion and also vorticity creation at the no-slip boundaries. In the following, we briefly describe the key features of the viscous vortex particle method focusing particularly on the key advances in the implementational aspects that have allowed us to perform GPU-accelerated simulations at substantially reduced total run times. For an introduction and detailed description of the vortex particle methods, their theoretical analysis and criteria for convergence including elaborate description of the techniques that enable their efficient use on conventional CPU based architectures (e.g. adaptive fast multipole method, remeshing, particle strength exchange etc.) we refer the interested readers to the book by Cottet & Koumoutsakos (2000) and the review article by Koumoutsakos (2005).

The VVPM method approximates the vorticity formulation of the two-dimensional incompressible Navier-Stokes equations given below:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot (\nabla \omega) = \nu \nabla^2 \omega, \tag{1.1}$$

where ω and ν are the vorticity and kinematic viscosity of the incompressible fluid, respectively. The boundary conditions that account for the rigid body kinematics (pitching motion of the airfoil) and the far-field uniform flow are given by

$$\begin{array}{ll} \omega(\mathbf{x}, 0) = 0 & \text{everywhere} \\ \mathbf{u} = \mathbf{U}_b(t) + \Omega(t)\hat{k} \times (\mathbf{x}_{pivot} - \mathbf{x}_b) & \text{on the airfoil surface} \\ \mathbf{u} \to U_\infty \hat{i} & \text{in the farfield } (|\mathbf{x}| \to \infty) \end{array} \tag{1.2}$$

where $U_\infty \hat{i}$ represents the free stream velocity field with $\mathbf{U_b}(t)$ and $\Omega(t)$ as the translational and the angular velocities of the body, respectively. In the above expressions, $\mathbf{x}_{pivot}$ denotes the coordinates of the location about which the rigid body rotates (quarter chord for pitching airfoil) and $\mathbf{x}_b$ represents a point on the surface of the rigid body with $\hat{i}$ and $\hat{k}$ as the unit vectors along x and z directions, respectively. We begin all the simulation runs from an initially inviscid uniform flow field with zero vorticity that corresponds to an impulsive start of the initially stationary rigid body.

† Email address for correspondence: ratnesh@mecheng.iisc.ernet.in
‡ Email address for correspondence: raghu@mecheng.iisc.ernet.in

The vortex particle method (Cottet & Koumoutsakos 2000; Koumoutsakos 2005) relies on a Lagrangian description of the vorticity field through discrete vortex blobs as follows

$$\omega(\mathbf{x},t) = \sum_{i=1}^{N} \Gamma_i \, \eta_\epsilon(\mathbf{x} - \mathbf{x}_i(t)) \tag{1.3}$$

where Γ_i denotes the circulation associated with the i^{th} blob with N as the total number of blobs employed for representing ω. The net vorticity carried by each blob and its distribution inside the blob is determined by the function η_ϵ, where the parameter ϵ determines the core size of equisized blobs. Use of finite sized blobs through the desingularized function η_ϵ of the following form

$$\eta_\epsilon = \frac{1}{2\pi\epsilon^2} \exp\left(\frac{-\|\mathbf{x} - \mathbf{x_i}\|^2}{2\epsilon^2}\right)$$

ensures that the induced velocity remains bounded even when two blobs are located infinitesimally close to each other. In order to model advection and diffusion of vorticity the vortex blobs are allowed to advect with the flow and exchange their strengths respectively as described below.

The governing equation for the evolution of vorticity (1.1) can be expressed in the Lagrangian framework as

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}(\mathbf{x},t) \qquad \text{Advection}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \nu\nabla^2\omega \qquad \text{Diffusion}$$

In the advection substep of the VVPM the positions of the vortex blobs are updated based on the local fluid velocity. In our implementation, we employ the classical fourth-order Runge-Kutta integrator to update the positions of the vortex blobs. In the diffusion substep of the VVPM the vortex blob strengths are updated to account for viscous diffusion while enforcing the no slip boundary condition at the surface of the rigid body.

The advection substep involves recovery of the velocity field from the vorticity through application of Biot-Savart's law with an infinite domain Green's function (see Cottet & Koumoutsakos 2000; Koumoutsakos 2005). The fluid velocity $\mathbf{u}(\mathbf{x},t)$ is given by (Eldredge 2007)

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_\omega + \mathbf{u}_\gamma + \mathbf{u}_\Omega + U_\infty\hat{i}$$

where $\mathbf{u}_\omega$ denotes the velocity induced by all the vortex blobs in the domain, $\mathbf{u}_\gamma$ denotes the velocity induced by the bound vortex sheet on the rigid body surface, and $\mathbf{u}_\Omega$ denotes the velocity due to rotation of the rigid body.

Recovery of $\mathbf{u}_\omega$ from the vorticity field obtained from (1.3) through a direct summation over all the vortex blobs is a computationally expensive $\mathcal{O}(N^2)$ operation as one must account for the mutual interaction between each of the N blobs present in the domain. In our implementation, we employ the adaptive fast multipole method (AFMM) (Carrier *et al.* 1988) to reduce computational expense associated with calculation of $\mathbf{u}_\omega$ drastically to $\mathcal{O}(N)$. Moreover, use of a GPU-accelerated implementation of AFMM (Goude & Engblom 2013) allows us to achieve more than an order of magnitude speedup over the standard CPU implementation. With an increase in the number of particles we find calculation of not just $\mathbf{u}_\omega$ but also $\mathbf{u}_\gamma$ and $\mathbf{u}_\Omega$ to be computationally demanding. Therefore, we have extended the CUDA based GPU implementations to accelerate the calculation of $\mathbf{u}_\gamma$ and $\mathbf{u}_\Omega$ significantly. These critical advances allow us to efficiently

perform long time highly-resolved simulations of pitching foil over more than a hundred cycles using vortex blobs in excess of a few million. In contrast with the conventional VVPM implementation on multicore CPU architectures the advances outlined above allow us to perform computations on thousands of cores that are available on modern general purpose graphics processing units and thus achieve over an order of magnitude speed up (compared to an OPENMP implementation on 12 cores we observe over a ten fold speed up using a Tesla K40 GPU with 2880 cores).

To account for the viscous diffusion, we approximate the diffusion operator using the particle strength exchange method (PSE) (Koumoutsakos & Leonard 1995) wherein the strength of the individual vortex blobs is updated using the classical fourth-order Runge-Kutta integrator. The compact support of the PSE operator allows for fast computations of this substep. To increase computational efficiency, the unbalanced quad-tree build from the advection substep is used to build an adjacency list for each particle which in turn is employed in the PSE. To enforce no-slip boundary condition on the rigid body surface, we follow the earlier approaches that rely on Lighthill's vorticity creation mechanism to eliminate the spurious slip through a vortex sheet (Koumoutsakos *et al.* 1994; Ploumhans & Winckelmans 2000). The strength of this vortex sheet is determined using the boundary element method (BEM). The sheet is subsequently diffused into the flow using a vorticity flux as suggested in Ploumhans & Winckelmans (2000).

The accuracy of PSE operator depends crucially on a proper overlap between the vortex blobs failing which a convergence to the actual vorticity distribution is not guaranteed. To maintain an overlap between the vortex blobs and to avoid unnecessary growth in the number of vortex blobs as a result of the skewness caused by the Lagrangian treatment, we apply remeshing every few timesteps by mapping the vorticity field back onto a regular Cartesian grid. Here, we follow the remeshing methodology developed in Ploumhans & Winckelmans (2000) for a generic body to ensure a more or less uniform distribution of particles throughout the simulation. Finally, the forces and moment experienced by the translating and rotating rigid body (pitching airfoil) are computed from the integral formulations involving time derivative of the impulse and angular-impulse of the total vorticity, respectively (Wu 1981).

2. Numerical tests

To validate our implementation, we have applied the GPU-accelerated VVPM to simulate flow past translating and rotating rigid bodies. In this section we present results from three such demonstration runs and compare them with prior vortex particle and grid based simulation results. In all the runs, the time step and spatial resolution (h) are carefully chosen such that $Re_\Gamma = |\omega|h^2/\nu \sim \mathcal{O}(1)$ as suggested by Ploumhans & Winckelmans (2000) for a well resolved vortex particle simulation. These parameters vary from run to run both with respect to Re and St. Furthermore, the size of the largest panel (Δs_{max}) used in the BEM is such that the ratio $h/\Delta s_{max} \sim 1$ following the recommendation of Eldredge (2007). We use an overlap ratio (h/ϵ) of 0.8, which falls between 0.7 and 1.0, as suggested by Barba (2004) for high accuracy simulation with Gaussian blobs. In all the simulation runs, we perform remeshing every six time steps. During remeshing, we remove particles whose strength scaled by the kinematic viscosity (Γ/ν) falls below a certain user-prescribed cutoff. In all our simulations, we use a cutoff of 10^{-7}. This cutoff criterion along with $Re_\Gamma \sim \mathcal{O}(1)$ ensures that particles spanning at least seven orders of magnitude in circulation are retained in all our simulation runs. Further, during remeshing we add particles with zero strength around the existing particles so that there are enough particles to ensure diffusion of vorticity along the domain periphery.

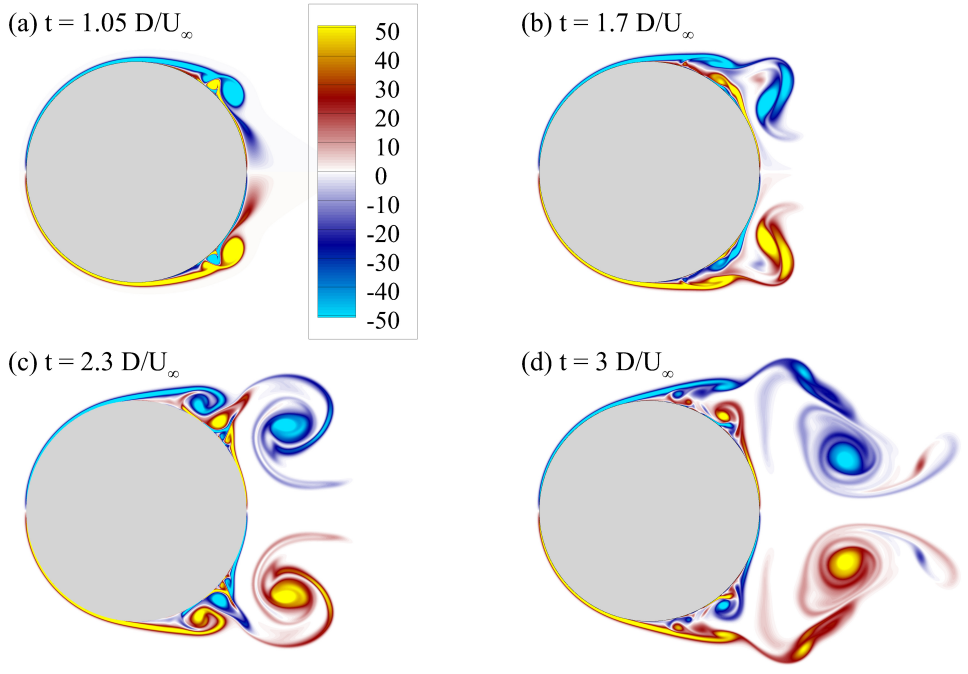

FIGURE 1. Vorticity contours at four successive time instants for the impulsive start of a circular cylinder at $Re = 9500$.

This prevents spurious artificial accumulation of vorticity along the domain boundary. The number of layers of zero strength particles to be added are decided based on the remesh frequency.

2.1. *Flow past an impulsively started circular cylinder at $Re_D = 9500$*

To begin with we consider flow past an impulsively started circular cylinder of diameter D at Re_D of 9500. Reynolds number based on cylinder diameter as the characteristic length scale $Re_D = U_\infty D/\nu$, where U_∞ denotes the velocity of the circular cylinder. The test case has been simulated previously with the vortex particle method (Koumoutsakos & Leonard 1995) and more recently using an advanced vorticity-based method that relies on wavelets to achieve high adaptivity (Rossinelli *et al.* 2015).

Figure 1 depicts the vorticity contours computed from our GPU-accelerated implementation of the VVPM. The VVPM effectively captures the evolution of the strong interacting vortices that result from unsteady flow separation over the cylinder surface while maintaining perfect symmetry that is expected to hold during the initial stages of the flow evolution. As a further check, figure 2 depicts a comparison of the evolution of the drag coefficient obtained from our runs with the results reported in Rossinelli *et al.* (2015). The excellent agreement between the two results, as evidenced from figure 2, serves to validate our VVPM implementation for rigid translating bodies.

2.2. *Flapping elliptical wing*

To assess the accuracy and robustness of our implementation on more generic rigid body motions that involve combined rotation and translation, we next consider the flapping elliptical wing test case that is widely used as a model problem for investigation of aerodynamics of insect hovering. The setup of this test case is exactly the same as

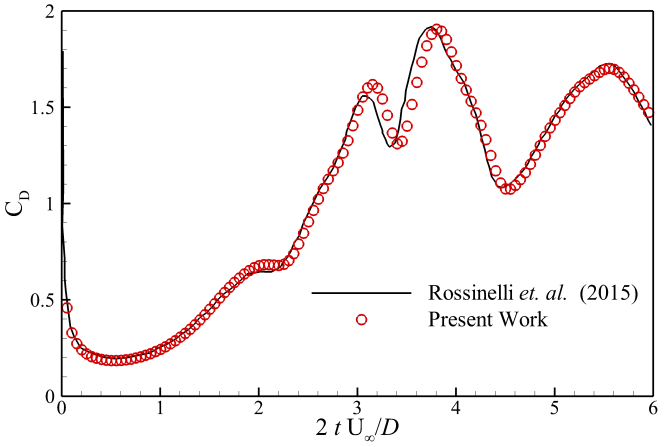

FIGURE 2. Drag coefficient as a function of time for an impulsively started circular cylinder at $Re_D = 9500$.

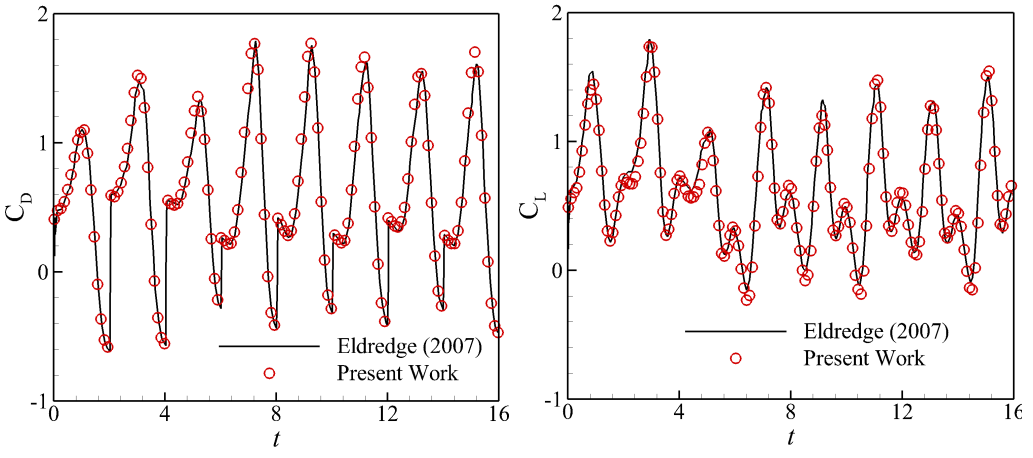

FIGURE 3. Drag (left) and lift (right) coefficients as a function of time for the flapping elliptical wing compared with the VVPM simulation results from Eldredge (2007).

the one considered in Eldredge (2007). An ellipse of aspect ratio 10 undergoes combined sinusoidal translational and rotational motions given by

$$\mathbf{X}_0(t) = 1.4c\,\cos\left(\frac{\pi t}{2}\right)\hat{i}, \quad \theta_0(t) = \frac{\pi}{2} + \frac{\pi}{4}\sin\left(\frac{\pi}{4} + \frac{\pi t}{2}\right), \tag{2.1}$$

where $\mathbf{X}_0$ denotes the coordinates of the center of the ellipse with c as the chord length. The Reynolds number based on the maximum translational velocity of $2.2c$ is set to 75 as in Eldredge (2007). To account for the far field quiescent conditions corresponding to zero flow we equate U_∞ to zero.

Figure 3 presents a comparison between the temporal evolution of the lift and drag forces experienced by the flapping ellipse obtained from our simulation runs with the values reported in Eldredge (2007). An excellent agreement between the two results certifies the accuracy of our GPU-accelerated approach for general rigid body motions. The vorticity contours for the flapping elliptical wing depicted in figure 4 compare favorably with the ones presented in figure 7 in Eldredge (2007).

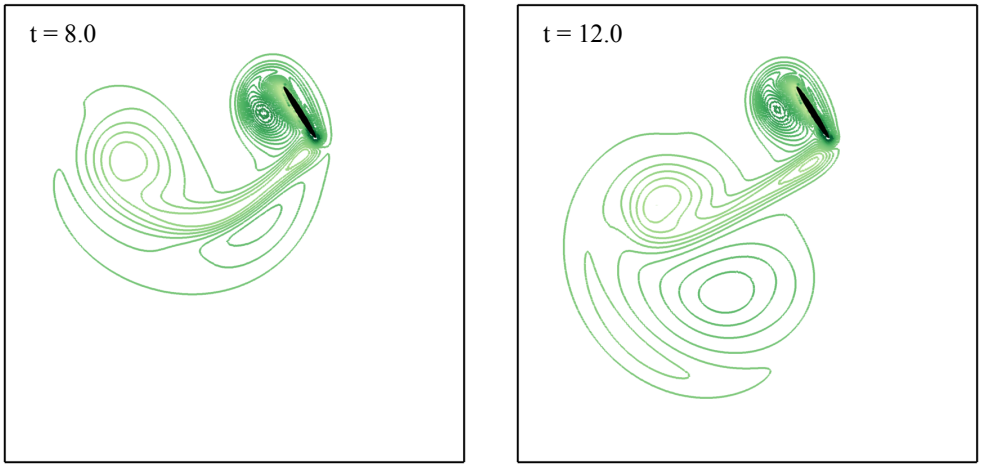

FIGURE 4. Vorticity contours at $t = 8$ and 12 for the flapping elliptical wing test case. Equispaced contour levels range from -20 to 20.

2.3. *Flow past a pitching airfoil at $Re_c = 1000$*

Finally, we assess the efficacy of the GPU-accelerated viscous vortex particle method in simulating highly separated flow over a rigid pitching airfoil that experiences highly fluctuating drag and lift forces and moments. We consider flow past a NACA 0012 airfoil pitching about its center at a Reynolds number Re_c of 1000 ($Re_c = U_\infty c/\nu$, with c as the chord length, U_∞ as the freestream velocity and ν as the kinematic viscosity of the fluid). The sinusoidal pitching motion is given by

$$\theta_0(t) = \frac{\theta_{min} + \theta_{max}}{2} - \frac{\theta_{min} - \theta_{max}}{2} \cos(2\pi f t), \tag{2.2}$$

where $\theta_{min} = 10^\circ, \theta_{max} = 30^\circ$ and $f = U_\infty/c$.

Figure 5 depicts the drag and lift coefficients along with the non-dimensional moment as a function of time computed from our GPU-accelerated implementation of the VVPM. For comparison results from a grid-based finite element simulation method (Mittal & Tezduyar 1992) have also been included. Despite vast differences between the two approaches we find a reasonable agreement in the drag, lift and moment predictions. The flow fields depicted in figure 6 illustrate the complex vortical structures and their interactions that are effectively captured by our high resolution simulation run.

REFERENCES

BARBA, L. A. 2004 Vortex method for computing high-Reynolds number flows: Increased accuracy with a fully mesh-less formulation. PhD thesis, California Institute of Technology.

CARRIER, J., GREENGARD, L. & ROKHLIN, V. 1988 A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. and Stat. Comput.* **9** (4), 669–686.

COTTET, G.-H. & KOUMOUTSAKOS, P. 2000 *Vortex methods: theory and practice*. Cambridge university press.

ELDREDGE, J. D. 2007 Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method. *J. Comput. Phys.* **221** (2), 626–648.

GOUDE, A. & ENGBLOM, S. 2013 Adaptive fast multipole methods on the GPU. *J. Supercomput.* **63** (3), 897–918.

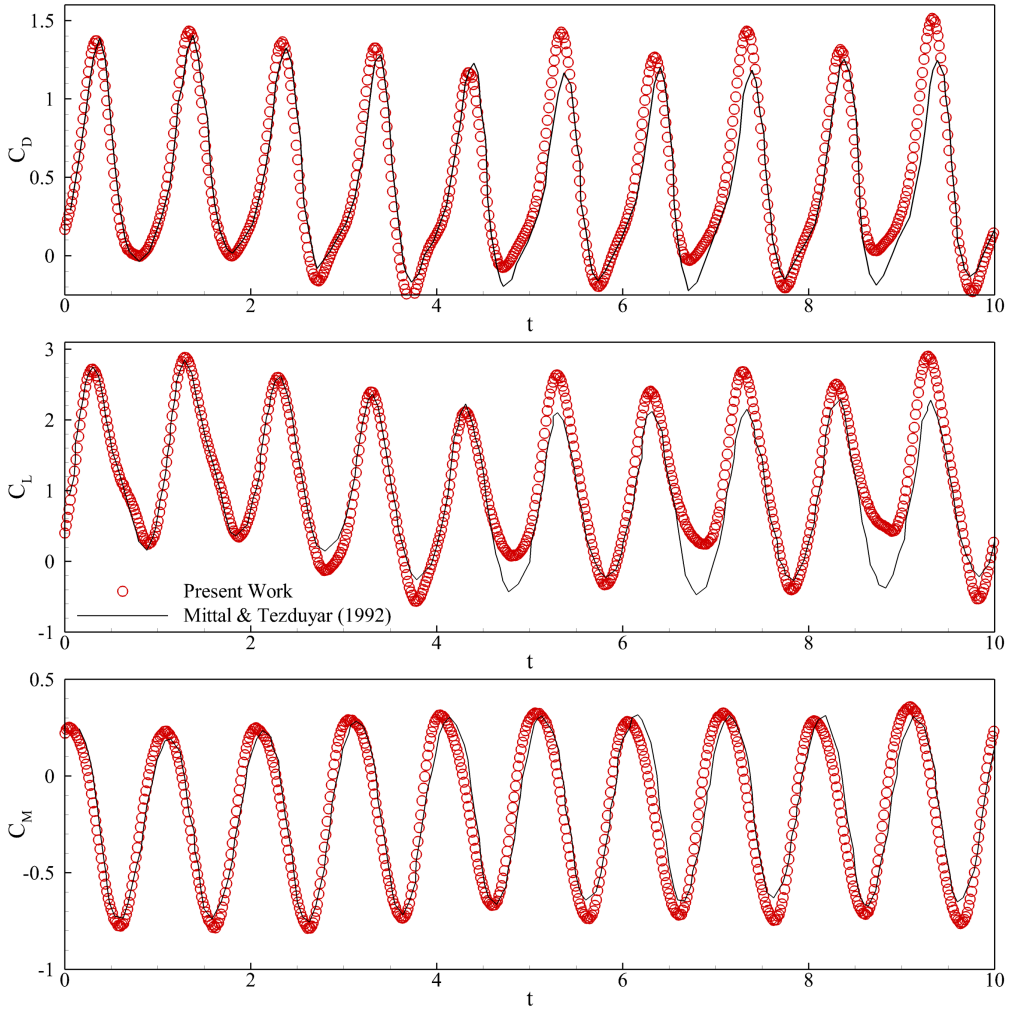

FIGURE 5. Drag (C_D), lift (C_L) and moment (C_M) coefficients as a function of time for the flow past a pitching airfoil at $Re_c = 1000$ compared with the finite-element simulation results from Mittal & Tezduyar (1992).

KOUMOUTSAKOS, P. 2005 Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.* **37** (1), 457–487.

KOUMOUTSAKOS, P. & LEONARD, A. 1995 High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.* **296**, 1–38.

KOUMOUTSAKOS, P., LEONARD, A. & PÉPIN, F. 1994 Boundary conditions for viscous vortex methods. *J. Comput. Phys.* **113**, 52–61.

MITTAL, S. & TEZDUYAR, T. E. 1992 A finite element study of incompressible flows past oscillating cylinders and aerofoils. *Int. J. Numer. Meth. Fluids* **15** (9), 1073–1118.

PLOUMHANS, P. & WINCKELMANS, G. S. 2000 Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *J. Comput. Phys.* **165** (2), 354–406.

ROSSINELLI, D., HEJAZIALHOSSEINI, B., VAN REES, W., GAZZOLA, M., BERGDORF, M. & KOUMOUTSAKOS, P. 2015 MRAG-I2D: multi-resolution adapted grids for remeshed vortex methods on multicore architectures. *J. Comput. Phys.* **288**, 1–18.

WU, J. C. 1981 Theory for aerodynamic force and moment in viscous flows. *AIAA J.* **19** (4), 432–441.

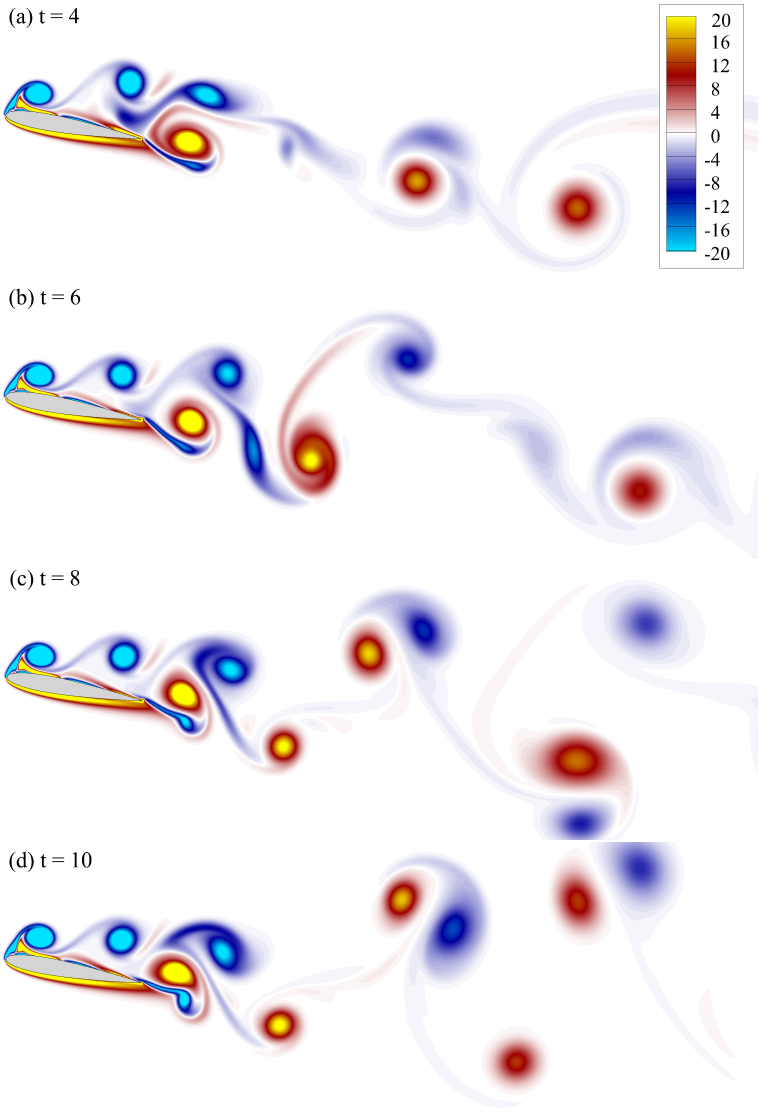

FIGURE 6. Vorticity contours at four successive time instants for the flow past a pitching airfoil at $Re_c = 1000$.