Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 9091–9120

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations

Roger Cocle *, Grégoire Winckelmans, Goéric Daeninck

Mechanical Engineering Department, Division TERM, and Center for Systems Engineering and Applied Mechanics (CESAME), Université Catholique de Louvain (UCL), 1348 Louvain-la-Neuve, Belgium

Received 23 April 2007; received in revised form 1 October 2007; accepted 8 October 2007
Available online 23 October 2007

Abstract

A new combination of vortex-in-cell and parallel fast multipole methods is presented which allows to efficiently simulate, in parallel, unbounded and half-unbounded vortical flows (flows with one flat wall). In the classical vortex-in-cell (VIC) method, the grid used to solve the Poisson equation is typically taken much larger than the vorticity field region, so as to be able to impose suitable far-field boundary conditions and thus approximate the truly unbounded (or half-unbounded) flow; an alternative is to assume periodicity. This approach leads to a solution that depends on the global grid size and, for large problems, to unmanageable memory and CPU requirements. The idea exploited here is to work on a domain that contains tightly the vorticity field and that can be decomposed in several subdomains on which the exact boundary conditions are obtained using the parallel fast multipole (PFM) method. This amounts to solving a 3-D Poisson equation without requiring any iteration between the subdomains (e.g., no Schwarz iteration required): this is so because the PFM method has a global view of the entire vorticity field and satisfies the far-field condition. The solution obtained by this VIC–PFM combination then corresponds to the simulation of a truly unbounded (or half-unbounded) flow. It requires far less memory and leads to a far better computational efficiency compared to simulations done using either (1) the VIC method alone, or (2) the vortex particle method with PFM solver alone. 3-D unbounded flow validation results are presented: instability, non-linear evolution and decay of a vortex ring (first at a moderate Reynolds number using the sequential version of the method, then at a high Reynolds number using the parallel version); instability and non-linear evolution of a two vortex system in ground effect. Finally, a space-developing simulation of an aircraft vortex wake in ground effect is also presented.

Keywords: Vortex methods; Parallel fast multipole method; Vortex-in-cell method; Particle methods; Lagrangian methods; Incompressible flows; Unsteady flows; Unbounded flows; Wall-bounded flows

* Corresponding author. Tel.: +32 10472151; fax: +32 10452692.
E-mail address: roger.cocle@uclouvain.be (R. Cocle).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.10.010

1. Introduction

The method developed here is a combination of the parallel fast multipole (PFM) method and of the vortex-in-cell (VIC) method that allows to efficiently simulate, in 2-D and in 3-D, incompressible unsteady flows that are unbounded or half-unbounded.

Vortex particle methods are based on the vorticity–velocity formulation of the Navier–Stokes equations and on the fact that, for incompressible flows, it is sufficient to follow the evolution of the vorticity field, the velocity field being recovered from the vorticity. See, e.g., Cottet and Koumoutsakos [9], Winckelmans [40] for global reviews. The Lagrangian treatment of the convection term leads to methods with negligible dispersion error. It also eliminates the CFL stability constraint; however, there is still an accuracy constraint that limits the relative rotation of vortex particles ($|\boldsymbol{\omega}|\Delta t$ must remain moderate). Vortex methods also have good energy conservation properties. These two qualities (negligible dissipation and negligible dispersion) make vortex methods suitable candidates for direct numerical simulation (DNS) and/or large-eddy simulation (LES) of complex convection dominated flows.

When using the vorticity–velocity formulation of the incompressible Navier–Stokes equations, one needs to solve a Poisson equation for the streamfunction. Two different approaches are commonly used:

- *Lagrangian vortex method*: the Poisson equation is solved using the unbounded Green's function approach (Biot-Savart). The unbounded domain is thus taken into account implicitly. Thanks to PFM methods [13,6], one is able to obtain, both in 2-D and in 3-D, the streamfunction (and thus the velocity, its gradient, etc.) with an $\mathcal{O}(N \log N)$ computational cost, where N is the number of particles. Those approaches make use of outer multipole expansions (i.e., expansions representing the field outside of a ball). There are also implementations that make use of both inner and outer expansions, allowing to reduce the cost to $\mathcal{O}(N)$. Our parallel 3-D implementation is based on an oct-tree and outer expansions; it also uses active error control based on tight error bounds, allowing to minimize the computational cost required to obtain the field with an error that is uniformly bounded in space at a prescribed level (the error tolerance being an input to the code): see [32,33,27,28]. Another parallel 3-D implementation is that by Krasny et al. [19].
- *Vortex-in-cell method*: in such hybrid Lagrangian–Eulerian methods, the Poisson equation is solved on a grid, see [7,11,12,25]. This is done using fast Poisson solvers. Those also have a computational cost $\mathcal{O}(M \log M)$, where M is the number of grid points, but are considerably faster than the PFM methods. However, boundary conditions are required. Hence, for unbounded (or half-unbounded) flows problems, the grid must be taken much larger than the vorticity field region, so that approximate far-field boundary conditions can be used. In some cases, one assumes periodicity: this too calls for a large grid. In both cases, the obtained solution is dependent on the global grid size, and so are the computational memory and cost requirements. In other implementations, a slightly modified Schwarz alternating algorithm can be used: an iterative method to obtain consistent boundary conditions on each subdomains and that avoids solving a boundary integral problem while still retaining good convergence properties [3,25]. Nevertheless, it results that the VIC approach used "alone" is not really appropriate for unbounded flows, as it does not retain the ability of vortex methods to exactly satisfy the far-field condition.

The present "VIC–PFM" method constitutes a combination of these two approaches. It uses the "smallest" possible domain (i.e., the "smallest" grid): one that tightly contains the vorticity region. In parallel, this domain is further decomposed by splitting it into several subdomains, with one subdomain per processor. The boundary conditions required on the sides of these subgrids are obtained "exactly" by using the PFM method, which has a global view of the entire vorticity field. The Poisson equation is then solved on each subgrid, i.e., locally on each processor, and the combination of all subgrid solutions still provides the solution of a truly unbounded flow. Moreover, from a computational point of view, the efficiency of this method is better than that obtained using a pure PFM method, while benefiting from the powerful parallelization of the fast multipole algorithm. As the grid can be taken compact, it is also better than that obtained using a pure VIC method (when the grid is taken large enough to properly approximate an unbounded domain). The present

VIC–PFM methodology is here also extended to unbounded flows with one periodic direction (e.g., useful for simulations of aircraft vortex wakes) and also to half-unbounded flows: those with a flat wall (e.g., simulations of aircraft wakes vortices in ground effect). A last variant, used to perform space-developing simulations (thus, simulations with inflow and outflow boundary conditions), is also implemented and is here illustrated on simulating a space-developing aircraft vortex wake in ground effect.

2. The VIC–PFM method

The VIC–PFM approach will be explained for the 3-D case only; the 2-D method can easily be deduced from it (and was also implemented). In this section, we first present the basic equations. Then, we explain the VIC–PFM methodology by presenting a global time step. Finally, we describe some essential "tools" used to ensure the method's accuracy/efficiency or required for some specific applications.

2.1. Basic equations

Let us first recall the vorticity–velocity formulation for the incompressible Navier–Stokes equations:

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

where D/Dt is the convective derivation, $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $\nabla \cdot (\mathbf{u}\boldsymbol{\omega})$ is the conservative form of $(\nabla \mathbf{u}) \cdot \boldsymbol{\omega}$, ν is the kinematic viscosity, and $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity field. In 2-D, $\boldsymbol{\omega} = \omega \hat{e}_z$ and there is no stretching term ($\nabla \cdot (\mathbf{u}\boldsymbol{\omega}) = 0$). The flow being incompressible, the velocity can be expressed as

$$\mathbf{u} = \nabla \times \boldsymbol{\psi} + \mathbf{U}_\infty \tag{2}$$

where $\boldsymbol{\psi}$ is the associated streamfunction, related to $\boldsymbol{\omega}$ by the Poisson equation,

$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega} \tag{3}$$

and $\mathbf{U}_\infty$ a freestream velocity, added when performing space-developing simulations (see Section 2.7.3).

In vortex methods, the vorticity field is discretized using a set of N Lagrangian particles, see Leonard [20] and global presentations in Leonard [21], Cottet and Koumoutsakos [9], and Winckelmans [40]. Here, we use the Gaussian function as regularization, so that the regularized particle vorticity field is

$$\boldsymbol{\omega}_\sigma(\mathbf{x}, t) = \sum_{p=1}^{N} \frac{1}{\left(\sqrt{2\pi}\sigma_p\right)^3} \exp\left(-\frac{1}{2}\frac{|\mathbf{x} - \mathbf{x}_p|^2}{\sigma_p^2}\right) \boldsymbol{\alpha}_p \tag{4}$$

where $\mathbf{x}_p$ is the position of particle p and $\boldsymbol{\alpha}_p = \int \boldsymbol{\omega}\, d\mathbf{x} = \boldsymbol{\omega}_p h_p^3$ is the particle's strength (with h_p the local grid size). The vector streamfunction, $\boldsymbol{\psi}_\sigma$, on the sides of the grid of each subdomain is thus obtained using the PFM method. To ensure a good agreement between the grid solver and the PFM solver, the core size σ_p of the regularized particles has to be chosen such that $\boldsymbol{\psi}_\sigma = \boldsymbol{\psi}_h$ (where $\boldsymbol{\psi}_h$ is the streamfunction obtained using the grid Poisson solver). In Fig. 1, we show the value of ψ_h obtained with the grid solver for a unit vorticity strength α_0 placed at $x = 0$. In the multipole method, for the direct interactions, the streamfunction is obtained as

$$\boldsymbol{\psi}_\sigma(\mathbf{x}, t) = \frac{1}{4\pi} \sum_p \operatorname{erf}\left(\frac{1}{\sqrt{2}}\frac{|\mathbf{x} - \mathbf{x}_p|}{\sigma_p}\right) \boldsymbol{\alpha}_p \tag{5}$$

We can then determine the proper σ_p that leads to the same regularization as the "intrinsic" regularization of the Poisson grid solver, which is due to the finite differences schemes used in those solvers. With the FishPack Poisson solver used here, we find $\sigma_p = 0.251 h_p$. The resulting $\boldsymbol{\psi}_\sigma$ is shown in Fig. 1. We observe that the two curves, $\boldsymbol{\psi}_\sigma$ and $\boldsymbol{\psi}_h$, are indeed in good agreement, which ensures the continuity of the streamfunction obtained with the Lagrangian field (i.e., the particles used in the PFM method) and with the Eulerian field (i.e., the field as seen by the grid used in the VIC method).

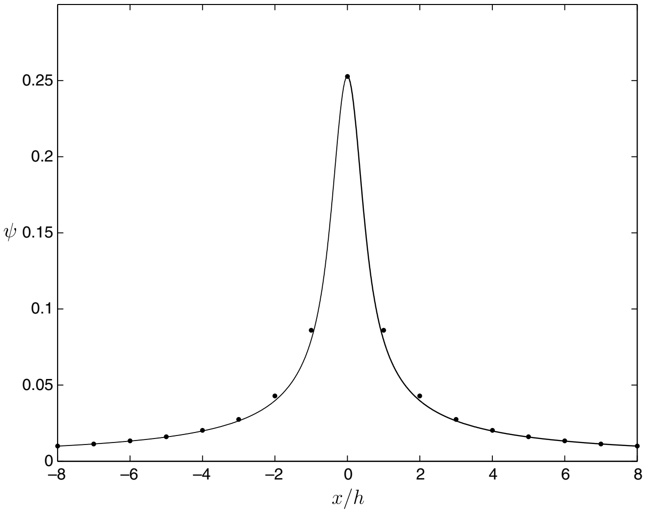


Fig. 1. The streamfunction obtained for a unit vorticity strength α_0 placed at $x = 0$: ψ_h obtained using the grid solver (dot) and ψ_σ obtained using the PFM method (solid).

2.2. Global time step

We now explain the VIC–PFM methodology. We consider the case with uniform grid (of grid size h); the case with non-uniform grid was also considered but the results are not presented here. The initial information is a set of particles of position $\mathbf{x}_p$ and strength $\boldsymbol{\alpha}_p$ which represents the complete vorticity field. Depending on the number of processors p used, the spatial domain defined by the particles, chosen so that it tightly includes the vorticity field, is split equally into several intervals in each direction (p_x intervals in the x-direction and equivalently, p_y and p_z for the others two directions) so that $p_x \times p_y \times p_z = p$ subdomains are created and allotted to the p processors. This is illustrated in Fig. 2, using a 2-D example to make it easier. For interpo-

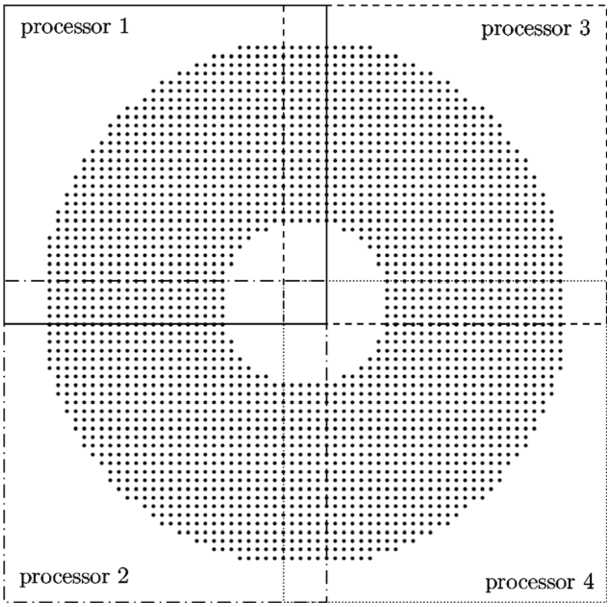


Fig. 2. Example of domain decomposition for a vorticity field shared between four processors. For proper interpolation reasons, an overlapping between the subdomains is required.

lation reasons that will be explained later, an overlapping between the subdomains is required (minimum $2h$). The particles are then distributed among the processors (i.e., the subdomains) depending on their positions. To prevent useless communications, some particles are repeated on several processors: we call them the ‘‘phantom’’ particles, in opposition to the ‘‘real’’ particles that are only present in one processor. Fig. 3 illustrates this distribution. We observe that the particles located inside the overlapping region are repeated between the different subdomains. It is also the case for the particles located up to a distance of $2h$ outside the grid limits. We will explain below, in the detailed time step explanation, how these ‘‘phantom’’ particles are useful during the interpolation/redistribution steps.

We also recall that, in Lagrangian methods, particle redistribution is an essential operation to ensure a good representation of the vorticity field and thus maintain the accuracy of the simulation as time proceeds. A particle redistribution is thus done every few time steps (e.g., typically 5, see [27,28]). When it is time to do a particle redistribution before the time step, one first creates a new set of particles on a regular lattice: here based on the VIC grid. This is done using the M'_4 high order interpolation/redistribution scheme (see Section 2.3), which preserves the moments of order zero, one and two of the vorticity field (corresponding, respectively, to the total vorticity $\mathbf{\Omega} = \int \boldsymbol{\omega}\,\mathrm{d}\mathbf{x}$, the linear impulse $\mathbf{I} = \frac{1}{2}\int \mathbf{x} \times \boldsymbol{\omega}\,\mathrm{d}\mathbf{x}$, and the angular impulse $\mathbf{A} = -\frac{1}{2}\int |\mathbf{x}|^2 \boldsymbol{\omega}\,\mathrm{d}\mathbf{x}$). In case no particle redistribution is done before the time step, the particles are not on a regular lattice, yet they are not much distorted.

In either case, at the beginning of the time step, each processor has a set of particles (‘‘real’’ and ‘‘phantom’’) of position $\mathbf{x}_p$ and strength $\boldsymbol{\alpha}_p$. The time step in each subdomain is then as follows:

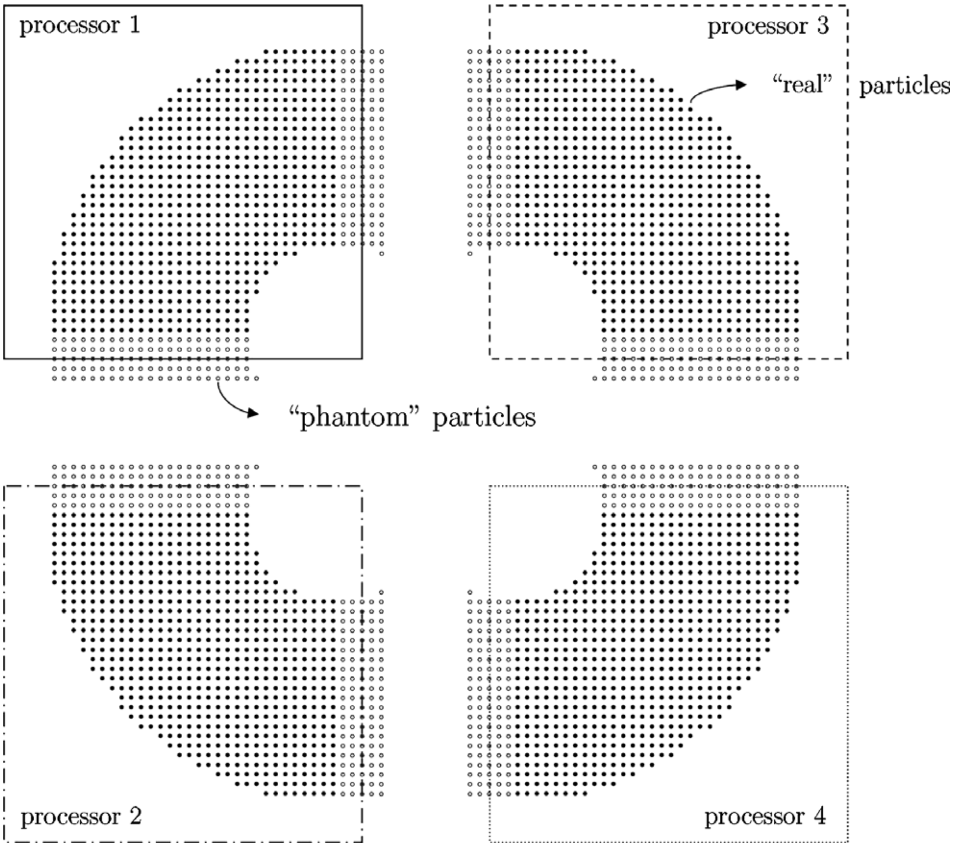


Fig. 3. Example of distribution of vortex particles between the processors. The ‘‘real’’ particles (●) are those which represent the vorticity field in the physical part of each subdomain. Some of these particles, the ‘‘phantom’’ particles (○), are repeated between the processors.

(1) Create the subgrid used to solve the Poisson equation (resulting from the spatial subdivision of the global vorticity field). It is chosen so that it tightly includes the vorticity field defined by the "real" particles domain. In practice, as shown in Fig. 4, a "buffer" zone of a few h (typically $5h$) is added in the "open-domain" directions to ensure that the VIC grid is not changed too often. As already mentioned, an "overlapping" zone is also added on all "subdomain interfaces".

(2) Interpolate the particle's strengths onto the grid, to obtain the vorticity field $\boldsymbol{\omega}$ on the grid. The M_4' high order interpolation scheme is used (see Section 2.3). As the stencil width of this scheme is $4h$, one notes the utility of the "phantom" particles, located in the region $2h$ outside of the grid limits, to properly obtain the interpolated vorticity field everywhere on the grid.

(3) Obtain the Dirichlet boundary conditions for $\boldsymbol{\psi}$ on the sides of the VIC grid, using the PFM method. The oct-tree and outer multipole expansions are computed using the set of grid points (i.e., the grid's strength previously interpolated) located in the "real" subdomain of each processor.

(4) Solve, locally on each processor, the Poisson equation $\nabla^2\boldsymbol{\psi} = -\boldsymbol{\omega}$ on the VIC grid using a fast Poisson solver (e.g., public domain libraries, such as Fishpack [1] and Mudpack [2]). The solver uses the grid with $\boldsymbol{\psi}$ provided on the boundaries and with $-\boldsymbol{\omega}$ known inside. It returns the $\boldsymbol{\psi}$ field on the grid. In 3-D, the Poisson solver is called three times (as $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ are vector fields).

(5) Evaluate the velocity field, $\mathbf{u}$, from $\boldsymbol{\psi}$, using finite differences (here second order, one-sided on the boundaries and centered elsewhere).

(6) Evaluate the evolution (without convection) of the vorticity field on the grid, $\mathrm{d}\boldsymbol{\omega}/\mathrm{d}t = \nabla_h \cdot (\mathbf{u}\boldsymbol{\omega}) + \nu\nabla_h^2\boldsymbol{\omega}$:

- The diffusion term, $\nu\nabla_h^2\boldsymbol{\omega}$, is here evaluated using a second order scheme using the 27 points of a cube (in 2-D, the 9 points of a square), see Section 2.6. These "isotropic" schemes have the advantage that their truncation error is independent of the local grid orientation (as opposed to the classical "cross" schemes).
- The stretching term, $\nabla_h \cdot (\mathbf{u}\boldsymbol{\omega})$, is also evaluated on the grid using finite differences (also second order, one-sided on the boundaries and centered elsewhere). It is interesting to note that using the conservative form of the stretching term ($\nabla \cdot (\mathbf{u}\boldsymbol{\omega})$) leads to better results (i.e., moments, energy and zero vorticity divergence conservation) than using the non-conservative form ($(\nabla\mathbf{u}) \cdot \boldsymbol{\omega}$).

At this stage, the fields $\mathbf{u}$ and $\mathrm{d}\boldsymbol{\alpha}/\mathrm{d}t = h^3\mathrm{d}\boldsymbol{\omega}/\mathrm{d}t$ are known at each grid point. It remains to "send back" the information (i.e., $\mathbf{u}$ and $\mathrm{d}\boldsymbol{\alpha}/\mathrm{d}t$) to the "real" particles and update their position and strength. The "phantom" particles are here deleted as they are no longer useful.

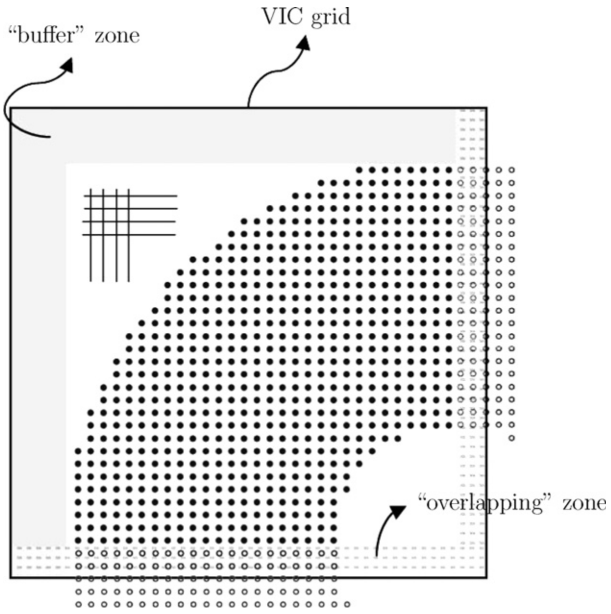


Fig. 4. Illustration of the Lagrangian and Eulerian "configuration" inside a processor.

(7) Update, still locally on each processor, the "real" particles' position and strength:

- In the case particle redistribution has been done prior to the time step, the particles are aligned with the grid. So, no interpolation is needed: $\mathbf{u}$ and $\mathrm{d}\boldsymbol{\alpha}/\mathrm{d}t$ are simply copied from the grid points to the corresponding particles. We then use the second order Runge–Kutta scheme (i.e., Euler as first step predictor and trapezoidal rule as second step corrector), as we do not have the information at the previous time step. The Euler predictor is

$$\mathbf{x}_p^{n+1,*} = \mathbf{x}_p^n + \Delta t\ \mathbf{u}_p\Big|_{(\mathbf{x}^n,\boldsymbol{\alpha}^n)} \tag{6}$$

$$\boldsymbol{\alpha}_p^{n+1,*} = \boldsymbol{\alpha}_p^n + \Delta t\ \frac{\mathrm{d}\boldsymbol{\alpha}_p}{\mathrm{d}t}\bigg|_{(\mathbf{x}^n,\boldsymbol{\alpha}^n)} \tag{7}$$

  The trapezoidal rule corrector first requires redoing steps (1)–(6). Moreover, as the particles are no longer aligned with the grid, the grid fields $\mathbf{u}$ and $\mathrm{d}\boldsymbol{\omega}/\mathrm{d}t$ are interpolated from the grid to the particles (using again M_4'). We see here the interest of the minimum $2h$ overlapping zone that permits to correctly interpolate the information from the grid to the particles located at the border of the "real" subdomain. After that, the correction step is as follows:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{1}{2}\Delta t\Big(\mathbf{u}_p|_{(\mathbf{x}^n,\boldsymbol{\alpha}^n)} + \mathbf{u}_p|_{(\mathbf{x}^{n+1,*},\boldsymbol{\alpha}^{n+1,*})}\Big) \tag{8}$$

$$\boldsymbol{\alpha}_p^{n+1} = \boldsymbol{\alpha}_p^n + \frac{1}{2}\Delta t\left(\frac{\mathrm{d}\boldsymbol{\alpha}_p}{\mathrm{d}t}\bigg|_{(\mathbf{x}^n,\boldsymbol{\alpha}^n)} + \frac{\mathrm{d}\boldsymbol{\alpha}_p}{\mathrm{d}t}\bigg|_{(\mathbf{x}^{n+1,*},\ \boldsymbol{\alpha}^{n+1,*})}\right) \tag{9}$$

- In the case particle redistribution has not been done prior to the time step, $\mathbf{u}$ and $\mathrm{d}\boldsymbol{\omega}/\mathrm{d}t$ are interpolated from the grid, as for the second substep of the Runge–Kutta scheme. Now, as we have the information from the previous time step, we use second order two-step schemes. For $\mathrm{d}\boldsymbol{\alpha}/\mathrm{d}t$, we use Adams-Bashforth. For particle convection, we use Leap-Frog:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^{n-1} + 2\Delta t\mathbf{u}_p\Big|_{(\mathbf{x}_p^n,\boldsymbol{\alpha}_p^n)} \tag{10}$$

$$\boldsymbol{\alpha}_p^{n+1} = \boldsymbol{\alpha}_p^n + \frac{1}{2}\Delta t\left(3\frac{\mathrm{d}\boldsymbol{\alpha}_p}{\mathrm{d}t}\bigg|_{(\mathbf{x}_p^n,\boldsymbol{\alpha}_p^n)} - \frac{\mathrm{d}\boldsymbol{\alpha}_p}{\mathrm{d}t}\bigg|_{(\mathbf{x}_p^{n-1},\boldsymbol{\alpha}_p^{n-1})}\right) \tag{11}$$

(8) Finally, the particles are distributed among the processors according to their new position. If the vorticity field has grown and/or moved, the computational VIC domain and its subdomains limits are redefined.

It is interesting to note that communications are needed only twice per time step: when the boundary conditions are determined using the PFM method (3) and during the particles distribution (8). It leads to a good speedup, in fact nearly the same as the speedup of the PFM method used alone but, as we will see, with a better global computational time.

We should also mention that, in order to limit the number of particles, a "cut" is done every time step. This operation consists in removing the particles having a strength weaker than a fraction of the maximum strength present in the flow. In the case where strong stretching occurs, the maximum strength can rapidly grows up (and thus leading to a too important cut). In that case, the maximum strength taken into account is the maximum strength of the initial condition. Typically, the limit of the cut is taken as $10^{-5}\|\boldsymbol{\alpha}\|_{\max}$. To ensure the total vorticity conservation, the sum of the "cut" vorticity is uniformly distributed to the remaining particles.

2.3. Particle redistribution/interpolation scheme

As already mentioned, Lagrangian methods require particle redistribution, see [9,40]. It is an essential operation to maintain a good overlap of the particle regularization functions, and thus maintain a good representation of the vorticity field. In the VIC–PFM method, as we continuously go between the particles and the grid, we also need a good quality scheme to bring the particles information to the grid, and the grid information to the particles. The scheme chosen is the M'_4 scheme [23] (the Λ_3 scheme was also studied but, in 3-D, some noise was noticed, see also [9]). In 1-D, the M'_4 scheme is as follows:

$$M'_4(X) = \begin{cases} \frac{1}{2}(2 - 5X^2 + 3X^3) & \text{if } 0 \leqslant X \leqslant 1 \\ \frac{1}{2}(1 - X)(2 - X)^2 & \text{if } 1 < X \leqslant 2 \\ 0 & \text{otherwise} \end{cases}$$

where X is the distance between the particle and the grid node, normalized by h. The fraction of the particle information which is attributed to the grid node (or conversely) is given by $M'_4(X)$. In 2-D and 3-D, one uses tensor products.

The M'_4 scheme is continuous and so is its first derivative. It is a second order scheme: it conserves $\mathbf{\Omega}$, $\mathbf{I}$ and $\mathbf{A}$. Of course, the use of such scheme introduces some dissipation. One can show that, on average, it acts as a hyper-viscosity. For a $h/2$ shift between particle positions and grid (which represents the worst situation), it corresponds to a hyper-viscosity operator discretized to second order and with effective $\nu^{(2)}\Delta t$ equal to $3h^4/128$. Considering all possible shifts statistically, the net hyper-viscosity obtained is only a fraction of that.

2.4. Projection scheme used to maintain zero vorticity divergence

In 3-D, vortex particle methods do not guarantee that the divergence of the vorticity field remains zero in long-time simulations and/or in under-resolved simulations. In the VIC method (thus also in the VIC–PFM method), a good way to prevent any possible bad behavior consists in regularly re-projecting the vorticity field onto a divergence free basis [12,29]:

$$\boldsymbol{\omega}^{\text{new}} = \boldsymbol{\omega} - \nabla_h F \tag{12}$$

where F is obtained by solving

$$\nabla_h^2 F = \nabla_h \cdot \boldsymbol{\omega} \tag{13}$$

on the grid using, again, the efficient Poisson solver, with Dirichlet boundary condition $F = 0$. This operation is done as soon as the vorticity divergence $\nabla_h \cdot \boldsymbol{\omega}$ is seen to exceed a prescribed value, and when the particles are on a regular lattice (i.e., after a redistribution). The new vorticity field is such that $\nabla_h \cdot \boldsymbol{\omega}^{\text{new}} = 0$ and the particle strengths are reset to $\boldsymbol{\alpha}^{\text{new}} = \boldsymbol{\omega}^{\text{new}} h^3$. This operation provides an efficient and essentially non-diffusive "relaxation" scheme, which allows for simulations at high Reynolds number and for arbitrarily long times.

2.5. Subgrid-scale modelling for LES

The aim of this new VIC–PFM method is also to be able to simulate complex 3-D flows at high Reynolds number, thus also using Large-Eddy Simulation (LES) approaches. In this case, only the most energetic scales are captured and the effect of the subgrid-scales (SGS) on the captured scales is modelled.

The capabilities of different SGS models for LES, here implemented in the VIC method, have been investigated in [8,10,15–17,37,39]. Their applicability to the simulation of decaying homogeneous isotropic turbulence (HIT) was also tested. In [8], various SGS models were tested and compared on a more complex and transitional flow: the multiscale dynamics developing in a counter-rotating four-vortex system that evolves from a simple state to a turbulent state. Here, we use a simplified version of the regularized variational multiscale model (here called RVM), where the diffusion, using an effective eddy viscosity ν_{sgs}, operates solely on the high frequency part of the LES vorticity field. In the vorticity–velocity formulation, this gives:

$$\frac{D\boldsymbol{\omega}}{Dt} = \nabla \cdot (\mathbf{u}\boldsymbol{\omega}) + \nu \nabla^2 \boldsymbol{\omega} + \nabla \cdot (\nu_{\text{sgs}}(\nabla \boldsymbol{\omega}^{\text{s}} + (\nabla \boldsymbol{\omega}^{\text{s}})^{\text{T}})) \tag{14}$$

with $\boldsymbol{\omega}^{\text{s}}$ the "small-scale" vorticity field and $\nu_{\text{sgs}} = C_R^{(n)} \Delta^2 (2 S_{ij} S_{ij})^{1/2}$ the subgrid viscosity. The LES cut-off length Δ is taken equal to the grid size h. The coefficients $C_R^{(n)}$ are taken from [8]. S_{ij} is the strain rate tensor:

$S_{ij} = \frac{1}{2}(g_{ij} + g_{ji})$ with $g_{ij} = \partial u_i/\partial x_j$. The rhs of Eq. (14) is discretized as presented in Section 2.6. Here the small-scale fields are obtained using a compact (stencil 3) tensor-product explicit discrete filter which is iterated n times to produce an order $2n$ filter, see [17] for more details, and also [16,39].

In some simulations, done prior to the RVM model, we also performed LES using a simple "hyper-viscosity" SGS dissipation model. It was also first chosen for its simplicity of implementation and its computational efficiency. The LES equations are then taken as

$$\frac{D\boldsymbol{\omega}}{Dt} = \nabla \cdot (\mathbf{u}\boldsymbol{\omega}) + \nu\nabla^2\boldsymbol{\omega} - \frac{C}{T_0}(h^2\nabla^2)^2\boldsymbol{\omega} \tag{15}$$

with T_0 a global time scale and C a coefficient. The $(h^2\nabla_h^2)^2$ operator is evaluated by applying twice the discrete Laplacian of Section 2.6.

The global and spectral behavior of these models, as well as others, are discussed in detail in [17]. The present brief presentation is sufficient for the examples of VIC–PFM applications presented here.

2.6. Laplacian schemes that are less sensitive to the grid orientation

We here present the 2-D and 3-D "isotropic" Laplacian schemes for which the performance is less dependant of the local grid orientation than the classical "cross" scheme. They were designed so that the truncation error is itself a Laplacian: see Fig. 5 for the coefficients. It is easily verified, using Taylor series, that we then obtain

$$(\nabla_h^2\boldsymbol{\omega})_{i,j,k} = \left(\nabla^2\left(\boldsymbol{\omega} + \frac{h^2}{12}\nabla^2\boldsymbol{\omega}\right)\right)_{i,j,k} + \mathcal{O}(h^4) \tag{16}$$

The leading truncation error term is thus $(h^2/12)$ $(\nabla^4\boldsymbol{\omega})_{i,j,k}$ which is not sensitive to the orientation of the grid with respect to the solution $\nabla^2\boldsymbol{\omega}$. This is not the case when using the usual cross scheme.

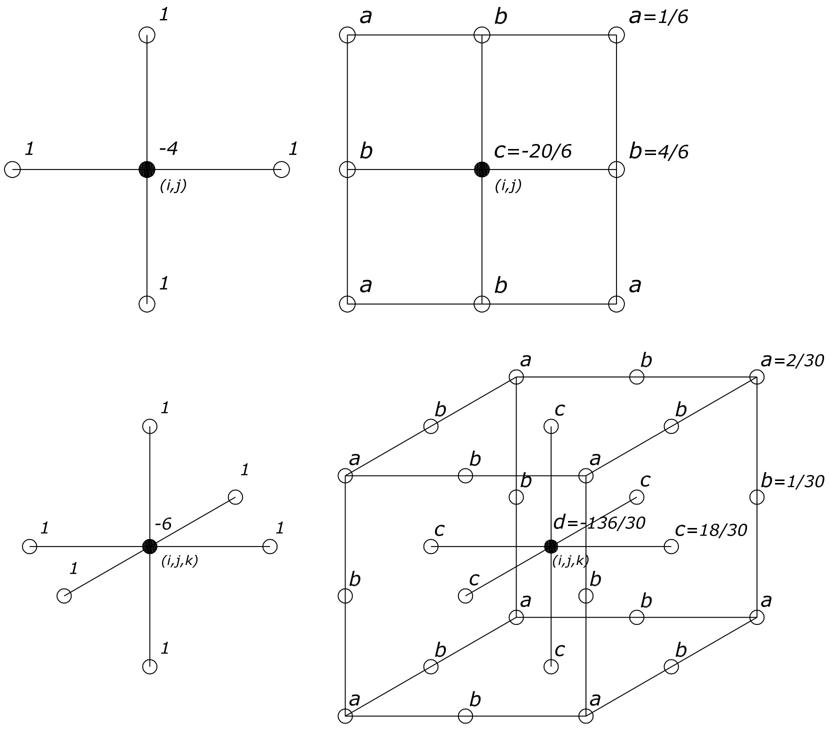


Fig. 5. Illustration of 2-D and 3-D second order Laplacian schemes: the "cross" schemes (left), and the "isotropic" schemes (right).

For LES, the RVM model (rhs of Eq. (14)) is programmed as follows:

$$h^2\nabla\cdot(\nu^{\mathrm{sgs}}(\nabla_h\boldsymbol{\omega}^{\mathrm{s}}+(\nabla_h\boldsymbol{\omega}^{\mathrm{s}})^{\mathrm{T}})) = \frac{2}{30}\left[\frac{(\nu^{\mathrm{sgs}}_{i+1,j+1,k+1}+\nu^{\mathrm{sgs}}_{i,j,k})}{2}(\boldsymbol{\omega}^{\mathrm{s}}_{i+1,j+1,k+1}-\boldsymbol{\omega}^{\mathrm{s}}_{i,j,k})+\cdots\right]+\frac{1}{30}[\cdots]+\frac{18}{30}[\cdots] \tag{17}$$

2.7. Beyond purely open-domain flows

There is a large class of purely open-domain flows which can be solved efficiently using the VIC–PFM method as such. We here describe how the method can be extended to further address problems with one periodic direction, problems bounded by a ground, or problems evolving jointly in space and time (space-developing flows).

2.7.1. Case of flows with one periodic direction

As we will illustrate in Section 4.2 using an aircraft vortex wake simulation, the VIC–PFM method is also implemented with one periodic direction. This allows to determine the $\boldsymbol{\psi}$ field on the four faces of the grid corresponding to the unbounded directions.

When the code is used on a single processor or when the domain is not split in the periodic direction, the Poisson solver used (here FishPack [1]) is able to take into account one direction with periodic condition and the two others with Dirichlet conditions. A fast and optimal way to determine the $\boldsymbol{\psi}$ field is illustrated in Fig. 6. To determine the $\boldsymbol{\psi}$ field at a point P, one has to compute the influence of the main box Ω (i.e., the computational domain) and also of a series of equivalent boxes repeated periodically on both sides of the main box. We use the fact that the influence of the box $\Omega_{1_{\mathrm{left}}}$ on the point P is exactly the same as the influence of the main box Ω on the point $P_{1_{\mathrm{right}}}$, and conversely; and so on for all points on both sides of Ω. In this way, the $\boldsymbol{\psi}$ field at P can be computed by summing the contributions of Ω on P, $P_{i_{\mathrm{left}}}$ and $P_{i_{\mathrm{right}}}$. Only the multipole expansions for the computational domain Ω are required. Experience has shown that taking 50 image points on each side provides a well converged solution (i.e., a periodic solution). Compared with the non-periodic case, this periodic evaluation has an additional computational cost of only about 25%. It is thus still very acceptable.

In the parallel case (i.e., when the domain Ω is itself split in the periodic direction in various subdomains), the Poisson equation is solved in two steps to ensure a perfect continuity between each subdomain:

$$\begin{cases}\nabla^2\overline{\boldsymbol{\psi}} = -\overline{\boldsymbol{\omega}}\\ \nabla^2\boldsymbol{\psi}' = -\boldsymbol{\omega}'\end{cases}$$

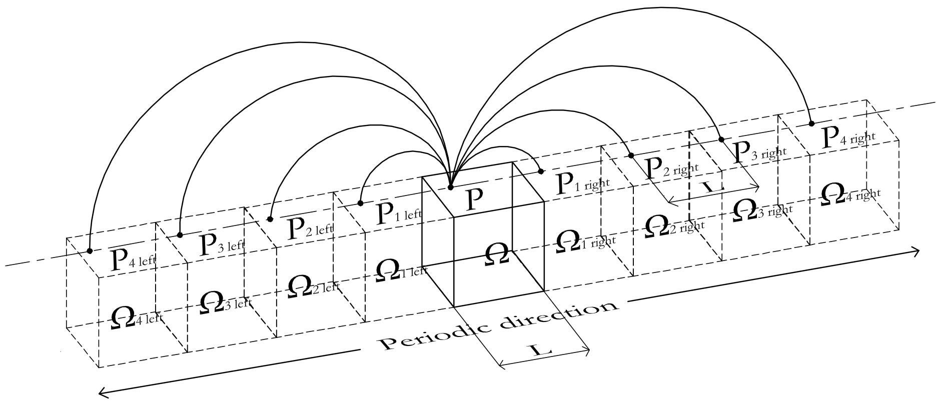


Fig. 6. Illustration of the PFM method used to compute flows with one periodic direction. The $\boldsymbol{\psi}$ field at P is obtained by summing the contributions of Ω on the "shifted" points.

where $\overline{\boldsymbol{\varpi}}$ is the 2-D vorticity "base flow" field (i.e., the vorticity field space averaged in the periodic direction) and $\boldsymbol{\omega}' = \boldsymbol{\omega} - \overline{\boldsymbol{\varpi}}$. The base flow is first evaluated on the grid and is used to obtain the boundary conditions $\overline{\boldsymbol{\psi}}$ required to solve the first Poisson equation. Then, the $\boldsymbol{\omega}'$ field is evaluated and is used to obtain the related $\boldsymbol{\psi}'$ boundary condition to solve the second Poisson equation. The total streamfunction $\boldsymbol{\psi}$ is then simply the sum of $\overline{\boldsymbol{\psi}}$ and $\boldsymbol{\psi}'$.

In that way, we impose that each processor in the periodic direction "sees" exactly the same base flow field. Without such procedure, we have noticed, in long-time simulations, the possible apparition of some small discontinuities at the subdomain interfaces. This behavior is due to the method used to compute the periodic boundary conditions. Indeed, even if 50 image points are sufficient to have a well converged periodic solution, it does not guarantee that the solution obtained on the ends of the periodic domain are exactly the same, the multipole summations being not exactly equivalent. Imposing the same base flow field all over the periodic domain avoids this problem and leads to a continuous periodic solution.

2.7.2. Case of flows in semi-open domains

As we have already mentioned, the PFM method is ideal to take into account boundary conditions for open-domain flows, also those with one periodic direction. However, in some cases (e.g., flows bounded by a flat ground, space-developing simulations with inflow and outflow, etc.), we also may want to impose:

- *A no-through flow plane*: this is used for bounded flows where we need to enforce the no-through flow condition on a flat plane. For that, we symmetrize the vorticity field with respect to the plane (here written for a x–y plane located at $z = 0$):

  $$\begin{cases} \omega_n(x,y,-z) = \omega_n(x,y,z) \\ \omega_t(x,y,-z) = -\omega_t(x,y,z) \end{cases}$$

  where ω_n and ω_t are respectively the normal component and the tangential components of the vorticity field. This leads to $u_n(x,y,0) = 0$.
- *A through flow plane*: this is used for outflow where the tangential velocity is enforced to zero. This is also done by enforcing symmetries on the flow with respect to a flat plane. It leads to the following constraints on the vorticity field (also written for a x–y plane located at $z = 0$):

  $$\begin{cases} \omega_n(x,y,-z) = -\omega_n(x,y,z) \\ \omega_t(x,y,-z) = \omega_t(x,y,z) \end{cases}$$

  This leads to $u_t(x,y,0) = 0$.

In practice, we do not have to build the symmetrized vorticity field explicitly: when one evaluates, using the PFM method, the $\boldsymbol{\psi}$ field at a point $\mathbf{x}$ (for the boundary conditions) induced by the vorticity field, one also makes evaluations at symmetric locations and adds it to the first contribution, changing adequately the sign of each component. In the case of more complex bodies than flat planes, boundary element methods are then used (see [28,40]).

In order to simulate a flow above a viscous ground, we also need to enforce a zero slip velocity at the wall. It is done in two steps. First, we use the "no-through flow" symmetrization to enforce a zero normal velocity over the whole ground plane. After the Poisson equation has been solved, there is however still a spurious slip velocity, $\mathbf{u}_{\mathrm{slip}}$, at the wall. The second step is then to enforce the no-slip by emitting a vorticity flux at the wall:

$$\nu \frac{\partial \boldsymbol{\omega}}{\partial \mathbf{n}} = \frac{\Delta \boldsymbol{\gamma}}{\Delta t}, \qquad \Delta \boldsymbol{\gamma} = 2\mathbf{u}_{\mathrm{slip}} \tag{18}$$

Since the ground is flat, there is no need to solve numerically a boundary integral equation for the panel strengths: the solution is trivially obtained as $\Delta \boldsymbol{\gamma} = 2\mathbf{u}_{\mathrm{slip}}$. This flux must be emitted during a time Δt, see [9,18,28,40]. As this operation is done on the VIC grid, we have square panels of strength $\Delta \boldsymbol{\gamma}$ and size $h \times h$. We here consider that the ground is located on the XY plane and that the panels diffuse towards the positive Z direction. The "amount" of strength diffused, from one panel, to a grid point located at $(x, y, z > 0)$, during a time Δt, is given by:

$$\Delta\boldsymbol{\alpha} = \int_0^{\Delta t} \frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}t}\,\mathrm{d}t \tag{19}$$

with

$$\frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}t}(x,y,z) = \int_{x-h/2}^{x+h/2}\int_{y-h/2}^{y+h/2}\int_{z-h_l/2}^{z+h/2} \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x \tag{20}$$

$$\begin{aligned} &= \frac{\Delta\boldsymbol{\gamma}}{\Delta t}\left([\mathrm{erfc}(u)]_{(z+h/2)/\sqrt{4\nu t}}^{(z-h_l/2)/\sqrt{4\nu t}}\right)\left(\sqrt{4\nu t}\,\frac{1}{2}\left([\mathrm{ierfc}(u)]_{(x-h)/\sqrt{4\nu t}}^{x/\sqrt{4\nu t}} - [\mathrm{ierfc}(u)]_{x/\sqrt{4\nu t}}^{(x+h)/\sqrt{4\nu t}}\right)\right) \\ &\times \left(\sqrt{4\nu t}\,\frac{1}{2}\left([\mathrm{ierfc}(u)]_{(y-h)/\sqrt{4\nu t}}^{y/\sqrt{4\nu t}} - [\mathrm{ierfc}(u)]_{y/\sqrt{4\nu t}}^{(y+h)/\sqrt{4\nu t}}\right)\right) \end{aligned} \tag{21}$$

where $\mathrm{ierfc}(u) = \int_u^\infty \mathrm{erfc}(v)\,\mathrm{d}v = \frac{1}{\sqrt{\pi}}\exp(-u^2) - u\,\mathrm{erfc}(u)$. Notice that $h_l/2 = h$ for the "first layer" of grid points (i.e., located at $z = h$) and $h_l/2 = h/2$ otherwise. The time integral in Eq. (19) is evaluated numerically using a Gauss quadrature (4 points). The vorticity is generally distributed on a "region" of size $7h \times 7h \times 4h$ centered on the panel, so that we ensure that the whole panel strength is distributed. As this operation is done on the grid, most of the terms of Eq. (21) can be precomputed for all panels. The no-slip condition is enforced at each time step, just after step (6).

When the divergence is reset to zero, using the projection scheme presented before, the boundary conditions for the resolution of the Poisson equation also need to be changed. At the wall, we impose Neumann boundary conditions, $\partial F/\partial n = 0$, so that the normal vorticity remains zero: $\omega_z^{\mathrm{new}} = \omega_z - \partial F/\partial z = \omega_z = 0$.

2.7.3. Case of space-developing simulations

The setup for space-developing simulations is presented in Fig. 7. It essentially consists on an inflow condition that dynamically evolves in space and time, due to the vorticity and the induced velocity of an added potential flow ϕ (and maybe time dependent).

In order to "close" the problem, a fixed vorticity field is added in front of the inflow plane to close the vortex lines coming from the "main" vorticity domain. The injection of the entry condition is ensured by buffer particles, located in front of the entry plane and convected with the velocity computed on the inflow plane. As soon as their position is inside the main vorticity domain, they become "real" particles. As we do not want to keep the vortex wake particles "forever", an outflow boundary condition is also used; it is imposed by enforcing symmetries on the flow.

When the boundary conditions are obtained using the PFM method, the set of particles representing the fixed vorticity field is used, in addition to the particles from the main vorticity domain. Then, the symmetrizations previously presented are used to impose the correct outflow conditions and/or the presence of a ground, as shown in Fig. 7. The choice of the outflow conditions is dictated by the vorticity field. If the normal

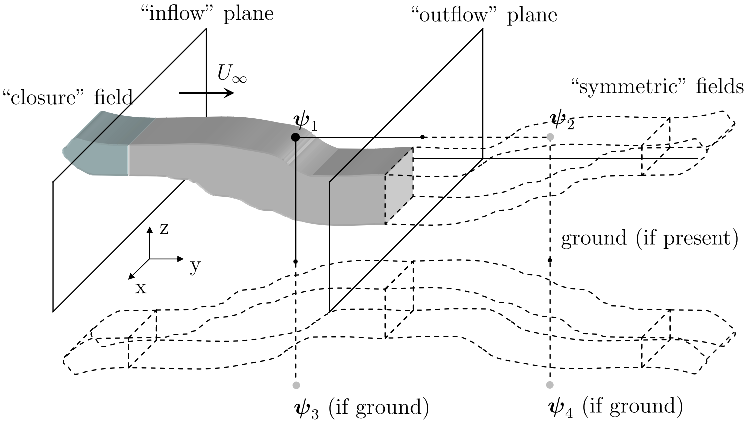


Fig. 7. Illustration of the configuration used for space-developing simulations using the VIC–PFM method.

component of the vorticity distribution is dominating (e.g., an aircraft wake vortex system), the “no-through flow” condition is used. It means that the normal velocity obtained from the Poisson equation is set to zero, and thus that the total normal velocity on the outflow plane is equal to the velocity of the added potential flow. Conversely, if the tangential components of the vorticity distribution is dominating (e.g., a shear layer flow, or a jet flow), the “through flow” condition is used. The velocity field on the outflow plane is then purely normal (unless the added potential flow also has a tangential component).

3. Efficiency analysis of the VIC–PFM method

The aim of this section is to analyse the CPU time cost and the memory requirements for the VIC–PFM method, as compared to the PFM method alone (which also solves the unbounded flow case) and to the VIC method alone (which here solves a periodic flow problem). To make the comparison, we consider a generic flow: a vortex ring of centerline radius R, of Gaussian vorticity distribution as in Eq. (28), and of core size $\sigma_0/R = 0.21$. The discretization is $h/R = 0.02$, the number of particles is $\sim 7.8 \times 10^5$ and the VIC–PFM grid is $166 \times 166 \times 64$ ($\sim 1.8 \times 10^6$ grid points).

Table 1 provides, for one time step, the computational times and the memory requirements obtained using the different methods, running on one processor. For the periodic VIC method alone, three different grid sizes are considered: they are illustrated in Fig. 8.

First, one notices that the VIC–PFM method is 17 times faster than the PFM method alone. We clearly see the interest to combine the two methods. Both solvers are $C * \mathcal{O}(N\log(N))$, but the Poisson solver used in the VIC method has a much smaller C than the PFM method. As we use the PFM method only for obtaining $\boldsymbol{\psi}$ at

Table 1
Efficiency analysis: CPU times (computations ran on one Intel Pentium IV processor at 2.4 GHz) and memory requirements (based on the storage of $\mathbf{u}$, $\boldsymbol{\psi}$ and $\boldsymbol{\omega}$ in double precision) for the hybrid VIC–PFM method, the PFM method alone, and the periodic VIC method alone (for three grids)

	VIC–PFM	PFM	VIC (1)	VIC (2)	VIC (3)
L/D	1.66	–	1.66	2.28	3.20
H/D	0.64	–	0.64	1.28	1.92
Grid size	166 × 166 × 64	–	166 × 166 × 64	228 × 228 × 128	320 × 320 × 192
CPU times	55.0 s	926.0 s	12.5 s	24.7 s	72.0 s
Memory requirements	172 MB	52 MB	120 MB	456 MB	1350 MB

the boundary grid points, we are able to save a lot of CPU time and finally obtain a large factor (17) between the two methods.

It is interesting to measure where the computational time is spent in this hybrid method: roughly half of the time is devoted to the determination of the proper ψ boundary conditions using the PFM method, a quarter is devoted to the grid Poisson solver, and a quarter to the interpolation operations between the particles and the grid; the other operations (finite differences for velocity, stretching term, particles distribution among the processors, etc.) have a negligible cost.

We also compare with the VIC method alone, assuming periodic conditions. One clearly sees, in Fig. 8, that the Cases 1 and 2 are still very far from representing a vortex ring in unbounded space. Case 3 could be considered as closer to an unbounded flow, but the memory requirements then become very large.

Besides the efficiency, the accuracy of the results between the different cases was also considered. The limitation was that the VIC method solves a full periodic problem whereas the PFM and the VIC–PFM methods solve an open-domain problem. A reference calculation was first performed using the PFM method with a very low error criteria. The efficiency of the VIC–PFM and the PFM methods was then compared using the same accuracy criteria. For the VIC method, we are not able to reach the same accuracy because the grid that would be required to obtain the equivalent open-domain condition would be too large. To make the comparison as fair as possible, we consider Case 3 closer to an unbounded flow, even if the solution is still obviously polluted by the periodicity. One can also noticed that the Poisson solver and the methodology used in the VIC and the VIC–PFM methods are exactly the same, and so their accuracy, but applied on a problem with different boundary conditions. It is thus mainly important to ensure that the accuracy criteria is equivalent between the VIC–PFM and the PFM methods, which solve the same problem.

Another interesting point is the speedup performance of the VIC–PFM implementation. Two cases are illustrated to distinguish the gain obtained with two different scalings: the weak scaling, where the problem size is doubled when the number of processors is doubled, and the strong scaling, where the problem size is held fixed. When a $\mathcal{O}(N)$ method is used, a perfect parallelization would lead to a speedup equal to the number of processors. Here, the method is the combination of two $\mathcal{O}(N\log(N))$ methods, the best speedup that can be obtained is thus not as good. In [11], the computational cost of the VIC method is fitted with $\mathcal{O}(N^a)$, where $a \approx [1.10–1.15]$. In the range of N classically used (i.e., $N \in [10^4–10^8]$), both writings are essentially equivalent: $\mathcal{O}(N\log(N)) \approx \mathcal{O}(N^a)$ for $a \approx [1.11–1.15]$. Here, we used the $\mathcal{O}(N\log(N))$ fit.

3.1. Weak scaling

For the weak scaling, the computational time using one processor and p processors are respectively:

$$T_1 = C_P 6N^{2/3}\log(rN) + C_V N\log(N) \tag{22}$$

$$T_p = C_P 6N^{2/3}\log(prN) + C_V N\log(N) \tag{23}$$

where C_P and C_V are the coefficients related to the PFM method and to the VIC method, r is a “filling” coefficient that gives the ratio between the number of particles and the number of grid points (typically $r \approx 0.5$) and N is the number of grid points, here with a grid assumed of cubical shape. The first term in the rhs of Eq. (23) is the cost of the boundary conditions computed on the 6 faces of the grid (composed of $N^{2/3}$ grid points) using the prN particles. The second term is the cost of the Poisson equation resolution: for the weak scaling, it remains constant with the number of processors. The best reachable speedup is then given by

$$S_{\max}(p) = p\frac{T_1}{T_p} = p\frac{1}{1 + \frac{\log(p)}{\log(r) + \left(1 + \frac{N^{1/3}}{6(C_P/C_V)}\right)\log(N)}} \tag{24}$$

The curve is shown in Fig. 9 and is compared with the speedup effectively obtained with a simulation on a single domain size of $N = 1.2 \times 10^6$, $r \approx 0.5$ and with $C_P/C_V = 100$. This last value is deduced from sequential simulations and is in good agreement with [11]. The speedup obtained is good: even with 128 processors and more than 150 million grid points, the efficiency is still very acceptable: $\epsilon(p) = S(p)/S_{\max}(p) \approx 0.63$.

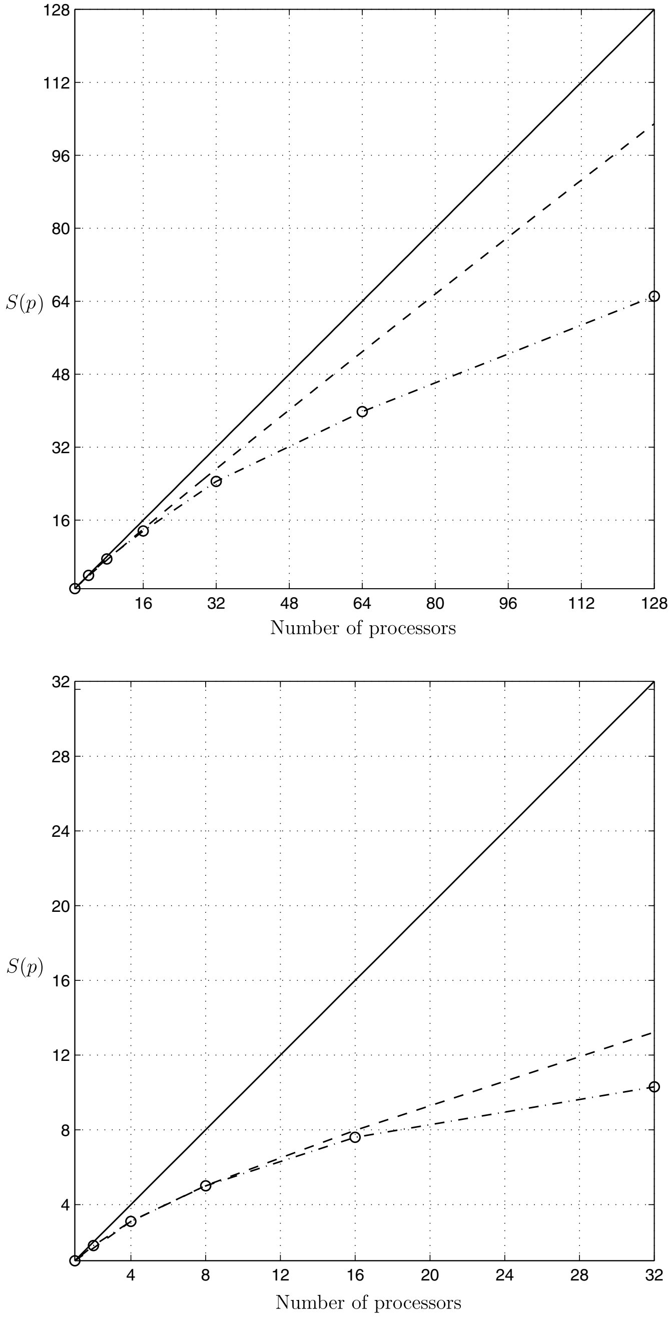


Fig. 9. Speedup performance for the weak scaling (top) and for the strong scaling (bottom): the perfect parallelization of a $\mathcal{O}(N)$ method (solid), the best reachable parallelization of a VIC–PFM method (dashed) and the speedup obtained with our code (dash-dot).

3.2. Strong scaling

For the strong scaling, the computational time using one processor and p processors are here respectively:

$$T_1 = C_{\mathrm{P}} 6 N^{2/3} \log(rN) + C_{\mathrm{V}} N \log(N) \tag{25}$$

$$T_p = C_{\mathrm{P}} A(p) N^{2/3} \log(rN) + C_{\mathrm{V}} \frac{N}{p} \log\left(\frac{N}{p}\right) \tag{26}$$

where $A(p)$ is the coefficient taking into account the variation of the number of grid points where the boundary conditions have to be computed (depending on the number of processors used). Again, the first term in the rhs of Eq. (26) is the computational cost related to the boundary conditions evaluated using the rN particles. The second term is the cost of the Poisson equation resolution. Here, the grid is split between the p processors and becomes smaller when p increases. The best reachable speedup is then given by

$$S(p) = \frac{T_1}{T_p} = \frac{(C_\mathrm{P}/C_\mathrm{V})6\log(rN) + N^{1/3}\log(N)}{(C_\mathrm{P}/C_\mathrm{V})A(p)\log(rN) + \frac{N^{1/3}}{p}\log\left(\frac{N}{p}\right)} \tag{27}$$

The curve is illustrated in Fig. 9. The real speedup is here obtained using a simulation with $N = 4.8 \times 10^6$ and $r \approx 0.5$. We observe that, up to 16 processors, both speedup curves collapse very well. With 32 processors, the real curve saturates because the grid allocated to each processor becomes too small: no significant further gain can then still be obtained.

We can finally conclude that the coupling of the PFM and VIC methods provides an efficient algorithm for truly unbounded flow problems, with good computational efficiency and much acceptable memory requirements.

4. Validations and illustrations

4.1. Instability of vortex rings at moderate and high Reynolds numbers

The first application we consider is the instability of vortex rings. The objective is not to carry a detailed study on vortex rings instabilities, but to show that the VIC–PFM method is competitive for this type of flow because of its Lagrangian framework (low dispersion) and its unbounded domain capacity. We first examine a configuration that was previously studied by Shariff et al. [36]: a vortex ring at moderate Reynolds number, where we examine the most amplified waves and their growth rate. Here, we will also simulate the complete non-linear phase, up to the saturation of the instabilities, and also capture the following global decay. The second case is far more challenging: a much thinner vortex ring and at higher Reynolds number. We will also examine the most amplified waves, their growth rate and compare them to those predicted using Saffman's theory.

The initial azimuthal vorticity distributions are Gaussian, function of the distance to the ring centerline:

$$\omega_\theta = \frac{\Gamma_0}{\pi\sigma_0^2}\exp(-s^2/\sigma_0^2) \tag{28}$$

with $s^2 = z^2 + (r - R)^2$, Γ_0 the initial circulation, σ_0 the initial core size, and R the ring radius. The time scale used for normalization is taken as $t_0 = R^2/\Gamma_0$. This distribution does not constitute a steady solution, but for core rotation frequencies not small compared to the growth rate, this unsteadiness will act as an intrinsic forcing on the instability waves [35,36]. The initial perturbation is done by displacing the vortex lines radially, from their unperturbed position r to their perturbed r_pert, as follows:

$$r_\mathrm{pert}(\theta) = r(1 + \epsilon g(\theta)) \tag{29}$$

where $g(\theta)$ is a sum of Fourier modes (with unit amplitudes and random phases) and ϵ is the perturbation amplitude. To maintain the zero divergence of the vorticity field after applying the perturbation, ω_θ and ω_r are realigned along the perturbed vortex lines.

The first vortex ring studied has a $\sigma_0/R = 0.4131$ and is perturbed on the 24 first modes with an amplitude $\epsilon = 0.0002$; the Reynolds number is $Re = \Gamma_0/\nu = 5500$ (corresponding to Case 1 in Shariff et al. [36]). The discretization is $h/R = 0.040$ and the time step used is $\Delta\tau = \Delta t/t_0 = 0.05$. To be able to better capture the turbulent decay phase, we add some hyper-viscosity SGS dissipation (with $C = 2.5 \times 10^{-2}$). Without such LES model, one cannot go much further than $\tau = 80$. We also use the projection scheme described in Section 2.4.

The simulation was carried out for 4700 time steps and took ~64 h on a single Pentium IV at 2.4 GHz. The number of particles goes from $\sim 3.6 \times 10^5$ up to $\sim 1.0 \times 10^6$, and the number of grid points grows from $\sim 9.0 \times 10^5$ to $\sim 4.4 \times 10^6$. At $\tau = 100$, a part of the distant wake left by the vortex ring was removed, to be able to continue the simulation on a single processor. Because of the very low energy level of this far wake, the flow around the vortex ring is not affected by this cut.

Before presenting the results, we provide definitions regarding the characterization of the instabilities and the evaluation of the energy diagnostics. Formulating the velocity field in cylindrical coordinates, we can write

$$\mathbf{u}(r,\theta,z,t) = \sum_{n=-N_\theta/2}^{N_\theta/2} \widehat{\mathbf{u}}(r,n,z,t)\mathrm{e}^{\mathrm{i}n\theta} \tag{30}$$

with N_θ the number of azimuthal grid points; a similar development applies to $\boldsymbol{\psi}$ and $\boldsymbol{\omega}$. The azimuthal Fourier transform of $\mathbf{u}$ is given by

$$\widehat{\mathbf{u}}(r,n,z,t) = \frac{1}{2\pi}\int_0^{2\pi} \mathbf{u}(r,\theta,z,t)\mathrm{e}^{-\mathrm{i}n\theta}\,\mathrm{d}\theta \tag{31}$$

The azimuthal modal energy E_n is accordingly defined by

$$E_n(t) = \frac{1}{2}\iint \widehat{\mathbf{u}}\cdot\widehat{\mathbf{u}}^* r\,\mathrm{d}r\,\mathrm{d}z = \frac{1}{4}\iint \left[\widehat{\boldsymbol{\psi}}\cdot\widehat{\boldsymbol{\omega}}^* + \widehat{\boldsymbol{\omega}}\cdot\widehat{\boldsymbol{\psi}}^*\right] r\,\mathrm{d}r\,\mathrm{d}z \tag{32}$$

with superscript * denoting the complex conjugate. Using the symmetry ($E_n = E_{-n}$), we consider only the modes with $n \geqslant 0$. Accordingly, the modal energy becomes

$$E_n(t) = \iint \widehat{\mathbf{u}}\cdot\widehat{\mathbf{u}}^* r\,\mathrm{d}r\,\mathrm{d}z = \frac{1}{2}\iint \left[\widehat{\boldsymbol{\psi}}\cdot\widehat{\boldsymbol{\omega}}^* + \widehat{\boldsymbol{\omega}}\cdot\widehat{\boldsymbol{\psi}}^*\right] r\,\mathrm{d}r\,\mathrm{d}z \tag{33}$$

The total energy is then

$$E(t) = \sum_{n=0}^{N_\theta/2} E_n(t) \tag{34}$$

The modal energy $E_n(t)$ is here computed using the stream-vorticity fields. At the highest wavenumbers, a difference is noticed with the energy computed using the velocity field. It is due to the truncation error introduced when the velocity is computed from the stream field using finite differences. Fig. 10 shows the evolution of the modal energies E_n. One can observe different phases: the transient phase for $\tau \leqslant 25$; the linear phase, when linearly fastest growing eigenmodes begin to dominate over transients ($25 \leqslant \tau \leqslant 80$); the non-linear stage, when highest wavenumber modes begin to grow more rapidly than the first eigenmodes ($80 \leqslant \tau \leqslant 150$); the saturation which occurs at $\tau \approx 150$; and finally, the global decay for $\tau \geqslant 150$. The most amplified modes during the linear phase are $n = 6$, $n = 7$, and, a little less, $n = 8$. Shariff et al. found the $n = 5$, $n = 6$ and $n = 7$ modes. In our simulation, the $n = 7$ mode is dominant whereas it was the $n = 6$ in Shariff's simulation. But, if we compare the growth rates, $\alpha_n = 1/(2E_n)(\mathrm{d}E_n/\mathrm{d}t)$, evaluated at $\tau = 45$, we obtain very similar values. These are illustrated in Fig. 10a. We obtain a growth rate equal to ~0.08 for the $n = 6$ and $n = 7$ modes (with a small preference for $n = 7$), which is the same value as Shariff found (with a small dominance for the $n = 6$ mode). The small difference observed could be due to the periodic condition in the z-direction in Shariff's simulation, contrary to the unbounded domain condition here. Indeed, his box size in this direction was equal to $6R$; but, at $\tau = 45$, the ring has travelled over $8R$. The ring has then entered in contact with his own wake, which is dominated by the $n = 6$ mode. This could explain why the $n = 6$ mode was a little more present in Shariff's simulation than in ours.

Fig. 10b provides the growth rates during the non-linear phase at $\tau = 100$. One observes that $n = 1$ and $n = 2$ modes are significantly amplified (as also observed by Shariff). One also sees that the next most non-linearly amplified waves are $n = 12$ and $n = 13$: the harmonics of the most linearly amplified modes, as also observed by Shariff. The next most amplified wave is $n = 19$, which is close to the second harmonic.

Fig. 11a and b show isocontours of $\omega_z\ t_0$, indicating the presence of axially elongated structures, in the plane of maximum $\|\boldsymbol{\omega}\|$ during the linear stage ($\tau = 70$), and at the early non-linear stage ($\tau = 100$). One sees that distinct patterns emerge. The $n = 6$ and $n = 7$ waves are clearly dominant and have the signature of second radial mode (i.e., three layers arranged radially), also consistent with flow visualization experiments. Fig. 11c–f also give isocontours of $\omega_z t_0$ (levels have changed) but for longer times, during the non-linear phase

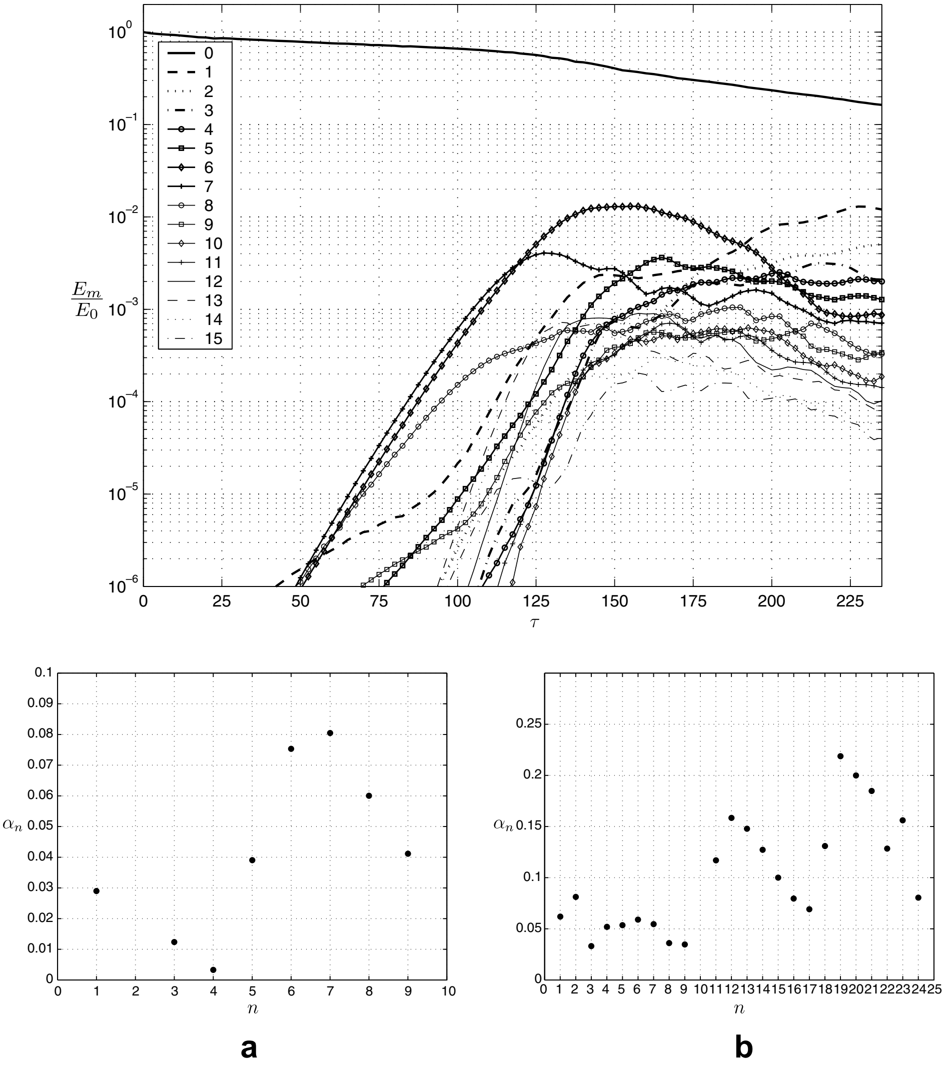
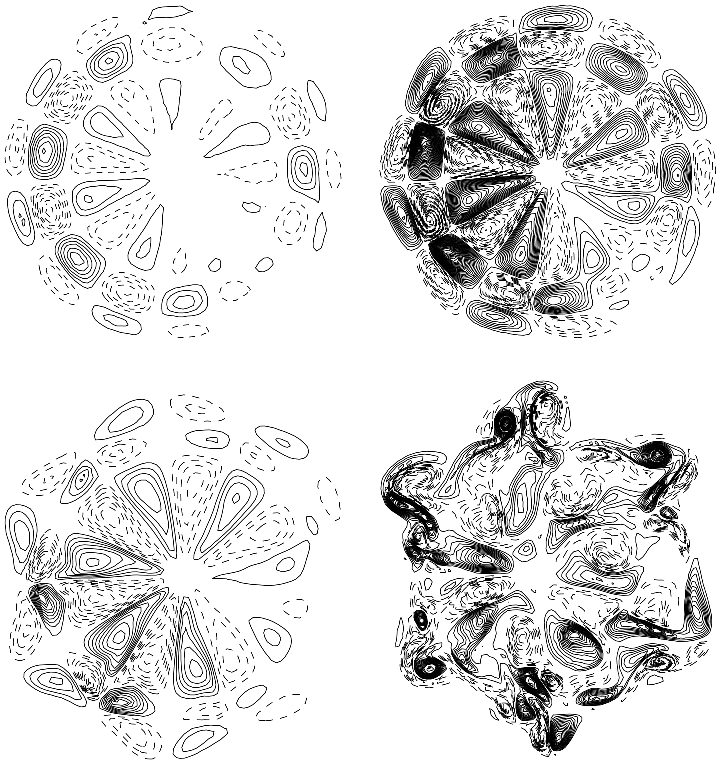


Fig. 10. Vortex ring at $Re = 5500$. Top: evolution of the modal energies, for modes 0–15. Bottom: growth rates for the modal energy at $\tau = 45$ (a) and $\tau = 100$ (b). The modes which still have an oscillatory behavior are not shown.

and after the saturation. One observes the rapid growth of axial vorticity, and for (c) and (d), the loss of well structured patterns: the "turbulent" phase, with strong decay. Fig. 12 provides the front three-dimensional views of the ring evolution for the same times as in Fig. 11. One first observes the distortion of the isosurfaces due to the linear instability, then the more complex flow resulting from the non-linear evolution. At $\tau = 175$, we also see an ejection of hairpin vortices into the wake: this behavior was observed in experiments by Schneider [34], Auerbach [4] and Weigand and Gharib [38].

For the second case, we choose a much thinner vortex ring. The initial core size is here set to $\sigma_0/R = 0.2$. The ring is also perturbed on the first 24 azimuthal modes, with an amplitude $\epsilon/R = 0.0001$. The Reynolds number is also increased to $Re = 25{,}000$. The discretization is quite fine, $h/R = 0.010$, and the time step is $\Delta\tau = \Delta t/t_0 = 0.01$. To be able to also capture the turbulence and the turbulent decay phase, we add a SGS model: we here use the advanced RVM model presented in Section 2.5, which is guaranteed not to dissipate in the well-resolved flow region.

The simulation was carried out for 13,500 time steps and took 250 h on 9 Opteron at 2.6 GHz. The number of vortex particles grew from 5.5 million to 10 million, and the number of grid points grew from 12 million to



24 million. We see in such case the interest to use a compact grid: the same simulation using the VIC method with periodic boundary conditions and using a grid size corresponding to Case 3 of Fig. 8, would lead to more than 240 million grid points and still would not represent well an unbounded flow. Another interesting comparison can be made with the vortex ring simulation done by M. Bergdorf, P. Koumoutsakos and A. Leonard (private communication, submitted 2007) using the VIC method with periodic boundary conditions and $h/R = 0.014$: the grid was $L/D = 2.8$ and $H/D = 11.2$ (as $H = 4L$, it indeed better represents an unbounded flow) and it required $400 \times 400 \times 1600 = 256$ million grid points.

Fig. 13 shows the evolution of the modal energies E_n. The same behavior as for the first case is observed. The most amplified mode during the linear phase is here the mode 11. We can compare this result with the most unstable mode expected using Saffman's theory [30,36]:

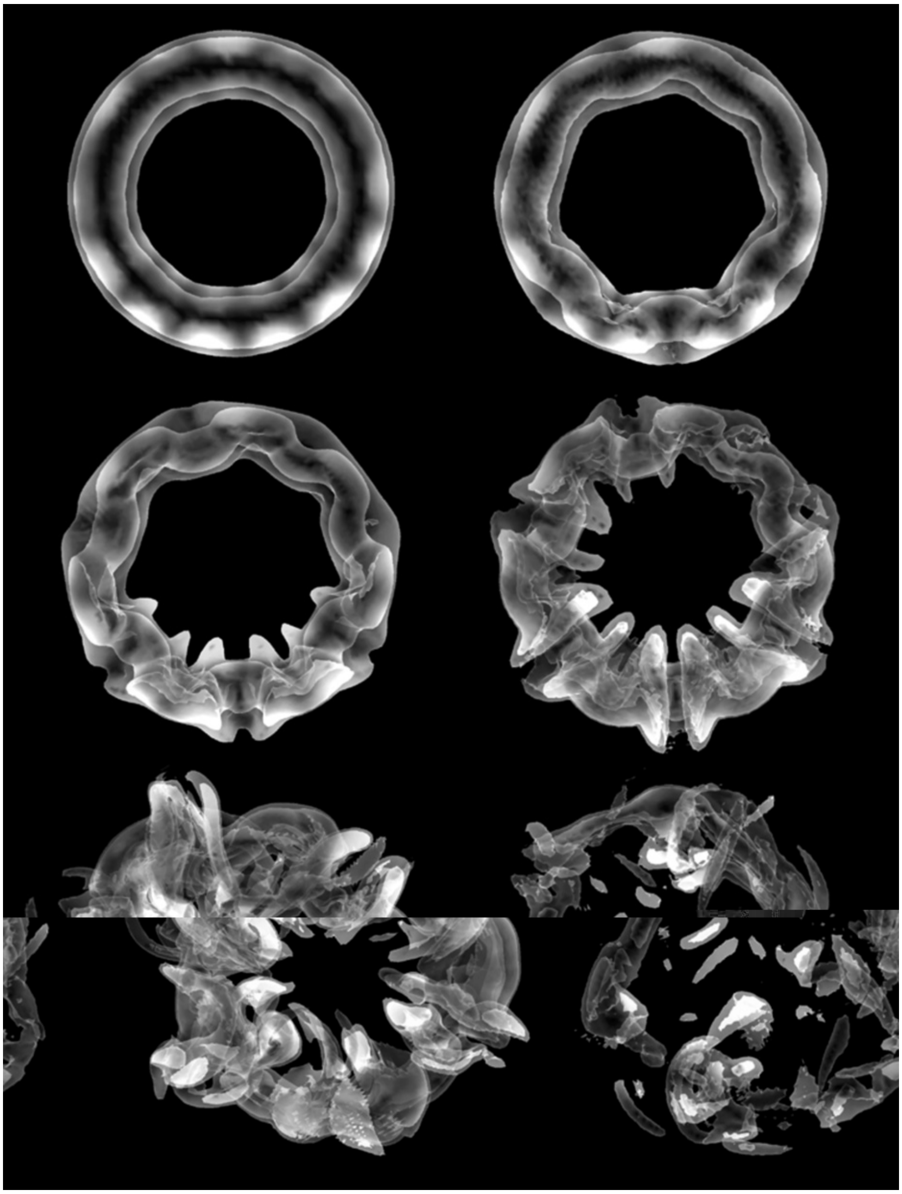


Fig. 12. Vortex ring at $Re = 5500$: 3-D front view of the instabilities. Two isosurfaces of vorticity shown: $\|\boldsymbol{\omega}\|t_0 = 1.0$ (opaque) and $\|\boldsymbol{\omega}\|t_0 = 0.686$ (semi-transparent). The grey scale is proportional to ω_z. Times shown are: $\tau = 70$, $\tau = 100$, $\tau = 115$, $\tau = 130$, $\tau = 175$ and $\tau = 220$.

$$n(t) = \frac{2.51}{1.12091 s(t)/R} \quad \text{with } s^2(t) = \sigma_0^2 + 4\nu t \tag{35}$$

For $t \in [0; 40]$ (i.e., during the linear phase), $n(t) \in [10.4; 11.2]$ which is exactly the most unstable mode obtained numerically. Fig. 13 illustrates the energy spectrum at three times. We here also see the most unstable mode $n = 11$ and the next most non-linearly amplified mode, its first harmonic $n = 22$. We observe that, for the last time, the spectra has a fair $k^{-5/3}$ behavior, arguing that we tend to a fully turbulent decaying flow.

Fig. 14 shows isocontours of $\omega_z t_0$ in the plane of maximum $\|\boldsymbol{\omega}\|$ during the linear stage ($\tau = 50$), the early non-linear stage ($\tau = 76$), at the saturation ($\tau = 92$), during the decay phase ($\tau = 102$) and when the flow becomes fully turbulent ($\tau = 112$ and $\tau = 134$). Fig. 15 provides the front three-dimensional views of the vortex ring evolution, shown at half the grid resolution. The different phases are also well observed.

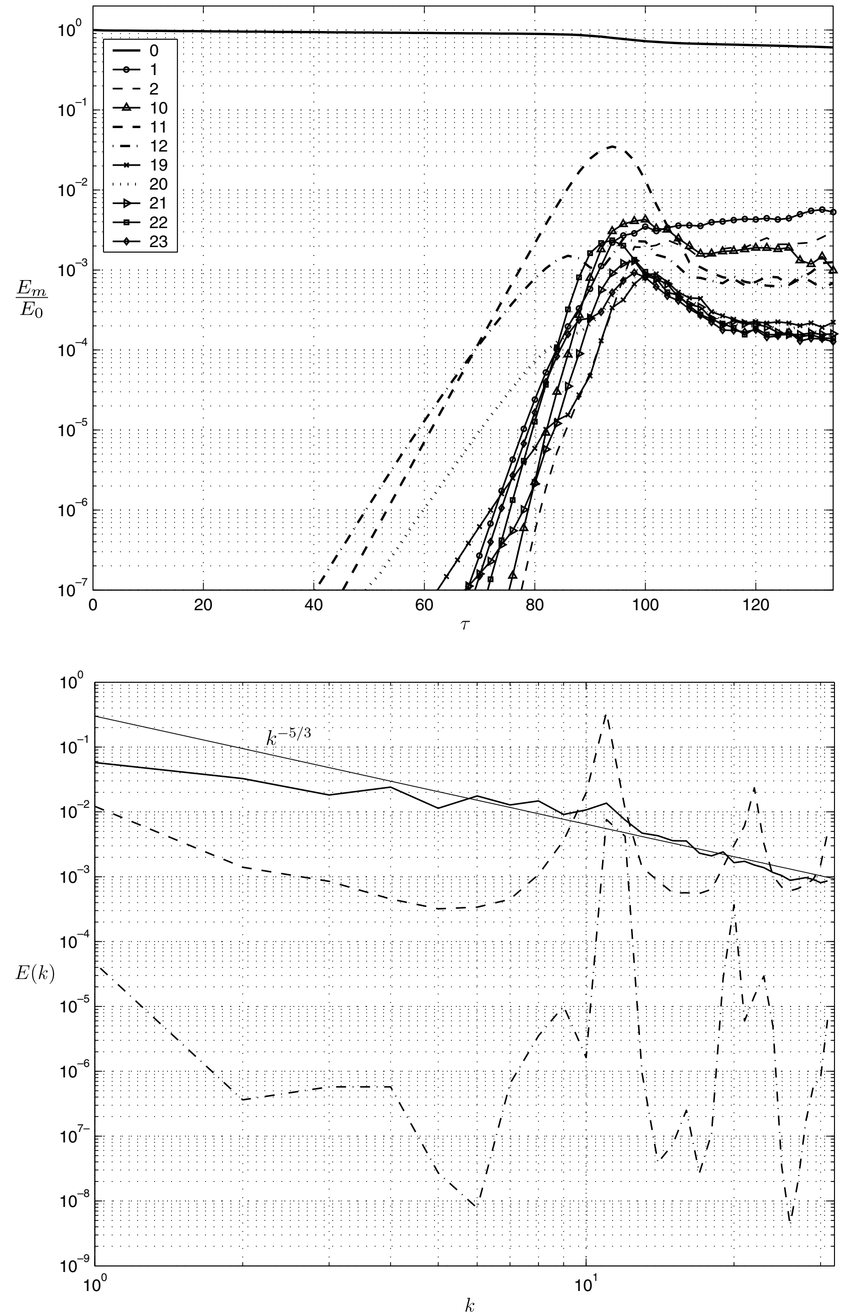


Fig. 13. Thin vortex ring at $Re = 25{,}000$. Top: evolution of the modal energies, for modes in the range 0–23. Bottom: energy spectrum at $\tau = 76$ (dash-dotted), $\tau = 92$ (dashed) and $\tau = 134$ (solid).

4.2. Two vortex system in ground effect

As an illustration of periodic and half-unbounded flows, we present a DNS of a two vortex system in ground effect at moderate Reynolds number. The thorough understanding of aircraft wake dynamic is largely studied in several research programs, especially in order to reduce the separation standards while maintaining or increasing the safety level during landing and take-off phases. The far wake of an aircraft is mainly constituted of two counter-rotating vortices (of circulation Γ_0 and separated by b_0). This event can be hazardous,

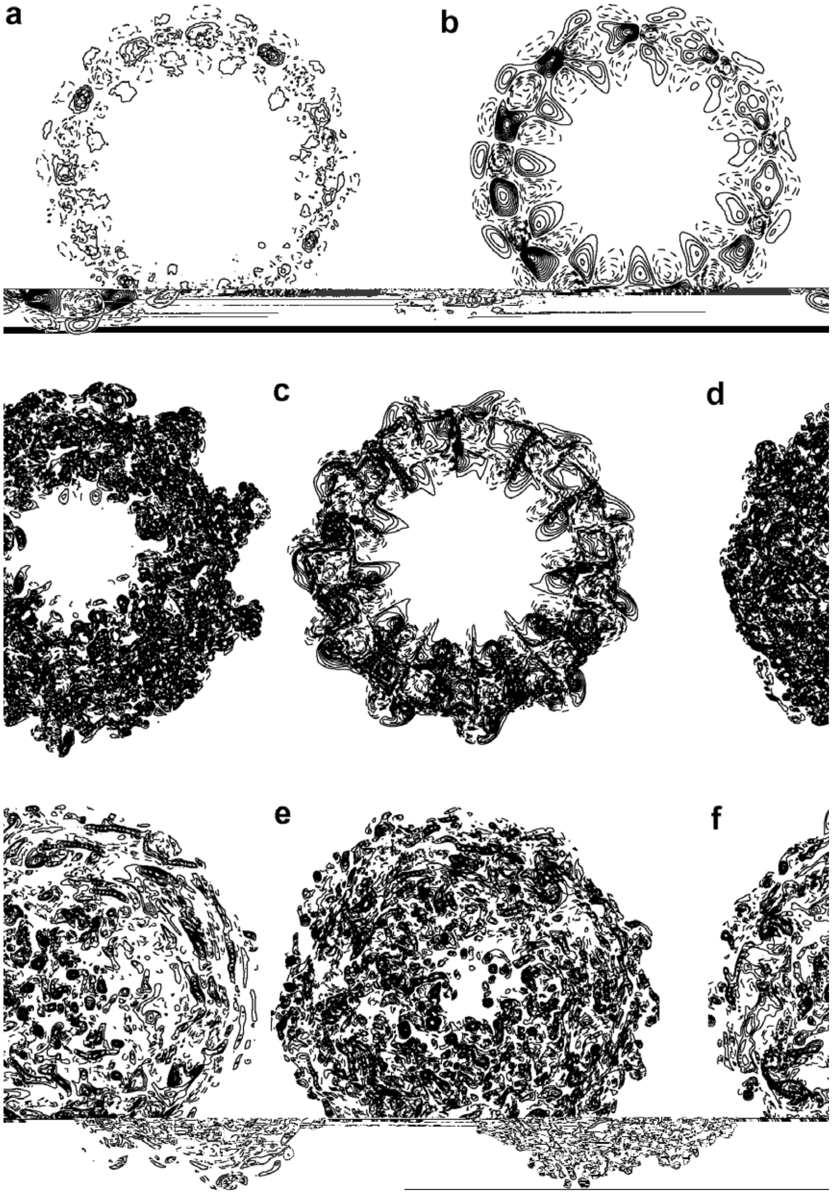


Fig. 14. Thin vortex ring at $Re = 25{,}000$: isocontours of $\omega_z t_0$ in the plane of maximum $\|\boldsymbol{\omega}\|$. Dashed and dotted lines represent positive and negative values, respectively. Levels are first by step of 0.02: (a) $\tau = 50$ (linear stage), levels are then by step of 0.16, (b) $\tau = 76$ (beginning of the non-linear stage). Levels are finally by step of 0.64: (c) $\tau = 92$ (saturation), (d) $\tau = 102$ (decay phase), (e) $\tau = 112$ and (f) $\tau = 134$ (turbulent and decay phase).

especially near the ground. A follower aircraft entering such wake could experience a strong rolling moment. Harvey and Perry [14] performed first experimental tests of a wing in ground effect where they observed the well-known rebound of the vortices due to the separation of the boundary layer created on the no-slip wall. Later, Barker and Crow [5] also observed experimentally the rebound phenomenon. Several numerical studies, in 2-D, were also performed by Peace and Riley [26], confirmed analytically by Saffman [31] and Orlandi [24]. Luton and Ragab [22] showed, using 3-D numerical simulations at low Reynolds number, that the interaction between a two vortex system and a viscous ground leads to a complex three-dimensional flow due to the appearance of short-wavelength instabilities (i.e., Widnall instabilities) on the secondary vortices. Here, we performed a similar simulation but at a higher (yet still moderate) Reynolds number, to obtain a stronger interaction between the secondary and primary vortices compared with [22].

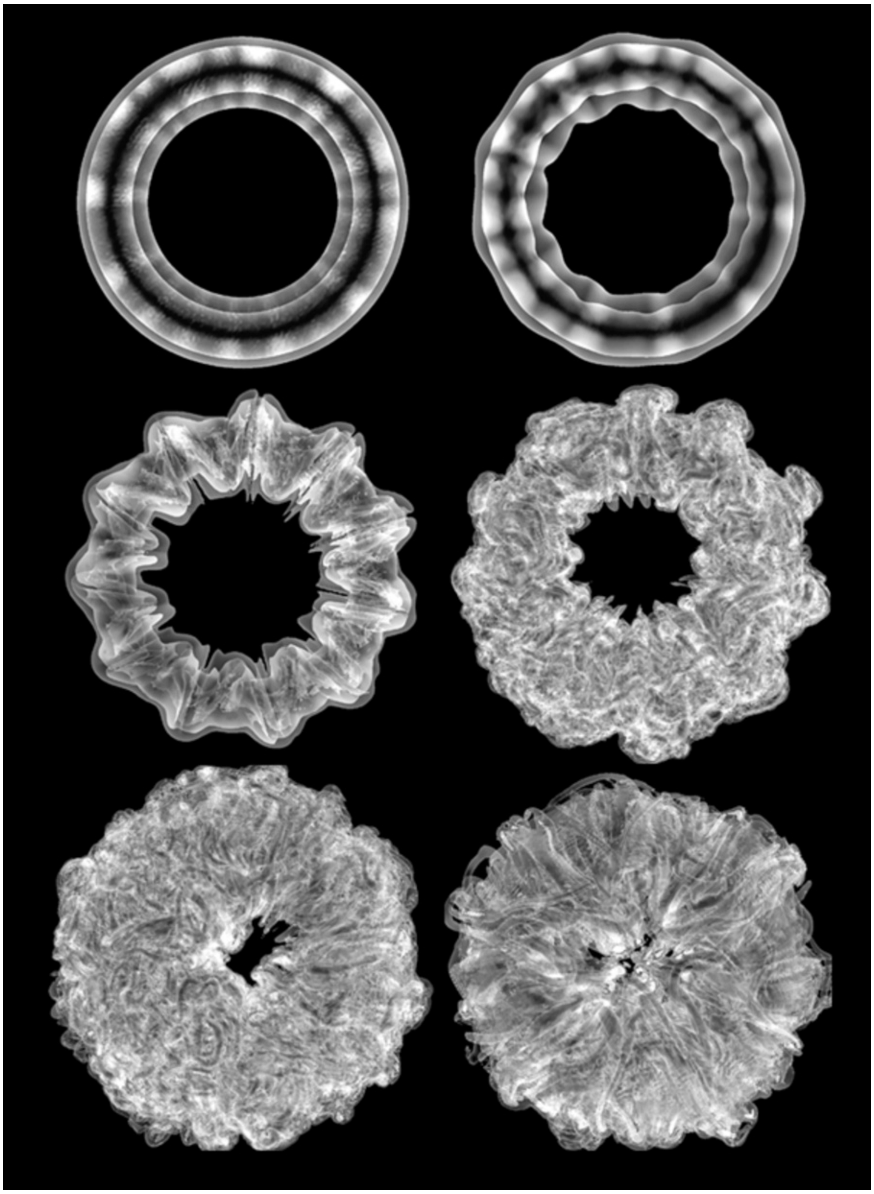


Fig. 15. Thin vortex ring at $Re = 25{,}000$: 3-D front view of the instabilities. Two isosurfaces of vorticity shown: $\|\boldsymbol{\omega}\| t_0 = 3.0$ (opaque) and $\|\boldsymbol{\omega}\| t_0 = 1.0$ (semi-transparent). The grey scale is proportional to ω_z. Times shown are: $\tau = 50$, $\tau = 76$, $\tau = 92$, $\tau = 102$, $\tau = 112$ and $\tau = 134$.

The problem is taken periodic in x and of large extent $L_x = 4b_0$. Initially, the vortices are algebraic vortices:

$$\omega_x(r) = \frac{\Gamma_0}{\pi} \frac{\sigma^2}{(r^2 + \sigma^2)^2} \tag{36}$$

with Γ_0 the total circulation and with a small core size $\sigma/b_0 = 0.05$. The dimensionless time is defined as $\tau = t/t_0$ with $t_0 = b_0/V_0$ (based on the descent velocity of a vortex pair away from the ground: $V_0 = \Gamma_0/(2\pi b_0)$). They are separated by a distance b_0 and are placed at height $h_0 = b_0$. The discretization is $h/L_x = 1/256$. The simulation is a DNS at $Re = \Gamma_0/\nu = 5000$, still low enough to avoid the use of a SGS model. The run was performed on 16 Opteron 2.6 GHz, 8 in the periodic direction and 2 in the spanwise direction.

Fig. 16 shows a view of the solution at different times. One sees that secondary vortices emanate from the ground because of the detachment of the boundary layer and are responsible for the rebound of the main vortices. During their orbital movement, short-wave (elliptic) instabilities develop on these secondary vortices. The non-linear interactions of these unstable modes and their saturation lead to a strong interaction with the main vortices, then to a complex 3-D flow and eventually, to the turbulent decay of the main vortices. Fig. 17 shows, at the same times, the space averaged vorticity for the right half plane. We observe more precisely the detachment of the boundary layer, the resulting orbital movement and, eventually, the turbulent vortex system.

We also compare our results to the solution obtained using a 2-D finite differences (FD) method that is energy conserving: it is thus a good reference to further validate our results. In this code, the computational box is also periodic in the spanwise direction: the box size was thus taken large “enough” ($L_y = 8b_0$) to mitigate the effect of the periodic images. The domain in the wall normal direction is taken as $L_z = 3b_0$ and a slip condition is used on top. The discretization is $h/b_0 = 0.004$ (thus $N_y = 2000$ and $N_z = 750$).

Fig. 18 shows the energy evolution. The curves compare quite well before the saturation of the secondary vortex instability (i.e., before the flow becomes 3-D). Indeed, once the short-wavelength instabilities appear on the secondary vortices, leading to the strong interaction, the dissipation rate increases strongly: it results to an enhanced decreased of the intensity of the vortex system compared to the 2-D case. Fig. 18 also compares the trajectories of the center of the main vortices (defined as the maximum of the averaged axial vorticity in the 3-D case, and as the maximum vorticity in the 2-D case). Even during the essentially 2-D phase, a difference is observed, due to the artificial confinement effects in the FD code: it leads to a slightly different displacement of the vortices (i.e., a different “rebound”). We see again the usefulness of an open-domain simulation performed by the VIC–PFM combination.

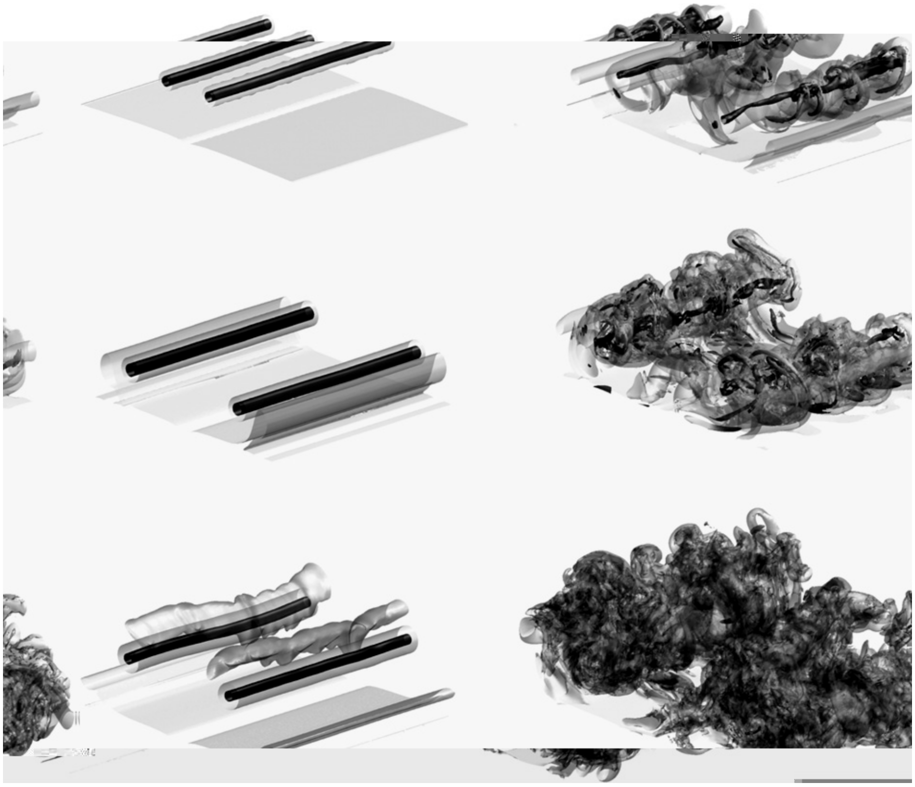


Fig. 16. Two vortex system in ground effect at $Re = 5000$: two isosurfaces of vorticity are shown (for one periodicity length): $\|\boldsymbol{\omega}\| b_0^2/\Gamma_0 = 10.0$ (opaque) and 1.0 (semi-transparent). The grey scale is proportional to ω_x. Times shown are: $\tau = 0.08$; $\tau = 1.6$ (secondary vortices emanate from the ground); $\tau = 2.9$ (short-wave instabilities visible on the secondary vortices); $\tau = 3.5$ and $\tau = 4.2$ (strong interactions between the vortices); $\tau = 6.1$ (“turbulent” and decay phase).

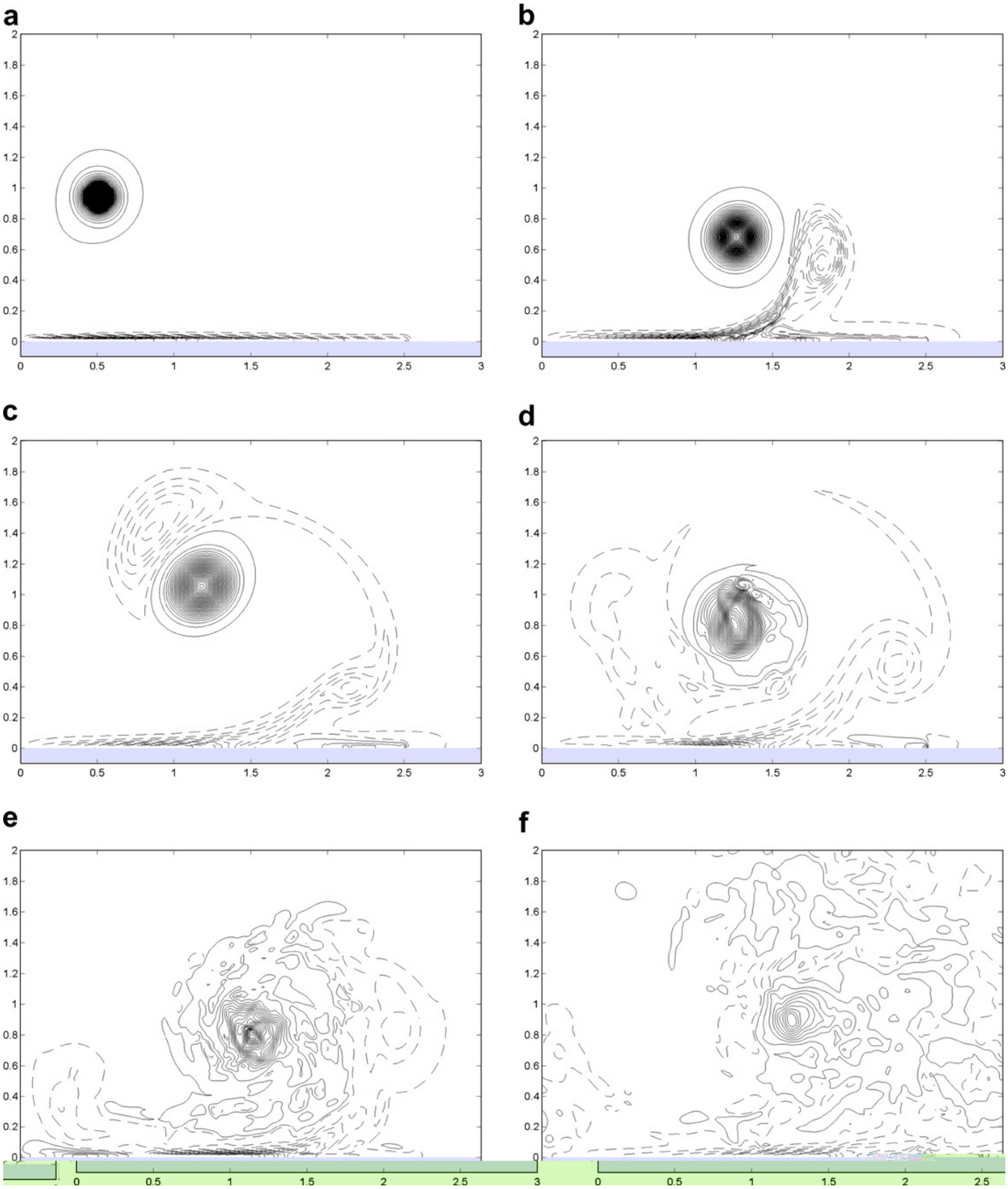


Fig. 17. Two vortex system in ground effect at $Re = 5000$: isocontours of longitudinally averaged vorticity $\omega_x t_0$ for the right half plane. Dashed and dotted lines represent positive and negative values, respectively. Levels are by step of 0.8π. The figures are shown at the same times as in Fig. 16.

4.3. Space-developing wake behind an elliptically loaded wing in ground effect

The last illustration of application is the case of a space-developing (SD) wake generated by an elliptically loaded wing in ground effect at a moderate (yet significant) Reynolds number ($\Gamma_0/\nu = 10{,}000$), and using LES. One important feature, that is not present in the previous time-developing (TD) simulations is the fact that the SD wake induces an axial velocity defect in the vortices. It therefore also exhibits an earlier and more violent transition to turbulence than the TD simulations. The SD configuration further differs from the TD case by

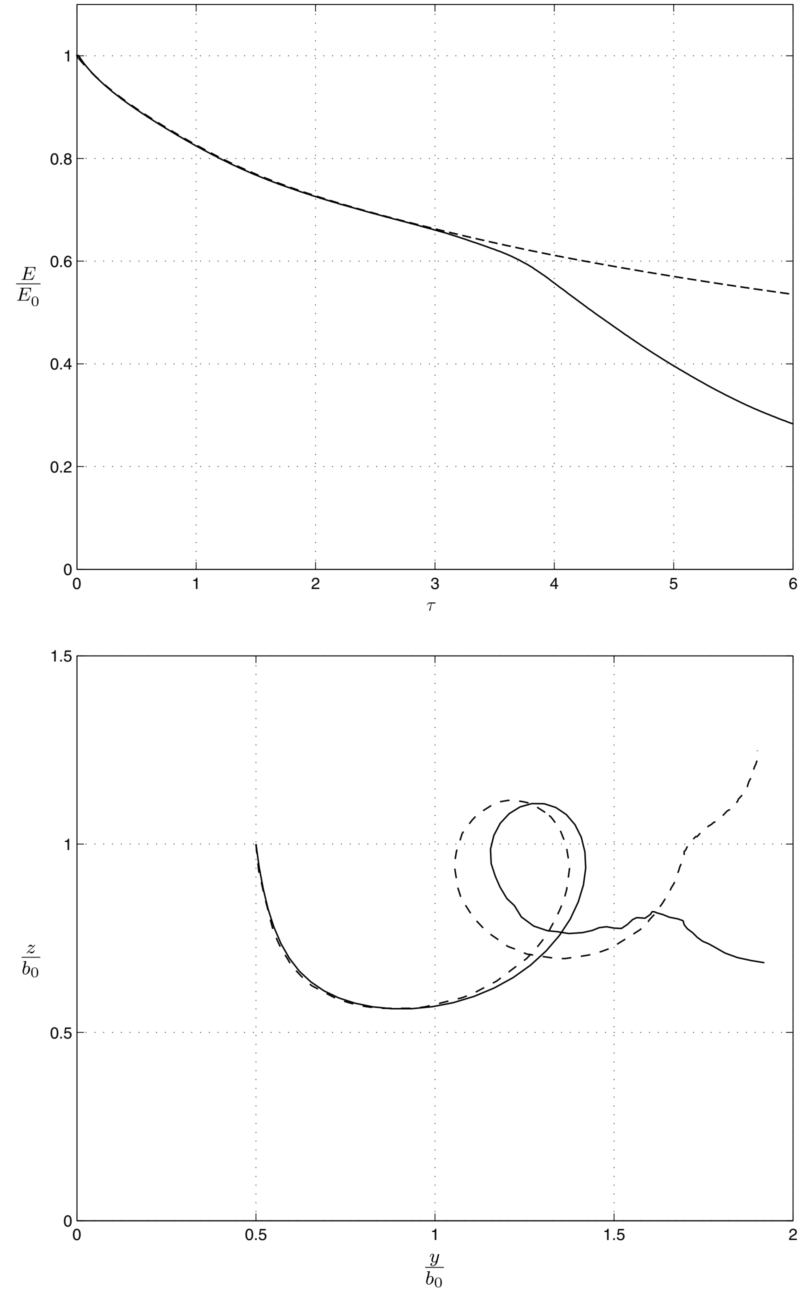


Fig. 18. Two vortex system in ground effect at $Re = 5000$. Top: total energy evolution using the VIC–PFM method (solid) and using a 2-D FD method with spanwise periodicity (dashed). Bottom: trajectory of the primary vortex.

the fact that the vortex tubes describe a spiraling motion. This feature is also suspected to trigger some kind of vortex meandering.

We consider the wake generated by an elliptically loaded wing of span b. The initial vortex sheet produced by the wing is as follows: its circulation per unit length is $\gamma(y) = -\frac{\mathrm{d}\Gamma}{\mathrm{d}y}(y)$ with

$$\Gamma(y) = \Gamma_0\left(1 - \left(\frac{y}{b/2}\right)^2\right)^{1/2}$$

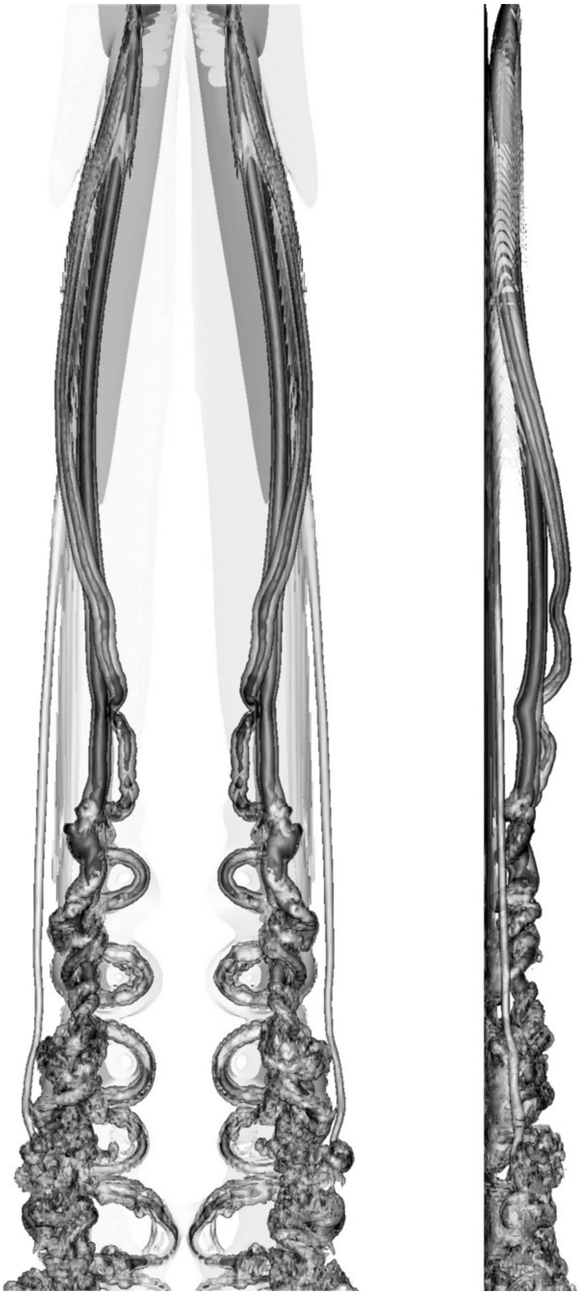


Fig. 19. 3-D space-developing LES of wake vortices behind an elliptically loaded wing in ground effect: isosurfaces of vorticity norm, $\|\boldsymbol{\omega}\| t_0 = 80$ (opaque) and 25 (semi-transparent), colored by the longitudinal component of vorticity, when the flow has reached it's regime. Top and side views.

In reality, $\Gamma(y)$ is slightly modified to take into account the ground effect using a modified "lifting line" theory based on Prandtl's theory and the use of "images", but we do not enter into the details here. This singular vortex sheet is then regularized in order to produce a regular vorticity field. For this, we use convolution with a Gaussian:

$$\omega_\sigma(y,z) = \frac{1}{2\pi\sigma^2} \int_{-b/2}^{b/2} \exp\left(-\frac{(y-y')^2 + (z-h_0)^2}{2\sigma^2} \right) \gamma(y')\,\mathrm{d}y' \tag{37}$$

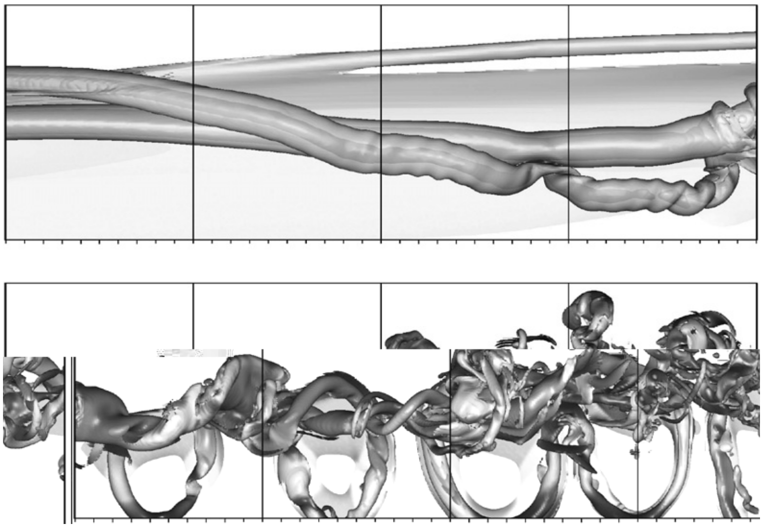


Fig. 20. 3-D space-developing LES: top view, here shown at full resolution, of the regions $x/b = [4, 8]$ and $[8, 12]$ at an arbitrary time once in regime.

where h_0 is the distance to the ground and σ is the regularization parameter. We here used $\sigma/b = 0.02$. The regularized vorticity field is then further discretized using particles put on multiple layers.

The height of the wing flying above the ground is $h_0/b = 0.25$. Fig. 19 shows a snapshot of the spatially developing flow when established. The effective computational domain extends between the inflow and outflow planes. The flight velocity, U_∞, was here taken lower than that of a realistic case: U_∞/V_0 is about 4 times lower than it should (where $V_0 = \Gamma_0/(2\pi b_0)$ and $b_0 = (\pi/4)b$). That choice limits the computational requirements in the longitudinal direction. A more realistic case would require a four times longer domain. The SD simulation presented here is thus to be taken as a "proof of concept" of what the VIC–PFM method can do. The resolution used is $h/b = 0.01$. The time step was $\Delta t/t_0 = 5 \times 10^{-4}$, with $t_0 = b_0/V_0$. The number of particles reached about 14 million and the size of the grid was $L_x/b = 12.5$, $L_y/b = 4$ and $L_z/b = 1.1$ (thus $N_x = 1250$, $N_y = 400$ and $N_z = 110$ for a total of 55 million grid points).

Compared to TD simulations, the computation has here to be carried on for a sufficiently long time for the flow to reach its regime state (i.e., its statistical equilibrium). Indeed, the flow being initialized using a spatially uniform wake, at least a complete flow through time, $t = L_x/U_\infty$, is required to "evacuate" this initial condition. A visualization of the space-developing simulation showed that the flow had reached a well established regime after about 1.6 flow through times. Once this regime state was attained, the flow was computed further, to attain statistical convergence of the time averaged fields of interest at different x/b stations (not shown). The complete simulation ran 10^4 time steps.

We see, in Fig. 19, the global spiraling motion of the wake vortices. The secondary vortex interacts with the primary vortex before having performed a complete revolution around it. The interaction is initiated by a medium wave instability that propagates on the secondary vortex. There is also some kind of vortex "meandering" of the secondary vortex, seen when examining time sequences (not shown here). Fig. 20 provides zooms of the flow in the mid wake and in the far wake: this also illustrates the good resolution quality of this simulation. It shows how the secondary vortex partially reconnects with the primary one through a complex spiraling motion. Due to this interaction, the primary vortex appears to "burst" into two or even three helical vortices which then further "break down" into small-scales. This simulation, and its detailed analysis (not shown), is part of the European project FAR-Wake.

5. Conclusions

We have presented a new approach, the VIC–PFM method, that efficiently combines a vortex-in-cell (VIC) method and a parallel fast multipole (PFM) method. It allows to efficiently simulate unbounded or half-unbounded flows, by using the advantages of each method: the efficiency of grid Poisson solvers used in VIC methods, and the implicit unbounded domain condition of PFM methods. As the open-domain boundary conditions are obtained correctly using the PFM method, it avoids any iterative method and it permits to use a computational domain (i.e., the VIC grid) as compact as the vorticity field, leading to much smaller memory requirements.

For large problems, the method was further parallelized. The grid is then split into several subdomains, each subgrid being allotted to one processor. The continuity between the subdomains is ensured by the PFM method, that always has a global view of the entire vorticity field.

The VIC–PFM is able to simulate complex flows, also turbulent flows using Large-Eddy Simulation (LES) approaches. Different SGS models were also implemented. An efficient relaxation scheme was also implemented (following Cottet and Poncet [12]): it enforces that the vorticity field remains divergence free at all times, which significantly enhances the accuracy and robustness of the method.

Various applications were presented and some compared with other methods and/or existing results. The VIC–PFM method shows its efficiency and its ability to simulate complex and turbulent unbounded flows, as shown by the simulation of a thin vortex ring at high Reynolds number. The VIC–PFM method was also implemented with one periodic direction and was here illustrated on the simulation of a two vortex system in ground effect. A space-developing simulation was also presented, for the longitudinal development, in ground effect, of the vortex wake emitted by an elliptically loaded wing.

Acknowledgement

R. Cocle was a research and teaching assistant funded by UCL. G. Daeninck was a researcher funded by the FRIA, then further funded by the FAR-Wake project (Fundamental Research on Aircraft Wake Phenomena), contract AST4-CT-2005-012238, of the European Commission within the Sixth Framework Programme. We also acknowledge useful discussion with F. Thirifay, who also used the combined VIC-FM method in the framework of his Ph.D. thesis.

References

[1] J. Adams, P. Swarztrauber, R. Sweet, Efficient FORTRAN subprograms for the solution of elliptic partial differential equations (FISHPACK ver.4.0), NCAR Technical Note – TN/IA-109, 1975.

[2] J. Adams, Multigrid Fortran software for the efficient solution of linear elliptic partial differential equations (MUDPACK), Appl. Math. Comput. 34 (1989) 113.

[3] C. Albuquerque, G.-H. Cottet, Coupling finite difference methods and integral formulas for elliptic problems arising in fluid mechanics, Numer. Methods PDE 20 (2003) 199–229.

[4] D. Auerbach, Stirring properties of vortex rings, Phys. Fluids A 3 (1991) 1351.

[5] S.J. Barker, S.C. Crow, The motion of two-dimensional vortex pairs in a ground effect, J. Fluid Mech. 82 (4) (1977) 659–671.

[6] J.E. Barnes, P. Hut, A hierarchical $\mathcal{O}(N\log(N))$ force calculations algorithm, Nature 324 (1986) 446.

[7] J.P. Christiansen, Numerical solution of hydrodynamics by the method of point vortices, J. Comput. Phys. 13 (1973) 363.

[8] R. Cocle, L. Dufresne, G. Winckelmans, Investigation of multiscale subgrid scale models for LES of instabilities and turbulence in wake vortex systems, Complex Effects in Large Eddy Simulations, Lecture Notes in Computational Science and Engineering (LNCSE), vol. 56, Springer, 2007.

[9] G.-H. Cottet, P.D. Koumoutsakos, Vortex Methods, Theory and Practice, Cambridge University Press, 2000.

[10] G.-H. Cottet, D. Jiroveanu, B. Michaux, Vorticity dynamics and turbulence models for large-eddy simulations, Math. Modell. Numer. Anal. 37 (2003) 187–207.

[11] G.-H. Cottet, P. Poncet, Particle methods for direct numerical simulations of three-dimensional wakes, J. Turbulence 3 (028) (2003) 1 (htpp://jot.iop.org/).

[12] G.-H. Cottet, P. Poncet, Advances in direct numerical simulations of 3-D wall-bounded flows by vortex-in-cell methods, J. Comput. Phys. 193 (2004) 136.

[13] L. Greengard, V. Rohklin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325.

[14] J.K. Harvey, F.J. Perry, Flowfield produced by trailing vortices in the vicinity of the ground, AIAA J. 9 (8) (1971) 1659–1660.

[15] T.J.R. Hughes, L. Mazzei, A.A. Oberai, A.A. Wray, The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence, Phys. Fluids 13 (2001) 505–512.

[16] H. Jeanmart, G. Winckelmans, Comparison of recent dynamic subgrid-scale models in turbulent channel flow, in: Proceedings of Summer Program 2002, Center for Turbulence Research, Stanford University and NASA Ames, 2002, pp. 105–116.

[17] H. Jeanmart, G. Winckelmans, Investigation of eddy-viscosity models modified using discrete filters: a simplified “regularized variational multiscale model” and an “enhanced field model”, Phys. Fluids 19 (5) (2007) 055110.

[18] P. Koumoutsakos, A. Leonard, High resolution simulations of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech. 296 (1995) 1–38.

[19] K. Lindsay, R. Krasny, A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow, J. Comput. Phys. 172 (2001) 879.

[20] A. Leonard, Vortex methods for flow simulation, J. Comput. Phys. 37 (1980) 289.

[21] A. Leonard, Computing three-dimensional incompressible flows with vortex elements, Annu. Rev. Fluid Mech. 17 (1985) 523.

[22] J.A. Luton, S.A. Ragab, Three-dimensional interaction of a vortex pair with a wall, Phys. Fluid 9 (10) (1997) 2967–2980.

[23] J.J. Monaghan, Extrapolating B splines for interpolation, J. Comput. Phys. 60 (1985) 253–262.

[24] P. Orlandi, Vortex dipole rebound from a wall, Phys. Fluid A 2 (1990) 1429–1436.

[25] M.L. Ould-Salihi, G.-H. Cottet, M. El Hamraoui, Blending finite-difference and vortex methods for incompressible flow computations, SIAM J. Scient. Comput. 22 (5) (2000) 1655–1674.

[26] A.J. Peace, N. Riley, A viscous vortex pair in ground effect, J. Fluid Mech. 129 (1983) 409–426.

[27] P. Ploumhans, G.S. Winckelmans, Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry, J. Comput. Phys. 165 (2000) 354.

[28] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, M.S. Warren, Vortex methods for direct numerical simulation of three-dimensional bluff body flows: application to the sphere at $Re = 300$, 500 and 1000, J. Comput. Phys. 178 (2002) 427.

[29] P. Poncet, Vanishing of mode B in the wake behind a rotationally oscillating circular cylinder, Phys. Fluids 14 (6) (2002) 2021.

[30] P.G. Saffman, The number of waves on unstable vortex rings, J. Fluid Mech. 84 (4) (1978) 625–639.

[31] P.G. Saffman, Approach of a vortex pair to a rigid free surface in viscous fluid, Phys. Fluid A 3 (5) (1991) 984–985.

[32] J.K. Salmon, M. Warren, Skeletons from the treecode closet, J. Comput. Phys. 111 (1994) 136.

[33] J.K. Salmon, M. Warren, G.S. Winckelmans, Fast parallel tree codes for gravitational and fluid dynamical N-body problems, Int. J. Supercomputer Appl. High Perform. Comput. 8 (2) (1994) 129.

[34] P.E.M. Schneider, Sekundarwirbelbildung bei Ringwirbeln und in Freistrahlen, Z. Flugwiss Weltraumforsch 4 (1980) 307.

[35] K. Shariff, A. Leonard, Vortex rings, Annu. Rev. Fluid Mech. 24 (1992) 235.

[36] K. Shariff, R. Verzicco, P. Orlandi, A numerical study of three dimensional vortex ring instabilities: viscous corrections and early nonlinear stage, J. Fluid Mech. 279 (1994) 351.

[37] S. Stolz, P. Schlatter, D. Meyer, L. Kleiser, High-pass filtered eddy-viscosity models for LES, in: R. Friedrich, B.J. Geurts, O. Métais (Eds.), Direct and Large-Eddy Simulation V, Kluwer, 2004, pp. 81–88.

[38] A. Weigand, M. Gharib, On the decay of a turbulent vortex ring, Phys. Fluids 6 (1994) 12.

[39] G.S. Winckelmans, H. Jeanmart, Assessment of some models for LES without and with explicit filtering, in: B.J. Geurts, R. Friedrich, O. Métais (Eds.), Direct and Large-Eddy Simulation IV, ERCOFTAC Series, vol. 8, Kluwer, 2001, pp. 55–66.

[40] G.S. Winckelmans, Vortex methods, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, vol. 3 (Fluids), John Wiley & Sons, 2004.