HAL
open science

A Hybrid Grid-Particle Method for Moving Bodies in 3D Stokes Flow with Variable Viscosity

Robin Chatelin, Philippe Poncet

To cite this version:

Robin Chatelin, Philippe Poncet. A Hybrid Grid-Particle Method for Moving Bodies in 3D Stokes Flow with Variable Viscosity. SIAM Journal on Scientific Computing, 2013, 35 (4), pp.B925-B949. 10.1137/120892921 . hal-02010645

HAL Id: hal-02010645

https://hal.science/hal-02010645v1

Submitted on 7 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SIAM J. Sci. Comput.
Vol. 35, No. 4, pp. B925–B949
© 2013 Society for Industrial and Applied Mathematics

A HYBRID GRID-PARTICLE METHOD FOR MOVING BODIES IN 3D STOKES FLOW WITH VARIABLE VISCOSITY*

ROBIN CHATELIN† AND PHILIPPE PONCET‡

Abstract. This article presents a new approach for the resolution of large three-dimensional Stokes equations with variable viscosity fluids, coupled with transport equations. After building the model, we will write these equations in the context of highly viscous flows and penalization, in order to consider complex geometries moving in a fluid. From a mathematical point of view, the solutions show nonlinear dynamics. Beside the use of standard tools such as finite differences and staggered grids, we have built a new methodology based on large three-dimensional simulations, including operators splitting for an efficient use of fast solvers, multi-index fixed point methods, Lagrangian methods with fast and accurate grid-particle transfers, and the multiresolution description of variables. Among the main original aspects of this method, both accurate incompressibility and variable viscosity are treated in the same fixed point. Hence the computation costs for variable and constant viscosity flows are similar. Several examples are given to validate the order of convergence and conservation rates. Such models are used, among other examples, in biological computing at the cellular scale. The present article eventually describes the ciliated epithelium cells covering a mammal's lungs, beating in a mucus film. This study of human lung diseases explores the efficiency of the mucociliary clearance, a challenging problem in health sciences, especially for the investigation of cystic fibrosis and various chronic obstructive pulmonary diseases.

Key words. particle methods, three-dimensional flows, Stokes equations, complex geometry, variable viscosity flows, biomathematics, mucus propulsion, mucociliary clearance

AMS subject classifications. 65M06, 65M12, 65M25, 68U20, 76D07, 76D17, 76Z05, 92B05

DOI. 10.1137/120892921

1. Introduction.

The numerical resolution of three-dimensional (3D) Stokes and Navier–Stokes equations has undergone significant advances in the last decades, and is still a challenge for the understanding of fluid dynamics in both the industrial and the academic worlds. Beyond the classical problems arising in fluid mechanics of homogeneous media, there is now a need for efficient algorithms applied to more complex flows, such as biological flows and medical flows, microfluidics, or porous media. This study presents a new Lagrangian/Eulerian method for complex moving geometries, applied to biological mucus flows in human lungs, at the scale of ciliated epithelium cells.

To compute the coupling of Stokes or Navier–Stokes equations with other partial differential equations, most numerical techniques consider a unique numerical method for the whole set of equations. This article focuses on 3D Stokes problems in moving geometry coupled with transport equations, which are of a very different nature. In the spirit of what has been developed in particle methods for Navier–Stokes equations since the early nineties [15, 32], we present a hybrid numerical method, mixing a novel

*Submitted to the journal's Computational Methods in Science and Engineering section September 26, 2012; accepted for publication (in revised form) June 17, 2013; published electronically August 22, 2013. This work was supported by the ANR Grant BioFiReaDy, under the contract ANR-2010-JCJC-0113-01.

http://www.siam.org/journals/sisc/35-4/89292.html

†Toulouse Institute of Mathematics, UMR CNRS 5219, Team MIP, F-31077 Toulouse, France and Université de Toulouse, F-31077 Toulouse, France (robin.chatelin@gmail.com).

‡Corresponding author. Laboratoire de Mathématiques et de leurs Applications, UMR CNRS 5142, IPRA, Université de Pau et des Pays de l'Adour, F-64013 Pau, France (philippe.poncet@univ-pau.fr).

efficient Eulerian method for the elliptic Stokes equations and a particle method for the transport equation [17, 43].

The interaction between fluid and complex moving geometries is performed by means of a penalization method [4]. This is obtained by giving a strong weight to the terms dealing with objects immersed in the fluid, through their characteristic function. There is consequently no need to get a precise mesh of obstacles and of their surfaces (hence it is not necessary to remesh periodically if they are moving in the computational domain). This is a good alternative to arbitrary Lagrangian–Eulerian (ALE) formulations, Lagrange-multiplier methods, or immersed boundary methods, especially combined with Cartesian grids and fast solvers (see [4, 14] for discussion and comparison with other numerical methods).

The Lagrangian property of particles makes the transport terms vanish from the equations so that the robustness of these methods is not subject to any transport stability conditions [3, 6, 18]. This helps to perform numerical simulations with large time steps without losing accuracy [42, 46], which is especially useful to study flows beyond transient regimes [48]. In conventional vortex methods, the quantity transported and the velocity field are linked through Poisson equations, thus naturally satisfying divergence-free conditions [15, 46]. Here we consider the problems of variable viscosity fluids, and taking the curl of the momentum equation will exhibit stiff second order derivatives of viscosity, so the velocity-pressure formulation is the correct framework. It can also be coupled with Lagrangian transport methods.

Special attention is given to producing an algorithm giving divergence-free velocity fields while being sufficiently efficient to handle large 3D simulations without ruining the efficiency of particle methods. The projection on divergence-free fields usually induces tangential spurious velocities [26]. Algorithms based on MAC schemes [44, 41], integral methods using local [13, 47] or global incidence [45] are useful in the context of parabolic partial differential equations, but are not valid for elliptic problems. No-slip boundary conditions often require implicit formulations which leads to large and full linear systems. In order to avoid implicit formulations and nonetheless satisfy boundary conditions, we set up a novel fixed point algorithm to deal with residual velocity at boundaries, divergence-free conditions, and at the same time, extra terms from the nonhomogeneity of the fluid. The numerical resolution of 3D Stokes equations is reduced to a sequence of Poisson and Helmholtz equations. These equations allow the use of fast solvers, and avoid the storage of matrices coming from large linear systems, which would be prohibitive even with sparse storage.

In modern particle methods, there has been a controversy about how to compute the solution of Poisson equations: is it preferable to use kernel methods [25] and multipole decomposition [35, 53], or hybrid grid/particle methods [15, 42, 49] with high order interpolation between grids and particles [16, 39]? While both approaches have known dramatic improvements over the last twenty years, in the present study we consider a variable diffusion coefficient μ, and there is no available Green's kernels associated with operators $(-\mathrm{div}(\mu\nabla u))$. Indeed, under some assumptions on the mechanical laws considered, μ follows a transport or diffusion-transport equation [10, 55], which prevents using kernel methods (such as Stokeslets or Biot–Savart) since any structure on μ can be supposed in a very artificial way.

In this article, a new Eulerian approach combining penalization and projection methods is introduced for Stokes problem. A fixed point algorithm is used to split the differential operators into parts allowing the use of fast solvers (multiscale iterative solver for Helmholtz equations [1] and Fourier–Chebyshev for Poisson equations [60, 59]). Another original aspect of this work is the nonhomogeneity of fluid

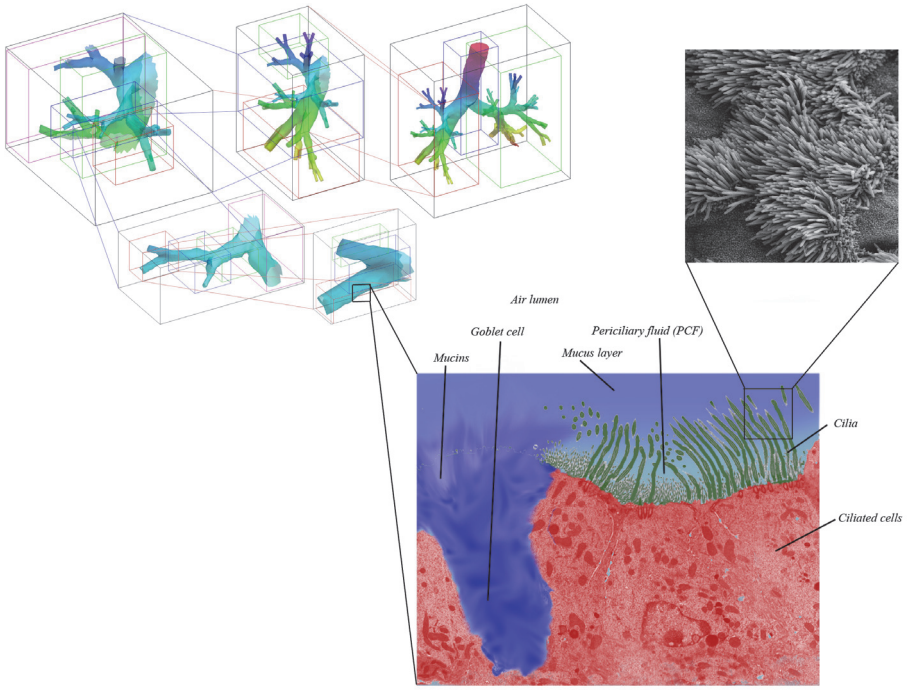

FIG. 1.1. *Sketch of human lungs at different scales, from trachea* ($\simeq 1cm$) *to ciliated cells* ($\simeq 5\mu m$). *Courtesy of C. Daghlian, Dartmouth University, for original versions of right-hand images.*

viscosity, which adds some terms to the Stokes equation. The modern features of this algorithm give similar computational costs for constant viscosity flows and for variable viscosity flows (since the extra terms are also included in the fixed point). The Stokes problem is coupled with transport equations for viscosity and solved with Lagrangian methods, as they present a lot of advantages for such equations. The coupling generates nonlinear effects, which are not contained in classical Stokes problem. The resulting method is second order with a complexity $\mathcal{O}(n)$. Moreover, Lagrangian methods allow the use of large time steps, without significant loss of accuracy (that is to say the constants in error estimation are small in practice), and without unnecessary resorting to the Stokes solver.

At a scale between 100 nm and 10 μm, equations of continuum mechanics are still valid, and the Reynolds number is small enough to be under the assumption of Stokes flows [51]. The present work will focus on biological applications at the cellular scale, but the methodology developed here is also valid for a wide range of applications such as porous media (Darcy–Stokes or Darcy–Brinkman), heterogeneous microfluidics, or plasma physics.

This numerical application investigates ciliated cells covering human (and mammal) lung walls, and moving into pulmonary mucus, a fluid of variable rheology [33, 50], as displayed in Figure 1.1. Numerical simulations are used to study different mechanisms separately and understand the phenomena occurring more specifically in cystic fibrosis (mucoviscidosis) [40], but also in various chronic obstructive pulmonary diseases. Indeed, understanding the contributions of different biological parameters helps to identify the ones which are dominant or sensitive, and consequently identify the alterations leading to pathological configurations.

In section 2, we will build a model based on physical and mechanical considerations, and discuss this set of partial differential equations in its penalized formulation for complex moving geometries.

In section 3, we will focus on fast methods to solve the penalized 3D Stokes equations, involving well-chosen fixed points and staggered interpolations in order to use fast multiscale Helmholtz and Fourier solvers.

Section 4 will deal with particle methods for viscosity transport, and its coupling with the previously introduced Eulerian Stokes solver. Both sections 3 and 4 will contain numerical validation, convergence, and scalability. All these elements will eventually be put together and validated with the simulation of a sphere moving in a variable viscosity 3D Stokes flow at high resolution.

Finally, section 5 will describe the application to biological films, with a simulation of mucus around a ciliated cell, in the context of a human lung.

2. Problem setup.

2.1. Model setup. The usual mathematical model for the dynamics of a 3D flow are the Navier–Stokes equations. For a Newtonian fluid of velocity $u : \mathbb{R}^3 \times]0, T[\longrightarrow \mathbb{R}^3$ and variable density ρ, they read as conservation equations of mass and momentum:

$$\begin{cases} \dfrac{\partial \rho u}{\partial t} + \mathrm{div}(\rho u \otimes u) - \mathrm{div}(\sigma) = f, \\ \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) = 0, \end{cases} \tag{2.1}$$

where u the velocity field is satisfying incompressibility condition $\mathrm{div} u = 0$ all over the domain, p is the pressure, $\sigma = 2\mu \mathrm{D}(u) - pI$ is the stress tensor, μ is the viscosity, and the strain tensor $\mathrm{D}(u)$ is the symmetric part of ∇u, that is to say $\mathrm{D}(u) = (\nabla u + \nabla u^T)/2$.

In the case of highly viscous flows and/or microfluidic scales, the Reynolds number is sufficiently small to neglect convection $\mathrm{div}(\rho u \times u)$ and time dependency in (2.1). The conservation of momentum reduces to the following Stokes quasi-static equation:

$$-\mathrm{div}(2\mu \mathrm{D}(u)) = f - \nabla p. \tag{2.2}$$

In (2.2) the density ρ only remains in the external force f (in the barotropic $f = \rho g z$ expression of gravity, for example). The simulations presented in the second part of this paper will focus on microfluidics applications. At these scales, gravity does not influence the flow, which is why the density will not appear in the equations but the fluid is still nonhomogeneous and its density stays variable.

In order to model the medium with variable rheology, we can introduce a massive fraction α quantifying the heterogeneity of the medium, which follows the same kind of equation as density:

$$\frac{\partial \alpha}{\partial t} + \mathrm{div}(\alpha u) = \eta \Delta \alpha, \tag{2.3}$$

where η is a diffusion coefficient of heterogeneity, corresponding to molecular diffusion. In biological applications, the massive fraction can be the proportion of a given protein (here mucins) in an aqueous medium. The viscosity of the medium can be considered as a function of the massive fraction α, that is to say $\mu = \phi(\alpha)$. Under

these assumptions, the medium considered here is still Newtonian. This model is a good compromise between non-Newtonian constitutive laws of rheology and constant viscosity fluid models.

For nonreactive Newtonian flows or nonmiscible media, η can be assumed to be zero, and consequently the viscosity μ also follows a transport equation [10, 55] like α and ρ, with adequate initial and boundary conditions. In some works, μ is directly assumed to be a function of ρ, and/or reciprocally ρ a function of μ. In this case, existence and uniqueness in adequate Sobolev spaces can be found in [8].

This assumption is made in the present study so μ follows a pure transport equation. This case is numerically more difficult than diffusion transport since there is no regularization due to the Laplacian and consequently function μ can exhibit much larger gradients.

Eventually density has disappeared from the equations but it is still variable and can be computed (if needed) as a function of μ.

2.2. Governing equations. One considers (2.2) in a time-dependent domain $\Omega(t) \subset \mathbb{R}^3$. This is the 3D Stokes quasi-static inviscid problem. It is completed by a transport equation for viscosity:

$$\begin{cases} \dfrac{\partial \mu}{\partial t} + u \cdot \nabla \mu = 0 & \text{in } \Omega(t) \times]0, T[, \\ -\operatorname{div}(2\mu \mathrm{D}(u)) + \nabla p = f \quad \text{and} \quad \operatorname{div} u = 0 & \text{in } \Omega(t) \times]0, T[, \\ u(x,t) = \overline{u}(x,t) & \text{on } \partial\Omega(t) \times]0, T[, \end{cases} \tag{2.4}$$

where $\overline{u}$ is a given velocity at boundaries, with initial conditions and boundary conditions on μ yet to be set.

To take into account the motion of a complex moving geometry immersed in this Stokes fluid, a penalization method [4] is used. The goal of penalization is to satisfy (up to an approximation) an equality of the type $u = \overline{u}$ inside a region $B(t)$ (often called a "solid region"), where $\overline{u}$ is a given divergence-free field.

Let Q denote the computational box and $B(t)$ the penalized region: the fluid domain is then $\Omega(t) = Q \setminus B(t)$. Let also function χ be the characteristic function of $B(t)$, ε the penalization parameter satisfying $\varepsilon \ll 1$, and $\overline{u}$ the velocity field inside the solid which is assumed to be known at each time step. Penalization transforms the momentum conservation equation of system (2.4) into the following equation:

$$-\operatorname{div}(2\mu \mathrm{D}(u)) + \frac{\chi}{\varepsilon}(u - \overline{u}) = f - \nabla p \quad \text{in } Q \times]0, T[, \tag{2.5}$$

where all the functions depend on time and space.

Using this method, the error produced on the computed solution has an order $\sqrt{\varepsilon}$ inside the fluid and a first order accuracy with respect to spatial discretization due to an error present in a small layer around the interface. In order to improve accuracy, interpolation with boundary condition on $\partial\Omega(t)$ can be done in the neighborhood of the jump of $\chi(\cdot, t)$, such as the one used in [7].

3. Numerical method for 3D Stokes equation in variable viscosity. In this part we consider the momentum conservation equation at a given time t. This means $\mu(\cdot, t)$ is perfectly defined in the whole domain. A novel algorithm able to handle accurate incompressibility, boundary conditions, penalization, and nonhomogeneous viscosity is developed. Problem (2.5) is split in order to have the best possible performances and to perform large 3D simulations with a guaranteed order of convergence and conservation.

The first strategy is to use a projector on divergence-free fields: it splits (2.5) into two coupled velocity-pressure equations. The main advantage of this strategy is to avoid the assembly of a rigidity matrix for the global Stokes problem (2.5), which usually leads to prohibitive computational cost for large 3D simulations, even when using modern linear solvers (such as [5] for example). Indeed, when using such splitting, we obtain four Laplace equations solved using fast solvers.

Projection on divergence-free fields generates a spurious tangential velocity on the computational box boundary ∂Q. Thereafter, we will discuss how to satisfy the correct boundary conditions.

3.1. Divergence-free projection algorithm and boundary conditions. One can develop

$$\mathrm{div}\,(2\mu \mathrm{D}(u)) = \mu\Delta u + \left(\nabla u + \nabla u^T\right)\nabla\mu + \mu\nabla\mathrm{div}u \tag{3.1}$$

with $\nabla\mathrm{div}u \equiv 0$ since u is divergence free. The penalized Stokes problem is then written:

$$\begin{cases} -\mu\Delta u - 2\mathrm{D}(u)\nabla\mu + \nabla p + \dfrac{\chi}{\varepsilon}(u-\bar{u}) = f \quad \text{and} \quad \mathrm{div}u = 0 & \text{in } Q, \\ u(\cdot,t) = g(\cdot,t) & \text{on } \partial Q, \end{cases} \tag{3.2}$$

where f and g are given functions.

To solve this velocity-pressure problem a projection method is used. It is an alternative to Lagrange multiplier methods. In these methods, the solution to the Stokes problem is seen as the solution to a saddle-point problem. It is very natural to implement using finite element discretization as it is derived from the variational formulation of the Stokes problem (for details, see [2, 52]). In [26] many variants of Chorin's original algorithm [13] are presented for unsteady Navier–Stokes equations and a novel variant for quasi-static flows is now presented.

A projection on divergence-free fields is defined as follows: let $v : Q \to \mathbb{R}^3$ be a vector field, and $\zeta : Q \to \mathbb{R}$ the solution (unique up to a constant) of the following equation:

$$\begin{cases} -\Delta\zeta = -\mathrm{div}v & \text{in } Q, \\ \dfrac{\partial\zeta}{\partial n} = v\cdot n & \text{on } \partial Q. \end{cases} \tag{3.3}$$

$\mathbb{P}$ denotes the projection operator on the divergence-free fields of $\mathbb{R}^3$, that is to say, the function of v given by $\mathbb{P}(v) = v - \nabla\zeta$.

The first step of the splitting algorithm consists in solving the previous equation without a pressure term (and with an additional expression to guarantee consistency with variable viscosity as explained below). u^* is the solution of the following equation:

$$\begin{cases} -\mu\Delta u^* + \dfrac{\chi}{\varepsilon}(\mathbb{P}(u^*) - \bar{u}) = f + [2\mathrm{D}(\mathbb{P}(u^*)) + (\mathrm{div}u^*)\mathrm{Id}]\,\nabla\mu & \text{in } Q, \\ u^*(x,t) = \vartheta(x,t) & \text{on } \partial Q. \end{cases} \tag{3.4}$$

The velocity field u^* does not satisfy the relation $\mathrm{div}u^* = 0$, and is projected on divergence-free fields by means of projection $\mathbb{P}$: the solution of problem (3.2) is then given by

$$u = \mathbb{P}(u^*) = u^* - \nabla\zeta \tag{3.5}$$

and immediately satisfies $\mathrm{div}u = \mathrm{div}\mathbb{P}(u^*) = \mathrm{div}u^* - \Delta\zeta \equiv 0$.

Indeed, using $u = u^* - \nabla\zeta$ leads to

$$\mathrm{div}(2\mu \mathrm{D}(u)) = \mu\Delta(u^* - \nabla\zeta) + 2\mathrm{D}(u)\nabla\mu = \mu\Delta u^* - \mu\nabla\Delta\zeta + 2\mathrm{D}(u)\nabla\mu. \tag{3.6}$$

Noticing that $\mu\nabla\Delta\zeta = \nabla(\mu\Delta\zeta) - \Delta\zeta\nabla\mu$, setting the pressure to $p = -\mu\Delta\zeta$, and using that $\Delta\zeta = \mathrm{div}u^*$, one gets

$$\mathrm{div}(2\mu \mathrm{D}(u)) - \nabla p = \mu\Delta u^* + \mathrm{div}u^*\nabla\mu + 2D(u)\nabla\mu \tag{3.7}$$

and consequently

$$-\mathrm{div}(2\mu \mathrm{D}(u)) + \nabla p = -\mu\Delta u^* - [2\mathrm{D}(\mathbb{P}(u^*)) + (\mathrm{div}u^*)\mathrm{Id}]\,\nabla\mu. \tag{3.8}$$

This gives the equivalence between (3.2) and (3.4), except for boundary conditions. This is discussed below.

Here, it is necessary for boundary condition $tr(u) = \vartheta - tr(\nabla\zeta)$ to be equal to g (here $tr(u)$ denotes the restriction of u on the boundary). The choice of ϑ to satisfy this boundary condition is not obvious, even if $\vartheta = g$ is often chosen and $tr(u) - g = -tr(\nabla\zeta)$ is then considered as a lack of accuracy of the scheme.

This lack of accuracy in tangential velocity in projection algorithms has been reported in [26] and is more important in quasi-static problems than for unsteady flows, for which projection is incremental in time so information from the previous time step is used.

Furthermore, for time incremental problems (with increment δt), panel methods can be used to satisfy boundary conditions, using the explicit solutions of integral equations on the boundary neighborhood, scaling as $\sqrt{2\mu\delta t/\rho}$, and are consequently very fast. By panel methods, we mean any linear method correcting the boundary conditions by means of a flux on the boundary neighborhood, most of the time with the potential solution of an integro-differential equation (see [47, 45, 15] for instance). In the present situation, panel methods are not efficient since the Green's kernel support, for the Laplace equation, fills the whole domain. In that case the incidence of correction on boundary conditions is not confined to the boundary neighborhood and the computational efficiency is dramatically reduced.

Consequently, we prefer the implicit formulation of boundary conditions on ∂Q, that is to say finding the correct ϑ of (3.4) such as $tr(\mathbb{P}(u^*)) = g$. In this case, the application $\vartheta \longmapsto tr(u^*)$ is affine, which leads $tr(\mathbb{P}(u^*)) = g$ to be a linear equation of variable ϑ, since u^* is a function of ϑ through (3.4).

We have explored three ways to solve this problem: assembling its matrix in order to use a direct solver, identifying the linear part of the affine function and using an iterative method such as the GMRES algorithm, and finally using a fixed point algorithm directly on the affine function. The second and third strategies are orders of magnitude faster than the first one.

Moreover, we prefer using a fixed point algorithm, which can be mixed with another fixed point treating the splitting of the sum $\mathrm{div}(\mu D(u)) = \mu\Delta u + D(u)\nabla u$. Indeed, using a GMRES algorithm embedded with a fixed point makes GMRES dramatically lose its efficiency, and using full GMRES would require a large Krylov space dimension, which means a very large amount of memory.

3.2. Fixed point algorithm. The fixed point iteration comes from rewriting the term $u - \overline{u} = \mathbb{P}(u^*) - \overline{u}$ as a sum between $u^* - \overline{u}$ of the present iteration on the one hand, and $\nabla\zeta$ of the previous iteration on the other hand. The algorithm is then defined as follows:

1. $u_0^* = u_0$ is provided for initialization;
2. fixed point loop:
   (a) u_k^* is assumed to be known, as well as $u_k = \mathbb{P}(u_k^*)$ and $\nabla\zeta_k = u_k^* - \mathbb{P}(u_k^*)$,
   (b) u_{k+1}^* is the solution of

$$\begin{cases} -\mu\Delta u_{k+1}^* + \dfrac{\chi}{\varepsilon}(u_{k+1}^* - \nabla\zeta_k - \bar{u}) = f - [2\mathrm{D}(u_k) + (\mathrm{div} u_k^*)\mathrm{Id}]\,\nabla\mu & \text{in } Q, \\ u_{k+1}^* = g - \nabla\zeta_k & \text{on } \partial Q; \end{cases}$$

   (c) compute projector ζ_{k+1} and project $u_{k+1} = \mathbb{P}(u_{k+1}^*)$;
   (d) if an exit test (such as $\|u_{k+1} - u_k\| \leqslant \varepsilon$) is positive, then go to 3., otherwise increment k and restart 2;
3. the final numerical solution of (3.2) is given by u_{k+1}.

In step 2(b), it is absolutely necessary to keep the term $\chi\varepsilon^{-1}u_{k+1}^*$ implicit. This introduces the difficulty of the Helmholtz equation with coefficient jumps, while the explicit version using $\chi\varepsilon^{-1}(u_k^* - \nabla\zeta_k) = \chi\varepsilon^{-1}\mathbb{P}(u_k^*)$ would lead to a simple Poisson equation. Nevertheless, there is no longer any convergence of such an explicit algorithm as soon as ε is under a certain (insufficient) threshold (10^{-4} in the example shown in section 3.3).

Furthermore, the boundary condition for u^* can be relaxed into the expression

$$u_{k+1}^* = g - (1-\theta)\nabla\zeta_k - \theta\nabla\zeta_{k-1}. \tag{3.9}$$

Here a Cartesian discretization is considered: the computational box is a parallelepiped $Q = [x_m, x_M] \times [y_m, y_M] \times [z_m, z_M]$ discretized using n_x, n_y, and n_z points, respectively, in the x, y, and z directions.

The main advantage in this formulation is that the equation in step 2(b) is a Helmholtz equation, and projection equation (3.3) is a Poisson equation. For both these equations, fast solvers can be used, thus avoiding matrix assembly since the resulting linear systems are standard problems with explicitly known coefficients. This leads to fast resolution with low storage memory, which suits for large three-dimensional problems very well.

In practice, the MUDPACK solver is used for the Helmholtz equation [1]. It relies on multiscale formulation and allows the resolution of general elliptic partial differential equations with nonhomogeneous coefficients (this is very interesting for both the penalization term and the viscosity gradients). Moreover, the FISHPACK solver is used for the Poisson equation [59, 60]. It is based on fast Fourier transforms and Chebyshev polynomial decompositions. These solvers are sufficiently fast to offset the number of fixed point iterations by orders of magnitude.

The projection step does not have sufficient accuracy if defined on collocated grids, because it generates a red/black synchronization on discretized knots, as the discretization of operators Δ and $\mathrm{div}\nabla$ does not coincide on such grids. This usual problem for velocity-pressure problems is solved by using staggered grids: the three components of velocity do not reside on the same grids. Those grids have a half-step space shift in one direction.

In our present study, instead of considering the centered and the three staggered grids physically, we only consider centered grids and the values on the staggered grids are computed by means of accurate interpolations, as suggested in [7]. We use the third order kernel M_4' for these interpolations between centered and staggered grids, which is defined by (4.5) in section 4.1.

Two reasons justify this choice. First in the staggered direction, the physical boundary does not match with the grid boundary so an extrapolation of boundary condition have to be performed and results in the transformation of a Dirichlet boundary condition into a Robin boundary condition. Thus the treatment of the Robin boundary condition in the fixed point is nontrivial and some terms derived from the Hessian of ζ have to be considered to guarantee the required accuracy in tangential velocity. The second reason is also a consequence of the transformation of the Dirichlet boundary condition into a Robin boundary condition. The Robin boundary condition has to be imposed *inside* the domain to keep the right flux sign, and consequently to keep the associated variational formulation coercive. It means that if N denotes the number of grid points in the nonstaggered direction, there are $N-1$ grid points in the staggered one. This is a serious drawback in using the MUDPACK solver (or any multiscale solver) which requires a number of grid points of the form $N = 2^p + 1$ in each direction to get optimal multigrid performances.

Furthermore, the algorithm's efficiency and accuracy are measured by monitoring the following four quantities:

- the divergence of the flow $\|\mathrm{div} u^{(k)}\|_Q$;
- the residual tangential velocity $\|u^{(k)} - g(\cdot, t)\|_{\partial Q}$;
- the residual velocity inside solid $\|u^{(k)} - \bar{u}\|_{B(t)}$;
- the residual strain $\|\mathrm{D}(u^{(k)}) - \mathrm{D}(u^{(k-1)})\|_Q$.

Convergence is analyzed in section 3.3, by means of a Green–Taylor vortex with variable viscosity. It will be shown that the number of fixed point iterations, thus the computation cost, can be reduced by using adequate preconditioning.

3.3. Scalability, convergence, and preconditioning. The convergence of the Stokes solver is validated by means of a 3D generalization of the Green–Taylor vortex. This is a divergence-free 1-periodic velocity field. It has homogeneous Dirichlet boundary conditions on $Q = [0,1]^3$ and nonhomogeneous on $Q = [-\frac{1}{2}, \frac{1}{2}]^3$.

It enables us to test various kinds of boundary conditions. A pressure field is arbitrarily defined. Then the right-hand side of (3.2) is forced with the exact value of $-\mathrm{div}(2\mu D(u)) + \nabla p$ and $\bar{u}$ is set to the exact solution. This allows us to test the Stokes solver for a variable viscosity, as well as penalization and the projection iterative process. The expression of this theoretical solution is given by

$$
\begin{aligned}
u(x,y,z) &= \begin{cases} 2(\cos(2\pi x) - 1)\sin(2\pi y)\sin(2\pi z), \\ -(\cos(2\pi y) - 1)\sin(2\pi x)\sin(2\pi z), \\ -(\cos(2\pi z) - 1)\sin(2\pi x)\sin(2\pi y), \end{cases} \\
p(x,y,z) &= \sin^2(2\pi x)\sin^2(2\pi y)\sin^2(2\pi z), \\
\mu(x,y,z) &= 2 + \sin(2\pi x)\sin(2\pi y)\sin(2\pi z).
\end{aligned} \tag{3.10}
$$

For a resolution of 256^3, one can notice on Figure 3.1 that all fixed point residuals reach a threshold of 5×10^{-6}. Furthermore, the relative error with respect to this solution is computed for various resolutions (using from $(2^4)^3$ to $(2^8)^3$ degrees of freedom) and is reported on Figure 3.2. The points are sufficiently aligned and the slope is computed using mean square first order polynomial interpolation. As expected, the slopes are very close to two, which is the expected value (that is to say compatible with discretization orders). It validates the accuracy of the solver.

The previous iterative method can be preconditioned if an accurate estimation of $\nabla\zeta^{(0)}$ and $D(u^{(0)})$ is provided. One way to obtain such an estimation is to first solve the problem on a coarser grid and then to interpolate the resulting projector

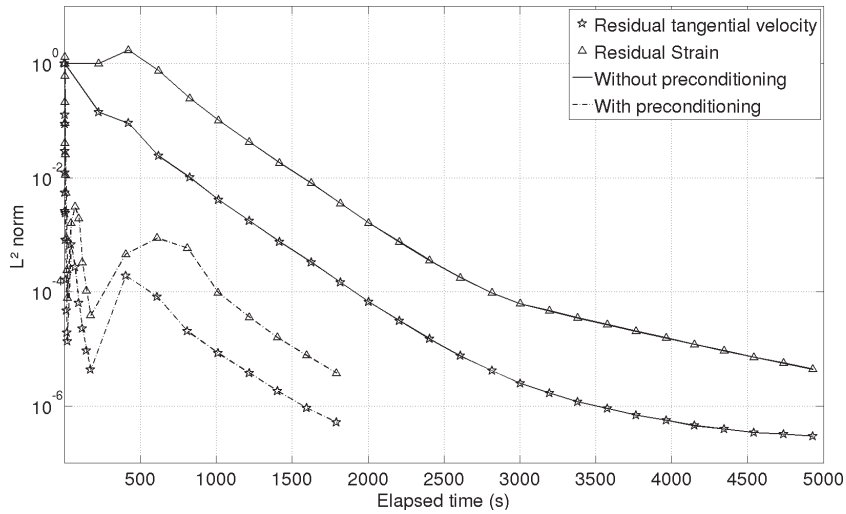

FIG. 3.1. *Comparison of residuals between the preconditioned algorithm and the original one using the same flow configuration as for Figure* 3.2. *Only two residuals are presented to improve readability. Simulation resolution is* 256^3.

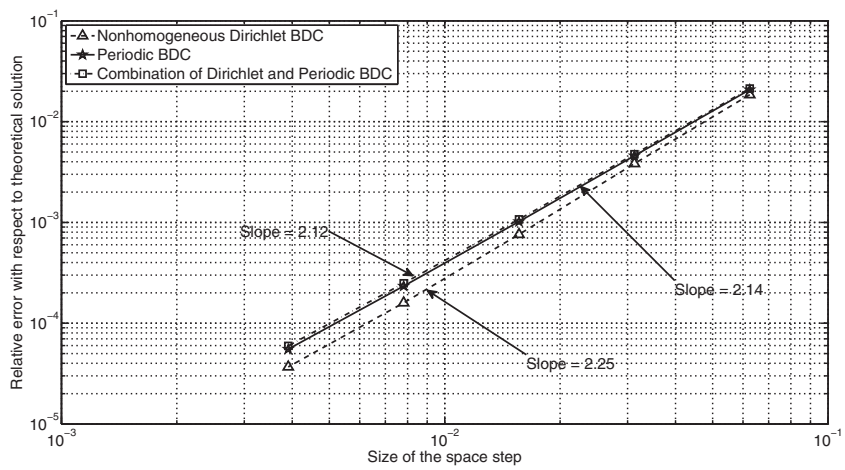

FIG. 3.2. *Convergence plot of Stokes solver for a nonhomogeneous flow. Different configurations of boundary conditions are presented. The analytical solution is the Green–Taylor vortex and the computation domain is* $Q = [-\frac{1}{2}, \frac{1}{2}]^3$. *Penalization is tested forcing an exact solution in a sphere of radius* 0.1 *centered at space origin. Second order accuracy is obtained for all configurations.*

and strain. Interpolation kernels have to be accurate enough (see the next section for details on interpolation kernels).

3D computations are performed using Cartesian grids with the number of intervals in one direction set from $N = 2^p + 1$. Solving the equation on a grid of $2^{p-1} + 1$ intervals in each direction allows us to estimate $\nabla\zeta$ accurately. The process can be performed down to a very coarse grid such as 32^3, for example. For a coarse resolution the preconditioning does not make a significant difference in computational time (see Figure 3.3). However it leads to a substantial gain of iterations at a fine level and

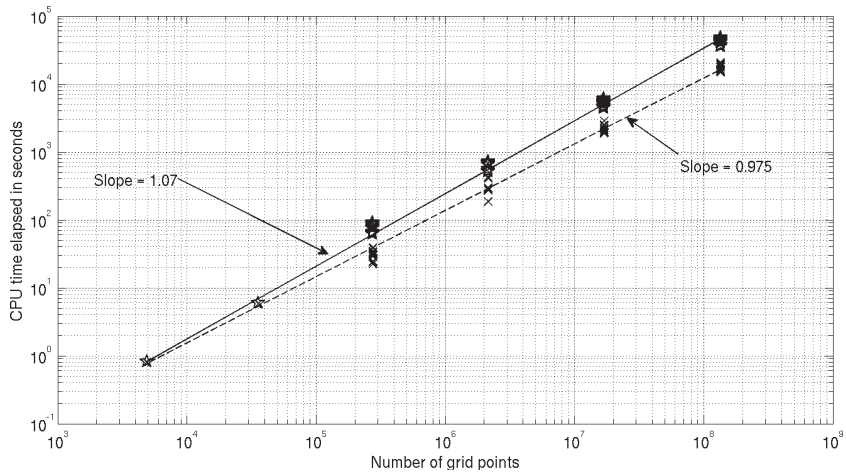

FIG. 3.3. *Scalability of quasi-static Stokes algorithm based on fixed point. These results are presented for the Green–Taylor vortex with variable viscosity, homogeneous Dirichlet boundary conditions on z faces, and periodic boundary conditions in x and y directions. Results compare performance using a coarser grid preconditioning (dotted line) or not using it (solid line). Computation is performed on a single CPU Intel Xeon X*5450 *(*3.00 *GHz). Computation time is given in seconds.* 15 *simulations have been performed for each resolution to get an average.*

consequently decreases the computational cost of the Stokes solver, up to a factor 3 for a 256^3 simulation (also shown on Figure 3.1) and a factor 4 for a 512^3 simulation.

4. Coupling particle-in-cell methods with 3D Stokes equation. This section focuses on the resolution of the convection equation of viscosity,

$$\partial_t \mu + u \cdot \nabla \mu = 0 \quad \text{in } Q. \tag{4.1}$$

A particle-in-cell method is used to transform this partial differential equation into a set of ordinary differential equations.

4.1. Conventional remeshed particle-in-cell methods. The particle-in-cell method consists in a Lagrangian discretization of the computational box, as a set of $k = 1 \dots K$ particles of volume v_k, located at position ξ_k, and holding a viscosity μ_k.

A measure solution of (4.1) is defined by

$$\mu(t) = \sum_{k=1}^{K} \mu_k(t) \delta_{\xi_k(t)} v_k(t), \tag{4.2}$$

where δ_ξ is the Dirac function $\delta_\xi(\phi) = \phi(\xi)$, so that (4.1) is rewritten using this Lagrangian discretization:

$$\begin{cases} \dfrac{d\xi_k}{dt} = u(\xi_k(t), t), \\ \dfrac{d\mu_k}{dt} = 0 \quad \text{and} \quad \dfrac{dv_k}{dt} = 0. \end{cases} \tag{4.3}$$

The volume equation's right-hand side is zero because the velocity field u is a divergence-free solution of the Stokes equations (the Jacobian of the flow in the volume integral remains constant, with an initial condition being the identity matrix; see Annex A.2 of [15] for details).

No additional boundary conditions are required for this dynamics system; it is implicitly contained in the definition of the velocity field u and interpolation kernels.

With this Lagrangian formulation the convection term $u \cdot \nabla\mu$ vanishes as well as its related stability condition (CFL condition). Consequently, one of the main advantages of such a Lagrangian method is the ability to use large time steps with no significant loss of accuracy.

Coherence between the Lagrangian feature of the transport equation and the Eulerian feature of Stokes equations has to be set up; a hybrid grid-particle algorithm is used (introduced in [15, 17] for a similar transport equation). Details of a second order midpoint scheme for temporal integration is summarized as follows:

1. viscosity μ is given on a grid, and grid points ξ_k define particles of viscosity μ_k;
2. velocity u is computed on the grid by the Stokes solver;
3. particles are pushed along the velocity field, by means of an explicit Euler scheme for (4.3) on half a time step. Values of μ_k are unchanged but particle positions ξ_k are changed by $\xi_k^* = \xi_k + \delta t u(\xi_k)/2$. We consequently have the intermediate time-step measure solution $\mu^* = \sum_{k=1}^K \mu_k \delta_{\xi_k^*} v_k$;
4. particles are then interpolated back to grid points by means of the convolution formula $\Lambda^\epsilon * \mu(x) = \sum_{k=1}^K \Lambda^\epsilon(x - \xi_k)\mu_k v_k$;
5. velocity u^* is computed using this intermediate time-step viscosity on the grid;
6. initial particles at position ξ_k are pushed along velocity field u^* using an explicit Euler scheme over a full time step by $\xi_k + \delta t u^*(\xi_k)$.

This midpoint formula is second order in time. A key point is of course the choice of interpolation kernel. As considered in [17, 42], we use Monaghan's M_4' kernel [39]. Indeed, we set the following rescaling, tensorialized on the three variables of space:

$$\Lambda^\epsilon(x) = \frac{1}{\epsilon^3} M_4'^{\otimes 3}(x/\epsilon), \tag{4.4}$$

where M_4' is defined as:

$$M_4'(x) = \begin{cases} (3x^3 - 5x^2 + 2)/2 & \text{if } 0 \leqslant |x| \leqslant 1, \\ (2-x)^2(1-x)/2 & \text{if } 1 \leqslant |x| \leqslant 2, \\ 0 & \text{if } |x| \geqslant 2. \end{cases} \tag{4.5}$$

It keeps the three first moments of distribution, it is symmetric, continuously differentiable two times (except at zero), and has a short compact support. Close to the wall, the one-sided kernel given in [17] based on M_4' can be used. Rescaling parameter ϵ is set at the same value as the grid step size, so that there are 4 grid points in each direction interacting with a particle. Consequently, the scalability of interpolation between grid and particles is linear. Moreover the conservation of the discrete integrals is ensured by the spline-like property of this kernel:

$$\sum_{k \in \mathbb{Z}} \Lambda(x + j) = 1. \tag{4.6}$$

The convergence and consistence order of hybrid grid-particle methods have been discussed in [15, 17, 42] in the context of Navier–Stokes equations. Here we have reformulated it to Stokes-transport coupling, where velocity is a solution of the 3D Stokes equation.

Convergence, consistence, and conservation for the overall problem of coupling 3D Stokes and transport of viscosity are discussed in the next section.

4.2. Convergence in the variable viscosity flow. In this section the coupling between the Lagrangian transport for viscosity and the Stokes solver is tested: a second flow configuration, the Poiseuille profile, is presented. The flow takes place between two planes (located at $z = z_m$ and $z = z_M$) where the velocity is zero and a time oscillating external force is generating a parabolic velocity profile between these planes. On other faces of the computation domain, boundary conditions are periodic. In this simplified test, viscosity is only transported as a passive tracer. Velocity is the solution of the Poisson problem $-\Delta u(x, y, z, t) = (2\theta(t), 0, 0)^T$, so analytical solutions of both velocity and viscosity are available for all time t.

Viscosity at initial time $t = 0$ has the same expression as in the previous configuration and the characteristic method enables a theoretical solution of viscosity for all times $t > 0$. The expression of the velocity field, pressure gradient and viscosity at time t for this oscillating Poiseuille flow is the following:

$$
\begin{aligned}
Q &= [x_m, x_M] \times [y_m, y_M] \times [z_m, z_M], \\
u(x, y, z, t) &= \begin{cases} \theta(t)(z - z_m)(z - z_M), \\ 0, \\ 0, \end{cases} \\
\nabla p(x, y, z) &= 0, \\
\mu(x, y, z, t = 0) &= \mu_0(x, y, z) = 2 + \sin(2\pi x)\sin(2\pi y)\sin(2\pi z), \\
\mu(x, y, z, t) &= \mu_0(x_0(t), y_0(t), z_0(t)), \\
x_0(x, y, z, t) &= x - \Theta(t)(z - z_m)(z - z_M) \quad \text{with} \quad \Theta'(t) = \theta(t), \\
y_0(x, y, z, t) &= y, \qquad z_0(x, y, z, t) = z.
\end{aligned}
\tag{4.7}
$$

The transport equation of viscosity is discretized using the measure formulation, and the midpoint formula is used for the resulting ordinary differential equations. The velocity field is computed with the Stokes solver of section 3.

Results are presented for different space step sizes on Figure 4.1. If the grid is too coarse, the error from spatial discretization (committed performing interpolations

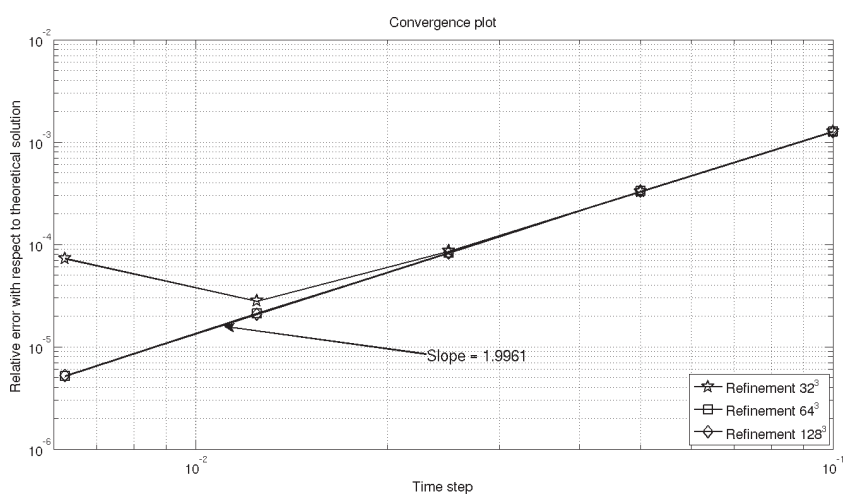

FIG. 4.1. *Convergence of the Lagrangian method using the Stokes solver for computing the velocity field. The error is plotted with respect to time step size. Results are presented for different values of spatial resolution. Second order accuracy is reached using fine grids so that the time discretization error dominates the spatial discretization error.*

and computations of velocity) dominates. Using a fine enough spatial discretization shows second order accuracy.

In addition, in order to get accurate interpolations and to handle viscosity gradients carefully, one can discretize viscosity on a particle cloud with a resolution two times finer than that of velocity [16]. This selective refinement is used and discussed in conservation benchmarks described in the next section.

One can notice that the error never reaches an unphysical high value, featuring numerical instabilities. Indeed, the Lagrangian formulation of the transport equation has no stability condition, which leads to a robust method, that is to say, with unconditional stability.

4.3. Conservation in the transport equation and scalability. This test case is a sphere spinning in the computational box, starting in a stratified viscosity field. The goal is to let it spin for several cycles, as displayed in Figure 4.2, to check whether viscosity is conserved or not. Indeed, in the resolution of the transport equation, diffusion phenomena can appear and viscosity can be numerically depleted.

The sphere is moving in the computation box $Q =]-1/2, 1/2[^3$, following the trajectory $\bar{X}(t)$ with the velocity $\bar{U}(t) = \bar{X}'(t)$ defined by

$$\bar{X}(t) = \left(\frac{1}{4}\cos 2\pi t, 0, \frac{1}{4}\sin 2\pi t\right) \quad \text{and} \quad \bar{U}(t) = \left(-\frac{\pi}{2}\sin 2\pi t, 0, \frac{\pi}{2}\cos 2\pi t\right). \tag{4.8}$$

Viscosity is initially stratified as

$$\mu(x, y, z, t = 0) = \mu_0(x, y, z) = 1 + z \tag{4.9}$$

and is then transported using the Stokes flow in its full variable viscosity formulation (2.4) with no external forces on the right-hand side.

Homogeneous Dirichlet boundary conditions are set on velocity for the left, right, top, and bottom boundaries (that is to say $x = \pm 0.5$ and $z = \pm 0.5$), while periodic boundary conditions are set for front and back planes ($y = \pm 0.5$). Boundary conditions on viscosity are also periodic in the y direction, and no boundary condition is required in other directions since $u = 0$ on these boundaries (so the domain entrance boundary is the empty set).

One can introduce the mean value of viscosity

$$\Gamma(t) = \frac{1}{\text{meas}(Q)} \int_Q \mu(x, y, z, t) dx dy dz. \tag{4.10}$$

No exact solution to the flow defined above can be analytically found, but the viscosity has to satisfy the conservation law $\Gamma(t) = \Gamma(0)$.

The ratio $\big(\Gamma(t) - \Gamma(0)\big)/\Gamma(0)$ is consequently computed at each time step and is presented on Figure 4.3. Computations are performed with $\delta t = 0.01$ s and 15 cycles are performed (one cycle is the unit of time). Results are presented for two different refinements: 64 and 128 points in each direction for velocity, and viscosity is double resolution (128 and 256).

One can notice on Figure 4.3 that viscosity is conserved throughout the simulation with an error decreasing when the refinement is finer: the smallest accumulation is 0.005% which is very small for such configurations exhibiting large gradients for μ and u as time increases. A usual finite difference discretization of this equation would have led to a CFL number of 1.92 (which is not stable by a factor of 4), and smaller time steps would have been required.

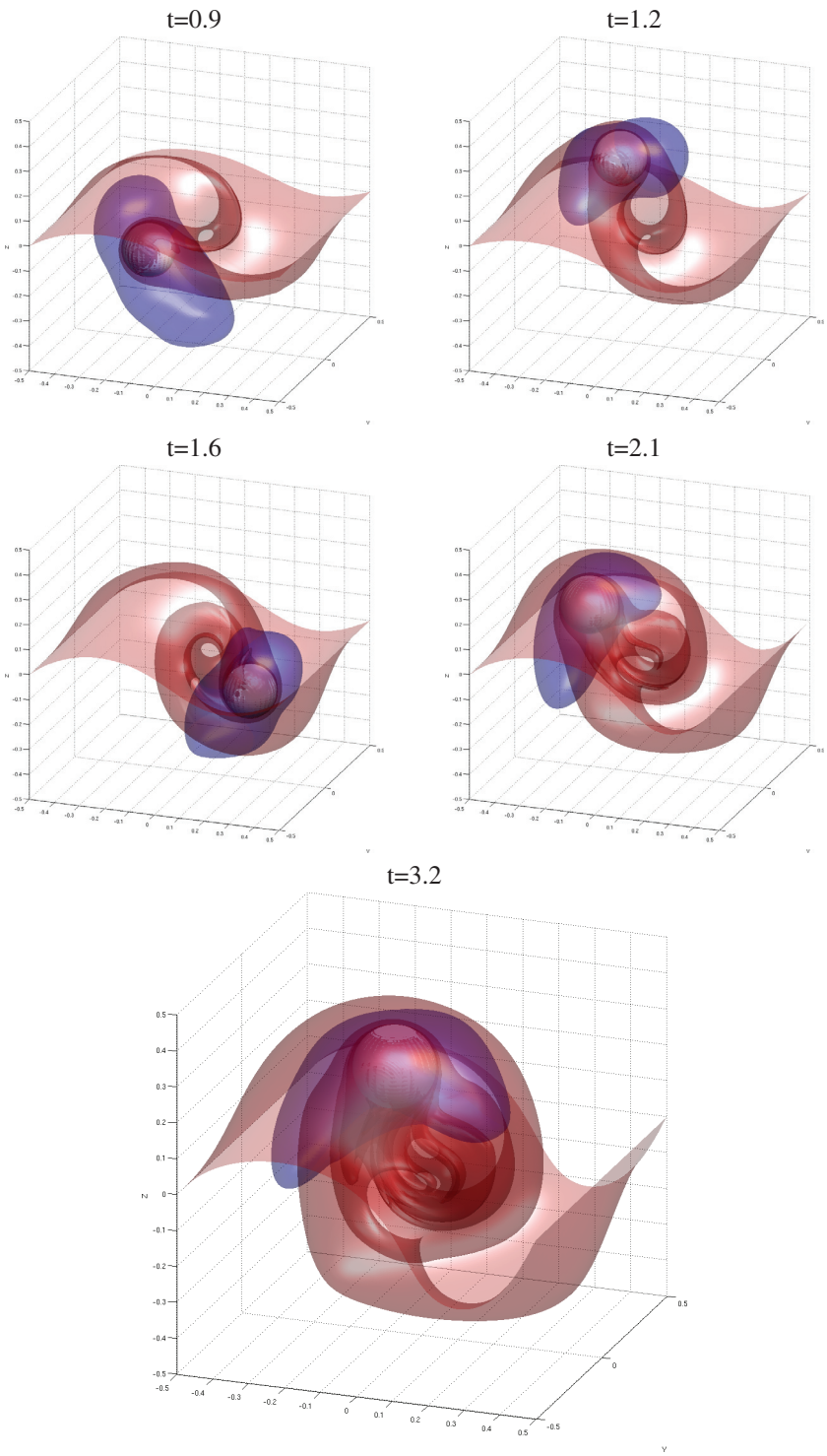

FIG. 4.2. *Sphere spinning in a variable viscosity Stokes flow, at different times. The coupling between the Stokes and transport equations generates strong nonlinear effects. The pictures show isosurfaces of viscosity at level* $1.5\ Pa \cdot s$ *(in red) and the velocity norm (in blue) at level* $0.5\ m \cdot s^{-1}$ *for a* 256^3 *simulation.*

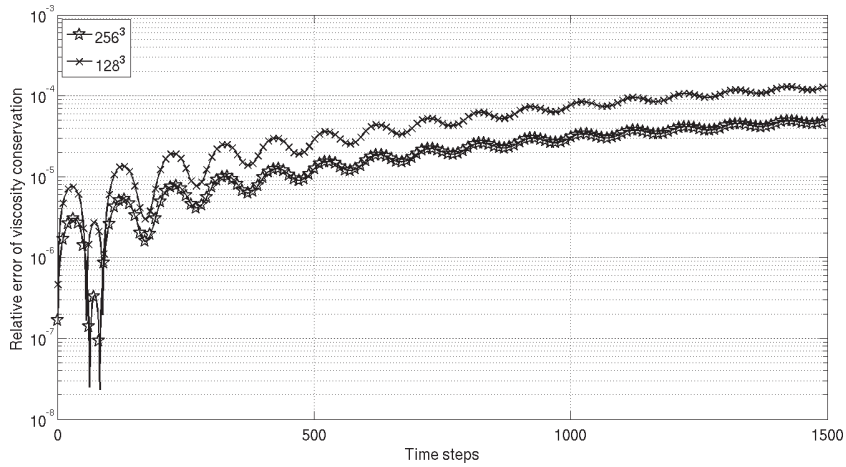

FIG. 4.3. *Evolution of the ratio* $(\Gamma(t) - \Gamma(0))/\Gamma(0)$ *during the simulation, using* $\delta t = 0.01$ *s time steps. The sphere is performing* 15 *turns according to* (4.8).

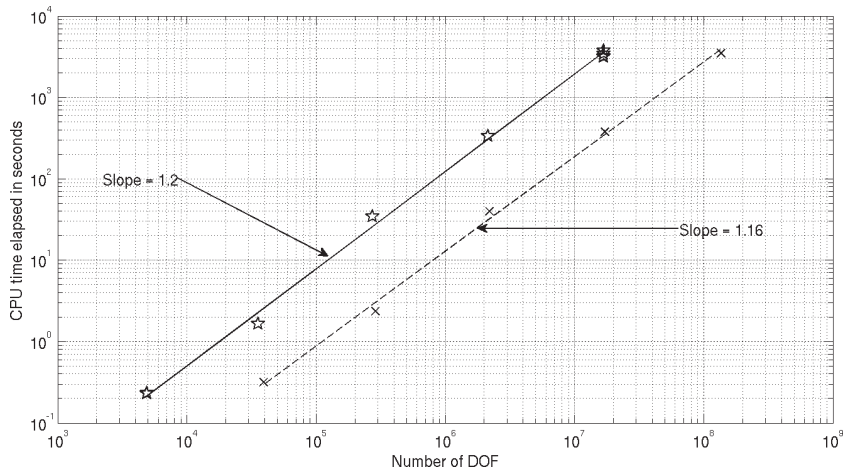

FIG. 4.4. *Scalability of the quasi-static Stokes algorithm based on fixed point transporting a viscosity field using Lagrangian methods. These results are presented for the spinning sphere in a stratified viscosity field. Computation cost per time step is considered for the fifteen first steps. Results compare performances using a twice refined viscosity field (dotted line) or not twice refined (solid line). Abscissa indicates this refinement. Computation performed on single CPU Intel Xeon X*5450 *(*3.00 *GHz). Computation time is given in seconds.*

Figure 4.4 presents the computation time of this turning sphere simulation for one Runge–Kutta 2 step of Lagrangian convection (so it represents two calls of the Stokes solver) using different spatial refinements. Results present a simulation using the same grid size for both viscosity and velocity, and a simulation where viscosity has a refinement two times finer than that of velocity. It shows the computation time is quite similar: the most expensive part of the algorithm is the computation of velocity. With this technique we gain a more accurate computation of interpolations and viscosity gradients for almost the same computation time.

5. Applications to biological films. In this section we are interested in the dynamics of mucus film flow around ciliated epithelium cells covering mammal lungs.

Epithelial cells are of fundamental importance in human lung efficiency (see Figure 1.1 for the anatomical configuration), as mucus acts as a barrier to protect bronchial walls from inhaled dust and pathogen agents [40].

Modeling ciliated cells in motion in a complex fluid (the mucus), on biological interfaces (lung/mucus and mucus/air), is a very challenging problem [9] and the complexity of the full model, that is to say involving all the physics, is still out of reach despite recent dramatic advances in mucus and cilia modeling [58]. Nevertheless, several paths can be followed toward this goal. Some models focus on the mucus itself as a visco-elastic and/or shear-thinning fluid, with the no-slip boundary condition at flat walls [36], or with a continuous boundary condition that mimics a propulsive motion [30]. Another approach is experimental reproduction of ciliated structures beating in a fluid [63, 29]. Yet another strategy is to mesh cilium with accurate laws of elastic beams with three-dimensional [38] or one-dimensional [20] models moving in a homogeneous fluid of constant viscosity [31, 56]. Variable viscosity problems have been explored without geometry [37, 21] or using one-dimensional (1D) or two-dimensional computations [57, 27, 34]. Finally some other works focus on surfactant transport in reduced models compared to experimentations [19] and interaction with branching networks of airways [28].

Mucus is a highly viscous gel essentially composed of polymers and proteins. Out of these proteins, mucins play an important role: they are released by goblet cells [62] situated on bronchial walls. The film is hydrated at the bottom whereas lung air flow dehydrates the top of the film. Thus it is much less viscous at the bottom (where the viscosity is very close to water's) than at the top (where it can be 10 to 10000 times more viscous when pathologies such as cystic fibrosis occur). Hence the rheology of mucus is mainly driven by mucin concentration and quantity of water.

Mucus is a 7–12-μm-thick variable viscosity film covering the lungs, where 4–7-μm-long ciliated cells vibrate at 4 to 20 Hz [61]. This results in a complex geometry in motion (a network of beating cilia) surrounded by a variable viscosity fluid.

The numerical method introduced in the previous sections is especially useful for this kind of large three-dimensional computation, with complex rheology and moving geometry. We consider a fluid with variable viscosity and a set of beating cilia with one way fluid-structure interaction. Indeed, cilia beating is produced by a strong polymerization which makes the one-way coupling acceptable.

Furthermore, shear-thinning and visco-elasticity properties of mucus are not considered here. It will be the focus of future work. These rheology constitutive laws can be brought to the model as supplementary terms in equations. It is much less of a challenge than building efficient algorithms for variable viscosity flows. It will be possible to use the method developed presently for the numerical simulation of mucus in human lungs immediately.

In practice, an average displacement of a few micrometers per second of the mucus film has been observed clinically [61]. So the characteristic Reynolds number of this flow is about 10^{-7} close to the wall and decreases as viscosity increases (in pathological situations).

Computation and characterization of displacement and nonreversibility are important matters in physics and mechanics of propulsion. Nonreversibility is almost systematic in slightly viscous flows since convection terms in Navier–Stokes equations give a naturally nonlinear equation in velocity u. For highly viscous flows, nature has to find specific movements with time-symmetry breaking strategies [51].

Thereafter the model for cilium beating is developed and used for computation of a mucus film with stratified viscosity. The cell is composed of 36 cilia. The beating

model of each cilium is shown to follow a different path on the way back, which induces a positive mean velocity of fluid. This is a crucial point to avoid stagnation of mucus and proliferation of pathogen agents. The resulting velocity is also shown to be comparable with clinical observations.

5.1. A first order hyperbolic model for cilia dynamics and one-way coupling. A cilium beating cycle contains two phases:

- stroke, resulting from a polymerization of microtubules inside the cilium;
- recovery, resulting from the inverse mechanism (depolymerization).

Polymerization is a sufficiently strong chemical reaction to assume a one-way solid-fluid interaction [61, 11].

To compute cilium motion a 1D transport equation is set up along a parameterized curve which describes the cilium. Let us call $P(\zeta, t)$ the curve position describing the cilium at time t at a length l from the cilium basis with $P :]0, 1[\longrightarrow \mathbb{R}^3$. Then $L = \partial_\zeta P$ is the solution to a first order partial differential equation

$$\frac{\partial L}{\partial t} + v(t)\frac{\partial L}{\partial \zeta} = 0. \tag{5.1}$$

The solution to this first order hyperbolic equation is also the solution to the following damped wave equation:

$$\frac{\partial^2 L}{\partial t^2} - v(t)^2\frac{\partial^2 L}{\partial \zeta^2} = -v'(t)\frac{\partial L}{\partial \zeta}. \tag{5.2}$$

Here $v(t)^2$ is an oscillating Young's modulus, quantifying elasticity.

Boundary conditions of this equation are naturally induced by the physics of the problem. $P(0, t) = 0$ ensures that the cilia basis is not moving as it is built into the bronchial wall, and $L(0, t) = \partial_\zeta P(0, t) = g(t)$ generates an oscillating motion up to an angle of $\pi/3$ (from [61]). This combination of boundary conditions ensures that $P(1, t)$ is free to evolve; there is no need for a boundary condition for the outer boundary of a transport equation. This is guaranteed by the positivity of transport field v.

The unconditionally stable implicit Euler method is used for temporal discretization, and upstream finite difference schemes are used for spatial derivatives, since v is positive.

The solution obtained with the first order differential model is very close to the motion obtained by other numerical models and experimental measurements on cultured cells, from the literature. A comparison with 4 different works is displayed on Figure 5.1 (see caption references for instance). Such a reduced 1D beating model is computationally very cheap.

5.2. Full ciliated cell in stratified film. In the following simulation, mucus motion is computed around one epithelial cell assuming periodic boundary conditions with respect to the x and y directions. This states identical cells are periodically located on the Oxy plane, beating in a coordinated manner, following the equations of the last section.

The curvature of the bronchial wall is neglected as computations focus on very small characteristic sizes: a cell width is a few micrometers, while characteristic airway diameter is several millimeters to a centimeter. The computational box is $22 \times 22 \times 10\ \mu m$, with a resolution of $256 \times 256 \times 128$. The top of the box can be considered as

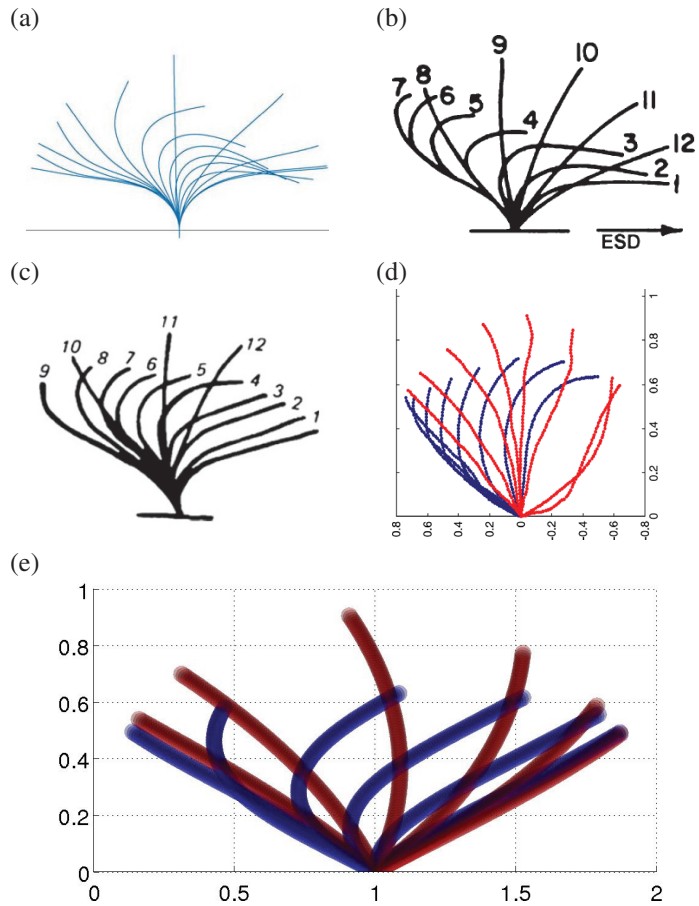

FIG. 5.1. *Trajectory followed by cilia motion. The present model is figure* (e), *in which different states of recovery are displayed in blue, and states of stroke are in red. For a qualitative comparison, figures* (a) *to* (d) *are, respectively, models from Fauci and Dillon* [23], *Gheber and Priel* [24], *Sanderson and Sleigh (culture of rabbit cells)* [54], *and Mitran (3D elastic beams model)* [38].

the top of the biofilm. As the dynamic viscosity ratio between air and mucus is very high, surface tension is also very high, and the curvature of the biofilm surface is very small.

The bottom plane of the computational box is identified as the bronchial wall inducing a no-slip boundary condition. At the top of the computational box mucus is authorized to displace in a tangential direction; it is done using a homogeneous Dirichlet boundary condition for u_z and a homogeneous Neumann boundary condition for both the other components.

To match microscopic observations, the cell is composed of several cilia organized in an elliptic pattern—see Figure 5.2. Each cilia has a different inclination (controlled by initial conditions) and the beating is coordinated.

At initialization, the fluid is two times more viscous at the surface with strong gradients concentrated around cilia tips during recovery. It is expressed with a logistic function of z:

$$\mu(x, y, z, t = 0) = \mu_{water} \left(1 + \frac{1}{1 + \exp(-10(z - 6))}\right).$$

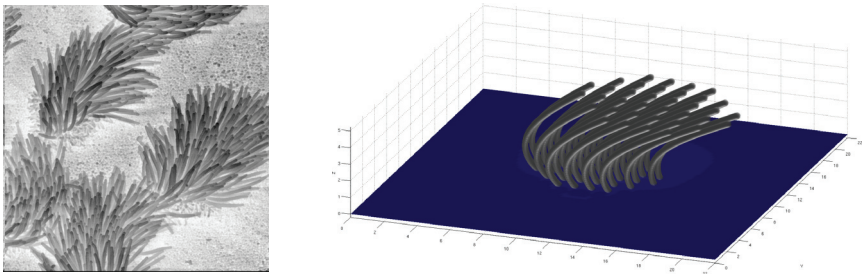

FIG. 5.2. *A scanning electron microscope image of lung trachea epithelium (left, courtesy of C. Daghlian, Dartmouth University), and a snapshot of cell geometry involving* 36 *cilia (right).*

Several snapshots of full cell simulation are displayed on Figure 5.3. Velocity at the biofilm surface is plotted on Figure 5.4, for one cilium and this full ciliated cell.

The mean velocities found are, respectively, 1.6 and 3.2 $\mu m/s$. The beating of the full cell generates much more displacement of the mucus film, which is consistent with medical observations [61]. The model presented in this paper, combining both asymmetric beating and variable viscosity, enables us to model mucociliary clearance with a positive net displacement (in the direction of the trachea). The simplicity of our beating model and fast algorithm developed for both velocity computing and convection enable us to perform 3D computations in a sequential way on a single CPU with a good treatment of nonlinearities.

From a stability point of view, the Lagrangian method enables us to perform very large time steps as there is no transport CFL constraint. For the full cell simulation a standard explicit straightforward discretization of this convection equation with these characteristic time steps would have led to a CFL number of 19. The robustness of Lagrangian methods, even with explicit time integration, make them very reliable for these simulations, where transport is dominant.

Previous simulations have used parameters corresponding to healthy lung conditions with a thin (but not excessively so) mucus film, and reasonable viscosity values. Experiments have shown that mucociliary clearance fails when these parameters change too much: if the mucus film is too thin pathogens can reach the bronchial wall and infections develop whereas stagnation situations (with pathogen proliferation) can occur when both biofilm thickness and viscosity are too large. The next paragraph presents the influence of viscosity gradients and future simulations using various parameter ranges will enable us to identify these pathologic situations, based on the resulting velocity of the mucus film.

Preliminary results discussing the influence of the viscosity gradient on mucociliary clearance with this numerical algorithm are now presented. A stratified initial viscosity field is used:

$$\mu(x, y, z, t = 0) = \mu_{water} \left(1 + \beta \frac{z}{z_M - z_m}\right).$$

So the fluid is β times more viscous at the top of the computational box ($z = z_M$) than at the bottom ($z = z_m$). The mean velocity over four cilium beating periods is computed and presented on Figure 5.5. This result is also compatible with medical observations: when mucus becomes more viscous (in pathologies like cystic fibrosis [12]) mucociliary clearance is less effective.

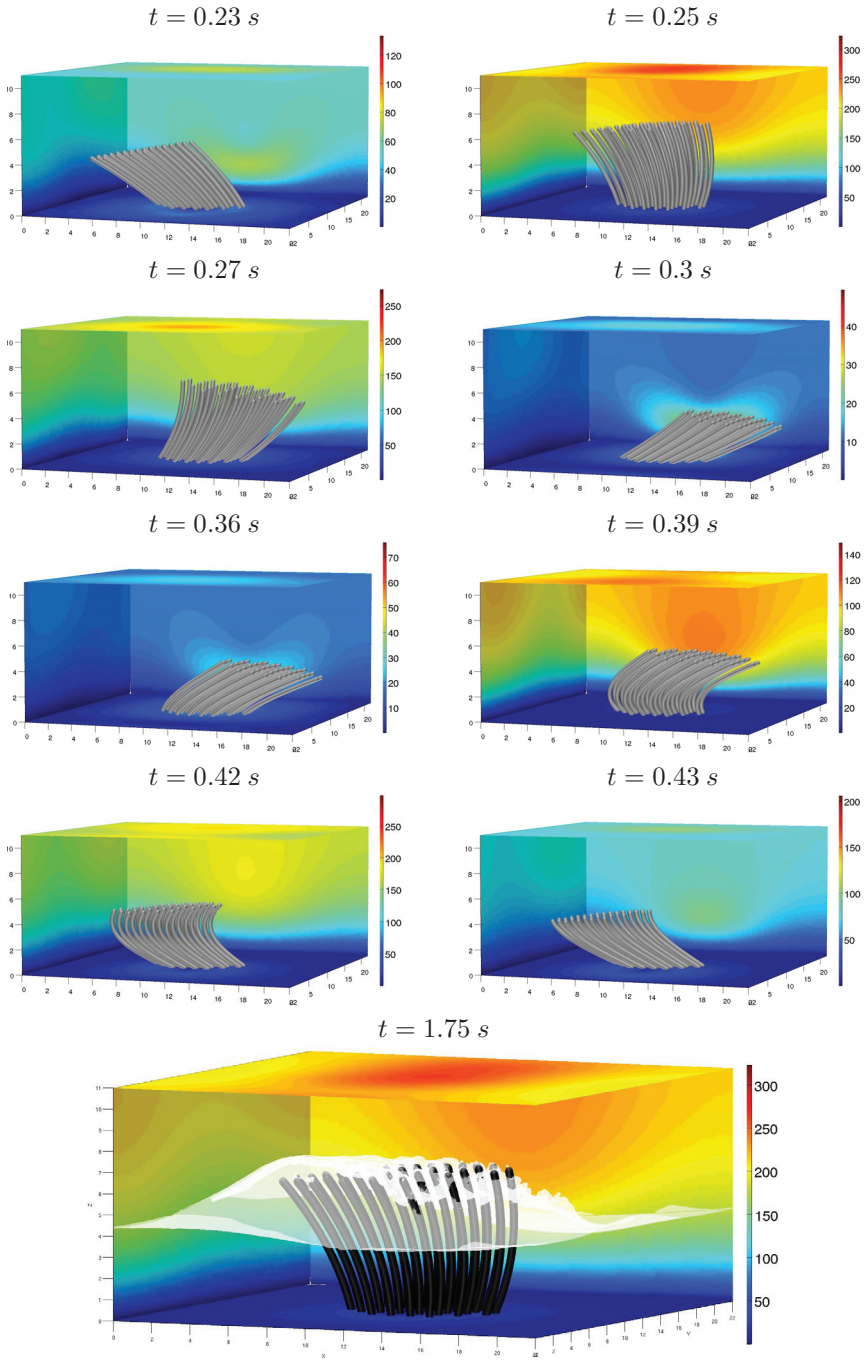

FIG. 5.3. *Velocity norm during the beating of a ciliated epithelium cell composed of* 36 *cilias in a variable viscosity mucus at different times. At initialization fluid is ten times more viscous at the surface than at the bottom (where it is equal to the viscosity of water:* $10^{-3}\ Pa \cdot s$*). The legend is using micrometers as the length unit and micrometers per second as the velocity unit. The bottom picture at time* $t = 1.75\ s$ *(after* 7 *cycles) contains an additional gray isosurface of viscosity at a level of* $4 \cdot 10^{-3}\ Pa \cdot s$*.*

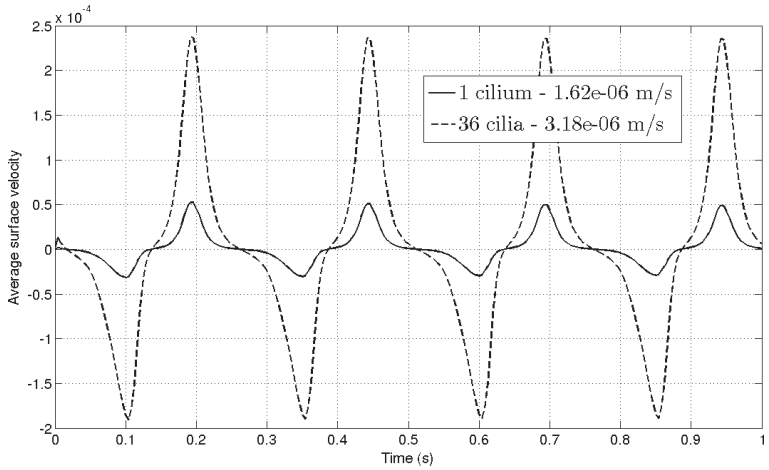

FIG. 5.4. *Surface velocity with respect to time over four periods, for* 1 *cilium (dotted line) and one cell containing* 36 *cilia (solid line). Average velocity over simulation at biofilm surface is reported in the legend.*

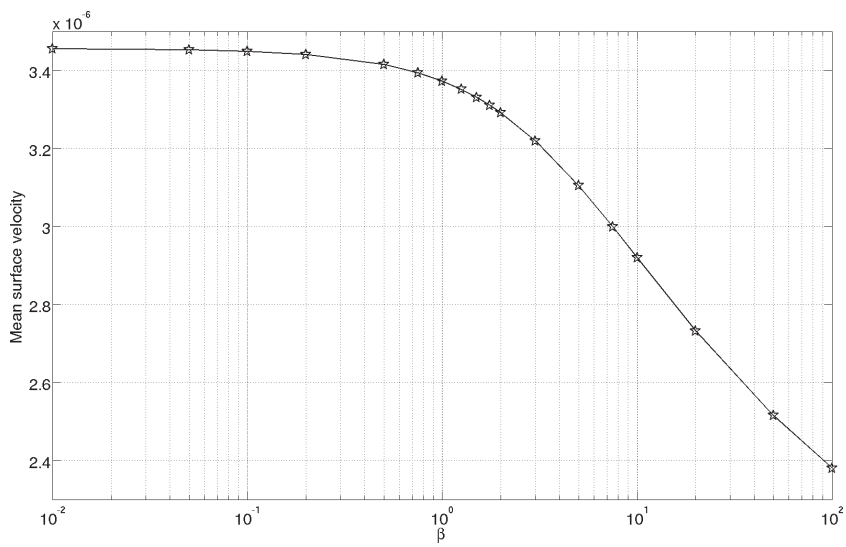

FIG. 5.5. *Influence of viscosity gradient on mean surface velocity over four beating periods. Computations performed for one cell containing* 36 *cilia.*

6. Conclusion. In this article, we have built a new numerical scheme dedicated to the 3D Stokes equations with variable viscosity, coupled with moving geometries. Viscosity follows the transport equation, and consequently solutions to these coupled equations can produce a wide range of nonlinear behaviors.

The method introduced is a hybrid scheme combining an Eulerian scheme for the Stokes equations and a particle method for transport equations. The Stokes equations are split into a pressureless part and a divergence-free projective part. With a structured discretization, fast solvers are used to solve resulting Helmholtz and Poisson equations. An original fixed point algorithm aims at satisfying divergence-free, accurate Dirichlet boundary conditions and nonlinearity induced by variable

viscosity. A penalization method is also coupled with the Stokes momentum equation and a one-way interaction between a moving solid geometry and nonlinear viscous flow is computed. The particle method used for transport equations has good conservation properties, and is very robust as characteristic curves are computed accurately using this novel Stokes solver. The resulting method has been proved to be overall second order. Moreover, the lack of stability conditions due to the Lagrangian feature of the scheme allows the use of large time steps.

Such an efficiency allows us to perform these simulations using sequential computing with a good treatment of nonlinearities. This feature in especially interesting for future developments: massive parallel use of this method for data assimilation.

This scheme allows us to consider large 3D numerical simulations of nonhomogeneous highly viscous flows driven by a moving geometry: here, up to 512^3 simulations over thousands of time steps have been performed. The set of equations is a realistic model for biological flows on the scale of cells, here the ciliated cells of lungs. The resolution of a set of 1D equations generates an asymmetric beating motion used to model the cilia dynamics of a full epithelial cell. By coupling this geometry motion with the Stokes flow, nonreversibility of this system can be studied through the mean velocity of the mucus film. Our results are shown to be coherent with observations; a net displacement of the film surface in the direction of proximal airways is observed, ensuring an efficient mucociliary clearance.

This novel model will be used in order to distinguish different mechanisms of mucus propelling, aiming to identify dominant factors in mucociliary clearance, particularly in the context of cystic fibrosis and aerosol therapy. Non-Newtonian aspects will also be considered in these models without altering the efficiency of the numerical strategy developed here.

This numerical tool could be useful for any simulation of highly viscous flows, where viscosity is transported by the flow, such as for porous media, many biological configurations, or process engineering.

REFERENCES

[1] J. Adams, *MUDPACK: Multigrid Fortran software for the efficient solution of linear elliptic partial differential equations*, Appl. Math. Comput., 34 (1989), pp. 113–146.

[2] G. Allaire, *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*, Oxford University Press, Oxford, 2007.

[3] C. Anderson and C. Greengard, *On vortex methods*, SIAM J. Numer. Anal., 22 (1985), pp. 413–440.

[4] P. Angot, C.-H. Bruneau, and P. Fabrie, *A penalization method to take into account obstacles in incompressible viscous flows*, Numer. Math., 81 (1999), pp. 497–520.

[5] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15-41.

[6] J.T. Beale and A. Majda, *Rates of convergence for viscous splitting of the Navier-Stokes equations*, Math. Comp., 37 (1981), pp. 243–259.

[7] M. Bergmann and A. Iollo, *Modeling and simulation of fish-like swimming*, J. Comput. Phys., 230 (2011), pp. 329–348.

[8] F. Boyer and P. Fabrie, *Eléments d'Analyse pour l'Étude de Quelques Modèles d'Écoulements de Fluides Visqueux Incompressibles*, Springer-Verlag, Berlin, 2006.

[9] R.C. Causey, *Mucus and the mare: How little we know*, Theriogenealogy, 68 (2007), pp. 386–394.

[10] P. Chassaing, R.A. Antonia, F. Anselmet, L. Joly, and S. Sarkar, *Variable Density Fluid Turbulence*, Kluwer Academic, Dordrecht, 2002.

[11] R. Chatelin, P. Poncet, and M. Tokman, *Computational aspects of mucus propulsion by ciliated epithelium*, in Proceedings of 2nd European Microfludics Conference, Toulouse, 2010.

[12] R. Chatelin, P. Poncet, A. Didier, M. Murris-Espin, D. Anne-Archard, and M. Thiriet, *Mucus and ciliated cells of human lung: Splitting strategies for particle methods and* 3*D stokes flows*, IUTAM Symposium on Particle Methods in Fluid Mechanics, Copenhagen, 2012.

[13] A.J. Chorin, *Numerical solution of the Navier-Stokes equations*, Math. Comp., 104 (1968), pp. 745–762.

[14] M. Coquerelle and G.-H. Cottet, *A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies*, J. Comput. Phys., 227 (2008), pp. 9121–9137.

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

[16] G.-H. Cottet and A. Magni, *Accurate, non-oscillatory, remeshing schemes for particle methods*, J. Comput. Phys., 231 (2012), pp. 152–172.

[17] G.-H. Cottet and P. Poncet, *Advances in direct numerical simulations of three-dimensional wall-bounded flows by particle in cell methods*, J. Comput. Phys., 193 (2003), pp. 136–158.

[18] B. Couët, O. Buneman, and A. Leonard, *Simulation of three-dimensional flows with vortex-in-cell methods*, J. Comput. Phys., 39 (1981), pp. 305–308.

[19] R.V. Craster and O.K. Matar, *Surfactant transport on mucus films*, J. Fluid Mech., 425 (2000), pp. 235–258.

[20] A. Dauptain, J. Favier, and A. Bottaro, *Hydrodynamics of ciliary propulsion*, J. Fluids Struct., 24 (2008), pp. 1156–1165.

[21] S. Enault, D. Lombardi, P. Poncet, and M. Thiriet, *Mucus dynamics subject to air and wall motion*, ESAIM Proc., 30 (2010), pp. 125–141.

[22] J.V. Fahy and B.F. Dickey, *Airway mucus function and dysfunction*, New Engl. J. Med., 363 (2010), pp. 2233–2247.

[23] L.J. Fauci and R. Dillon, *Biofluidmechanics of reproduction*, Annu. Rev. Fluid Mech., 38 (2006), pp. 371–394.

[24] Gheber and Priel, *Extraction of cilium beat parameters by the combined application of photoelectric measurements and computer simulation*, Biophys. J., 72 (1997), pp. 449–462.

[25] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulation*, J. Comput. Phys., 73 (1987), pp. 325–348.

[26] J.L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 6011–6045.

[27] S. Gueron and N. Liron, *Ciliary motion modeling, and dynamic multicilia interactions*, Biophys. J., 63 (1992), pp. 1045–1058.

[28] D. Halpern, J.L. Bull, and J.B. Grotberg, *The effect of airway wall motion on surfactant delivery*, J. Biomech. Engrg., 126 (2004), pp. 410–419.

[29] J. Hussong, N. Schorr, J. Belardi, O. Prucker, J. Rühe, and J. Westerweel, *Experimental investigation of the flow induced by artificial cilia*, Lab. Chip, 11 (2011), p. 2017.

[30] J. Hussong, W.-P. Breugem, and J. Westerweel, *A continuum model for flow induced by metachronal coordination between beating cilia*, J. Fluid Mech., 684 (2011), pp. 137–162.

[31] P.G. Jayathilake, Z. Tan, D.V. Le, H.P. Lee, and B.C. Khoo, *Three-dimensional numerical simulations of human pulmonary cilia in the periciliary liquid layer by the immersed boundary method*, Comput. Fluids, 67 (2012), pp. 130–137.

[32] P. Koumoutsakos, *Multiscale flow simulations using particles*, Annu. Rev. Fluid Mech., 37 (2005), pp. 457–487.

[33] S. K. Lai, Y.-Y. Wang, D. Wirtz, and J. Hanes, *Micro- and macrorheology of mucus*, Adv. Drug Delivery Rev., 61 (2009), pp. 86–100.

[34] W.L. Lee, P.G. Jayathilake, Z. Tan, D.V. Le, H.P. Lee, and B.C. Khoo, *Muco-ciliary transport: Effect of mucus viscosity, cilia beat frequency and cilia density*, Comput. Fluids, 49 (2011), pp. 214–221.

[35] P. Li, H. Johnston, and R. Krasny, *A Cartesian treecode for screened Coulomb interactions*, J. Comput. Phys., 228 (2009), pp. 38–58.

[36] O.K. Matar and P.D.M. Spelt, *Dynamics of thin free films with reaction-driven density and viscosity variations*, Phys. Fluids, 17 (2005), 122102.

[37] B. Mauroy, C. Fausser, D. Pelca, J. Merckx, and P. Flaud, *Towards the modeling of mucus draining from human lung: Role of the geometry of the airway tree*, Phys. Biol., 8 (2011), 056006.

[38] S. Mitran, *Metachronal wave formation in a model of pulmonary cilia*, Comput. Struct., 85 (2007), pp. 763–774.

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

[40] M. Murris-Espin, M. Aubert, E. Bosdure, and J.-C. Dubus, *Influenza vaccination coverage in patients with cystic fibrosis followed at* 12 *care centers in the Greater South Region of France for the season* 2005/2006, Vaccine, 26 (2008), pp. 5612–5618.

[41] R.A. Nicolaides and X. Wu, *Analysis and convergence of the MAC scheme* II: *Navier-Stokes equations*, Math. Comp., 65 (1996), pp. 29–44.

[42] M. El Ossmani and P. Poncet, *Efficiency of multiscale hybrid grid-particle vortex methods*, Multiscale Model. Simul., 8 (2010), pp. 1671–1690.

[43] M.L. Ould-Sahili, G.-H. Cottet, and M. El Hamraoui, *Blending finite-difference and vortex methods for incompressible flow computations*, SIAM J. Sci. Comput., 22 (2000), pp. 1655–1674.

[44] R. Peyret and T.D. Taylor, *Computational Methods for Fluid Flow*, Springer-Verlag, New York, 1983.

[45] P. Ploumhans and G.S. Winckelmans, *Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry*, J. Comput. Phys., 165 (2000), pp. 354–406.

[46] P. Poncet, *Topological aspects of three-dimensional wakes behind rotary oscillating circular cylinders*, J. Fluid Mech., 517 (2004), pp. 27–53.

[47] P. Poncet, *Analysis of direct three-dimensional parabolic panel methods*, SIAM J. Numer. Anal., 45 (2007), pp. 2259–2297.

[48] P. Poncet, R. Hildebrand, G.-H. Cottet, and P. Koumoutsakos, *Spatially distributed control for optimal drag reduction in cylinder wakes*, J. Fluid Mech., 599 (2008), pp. 111–120.

[49] P. Poncet, *Analysis of an immersed boundary method for three-dimensional flows in vorticity formulation*, J. Comput. Phys., 228 (2009), pp. 7268–7288.

[50] E. Puchelle, J.M. Zahm, and C. Duvivier, *Spinability of bronchial mucus. Relationship with visco-elasticity and mucous transport properties*, Biorheology, 20 (1983), pp. 239–250.

[51] E.M. Purcell, *The efficiency of propulsion by a rotating flagellum*, Proc. Natl. Acad. Sci. USA, 94 (1997), pp. 11307–11311.

[52] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer, Berlin, 2008.

[53] J.K. Salmon and M.S. Warren, *Skeletons from the treecode closet*, J. Comput. Phys., 111 (1994), pp. 136–155.

[54] M.J. Sanderson and M.A. Sleigh, *Ciliary activity of cultured rabbit tracheal epithelium, Beat pattern and metachrony*, J. Cell. Sci., 47 (1981), pp. 331–334.

[55] D.L. Sandoval, *The Dynamics of Variable-Density Turbulence*, Ph.D. thesis, Los Alamos National Laboratory, Albuquerque, NM (United States), 1995.

[56] D.J. Smith, E.A. Gaffney, and J.R. Blake, *Discrete cilia modelling with singularity distributions: Application to the embryonic node and the airway surface liquid*, Bull. Math. Biol., 69 (2007), pp. 1477–1510.

[57] D.J. Smith, E.A. Gaffney, and J.R. Blake, *A viscoelastic traction layer model of muco-ciliary transport*, Bull Math Biol., 69 (2007), p. 1449.

[58] D.J. Smith, E.A. Gaffney, and J.R. Blake, *Modelling mucociliary clearance*, Resp. Physio. Neuro., 163 (2008), pp. 178–188.

[59] P. Swarztrauber and R. Sweet, *Efficient FORTRAN Subprograms for the Solution of Elliptic Partial Differential Equations*, Technical report NCAR Technical Note-TN/IA-109, Boulder, CO, 1975.

[60] R. A. Sweet, *A parallel and vector variant of the cyclic reduction algorithm*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 761–765.

[61] M. Thiriet, *Tissue Functioning and Remodeling in the Circulatory and Ventilatory Systems (Vol.* 5*)*, Ser. Biomath. Biomechan. Model. Circulatory Ventilatory Systems, Springer, New York, 2012.

[62] D.J. Thornton and J.K. Sheehan, *From mucins to mucus: Toward a more coherent understanding of this essential barrier*, Proc. Amer. Thoracic Soc., 1 (2004), pp. 54–61.

[63] T.S. Yu, E. Lauga, and A.E. Hosoi, *Experimental investigations of elastic tail propulsion at low reynolds number*, Phys. Fluids, 18 (2006), p. 091701.