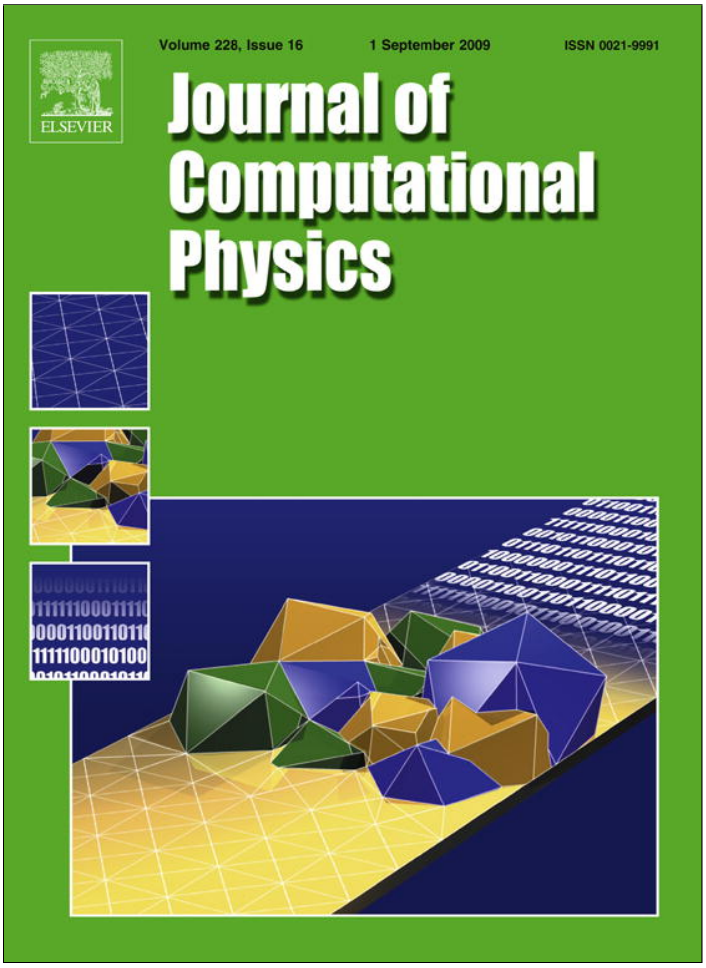
Volume 228, Issue 16
1 September 2009
ISSN 0021-9991
ELSEVIER
Journal of Computational Physics

Journal of Computational Physics 228 (2009) 5687–5709

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A Fourier spectral method for the Navier–Stokes equations with volume penalization for moving solid obstacles

Dmitry Kolomenskiy [a,*], Kai Schneider [a,b]

[a] Laboratoire de Mécanique, Modélisation et Procédés Propres (M2P2), CNRS et Universités d'Aix-Marseille, 38 rue Frédéric Joliot-Curie, 13451 Marseille cedex 20, France
[b] Centre de Mathématiques et d'Informatique (CMI), Université d'Aix-Marseille I, 39 rue Frédéric Joliot-Curie, 13453 Marseille cedex 13, France

ARTICLE INFO

Article history:
Received 2 May 2008
Received in revised form 6 April 2009
Accepted 16 April 2009
Available online 3 May 2009

Keywords:
Fluid–solid interaction
Volume penalization method
Spectral method
Freely falling bodies

ABSTRACT

An efficient numerical scheme to compute flows past rigid solid bodies moving through viscous incompressible fluid is presented. Solid obstacles of arbitrary shape are taken into account using the volume penalization method to impose no-slip boundary condition. The 2D Navier–Stokes equations, written in the vorticity-streamfunction formulation, are discretized using a Fourier pseudo-spectral scheme. Four different time discretization schemes of the penalization term are proposed and compared. The originality of the present work lies in the implementation of time-dependent penalization, which makes the above method capable of solving problems where the obstacle follows an arbitrary motion. Fluid–solid coupling for freely falling bodies is also implemented. The numerical method is validated for different test cases: the flow past a cylinder, Couette flow between rotating cylinders, sedimentation of a cylinder and a falling leaf with elliptical shape.

1. Introduction

The concept of fluid-structure interaction covers a wide variety of phenomena involving solid boundaries immersed in a fluid flow. These boundaries may have complex geometrical shape, may move and even deform, as it occurs in numerous applications, thus resulting in challenging tasks for computational fluid dynamics.

Nowadays, there exists a considerable number of techniques to introduce solid boundaries in a numerical model of the flow. Among those, immersed boundary and fictitious domain methods are becoming increasingly popular (see, *e.g.* [17,16]). The motivation is that for complex fluid-structure interaction problems these methods have a potential to be more easy-to-implement and more efficient than classical approaches such as body-fitted curvilinear grids. Thus one of the earliest works is by Peskin who introduced an immersed boundary method to explore flow patterns around heart valves [18]. Glowinski et al. developed a Lagrangian-multiplier-based fictitious domain method and applied it to a wide range of problems [33]. Among the recent publications there are simulations of fish and insect locomotion by Mittal et al. [31], studies of dragonfly flapping flight by Russell and Wang [19], and applications to particulate flows by Sharma and Patankar [32] and Uhlmann [28].

The common feature of the mentioned methods is that the physical solid boundary does not necessarily conform to the computational domain boundary. As a consequence, Cartesian grids and efficient numerical schemes can be implemented.

The volume penalization method considered in this paper is based on the idea of modelling solid bodies as porous media whose permeability tends to zero. The Navier–Stokes/Brinkman model, where the penalization source term in the momentum equation corresponds to the Darcy drag, was first proposed by Arquis and Caltagirone [1] in the context of the natural convection flow inside a fluid–porous cavity. It was then generalized to study the fluid–porous–solid systems [36]. In addition

[*] Corresponding author. Tel.: +33 4 91 11 85 34.
E-mail addresses: dkolom@gmail.com, dkolom@L3m.univ-mrs.fr (D. Kolomenskiy), kschneid@cmi.univ-mrs.fr (K. Schneider).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.04.026

to being physically motivated, this model is mathematically justified, due to the convergence property rigorously proved by Angot et al. [2]. Recent improvements of this fictitious domain approach include mixed boundary conditions [21] and higher order accuracy [20].

The volume penalization is an essential component of the present work. It allows such features as arbitrary shape and number of obstacles. Their motion and interaction with the fluid can be efficiently implemented in a relatively straightforward manner. Thus it represents a flexible tool for numerical simulation of fluid-structure interaction.

As a starting point we take the numerical scheme developed in [4]. It consists in solving the two-dimensional (2D) incompressible Navier–Stokes equations using a Fourier pseudo-spectral method together with the volume penalization method. Spectral discretizations are known as efficient tools for modelling of turbulent and transitional flows [6]. Fast Fourier transform software packages are nowadays well-optimized both for single-processor and parallel computations. Therewith, the numerical method is limited to periodic computational domains, which suggests considering fictitious domain techniques to simulate bounded flows. This approach was used by various authors to model flows past fixed obstacles, such as flows past tube bundles [8,9] and flat plates [10], confined turbulence in a circular container [25], and a dipole-wall collision [22]. In particular, the latter paper addresses the issue of higher order convergence.

The aim of the present paper is to develop further the above model, which consists in the implementation of time-dependent penalization to solve moving obstacle problems. We also explore possible improvements of time-stepping. In addition, for fluid–solid interaction we introduce coupling with the equations of solid body dynamics. Thus we obtain an instrument to perform direct numerical simulations (DNS) of a falling leaf, and reveal some features of this phenomenon. This latter interest is inspired by the beautiful simulations of falling leaves performed by Pesavento and Wang [13].

The paper is organized as follows. In Section 2, we briefly recall the physical model and its space discretization. The reader is referred to [4] for more details at this point. Section 3 presents a discussion of different time discretization schemes to treat the penalization term. In Section 4, we describe in detail our approach to time-dependent penalization for moving obstacles. It is validated by comparing moving and fixed cylinder simulations, and also by performing a convergence test for Couette flow between rotating cylinders. In Section 5, we present some results for freely falling solid bodies. Finally, in Section 6, we draw conclusions on the present results and outline perspectives for future work.

2. Physical model and spatial discretization

In this section we recall the equations governing the motion of viscous incompressible fluid, followed by presenting the volume penalization method. Then, the space discretization of the problem using a Fourier pseudo-spectral method is briefly described.

2.1. Governing equations

We consider interactions between viscous incompressible fluid and rigid solids moving in it. In this case the fluid motion is governed by the incompressible Navier–Stokes equations – the momentum equation

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p - \nu \nabla^2 \mathbf{u} = \mathbf{f} \tag{1}$$

and the continuity equation

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

with no-slip boundary conditions on the solid walls moving with velocity $\mathbf{V}_{\partial\Omega_s}$

$$\mathbf{u}|_{\partial\Omega_s} = \mathbf{V}_{\partial\Omega_s} \tag{3}$$

and completed with a suitable initial condition.

Let Ω_f denote the fluid domain, Ω_s the solid one and $\Omega = \Omega_f \bigcup \Omega_s$ the entire domain (see Fig. 1). Only two-dimensional flows are studied in this work. The spatial coordinates in the two-dimensional domain are $\mathbf{x} = (x, y) \in \Omega$ and time is

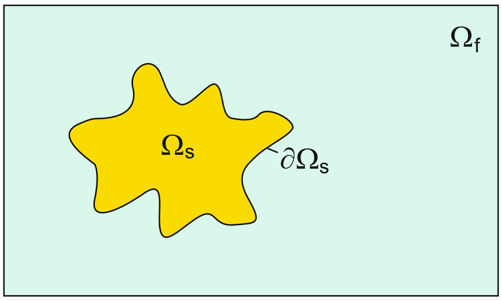

Fig. 1. The computational domain Ω contains the fluid domain Ω_f, the solid obstacle Ω_s and its boundary $\partial\Omega_s$.

$t \in [0, t_{max}]$. Note that in general the fluid and the solid domains, the solid boundary $\partial\Omega_s$ and its velocity $\mathbf{V}_{\partial\Omega_s}$ depend on time.

The equations are written for dimensionless variables. Units are chosen in order to set the dimensionless density $\rho = 1$. The flow velocity $\mathbf{u}(\mathbf{x}, t)$ and the pressure $p(\mathbf{x}, t)$ are unknowns, while $\mathbf{f}(\mathbf{x}, t)$ is a source term (such as gravitational force). The parameter $\nu > 0$ is the kinematic viscosity.

The fluid–solid interaction simulations presented in this paper invoke the volume penalization method, originally proposed by Arquis and Caltagirone [1] for the natural convection flow inside a fluid–porous cavity, and then generalized to study the fluid–porous–solid systems [36]. The method is based on the idea of modelling solid obstacles as porous media. When the permeability η tends to zero the obstacle tends to be a solid with no-slip boundary conditions on its walls. The flow is considered in a unified domain in which both fluid and solid domains are imbedded. The difference between them is the permeability. The momentum equation (1) is modified by adding a penalization term on the velocity:

$$\partial_t \mathbf{u}_\eta + \mathbf{u}_\eta \cdot \nabla \mathbf{u}_\eta + \nabla p_\eta - \nu \nabla^2 \mathbf{u}_\eta + \frac{\chi_\Omega}{\eta}(\mathbf{u}_\eta - \mathbf{u}_s) = \mathbf{f}, \tag{4}$$

where $\chi_\Omega(\mathbf{x}, t)$ is the mask function describing the geometry of the obstacle

$$\chi_\Omega = \begin{cases} 1 & \text{for } \mathbf{x} \in \overline{\Omega}_s, \\ 0 & \text{for } \mathbf{x} \in \Omega_f \end{cases} \tag{5}$$

and $\mathbf{u}_s(\mathbf{x}, t)$ is its velocity.

According to Eq. (4), for small η the flow is governed by the Navier–Stokes equations in Ω_f, and by Darcy's law in Ω_s. The convergence of $\mathbf{u}_\eta$ to $\mathbf{u}$ in the limit of vanishing η was rigorously proved by Angot et al. [2] for fixed obstacles. The estimates were then refined by Carbou and Fabrie [34], who demonstrated $\eta^{1/2}$ rate of convergence.

The hydrodynamic forces and moment on the obstacle can be computed by integrating the penalized velocity over the obstacle volume, instead of evaluating surface integrals of the strain $\sigma(\mathbf{u}, p) = \frac{1}{2\nu}(\nabla\mathbf{u} + (\nabla\mathbf{u})^t) - pI$ [2,3]:

$$\mathbf{F} = \int_{\partial\Omega_s} \sigma \mathbf{n}_f \, d\gamma = \lim_{\eta\to 0} \int_\Omega \frac{\chi_\Omega}{\eta}(\mathbf{u}_\eta - \mathbf{u}_s) d\Omega + V_c \ddot{\mathbf{x}}_c, \tag{6}$$

$$M_{x_c} = \int_{\partial\Omega_s} (\mathbf{x} - \mathbf{x}_c) \times \sigma \mathbf{n}_f \, d\gamma = \lim_{\eta\to 0} \int_\Omega \frac{\chi_\Omega}{\eta}(\mathbf{x} - \mathbf{x}_c) \times (\mathbf{u}_\eta - \mathbf{u}_s) d\Omega + I_c \ddot{\theta}_c, \tag{7}$$

where $V_c = \int_{\Omega_s} d\mathbf{x}$ is the volume of the solid, $I_c = \int_{\Omega_s} (\mathbf{x} - \mathbf{x}_c)^2 \, d\mathbf{x}$ is the geometrical moment of inertia, $\mathbf{x}_c$ is the center of gravity position vector (we assume the uniform density distribution of the solid), and θ_c is the angle of rotation with respect to the center of gravity. Dots denote derivation with respect to time.

In 2D it is worthwhile to introduce the vorticity-streamfunction formulation of the penalized Navier–Stokes equation [4]. We take the curl of (4) to obtain

$$\partial_t \omega_\eta + \mathbf{u}_\eta \cdot \nabla \omega_\eta - \nu \nabla^2 \omega_\eta + \nabla \times \left(\frac{\chi_\Omega}{\eta}(\mathbf{u}_\eta - \mathbf{u}_s) \right) = \nabla \times \mathbf{f}, \tag{8}$$

where $\omega_\eta = \nabla \times \mathbf{u}_\eta$ denotes the vorticity, which is scalar-valued, since only the component perpendicular to the xy-plane is non-zero in 2D.

The source term $\mathbf{f}$ is assumed irrotational and the velocity is determined as a sum $\mathbf{u}_\eta = \nabla^\perp \Psi + \mathbf{U}_\infty$, with $\mathbf{U}_\infty$ being the free-stream velocity and Ψ being the stream function, satisfying

$$\nabla^2 \Psi = \omega_\eta, \tag{9}$$

where $\nabla^\perp \Psi = (-\partial_y \Psi, \partial_x \Psi)$ stands for the orthogonal gradient of the stream function.

The equations (8) and (9) are solved numerically using a Fourier pseudo-spectral technique [6]. In this connection it is relevant to make a remark that as long as the numerical method implies periodic boundary conditions, the irrotational part of the velocity has to be constant [5], so it is interpreted as $\mathbf{U}_\infty$ and corresponds to the mean velocity.

2.2. Pseudo-spectral discretization

For the spatial discretization of (8) and (9) we use a classical Fourier pseudo-spectral method on a 2π-periodic domain Ω [6]. The vorticity field is transformed into Fourier space in order to compute the spatial derivatives and evolve the vorticity field in time. The non-linear and penalization terms are calculated in physical space. Following [4], the vorticity field and the other variables are represented as Fourier series,

$$\omega_\eta(\mathbf{x}, t) = \sum_{\mathbf{k} \in \mathbb{Z}^2} \widehat{\omega}_\eta(\mathbf{k}, t) e^{i\mathbf{k}\cdot\mathbf{x}}, \tag{10}$$

with $i^2 = -1$, and where the Fourier transform of ω_η is defined as

$$\widehat{\omega}_\eta(\mathbf{k}, t) = \frac{1}{4\pi^2} \int_\Omega \omega_\eta(\mathbf{x}, t) e^{-i\mathbf{k}\cdot\mathbf{x}} \, d\Omega, \tag{11}$$

with $\mathbf{k} = (k_x, k_y)$ and $\mathbf{x} \in [0, 2\pi]^2$ (general domains $[0, L_x] \times [0, L_y]$ may be considered by rescaling). The Fourier discretization is uniform in space and is truncated at $k_x = -N_x/2$ and $k_x = N_x/2 + 1$, $k_y = -N_y/2$ and $k_y = N_y/2 + 1$, where N_x and N_y are the number of grid points in x and y direction, respectively. The gradient of ω_η is computed by multiplication of $\widehat{\omega}_\eta$ by $i\mathbf{k}$, the Laplacian corresponds to multiplication by $-|\mathbf{k}|^2$. The velocity $\mathbf{u}$ is computed from the vorticity using Biot-Savart's law in Fourier space,

$$\mathbf{u}_\eta(\mathbf{x}, t) = -\sum_{\mathbf{k}\in\mathbb{Z}^2, \mathbf{k}\neq 0} \frac{i\mathbf{k}^\perp}{|\mathbf{k}|^2} \widehat{\omega}_\eta(\mathbf{k}, t) e^{i\mathbf{k}\cdot\mathbf{x}} + \mathbf{U}_\infty, \tag{12}$$

where $\mathbf{k}^\perp = (-k_y, k_x)$ and $\mathbf{U}_\infty$ is the potential part.

To avoid aliasing errors, *i.e.*, the production of small scales due to non-linear terms which are not resolved on the grid, we dealias the vorticity at each time step by truncating its Fourier coefficients using the 2/3 rule. For the transformation between physical and Fourier space we use the Fast Fourier transform (FFT) with an order of complexity $\mathbf{N}\log_2\mathbf{N}, \mathbf{N} = N_x N_y$ [6].

3. Time discretization

In this section first we describe the time integration of (8) using explicit treatment of the non-linear and penalization terms and exact integration of the diffusion term. We note that along with its simplicity and robustness this approach has a drawback: the explicit treatment of the penalization term imposes a stability condition $\Delta t < \eta$.

Then we investigate different ways to overcome this limitation. One approach consists in exponential propagation of the whole linear part of the momentum equations, including the penalization term, and not only of the diffusion term. In this case exact evaluation of exponential integrating factors is not practical, since diagonalization of the linear operator is no more straightforward. Hence, we consider various approximations [3,23,24]. Another way is to treat the penalization term implicitly, as proposed in [22].

In the following we choose four different time schemes, compare their performance when applied to a simplified 1D model problem, the penalized Burgers equation, and discuss numerous difficulties encountered in their generalization for the Navier–Stokes equations.

3.1. Adaptive second order Adams–Bashforth scheme

The basic time-stepping implemented in our code is an adaptive second order Adams–Bashforth method (AB2) with exact integration of the diffusion term and explicit integration of the penalization term [4]. All simulations in the present paper are performed using this scheme, unless otherwise indicated. It fits well into our general concept of compromise between the ease of implementation and computational efficiency.

Exact integration of the diffusion term is feasible because the Laplace operator is diagonal in Fourier space and hence no linear system has to be solved. It improves stability of the scheme, avoiding the stability condition $\Delta t < \Delta x^2/\nu$. The remaining terms are discretized explicitly to avoid the solution of non-linear equations, however it implies a CFL condition on the time step size Δt and also a condition due to the explicit discretization of the penalization term $\Delta t < \eta$, as linear stability analysis shows (see Appendix B).

To simplify notations let us rewrite the Eq. (8) in the form of a non-linear evolution equation

$$\partial_t \omega - \nu \nabla^2 \omega = N(\omega), \tag{13}$$

where $N(\omega) = -\mathbf{u} \cdot \nabla\omega - \nabla \times \left(\frac{\chi_\Omega}{\eta}(\mathbf{u} - \mathbf{u}_s)\right)$ and we dropped the index η for ease of notation.

Transforming the above into Fourier space we obtain the equation

$$\partial_t \widehat{\omega} + \nu|\mathbf{k}|^2 \widehat{\omega} = \widehat{N}(\omega). \tag{14}$$

For the initial condition $\widehat{\omega}(\mathbf{k}, t_n)$, Eq. (14) has the solution

$$\widehat{\omega}(\mathbf{k}, t_{n+1}) = e^{-\nu\Delta t_{n+1}|\mathbf{k}|^2} \widehat{\omega}(\mathbf{k}, t_n) + \int_{t_n}^{t_{n+1}} e^{-\nu(t_{n+1}-s)|\mathbf{k}|^2} \widehat{N}(\widehat{\omega}(\mathbf{k}, s)) ds. \tag{15}$$

The integral in (15) is discretized using the AB2 scheme with adaptive time steps, thus giving a fully discretized equation

$$\widehat{\omega}(\mathbf{k}, t_{n+1}) = e^{-\nu\Delta t_{n+1}|\mathbf{k}|^2} \left(\widehat{\omega}(\mathbf{k}, t_n) + \beta_{10}\widehat{N}^n + \beta_{11} e^{-\nu\Delta t_n|\mathbf{k}|^2} \widehat{N}^{n-1} \right), \tag{16}$$

with $\widehat{N}^n = \widehat{N}(\widehat{\omega}(\mathbf{k}, t_n))$ denoting the value of non-linear term at the time instant t_n and the Adams–Bashforth coefficients β_{10}, β_{11} being given by

$$\beta_{10} = \frac{1}{2} \frac{\Delta t_{n+1}}{\Delta t_n} (\Delta t_{n+1} + 2\Delta t_n), \quad \beta_{11} = -\frac{1}{2} \frac{\Delta t_{n+1}^2}{\Delta t_n}, \tag{17}$$

where $\Delta t_n = t_n - t_{n-1}$. For startup a first order scheme is used.

The time step size control is based on the CFL stability limit of the explicit discretization of the non-linear term. Therefore, at each time step t_n, the maximal pointwise velocity is computed:

$$U_{max} = \max_{\mathbf{x}\in Grid} |\mathbf{u}(\mathbf{x}, t_n)| \tag{18}$$

and the new time step is given by

$$\Delta t_{n+1} = C\Delta x / U_{max}, \tag{19}$$

where $C < 1$ is the CFL constant (we use $C = 0.1$) and $\Delta x = \min\left(\frac{L_x}{N_x}, \frac{L_y}{N_y}\right)$ is the minimal spatial grid size, with L_x, L_y denoting the length of the domain and N_x, N_y the number of points in x and y directions, respectively. Moreover, the time step is limited to verify the condition $\Delta t_{n+1} < \eta$.

3.2. *Exponential propagation and implicit treatment of the penalization term: application to Burgers equation*

As mentioned above the explicit treatment of the penalization term imposes a stability restriction $\Delta t < \eta$. It can be seen from (16) that when η tends to zero the terms $\widehat{N}^n, \widehat{N}^{n-1}$ become unbounded. To overcome this difficulty, several approaches have been proposed in the literature [3,23,22]. In this section, we briefly discuss them and study them, for the sake of simplicity, for a one-dimensional model problem.

Namely, we consider the viscous Burgers equation

$$\partial_t u + u\partial_x u - \nu\partial_{xx} u = 0 \tag{20}$$

for $x \in [-1, 1]$ and $t \in [0, t_{max}]$. We impose homogeneous Dirichlet boundary conditions

$$u(-1, t) = u(1, t) = 0 \tag{21}$$

and the initial condition given by

$$u(x, 0) = -\sin(\pi x). \tag{22}$$

To solve the above problem using the volume penalization method, the initial domain is imbedded into a larger domain $x \in [-2, 2]$, in which the penalized solution u_η is computed. We add a penalization term $\frac{\chi_\Omega}{\eta} u_\eta$ to Eq. (20) and obtain

$$\partial_t u_\eta + u_\eta \partial_x u_\eta - \nu\partial_{xx} u_\eta + \frac{\chi_\Omega}{\eta} u_\eta = 0, \tag{23}$$

where the mask function

$$\chi_\Omega = \begin{cases} 0 & \text{for } x \in]-1, 1[\\ 1 & \text{for } x \in [-2, -1] \bigcup [1, 2] \end{cases} \tag{24}$$

serves to model the Dirichlet conditions (21). Periodic boundary conditions at $x = -2$ and $x = 2$ are imposed to enable us using a Fourier pseudo-spectral method, and the initial condition (22) is extended periodically through the larger domain:

$$u_\eta(x, 0) = -\sin(\pi x). \tag{25}$$

Let us mention that in practice the penalized domain does not need to be twice as large as the initial domain, it is sufficient to cover the high-gradient boundary layer inside the solid obstacle with a few grid points. Here we choose this size only for illustrative purposes.

For $t > 0$, $\eta \to \infty$ the solution of (23)–(25) tends to Cole's exact solution of (20)–(22) [7] in $]-1, 1[$ and to zero in $[-2, -1] \bigcup [1, 2]$.

Numerical tests presented here are performed for $\nu = 0.01/\pi$, and the solution is computed up to $t_{max} = 1.6037/\pi$. It is discretized in space using $N = 2^{12}$ Fourier modes.

Since the following discussion is focused on time-stepping, spatial discretization errors are supposed to be sufficiently small. The equations in the following subsections should be read then in spectral collocation sense, with all the operators acting in finite-dimensional spaces.

3.2.1. *Explicit integration of the penalization term (EIP)*

We start with the second order Adams–Bashforth scheme where the penalization term is treated explicitly, as described in the previous subsection. Fig. 2 shows convergence of the approximate solution of (23)–(25) obtained with this scheme. Its L_∞ error versus time step is traced for four values of the penalization parameter: $\eta_1 = 1.6037 \cdot 2^{-7}/\pi$, $\eta_2 = 1.6037 \cdot 2^{-9}/\pi$, $\eta_3 = 1.6037 \cdot 2^{-11}/\pi$ and $\eta_4 = 1.6037 \cdot 2^{-13}/\pi$.

Let us first consider the curve η_1. The rightmost point of this curve corresponds to $\Delta t \approx 0.002$, which is smaller than $\eta_1 \approx 0.004$, thus the solution is stable. When Δt is decreasing to 0.001, the error is also decreasing, somewhat slower than $O(\Delta t^2)$. This is not surprising, since higher order truncation error terms are significant with these rather coarse time steps. These terms vanish for smaller time steps, but the penalization error (the error due to finite η) becomes dominant. It is independent of Δt and equals 0.0036.

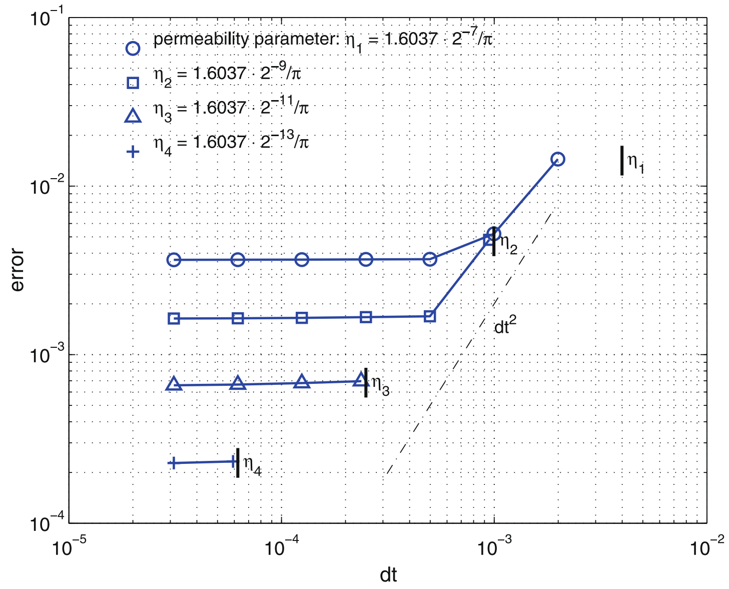

Fig. 2. Decay of the approximation error. AB2 scheme with explicit treatment of the penalization term (EIP). The L^∞-error between the exact and approximate solutions at $t_{max} = 1.6037/\pi$ is traced versus time step for a series of penalization parameter values. Vertical bars mark the stability restriction $\Delta t < \eta$.

Now we consider a smaller value of the penalization parameter, $\eta_2 \approx 0.001$. The penalization error decreases down to 0.0016, and it dominates the total error only when $\Delta t < 0.005$. For larger time steps the curve has a slope close to 2. However, large time steps $\Delta t > 0.001$ become impossible due to lack of stability.

For even smaller values of the penalization parameter, η_3 and η_4, the $O(\Delta t^2)$ part of the curves is absent. In some sense this means loosing the order of convergence of the temporal discretization. Therefore we can conclude that the explicit integration of the penalization term is only adequate for moderate values of η and Δt.

3.2.2. Exponential propagation of the penalization term (ExP)

One of the possible ways to overcome the above mentioned difficulty is to note that the penalization term in (23) is linear, so that we can follow the same steps as in the previous subsection to obtain

$$u_\eta(x, t_{n+1}) = e^{\Delta t_{n+1} A} u_\eta(x, t_n) + \int_{t_n}^{t_{n+1}} e^{(t_{n+1}-s)A} Q(u_\eta(x, s))\mathrm{d}s, \tag{26}$$

where $A = \nu\partial_{xx} - \frac{\chi_\Omega}{\eta}$ is the linear operator and $Q = -u_\eta \partial_x u_\eta$ is the non-linear term. However in this case, A does not yield a diagonal matrix in Fourier space, and the evaluation of its exact exponent becomes prohibitive.

Nevertheless it still can be evaluated approximately at a reasonable cost. This idea was exploited in [3], where the following approximations were proposed:

$$e^{tA} = \mathcal{F}^{-1} e^{tL} \mathcal{F} e^{-tP} + O(t^2) \tag{27}$$

or

$$e^{tA} = -I + \mathcal{F}^{-1} e^{tL} \mathcal{F} + e^{-tP} + O(t^2), \tag{28}$$

where $\mathcal{F}$ is the discrete Fourier transform (DFT) matrix, $\mathcal{F}^{-1}$ is its inverse, L is a matrix corresponding to the viscous term in Fourier space, P is the penalization matrix and I is the identity matrix. Note that L and P are both diagonal matrices: the former has values $-\nu|k|^2$ on its diagonal, with k being the wavenumbers; the diagonal elements of the latter are grid values of $\frac{\chi_\Omega(x)}{\eta}$. However, the matrix A is full, as well as its DFT.

Note that (27) or (28) injected into (26) yields a first order scheme only, since the error accumulates over time. We can obtain a second order scheme by following the same ideas. The time integral in (26) is approximated using the above mentioned AB2 scheme

$$u_\eta^{n+1} = e^{\Delta t_{n+1} A}(u_\eta^n + \beta_{10} Q^n + \beta_{11} e^{\Delta t_n A} Q^{n-1}) \tag{29}$$

and the exponentials are computed as

$$e^{tA} = e^{-tP/2} \mathcal{F}^{-1} e^{tL} \mathcal{F} e^{-tP/2} + O(t^3). \tag{30}$$

The approximation (30) is easily proved by developing its both sides into Taylor series.

The convergence of this scheme is illustrated in Fig. 3. The penalization parameter η does not restrict stability of the scheme. Moreover, it appears to be decoupled from the time step, *i.e.*, its change has no distinct influence on the truncation

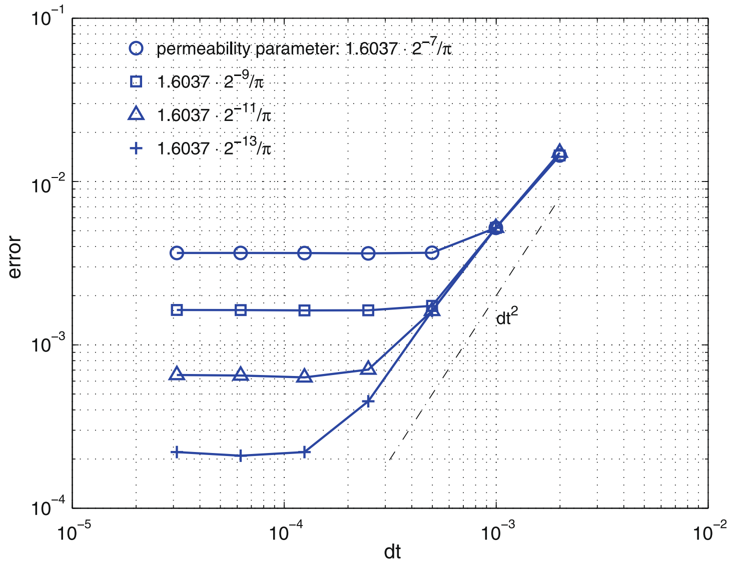

Fig. 3. Decay of the L^∞-error of (29) and (30) (ExP).

error of the time-stepping scheme. Each curve exhibits a Δt^2 decay and then a saturation when the time discretization error becomes smaller than the penalization error.

3.2.3. Krylov method (KrM)

Krylov space methods are an alternative way to approximate the desired exponential (see [23,24]). Let u be a vector of N components, real or complex-valued, and A be an $N \times N$ matrix. The following procedure gives an approximation of $e^{tA}u$ to order t^K, where t is a real parameter (time) and K is a positive integer, which is typically smaller than N. The K-dimensional Krylov subspace is defined as a linear span of vectors $\{u, Au, \ldots, A^{K-1}u\}$. Its orthonormal basis $\{v_1, v_2, \ldots, v_K\}$ is generated by the Arnoldi process [26]:

- $w_1 = u$
- For $j = 1, 2, \ldots, K$

$$\begin{aligned} v_j &= w_j / \|w_j\|, \\ w_{j+1} &= A v_j - \sum_{l=1}^{j} v_l (v_l, A v_j). \end{aligned} \tag{31}$$

This process results in a $N \times K$ matrix V whose columns are $v_1, \ldots, v_K$ and a $K \times K$ matrix $H = V^T A V$, whose elements are $H_{lj} = (v_l, A v_j)$ and which is an upper Hessenberg matrix. We find then e^{tH} by a series approximation with scaling and squaring as described in [24].

Finally, we compute

$$e^{tA} \approx V e^{tH} a_1 \|u\|, \tag{32}$$

where a_1 is the first unit vector of length K.

The above procedure is used for computing the exponentials both of the linear and non-linear terms of (26). The convolution integral arising from the non-linear term is evaluated using the previously described Adams–Bashforth discretization (*cf.* Eqs. (15) and (16)), yielding a second order scheme. Note that this approach differs from the one reported in [23], where quadrature formulae are used with exponential propagation being applied to each of the quadrature weights.

The Krylov subspace dimension K should be set to 3 or higher to achieve second order convergence of (29). Fig. 4 shows two curves, corresponding to $K = 3$ and $K = 20$. Both of them indicate an error of the approximate solution decreasing with Δt^2, provided Δt is small enough. However, the former one is several orders of magnitude higher for coarse discretizations.

An adequate choice of K is a non-trivial task, since it affects the computational cost. In addition, it depends on the parameters of the problem.

3.2.4. Implicit integration of the penalization term (IIP)

So far we considered exponential propagation of the linear terms. It was derived under the assumption that the penalization operator is linear and constant in time. This appears false in case of moving obstacles, as one can see from the penalization

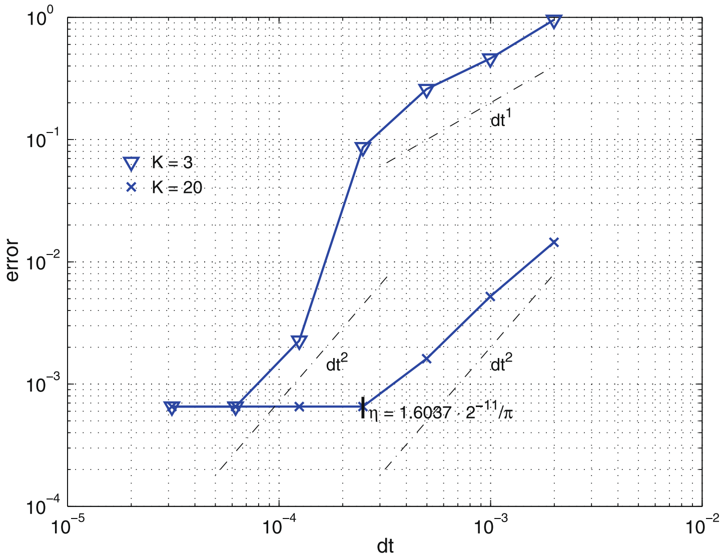

Fig. 4. Decay of the L^∞-error of the Krylov method (KrM) for $K = 3$ and $K = 20$. Penalization parameter is $\eta = 1.6037 \cdot 2^{-11}/\pi$.

term in (4). Therefore it is worthwhile considering another approach which consists in an implicit integration of the penalization term.

Such a scheme was proposed in [22]. When applied to (23) it reads

$$\begin{aligned}\widehat{u}_\eta^{n+1} = \mathcal{F}\Bigg[\mathcal{F}^{-1}\Bigg\{&3e^{-\nu\Delta t|k|^2}\widehat{u}_\eta^n - \frac{3}{2}e^{-2\nu\Delta t|k|^2}\widehat{u}_\eta^{n-1} + \frac{1}{3}e^{-3\nu\Delta t|k|^2}\widehat{u}_\eta^{n-2} \\ &+\Delta t(3e^{-\nu\Delta t|k|^2}\widehat{Q}^n - 3e^{-2\nu\Delta t|k|^2}\widehat{Q}^{n-1} + e^{-3\nu\Delta t|k|^2}\widehat{Q}^{n-2})\Bigg\}\Bigg/\left(\frac{11}{6} + \Delta t\frac{\chi_\Omega(t^{n+1})}{\eta}\right)\Bigg],\end{aligned} \tag{33}$$

where $\widehat{Q}^n = \mathcal{F}(-u_\eta^n \cdot \partial_x u_\eta^n)$, $\widehat{u}_\eta^n = \mathcal{F}(u_\eta^n)$ and $\mathcal{F}$ stands for the Fourier transform. Note that in this scheme the time step Δt is fixed.

This semi-implicit backward differentiation scheme of third order requires values of $\widehat{u}_\eta^0, \widehat{u}_\eta^1$ and $\widehat{u}_\eta^2$ for its startup. The first is given by the initial condition at $t = 0$: $\widehat{u}_\eta^0 = \mathcal{F}\{u_\eta(x,0)\}$. To obtain the next two we use a second order semi-implicit Runge–Kutta scheme

$$\begin{aligned}\widehat{u}_\eta^I &= \mathcal{F}\left[\mathcal{F}^{-1}\{e^{-\nu\Delta t|k|^2}(\widehat{u}_\eta^n + \Delta t\widehat{Q}^n)\}/\left(1 + \Delta t\frac{\chi_\Omega(t^{n+1})}{\eta}\right)\right]; \\ \widehat{u}_\eta^{II} &= \mathcal{F}\left[\mathcal{F}^{-1}\{e^{-\nu\Delta t|k|^2}(\widehat{u}_\eta^I + \Delta t\widehat{Q}^I)\}/\left(1 + \Delta t\frac{\chi_\Omega(t^{n+2})}{\eta}\right)\right]; \\ \widehat{u}_\eta^{n+1} &= \frac{1}{2}(e^{-\nu\Delta t|k|^2}\widehat{u}_\eta^n - e^{\nu\Delta t|k|^2}\widehat{u}_\eta^{II}) + \widehat{u}_\eta^I + \Delta t\widehat{Q}^I.\end{aligned} \tag{34}$$

In absence of penalization, it reduces to Heun's scheme. Convergence of order higher than 2 is seen in Fig. 5, though the curves do not reach the Δt^3 slope before the penalization error lines up with the spatial discretization error.

For Burgers equation the scheme (33) requires no additional FFTs compared with an explicit treatment of the penalization term.

3.3. Exponential propagation and implicit treatment of the penalization term. Extension to Navier–Stokes equations

Studying a simplified problem helped us to outline different possibilities to overcome a stability restriction inherent to the penalization method. Apparently, time integration of the Navier–Stokes equations originates some extra points to be discussed.

3.3.1. Vorticity–streamfunction form versus velocity–pressure form

Incompressible Navier–Stokes equations can be formulated in a number of ways, which are equivalent but whose properties differ from the numerical point of view. In particular, there is an essential difference between vorticity–streamfunction '$\omega - \psi$' and velocity–pressure '$u - p$' formulations for 2D flows, since the former requires less storage and less computation.

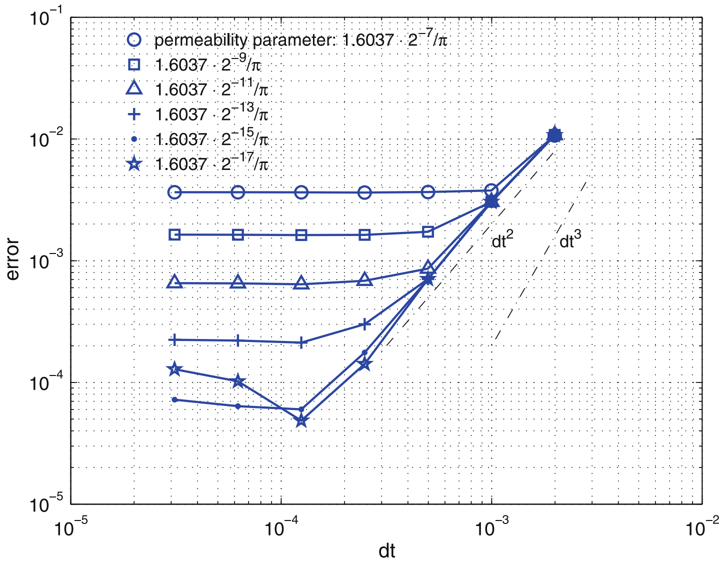

Fig. 5. Decay of the L^{∞}-error of (33) and (34) (IIP).

This argument still holds for the penalized equations when an explicit scheme is used (*e.g.* the EIP method described in the previous subsection). Meanwhile these too formulations imply two different forms of the penalization term:

- $\frac{\chi_{\Omega}}{\eta}(\mathbf{u}_{\eta} - \mathbf{u}_s)$ for the '$u - p$' (4);
- $\nabla \times \left(\frac{\chi_{\Omega}}{\eta}(\mathbf{u}_{\eta} - \mathbf{u}_s)\right)$ for the '$\omega - \psi$' (8).

It may seem that the latter formulation is disadvantageous since it contains derivatives of a discontinuous function. In this connection let us note that the former one is a discontinuous function itself, and for spectral methods both will produce Gibbs oscillations. Thereby it was explored in [9] that this truncation error affects neither flow dynamics, nor forces exerted on solid bodies. Moreover, there exists a number of post-processing techniques to filter out oscillations (see also the related discussion in [22]).

A more important difference lies in the fact that the penalization matrix corresponding to '$u - p$' is diagonal in physical space, while it is not true for '$\omega - \psi$'. As a consequence, generalization of the methods described in the previous subsection is not straightforward. Namely, the IIP method fails for '$\omega - \psi$', and approximations of matrix exponents (27) or (30) in the ExP method do not hold. In this context a first order scheme for (14) may be obtained by generalization of (28).

$$\widehat{\omega}^{n+1} = -ik_x\mathcal{F}e^{-\Delta t\frac{\chi_{\Omega}}{\eta}}\left(\mathcal{F}^{-1}\frac{ik_x}{|\mathbf{k}|^2}\widehat{\omega}^n - U_{\infty y}\right) - ik_y\mathcal{F}e^{-\Delta t\frac{\chi_{\Omega}}{\eta}}\left(\mathcal{F}^{-1}\frac{ik_y}{|\mathbf{k}|^2}\widehat{\omega}^n + U_{\infty x}\right) + e^{-\nu\Delta t|\mathbf{k}|^2}\widehat{\omega}^n - \widehat{\omega}^n + \Delta te^{-\nu\Delta t|\mathbf{k}|^2}\widehat{Q}(\widehat{\omega}^n), \tag{35}$$

where the non-linear term $Q = -\mathbf{u} \cdot \nabla\omega$ is evaluated in physical space. Here we use the notation $\widehat{\omega}^n = \widehat{\omega}(\mathbf{k}, t_n)$ and assume $\mathbf{u}_s = 0$. This first order method requires an additional pair of FFTs compared with the explicit scheme. An analogous second order scheme will use even more FFTs.

On the contrary, the KrM algorithm can be implemented without major modification for any formulation, since it only requires a consequent application of the linear operator, whatever its form is. The choice of formulation may influence the appropriate choice of the Krylov subspace dimension K, but in both cases K should be set high enough to suppress coupling between η and Δt. In this view the Krylov method represents a flexible, but computationally not very efficient tool, as stated in [23].

3.3.2. *Imposing incompressibility*

The '$u - p$' formulation (4) is naturally convenient for implicit or exact integration of the penalization term, and for extending the current approach to three spatial dimensions. The evaluation of the pressure gradient reduces to imposing the divergence free condition $\nabla \cdot \mathbf{u}_{\eta} = 0$ on the velocity using, *e.g.* the fractional step method. In this connection it is important to note that the penalization term in (4) is not divergence free, and makes hence its contribution to the pressure gradient.

3.3.3. *Imposing the mean flow*

In the '$u - p$' formulation, a constant upstream flow can be imposed by setting the zero wavenumber mode of the velocity to a desired value. This action is more subtle for the '$\omega - \psi$' formulation, since it introduces a bias in the penalization term, and then the corresponding particular solution should be incorporated.

Hence, KrM will yield the following time-stepping scheme:

$$\widehat{\omega}^{n+1} = \widehat{\omega}^n + (e^{\Delta t^{n+1}A} - I)A^{-1}(A\widehat{\omega}^n + b) + e^{\Delta t^{n+1}A}(\beta_{10}\widehat{Q}(\widehat{\omega}^n) + \beta_{11}e^{\Delta t^n A}\widehat{Q}(\widehat{\omega}^{n-1})), \tag{36}$$

where

$$\begin{aligned} A &= ik_x\mathcal{F}\frac{\chi_\Omega}{\eta}\mathcal{F}^{-1}\frac{ik_x}{|\mathbf{k}|^2} + ik_y\mathcal{F}\frac{\chi_\Omega}{\eta}\mathcal{F}^{-1}\frac{ik_y}{|\mathbf{k}|^2} - \nu|\mathbf{k}|^2, \\ A\widehat{\omega}^n + b &= ik_x\mathcal{F}\frac{\chi_\Omega}{\eta}\left(\mathcal{F}^{-1}\frac{ik_x}{|\mathbf{k}|^2}\widehat{\omega}^n - U_{\infty y}\right) + ik_y\mathcal{F}\frac{\chi_\Omega}{\eta}\left(\mathcal{F}^{-1}\frac{ik_y}{|\mathbf{k}|^2}\widehat{\omega}^n + U_{\infty x}\right) - \nu|\mathbf{k}|^2\widehat{\omega}^n, \end{aligned} \tag{37}$$

and the operators $e^{\Delta t^n A}, e^{\Delta t^{n+1}A}$ and $(e^{\Delta t^{n+1}A} - I)A^{-1}$ are evaluated using an approximation in a Krylov subspace.

3.3.4. Moving obstacles

In the context of the penalization method 'moving obstacle' means that in Eq. (4)

- the mask function χ depends on time;
- the term u_s is non-zero and, in general, also depends on time.

The first point has an important consequence for the ExP and KrM approaches, since they require the linear operator to be constant in time. It can be shown that for time-dependent linear operators the accuracy of these methods reduces to first order.

The IIP method does not invoke exact integration of the penalization term, and therefore it does not encounter this problem. For Eq. (4) we generalize the IIP method to obtain the following scheme

$$\widehat{\mathbf{u}}_\eta^{n+1} = \mathcal{F}\left\{\mathbf{u}_s^{n+1} + \left[\mathcal{F}^{-1}\left(3e^{-\nu\Delta t|\mathbf{k}|^2}\widehat{\mathbf{u}}_\eta^n - \frac{3}{2}e^{-2\nu\Delta t|\mathbf{k}|^2}\widehat{\mathbf{u}}_\eta^{n-1} + \frac{1}{3}e^{-3\nu\Delta t|\mathbf{k}|^2}\widehat{\mathbf{u}}_\eta^{n-2} + \Delta t\mathcal{Q}\right) - \frac{11}{6}\mathbf{u}_s^{n+1}\right]\Big/\left(\frac{11}{6} + \Delta t\frac{\chi_\Omega(t^{n+1})}{\eta}\right)\right\}, \tag{38}$$

where $\mathcal{Q} = 3e^{-\nu\Delta t|\mathbf{k}|^2}\widehat{\mathbf{Q}}^n - 3e^{-2\nu\Delta t|\mathbf{k}|^2}\widehat{\mathbf{Q}}^{n-1} + e^{-3\nu\Delta t|\mathbf{k}|^2}\widehat{\mathbf{Q}}^{n-2}$ and $\mathbf{Q} = -\mathbf{u}\cdot\nabla\mathbf{u}$.

We should also mention in this connection that in principle the 'moving obstacle' problem can be reduced to the 'fixed obstacle' problem by rewriting the governing equations in a moving reference frame. This approach, however, contradicts to the essential ideas of this work and restricts capabilities of the proposed model (for instance, it is not quite suitable for two moving obstacles).

3.3.5. Conclusions on time-stepping schemes

Above we listed numerous difficulties related to implicit and exact integration of the penalization term, and we found few advantages with respect to the original explicit scheme described in Section 3.1. In our opinion, these approaches need further development in order to be used for moving obstacle simulations without loosing efficiency. All the results presented in the following are obtained using the explicit AB2 scheme.

4. Moving obstacles: time-dependent penalization

An attractive feature of the volume penalization method is that it potentially offers a capability of moving and deforming solid obstacles while keeping unchanged the computational mesh. To do this in practice, we need to take care of the time-dependent penalization term.

Note that in (8) both the obstacle penalization mask $\chi_\Omega(\mathbf{x},t)$ and the velocity $\mathbf{u}_s$ depend on time and on the spatial coordinate. As long as the obstacle is rigid, they are fully described by the coordinates of the obstacle's center of gravity (c.g.) $\mathbf{x}_{cg}(t) = (x_{cg}(t), y_{cg}(t))$, its translational velocity $\dot{\mathbf{x}}_{cg}(t)$, the angle of inclination of the obstacle $\alpha_{cg}(t)$, its rotational velocity $\dot{\alpha}_{cg}(t)$, and the initial mask function χ_Ω at $t = 0$:

$$\begin{aligned} &\mathbf{u}_s = (u_{sx}, u_{sy}), \\ &\text{where} \\ &u_{sx}(\mathbf{x},t) = \dot{x}_{cg}(t) - \dot{\alpha}_{cg}(t)\cdot(y - y_{cg}(t)), \\ &u_{sy}(\mathbf{x},t) = \dot{y}_{cg}(t) + \dot{\alpha}_{cg}(t)\cdot(x - x_{cg}(t)), \end{aligned} \tag{39}$$

$$\begin{aligned} &\chi_\Omega(\mathbf{x},t) = \chi_\Omega(\mathbf{x}',0), \\ &\text{where} \\ &\mathbf{x}' = (x', y'), \\ &x' = (x - x_{cg}(t))\cos\alpha_{cg}(t) + (y - y_{cg}(t))\sin\alpha_{cg}(t) + x_{cg}(0), \\ &y' = -(x - x_{cg}(t))\sin\alpha_{cg}(t) + (y - y_{cg}(t))\cos\alpha_{cg}(t) + y_{cg}(0), \end{aligned} \tag{40}$$

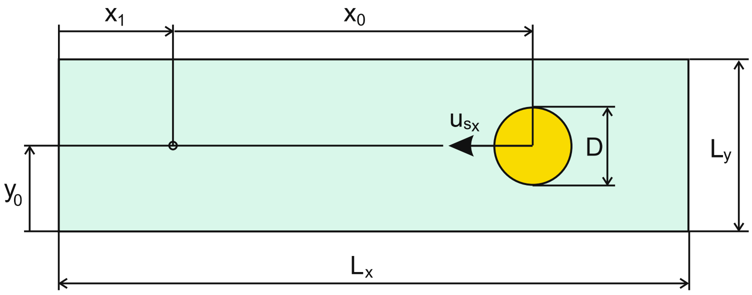

Fig. 6. Computational domain of a moving cylinder.

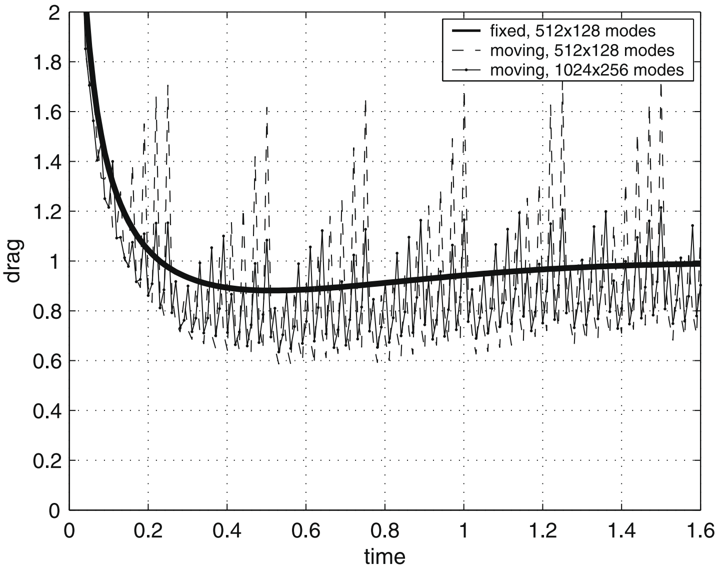

Fig. 7. Drag of a moving cylinder. Penalization mask is translated in physical space.

The numerical solution of (8) and (9) will require a suitable discretization of (39) and (40) and, consequently, two-dimensional interpolation of the mask function χ_Ω.

4.1. Discretization in physical space

We first imagine an obstacle moving along the x-axis (see Fig. 6). At the time instant t_1 its center of gravity is translated from its initial position (x_0, y_0) to the point (x_1, y_0). The initial mask function $\chi_\Omega(\mathbf{x}, t = 0)$ is given at the grid points. Then we can obviously approximate $\chi_\Omega(\mathbf{x}, t_1)$ by shifting the elements of the initial mask matrix by N_{dx} cells along x-direction, with

$$N_{dx} = \text{modulo}([(x_1 - x_0)/\Delta x], N_x), \tag{41}$$

where N_x is the number of grid points in x and Δx is the grid step along x. The penalization velocity field $\mathbf{u}_s$ is simply given by

$$\mathbf{u}_s(\mathbf{x}, t) = (u_{sx}, 0) \tag{42}$$

for all $\mathbf{x} = (x, y) \in [0, L_x] \times [0, L_y]$.

Hereafter this approach is called 'moving penalization mask in physical space'. In order to verify its validity, let us consider a cylinder impulsively starting and moving in a fluid at rest with constant speed $u_{sx} = 1$ along the x-axis. Ideally, hydrodynamic forces must be invariant to the Galilean change of reference frame. Fig. 7 displays the evolution of the drag coefficient, computed in such simulations for three different levels of spatial resolution. The domain size is $L_x \times L_y = 16 \times 4$, the diameter of the cylinder $D = 1$, the viscosity $\nu = 10^{-2}$ and the penalization parameter $\eta = 10^{-3}$. The drag of a fixed cylinder in a uniform flow is plotted for reference. These curves reveal a problem of the present numerical method: oscillations of the hydrodynamic forces. With an increase in resolution they diminish but still remain unacceptable.

The origin of these oscillations can be explained as follows. The volume penalization method implies that the obstacle is slightly permeable, and the whole domain is occupied by the fluid. A vortical boundary layer is formed both outside and inside of the body. Let us consider a solid body which is fixed, and the fluid flowing around it. At each time step the boundary layer tends to be advected inside the body within a distance proportional to Δt, the time step. For numerical stability this distance should be smaller (and in practice much smaller) than the spatial grid sizes Δx, Δy. On the contrary, when the body moves in a fluid at rest, using the above method it can only be translated by an integer number of grid points, thus violating the CFL condition.

In the next subsection we discuss a way to simulate 'smoother' motion of obstacles and to avoid oscillations of hydrodynamic forces caused by the mentioned jerky motion.

4.2. Discretization in Fourier space

The aim of this subsection is to find a way of translating the obstacles by an increment as small as desired, and not limited by the spatial grid resolution. This can be done basing on the fact that in our model the mask function is approximated by a smooth periodic function given by a finite number of Fourier coefficients. So the shift of $\chi_\Omega(\mathbf{x}, t)$ in physical space corresponds to multiplication of its Fourier transform by a complex exponential factor. We consider again an obstacle moving along the x-axis with a constant speed V, so that after a time δt its displacement will be $\delta x = V\delta t$. The corresponding mask function will be

$$\chi_\Omega(x,t) = \chi_{\Omega 0}(x - \delta x), \tag{43}$$

where $\chi_{\Omega 0}(x)$ is the initial mask function given at $t = 0$. Performing a Fourier transform of (43) in x we obtain

$$\hat{\chi}_\Omega(k,t) = e^{-ik\delta x}\hat{\chi}_{\Omega 0}(k), \tag{44}$$

where k is the wavenumber.

A similar relation may be used to approximate the mask function $\chi_\Omega(x_i, t)$ at discrete points $x_i, i = 1, \ldots, N$ and $t > 0$, from the discrete Fourier coefficients of the initial mask at $t = 0$. In this case the wavenumbers k are discrete and bounded, and initially at $t = 0$, the mask function is given by its grid point values $X_{\Omega 0} = \chi_{\Omega 0}(x_i)$. So we obtain an expression to compute the discrete mask function X_Ω

$$X_\Omega = \mathcal{F}^{-1}(e^{-ik\delta x^n} \cdot \mathcal{F}(X_{\Omega 0})), \tag{45}$$

with $\delta x^n = Vt^n$.

By construction this approximation converges in L^2 to (43) with increasing number of modes taken into consideration.

The discussed algorithm has been applied to the moving cylinder test case from the previous subsection. Fig. 8 shows the evolution of drag in this case, and should be compared with Fig. 7. Oscillations are still present, but their amplitude is much smaller than observed previously in the case of translation in physical space. We can also see convergence of drag with

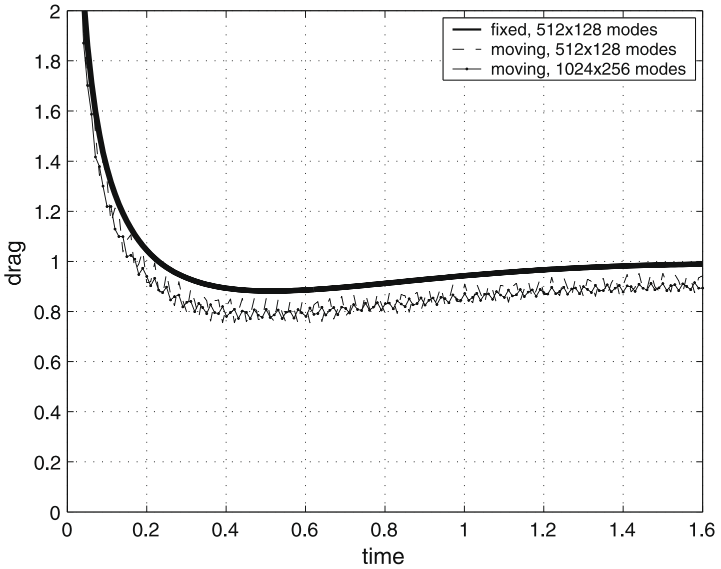

Fig. 8. Drag of a moving cylinder. Translation is modelled by changing the Fourier coefficients of the mask function.

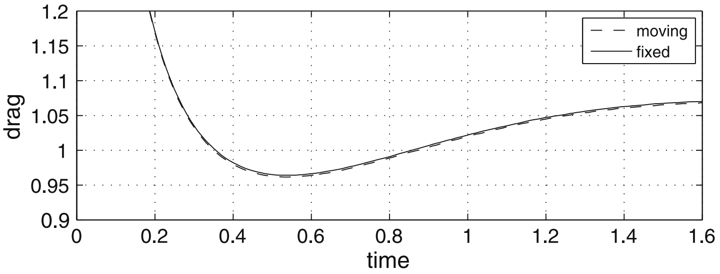

Fig. 9. Drag of fixed and moving cylinders modelled with a smooth mask function.

increasing resolution, but the limit seems to differ slightly from the value corresponding to a fixed cylinder. The latter fact may be explained by the Gibbs oscillations of the mask function, which are not present in the fixed cylinder case (even if the vorticity fields exhibit some small oscillations in both cases).

Oscillations disappear when using a smooth mask function rather than a discontinuous one, as displayed in Fig. 9. Therefore we can conclude that smoothing, *i.e.*, constructing a smooth mask function from the initial discontinuous mask function, is important for the present moving obstacles model. This can be done by solving a heat equation with the discontinuous mask function taken as an initial condition. Details of this process are described in Appendix A. A permeable layer which is generated around the obstacle can be interpreted as wall roughness, according to [9].

Noteworthily, the two curves in Fig. 9 are not exactly identical even for very smooth mask functions. This is due to different time-stepping truncation errors in these two cases. The results shown in Fig. 9 are obtained with the CFL number $C = 0.01$, which yielded $\Delta t \approx 10^{-4}$. For $\Delta t = 10^{-3}$ the difference increases to 2%.

The discussed approach of modelling motion of an obstacle by manipulating its mask function in Fourier space can be generalized to perform arbitrary motion of a solid. In 2D, we need to shift the mask function by distances δx and δy along x and y respectively, and rotate it about a given point (the solid c.g.) in the xy-plane by an angle $\delta\theta$.

Computational costs of this Fourier interpolation can be significant. At this point it is useful to remark that often, and typically when external flows are considered, the obstacles are several times smaller than the computational domain. Thus the Fourier interpolation can be performed at little cost within a small box bounding the obstacle, and then this box is shifted by an integer number of grid points, as described in the previous subsection.

4.3. Rotation and arbitrary two-dimensional motion

Rotation, just like translation, requires the values of the mask function to be interpolated between the grid points. The difference is that rotation is a 2D operation. Nevertheless, it can be decomposed into a sequence of 1D transformations along x and y axes. These decompositions are well known in image processing. We adopt the three-pass rotation, which is both easy-to-implement and efficient [11]. It makes use of the rotation matrix factorization:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{bmatrix}, \tag{46}$$

which decomposes the rotation by an angle θ into skewing in the x and y directions. Each row (column) is translated by an offset $\delta x = -y\tan(\theta/2)(\delta y = x\sin(\theta))$ that is proportional to its vertical (horizontal) coordinate.

The above decomposition can be used for $\theta \in]-\pi, \pi[$. For θ outside of this interval one can decompose it as

$$\theta = \theta_1 + n\pi, \tag{47}$$

with $n \in \mathbb{N}$ and $\theta_1 \in]-\pi, \pi[$, and mirror the mask function instead of rotating it at $n\pi$. The latter corresponds to taking the complex conjugate in Fourier space.

Combining the described translation and rotation we obtain Algorithm 1 to compute the discretized mask function corresponding to a solid obstacle in arbitrary 2D motion.

It is worth mentioning an alternative algorithm that has been developed for the same problem of arbitrary solid motion, and requires only one pair of FFT in each direction. The algorithm is based on an approximate factorization of the rotation matrix

$$R(\theta) = \begin{bmatrix} 1 & -\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \theta & 1 \end{bmatrix} + O(\theta^2) \tag{48}$$

and yields a first order scheme when applied to a small increment of the rotation angle. Nonetheless, all of the present results in this paper are obtained with the exact factorization (46), for the sake of accuracy.

Algorithm 1. Arbitrary 2D motion of the penalization mask function

```
Require: Coordinates of the obstacle $x_2 = x_{cg}(t_2), y_2 = y_{cg}(t_2), \alpha_2 = \alpha_{cg}(t_2)$ at the current time instant $t_2$, and $x_0 = x_{cg}(0)$,
         $y_0 = y_{cg}(0), \alpha_0 = \alpha_{cg}(0)$ at the initial time $t = 0$, as well as the initial penalization mask matrix $X_0$ containing the
         grid point values of the penalization mask function $\chi_\Omega(\mathbf{x}, 0)$. The fluid domain occupies a rectangle with coordinates
         $[0, L_x] \times [0, L_y]$.
Step 1: Find $\theta_1$ satisfying (47).
Step 2: Perform an FFT of $X_0$ in $x$-direction. Multiply the result by
        $e^{-ik_x(L_x/2-x_0)}$
        (translation of c.g. from $x_0$ to the center of domain along $x$). If $\cos(\theta) < 0$, perform complex conjugate (mirror about
        $y$-axis). Then multiply by
        $e^{\pm ik_x(y-y_0)\tan(\theta_1/2)}$
        (shear along $x$-direction, "–" sign corresponds to the case when $\cos(\theta) < 0$). Perform an inverse FFT in $x$-direction.
Step 3: Perform an FFT of $X_0$ in $y$-direction. Multiply the result by
        $e^{-ik_y(L_y/2-y_0)}$
        (translation of c.g. from $y_0$ to the center of domain along $y$). If $\cos(\theta) < 0$, perform complex conjugate (mirror about
        $x$-axis). Then multiply by
        $e^{-ik_y(x-L_x/2)\sin\theta_1}$
        (shear along $y$-direction). Perform an inverse FFT in $y$-direction.
Step 4: Perform an FFT in $x$-direction. Multiply by
        $e^{-ik_x(x_2-L_x-(y-L_y/2)\tan(\theta_1/2))}$
        (translation of c.g. from the center of domain to $x_2$ and shear along $x$-direction). Perform an inverse FFT in $x$-
        direction.
Step 5: Perform an FFT of $X_0$ in $y$-direction. Multiply the result by
        $e^{-ik_y(y_2-L_y/2)}$
        (translation of c.g. from the center of domain to $y_2$). Perform an inverse FFT in $y$-direction.
Ensure: Discretized penalization mask $X_2$ approximating $\chi_\Omega(\mathbf{x}, t_2)$ at $t_2$.
```

Example: To see Algorithm 1 working, let us consider an assembly of three plates with rectangular cross-section. Their dimensions are 0.35×0.05 in terms of the unit length. One end of each is located on a circle of radius 0.5, the other end is on a circle of radius 0.15, and the angle between the plates equals $2\pi/3$. The three plates are rotating as a single unit with an angular velocity $\dot{\alpha}_{cg} = \pi/2$ (counter-clockwise) about an axis which is moving in the x-direction with a speed $\dot{x}_{cg} = -1$ (to the left).

The domain dimensions are 6×3 and the resolution is 2048×1024. The Reynolds number based on the unit length and speed is $Re = 500$.

In Fig. 10, one can see four snapshots of the vorticity produced by the plates at four different time instants. The plates are coloured in grey, the colour which corresponds to their constant angular velocity. One can see vortices shedding from the edges of the plates. The strongest vortices are generated by a plate which is on the top, because it moves faster. Then they collide with the next plate which is approaching, as it can be observed at $t = 2.3$ or $t = 3.8$.

Gibbs oscillations are not present, due to smoothing of the mask function. It should be noted that this procedure slightly changes the edge of the plate, so it will affect the properties (intensity, stability, etc.) of the vortex shedding (see [10] on the effect of the edge geometry in this kind of numerical simulations).

Fig. 11 displays the resultant instantaneous forces and moment exercised by the plates, where positive y direction is upwards, positive x is to the right and positive moment is counter-clockwise. In spite of the complexity of the flow, there are no spurious oscillations of the hydrodynamic forces, and this is the essential conclusion of the present example.

In the present section we have dealt with the problem of modelling the flow around solid abstacles in arbitrary prescribed 2D motion. Using the algorithm based on Fourier transforms of the obstacle mask function we attain a smooth evolution of its hydrodynamic forces. The algorithm is relatively straightforward and easy-to-implement. It requires eight one-dimensional FFTs in a rectangular sub-domain which bounds the solid obstacle.

4.4. Couette flow between cylinders

Our main motivation in this subsection is to show the validity of the present model when the obstacles are rotating. We consider two co-axial cylinders, having radii $R_1, R_2 (R_1 < R_2)$, and rotating with angular velocities Ω_1, Ω_2. There exists an analytical solution of the Navier–Stokes equations for a steady flow between the cylinders (*e.g.* see [27]):

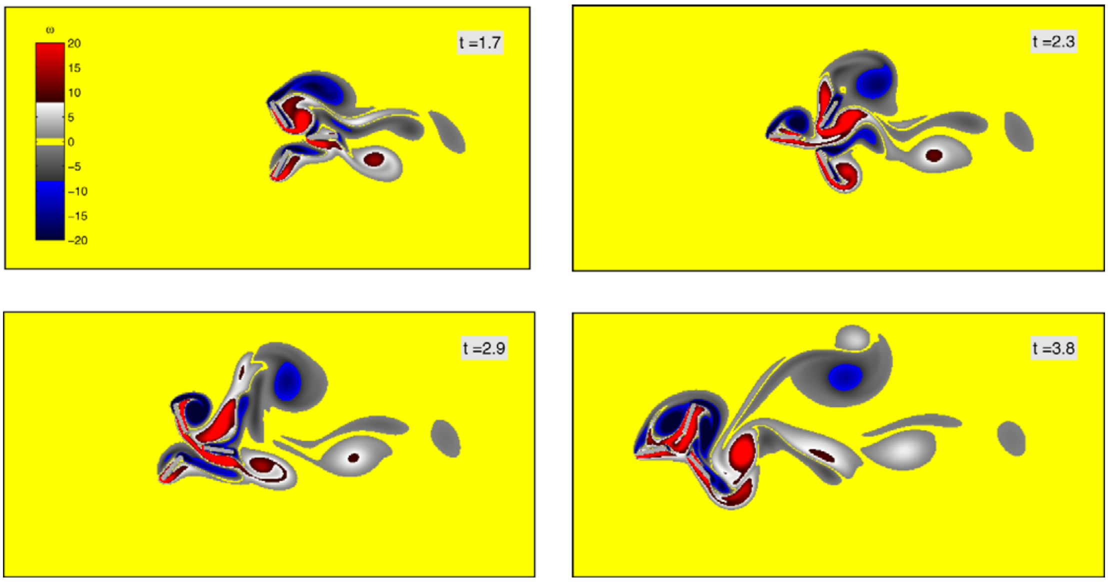

Fig. 10. Plots of vorticity produced by the moving plates assembly at subsequent time instants.

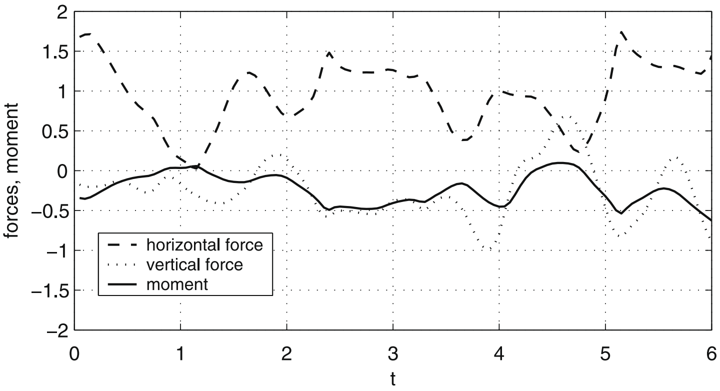

Fig. 11. Forces and moment of the moving plates.

$$u(r) = \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2} r + \frac{(\Omega_1 - \Omega_2) R_1^2 R_2^2}{R_2^2 - R_1^2} \frac{1}{r}, \tag{49}$$

where u is the tangential component of the velocity, and the other components are zero.

For our numerical tests we set $R_1 = 0.4, R_2 = 1.0, \Omega_1 = 1.25, \Omega_2 = 0$. The fluid viscosity, although not present in (49), is important for the numerical solution. Here it equals $\nu = 0.1$. The domain is $[-1.1, 1.1] \times [-1.1, 1.1]$, hence the mask function is

$$\chi_\Omega(\mathbf{x}) = \begin{cases} 0 & \text{for } R_1^2 < x^2 + y^2 < R_2^2, \\ 1 & \text{elsewhere.} \end{cases} \tag{50}$$

Since the mask function does not change in time, the interpolation described above is not applied for this test. However, the penalization velocity $\mathbf{u}_s = (-\Omega_1 y, \Omega_1 x)$ is non-zero, since it models the non-homogeneous Dirichlet boundary condition for the fluid velocity. This makes a difference with respect to the fixed obstacle case. The penalization parameter is set to $\eta = 10^{-5}$, so that the penalization error is expected to be smaller then those appearing from the discretization.

Fig. 12 shows a cut of the azimuthal velocity for $x \in [0, 1.1]$ at $y = 0$. The numerical solution is computed with $N^2 = 256^2$ grid points and corresponds to time $t = 6$, when the steady-state is well established (for $t > 6$ the velocity change is less then 10^{-3}). It is practically equal to u_s within the solid domain $x \in [0, 0.4] \bigcup [1, 1.1]$. Within the fluid domain $x \in]0.4, 1[$ it is close to the exact solution, but a difference is visible, which has its maximum near the edge of the inner cylinder.

Fig. 13 displays the L^2-error between the approximate solution and the exact one as a function of the number of grid points, *i.e.*, for decreasing grid size. Simulations with and without smoothing of the mask function are compared. Both curves indicate first to second order of convergence. The smooth mask function results in a monotonous error decay, which is not

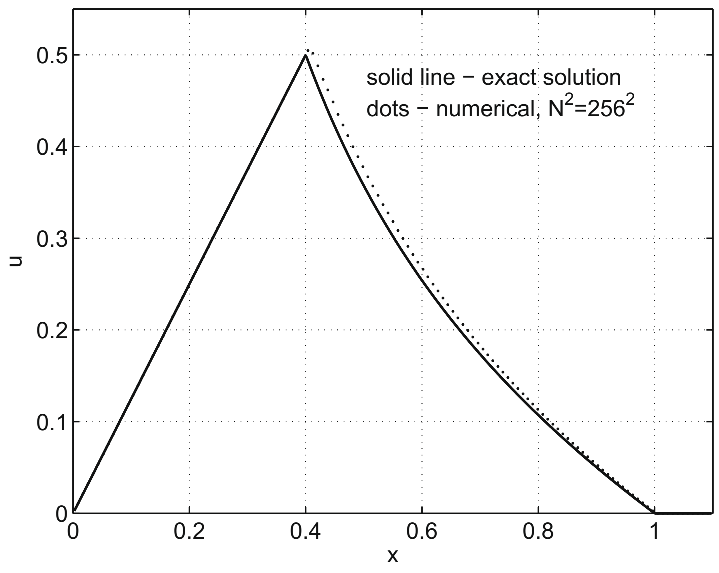

Fig. 12. Couette flow between cylinders: cuts of the azimuthal velocity at $y = 0$.

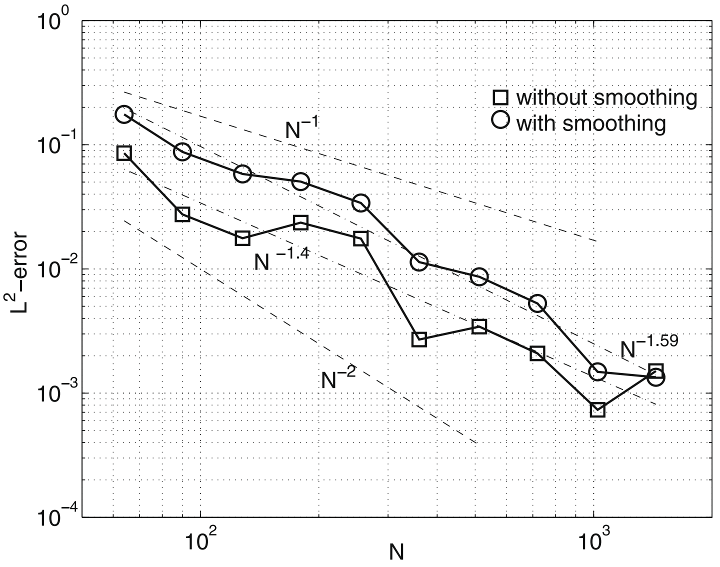

Fig. 13. L^2-error decay in the Couette flow simulations.

the case with the discontinuous mask. These oscillations arise from the approximation of the interface with a cartesian grid (*cf.* the related discusion in [28]). Meanwhile, smoothing also increases the error.

The first to second order of convergence of our numerical method is consistent with the $C^1(\Omega)$ continuity of the penalized velocity through the fluid–solid interface. Higher order convergence of spectral approximation can be recovered by applying post-processing techniques [22,35].

We conclude that the volume penalization method is able to model rotating obstacles, and the numerical solution converges to the exact one. Smoothing of the mask function, though optional, helps to avoid numerical oscillations of the result.

4.5. Flapping wings

The penalized momentum Eq. (4) can be naturally generalized for the case of multiple moving obstacles $\Omega_{sj}, j = 1 \ldots n_{obst}$. In this case it is convenient to rewrite the penalization term as

$$\sum_{j=1}^{n_{obst}} \frac{\chi_{\Omega_j}}{\eta}(\mathbf{u}_\eta - \mathbf{u}_{sj}), \tag{51}$$

where χ_{Ω_j} equals one inside of the jth obstacle, and zero elsewhere. The velocities $\mathbf{u}_{sj}$ are imposed by the solid body motion of each of the obstacles.

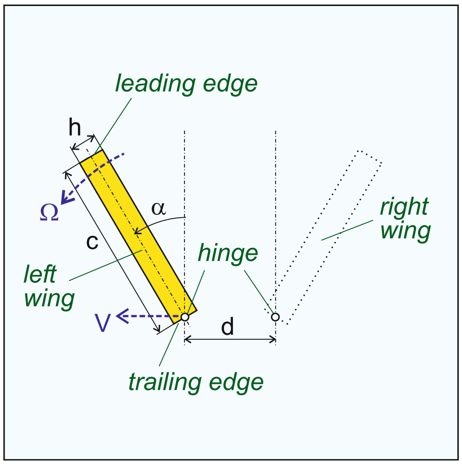

Fig. 14. Sketch of the flapping wings.

Numerical simulation of flows induced by multiple bodies moving with respect to each other is of particular interest to applications in biological fluid mechanics. Flying insects, for instance, make use of mutual aerodynamic interactions and unsteady motion of their wings to get better flight performance. An interesting and, in a certain sense, extreme case of these wing–wing interactions is the Lighthill–Weis-Fogh mechanism [38], which involves a change of topology of the flow. Our study of this flow will be reported elsewhere, but it would be appropriate to provide in this paper a kind of validation of the present numerical method for a multi-body problem.

We make a comparison with numerical simulations performed by Miller and Peskin [37] using an immersed boundary method. The wings are rectangular, with chord $c = 1$ and thickness $h = 1/32$ (see Fig. 14). The kinematics are given as functions of dimensionless time $\tau = tV_{max}/c$, where the representative velocity $V_{max} = 1$ is the maximum translational velocity. The dimensionless time varies from 0 to $\tau_{max} = 6$. Initially the wings are parallel ('clapped'), and we consider two cases, where we vary the initial distance d_0 between the hinges: $d_0 = c/6$ matches [37], and $d_0 = c/32$ is an interesting extreme case when the wings are fully clapped. Between $\tau = 0$ and $\tau = 1.74$ the wings rotate in opposite directions with angular speed

$$\Omega(\tau) = \frac{1}{2}\Omega_{rot}\left[1 - \cos\left(2\pi\frac{\tau - \tau_{turn}}{\Delta\tau_{rot}}\right)\right], \tag{52}$$

where $\Omega_{rot} = 2\Delta\theta V_{max}/\Delta\tau_{rot}c, \Delta\theta = \pi/4, \Delta\tau_{rot} = 1.74$ and $\tau_{turn} = 0$. The translational motion begins at $\tau_{accel} = 0.86$. At $0.86 < \tau < 2.16$ the wings accelerate. The wings start sweeping apart with increasing speed

$$V(\tau) = \frac{1}{2}V_{max}\left[1 + \cos\left(\pi + \pi\frac{\tau - \tau_{accel}}{\Delta\tau_{accel}}\right)\right], \tag{53}$$

where $\Delta\tau_{accel} = 1.3$. After $\tau = 2.16$ the speed remains constant and equal to V_{max}. The kinematic viscosity, $\nu = 0.0078125$, yields the Reynolds number $Re = 128$.

The periodic domain size in our simulations is $L_x \times L_y = 10 \times 10$, it is discretized with $N_x \times N_y = 2048 \times 2048$ grid points. The permeability parameter is $\eta = 5 \cdot 10^{-4}$. Smoothing of the mask function is applied (see Appendix A).

The flow field corresponding to the case $d_0 = c/32$ is visualized in Fig. 15 at three time instants. During 'fling', at $\tau = 0.6$, the fluid fills the opening space between the wings. The velocity is important near the leading edges (or tips), and reaches its maximum on the symmetry axis. Two strong counter-rotating vortices are generated at the tips. Towards the hinge point the velocity is decreasing. In the beginning of 'sweep', at $\tau = 1.2$, the velocity of the flow past the just-separated trailing edges is still small. At the same time, the leading edge vortices separate, and they detach as the wings move further apart at $\tau = 2$. One can also see newly-formed leading edge vortices and trailing edge vortices. It is worthwhile to mention that the numerical solution is stable during the change of topology.

The lift coefficient per wing, calculated as $c_L = 2F_L/\rho V^2 c$ with ρ=1, is shown in Fig. 16. It has one peak during 'fling', another one during the initiation of 'sweep', then it gradually decreases to a value which corresponds to an isolated plate. The $d_0 = c/32$ case displays 20% higher lift peaks. Note the significant increase in lift during the initial portion of sweep, which is due to stronger vorticity generation in the preceding fling motion. However, the lift drops more rapidly as the wings move further apart. Near $\tau = 0$ the lift coefficient vanishes in both simulations, as can be expected for finite-thickness wings. Time evolution of the lift coefficient reported in [37] and obtained in our simulation with $d_0 = c/6$ is in reasonable agreement, the discrepancy of 10% can be attributed to the differences in the numerical models.

5. Freely falling bodies

In this section we present a numerical model describing the motion of freely falling solid bodies.

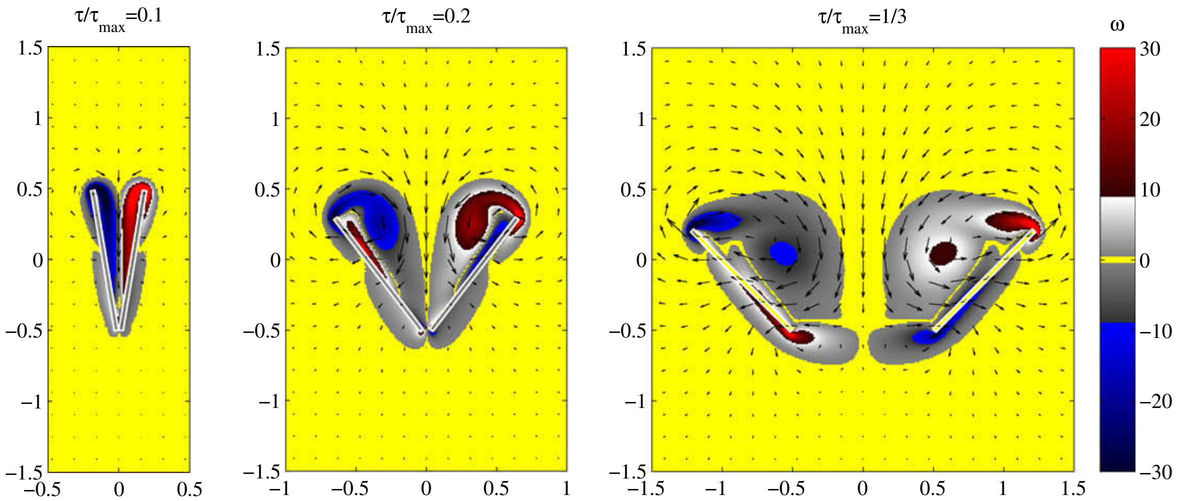

Fig. 15. Time evolution of vorticity with superimposed corresponding velocity vectors. Only zooms of the computational domain are shown.

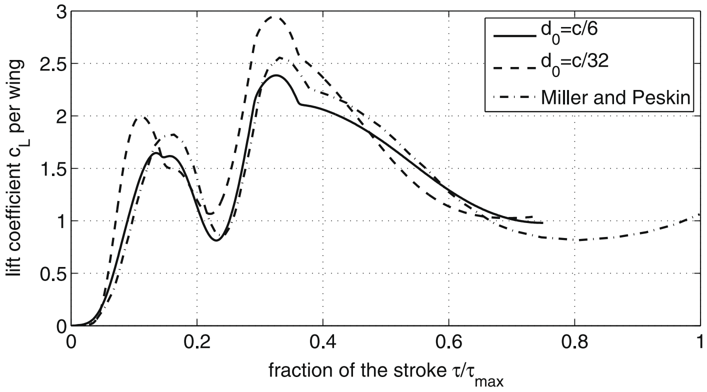

Fig. 16. Comparison of the lift coefficient obtained in the present simulations (solid and dashed lines) with the corresponding curve in [37] (dash-dot).

5.1. Fluid–solid interaction model

The general problem of an arbitrary solid body falling through a viscous incompressible fluid requires to supplement the equations of fluid mechanics by an appropriate description of the solid body dynamics. The latter is governed by Newton's 2nd law, yielding in 2D

$$m\frac{\mathrm{d}^2\mathbf{x}_{cg}}{\mathrm{d}t^2} = \mathbf{F} + \mathbf{G}, \quad J_p\frac{\mathrm{d}^2\theta_{cg}}{\mathrm{d}t^2} = M_{xcg}, \tag{54}$$

where m is the mass of the body, J_p is its moment of inertia, $\mathbf{x}_{cg}$ is the position vector of its center of gravity (c.g.), θ_{cg} is the angle of rotation of the body around c.g., $\mathbf{F}$ and M_{xcg} are, respectively, fluid force and torque, and $\mathbf{G}$ is the gravity force. Note that the solid is assumed to be homogeneous, such that both gravity and hydrostatic forces are applied at the c.g., and we can expel the gravitational source term from (1) by introducing the buoyancy-corrected gravity in (54).

Eqs. (6) and (7) provide the fluid forces and moment necessary to integrate the system of ODEs (54) governing the solid body motion. Appropriate initial conditions close the problem.

To discretize the Eq. (54) we rewrite them as a system of six first order ODEs

$$\begin{cases} \mathrm{d}X/\mathrm{d}t = V, \\ \mathrm{d}V/\mathrm{d}t = F, \end{cases} \tag{55}$$

where $X = (x_{cg}, y_{cg}, \theta_{cg})^t$, $F = (F_{xcg}, F_{ycg} + G_y, M)^t$, G_y is the buoyancy-corrected gravity force and V contains the corresponding velocities.

In the present computations we use a first order scheme:

$$\begin{cases} X^{n+1} = X^n + V^n\Delta t + \frac{1}{2}F^n\Delta t^2, \\ V^{n+1} = V^n + F^n\Delta t, \end{cases} \tag{56}$$

We observed that this scheme results in a slightly better accuracy compared with the first order explicit Euler discretization.

The first order fluid–solid coupling seems to be adequate at this stage. Implementation of higher order schemes is seen as a possible future improvement of the present code.

5.2. Sedimentation of a circular cylinder

To validate the fluid–solid interaction model we now consider the sedimentation of a solid cylinder body immersed in a fluid, starting from rest and falling down perpendicularly to its longitudinal axis. For this simulation we have chosen the following dimensional parameters: fluid viscosity $\nu = 0.03926$, fluid density $\rho = 1$, solid density $\rho_s = 2$, acceleration of gravity $g = 9.81$ and the cylinder diameter $D = 1$. The periodic domain is oriented vertically, having dimensions $L_x \times L_y = 10 \times 40$. The spatial resolution is $N_x \times N_y = 512 \times 2048$ and the penalization parameter is $\eta = 10^{-3}$.

Assuming that the cylinder achieves its steady-state, the terminal velocity u_t can be estimated from the balance between drag and buoyancy-corrected weight,

$$c_d \frac{\rho u_t^2}{2} D + (\rho - \rho_s) g \frac{\pi D^2}{4} = 0, \tag{57}$$

where c_d is the drag coefficient which can be determined as a function of $Re = u_t D/\nu$ for a cylinder in a steady inflow.

For the above choice of parameters we can judge that the cylinder will fall with a speed equal to $u_t = 3.34$ which corresponds to $Re = 85$. This is confirmed by the simulation results in Fig. 17, where the vertical velocity of the cylinder is shown as a function of time. Indeed, after being released, the cylinder accelerates and achieves the indicated terminal speed. At the same time its vortical wake develops, becomes unstable and forms a von Kármán vortex street, which makes the forces oscillate. The latter results in slight deviations of the trajectory from a straight line, and the velocity also slightly oscillates around u_t.

5.3. Falling leaves

In this subsection we present two numerical simulations of a falling leaf, showing different patterns of its behaviour. Following [13–15], we consider a plate with an elliptical cross-section. The problem admits three dimensionless parameters:

- the eccentricity of the ellipse $e = b/a$,
- the Reynolds number $Re = 2u_t a/\nu$,
- the dimensionless moment of inertia $I^* = \frac{b(a^2+b^2)\rho_s}{2a^3\rho}$.

In the above, a and b are, respectively, the major and minor semi-axes of the ellipse, ρ_s is its density, ρ and ν are the density and the kinematic viscosity of the fluid, respectively, and $u_t = \sqrt{\pi b g(\rho_s/\rho - 1)}$ is the terminal velocity estimated by balancing gravity, buoyancy and drag of a plate having the chord $2a$ and unit drag coefficient. In the first simulation we set $e = 0.125, Re = 1100$ and $I^* = 0.17$. The leaf is released from rest with an initial angle of 0.2 rad.

The following numerical parameters are used for both simulations in this subsection: the periodic domain size is $L_x \times L_y = 10 \times 20$ chord lengths, $L = 2a$, the resolution is $N_x \times N_y = 1024 \times 2048$ and the penalization parameter is $\eta = 10^{-3}$.

From Fig. 18 one can see that after being released the leaf starts oscillating from side to side (fluttering) with increasing amplitude, attaining increasingly higher angles at the turning points of its trajectory. After a while, this increase results in an upside-down turn, and then the leaf continues rotating in the same direction and drifting sideways (tumbling). This behav-

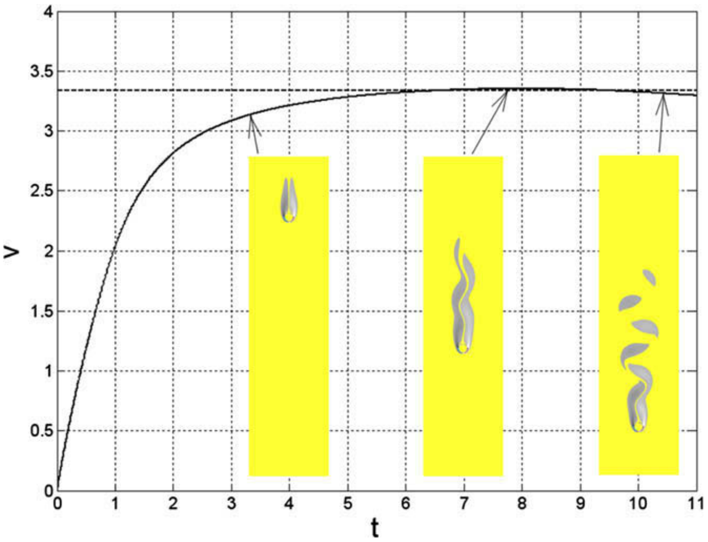

Fig. 17. Velocity of a falling cylinder starting from rest (solid line). Insets show its wake (vorticity field) at three distinct times. The horizontal dashed line indicates the steady-state estimate.

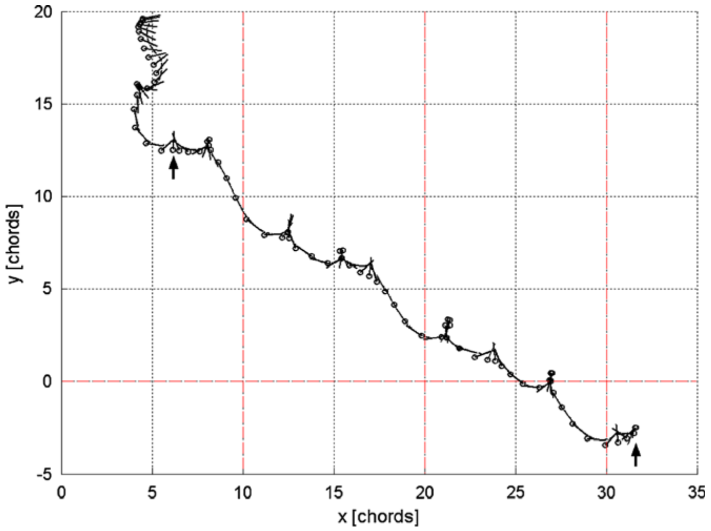

Fig. 18. Trajectory of a falling leaf. Arrows mark the tumbling zone corresponding to the average velocities in Table 1. Dashed red lines indicate the periodic box size.

Table 1
Averaged translational and angular velocities, and descent angle of the falling ellipse.

	U (cm/s)	V (cm/s)	Ω_z (1/s)	θ (deg)
Andersen et al. [14]	15.6	−7.4	18.0	25.3
Jin and Xu [29]	15.3	−11.2	16.9	36.2
Present computation	11.3	−7.2	19.4	32.5

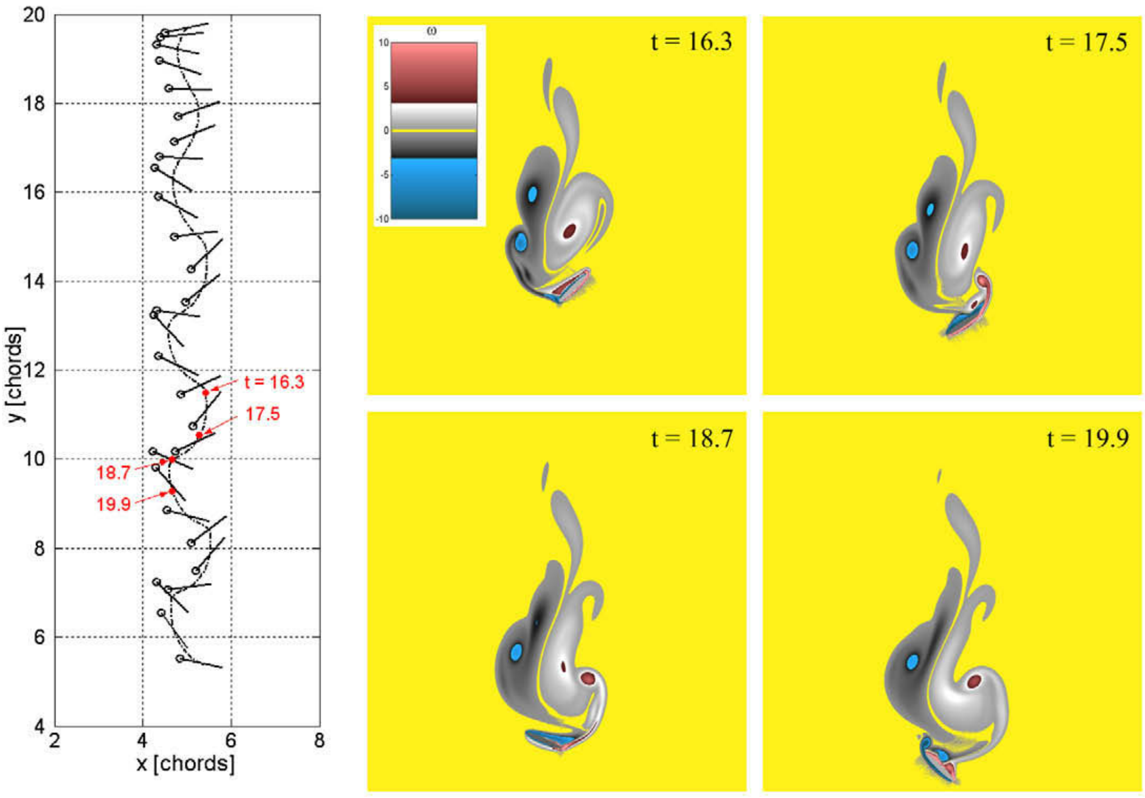

Fig. 19. Trajectory (left) and vorticity plots (right) of a fluttering leaf.

iour is qualitatively similar to that obtained in [13,14], but in the latter the transition occurred just after one cycle of fluttering. Such an important feature as the center of gravity elevation near the cusp-like turning points, first reported in [13] and attributed to the viscous effects, is clearly present in our simulation.

To justify the fact that the accordance between the trajectory in Fig. 18 and the one presented in [13] is only qualitative, but not point-to-point, we emphasize that the problem is very sensitive to perturbations. In this view, it is natural that two very different numerical methods predict quantitatively different trajectories. For instance, Jin and Xu [29] explored the same falling leaf using lattice Boltzmann simulations and did not observe the period-two structure reported in [14]. In addition, at longer times the present result is most probably affected by the boundary conditions. Although the tumbling regime is stable enough to persist these perturbations, no periodic final state establishes.

Nonetheless, we can examine some integral quantities, such as average velocities of tumbling motion. Table 1 shows a comparison of the horizontal U, vertical V and angular Ω_z mean velocities with the values presented in [14,29]. In the present computation they are obtained by averaging through an interval marked on Fig. 18 and converted to the dimensional form. The table also shows the descent angle, calculated as $\theta = \arctan(|V/U|)$. The agreement is reasonable, especially in view of the relatively low resolution (about 51 points per chord) and the coarse penalization parameter ($\eta = 10^{-3}$). Simulations with finer discretization and larger domains are anticipated.

An important feature of the 2D free fall is that it exhibits a number of distinct patterns, depending on the parameters of the problem (see, *e.g.* [14,15]). In particular, the experiments [12] show that, at $Re = 3 \cdot 10^3 \ldots 4 \cdot 10^4$, a rectangular plate either flutters or tumbles according to the similarity number $\sqrt{m/\rho L^2}$, where m is its mass per unit length and L is its chord.

We have seen in the above simulation an example when the fluttering motion diverges, and the leaf starts tumbling. In the second simulation we show that lighter leaves tend to flutter rather than tumble. We decrease the density ratio twice, leaving all the other dimensional properties unchanged, and thus setting $I^* = 0.085, Re = 494$. This yields $\sqrt{m/\rho L^2} = 0.3626$. One can see in Fig. 19 that this leaf indeed flutters. Periodic motion establishes after two or three cycles of transient. In contrast, the behaviour of the vortical wake is rather complex: the detached boundary layer rolls up into intensive vortices, which then pair and merge (*cf*. the results of the inviscid model [30]).

6. Conclusions and perspectives

A numerical method has been developed for simulations of solid bodies moving through a viscous incompressible fluid. It represents a further development of the technique presented in [4]. The 2D Navier–Stokes equations, written in the vorticity-streamfunction formulation, are discretized using the Fourier pseudo-spectral scheme. Therewith, solid obstacles of arbitrary shape can be taken into account using the volume penalization method.

Four distinct time discretization schemes have been implemented and assessed, which mainly differ in the integration of the penalization term. The results indicate that the schemes based on implicit integration and exponential propagation of the penalization term need further development in order to be used for moving obstacle simulations without loosing efficiency. The explicit scheme has been kept in this work for its precision for a given computational cost.

The originality of the present work lies in the implementation of time-dependent volume penalization, which makes this method capable of solving problems where the obstacle follows an arbitrary motion. Interpolation of a mask function in Fourier space helps to avoid oscillations of hydrodynamic forces when the obstacle translates with respect to the grid. This approach has been generalized for the arbitrary 2D motion by factorization of rotation into a sequence of 1D operations.

Couette flow between cylinders has been considered. The volume penalization method is found able to model rotating obstacles, and the numerical solution converges to the exact one. Smoothing of the mask function reduces numerical oscillations, while the order of convergence remains first to second. Noteworthily, there exists a possibility to recover the higher order of convergence by applying a post-processing technique [22,35]. We anticipate its implementation in our future work.

The penalization model has been extended to solve the problem of solid bodies falling through a fluid. An ODE system of the solid body dynamics has been introduced and discretized with first order accuracy. Higher order schemes are to be implemented in future.

The sedimentation of a cylinder has been considered. The terminal velocity achieved in the simulation agrees reasonably well with the value predicted from the balance between drag and buoyancy-corrected weight. Different simulations of falling leaves (plates) have been performed. For the effective choice of numerical parameters, the results are in reasonable agreement with the available data. Fluttering and tumbling regimes of the freely falling leaves have been explored, depending on the physical parameters. A detailed study is anticipated in future.

Generalization of the model to three spatial dimensions is envisaged. In that case we prefer the velocity–pressure formulation, since the vorticity equation is no longer scalar-valued. In addition, this formulation would be more convenient for an implicit integration of the penalization term. Incompressibility can be imposed using, *e.g.* fractional step method. Concerning the moving obstacles, generalization of the algorithm described in Section 4 is straightforward, the sole difference is that three translations and three rotations are required.

Acknowledgments

The authors acknowledge financial support from the Agence Nationale de la Recherche grant ‘M2TFP’, the French-German University Project S-GRK-ED-04-05, and the Institut du Développement et des Ressources en Informatique Scientifique for providing computer time on its NEC SX-8 vector computer under Project 0211664. We thank Alexandre Azzalini, Marie

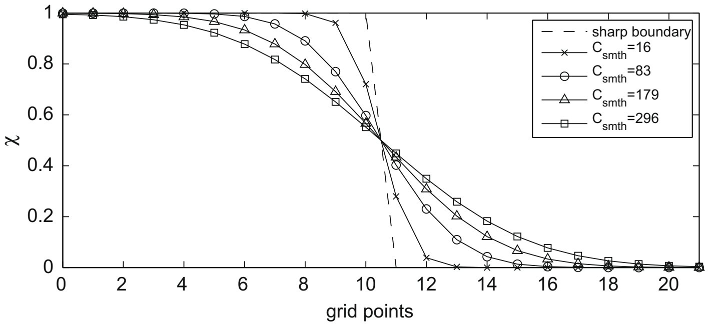

Fig. A.1. Normal profiles of a mask function corresponding to different amounts of smoothing.

Farge, Wouter Bos and Romain Nguyen van Yen for many discussions and useful suggestions, and Dominique Fougère for computational assistance. We thank Oleg Vasilyev for pointing out the unsteady correction of hydrodynamic forces and for other important remarks, Philippe Angot for clarifying some key points of the volume penalization method, and the Referee for many constructive comments improving the quality of the paper.

Appendix A. Smoothing of the mask function

As it was mentioned in Section 4.2, smoothing of the mask function is required to avoid spurious oscillations of the hydrodynamic forces on moving obstacles. Another reason is the need for de-aliasing of the penalization term, since the product $\chi_\Omega(\mathbf{u}_\eta - \mathbf{u}_s)$ can contain modes which are not resolved by the grid.

A convenient way to smooth the mask function is to convolve it with a Gaussian filter [39],

$$\widehat{\chi} = \exp(-C_{smth}(k_x^2/N_x^2 + k_y^2/N_y^2))\widehat{\chi}_{sharp}, \tag{A.1}$$

which can be seen as diffusing the initially discontinuous mask χ_{sharp} by applying several iterations of solving the heat equation. This procedure is straightforward with a spectral method, and the high-gradient region remains well localized in physical space and becomes localized in Fourier space. Our typical choice is $C_{smth} = 16$.

Alternatively, the profiles in Fig. A.1 can be used to “smear out” the mask function, *i.e.*, to impose a smooth transition from 0 to 1, which contains a given number of points in a normal direction to the boundary. However, this simpler approach is limited to simple geometries.

Appendix B. Stability of explicit integration of the penalization term

As we mentioned in Section 3.1, the second order Adams–Bashforth time integration of the penalization term in the Navier–Stokes equations implies a stability condition $\Delta t < \eta$. This condition is derived below from the linear stability analysis.

Let us consider Eq. (8) with $\nabla \times \mathbf{f} = 0$ and $\mathbf{u}_s = 0$, *i.e.*,

$$\partial_t \omega_\eta + \mathbf{u}_\eta \cdot \nabla \omega_\eta - \nu \nabla^2 \omega_\eta + \nabla \times \left(\frac{\chi_\Omega}{\eta} \mathbf{u}_\eta \right) = 0, \tag{B.1}$$

where $\mathbf{u}_\eta = \nabla^\perp \Psi + \mathbf{U}_\infty$ and Ψ satisfies $\nabla^2 \Psi = \omega_\eta$. Inside the obstacle the penalized velocity $\mathbf{u}_\eta$ is tending to zero with η (see [34]), and the non-linear term of (B.1) is of higher order in η compared to the linear terms. Let us restrict our attention to a particular case where the obstacle occupies the whole computational domain. The result obtained under this assumption seems to hold in general. The vorticity equation (B.1) is then much simplified:

$$\partial_t \omega_\eta - \nu \nabla^2 \omega_\eta + \frac{1}{\eta} \nabla \times \mathbf{u}_\eta = 0. \tag{B.2}$$

All spatial derivatives in (B.2) are evaluated with a Fourier collocation method, therefore a numerical approximation of $\nabla \times \mathbf{u}_\eta$ equals exactly ω_η. Then rewriting (B.2) in terms of its discrete Fourier modes yields

$$\partial_t \widehat{\omega}_\eta(\mathbf{k}) + \nu |\mathbf{k}|^2 \widehat{\omega}_\eta(\mathbf{k}) + \frac{1}{\eta} \widehat{\omega}_\eta(\mathbf{k}) = 0. \tag{B.3}$$

Now let us apply the AB2 scheme with exact treatment of the diffusion term to solve this equation. This means that the explicit time-stepping scheme is applied to the equation

$$\partial_t \widehat{W}_\eta(\mathbf{k}) = -\frac{1}{\eta} \widehat{W}_\eta(\mathbf{k}), \tag{B.4}$$

where $\widehat{W}_\eta(\mathbf{k}) = e^{\nu|\mathbf{k}|^2 t} \widehat{\omega}_\eta(\mathbf{k})$. The linear operator on the right-hand side of (B.4) is simply a constant equal to $-1/\eta$, and the stability condition of the AB2 scheme (see [6]) in this case is $0 \geqslant -\Delta t/\eta \geqslant -1$, which yields $\Delta t \leqslant \eta$. By writing $\Delta t < \eta$ we exclude the neutrally stable case.

References

[1] E. Arquis, J.-P. Caltagirone, Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide – milieux poreux: application à la convection naturelle, Comptes Rendus de l'Academie des Sciences Paris II 299 (1984) 1–4.

[2] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in viscous flows, Numerische Mathematik 81 (1999) 497–520.

[3] A. Azzalini, Étude des écoulements barotropes en rotation en présence ou non d'obstacles: simulations pseudo-spectrales et filtrages en ondelettes, Thèse de doctorat, Université Paris VI Pierre et Marie Curie, 2004.

[4] K. Schneider, Numerical simulation of the transient flow behaviour in chemical reactors using a penalization method, Computers and Fluids 34 (2005) 1223–1238.

[5] M. Farge, K. Schneider, Analyzing and computing turbulent flows using wavelets. Cours des Houches LXXIV, new trends in turbulence, in: M. Lesieur, A. Yaglom, F. David (Eds.), Springer, 2002, pp. 453–503.

[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, 1988.

[7] J.D. Cole, On a quasi-linear parabolic equation occurring in aerodynamics, Quarterly of Applied Mathematics 9 (1951) 225–236.

[8] K. Schneider, M. Farge, Numerical simulation of the transient flow behaviour in tube bundles using a volume penalization method, Journal of Fluids and Structures 20 (2005) 555–566.

[9] N.K.-R. Kevlahan, J.-M. Ghidaglia, Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization, European Journal of Mechanics B/Fluids 20 (2001) 333–350.

[10] K. Schneider, M. Paget-Goy, G. Pellegrino, A. Verga, M. Farge, Direct numerical simulation of impulsively started and uniformly accelerated plates using adaptive wavelet and Fourier methods with penalization, Proceedings of Turbulence and Shear Flow Phenomena III 1 (2003) 407–412.

[11] M. Unser, P. Thévenaz, L. Yaroslavsky, Convolution-based interpolation for fast, high-quality rotation of images, IEEE Transactions on Image Processing 10 (4) (1995) 1371–1381.

[12] A. Belmonte, H. Eisenberg, E. Moses, From flutter to tumble: inertial drag and Froude similarity in falling paper, Physical Review Letters 81 (1998) 345.

[13] U. Pesavento, Z.J. Wang, Falling paper: Navier–Stokes solutions model of fluid forces and center of mass elevation, Physical Review Letters 93 (2004) 144501-1.

[14] A. Andersen, U. Pesavento, Z.J. Wang, Unsteady aerodynamics of fluttering and tumbling plates, Journal of Fluid Mechanics 541 (2005) 65.

[15] A. Andersen, U. Pesavento, Z.J. Wang, Analysis of transitions between fluttering, tumbling and steady descent of falling cards, Journal of Fluid Mechanics 541 (2005) 91.

[16] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239.

[17] C.S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[18] C.S. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[19] Z.J. Wang, D. Russell, Effect of forewing and hindwing interactions on aerodynamic forces and power in hovering dragonfly flight, Physical Review Letters 99 (2007) 148101.

[20] A. Sarthou, S. Vincent, J.-P. Caltagirone, P. Angot, Eulerian–Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects, International Journal for Numerical Methods in Fluids 56 (8) (2008) 1093–1099.

[21] I. Ramière, P. Angot, M. Belliard, A general fictitious domain method with immersed jumps and multilevel nested structured meshes, Journal of Computational Physics 225 (2) (2007) 1347–1387.

[22] G.H. Keetels, U. D'Ortona, W. Kramer, H.J.H. Clercx, K. Schneider, G.J.F. van Heijst, Fourier spectral and wavelet solvers for the incompressible Navier–Stokes equations with volume penalization: convergence of a dipole-wall collision, Journal of Computational Physics 227 (2) (2007) 919–945.

[23] W.S. Edwards, L.S. Tuckerman, R.A. Friesner, D.C. Sorensen, Krylov methods for the incompressible Navier–Stokes equations, Journal of Computational Physics 110 (1) (1994) 82–102.

[24] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Review 45 (1) (2003) 3–49.

[25] K. Schneider, M. Farge, Decaying two-dimensional turbulence in a circular container, Physical Review Letters 95 (2005) 244502.

[26] W.E. Arnoldi, The principle of minimized iteration in the solution of the matrix eigenvalue problem, Quarterly of Applied Mathematics 9 (1951) 17–25.

[27] A.S. Monin, A.M. Yaglom, Statistical Fluid Mechanics: Mechanics of Turbulence, vol. 1, MIT Press, Cambridge, MA, 1971.

[28] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, Journal of Computational Physics 209 (2) (2005) 448–476.

[29] C. Jin, K. Xu, Numerical study of the unsteady aerodynamics of the freely falling plates, Communications in Computational Physics 3 (4) (2008) 834–851.

[30] M.A. Jones, M.J. Shelley, Falling cards, Journal of Fluid Mechanics 540 (2005) 393–425.

[31] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, Journal of Computational Physics 227 (10) (2008) 4825–4852.

[32] N. Sharma, N.A. Patankar, A fast computation technique for the direct numerical simulation of rigid particulate flows, Journal of Computational Physics 205 (2) (2005) 439–457.

[33] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, Journal of Computational Physics 169 (2) (2001) 363–426.

[34] G. Carbou, P. Fabrie, Boundary layer for a penalization method for viscous incompressible flow, Advances in Differential Equations 8 (2003) 1453–1480.

[35] E. Tadmor, J. Tanner, Adaptive mollifiers for high resolution recovery of piecewise smooth data from its spectral information, Foundations of Computational Mathematics 2 (2002) 155–189.

[36] P. Angot, J.-P. Caltagirone, New graphical and computational architecture concept for numerical simulation on supercomputers, in: Proceedings of 2nd World Congress on Computational Mechanics, vol. 1, 1990, pp. 973–976

[37] L.A. Miller, C.S. Peskin, A computational fluid dynamics of 'clap and fling' in the smallest insects, Journal of Experimental Biology 208 (2005) 195–212.

[38] M.J. Lighthill, On the Weis–Fogh mechanism of lift generation, Journal of Fluid Mechanics 60 (1) (1973) 1–17.

[39] R. Nguyen van Yen, Private communication.