Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 197 (2008) 1296–1304
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Billion vortex particle direct numerical simulations of aircraft wakes

Philippe Chatelain [a], Alessandro Curioni [b], Michael Bergdorf [a], Diego Rossinelli [a], Wanda Andreoni [b], Petros Koumoutsakos [a,*]

[a] Chair for Computational Science, ETH Zurich CH-8092, Switzerland
[b] Computational Sciences, IBM Research Division – Zurich Research Laboratory, Switzerland

Received 10 July 2007; received in revised form 23 October 2007; accepted 8 November 2007

Abstract

We present the Direct Numerical Simulations of high Reynolds numbers aircraft wakes employing vortex particle methods. The simulations involve a highly efficient implementation of vortex methods on massively parallel computers, enabling unprecedented simulations using billions of particles.

The method relies on the Lagrangian discretization of the Navier–Stokes equations in vorticity–velocity form and relies on remeshing of the particles in order to ensure the convergence of the method. The remeshed particle locations are utilized for the computation of the field quantities, the discretization of the differential operators for diffusion and vortex stretching, and the solution of the Poisson equation for the advection velocity field. The method exhibits excellent scalability up to 16k BG/L nodes. The results include unprecedented Direct Numerical Simulations of the onset and the evolution of multiple wavelength instabilities induced by ambient noise in aircraft vortex wakes at $Re = 6000$.

PACS: 01.30.−y

Keywords: Vortex method; Wake vortices; Particle method

1. Introduction

Particle methods are distinguished by their robustness and adaptivity in simulations of convection dominated flows. Vortex methods exemplify the computational advantages and challenges of particle methods in simulations of incompressible vortical flows. These simulations are based on the discretization of the vorticity–velocity formulation of the Navier–Stokes equations in a Lagrangian form. Vortex particles discretize only the part of the domain that is occupied by vorticity while the Lagrangian formulation allows for automatic adaptivity and allows for significantly larger time steps than the ones required for the stability of the corresponding Eulerian discretizations. At the same time vortex methods can be inaccurate due to the distortion of the Lagrangian computational elements and computationally inefficient due to the use of irregular particle locations for the discretization of the governing equations.

In the recent years hybrid techniques (see [1,2] and references therein) have been proposed where a mesh is used along with the particles in order to develop efficient and accurate computations of vortical flows. The mesh is used in order to reinitialize the distorted particle locations [3–6,1] thus ensuring the convergence of the method. In addition the mesh enables the efficient computation of differential operators, for diffusion and vortex stretching, and the use of fast Poisson solvers for the computation of the field

* Corresponding author.
E-mail addresses: pchatela@inf.ethz.ch (P. Chatelain), cur@zurich.ibm.com (A. Curioni), bergdorf@inf.ethz.ch (M. Bergdorf), diegor@inf.ethz.ch (D. Rossinelli), and@zurich.ibm.com (W. Andreoni), petros@ethz.ch (P. Koumoutsakos).
URL: http://www.cse-lab.ethz.ch (P. Chatelain).

0045-7825/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.11.016

equations. The particles and the mesh exchange field quantities and particle strengths via moment conserving interpolations.

These methodological advances introduce a number of challenges for the massively parallel implementation of vortex methods that have hindered in the past large scale Direct Numerical Simulations. In this paper we develop efficient domain decompositions and optimized data mappings that rely on the Message Passing Interface (MPI). The algorithm is implemented for the distributed-memory architecture of the IBM BG/L using up to 16K CPUs and involving up to six billion particles.

We demonstrate the novel capabilities of this method in the Direct Numerical Simulation (DNS) of aircraft wakes. These wakes consist of long trailing vortices that can subject the following aircraft to a large downwash. This effect imposes stringent safety requirements on distances between aircraft, limiting the landing and take-off capacities of airports. Several research efforts have focused on the identification of the governing physical mechanisms of wake evolution that would lead to design of vortex wake alleviation schemes. Such schemes include the modification of the lift distribution to enhance sensitivity to instabilities [7–10] and the introduction of suitable perturbations [11]. Flight realistic conditions involve turbulent flows ($Re \sim 10^6$) in unbounded domains for which DNS reference data is still lacking. State-of-the-art simulations have been presented for low resolution LES in large domains [12]. In addition Vortex method simulations [13,14] achieved $Re = 5000$ DNS in short domains and investigated various subgrid stress models for LES in long domains. The present work enables unprecedented resolutions for the DNS of long wavelength instabilities. The long domain calculation at $Re = 6000$ presented herein constitutes the largest DNS ever achieved for a vortex particle method.

The paper is structured as follows: Section 2 covers the method and its parallel implementation; in Section 3, we present validation and scalability results for our method, along with a moderately large DNS result of a wake instability.

2. Methodology

2.1. The vortex particle method

We consider a three-dimensional incompressible flow and the Navier–Stokes equations in its velocity ($\mathbf{u}$)-vorticity ($\boldsymbol{\omega} = \nabla \times \mathbf{u}$) form

$$\frac{\mathrm{D}\boldsymbol{\omega}}{\mathrm{D}t} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\nabla^2\boldsymbol{\omega}, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where $\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla$ denotes the Lagrangian derivative and ν is the kinematic viscosity.

Vortex methods discretize the vorticity field with particles, characterized by a position $\mathbf{x}_p$, a volume V_p and a strength $\boldsymbol{\alpha}_p = \int_{V_p}\boldsymbol{\omega}\,\mathrm{d}\mathbf{x} \approx \boldsymbol{\omega}_p V_p$. Particles are convected by the flow field and their strength is modified to account for vortex stretching and diffusion

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}_p), \tag{3}$$

$$\frac{\mathrm{d}\boldsymbol{\alpha}_p}{\mathrm{d}t} = \int_{V_p}(\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\nabla^2\boldsymbol{\omega}\,\mathrm{d}\mathbf{x} \tag{4}$$

$$\simeq \big((\boldsymbol{\omega}\cdot\nabla)\mathbf{u}(\mathbf{x}_p) + \nu\nabla^2\boldsymbol{\omega}(\mathbf{x}_p)\big)V_p. \tag{5}$$

Using the definition of vorticity and the incompressibility constraint the velocity field is computed by solving the Poisson equation

$$\nabla^2\mathbf{u} = -\nabla\times\boldsymbol{\omega}. \tag{6}$$

The solution of this equation can be computed by using the Green's function solution of the Poisson equation or via fast Poisson solvers on the mesh of the present hybrid formulation (as discussed below).

2.2. Remeshed vortex particle methods

2.2.1. Reinitialization

The adaptivity of computational elements in vortex methods comes at the expense of the convergence and accuracy of the method [15,16]. Particles follow the flow map, resulting in an inhomogeneous particle distribution and the generation of spurious vortical structures. Remeshing has been proposed [3,4,1] as a remedy to maintain adaptivity while reducing the artificial dissipation of vortex methods. Remeshing consists in the periodic regularization onto a grid of the particle set via high order interpolation

$$\boldsymbol{\alpha}_q = \sum_p \boldsymbol{\alpha}_p W\left(\frac{\mathbf{x}_q - \mathbf{x}_p}{h}\right), \tag{7}$$

where h is the mesh spacing, $\mathbf{x}_q$ lies on the mesh and W is the interpolation kernel. In the present work, remeshing is performed at the end of each time step and uses the third order accurate M'_4 interpolation formula of [17].

The remeshed vortex particle method thus constitutes a clear departure from the purely Lagrangian character of vortex methods. Adaptivity is not lost however: it is simply not implicit anymore. Remeshing actually allows its control, in a fashion consistent with Adaptive Multi-Resolution (AMR) approaches or multilevel particle methods [18].

2.2.2. Efficient computation

One can extend the interpolation procedure of Eq. (7) to the field quantities themselves

$$\boldsymbol{\omega}(\mathbf{x}_q) = \sum_p \boldsymbol{\alpha}_p \frac{1}{h^3} W\left(\frac{\mathbf{x}_q - \mathbf{x}_p}{h}\right) \tag{8}$$

thus enabling additional computational advances. Differential operators (such as those for stretching and diffusion in Eq. (4)) are carried out on a mesh using fourth order finite differences. This enables orders of magnitude improvement over computations involving pure particle descriptions

where local interactions are computed through the use of mollifying kernels and require costly neighbor finding loops.

The Poisson equation (6) is solved on the mesh too. In the present application, periodic boundary conditions are used in all directions; they allow the straightforward translation to Fourier space

$$\hat{\mathbf{u}} = -\frac{\imath\mathbf{k}}{|\mathbf{k}|^2} \times \hat{\boldsymbol{\omega}}. \tag{9}$$

For the wake geometries of Section 3, the use of periodic boundary conditions in the transverse directions constitutes an approximation of the physical unbounded conditions, an approximation which is mitigated by the large size of the periodic box. We note that transverse unbounded conditions can be achieved by using hybrid Fast Multipole-VIC methods [13,14] but also with fast convolutions in Fourier space [19]; this extension is an area of ongoing work.

Mesh quantities can then be interpolated onto particle locations through the converse of Eq. (8)

$$f(\mathbf{x}_p) = \sum_{i,j,k} f_{i,j,k} W\left(\frac{\mathbf{x}_p - \mathbf{x}^h}{h}\right), \tag{10}$$

where f represents in turns the velocity field $\mathbf{u}$ or the vorticity time-derivative $\mathrm{d}\boldsymbol{\alpha}/\mathrm{d}t$, obtained beforehand from the vorticity Laplacian and the stretching term on the mesh. The combination of the third order accurate interpolations and fourth order finite differences yields a scheme which is third order accurate in space.

Finally, we account for the important observation [20,13] that the discretization of the three-dimensional vorticity field is not divergence-free. This spurious divergence is controlled here by means of a projection algorithm applied every few (5–10) time-steps. In the Fourier space ($\hat{\boldsymbol{\omega}}$), this relaxation consists in the explicit enforcement of null divergence

$$\hat{\boldsymbol{\omega}}' = \hat{\boldsymbol{\omega}} - \frac{\mathbf{k}\cdot\hat{\boldsymbol{\omega}}}{|\mathbf{k}|^2}\mathbf{k}. \tag{11}$$

2.2.3. Time integration

An important aspect of vortex particle methods is the Lagrangian simulation of the convection (Eq. (3)) which enables the use of time steps that are orders of magnitude larger than the corresponding steps of Eulerian descriptions.

Particle methods are free of the linear advection stability constraint of explicit mesh-based methods, i.e. the usual CFL condition

$$\Delta t|\mathbf{u}|_\infty/h < \mathrm{O}(1). \tag{12}$$

In vortex particle methods stability constraints are dictated from higher order terms that correspond to the crossing of particle trajectories. The stability criterion does not involve the mesh resolution and relates the time step to the strain time scale

$$\Delta t|\nabla\mathbf{u}|_\infty < \mathrm{O}(1). \tag{13}$$

We mention that the viscous diffusion operator introduces an additional stability constraint

$$\Delta t\nu/h^2 < \mathrm{O}(1), \tag{14}$$

which combined with Eq. (13) yields a particle Reynolds number condition

$$|\nabla\mathbf{u}|_\infty h^2/\nu < \mathrm{O}(1). \tag{15}$$

The time integration of Eqs. (3) and (4) is carried out with a third order low-storage Runge–Kutta scheme [21].

2.3. Implementation for parallel computer architectures

The proposed hybrid vortex particle methods encounter a number of challenges in their implementation in massively parallel architectures. The simultaneous presence of grid and particles and the need for an efficient particle-grid communication and load distributions have hindered in the past large scale Direct Numerical Simulations using vortex methods.

2.3.1. Parallel particle mesh library

The method was implemented as a client application of the open source Parallel Particle Mesh (PPM) library [22]. PPM provides a general-purpose framework that can handle the simulation of particle-only, mesh-only or particle-mesh systems. The library can define topologies, i.e. space decompositions and the assignment of sub-domains to processors, which achieve particle- and mesh-based load balancing. PPM is written in Fortran 90 on top of the Message Passing Interface (MPI).

The data communication is organized in mappings which can be applied to either mesh points or particles. Mesh points need to be mapped within the same topology for the computation of finite difference stencils or into a different topology for the computation of Fourier transforms. A local or global mapping can be applied to particles. A local neighbor-to-neighbor mapping is needed to re-assign particles which are advected from a subdomain into a neighboring one within the same topology; global mappings are used for the initial insertion of the particles or for their mapping into a new topology.

All the mappings – for particles or mesh points – follow a stack paradigm. The user can *push* data onto the stack, e.g. blocks of the velocity and vorticity fields; the library consolidates it into a single send and receive communication. Once the communications are done, the user simply has to *pop* data off the stack.

The library organizes the resulting data fluxes into a sequence of point-to-point (`MPI_Sendrecv`) communications which can be blocking or non-blocking, at the user's discretion, depending on the architecture and the observed performance.

2.3.2. Client specific optimizations

The vortex client uses PPM inside a MPI Cartesian communicator and user-defined mesh-based decompositions. In

the present simulations, a Poisson equation is solved on the mesh in Fourier space at every right-hand side evaluation. The three-dimensional Fourier transforms are based on the FFTW library [23]; they are carried out in parallel in two or three steps which correspond to as many topologies. Two steps correspond to two-dimensional and one-dimensional transforms carried out in slab[1] and pencil[2] topologies respectively. In three steps, each transform is carried out on a pencil topology. Because the mappings inside the FFTs are global, we added a mapping option to PPM which does not rely on a communication sequence but uses a collective communication routine (`MPI_Alltoallv`).

The two-step transform requires one less mapping but limits the number of processors to the largest linear resolution of the problem. The distribution of data transfer however changes dramatically between the two solutions. For the slab–pencil method, every processor communicates with all the others, while for the pencil–pencil–pencil, it has to interact with $O(\sqrt{N_{CPU}})$ other processors. Finally, some precautions are taken in the ordering of the transforms. Because the first transform (real-to-complex) halves the problem size in that direction, we set the first and last transforms to the directions with the most and least grid points, respectively. This ensures that the problem granularity remains fine for the second and third transforms, thus helping the load balancing.

2.3.3. Platform

The code is run on an IBM Blue Gene/L solution with dual cores nodes based on the PowerPC 440 700 Mhz low power processor. Each node has 512 MB of memory. The computations are all carried out in co-processor mode: one of the two CPUs is fully devoted to the communications. The machine used for production was the BG/L at IBM T.J. Watson Research Center – Yorktown Heights[3] whereas porting, optimization and testing was done on the BG/L system of the IBM Zurich Research Laboratory. Machine dependent optimization consisted in

(1) data reordering and compiler directives to exploit the double floating point unit of the PowerPC 440 processors,
(2) mapping of the cartesian communicators to the BG/L torus,
(3) use of the BG/L tree network for global reductions.

3. Results

3.1. Problem description

In the present simulations of aircraft trailing vortices the coordinate axes are set so that the x-axis is along the streamwise direction, the y-axis is in the spanwise direction separating the vortical axes and z is in the vertical. In three dimensions, a finite span wing with non-zero lift sheds a vortex sheet along its trailing edge.

This sheet quickly rolls up to form vortex pairs that depend on the lift distribution. Extended flaps, for example, generate a sharp lift jump along the wing and consequently an additional vortex pair.

The evolution and eventual destruction of the trailing vortices is affected by several types of instabilities, usually classified according to their wavelength. Long wavelength instabilities are the most powerful to drive the collapse of a vortex pair albeit with a slow growth rate. The well-known Crow instability [24] is an example of such instabilities that deforms the vortex lines into sinusoidal structures until vortices of opposite sign reconnect and form rings.

More complex systems with multiple vortex pairs can undergo other instabilities. A rapidly growing, medium-wavelength instability has been the focus of recent experimental [8,10,25] and numerical studies [12–14]. This instability occurs in the presence of a secondary vortex pair that is counter-rotating relative to the main pair. These secondary vortices are generated by a sufficient negative load on the horizontal tail or the inboard edge of outboard flaps. Being weaker, they eventually wrap around the primary ones in so-called Ω-loops, leading to the reconnection of vortices of unequal circulations. This in turn triggers an accelerated vortex destruction.

3.2. Validation

3.2.1. Convergence

We use the geometry of this particular medium wavelength instability for the validation of our code. The geometry of the problem is taken from [13]. The vortices have algebraic cores

$$\omega(r) = \frac{1}{2\pi\sigma^2}(1 + (r/\sigma)^2)^{-2} \tag{16}$$

with cores sizes $\sigma_1/b_1 = 0.075$ and $\sigma_2/b_1 = 0.05$. The secondary pair has a span $b_2/b_1 = 0.3$ and a circulation $\Gamma_2/\Gamma_1 = -0.3$. The vortex center lines are perturbed by a sine function with an amplitude $10^{-6}b_1$. The non-dimensionalization is based on the equivalent vortex pair $\Gamma_0 = \Gamma_1 + \Gamma_2$, $b_0 = \frac{\Gamma_1 b_1 + \Gamma_2 b_2}{\Gamma_0} = 1.3b_1$ and $t_0 = \frac{2\pi b_0^2}{\Gamma_0}$. The Reynolds number is $Re = \Gamma_0/\nu = 3500$. The length of our periodic domain is chosen as $L_x/b_0 = 0.76$, the wavelength which maximizes the instability growth rate. The other directions are also periodic; their extents are set to $L_y/L_x = 5$ and $L_z/L_x = 3$ to mitigate the effect of the periodic images.

Three grid sizes were considered, $64 \times 320 \times 192$, $128 \times 640 \times 384$, and $256 \times 1280 \times 768$, resulting in 4, 32 and 252 million particles respectively. All three configurations were run on 1024 processors of IBM BG/L. The time-step was kept constant for all resolutions $\Delta t = 3.310^{-4}t_0$. We emphasize again that the time step in particle

[1] Topology with no division in two given directions.
[2] Topology with no division in one given direction.
[3] Compiled with XLF version 10.1, with BG/L driver V1.3 and FFTW 3.1.1.

methods is not explicitly linked to the grid resolution but rather to the strain field that is simulated.

Fig. 1 shows the evolution of vorticity iso-surfaces and the wrapping-around of the secondary vortices around the main ones. Diagnostics (Fig. 2) such as the evolution of enstrophy, which measures the energy decay and the evolution of the effective numerical viscosity confirm the convergence of the method.

3.2.2. Scalability

The parallel efficiency was assessed for $512 \leqslant N_{\mathrm{CPU}} \leqslant 16\,384$ on IBM BG/L. The client used blocking routines for local mappings and the collective solution (discussed in Section 2.3.2) for the transposition of data inside the Fourier transforms, carried out in three pencil decompositions. The collective call option was retained as it brought a 30% improvement in execution time over the point-to-point communication sequence at large N_{CPU}. This figure, which is consistent with the observations in [26], demonstrates the level of optimization of collective communication and the *alltoall* in particular on the BG/L torus.

We measure the strong efficiency as

$$\eta_{\mathrm{strong}} = \frac{N^{\mathrm{ref}}_{\mathrm{CPUS}} T(N^{\mathrm{ref}}_{\mathrm{CPUS}})}{N_{\mathrm{CPUS}} T(N_{\mathrm{CPUS}})}, \tag{17}$$

where T is the average computation time of one time step. In order to test the code up to the large sizes allowed by BG/L, we used $N^{\mathrm{ref}}_{\mathrm{CPUS}} = 2048$ and a problem size of $768 \times 1024 \times 2048$ or 1.6 billion particles. This brings the per-processor problem size from 786 432 down to 98 304 when we run on the maximum number of processors. The curve (Fig. 3) displays a plateau up to $N_{\mathrm{CPUS}} = 4096$, with the per-processor problem size becoming progressively smaller and communication overhead overwhelming the computing cycles.

From this result, we base our weak scalability study on a constant per-processor number of particles of $M_{\mathrm{per\ CPU}} \simeq 4 \times 10^5$. We used the following measure:

$$\eta_{\mathrm{weak}} = \frac{T(N^{\mathrm{ref}}_{\mathrm{CPUS}}, M^{\mathrm{ref}})}{T(N_{\mathrm{CPUS}}, \frac{N_{\mathrm{CPUS}}}{N^{\mathrm{ref}}_{\mathrm{CPUS}}} M^{\mathrm{ref}})}, \tag{18}$$

where we took $N^{\mathrm{ref}}_{\mathrm{CPUS}} = 512$. The code displays (Fig. 3) excellent scalability up to $N_{\mathrm{CPUS}} = 4096$. Eq. (18) assumes linear complexity for the problem at hand. There is however an $\mathrm{O}(N \log N)$ component to the overall complexity of the present problem as we are solving the Poisson equation for the convection velocity. The two curves (with and without the cost for the solution of the Poisson equation) are shown in Fig. 3; the relatively small gap between the two curves manifests the good performance of the Poisson solver and indicates that the scalability is dependent on the communication imbalance that occurs during the mapping of the convected particles. Finally we note that using 16K processors enables unprecedented simulations using $\mathrm{O}(10^{10})$ vortex particles.

3.3. Instability initiation by ambient noise

For our first application, we consider the configuration presented in the state of the art calculations in [12, see configuration 2] simulating the onset of instabilities of multiple wavelengths in a long domain. The domain length is chosen as the wavelength of maximum growth rate for the Crow instability, $L_x = 9.4285 b_1$. The transversal dimensions are $L_y = 1/2 L_x$ and $L_z = 3/8 L_x$. The vortices have Gaussian cores

$$\omega(r) = \frac{1}{2\pi\sigma^2} \exp(-(r/2\sigma)^2) \tag{19}$$

with $\sigma_1/b_1 = 0.05$ and $\sigma_2/b_1 = 0.025$. The secondary pair is located at $b_2/b_1 = 0.5$, with a relative strength $\Gamma_2/\Gamma_1 =$

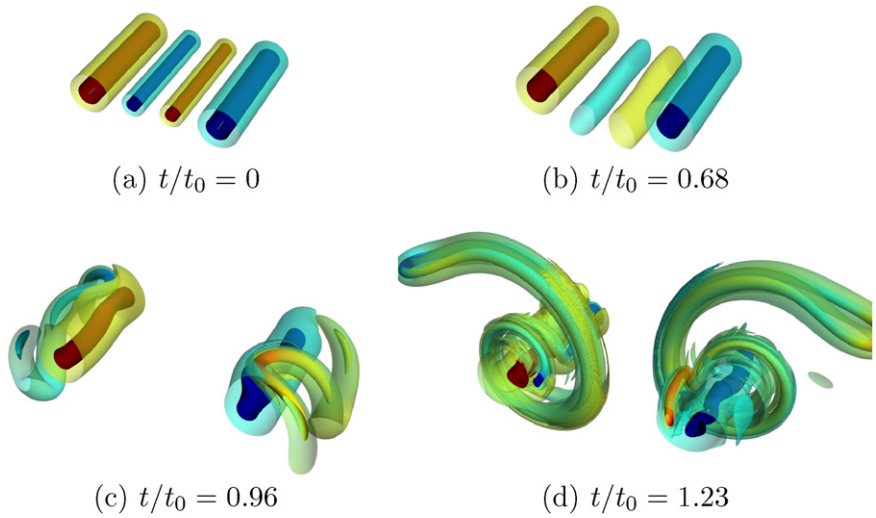
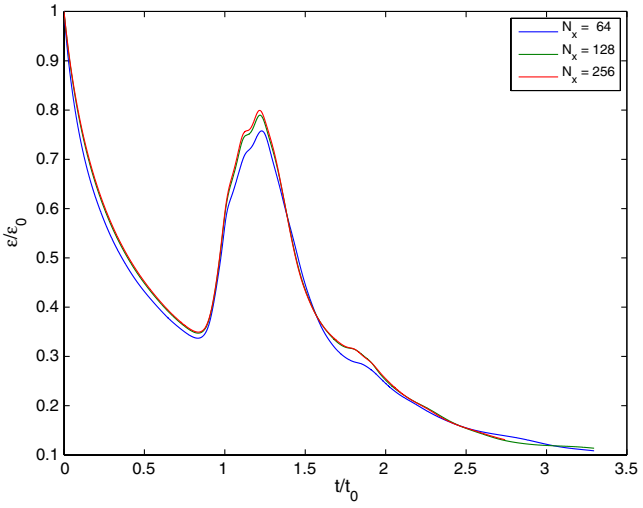
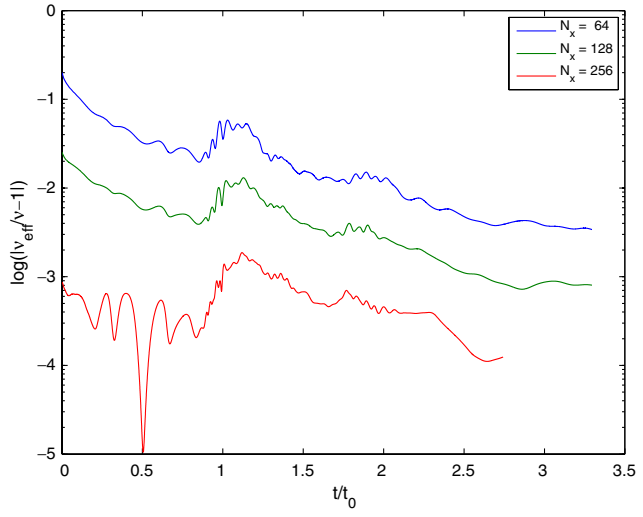


Fig. 1. Medium-wavelength instability of counter-rotating vortices, $128 \times 640 \times 384$-grid: evolution of vorticity iso-surfaces. The opaque surface corresponds to $|\boldsymbol{\omega}| = 10\Gamma_1/b_1^2$; the transparent one, to $|\boldsymbol{\omega}| = 2\Gamma_1/b_1^2$. (a) $t/t_0 = 0$, (b) $t/t_0 = 0.68$, (c) $t/t_0 = 0.96$, (d) $t/t_0 = 1.23$.

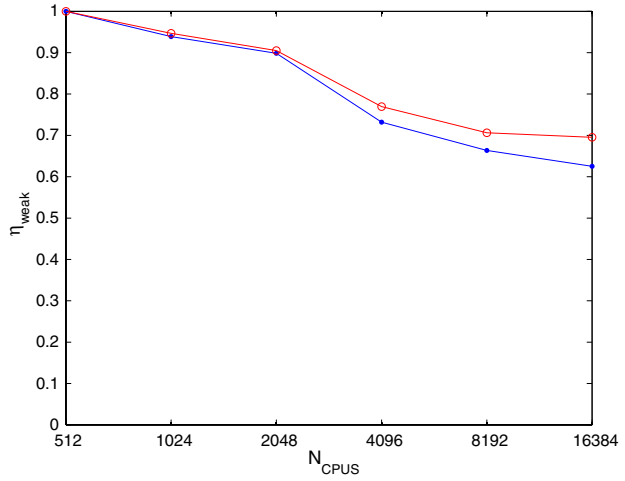


(a) Enstrophy, $\epsilon = \int \boldsymbol{\omega} \cdot \boldsymbol{\omega}\, dV$

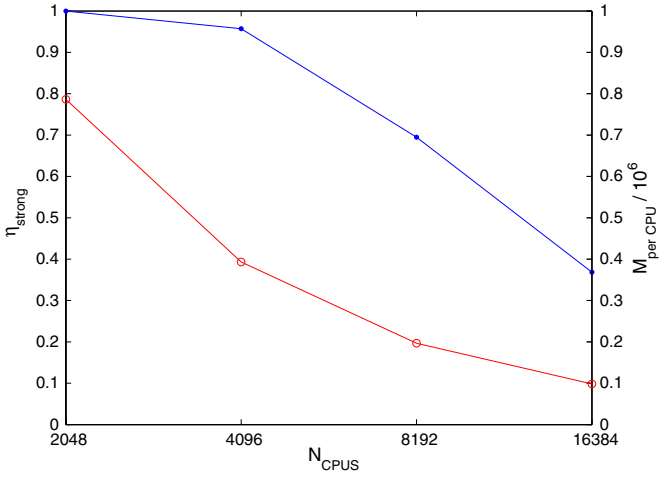


(b) Relative error in the effective kinematic viscosity, $\nu_{\text{effective}} = -\frac{dE/dt}{\epsilon}$

Fig. 2. Medium-wavelength instability of counter-rotating vortices: convergence and diagnostics for three spatial resolutions. (a) Enstrophy, $\epsilon = \int \boldsymbol{\omega} \cdot \boldsymbol{\omega} \mathrm{d}V$ and (b) relative error in the effective kinematic viscosity, $\nu_{\text{effective}} = -\frac{\mathrm{d}E/\mathrm{d}t}{\epsilon}$.



(a) Weak scalability for a per-processor problem size = $4\,10^5$;full problem (solid dots) and excluding the Poisson solver (circles)



(b) Strong scalability (solid dots) and per-processor size (circles)

Fig. 3. Medium-wavelength instability of counter-rotating vortices: parallel efficiencies on IBM BlueGene/L. (a) Weak scalability for a per-processor problem size = 4×10^5; full problem (solid dots) and excluding the Poisson solver (circles) and (b) strong scalability (solid dots) and per-processor size (circles).

-0.35. In addition to the initially unperturbed vortices, the vorticity field is filled with a white noise that produces $u_{\text{RMS}} = 0.005 u_{\max}$. We study this flow with DNS at $Re_{\Gamma_1} = 6000$. This represents a three-fold increase over previously reported Reynolds numbers [12]. In addition, these prior simulations used a coarse resolution and a crude LES model (MILES [27]) to model the high Reynolds number dynamics of the flow. High Reynolds number LES (accounting only for the SGS diffusivity have been reported in [14]. The present DNS is afforded due to a mesh resolution of $2048 \times 1024 \times 768$ and 1.6 billion particles. It is run on 4096 CPUs; the wall-clock computation time was 39 s on average per time step. With approximately 10 000 time steps, this represents a time-to-solution of 100 h.

The level of spurious dissipation is shown in Fig. 4. It remains at a low level, below 3%, through most of the simulation.

Fig. 5 shows that this system with a random initial condition picks up the medium-wavelength instability. At $t/t_0 = 0.25$ (Fig. 5), we count 10 and 11 Ω-loops along the two primary vortices. This corresponds to the average wavelengths $\lambda/b_1 = 0.943$ and 0.86. These values are sensibly different from the ones reported in [12], 1.047 and 1.309. This comparison, however, considers the problem at the end of the exponential growth and ignores the uneven distribution of loop wavelengths and hence, individual growth rates. An exhaustive investigation of this matter goes beyond the scope of the present work and will

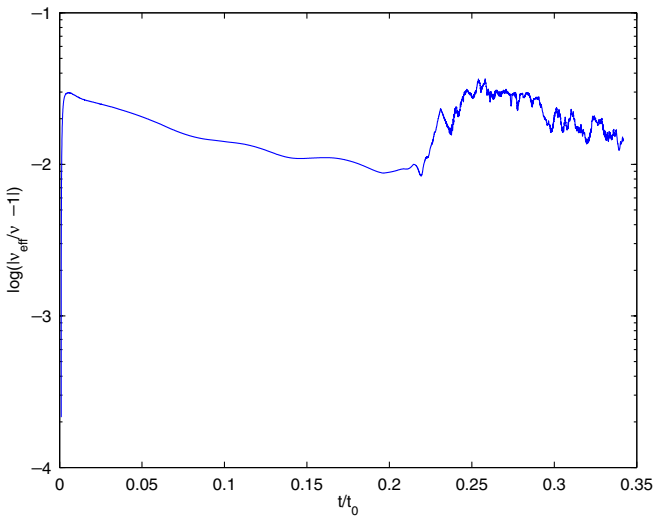


Fig. 4. Counter-rotating vortices in ambient noise: relative error in the effective kinematic viscosity.

be presented in a separate work. Fig. 6 presents the details of a reconnection region. In particular, one can observe the interaction of unequal strength tubes and the generation of fine scales, both at the reconnection and along the loops.

These observations are confirmed in frequency space. Fig. 7 shows the evolution of longitudinal energy modes

$$E(k_x) = \int \hat{\mathbf{u}} \cdot \hat{\mathbf{u}}^* \, \mathrm{d}y \, \mathrm{d}z, \tag{20}$$

where $\hat{\mathbf{u}}$ denotes the Fourier transform in the x-direction. The initial condition imposes that all modes start from a similar energy level. The high k_x modes undergo an slight decay before the expected exponential growth. The mode $k_x = 11(2\pi/L_x)$ is the fastest growing one and corresponds to as many initiations of Ω-loops (Fig. 5). The initial viscous decay affects less the Crow mode $k_x = (2\pi/L_x)$ which eventually catches up with the medium wavelength instability.

Conversely, in physical space, the kinetic energy of the two-dimensional flow in a cross-flow slice does not necessarily decay monotonically like the total energy, as shown in Fig. 8. This feature is clearly more pronounced in the present work than in the LES results of [12] and matches the behavior of the experimental results of [28].

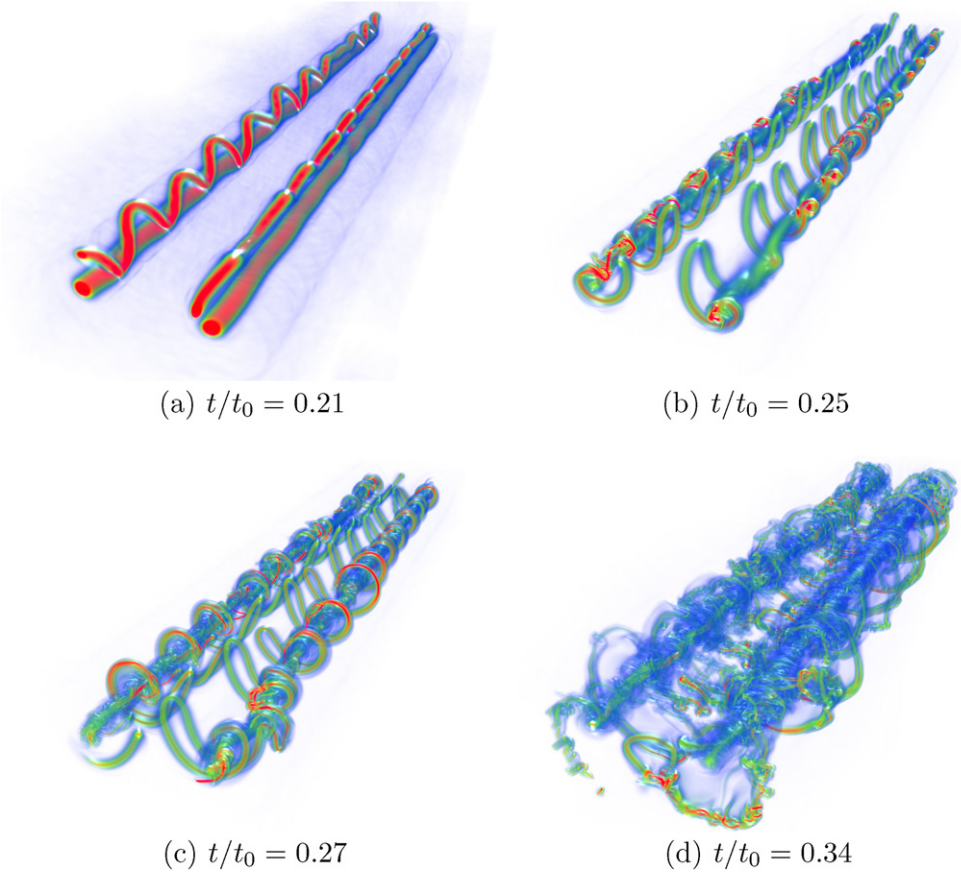


Fig. 5. Counter-rotating vortices, initiation by ambient noise: visualization of the vorticity structures by volume rendering. High vorticity norm regions correspond to red and opaque; low vorticity are blue and transparent. (a) $t/t_0 = 0.21$, (b) $t/t_0 = 0.25$, (c) $t/t_0 = 0.27$ and (d) $t/t_0 = 0.34$. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

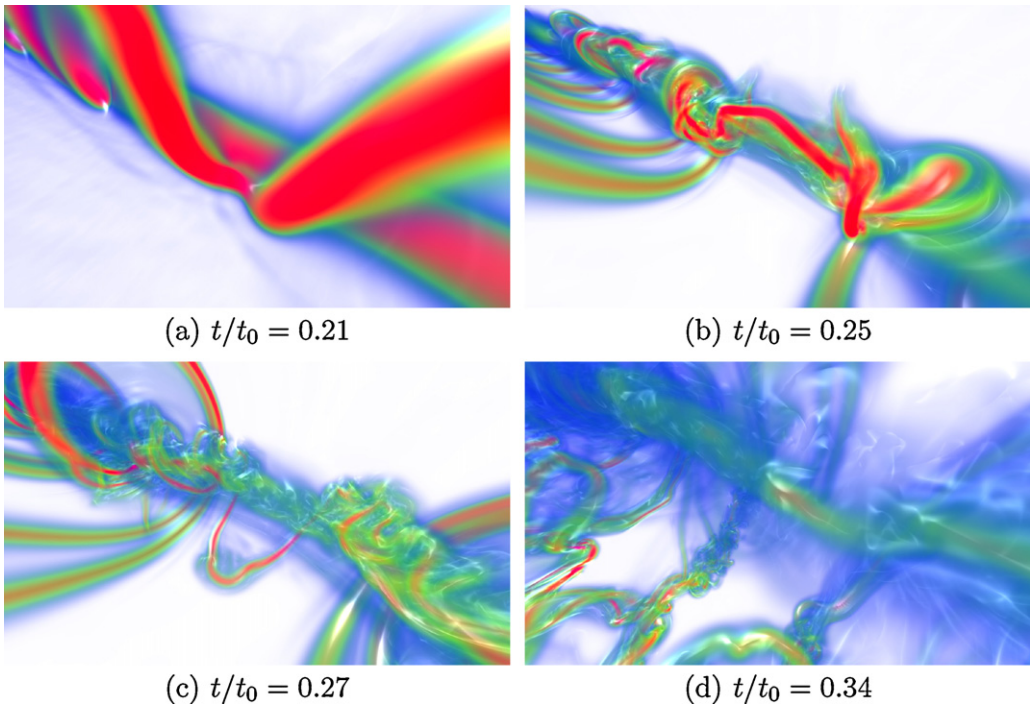


Fig. 6. Counter-rotating vortices, initiation by ambient noise: details of a reconnection region. (a) $t/t_0 = 0.21$, (b) $t/t_0 = 0.25$, (c) $t/t_0 = 0.27$ and (d) $t/t_0 = 0.34$.

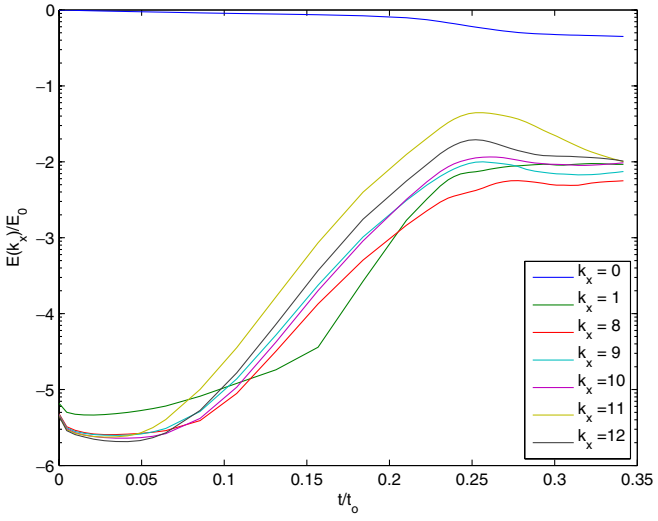


Fig. 7. Counter-rotating vortices in ambient noise: evolution of the stream-wise Fourier modes of kinetic energy.

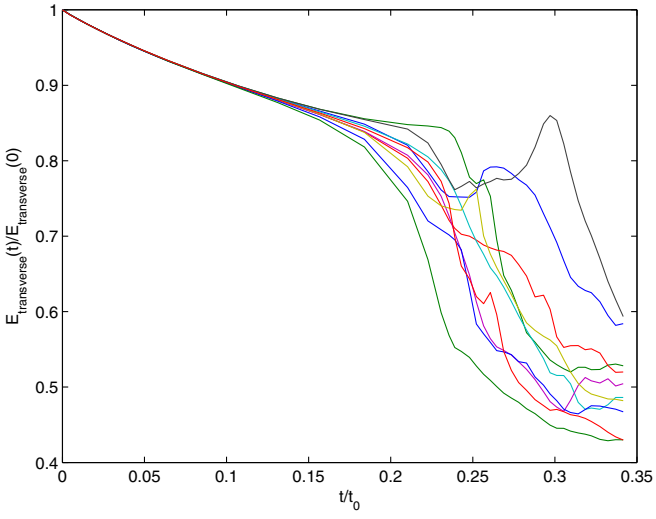


Fig. 8. Counter-rotating vortices in ambient noise: kinetic energy of the transverse flow $E_{\text{transverse}} = \int 0.5(u_y^2 + u_z^2)\mathrm{d}S$ in 10 equally spaced $x = \text{cst}$ planes.

4. Conclusions

We have implemented a hybrid particle-mesh vortex element method for large parallel distributed-memory computations enabling simulations using a billion vortex particles and providing us with unprecedented details of the evolved dynamics. The code exhibits reliability across platforms, very good efficiency and scalability, up to 16 k processors on the IBM BG/L. We note that the present count of operations in our code amounts to 2.6 TFlops. This relatively low count of operations is attributed to the large overhead in transfer of information between grid and particles. At the same time we wish to emphasize that this particle description is the one that allows time steps that are orders of magnitude larger than the ones employed in grid based methods. We continue to work on the further optimization of the code and we hope to exceed 5 TFlops in the near future as well as implement the method on 32K processors.

Our test and production runs on BG/L constitute the largest problem size ever achieved by a vortex method. We are in the process of investigating the underlying vorticity dynamics and we will report in the near future on the quantification of the dynamics of the flow. Ongoing work for the simulation of vortex wakes includes the implementation of unbounded boundary conditions, as discussed in Section 2.2.2. This will waive the need for a large periodic box, allowing for finer resolution of the governing physical mechanisms and consequently the simulation of even higher Reynolds number flows. Future work will include DNS at higher *Re*, LES model validation and wake alleviation optimization studies. Finally we note that the present work can be readily extended to other particle flow simulations such as Smooth Particle Hydrodynamics (SPH) and in particular its extension using remeshed particle methods [5,29]. In fact the present $O(10^{10})$ particle simulations could translate to even higher numbers in SPH where the requirement for an elliptic solver is relaxed. The extension of the present particle simulations to other areas of computational science, such as astrophysics, granular flows and cell biology, is a subject of ongoing investigations among several members of the PPM community.

Acknowledgement

The present simulations of aircraft wakes have been inspired by several discussions with Grégoire Winckelmans.

References

[1] G. Winckelmans, Vortex methods, in: Erwin Stein, René De Borst, Thomas J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, vol. 3, John Wiley and Sons, 2004.
[2] P. Koumoutsakos, Multiscale flow simulations using particles, Annu. Rev. Fluid Mech. 37 (2005) 457–487.
[3] G.-H. Cottet, Artificial viscosity models for vortex and particle methods, J. Comput. Phys. 127 (2) (1996) 299–308.
[4] P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, J. Comput. Phys. 138 (2) (1997) 821–857.
[5] A.K. Chaniotis, D. Poulikakos, P. Koumoutsakos, Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows, J. Comput. Phys. 182 (2002) 67–90.
[6] J.D. Eldredge, T. Colonius, A. Leonard, A vortex particle method for two dimensional compressible flow, J. Comput. Phys. 179 (2002) 371–399.
[7] J.D. Crouch, Instability and transient growth for two trailing-vortex pairs, J. Fluid Mech. 350 (1997) 311–330.
[8] D.A. Durston, S.M. Walker, D.M. Driver, S.C. Smith, Ö. Savas, Wake vortex alleviation flow field studies, J. Aircraft 42 (4) (2005) 894–907.
[9] W.R. Graham, S.-W. Park, T.B. Nickels, Trailing vortices from a wing with a notched lift distribution, AIAA J. 41 (9) (2003) 1835–1838.
[10] J.M. Ortega, Ö. Savas, Rapidly growing instability mode in trailing multiple-vortex wakes, AIAA J. 39 (4) (2001) 750–754.
[11] J.D. Crouch, G.D. Miller, P.R. Spalart, Active-control system for breakup of airplane trailing vortices, AIAA J. 39 (12) (2001) 2374–2381.
[12] E. Stumpf, Study of four-vortex aircraft wakes and layout of corresponding aircraft configurations, J. Aircraft 42 (3) (2005) 722–730.
[13] G. Winckelmans, R. Cocle, L. Dufresne, R. Capart, Vortex methods and their application to trailing wake vortex simulations, C.R. Phys. 6 (4–5) (2005) 467–486.
[14] R. Cocle, L. Dufresne, G. Winckelmans, Investigation of multiscale subgrid models for less of instabilities and turbulence in wake vortex systems, Lect. Notes Comput. Sci. Engrg. 56 (2007).
[15] J.T. Beale, A. Majda, Vortex methods I: convergence in 3 dimensions, Math. Comput. 39 (159) (1982) 1–27.
[16] J.T. Beale, On the accuracy of vortex methods at large times, in: B. Engquist et al. (Ed.), Proceedings of the Workshop on Comput. Fluid Dyn. and React. Gas Flows, IMA, University of Minnesota, 1986, Springer-Verlag, New York, 1988, p. 19.
[17] J.J. Monaghan, Extrapolating *b* splines for interpolation, J. Comput. Phys. 60 (2) (1985) 253–262.
[18] M. Bergdorf, G.-H. Cottet, P. Koumoutsakos, Multilevel adaptive particle methods for convection–diffusion equations, Multiscale Model. Simulat. 4 (1) (2005) 328–357.
[19] Roger Hockney, James Eastwood, Computer Simulation using Particles, Institute of Physics Publishing, 1988.
[20] G.S. Winckelmans, A. Leonard, Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows, J. Comput. Phys. 109 (2) (1993) 247–273.
[21] J.H. Williamson, Low-storage Runge–Kutta schemes, J. Comput. Phys. 35 (1) (1980) 48–56.
[22] I.F. Sbalzarini, J.H. Walther, M. Bergdorf, S.E. Hieber, E.M. Kotsalis, P. Koumoutsakos, PPM a highly efficient parallel particle mesh library for the simulation of continuum systems, J. Comput. Phys. 215 (2006) 566–588.
[23] M. Frigo, S.G. Johnson, FFTW: an adaptive software architecture for the FFT, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, Seattle, WA, May 1998, pp. 1381–1384.
[24] S.C. Crow, Stability theory for a pair of trailing vortices, AIAA J. 8 (12) (1970) 2172–2179.
[25] R. Stuff, The near-far relationship of vortices shed from transport aircraft, in: AIAA (Ed.), AIAA Applied Aerodynamics Conference, 19th, Anaheim, CA, AIAA, 2001, pp. AIAA-2001–2429.
[26] George Almasi, Gyan Bhanot, Alan Gara, Manish Gupta, James Sexton, Bob Walkup, Scaling physics and material science applications on a massively parallel Blue Gene/L system, in: Proceedings of the 19th Annual International Conference on Supercomputing, 2005, pp. 246–252.
[27] J.P. Boris, F.F. Grinstein, E.S. Oran, R.L. Kolbe, New insights into large eddy simulation, Fluid Dyn. Res. 10 (4–6) (1992) 199–228.
[28] Jason M. Ortega, Stability characteristics of counter-rotating vortex-pairs in the wakes of triangular-flapped airfoils, PhD thesis, University of California, Berkeley, 2001.
[29] P. Chatelain, G.-H. Cottet, P. Koumoutsakos, PMH: particle mesh hydrodynamics, Int. J. Mod. Phys. C 18 (4) (2007) 610–618.