INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **61**:1247–1272
Published online 20 February 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2007

On the Fourier representation of elastic immersed boundaries

F. Pacull[1] and M. Garbey[2,*,†]

[1]*Fluorem—Ecully, France*
[2]*Department of Computer Science, University of Houston, Houston, TX, U.S.A.*

SUMMARY

This paper presents an efficient treatment of fluid/elastic–structure interactions that takes advantage of the Fourier representation of immersed boundaries. We assume that the fluid is incompressible with uniform density and viscosity and that the immersed boundaries have fixed topologies. These elastic bodies can have large deformations and evolve anywhere within the fluid domain. They may be thick and are assumed to be piecewise smooth. We process the fluid–structure coupling with the immersed boundary method of C. S. Peskin. We can take advantage of the Fourier representation of the immersed bodies in many ways. First, the use of Fourier expansions allows us to filter out the high frequencies of the spatial oscillations along the boundary vectors. Second, we can work with a smaller number of boundary points to represent the interface, while preserving the same level of accuracy as long as enough points are used in the force spreading process. Finally, the harmonic information gathered by the Fourier coefficients is useful to control some global properties of the immersed boundaries. For example, we introduce a technique that corrects the volume conservation issue of closed immersed boundaries by performing constrained optimization in the Fourier space. We illustrate our method with two applications: one is a suspension flow with a large number of elastic 'bubbles', the other is an interesting case of artificial motion based on inertia rather than on flapping fins or flagella.

Received 13 April 2008; Revised 18 December 2008; Accepted 7 January 2009

KEY WORDS: fluid–structure interaction; immersed boundary method; Navier Stokes; PDE; elastic membrane; low Reynolds flow

1. INTRODUCTION

The main approaches to simulate fluid flows in complex moving geometries use either moving-grid or artificial boundary methods [1–6]. This former type of methods imply re-meshing processes, which are highly expensive computationally in the fluid/elastic–structure interaction cases that involve large structure deformations. On the other hand, in the artificial boundary methods, the effect of the boundary is remotely applied to the fluid by a constraint/penalty on the governing

*Correspondence to: M. Garbey, Department of Computer Science, University of Houston, Houston, TX, U.S.A.
†E-mail: garbey@cs.uh.edu

equations or a locally modified discretization/stencil: the fluid mesh is then globally independent of the moving interface, described by Lagrangian coordinates, and the effect of the interaction is artificially introduced into the fluid variables at the Eulerian grid points neighboring the interface.

Many of the applications of fluid/flexible–body interaction simulations with large deformation are within the field of bio-engineering. The accuracy of experimental data measurements of bio-fluids is usually not very high and one may prefer to emphasis the robustness of the numerical method over the high-order accuracy of the solution process. The simulation of large movements/deformations of realistic elastic boundaries should be versatile, efficient and easy to implement on modern computer architectures. This leads us to the immersed boundary method (IBM) pioneered by C. S. Peskin [5, 7] to model flow patterns around heart valves. A major advantage of the IBM is the high level of uniformity of mesh and stencil, avoiding the critical interpolation processes of the cut-cell/direct methods. Based on the standard finite-difference method, the IBM allows highly efficient domain decomposition techniques to be implemented. In other words, the difficulty of simulating dynamical interaction phenomena with complex geometries can be overcome by implementing, in a fast and easy way, large fine-grid parallel computations that takes full advantage of a uniform stencil on an extended regular domain, as described in [8, 9], for blood flow applications. While the examples of this paper are restricted to two-dimensional flows, the concepts introduced here apply to large three-dimensional bio-flows [10].

The plan of the paper is as follows. Section 2 describes a full-pressure fractional-step method for the incompressible Navier–Stokes equations, which achieves second-order temporal accuracy in fluid-only simulations. In Section 3, we recall two standard benchmark problems to illustrate some of the characteristics of the IBM. Section 4 defines the Fourier representation of the immersed boundary. Section 5 introduces a strict global volume conservation technique. Section 6 shows the benefit of reducing the Fourier space. Section 7 illustrates the technique for the two benchmark problems of Section 3. Section 8 gives two interesting applications. Section 9 concludes this study.

2. THE DISCRETIZATION OF THE IBM

We are going to recall the IBM formulation and present the spatial and temporal discretizations that we have chosen. The primitive variables are V and P, respectively, the velocity and pressure of the fluid, whose physical parameters are the uniform viscosity μ and the uniform density ρ. The fluid domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ is described by the Cartesian coordinate vector x. $\Gamma \subset \Omega$ is the immersed elastic boundary, whose curvilinear dimension is m $(m \leqslant d)$. X is the Lagrangian position vector of Γ, expressed in the d-dimensional Cartesian referential. The Lagrangian vector f is the local elastic force density along Γ, also expressed in the Cartesian referential. f is projected onto Ω to get the Eulerian vector field F, the fluid force applied by the immersed elastic boundary. While we restrict the presentation to one immersed elastic boundary there can be an arbitrary number of them in practice.

If we note $s \in (0, 1)^m$ the curvilinear coordinates of any points along Γ (normalized here in this section) and $t \in [0, t_{\max}]$ the time variable, we can summarize the different mappings as follows:

$$V : (x, t) \in \Omega \times [0, t_{\max}] \longrightarrow \mathbb{R}^d$$

$$P : (x, t) \in \Omega \times [0, t_{\max}] \longrightarrow \mathbb{R}$$

$$X : (s, t) \in (0, 1)^m \times [0, t_{\max}] \longrightarrow \Omega$$

DOI: 10.1002/fld

$$f:(s,t)\in(0,1)^m\times[0,t_{\max}]\longrightarrow\mathbb{R}^d$$

$$F(x,t)\in\Omega\times[0,t_{\max}]\longrightarrow\mathbb{R}^d$$

2.1. The IBM equations

A complete and accurate introduction to the IBM can be found in [7]. Here is a brief description of the fluid/elastic-interface model unified into a set of coupled PDEs. The incompressible Navier–Stokes system writes:

$$\rho\left[\frac{\partial V}{\partial t}+(V\cdot\nabla)V\right]=-\nabla P+\mu\Delta V+F \tag{1}$$

$$\nabla\cdot V=0 \tag{2}$$

The IBM requires the extrapolation of the Lagrangian vector f into the Eulerian vector field F from the RHS of Equation (1). In the IBM of Peskin, we use a distribution of Dirac delta functions δ for that purpose:

$$F(x,t)=\int_\Gamma f(s,t)\delta(x-X(s,t))\,\mathrm{d}s=\begin{cases}f(s,t) & \text{if } x=X(s,t)\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

The motion of the immersed boundary should match the motion of the neighboring fluid particles thanks to a no-slip boundary condition. Equation (4) approximates this no-slip boundary condition using the Dirac delta function as an interpolating tool for V, from Ω to Γ:

$$\frac{\partial X(s,t)}{\partial t}=\int_\Omega V(x,t)\delta(x-X(s,t))\,\mathrm{d}x=\begin{cases}V(X(s,t),t) & \text{if } x=X(s,t)\\ 0 & \text{otherwise}\end{cases} \tag{4}$$

The immersed boundary obeys a linear elastic model. We use Hooke's law of elasticity, i.e. the tension $\mathscr{T}$ of the immersed boundary is a linear function of the strain. For a one-dimensional boundary, we have:

$$\mathscr{T}(s,t)=\sigma\left|\frac{\partial X(s,t)}{\partial s}\right| \tag{5}$$

where σ is the boundary elasticity coefficient. The local elastic force density f is defined as:

$$f(s,t)=\frac{\partial(\mathscr{T}(s,t)\tau(s,t))}{\partial s},\quad \tau(s,t)=\frac{\partial X(s,t)/\partial s}{|\partial X(s,t)/\partial s|} \tag{6}$$

τ is the unit tangent vector to Γ. Finally, by plugging Equation (5) into Equation (6), we get:

$$f(s,t)=\sigma\frac{\partial^2 X(s,t)}{\partial s^2} \tag{7}$$

We note that this tension is never negative. But many other physical settings are equally feasible, we refer to [7] for a review. The practical implementation of the IBM of Peskin offers dozens of different possibilities regarding the choice of the temporal scheme, the space discretization, the discrete approximation of the Dirac function and so on. There is clearly a compromise between

DOI: 10.1002/fld

the stability of the scheme that suffers from the sharp numerical interface in the pressure field, theoretically discontinuous, and accuracy that needs this numerical feature. We refer to the thesis of the first author [11] and its bibliography for an extensive comparison of possible implementations against standard benchmark problems such as the oscillation/relaxation of a stretched elastic 'bubble' toward its equilibrium, or the motion of an elastic 'bubble' immersed in a cavity flow. These test cases are described in Section 3. We are now going to present the basic algorithmic choices made, which perform well. First we describe the time-stepping scheme.

2.2. The Temporal scheme

We use a projection method for the time integration of the incompressible Navier–Stokes equations. We have chosen the P2 fractional-step method described by S. Armfield *et al.* in [12] that is second order in time. The time integration scheme writes as follow. Using the momentum equations with the pressure given from a previous time step, one obtains a prediction V^*, of the velocity V^{n+1},

$$\rho\left[\frac{V^*-V^n}{\Delta t}\right]=-\nabla P^{n-1/2}+\frac{\mu}{2}\Delta(V^*+V^n)$$
$$-\frac{\rho}{2}[3H(V^n)-H(V^{n-1})]+\frac{3}{2}F^n-\frac{1}{2}F^{n-1} \tag{8}$$

where $H(V)$ is for the convective term and Δt the time step. One needs to add a correction to V^*, since it does not satisfy the divergence-free condition. If we note Π the corresponding scalar field, the Hodge decomposition of V^* writes:

$$V^*=V^{n+1}+\frac{\Delta t}{\rho}\nabla\Pi \tag{9}$$

$$\nabla\cdot V^{n+1}=0 \tag{10}$$

Taking the divergence of Equation (9) leads to the pressure correction solution of the Poisson problem:

$$\Delta\Pi=\frac{\rho}{\Delta t}\nabla\cdot V^* \tag{11}$$

V^{n+1} is the divergence-free exact projection of V^* and the corresponding projection operator is:

$$[I-\nabla\Delta^{-1}\nabla\cdot]$$

From the prediction Equation (8), the pressure Equation (11) and the correction Equation (9), one has formally the second-order time stepping scheme:

$$\rho\left[\frac{V^{n+1}-V^n}{\Delta t}\right]=-\nabla P^{n+1/2}+\frac{\mu}{2}\Delta(V^{n+1}+V^n)$$
$$-\frac{\rho}{2}[3H(V^n)-H(V^{n-1})]+\frac{3}{2}F^n-\frac{1}{2}F^{n-1} \tag{12}$$

DOI: 10.1002/fld

Following [12], the pressure field is updated with:

$$P^{n+1/2} = P^{n-1/2} + \Pi - \frac{\mu \Delta t}{2\rho} \Delta \Pi \tag{13}$$

If needed, P^{n+1} can be computed from an extrapolation formula as follows:

$$P^{n+1} = \tfrac{3}{2} P^{n+1/2} - \tfrac{1}{2} P^{n-1/2} \tag{14}$$

The elastic body motion in Equation (4) is discretized using the midpoint rule for the time derivative, the trapezoidal rule and an Adams–Bashforth scheme, respectively, for the fluid velocity and the boundary position extrapolated at time step $n+\frac{1}{2}$:

$$\frac{X^{n+1} - X^n}{\Delta t} = \int_\Omega \left(\frac{V^{n+1} + V^n}{2} \right) \delta \left(x - \frac{3}{2} X^n + \frac{1}{2} X^{n-1} \right) \mathrm{d}x \tag{15}$$

Finally, the local elastic force density f^{n+1} is computed as a function of X^{n+1} and extrapolated onto Ω:

$$F^{n+1} = \int_\Gamma f^{n+1} \delta(x - X^{n+1}) \, \mathrm{d}s \tag{16}$$

After the temporal scheme, here is a description of the spatial approximations.

2.3. Spatial discretization

The description of the spatial discretization is restricted here to two-dimensional space problems. We recall first the definition of the staggered grid discretization [13].

2.3.1. The staggered grid. Let us consider a rectangular domain $\Omega = [0, L] \times [0, L]$ and a uniform mesh in both space directions, with space step $h = L/(N-1)$. We use the notations (x_1, x_2) for the space coordinates, and (u, v) for the fluid velocity components. As shown in Figure 1, the primitive variables discretized on the staggered grid are:

- $u_{i,j} = u(ih, (j+\frac{1}{2})h)$ with $0 \leqslant i \leqslant N-1$ and $0 \leqslant j \leqslant N-2$,
- $v_{i,j} = v((i+\frac{1}{2})h, jh)$ with $0 \leqslant i \leqslant N-2$ and $0 \leqslant j \leqslant N-1$,
- $P_{i,j} = P((i+\frac{1}{2})h, (j+\frac{1}{2})h)$ with $0 \leqslant i, j \leqslant N-2$.

We chose the staggered over the collocated or regular finite-difference grids for several reasons. On the staggered grid with second-order discretization, the widths of the divergence operator and Laplace operator stencils are equal, respectively, to one space step and two space steps. The Laplace operator stencil is always larger than the divergence one in order to have a good propagation of the information in the discretized pressure equation (11). Thus, for regular grids, the discretization will require a four space-steps stencil width for the Laplace operator and a two space-steps stencil width for the divergence operator. The pressure accuracy benefits from the smaller stencil of staggered grids, specifically in the regions with sharp gradients. There is, in principle, an additional advantage in associating the IBM with staggered grids, which is that discrete Dirac delta functions with minimum support can be used. In principle, the smaller the Dirac delta function support, the less the smear-out we have in the numerical approximation of the pressure jump at the immersed

DOI: 10.1002/fld

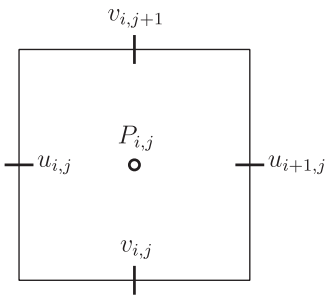

Figure 1. Staggered grid—2D primitive variables discrete locations.

boundary location. In reality, it is not so clear because of possible numerical instabilities that leads to oscillations. Finally, as shown in [14] for regular fluid cases, the collocated grid, which has the same pressure stencil as the staggered grid, is more dissipative and does not conserve well kinetic energy. It is only conserved through time to the first order with the collocated grid, against third order with staggered grids. We are now going to describe the discretization of the convective term of the momentum equation.

2.3.2. The Convective term. The advective form of the convective term is used, as described by Y. Morinishi *et al.* in [14]. We have:

$$H(V)=(V\cdot\nabla)V=u\frac{\partial}{\partial x_1}\begin{bmatrix}u\\v\end{bmatrix}+v\frac{\partial}{\partial x_2}\begin{bmatrix}u\\v\end{bmatrix} \tag{17}$$

where u and v are the velocity components of V. We introduce the following finite-difference notations:

- $D_{x_1}(.)$ and $D_{x_2}(.)$ are, respectively, the x_1 and x_2 first derivatives, evaluated at the middle of two neighboring points of the Cartesian grid,
- Similarly, $A_{x_1}(.)$ and $A_{x_2}(.)$ are, respectively, the x_1 and x_2 simple arithmetic mean, also evaluated at the middle of two neighboring points of the Cartesian grid.

The discretization of the convective term $H(V)=[H(V)^u, H(V)^v]^{\mathrm{T}}$, written in advective form, is:

$$H(V)^u=A_{x_1}(A_{x_1}(u)D_{x_1}(u))+A_{x_2}(A_{x_1}(v)D_{x_2}(u))$$

$$H(V)^v=A_{x_1}(A_{x_2}(u)D_{x_1}(v))+A_{x_2}(A_{x_2}(v)D_{x_2}(v))$$

which leads to, on the staggered mesh:

$$\begin{aligned}H(V)^u_{i,j}=\frac{1}{4h}[&(u^2_{i+1,j}-u^2_{i-1,j})\\&+(u_{i,j+1}-u_{i,j})(v_{i,j+1}+v_{i-1,j+1})+(u_{i,j}-u_{i,j-1})(v_{i,j}+v_{i-1,j})]\end{aligned} \tag{18}$$

$$\begin{aligned}H(V)^v_{i,j}=\frac{1}{4h}[&(u_{i+1,j}+u_{i+1,j-1})(v_{i+1,j}-v_{i,j})+(u_{i,j}+u_{i,j-1})(v_{i,j}-v_{i-1,j})\\&+(v^2_{i,j+1}-v^2_{i,j-1})]\end{aligned} \tag{19}$$

DOI: 10.1002/fld

Table I. Maximum stable time step and corresponding Courant number, with respect to the elasticity coefficient $\sigma \cdot \rho = 1$, $\mu = 0.1$.

σ	1	10	100	1000	10 000
$\Delta t_{\max}$	1.3×10^{-2}	2.5×10^{-3}	4.8×10^{-4}	1.1×10^{-4}	2.5×10^{-5}
max(Cour. numb.)	0.16	0.15	0.12	0.12	0.11

We implemented this scheme on the two standard test cases mentioned earlier at the end of Section 2.1, and came to the conclusion that the instability of centered convective schemes does not penalize the IBM: the stiffness of the fluid–structure coupling scheme is the bottleneck of the scheme stability. For example, in the numerical simulation of the unit square cavity flow problem carrying an elastic 'bubble', with parameter values $\rho = 1$, $\mu = 0.1$, we observe that the maximum 'stable' time step $\Delta t_{\max}$ allowed with explicit time stepping is a small fraction of the maximum time step allowed by the CFL condition based on the maximum velocity—see Table I.

A different staggered discretization than (18), (19), that is the divergence form of the convective term, conserves kinetic energy of the flow up to the third order with respect to time in fluid-only cases [14]. However, we did not measure any improvement over the standard advective form in our fluid/structure interaction test cases. We found that conservation of energy in the IBM is mainly limited by the boundary treatment rather than the convective term discretization. We are now going to describe the numerical representation of the Dirac delta function that plays a central role in the IBM method, linking the Eulerian to the Lagrangian quantities.

2.3.3. The discrete Dirac delta function. We use in this study the traditional function δ_h that has a $4h$ support, as described in [7]:

$$\delta_h(x_1, x_2) = \frac{1}{h}\phi\left(\frac{x_1}{h}\right)\frac{1}{h}\phi\left(\frac{x_2}{h}\right)$$

with:

$$\phi(r) = \begin{cases} \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}) & \text{if } |r| \leqslant 1 \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}) & \text{if } 1 < |r| \leqslant 2 \\ 0 & \text{if } |r| > 2 \end{cases} \tag{20}$$

The main cause of inaccuracy of the method is the smearing of the pressure jump, due to the regularized dipole source term in the pressure equation, corresponding to the divergence of the discrete force term equation (11). The choice (20) made in [11] seems overall to be a better compromise between the stability of the scheme, with respect to time stepping, and the accuracy of the solution than the approximation with $3h$ support introduced in [15] for staggered grids or the piecewise cubic representation of [16]. We are now going to comment on some implementation details.

DOI: 10.1002/fld

2.3.4. Implementation details. As the IBM requires the use of regular grids, Lagrangian variables are quite simple to manipulate in the code. It is particularly easy to compute the list of the Cartesian grid points that are inside of the discrete Dirac delta function support, for each discrete point of the immersed boundary. For example, for a general discrete Dirac delta function of support $2nh$, with $n=2,3,\ldots$, we use these inner loops, expressed here in the MATLAB language, to span the intersection of the u-grid with the delta support around $X \equiv X(s,t)$:

```
iX = round(X/h)+1;
jX = round(X/h-0.5)+1;
for i = max(1,iX-n):min(iX+n,N),
   for j = max(1,jX-n):min(jX+n,N-1),
:
   end
end
```

Similarly, the following inner loop is used for the v-grid:

```
iX = round(X/h-0.5)+1;
jX = round(X/h)+1;
for i = max(1,iX-n):min(iX+n,N-1),
    for j = max(1,jX-n):min(jX+n,N),
:
    end
end
```

Thus, each of the discretized boundary points will correspond to a span of $(2n+1)^2$ points in the 2D Cartesian grid, therefore, reducing the amount of computation for Equations (3), (4) of the immersed boundary treatment.

Dealing with regular grids permits the extensive use of fast parallel solver for the pressure. There are numerous efficient elliptic solvers available in the literature and we have used the Aitken-Schwarz algorithm [8, 9, 17] that performs very well on Beowulf cluster with Gigabit ethernet network. The parallel implementation of the time stepping in the momentum equation is straightforward with explicit time stepping but may require more work for semi-implicit versions. Because the time step must be small in practice compared with the one limited by the CFL condition, the operator corresponding to time stepping is close to identity. We found that the multigrid method with few levels of grids gives a very satisfactory efficient implementation for this type of low Reynolds number flow. One problem that we have not addressed in our parallel implementation is the possible unbalanced load of the computational nodes, due to the overhead of the local boundary treatment. The larger the number of processors, the more critical it becomes according to the Amdhal's law. This is not an issue for the suspension flow with a significant densities of immersed bodies we considered in Section 6. The bubbles in this cases are roughly equally distributed. On the contrary, one needs to use an adaptive domain decomposition method for the case of sparse isolated bubbles to load balance the parallel computation with a large number of processors.

We are now going to underline some particular features of the numerical solutions of the IBM, with some classical test cases that have been well documented in the literature [11, 18, 19], which also allowed us to validate the codes.

DOI: 10.1002/fld

3. TWO SIMPLE TEST CASES

In the following numerical experiments, we are particularly interested in regarding volume conservation, non-physical oscillations and stability.

The test case that we use is the 2D 'bubble' test-case: a closed infinitely thin elastic moving boundary is immersed inside an incompressible fluid. The domain is a closed square box $\Omega=[0,1]^2$. At the beginning the fluid velocity is null and the elastic boundary is stretched out, then the stored elastic energy is converted into kinetic energy. The boundary conditions are homogeneous:

$$V^n|_{\partial\Omega}=0 \tag{21}$$

$$V^*|_{\partial\Omega}=0 \tag{22}$$

$$\left.\frac{\partial\Pi}{\partial\eta}\right|_{\partial\Omega}=0 \tag{23}$$

where η is the normal vector to $\partial\Omega$.

A variation of this test case consists in imposing a quadratic stationary velocity profile ($\leqslant v_w$) on a wall of the squared box [20], in order to immerse the elastic boundary inside a driven cavity flow:

$$v(1,y)=4\,v_w\,y(1-y) \tag{24}$$

We will now present two different well-defined cases, named A and B, which emphasize on the volume conservation of the IBM.

3.1. Volume increasing with respect to time (Test case A)

The elastic bubble is initially a circle of radius 0.2 with Area ≈ 0.1257 and center location $(0.75, 0.5)$. The cavity flow is driven by a vertical velocity along the right wall. The discrete and physical parameters in the simulation are $\mathrm{d}t\approx 5.2\times10^{-4}$, $N=32$, $M=128$, $\sigma=1$, $\rho=1$, $\mu=0.1$, $v_w=-10$. The immersed boundary rapidly starts spinning on itself along a fixed streamline of the fluid (see Figure 2). However, we note that the volume of the 'bubble' is linearly increasing, as shown on Figure 3. We obtain the opposite feature with the following test case.

3.2. Volume decreasing with respect to time (Test case B)

The elastic bubble is initially a circle of radius 0.35 with Area ≈ 0.38485 and center location $(0.5, 0.5)$. The discrete and physical parameters in the simulation are $\mathrm{d}t\approx 5.2\times10^{-7}$, $N=32$, $M=128$, $\sigma=1000$, $\rho=1$, $\mu=0.1$, $v_w=0$. Unlike in test case A, the boundary is already at equilibrium at time zero, the fluid being at rest, and we note that the volume of the 'bubble' is slowly decreasing, as shown in Figure 4.

As the 'bubble' is not moving, we can easily observe the non-physical oscillations on the local elastic force density on Figure 5. The elasticity law having no dissipative term, the Lagrangian discretized points have a tendency to move toward their neighbors along the continuous boundary position when σ is large, even if there is no fluid or structure displacement. This is a by-product of the poor quality of the discrete approximation of the pressure around the jump as well as of the chaotic propagation of the error through the force and velocity interpolation processes. These

DOI: 10.1002/fld

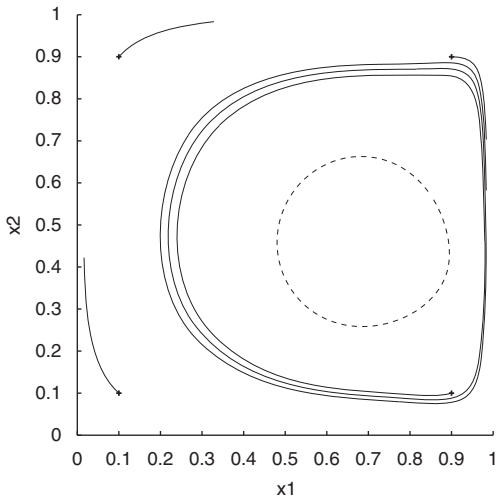

Figure 2. Test case A from Section 3.1. 4 streamlines are plotted. The immersed boundary is represented by the dotted line. $t=10$.

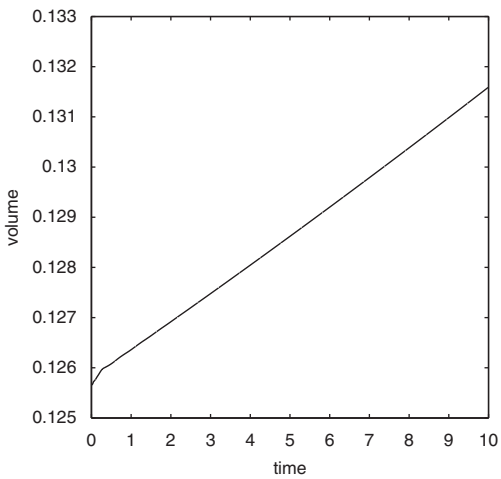

Figure 3. Evolution of the closed immersed area with respect to time in test case A from Section 3.1. $0\leqslant t\leqslant 10$.

oscillations are limited by a condition on the discrete approximation of δ stated by C. S. Peskin [7] ensuring that the interaction between the fluid and a boundary point is theoretically independent of where this point lies within a fluid mesh cell. Next we present the Fourier representation of the elastic immersed boundaries.

DOI: 10.1002/fld

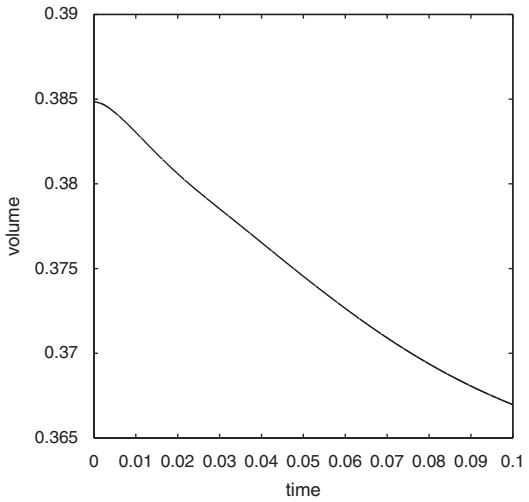

Figure 4. Evolution of the closed immersed area with respect to time in test case B from Section 3.2. $0 \leqslant t \leqslant 0.1$.

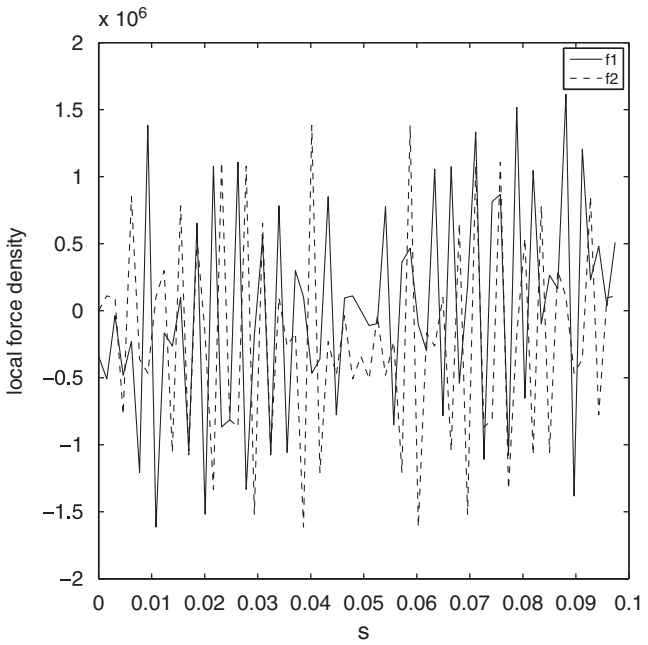

Figure 5. Local elastic force density components $f = [f_1, f_2]$ with respect to the curvilinear coordinate in test case B from Section 3.2 ($0 \leqslant s \leqslant 0.1$ is shown here). $t = 0.1$.

DOI: 10.1002/fld

4. THE FOURIER REPRESENTATION OF THE ELASTIC IMMERSED BOUNDARIES

The use of Fourier expansions is justified by the fact that initially the discretization points on the immersed boundary are equally spaced in the curvilinear space. Then, the motion of these discrete points follows the fluid flow thanks to the no-slip boundary condition. At any time, therefore, the position of the discretization points corresponds to a regular transformation of the original uniform distribution and can be used in the Fourier representation. We now describe the method.

The Fourier representation is formulated here for a 1D closed immersed elastic boundary. However, this could also apply to non-periodic 1D immersed membranes by subtracting a linear combination of the membrane boundary values and using an odd, periodic extension of the original data, with respect to the curvilinear coordinate. The formulation could also be extended to 2D curvilinear membranes with 2D Fourier expansions or spherical harmonics, in a similar manner. Some complex immersed boundaries are actually made of elementary curvilinear squares stitched together. The idea of applying an harmonic analysis to the immersed boundary position vector is quite general and should apply in most cases.

After discretization, the immersed boundary is represented by a finite set of grid points:

$$\{X_i\}_{0\leqslant i\leqslant M-1}=\{X_{1,i}, X_{2,i}\}_{0\leqslant i\leqslant M-1} \tag{25}$$

where $\{X_{1,i}\}_{0\leqslant i\leqslant M-1}$ is the vector of the horizontal components of the moving points and $\{X_{2,i}\}_{0\leqslant i\leqslant M-1}$ is the vector of their vertical components. We assume that M is even and then define $K\equiv M/2$. The Fourier coefficients are:

$$\{\alpha_k\}_{0\leqslant k\leqslant K}=\{\alpha_{1,k}, \alpha_{2,k}\}_{0\leqslant k\leqslant K}$$

with

$$\alpha_{j,k}=\{\alpha^A_{j,k}, \alpha^B_{j,k}\}, \quad 0\leqslant k\leqslant K, \quad j=1,2$$

For $0\leqslant k\leqslant K$, we have:

$$\alpha^A_{j,k}=\sum_{i=0}^{M-1} X_{j,i}\cos\left(2\pi k\frac{i}{M}\right), \quad j=1,2$$

For $1\leqslant k\leqslant K-1$, we have:

$$\alpha^B_{j,k}=\sum_{i=0}^{M-1} X_{j,i}\sin\left(2\pi k\frac{i}{M}\right), \quad j=1,2$$

and:

$$\alpha^B_{j,0}=\alpha^B_{j,K}=0, \quad j=1,2$$

We note:

$$\tilde{\alpha}=(\alpha^A_{1,0}\ldots\alpha^A_{1,K}\alpha^B_{1,1}\ldots\alpha^B_{1,K-1}\alpha^A_{2,0}\ldots\alpha^A_{2,K}\alpha^B_{2,1}\ldots\alpha^B_{2,K-1}) \tag{26}$$

and

$$\alpha=(\alpha^A_{1,1}\ldots\alpha^A_{1,K-1}\alpha^B_{1,1}\ldots\alpha^B_{1,K-1}\alpha^A_{2,1}\ldots\alpha^A_{2,K-1}\alpha^B_{2,1}\ldots\alpha^B_{2,K-1}) \tag{27}$$

DOI: 10.1002/fld

Then the Fourier expansion of X is, for $j=1,2$ and $0\leqslant i\leqslant M-1$:

$$\hat{X}_{j,i}(\tilde{\alpha})=\frac{1}{M}\left[\alpha^A_{j,0}+2\sum_{k=1}^{K-1}\left(\alpha^A_{j,k}\cos\left(2\pi k\frac{i}{M}\right)+\alpha^B_{j,k}\sin\left(2\pi k\frac{i}{M}\right)\right)+\alpha^A_{j,K}(-1)^i\right] \tag{28}$$

Now that we have an expansion of the position vector X, we can also differentiate twice analytically Equation (28) in order to get an expression for the immersed boundary local force density equation (7), instead of using second-order finite differences. This leads to the following formulae for the discrete local force density given by Equation (7), depending on the Fourier coefficients:

$$f|_{j,i}(\alpha)=-\frac{4\pi^2}{M}\sigma\left[\sum_{k=1}^{K-1}2k^2\left(\alpha^A_{j,k}\cos\left(2\pi k\frac{i}{M}\right)+\alpha^B_{j,k}\sin\left(2\pi k\frac{i}{M}\right)\right)+K^2\alpha^A_{j,K}(-1)^{i-1}\right] \tag{29}$$

with $j=1,2$ and $0\leqslant i\leqslant M-1$. We measured on our test cases that using Equation (29) instead of Equation (7) does not introduce any error on the method, the results being almost identical. The error introduced by the finite differencing of the local force density equation (7) is significantly below the one introduced by a poor discretization of the boundary itself. We actually used in this study a very small spacing between the Lagrangian points, in order to focus on the error of the IBM and eliminate any possibility of boundary discretization deficiency.

We note here that the above calculation of the local force density based on the Fourier coefficients could also be done for other type of force densities, since they all are functions of X. We just proposed to plug in the force formulae the continuous expansion of X based on the Fourier coefficients instead of using the discrete set of boundary points along with finite differences.

Another advantage of having a continuous expansion of the position vector X is that the boundary points used to spread the local force density can be chosen arbitrarily along Γ. Thus, more points can be used in the force spreading process than in the Fourier coefficients evaluation or the boundary motion process. This would also allow us to use discrete Dirac delta functions located only on the grid lines during the force spreading process, which might improve the accuracy order of the Poisson pressure equation due to a better discretization of the source term in the right-hand side of Equation (11). We describe, next, an example of use of the data given by the Fourier coefficient, to improve the volume conservation property of the IBM.

5. A VOLUME CONSERVATION METHOD

The 'numerical porosity' of the IBM is a drawback of this artificial boundary technique. The IBM requires the interpolation of an Eulerian variable into a Lagrangian one, when the no-slip boundary condition between the immersed boundary Γ and its neighboring fluid particles is enforced. This interpolation process introduces an error due to the use of discrete Dirac delta functions as interpolating tool. This error of discretization gives us a small degree of freedom

DOI: 10.1002/fld

to eliminate the 'porosity' of the elastic membrane through a correction step. This consists in minimizing the amplitude of a small change on the position vector of Γ, under the constraint of a fixed specified volume. This means that topologically, the domain must be divided into at least two parts totally separated by the immersed boundary. Let us observe that the immersed boundary does not need to be a closed surface and can be, for example, a membrane splitting the whole domain. As described in [19], most of the existing methods to improve volume conservation use a local change of stencil on the Eulerian grid, or require to compute the normal to the boundary at each of its points to build a constrained local interpolation operator for the boundary velocity. The following method is global and uses the initial volume enclosed by the immersed boundary as a control objective. We represent the position vector of the immersed boundary with a global polynomial. In the 2D 'bubble' test case, it is natural to use a Fourier expansion. For convenience, we are going to define the global volume conservation problem in the case of a closed immersed boundary similar to our benchmark problems above. It is easy to compute analytically the area of the 'bubble' using Equation (28) and Green's theorem. We get the formula:

$$\text{Area}(\alpha) = \frac{4\pi}{M^2} \sum_{k=1}^{K-1} k(\alpha_{1,k}^A \alpha_{2,k}^B - \alpha_{2,k}^A \alpha_{1,k}^B) \tag{30}$$

If $\text{Area}(\alpha) = V_0$ at the initial condition, we want this function $c(\alpha)$ to be null:

$$c(\alpha) = \text{Area}(\alpha) - V_0 = 0$$

As we want the correction on the position vector to be as small as possible, we perform a least-square minimization of the position change, constrained with the area preservation. The function to minimize is:

$$\begin{aligned} F(\tilde{\alpha}) &= \|\{X_i - \hat{X}_i(\tilde{\alpha})\}_{0 \leqslant i \leqslant M-1}\|_2^2 \\ &= \sum_{i=0}^{M-1} ((X_{1,i} - \hat{X}_{1,i}(\tilde{\alpha}))^2 + (X_{2,i} - \hat{X}_{2,i}(\tilde{\alpha}))^2) \end{aligned} \tag{31}$$

As the input variable of the constraint $c(\alpha) = 0$ is α and not $\tilde{\alpha}$, the search space of the minimization is $\mathbb{R}^{4(K-1)}$ and not $\mathbb{R}^{4K}$. We assumed that the coefficients

$$(\alpha_{1,0}^A, \alpha_{1,K}^A, \alpha_{2,0}^A, \alpha_{2,K}^A) = \tilde{\alpha} \backslash \alpha$$

are fixed. These coefficients are not related to the area, but control the global position of the immersed boundary in the domain. Then:

$$F(\alpha) = F(\tilde{\alpha})|_{(\alpha_{1,0}^A, \alpha_{1,K}^A, \alpha_{2,0}^A, \alpha_{2,K}^A)}$$

$$\hat{X}_i(\alpha) = \hat{X}_i(\tilde{\alpha})|_{(\alpha_{1,0}^A, \alpha_{1,K}^A, \alpha_{2,0}^A, \alpha_{2,K}^A)}$$

Finally, we can express the constrained minimization problem as follows:

$$\min_{\alpha / c(\alpha) = 0} F(\alpha) \tag{32}$$

We can show that this optimization problem has a unique solution. For that, we define the Lagrangian $L(\alpha, \lambda)$ with $\alpha \in \mathbb{R}^{4(K-1)}$, $\lambda \in \mathbb{R}$:

$$L(\alpha, \lambda) = F(\alpha) + \lambda c(\alpha) \tag{33}$$

DOI: 10.1002/fld

F and c are both twice continuously differentiable with respect to α. A nice property of the Hessian $\nabla_{\alpha\alpha}F$ is that it is independent of α and diagonal:

$$\nabla_{\alpha\alpha}F = \frac{4}{M} I_{4(K-1)} \tag{34}$$

This is also true for $\nabla_{\alpha\alpha}c$ in this particular test-case with area formula equation (30):

$$\nabla_{\alpha\alpha}c = \frac{4\pi}{M^2}\begin{bmatrix} 0 & & & +B_K \\ & & -B_K & \\ & -B_K & & \\ +B_K & & & 0 \end{bmatrix} \tag{35}$$

$$B_K = \begin{bmatrix} 1 & & & 0 \\ & 2 & & \\ & & \ddots & \\ 0 & & & K-1 \end{bmatrix} \tag{36}$$

Hence, for a given pair (α^*, λ^*) and for all α in the neighborhood of α^*, we have:

$$\alpha^{\mathrm{T}}\nabla_{\alpha\alpha}L(\alpha^*, \lambda^*)\alpha = \alpha^{\mathrm{T}}\nabla_{\alpha\alpha}F\alpha + \lambda^*\alpha^{\mathrm{T}}\nabla_{\alpha\alpha}c\alpha \tag{37}$$

$$= \frac{4}{M}\alpha^{\mathrm{T}}\alpha + \frac{8\pi\lambda^*}{M^2}\sum_{k=1}^{K-1} k(\alpha^A_{1,k}\alpha^B_{2,k} - \alpha^A_{2,k}\alpha^B_{1,k}) \tag{38}$$

$$= \frac{4}{M}\alpha^2 + 2\lambda^*\mathrm{Area}(\alpha) \tag{39}$$

Hence, $\lambda^* > 0$ implies:

$$\alpha^{\mathrm{T}}\nabla_{\alpha\alpha}L(\alpha^*, \lambda^*)\alpha > 0 \tag{40}$$

Whenever we find a pair (α^*, λ^*) such that $\nabla_\alpha F(\alpha^*) = 0$, $c(\alpha^*) = 0$ and $\lambda^* > 0$, the first-order Lagrangian sufficiency condition is satisfied. This implies that α^* is a strict local minimum.

We note here that $\nabla_{\alpha\alpha}F$ has the same form for any geometry of Γ in any dimension, since it simply represents the norm of the correction on X. We also have here an elegant analytical formulation for the constraint in the 'Bubble' test case. The functions $F(\alpha)$ and $c(\alpha)$ are easy to compute as well as $\nabla_\alpha F(\alpha)$, $\nabla_\alpha c(\alpha)$ and the Hessians of F and c, respectively $\nabla_{\alpha\alpha}F$ and $\nabla_{\alpha\alpha}c$, are constant. In the general situation, one may have to use a more complicated numerical method to get the elements of the minimization problem (32) regarding the constraint. In the 'Bubble' test case, we need to solve:

$$\begin{pmatrix} \nabla_\alpha L(\alpha, \lambda) \\ \nabla_\lambda L(\alpha, \lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{41}$$

This is done using the classical Newton–Raphson algorithm, although some more efficient algorithms could be implemented. A good initial solution for the iterative Newton algorithm is the

DOI: 10.1002/fld

α coefficients corresponding to the actual position of the boundary X before the correction. In most of the cases, the volume of the immersed boundary is found to evolve relatively slowly with respect to time. Then it is not required to perform this minimization at every time step.

Compared with the traditional method, the full Fourier expansion allows a compact representation of the interface to be used, without loss of accuracy. We are going to describe the reduction of the Fourier expansion technique, and apply it to the 'Bubble' test case.

6. REDUCTION OF THE FOURIER SPACE

To reduce the computational load of the minimization process, one may want to restrict the search space of the α coefficients to a space of dimension smaller than M, the number of discrete points that support the immersed interface. Because of the high-order accuracy of the Fourier expansion for smooth representations, we work with the first $K/4$ of the Fourier coefficients. This compact representation of the immersed closed boundary has several advantages. First, it filters out the high wave frequency components of the position vector, and removes most of the noise in the force term (see Figure 5). Second, it can also speed up an implicit IBM scheme, by reducing the search space for the Newton algorithm. In [11], we implemented the inexact Newton backtracking method of M. Pernice and H. F. Walker [21] in an implicit IBM scheme, in which the unknown is the position of the moving boundary at the next time step. We used a Krylov method to solve the inexact Newton condition and an associated Jacobian matrix approached by finite differences. Without a Fourier representation of the boundary position, the size of the Newton search space in two space dimensions is $2M$, while it becomes $M/2$ here.

To illustrate the numerical accuracy of this compact representation, we measured the error introduced by a truncated Fourier representation of the interface. We compared the results with a reference solution obtained on a fine mesh. We started the simulation with some smooth arbitrary shape immersed closed interface shown in Figure 6, in order to avoid a simple symmetric elliptic shape. Figure 7 reports the error on the interface representation for that particular configuration, with respect to the number Fourier modes used in the expansion. The parameters for this simulation are: $N=100$, $M=400$, $\Delta t \approx 5.1\times 10^{-8}$, $\rho=1$, $\mu=0.1$. In Figure 7, we plot, using the log scale, the discrete 2-norm error on the horizontal component X_1 of the boundary position after 200 time steps, with respect to the number of modes $K_F \leqslant M/2$ used in the Fourier representation of the interface. We have similar result for the second component X_2. The reference solution here, used to measure the error, is the standard IBM solution obtained with $M=400$, without Fourier representation of the interface. The vertical line corresponds to our cut-off choice, which is to keep one fourth of the Fourier coefficients. We note that going to high-order filtering [22] did not improve significantly the accuracy of the simulation.

Being able to reduce the number of Fourier modes also imply the possibility to use less boundary points with relatively more modes. However, it is important to follow the restriction, as stated originally by C. S. Peskin, which is that two neighboring boundary points are required to be within a distance of $h/2$ at most. This condition is required in the force spreading process only. Using the Fourier representation, the position of the boundary is known continuously and the arbitrary curvilinear position of any spreading point can be computed easily; thus, more points can be used only during this process. Regarding the other steps of the boundary treatment, the force density can be evaluated using the Fourier coefficients, while the accuracy of the immersed boundary motion equation depends on the quality of the Cartesian grid and not on the number of boundary points.

DOI: 10.1002/fld

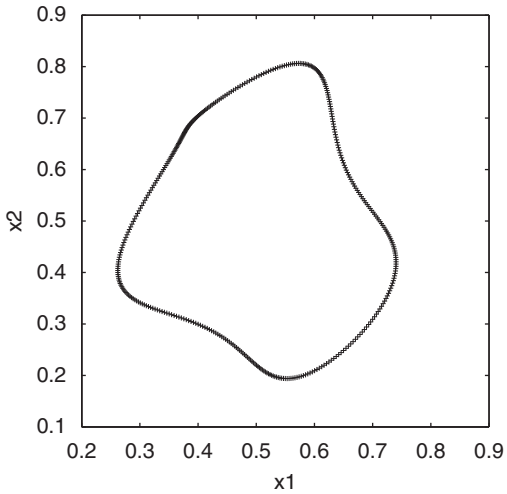

Figure 6. Unsymmetrical initial position of the 'bubble' for the Fourier space reduction study.

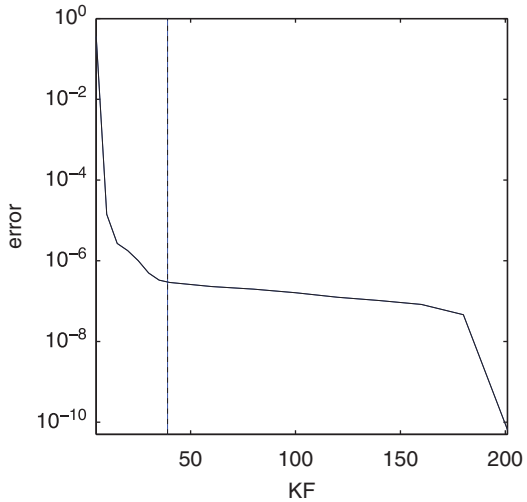

Figure 7. Error introduced by the reduction of the Fourier space with respect to K_F.

We now measure the benefit of the method on both test cases presented in Section 3.

7. RESULTS ON THE TEST CASES A AND B

We now report on the numerical accuracy of the overall method presented in previous sections for the 'bubble' test cases A and B. It is important to check both the volume area of the closed

DOI: 10.1002/fld

immersed boundary and the global energy of the system. In order to evaluate the total energy $E(t)$ of the fluid/structure system at a given time, we use the following formulae from [19]:

$$E(t) = \int_{\Omega} V(x,t)^2 \, \mathrm{d}x - \int_{\Gamma} f(s,t).X(s,t) \, \mathrm{d}s \tag{42}$$

We perform a volume correction every 100 temporal iterations. For each volume correction, most of the minimization is done at the very first Newton step and $c(\alpha) = \nabla_{\lambda} L(\alpha, \lambda)$ only requires two or three steps to become negligible. We used the reduced Fourier expansion, in order to cut off the oscillations.

For test case A, we observe on Figure 8 the very small periodic variations and corrections of the enclosed volume (between 0.125665 and 0.12575). Figure 8 is to be compared with Figure 3, which is the same case without volume correction (between 0.1256 and 0.1316 during the same time length). Figure 9 clearly shows that the system has reached an equilibrium.

For test case B, we observe in Figure 10 a perfect volume conservation. Figure 10 is to be compared with Figure 4, which is the same case without volume correction (between 0.385 and 0.367 during the same time length). Figure 11 clearly shows the improvement regarding the wiggles of the local force density shown in Figure 5. We recall that Figure 5 corresponds exactly to the same case and time as Figure 11, without the Fourier space reduction.

We are now going to present two applications of our IBM implementation that are non-trivial and can take advantage of the Fourier representation of immersed interfaces.

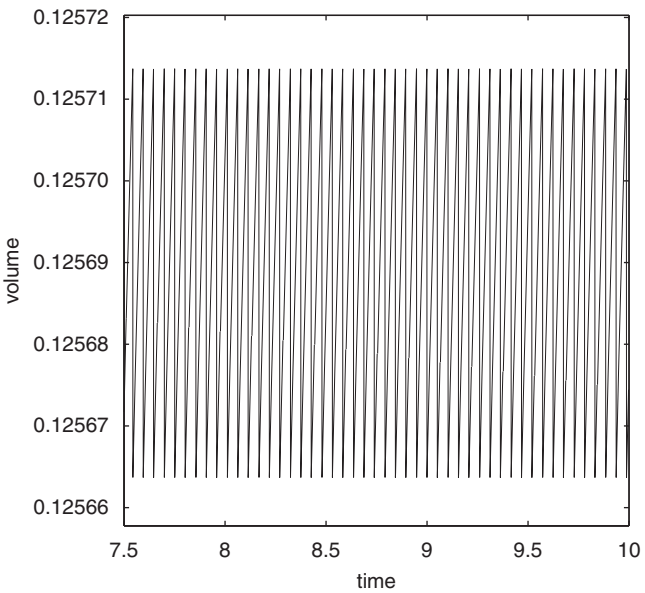

Figure 8. Evolution of the closed immersed area with respect to time in test case A from Section 3.1, with volume correction. $0 \leqslant t \leqslant 10$.

DOI: 10.1002/fld

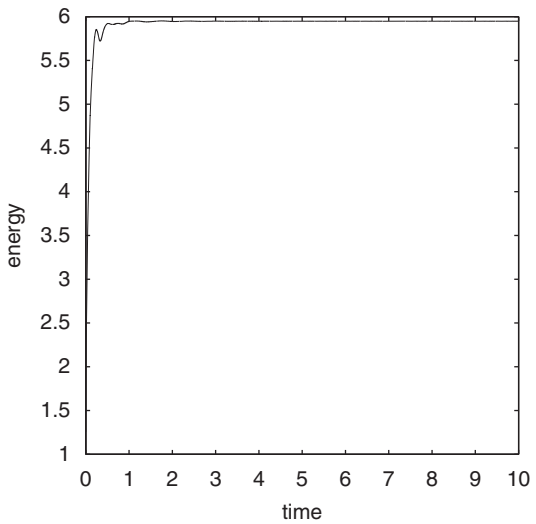

Figure 9. Evolution of the total energy of the fluid/structure system in test case A from Section 3.1. $0 \leqslant t \leqslant 10$.

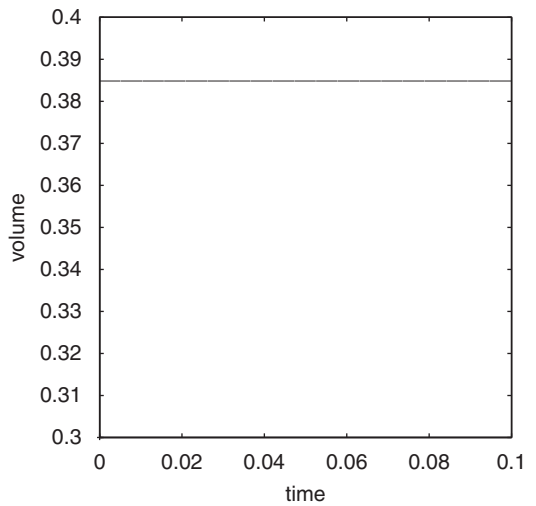

Figure 10. Evolution of the closed immersed area with respect to time in test case B from Section 3.2, with volume correction. $0 \leqslant t \leqslant 0.1$.

8. APPLICATIONS

Both applications deal with small-scale Bio-fluids. This means that the corresponding flows naturally have small Reynolds numbers, being dominated by viscous effects. In our first application we will consider the cavity flow problem with an internal obstacle and a large number of bubbles.

DOI: 10.1002/fld

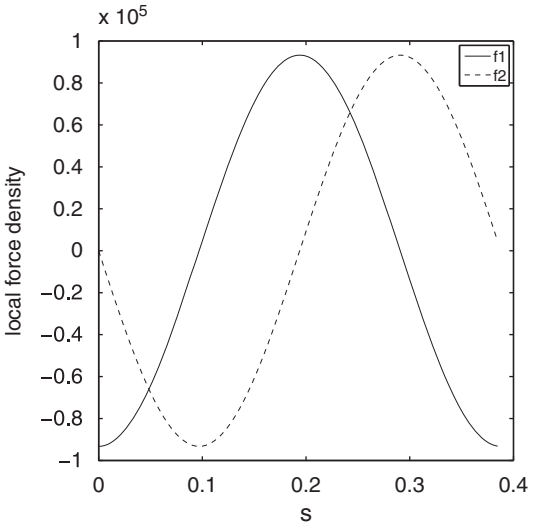

Figure 11. Local elastic force density components $f=[f_1, f_2]$ with respect to the curvilinear coordinate in test case B from Section 3.2, with a reduction of the Fourier space. $t=0.1$.

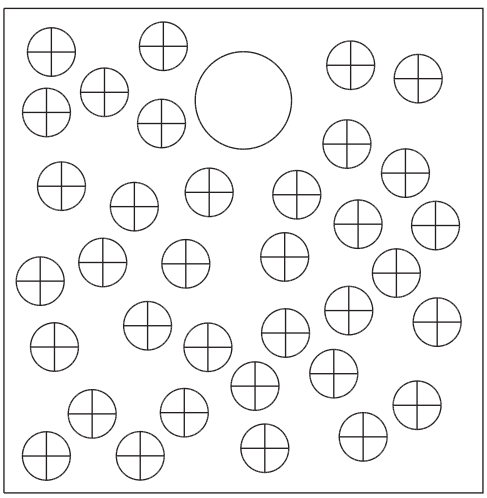

Figure 12. Application 8.1. $t=0$.

8.1. Toward large suspension flow simulations

Blood is a suspension flow. Its primary purpose is transporting cells: the proportion of blood volume that is occupied by red blood cells is normally between 40 and 50%. The dynamic of the flow becomes particularly interesting when the cells are squeezed by some artificial obstacles. To mimic this problem, we consider the following artificial setup that is an extension of our previous

DOI: 10.1002/fld

cavity flow problems. We also have introduced a fixed round shape obstacle close to the sliding wall, at a distance that is lower than the diameter of the elastic bubbles at rest.

To be more specific, we have $\Omega=[0,1]^2$ and a staggered mesh for the pressure of 481×481 points. The physical parameters are $\mu=\rho=1, \sigma=100$. The upper wall velocity is $u(x,1)=40x(1-x)$. Initially, the fluid is at rest. We have included 35 bubbles of radius 0.05 and use $M=240$ discrete points for each bubble's interface. The volume optimization is performed to all the bubbles every $\Delta t=0.005$, while the time step of the time integration scheme is $\mathrm{d}t\simeq 8.68\times10^{-7}$. The fixed obstacle is a cylinder of radius 0.1 and center $(0.5, 0.81)$. We use a different immersed boundary technique to impose the obstacle to the flow: the direct forcing method of J. Mohd-Yusof [23]. To prevent the moving bubbles to get attached to the fixed cylinder, which may happen when both kinds of force terms of the immersed boundary techniques overlap, we found that the obstacle force term in the momentum equation should be spread on the neighboring nodes. Non-physical contact is obviously an imperfection of the overall simulation. We have shown some snapshots of this simulation in Figures 12–14. To better visualize the dynamic of the flow we added the cross inside the bubbles that are not an actual elastic boundary, but visual markers. We believe that our post-processing technique enhances the quality of this simulation, but such complicated test case deserves more extensive validation. Naively, the suspension flow dynamic looks 'real', but we feel that this type of simulation is more likely to be used as an exploratory tool to study suspension flow. We should examine later on the impact of realistic surface tension or contact surface law that are missing here [24, 25].

Using our IBM simulation as an exploratory tool, we have actually (re)discovered a new motion mechanism that does not require flapping fins or flagella [26]. We are going to present this innovative idea in the following section.

8.2. Motion by inertia

Let us consider a single bubble in a long rectangular cavity at rest. The flow velocity at initial time is null and the bubble is a circle. We equip the moving bubble of a membrane that can contract or dilate periodically by forcing the boundary elasticity coefficient in the Hooke law. We set:

$$\sigma(\theta)=\sigma_0\left(1+\sigma_1\left(1+\sin\left(2\pi\frac{t}{P}\right)\right)(\cos(\theta)+1)\right) \tag{43}$$

where $\theta\in(0,2\pi)$ is the angle in the polar moving coordinate system attached to the bubble. To be more specific if X is a point attached to the membrane, the corresponding θ stays invariant with respect to X, no matter X motion. $\theta\in(\pi/2,3\pi/2)$ corresponds to the anterior side of the bubble, and the posterior side is the opposite side of the bubble.

The elasticity coefficient increases and decreases periodically in time with period P. The variation of the elasticity coefficient in time is the most pronounced around $\theta=0$, and has the less variation around $\theta=\pi$. The largest contraction and relaxation move of the membrane happens around $\theta=0$. The forcing of the elasticity coefficient (43) of the membrane makes an 'active bubble'.

During the contraction phase, the posterior side of the membrane projects the liquid inside the bubble toward the forward side. While this mass of liquid travels forward, the posterior side of the membrane relaxes. It results overall in the motion of the bubble forward. We observe from the numerical simulation that this motion is maintained by two nice symmetric vortices that companion the bubble motion. Figures 15 and 16 show an established forward motion of the

DOI: 10.1002/fld

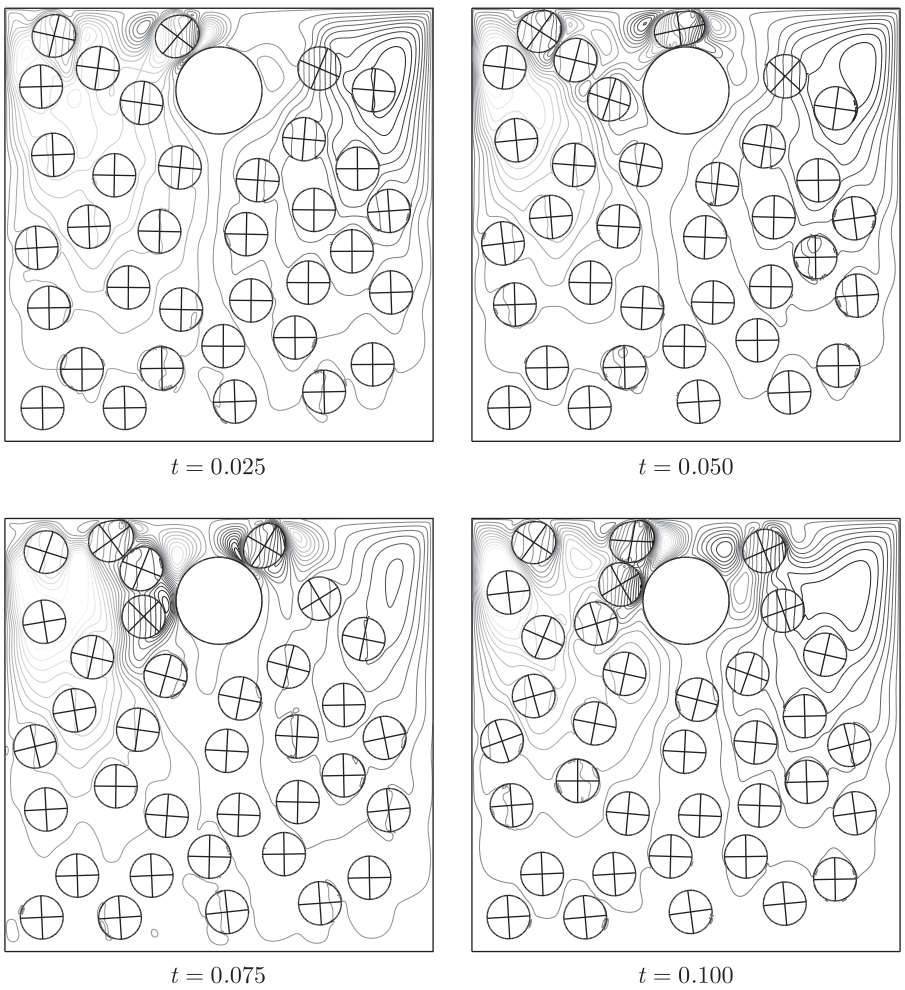

Figure 13. Application 8.1. Temporal evolution of the immersed boundaries and contour of v.

bubble with the bubble starting at rest near the right side of the cavity. In this simulation, we have $(\sigma_0, \sigma_1) = (200, 5)$, and the elasticity coefficient varies in the interval $200 \leqslant \sigma(\theta) \leqslant 4200$. The center of the bubble is at $t = 0$ centered on a symmetry axis of the rectangular cavity. The solution conserves this symmetry as time goes on. However, if the position of the bubble is slightly shifted away of this axis at time $t = 0$, the no-slip boundary flow condition rapidly breaks the symmetry. The direction of motion becomes unstable and the bubble starts to have (slow) chaotic motion. In practice a tail attached at the bubble at the $\theta = 0$ angle position should stabilize this motion thanks to viscosity forces. Alternatively, a tandem of two 'active' bubbles can pilot a larger body. We are currently studying the optimum design of this setup using our simulation tool.

This motion of the 'active bubble' with no flapping fins nor flagella, or helicoidal motion [26] is an amazing example of fluid dynamic that should be explored further. It has some analogy,

DOI: 10.1002/fld

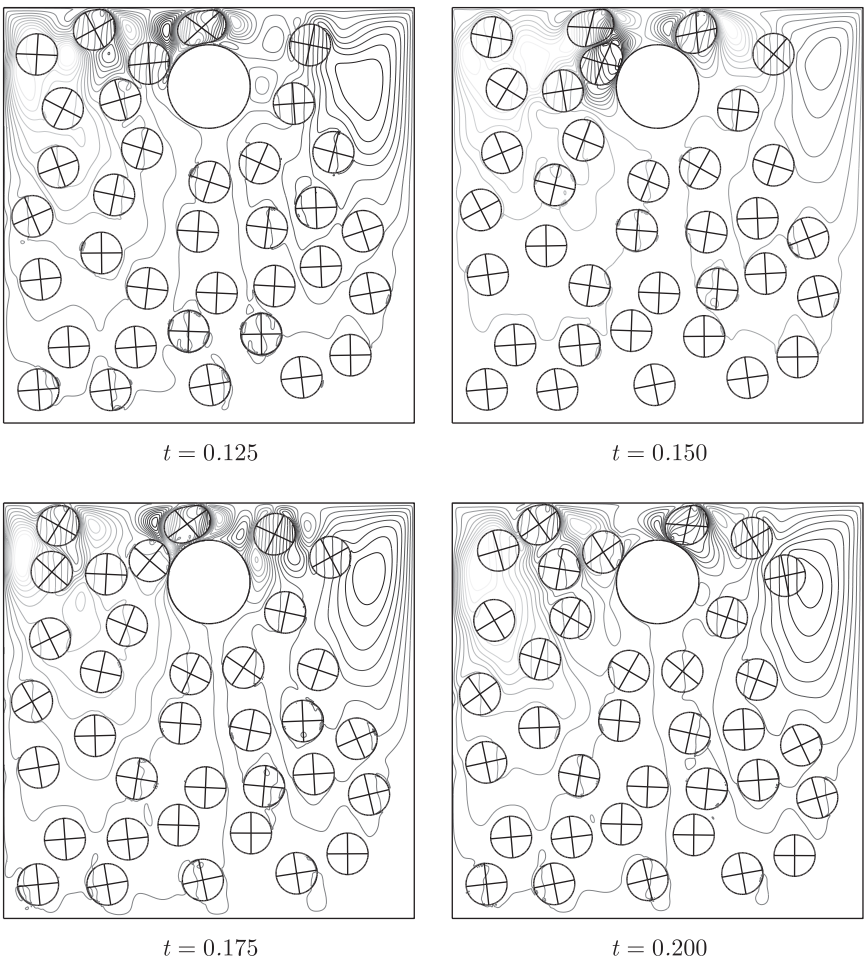

Figure 14. Application 8.1. Temporal evolution of the immersed boundaries and contour of v.

actually, with the motion of Amoeba in the sense that this organism projects its inside fluid toward the desired direction of motion. The internal mechanism of the motion of Amoeba by pseudo-pods is, however, much more complicated and does not seems to be realized by the contraction of an external membrane.

We propose the following conjecture. If one can make artificial membrane from a material with elasticity properties that can be changed by an external magnetic field, for example, we speculate that the mechanism of motion that we have presented here can be controlled from a distance. One may think of a drug encapsulated in a spherical membrane at microscale where the motion by inertia presented here can be controlled by an MRI machine. This might be an interesting drug delivery mechanical system.

We can now summarize our study that actually started from our motivation to improve the numerical efficiency of our IBM implementation.

DOI: 10.1002/fld

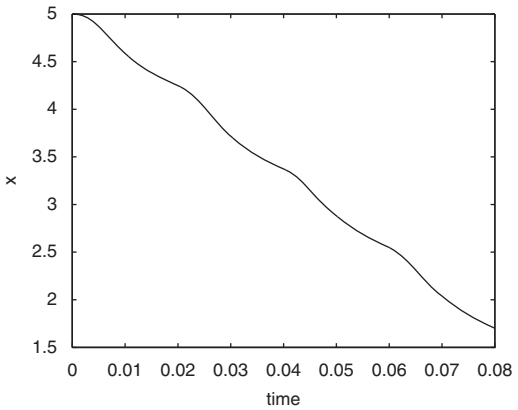

Figure 15. Application 8.2. Motion of the center of the active bubble.

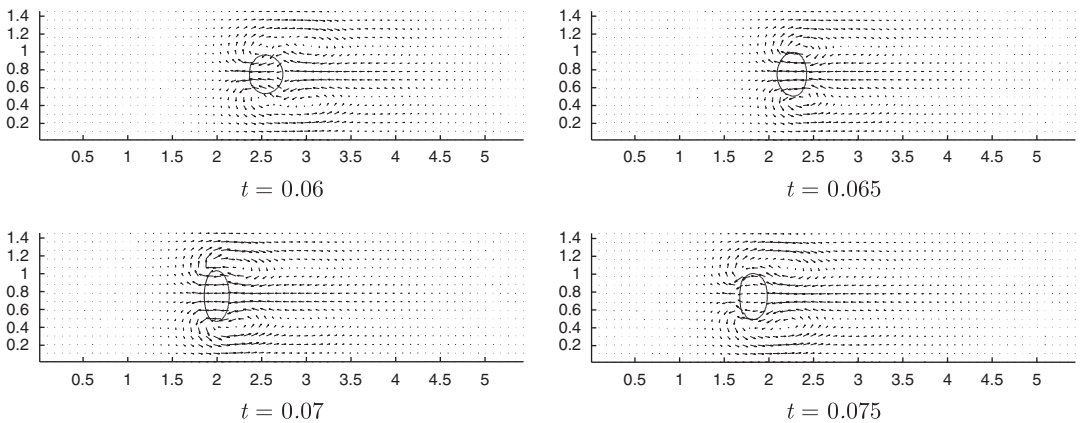

Figure 16. Application 8.2. Active bubble motion with time period 0.02.

9. CONCLUSION

We introduced a new Fourier representation of the discretized immersed boundary coordinates vector, which allows us to gather and control some important information about piecewise smooth elastic immersed bodies. It brings the possibility to easily filter non-physical high-frequency oscillations along the boundaries, to use less boundary points and choose their spreading position, to compute analytically the local force density, and also to perform the volume optimization of closed boundaries. The latter volume conservation method is computationally inexpensive and does not modify the interpolation stencil, but slightly corrects the position of Γ under a fixed specified volume constraint. We checked with some classic test cases that the Fourier representation does not introduce any error when all the modes are kept. In the case of reduced Fourier expansions,

DOI: 10.1002/fld

we studied the number of Fourier modes to keep in order to avoid introducing an non-negligible error. We concluded this study with two non-trivial applications of complex fluid/elastic–structure interactions.

We found particularly fascinating that the IBM technique that is relatively simple to implement gives access to such complex fluid flow problems. This is probably a very strong motivation to develop further the numerical analysis of this technique.

REFERENCES

1. Glowinski R. A fictitious domain approach to the direct numerical simulation of incompressible flow past moving rigid bodies: application to particle flow. *Journal of Computational Physics* 2001; **162**:363–426.
2. Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Review* 2003; **56**(3):331–347.
3. Lee L, LeVeque R. Immersed interface methods for incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing* 2003; **25**:832–856.
4. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
5. Peskin CS. Flow patterns around heart valves: a digital computer method for solving the equations of motion. *Ph.D. Thesis*, Albert Einstein College of Medicine, 1972.
6. Shyy W. Moving boundaries in micro-scale biofluid dynamics. *Applied Mechanics Review* 2001; **54-5**:405–453.
7. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
8. Garbey M, Pacull F. Toward a MatlabMPI parallelized immersed boundary method. In *Parallel Computational Fluid Dynamics*, Deane A, Brenner G, Ecer A, Tromeur-Dervout D, Emerson D, McDonough J, Periaux J, Satofuka N (eds), 2005.
9. Garbey M, Pacull F. A versatile incompressible Navier–Stokes solver for blood flow application. *International Journal for Numerical Methods in Fluids* 2007; **54**(5):473–496.
10. Rosar ME, Peskin CS. Fluid flow in collapsible elastic tubes: a three-dimensional numerical model. *New York Journal of Mathematics* 2001; **7**:281–302.
11. Pacull F. A numerical study of the immersed boundary method and application to blood flow. *Ph.D. Thesis*, Department of Mathematics, University of Houston, 2006.
12. Armfield S, Street R. The pressure accuracy of fractional-step methods for the Navier–Stokes equations on Staggered Grids. *ANZIAM Journal* 2003; **44**:C20–C39.
13. Harlow FH, Welch E. Numerical calculation of time-dependent viscous incompressible flow of fluids with free surface. *Physics of Fluids* 1965; **8**:2182–2189.
14. Morinishi Y, Lund TS, Vasilyev OV, Moin P. Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics* 1998; **143**:90–124.
15. Roma AM. A multilevel self-adaptative version of the immersed boundary method. *Ph.D. Dissertation*, New York University, 1996.
16. Tornberg AK, Engquist B. Numerical approximation of singular source in differential equations. *Journal of Computational Physics* 2004; **200**:462–488.
17. Garbey M, Tromeur Dervout D. On some Aitken like acceleration of the Schwarz method. *International Journal for Numerical Methods in Fluids* 2002; **40**(12):1493–1513.
18. Stockie JM. Analysis and computation of immersed boundaries, with applications to pulp fibers. *Ph.D. Thesis*, University of British Columbia, 1997.
19. Newren EP. Enhancing the immersed boundary method stability, volume conservation, and implicit solvers. *Ph.D. Thesis*, Department of Mathematics, University of Utah, 2007.
20. Peyret R, Taylor TT. *Computational Methods for Fluid Flow*. Springer Series in Computational Physics. Springer: Berlin, 1985.
21. Pernice M, Walker HF. NITSOL: a Newton iterative solver for nonlinear systems. *SIAM Journal on Scientific Computing* 1998; **19**(1):302–318.
22. Gottlieb D, Shu C-W. On the Gibbs phenomenon and its resolution. *SIAM Review* 1997; **39**(4):644–668.
23. Mohd-Yusof J. Combined immersed boundaries/b-splines methods for simulations of flows in complex geometries. *Ctr Annual Research Briefs*, Stanford University, NASA Ames, 1997.

DOI: 10.1002/fld

24. Francois M, Shyy W. Computations of drop dynamics with the immersed boundary method, part 1: numerical algorithm and buoyancy-induced effect. *Numerical Heat Transfer B* 2003; **44**:101–118.
25. Francois M, Shyy W. Computations of drop dynamics with the immersed boundary method, part 2: drop impact and heat transfer. *Numerical Heat Transfer B* 2003; **44**:119–143.
26. Cortez R, Cowen N, Dillon R, Fauci L. Simulation of swimming organisms: coupling internal mechanics with external fluid dynamics. *Computing in Science and Engineering* 2004; **6**(3):38–45.

DOI: 10.1002/fld