Chin. Ann. Math. Ser. B
39(2), 2018, 213–232
DOI: 10.1007/s11401-018-1061-9

Chinese Annals of Mathematics, Series B

An Energy Stable Monolithic Eulerian Fluid-Structure Numerical Scheme

Olivier PIRONNEAU[1]

(*Dedicated to Philippe G. Ciarlet on the occasion of his* 80*th birthday*)

Abstract The conservation laws of continuum mechanics, written in an Eulerian frame, do not distinguish fluids and solids, except in the expression of the stress tensors, usually with Newton's hypothesis for the fluids and Helmholtz potentials of energy for hyperelastic solids. By taking the velocities as unknown monolithic methods for fluid structure interactions (FSI for short) are built. In this paper such a formulation is analysed when the solid is compressible and the fluid is incompressible. The idea is not new but the progress of mesh generators and numerical schemes like the Characteristics-Galerkin method render this approach feasible and reasonably robust. In this paper the method and its discretisation are presented, stability is discussed through an energy estimate. A numerical section discusses implementation issues and presents a few simple tests.

Keywords Fluid-Structure interactions, Numerical method, Energy stability, Finite element method
2000 MR Subject Classification 65M60, 74F10, 74S30, 76D05, 76M25

1 Introduction

Currently two methods dominate FSI (Fluid-Structure-Interaction) science: Arbitrary Lagrangian Eulerian (ALE for short) methods especially for thin structures (see [17, 28]) and immersed boundary methods (IBM for short) (see [11, 29]), for which the mathematical analysis is more advanced (see [5]) but the numerical implementations lag behind. ALE for large displacements have meshing difficulties (see [25]) and to a lesser extent with the matching conditions at the fluid-solid interface (see [23]). Furthermore, iterative solvers for ALE-based FSI methods which rely on alternative solutions of the fluid and the structure parts are subject to the added mass effect and require special solvers (see [7, 16]).

Alternatives to ALE and IBM are few. One old method (see [2–3]) has resurfaced recently, the so-called actualized Lagrangian methods for computing structures (see [22, 26]), see also [10] although different from the present study because it deals mostly with membranes.

Continuum mechanics does not distinguish between solids and fluids till it comes to the constitutive equations. This has been exploited numerically in several studies but most often in the context of ALE (see [21, 24, 32]).

Manuscript received May 8, 2017. Revised August 8, 2017.
[1]Sorbonne Université, UPMC (Paris VI) Laboratoire Jacques-Louis Lions Place Jussieu, Boite 187, Paris 75252, France. E-mail: Olivier.Pironneau@upmc.fr

In the present study, which is a follow-up of [30] and [19], we investigate what Stephan Turek [21], Heil [20] and Wang [34] called a monolithic formulation but here in an Eulerian framework, as in [13–15, 31], following the displaced geometry of the fluid and the solid. In [13], the authors obtained excellent results with the fully Eulerian formulation adopted here but at the cost of meshing difficulties to handle the Lagrangian derivatives. Here we advocate the Characteristic-Galerkin method and obtain an energy estimate, which is not a proof of stability but a prerequisite for it.

2 Conservation Laws

Let the time dependent computational domain Ω_t be made of a fluid region Ω_t^f and a solid region Ω_t^s with no overlap: $\overline{\Omega}_t = \overline{\Omega}_t^f \cup \overline{\Omega}_t^s$, $\Omega_t^f \cap \Omega_t^s = \emptyset$ at any times $t \in (0,T)$. At initial time Ω_0^f and Ω_0^s are prescribed.

Let the fluid-structure interface be $\Sigma_t = \overline{\Omega}_t^f \cap \overline{\Omega}_t^s$ and the boundary of Ω_t be $\partial\Omega_t$. The part of $\partial\Omega_t$ on which either the structure is clamped or on which there is a no slip condition on the fluid, that part is denoted by Γ and assumed to be independent of time.

The following standard notations are used. For more details see one of textbooks: [1–2, 9, 27], or the following articles: [21, 24]. For clarity we use bold characters for vectors and tensors/matrices, with some exceptions, like $x, x^0 \in \mathbb{R}^d,\ d = 2$ or 3.

- $\mathbf{X} : \Omega_0 \times (0,T) \mapsto \Omega_t$: $\mathbf{X}(x^0,t)$, the Lagrangian position at t of x^0.
- $\mathbf{u} = \partial_t \mathbf{X}$, the velocity of the deformation.
- $\mathbf{F} = \nabla^{\mathrm{T}}\mathbf{X} = ((\partial_{x_i^0}\mathbf{X}_j))$, the Jacobian of the deformation.
- $J = \det_{\mathbf{F}}$.

We denote by tr_A and $\det_A$ the trace and determinant of A. To describe the fluid structure system we need the following:

- $\rho = \mathbf{1}_{\Omega_t^f}\rho^f + \mathbf{1}_{\Omega_t^s}\rho^s$, the density.
- $\sigma = \mathbf{1}_{\Omega_t^f}\sigma^f + \mathbf{1}_{\Omega_t^s}\sigma^s$, the stress tensor.
- $\mathbf{f}(x,t)$ the density of volumic forces at x,t.
- $\mathbf{d} = \mathbf{X}(x^0,t) - x^0$, the displacement.

Finally and unless specified all spatial derivatives are with respect to $x \in \Omega_t$ and not with respect to $x^0 \in \Omega_0$. Let ϕ a function of x,t; as $x = \mathbf{X}(x^0,t),\ x^0 \in \Omega_0$, ϕ is also a function of x^0 and we have

$$\nabla_{x^0}\phi = [\partial_{x_i^0}\phi] = [\partial_{x_i^0}\mathbf{X}_j\partial_{x_j}\phi] = \mathbf{F}^{\mathrm{T}}\nabla\phi.$$

When $\mathbf{X}$ is one-to-one and invertible, $\mathbf{d}$ and $\mathbf{F}$ can be seen as functions of (x,t) instead of (x^0,t). They are related by

$$\mathbf{F}^{\mathrm{T}} = \nabla_{x^0}\mathbf{X} = \nabla_{x^0}(\mathbf{d}+x^0) = \nabla_{x^0}\mathbf{d} + \mathbf{I} = \mathbf{F}^{\mathrm{T}}\nabla\mathbf{d} + \mathbf{I} \ \Rightarrow\ \mathbf{F} = (\mathbf{I} - \nabla\mathbf{d})^{-\mathrm{T}}$$

Time derivatives are related by (note the notation $\mathbb{D}_t$)

$$\mathbb{D}_t\phi := \frac{\mathrm{d}}{\mathrm{d}t}\phi(\mathbf{X}(x_0,t),t)_{|x=\mathbf{X}(x_0,t)} = \partial_t\phi(x,t) + \mathbf{u}\cdot\nabla\phi(x,t).$$

It is convenient to introduce (note the difference between $\mathbb{D}_t$ above and D here):

$$\mathrm{D}\mathbf{u} = \nabla\mathbf{u} + \nabla^{\mathrm{T}}\mathbf{u}.$$

Conservation of momentum and conservation of mass take the same form for the fluid and the solid:

$$\rho\mathbb{D}_t\mathbf{u} = \mathbf{f} + \nabla\cdot\sigma, \quad \mathbb{D}_t\rho + \rho\nabla\cdot\mathbf{u} = \mathbb{D}_t(J\rho) = 0.$$

So $J\rho = \rho_0$ at all times

$$J^{-1}\rho_0\mathbb{D}_t\mathbf{u} = f + \nabla\cdot\sigma \quad \text{in } \Omega_t, \quad \forall t\in(0,T) \tag{2.1}$$

with continuity of $\mathbf{u}$ and of $\sigma\cdot\mathbf{n}$ at the fluid-structure interface Σ in absence of interface constraint like surface tension. There are also unwritten constraints pertaining to the realisability of the map $\mathbf{X}$ (see [9, 27]).

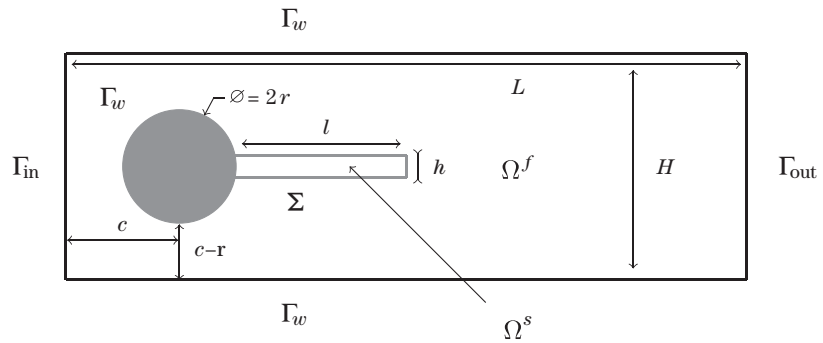

Figure 1 The geometry of the FLUSTRUK test (see [15]). The cylinder (in black) is fixed but the flag is a thick compressible Mooney-Rivlin material clamped to the cylinder by its left boundary; the outer rectangle is filled with a fluid which enters from the left Γ_{in} and leaves on the right Γ_{out}; the horizontal boundaries of the outer rectangle are walls, so they form together with the cylinder the boundary Γ_w. The flag is at time zero a rectangle of size $l\times h$. The outer rectangle has size $L\times H$. The center of the circle representing the cylinder is at (c,c) in a frame of reference which has the lower left corner at $(0,0)$; the cylinder has radius r and is fixed.

2.1 Constitutive equations

We consider a bi-dimensional geometry. For the 3d case (see [8]).

- For a Newtonian incompressible fluid : $\sigma^f = -p^f\mathbf{I} + \mu^f\mathrm{D}\mathbf{u}$,
- For an hyperelastic material : $\sigma^s = \rho^s\partial_{\mathbf{F}}\Psi\mathbf{F}^{\mathrm{T}}$,

where Ψ is the Helmholtz potential which, in the case of a S^t-Venant-Kirchhoff material, is (see [9])

$$\Psi(\mathbf{F}) = \frac{\lambda^s}{2}\mathrm{tr}_{\mathbf{E}}^2 + \mu^s \mathrm{tr}_{\mathbf{E}^2}, \quad \mathbf{E} = \frac{1}{2}(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}). \tag{2.2}$$

It is easy to see that $\mathrm{tr}_{\mathbf{E}} = \frac{1}{2}\mathrm{tr}_{\mathbf{F}^{\mathrm{T}}\mathbf{F}} - 1$ and

$$\begin{aligned}
\partial_{\mathbf{F}}\mathrm{tr}_{\mathbf{F}^{\mathrm{T}}\mathbf{F}} &= \Big(\Big(\partial_{\mathbf{F}_{ij}} \sum_{m,n} F_{m,n}^2\Big)\Big) = 2\mathbf{F} \Rightarrow \partial_{\mathbf{F}}\mathrm{tr}_{\mathbf{E}} = \mathbf{F}, \\
\partial_{\mathbf{F}}\mathrm{tr}_{(\mathbf{F}^{\mathrm{T}}\mathbf{F})^2} &= \Big(\Big(\partial_{\mathbf{F}_{ij}} \sum_{n,m,p,k} F_{n,k}F_{n,m}F_{p,m}F_{p,k}\Big)\Big) = 4\mathbf{F}\mathbf{F}^{\mathrm{T}}\mathbf{F},
\end{aligned} \tag{2.3}$$

which implies that $\partial_{\mathbf{F}}\mathrm{tr}_{\mathbf{E}^2} = 2\mathbf{F}\mathbf{E}$. Therefore

$$\partial_{\mathbf{F}}\Psi(\mathbf{F})\mathbf{F}^{\mathrm{T}} = (\lambda^s \mathrm{tr}_{\mathbf{E}}\mathbf{F} + 2\mu^s \mathbf{F}\mathbf{E})\mathbf{F}^{\mathrm{T}},$$

which in turn implies that

$$\sigma^s = \rho^s \mathbf{F}(\lambda^s \mathrm{tr}_{\mathbf{E}} + 2\mu^s \mathbf{E})\mathbf{F}^{\mathrm{T}} = J^{-1}\rho_0^s \mathbf{F}(\lambda^s \mathrm{tr}_{\mathbf{E}} + 2\mu^s \mathbf{E})\mathbf{F}^{\mathrm{T}}.$$

For a tensor $\mathbf{A}$ define $|\mathbf{A}| = \sum_{ij} A_{ij}^2$.

Remark 2.1 Some authors have a different definition for the Lamé coefficient $\lambda\rho_0^s \to \lambda$, $\mu\rho_0^s \to \mu$ which define σ^s.

Proposition 2.1 *Let* $\gamma = \mathrm{tr}_{\mathbf{F}\mathbf{F}^{\mathrm{T}}}$. *Then*

$$\gamma = \mathrm{tr}_{\mathbf{F}\mathbf{F}^{\mathrm{T}}} = (2 - 2\nabla\cdot\mathbf{d} + |\nabla\mathbf{d}|^2)J^2, \quad \widetilde{\gamma} = \gamma J^{-2}$$

and the following holds

$$\sigma^s = \rho^s(a\mathbf{I} + 2b(\mathrm{D}\mathbf{d} - \nabla\mathbf{d}\nabla^{\mathrm{T}}\mathbf{d}))$$

with

$$\begin{aligned}
a &= \lambda^s\Big(\frac{1}{2}\gamma - 1\Big)(\widetilde{\gamma} - 1) + \mu^s(\gamma - J^2 - 1)\widetilde{\gamma}, \\
b &= \frac{1}{2}\Big(\frac{\lambda^s}{2} + \mu^s\Big)(\gamma - 1) - \frac{\lambda^s}{4}.
\end{aligned} \tag{2.4}$$

Proof First note that if $\mathbf{B} = \mathbf{F}\mathbf{F}^{\mathrm{T}}$ then

$$\sigma^s = \rho^s\Big[\Big[\lambda^s\Big(\frac{1}{2}\gamma - 1\Big) - \mu^s\Big]\mathbf{B} + \mu^s\mathbf{B}^2\Big]. \tag{2.5}$$

Now by the Cayley- Hamilton theorem in 2-dimensions, $\mathbf{B}^2 - \gamma\mathbf{B} + J^2\mathbf{I} = 0$. As

$$\mathbf{B}^{-1} = \mathbf{I} - \mathrm{D}\mathbf{d} + \nabla\mathbf{d}\nabla^{\mathrm{T}}\mathbf{d},$$

let $\mathbf{C} = \mathbf{I} - \mathbf{B}^{-1} = \mathrm{D}\mathbf{d} - \nabla\mathbf{d}\nabla^{\mathrm{T}}\mathbf{d}$. Then

$$\mathbf{B} = \gamma\mathbf{I} - J^2\mathbf{B}^{-1} = (\gamma - J^2)\mathbf{I} + J^2\mathbf{C}, \quad \mathbf{B}^2 = (\gamma^2 - (1+\gamma)J^2)\mathbf{I} + \gamma J^2\mathbf{C}. \tag{2.6}$$

Therefore

$$\begin{aligned}\sigma^s &= \rho^s\Big[\Big[\lambda^s\Big(\frac{1}{2}\gamma-1\Big)-\mu^s\Big][(\gamma-J^2)\mathbf{I}+J^2\mathbf{C}]+\mu^s[(\gamma^2-(1+\gamma)J^2)\mathbf{I}+\gamma J^2\mathbf{C}]\Big]\\ &= \rho^s\Big[\Big[\Big(\lambda^s\Big(\frac{1}{2}\gamma-1\Big)\Big)(\gamma-J^2)+\mu^s\gamma(\gamma-1-J^2)\Big]\mathbf{I}\\ &\quad+\Big[\lambda^s\Big(\frac{1}{2}\gamma-1\Big)+\mu^s(\gamma-1)\Big]J^2\mathbf{C}\Big].\end{aligned} \tag{2.7}$$

2.2 Variational monolithic Eulerian formulation

From now on we limit our analysis to the case $\rho_0^s,\ \rho_0^f$ constant.

One must find $(\mathbf{u},p)$ with $\mathbf{u}_{|\Gamma}=0$, $\mathbf{d}$ and Ω_t^s, Ω^f, solution for all $(\widehat{\mathbf{u}},\widehat{p})$ with $\widehat{\mathbf{u}}_{|\Gamma}=0$ of

$$\begin{cases}\displaystyle\int_{\Omega_t^f}\Big[\rho^f\mathbb{D}_t\mathbf{u}\cdot\widehat{\mathbf{u}}-p\nabla\cdot\widehat{\mathbf{u}}-\widehat{p}\nabla\cdot\mathbf{u}+\frac{\mu^f}{2}\mathrm{D}\mathbf{u}:\mathrm{D}\widehat{\mathbf{u}}\Big]\\ \displaystyle+\int_{\Omega_t^s}\rho^s[\mathbb{D}_t\mathbf{u}\cdot\widehat{\mathbf{u}}+b(\mathrm{D}\mathbf{d}-\nabla\mathbf{d}\nabla^{\mathrm{T}}\mathbf{d}):\mathrm{D}\widehat{\mathbf{u}}+a\nabla\cdot\widehat{\mathbf{u}}]=\int_{\Omega_t}f\cdot\widehat{\mathbf{u}},\\ \mathbb{D}_t\mathbf{d}=\mathbf{u},\quad J^{-1}=\det_{\mathbf{I}-\nabla\mathbf{d}},\quad \rho^r=J^{-1}\rho_0^r,\\ \{\dot{x}(t)=\mathbf{u}(x(t),t),\ x(0)=x_0\in\Omega_0^r\Rightarrow x(t)\in\Omega_t^r\},\quad r=s,f.\end{cases} \tag{2.8}$$

For an existence result, up to time T^* (see [6, 12, 33]), provided a regularization term is added to the formulation to insure that $\partial_t\mathbf{d}$ has H^1-regularity; T^* is such that the solid does not touch the boundary and Σ_t does not buckle.

3 Numerical Scheme

For the stability of the numerical scheme, the problem is that even for small displacements the Lamé terms $\mu^s\nabla\mathbf{u}:\nabla\widehat{\mathbf{u}}+\lambda^s\nabla\cdot\mathbf{u}\nabla\cdot\widehat{\mathbf{u}}$ are hidden in $b\mathrm{D}\mathbf{d}:\mathrm{D}\widehat{\mathbf{u}}$ and $a\nabla\cdot\widehat{\mathbf{u}}$ in the above variational formulation (2.8).

But notice that

$$\begin{aligned}&J^2=1+2\nabla\cdot\mathbf{d}-2\det_{\nabla\mathbf{d}}+3(\nabla\cdot\mathbf{d})^2+o(|\nabla\mathbf{d}|^2),\\ &\gamma=2\Big(1+\nabla\cdot\mathbf{d}+(\nabla\cdot\mathbf{d})^2+\frac{1}{2}|\nabla\mathbf{d}|^2-2\det\nabla\mathbf{d}\Big)+o(|\nabla\mathbf{d}|^2),\\ &\Big(\frac{\gamma}{2}-1\Big)(\widetilde{\gamma}-1)=\nabla\cdot\mathbf{d}-(\nabla\cdot\mathbf{d})^2-\frac{1}{2}|\nabla\mathbf{d}|^2-2\det\nabla\mathbf{d}+o(|\nabla\mathbf{d}|^2).\end{aligned} \tag{3.1}$$

So it makes sense to define

$$c=a-\lambda^s\nabla\cdot\mathbf{d}. \tag{3.2}$$

To prepare the time discretisation of (2.8) with a given time step δt, let

$$\overline{\mathbf{d}}=\mathbf{d}-\delta t\mathbf{u}. \tag{3.3}$$

Then (2.8) becomes

$$\begin{cases} \int_{\Omega_t^f} \left[\rho \mathbb{D}_t \mathbf{u}\cdot\widehat{\mathbf{u}} - p\nabla\cdot\widehat{\mathbf{u}} - \widehat{p}\nabla\cdot\mathbf{u} + \frac{\mu^f}{2}\mathrm{D}\mathbf{u} : \mathrm{D}\widehat{\mathbf{u}}\right] \\ + \int_{\Omega_t^s} \rho\delta t[b(\mathrm{D}\mathbf{u} - \nabla\overline{\mathbf{d}}\nabla^{\mathrm{T}}\mathbf{u} - \nabla\mathbf{u}\nabla^{\mathrm{T}}\overline{\mathbf{d}} + \delta t\nabla\mathbf{u}\nabla^{\mathrm{T}}\mathbf{u}) : \mathrm{D}\widehat{\mathbf{u}} + \lambda^s\nabla\cdot\mathbf{u}\ \nabla\cdot\widehat{\mathbf{u}}] \\ + \int_{\Omega_t^s} \rho[\mathbb{D}_t\mathbf{u}\cdot\widehat{\mathbf{u}} + b(\mathrm{D}\overline{\mathbf{d}} - \nabla\overline{\mathbf{d}}\nabla^{\mathrm{T}}\overline{\mathbf{d}}) : \mathrm{D}\widehat{\mathbf{u}} + (c+\lambda^s\nabla\cdot\overline{\mathbf{d}})\nabla\cdot\widehat{\mathbf{u}}] = \int_{\Omega_t} f\cdot\widehat{\mathbf{u}}, \\ \mathbb{D}_t\mathbf{d} = \mathbf{u}, \quad \rho = \rho_0 \mathrm{det}_{\mathbf{I}-\nabla\mathbf{d}}. \end{cases}$$

Here linear elasticity is visible because the zero order term of b is $\frac{\mu^f}{2}$. From now on we do not use $\overline{\mathbf{d}}$ because the Characteristics-Galerkin discretisation of $\mathbb{D}_t\mathbf{d} = \mathbf{u}$ will give an analogue of (3.3).

3.1 Discretisation of total derivatives

Let $\Omega \subset \mathbb{R}^d$, $\mathbf{u} \in \mathbf{H}_0^1(\Omega) = (H_0^1(\Omega))^d$ ($d = 2$ here), $t \in (0,T)$ and $x \in \Omega$. Then let $\chi^t_{\mathbf{u},x}(\tau)$ be the solution at time τ of

$$\dot{\chi}(\tau) = \mathbf{u}(\chi(\tau),\tau) \quad \text{with } \chi(t) = x.$$

If $\mathbf{u}$ is Lipschitz in space and continuous in time the solution exists. The Characteristics-Galerkin method relies on the concept of total derivative:

$$\mathbb{D}_t\mathbf{v}(x,t) := \frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{v}(\chi(\tau),\tau)|_{\tau=t} = \partial_t\mathbf{v} + \mathbf{u}\cdot\nabla\mathbf{v}.$$

Given a time step δt, let us approximate

$$\chi^{(n+1)\delta t}_{\mathbf{u}^{n+1},x}(n\delta t) \approx \mathbb{Y}^{n+1}(x) := x - \mathbf{u}^{n+1}(x)\delta t.$$

Remark 3.1 Note also that, as $J\rho$ is convected by $\mathbf{u}$, that is $J\rho|_{\chi^t_{\mathbf{u},x}(\tau),\tau} = J\rho|_{x,t}$, so a consistent approximation is

$$(J^n\rho_n)\circ\mathbb{Y}^{n+1}(x) = J^{n+1}(x)\rho_{n+1}(x), \quad x \in \Omega_{n+1}.$$

Thus discretising the total derivative of $\mathbf{u}$ or the one of $\rho_0\mathbf{u}$ will give the same scheme.

$$\begin{aligned} &\rho_0(x)\frac{\mathbf{u}^{n+1}(x) - \mathbf{u}^n(\mathbb{Y}^{n+1}(x))}{\delta t} \\ &= J^{n+1}\rho_{n+1}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n\circ\mathbb{Y}^{n+1}}{\delta t} \\ &= \frac{J^{n+1}\rho_{n+1}\mathbf{u}^{n+1} - (J^n\rho_n\mathbf{u}^n)\circ\mathbb{Y}^{n+1}}{\delta t} \\ &= \frac{\rho_0\mathbf{u}^{n+1} - (\rho_0\mathbf{u}^n)\circ\mathbb{Y}^{n+1}}{\delta t}. \end{aligned} \tag{3.4}$$

3.2 Updating the fluid and solid domain

From the definition of $\mathbb{Y}$, notice that the only way to be consistent is to define Ω_{n+1} using $\mathbf{u}^{n+1}$, i.e., implicitly, since the later is defined also on Ω_{n+1}:

$$\Omega_{n+1} = (\mathbb{Y}^{n+1})^{-1}(\Omega_n) = \{x :\ \mathbb{Y}^{n+1}(x) := x - \mathbf{u}^{n+1}(x)\delta t \in \Omega_n\}.$$

3.3 The time discretised scheme

Let

$$\widetilde{\mathbf{d}}^n := \mathbf{d}^n \circ \mathbb{Y}^{n+1}, \quad \mathbf{d}^{n+1} = \widetilde{\mathbf{d}}^n + \delta t \mathbf{u}^{n+1}, \quad \rho_{n+1} = \rho_0 \det_{\mathbf{I}-\nabla \mathbf{d}^{n+1}}. \tag{3.5}$$

Let $\widetilde{b}_n, \widetilde{c}_n$ be given by (2.4) and (3.2) computed with $\widetilde{d}^n$. The following defines $\mathbf{u}^{n+1}, p^{n+1}$ with $\mathbf{u}^{n+1}|_\Gamma = 0$: $\forall \widehat{\mathbf{u}}, \widehat{p}$ with $\widehat{\mathbf{u}}|_\Gamma = 0$,

$$\begin{cases} \displaystyle\int_{\Omega_{n+1}} \rho_{n+1} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n \circ \mathbb{Y}^{n+1}}{\delta t} \cdot \widehat{\mathbf{u}} \\ \displaystyle + \int_{\Omega^f_{n+1}} \Big[- p^{n+1} \nabla \cdot \widehat{\mathbf{u}} - \widehat{p} \nabla \cdot \mathbf{u}^{n+1} + \frac{\mu^f}{2} \mathrm{D}\mathbf{u}^{n+1} : \mathrm{D}\widehat{\mathbf{u}} \Big] \\ \displaystyle + \int_{\Omega^s_{n+1}} \rho_{n+1} \delta t [\widetilde{b}_n (\mathrm{D}\mathbf{u}^{n+1} - \nabla \widetilde{\mathbf{d}}^n \nabla^{\mathrm{T}} \mathbf{u}^{n+1} - \nabla \mathbf{u}^{n+1} \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n) : \mathrm{D}\widehat{\mathbf{u}} \\ + \lambda^s \nabla \cdot \mathbf{u}^{n+1} \nabla \cdot \widehat{\mathbf{u}}] \\ \displaystyle + \int_{\Omega^s_{n+1}} [\widetilde{b}_n (\mathrm{D}\widetilde{\mathbf{d}}^n - \nabla \widetilde{\mathbf{d}}^n \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n) : \mathrm{D}\widehat{\mathbf{u}} + (\widetilde{c}_n + \lambda^s \nabla \cdot \widetilde{\mathbf{d}}^n) \nabla \cdot \widehat{\mathbf{u}}] \\ \displaystyle = \int_{\Omega_{n+1}} f \cdot \widehat{\mathbf{u}}. \end{cases} \tag{3.6}$$

3.4 Iterative solution by fixed point

The most natural method to solve the above is to freeze some coefficients so as to obtain a well posed linear problem and iterate:

(1) Start with $\mathbf{u} = \mathbf{u}^n$, $\mathbb{Y}(x) = x - \mathbf{u}\delta t$, $\Omega^r = \mathbb{Y}^{-1}(\Omega^r_n)$, $r = s, f$.

(2) Set $\widetilde{\mathbf{d}}^n = \mathbf{d}^n \circ \mathbb{Y}$, $\widetilde{\rho}_n = \rho_0 \det_{\mathbf{I}-\nabla \widetilde{\mathbf{d}}}$; compute $\widetilde{b}_n, \widetilde{c}_n$.

(3) Find $\mathbf{u}^{n+1}, p^{n+1}$ by solving

$$\begin{cases} \displaystyle\int_{\Omega} \widetilde{\rho}_n \frac{\mathbf{u}^{n+1} - \mathbf{u}^n \circ \mathbb{Y}}{\delta t} \cdot \widehat{\mathbf{u}} \\ \displaystyle + \int_{\Omega^f} \Big[- p^{n+1} \nabla \cdot \widehat{\mathbf{u}} - \widehat{p} \nabla \cdot \mathbf{u}^{n+1} + \frac{\mu^f}{2} \mathrm{D}\mathbf{u}^{n+1} : \mathrm{D}\widehat{\mathbf{u}} \Big] \\ \displaystyle + \int_{\Omega^s} \widetilde{\rho}_n \delta t [\widetilde{b}_n (\mathrm{D}\mathbf{u}^{n+1} - \nabla \widetilde{\mathbf{d}}^n \nabla^{\mathrm{T}} \mathbf{u}^{n+1} - \nabla \mathbf{u}^{n+1} \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n) : \mathrm{D}\widehat{\mathbf{u}} \\ + \lambda^s \nabla \cdot \mathbf{u}^{n+1} \nabla \cdot \widehat{\mathbf{u}}] \\ \displaystyle + \int_{\Omega^s} [\widetilde{b}_n (\mathrm{D}\widetilde{\mathbf{d}}^n - \nabla \widetilde{\mathbf{d}}^n \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n) : \mathrm{D}\widehat{\mathbf{u}} + (\widetilde{c}_n + \lambda^s \nabla \cdot \widetilde{\mathbf{d}}^n) \nabla \cdot \widehat{\mathbf{u}}] \\ \displaystyle = \int_{\Omega} f \cdot \widehat{\mathbf{u}}. \end{cases} \tag{3.7}$$

(4) Set $\mathbf{u} = \mathbf{u}^{n+1}$, $\mathbb{Y}(x) = x - \mathbf{u}\delta t$, $\Omega^r = \mathbb{Y}^{-1}(\Omega^r_n)$, $r = s, f$.

(5) If not converged return to Step 2 else set $\mathbf{d}^{n+1} = \mathbf{d}^n \circ \mathbb{Y} + \delta t \mathbf{u}^{n+1}$.

Notice that (3.7) is a well posed linear problem whenever

$$A(\mathbf{u}, \widehat{\mathbf{u}}) = \int_{\Omega^s} \Big[\frac{\rho}{\delta t} \mathbf{u} \cdot \widehat{\mathbf{u}} + \widetilde{b} (\mathrm{D}\mathbf{u} - \nabla \widetilde{\mathbf{d}}^n \nabla^{\mathrm{T}} \mathbf{u} - \nabla \mathbf{u} \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n) : \mathrm{D}\widehat{\mathbf{u}} + \lambda^s \nabla \cdot \mathbf{u} \nabla \cdot \widehat{\mathbf{u}} \Big]$$

is coercive. Then (3.7) gives a solution bounded in $\mathbf{H}^1(\Omega)$ and converging subsequences can be extracted from $\rho_{n+1}, \mathbf{u}^{n+1}, \Omega^r_{n+1}$ when $\overline{\Omega} = \overline{\Omega^f_n \cup \Omega^s_n}$ is fixed. Then convergence would occur if we could prove that Ω^r_{n+1} converges.

3.5 Spatial discretisation with finite elements

Let $\mathcal{T}^0_h$ be a triangulation of the initial domain. Spatial discretisation can be done with the most popular finite element for fluids: The Lagrangian triangular elements of degree 2 for the space V_h of velocities and displacements and Lagrangian triangular elements of degree 1 for the pressure space Q_h; later we will also discuss the stabilised $P^1 - P^1$ element. As the pressure is defined up to a constant, a small penalization term with parameter ϵ must be added to impose uniqueness.

This leads us to find $\mathbf{u}^{n+1}_h \in V_{h0_\Gamma}$, $p^{n+1}_h \in Q_h$, Ω_{n+1} such that for all $\widehat{\mathbf{u}}_h, \widehat{p}_h \in V_{h0_\Gamma} \times Q_h$ with

$$\widetilde{\mathbf{d}}^n_h := \mathbf{d}^n_h \circ \mathbb{Y}^{n+1},$$

where

$$\mathbb{Y}^{n+1}(x) = x - \mathbf{u}^{n+1}_h(x)\delta t,$$

the following holds:

$$\begin{cases} \mathbf{a}(\widetilde{\rho}_n, \widetilde{b}_n, \widetilde{c}_n; \mathbf{u}^{n+1}, \widehat{\mathbf{u}}) := \displaystyle\int_{\Omega_{n+1}} \widetilde{\rho}_n \frac{\mathbf{u}^{n+1}_h - \mathbf{u}^n_h \circ \mathbb{Y}^{n+1}}{\delta t} \cdot \widehat{\mathbf{u}}_h \\ + \displaystyle\int_{\Omega^f_{n+1}} \Big[- p^{n+1} \nabla \cdot \widehat{\mathbf{u}}_h - \widehat{p} \nabla \cdot \mathbf{u}^{n+1}_h + \frac{\mu^f}{2} \mathrm{D}\mathbf{u}^{n+1}_h : \mathrm{D}\widehat{\mathbf{u}}_h \Big] \\ + \displaystyle\int_{\Omega^s_{n+1}} \widetilde{\rho}^s_n \delta t [\widetilde{b}_n (\mathrm{D}\mathbf{u}^{n+1}_h - \nabla \widetilde{\mathbf{d}}^n_h \nabla^{\mathrm{T}} \mathbf{u}^{n+1}_h - \nabla \mathbf{u}^{n+1}_h \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n_h) : \mathrm{D}\widehat{\mathbf{u}}_h \\ + \lambda^s \nabla \cdot \mathbf{u}^{n+1}_h \nabla \cdot \widehat{\mathbf{u}}_h] \\ + \displaystyle\int_{\Omega^s_{n+1}} [\widetilde{b}_n (\mathrm{D}\widetilde{\mathbf{d}}^n_h - \nabla \widetilde{\mathbf{d}}^n_h \nabla^{\mathrm{T}} \widetilde{\mathbf{d}}^n_h) : \mathrm{D}\widehat{\mathbf{u}}_h + (\widetilde{c}_n + \lambda^s \nabla \cdot \widetilde{\mathbf{d}}^n_h) \nabla \cdot \widehat{\mathbf{u}}_h] \\ = \displaystyle\int_{\Omega_{n+1}} \mathbf{f} \cdot \widehat{\mathbf{u}}_h, \quad \Omega_{n+1} = (\mathbb{Y}^{n+1})^{-1}(\Omega_n) = \{x : \ \mathbb{Y}^{n+1}(x) \in \Omega_n\}. \end{cases} \tag{3.8}$$

Then $\mathbf{d}$,

$$\mathbf{d}^{n+1}_h = \widetilde{\mathbf{d}}^n_h + \delta t \mathbf{u}^{n+1}_h.$$

3.6 Implementation

The various tests we made lead us to recommend the following.

• Move the vertices of the triangles supporting the solid with their own velocity:

$$q^{n+1}_i = q^n_i + \mathbf{u}^{n+1}_h(q^{n+1}_i)\delta t, \tag{3.9}$$

which, as explained above has to be implemented through an iterative process.

• Remesh the fluid part at each iteration with a Delaunay-Voronoi mesh generator from the boundary vertices of Σ_{n+1}. However in a Eulerian formulation there is only one mesh, even if

the triangles are marked to be fluid or structure. Hence the fluid boundary must be identified computationally, its oriented edges and vertices; these are then input to the fluid mesh generator as if it was the boundary of an independent fluid domain. Finally two new meshes are merged into a unique fluid-structure mesh.

• In doing so, the discrete topological properties of the structural part are preserved and we have the important property that the value $\mathbf{d}[i]$ of $\mathbf{d}$ at vertex q_i in the computer implementation of $\mathbf{d}$ by an array of values at the nodes, satisfies

$$\mathbf{d}^{n+1}[i] = \mathbf{d}^n[i] + \delta t \mathbf{u}^{n+1}[i], \quad \forall i.$$

In other words, $\mathbf{d}^n \circ \mathbb{Y}^{n+1}$ is $\mathbf{d}^n[i]$ after moving the vertices by (3.9).

4 Energy Estimate

4.1 Stability of the scheme discretised in time

To conserve energy we need to change the scheme (3.8) slightly, from

$$\mathbf{a}(\widetilde{\rho}_n, \widetilde{b}_n, \widetilde{c}_n; \mathbf{u}^{n+1}, \widehat{\mathbf{u}}) = \int_{\Omega_{n+1}} \mathbf{f} \cdot \widehat{\mathbf{u}}_h$$

to

$$\begin{aligned} &\mathbf{a}(\rho_{n+1}, b_{n+1}, c_{n+1}; \mathbf{u}^{n+1}, \widehat{\mathbf{u}}) + \delta t^2 \int_{\Omega^s_{n+1}} \rho^s_{n+1} b_{n+1} \nabla \mathbf{u}_h^{n+1} \nabla^{\mathrm{T}} \mathbf{u}_h^{n+1} : \mathrm{D}\widehat{\mathbf{u}}_h \\ &= \int_{\Omega_{n+1}} \mathbf{f} \cdot \widehat{\mathbf{u}}_h. \end{aligned} \tag{4.1}$$

Lemma 4.1 *The mapping $\mathbf{X}^n : \Omega_0 \mapsto \Omega_n$ is also $\mathbf{X}^{n+1} = (\mathbb{Y}^{n+1})^{-1} \circ \mathbf{X}^n$, $n \geq 1$ and the Jacobian of the transformation is $\mathbf{F}^n := \nabla^{\mathrm{T}}_{x_0} \mathbf{X}^n = (\mathbf{I} - \nabla \mathbf{d}^n)^{-T}$.*

Proof Notice that $\mathbb{Y}^1(\mathbb{Y}^2(..\mathbb{Y}^{n-1}(\mathbb{Y}^n(\Omega_n))..)) = \Omega_0$. Hence

$$\mathbf{X}^{n+1} = [\mathbb{Y}^1(\mathbb{Y}^2(..\mathbb{Y}^n(\mathbb{Y}^{n+1})))]^{-1} = (\mathbb{Y}^{n+1})^{-1} \circ \mathbf{X}^n.$$

By definition of $\mathbf{d}^{n+1}$ in (3.5),

$$\begin{aligned} \mathbf{d}^{n+1}(\mathbf{X}^{n+1}(x_0)) &= \mathbf{d}^n(\mathbb{Y}^{n+1}(\mathbf{X}^{n+1}(x_0))) + \mathbf{u}^{n+1}(\mathbf{X}^{n+1}(x_0))\delta t \\ &= \mathbf{d}^n(\mathbf{X}^n(x_0)) + \mathbf{u}^{n+1}(\mathbf{X}^{n+1}(x_0))\delta t, \end{aligned} \tag{4.2}$$

and since $\mathbf{X}^{n+1}(x_0) = \mathbf{d}^{n+1}(\mathbf{X}^{n+1}(x_0)) + x_0$ we have

$$\begin{aligned} \mathbf{F}^{n+1} &= \nabla^t_{x_0}(\mathbf{d}^{n+1}((\mathbf{X}^{n+1}(x_0))) + x_0) \\ &= \nabla \mathbf{d}^{n+1^{\mathrm{T}}} \mathbf{F}^{n+1} + \mathbf{I} \;\Rightarrow\; \mathbf{F}^{n+1} = (\mathbf{I} - \nabla \mathbf{d}^{n+1})^{-\mathrm{T}}. \end{aligned} \tag{4.3}$$

Note that (4.2) shows also that

$$\mathbf{F}^{n+1} = \mathbf{F}^n + \delta t \nabla^{\mathrm{T}}_{x_0} \mathbf{u}^{n+1}. \tag{4.4}$$

Lemma 4.2 *With Ψ defined by* (2.2),

$$\int_{\Omega^s_{n+1}} \rho^s_{n+1}[b^{n+1}(\mathrm{D}\mathbf{d}^{n+1} - \nabla\mathbf{d}^{n+1}\nabla^{\mathrm{T}}\mathbf{d}^{n+1}) : \mathrm{D}\widehat{\mathbf{u}} + a^{n+1}\nabla\cdot\widehat{\mathbf{u}}]$$
$$= \int_{\Omega^s_0} \partial_{\mathbf{F}}\Psi^{n+1} : \nabla_{x_0}\widehat{\mathbf{u}}. \tag{4.5}$$

Proof By Proposition 2.1 and Lemma 4.1,

$$\int_{\Omega^s_{n+1}} \rho^s_{n+1}(a^{n+1}\mathbf{I} + 2b^{n+1}(\mathrm{D}\mathbf{d}^{n+1} - \nabla\mathbf{d}^{n+1}\nabla^{\mathrm{T}}\mathbf{d}^{n+1})) : \nabla\widehat{\mathbf{u}}$$
$$= \int_{\Omega^s_{n+1}} \sigma^s_{n+1} : \nabla\widehat{\mathbf{u}} = \int_{\Omega^s_{n+1}} [\rho^s_{n+1}\partial_{\mathbf{F}}\Psi\mathbf{F}^{\mathrm{T}}]|_{n+1} : \nabla\widehat{\mathbf{u}}$$
$$= \int_{\Omega^s_{n+1}} [J^{-1}_{n+1}\rho^s_0\partial_{\mathbf{F}}\Psi\mathbf{F}^{\mathrm{T}}]|_{n+1} : \nabla\widehat{\mathbf{u}} = \int_{\Omega^s_0} \rho^s_0\partial_{\mathbf{F}}\Psi^{n+1} : \nabla\widehat{\mathbf{u}}. \tag{4.6}$$

Theorem 4.1 *When $f = 0$ and ρ_0 is constant in each domain Ω^r_0, $r = s, f$, the numerical scheme* (4.1) *has the following property:*

$$\int_{\Omega_n} \frac{\rho_n}{2}|\mathbf{u}^n|^2 + \delta t\sum_{k=1}^{n}\int_{\Omega^f_k}\frac{\mu^f}{2}|\mathrm{D}\mathbf{u}^k|^2 + \int_{\Omega^s_0}\Psi^n \le \int_{\Omega_0}\frac{\rho_0}{2}|\mathbf{u}^0|^2 + \int_{\Omega^s_0}\Psi^0. \tag{4.7}$$

Proof Let $r = s$ or f. Let us choose $\widehat{\mathbf{u}} = \mathbf{u}^{n+1}$ in (4.1). By Schwartz inequality

$$\int_{\Omega^r_{n+1}} \rho_{n+1}(\mathbf{u}^n\circ\mathbb{Y}^{n+1})\cdot\mathbf{u}^{n+1}$$
$$= \rho_0\int_{\Omega^r_{n+1}} (J^{n+1})^{-1}(\mathbf{u}^n\circ\mathbb{Y}^{n+1})\cdot\mathbf{u}^{n+1}$$
$$\le \rho_0\Big(\int_{\Omega^r_{n+1}} (J^{n+1})^{-1}(\mathbf{u}^n\circ\mathbb{Y}^{n+1})^2\Big)^{\frac12}\Big(\int_{\Omega^r_{n+1}} (J^{n+1})^{-1}(\mathbf{u}^{n+1})^2\Big)^{\frac12}$$
$$= \Big[\int_{\Omega^r_n}\rho_n(\mathbf{u}^n)^2\int_{\Omega^r_{n+1}}\rho_{n+1}(\mathbf{u}^{n+1})^2\Big]^{\frac12}$$
$$\le \frac12\int_{\Omega^r_n}\rho^r_n\mathbf{u}^{n2} + \frac12\int_{\Omega^r_{n+1}}\rho^r_{n+1}\mathbf{u}^{n+1^2}.$$

Plugging this estimate in (4.1) with $\widehat{\mathbf{u}} = \mathbf{u}^{n+1}$ leads to

$$\int_{\Omega_{n+1}}\frac{\rho_{n+1}}{2}|\mathbf{u}^{n+1}|^2 + \delta t\int_{\Omega^f_{n+1}}\frac{\mu^f}{2}|\mathrm{D}\mathbf{u}^{n+1}|^2 + \int_{\Omega_0}\Psi^{n+1}$$
$$\le \int_{\Omega_n}\frac{\rho_n}{2}|\mathbf{u}^n|^2 + \int_{\Omega_0}\Psi^n.$$

4.2 Energy estimate for the fully discrete scheme

The proof for the spatially continuous case will work for the discrete case if

$$\mathbf{X}^n = \mathbf{X}^{n+1}\circ\mathbb{Y}^{n+1}. \tag{4.8}$$

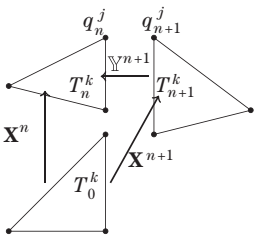

Figure 2 Sketch to understand if $\mathbf{X}^n = \mathbb{Y}^{n+1} \circ \mathbf{X}^{n+1}$ holds with the $P^1 - P1$ stabilised element. A triangle T_0^k in the reference domain (chosen here to be its initial position at time zero) becomes triangle T_n^k at t_n and T_{n+1}^k at time t_{n+1}: $T_n^k = \mathbf{X}^n(T_0^k)$ and $T_{n+1}^k = \mathbf{X}^{n+1}(T_0^k)$, respectively. Vertices are preserved by these transformations.

As discussed in [19] it may be possible to program an isoparametric $P^2 - P^1$ element for which (4.8) but it is certainly far from easy. On the other hand, consider the stabilised $P^1 - P^1$ element: The fluid pressure and the solid pressure are continuous and piecewise linear on the triangulation. The inf-sup condition for stability does not hold unless the incompressibility condition in the fluid, $\nabla \cdot \mathbf{u} = 0$, is changed to $-\alpha\Delta p + \nabla \cdot \mathbf{u} = 0$. In [4], for instance, more details are given explaining why α should be proportional to h^2, in 2D, h being the local size of the mesh edges. It amounts to adding $\alpha\nabla p^{n+1} \cdot \nabla\widehat{p}$ next to the term with μ^f in the variational formulations. Then (4.8) holds (see Figure 2) and the proof of the spatially continuous case can be adapted leading to (4.7) with an additional viscous term $\epsilon|\nabla p^{n+1}|^2$ next to the term with μ^f.

Remark 4.1 Because of energy preservation scheme (4.1), implemented via a fixed point algorithm as in (3.7), generates bounded sequences $\rho, \mathbf{u}, q^i$; it seems safe to assess that out of these bounded subsequences will converge to a solution of the problem discretised in space but continuous in time when $\delta t \to 0$.

5 Numerical Tests

In our tests we have used the $P^2 - P^1$ element with 2 iterations for the nonlinear system at each time step. In most cases 3 iterations are unnecessary but one iteration is not enough. At each iteration the linear system is solved with the library MUMPS – implemented in FreeFem++ (see [18])—and the condition number does not seem to be an issue, which is natural since the main contribution to the matrix is the mass matrix. Although mathematically better the stabilised $P^1 - P^1$ element with $\alpha = 10^{-4}$ did not perform better. For instance, on FLUSTRUK-FSI-2* below, the flag touches, correctly, the bottom boundary at 0.41 when a mesh with 9568 vertices is used and does not converge with a mesh of 2511 vertices.

We have also tested the effect of adding the term of order δt^2 to the nonlinear system: It made no visible differences.

5.1 The cylinder-flag test

A compressible hyperelastic Mooney-Rivlin material, shaped as a rectangle of size $[0,l]\times[0,h]$, is attached behind a cylinder of radius r and beats in tune with the Karman vortices of the wake behind the cylinder; the fluid in the computational rectangular domain $[0,L]\times[0,H]$ enters from the left and is free to leave on the right. The center of the cylinder is at (c,c) (see Figure 1). In [13] the following numerical values are suggested.

Geometry $l=0.35$, $h=0.02$, $L=2.5$, $H=0.41$, $c=0.2$ which puts the cylinder slightly below the symmetry line.

Fluid density $\rho^f = 10^3$ kg/m^3 and a reduced viscosity $\nu^f = \frac{\mu^f}{\rho^f} = 10^{-3}$ m^2/s; inflow horizontal velocity $\mathbf{u}(0,y) = \overline{U}\big(\frac{6}{H^2}y(H-y),0\big)^{\mathrm{T}}$ is a parabolic profile with flux $\overline{U}H$. Top and bottom boundaries are walls with no-slip conditions.

Solid $E=2\mu(1+\sigma)$, $\sigma=0.4$, $\lambda=\frac{E\sigma}{(1+\sigma)(1-2\sigma)}$.

Initial velocities and displacements are zero. In all cases the same mesh is used initially with 2511 vertices. The time step is 0.005.

5.1.1 Free fall of a thick flag

The gravity is $g=9.81$ in Ω_t. When $\overline{U}=0$, $\mu=1.5\,10^6$ and $\rho^s=20\rho^f$, the flag falls under its own weight; it comes to touch the lower boundary with zero velocity at time 0.4 and then moves up under its spring effect. This test is named FLUSTRUK-FSI-2* in [13] but we have used a different value for μ because the one reported in [13] does not give the value used for the gravity.

Figure 3 shows a zoom around the flag at the time when it has stopped to descend and started to move upward. Pressure lines are drawn in the flow region together with the mesh and the velocity vectors in the flag and drawn at each vertex. Figure 4 shows the coordinates of the upper right tip of the flag versus time. It shows also that mass is conserved because the

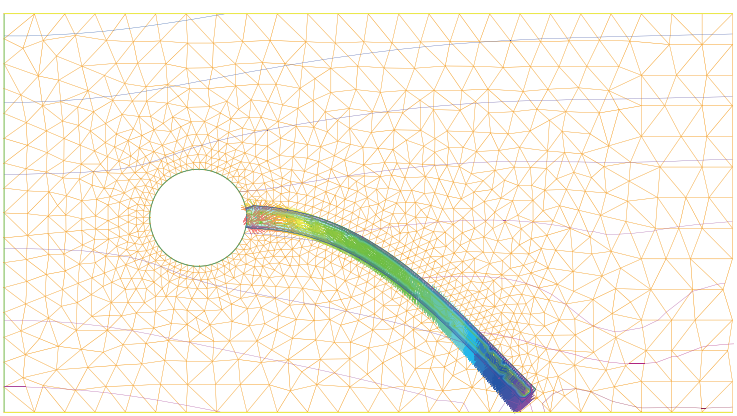

Figure 3 FLUSTRUK-FSI-2* (see [13]). Zoom near the flag at $t=0.4$ just as it begins to move up after the fall under its own weight in a flow initially at rest. Mesh and Pressure lines are shown in the fluid and velocity vectors in the solid.

integral in the solid of Jh is plotted. Finally to check the stability of the algorithm for high values of ρ^s we performed the same test but with $\rho^s = 2\,10^6$ and $\mu^s = 2.5\,10^6$ (with 1.5 instead of 2.5 the flag touches the bottom boundary). Then we compared with $\rho^s = 2$ for the same value of μ^s. Results are shown in Figure 5. Stability is established and a high value of ρ^s affects the frequency of the oscillations.

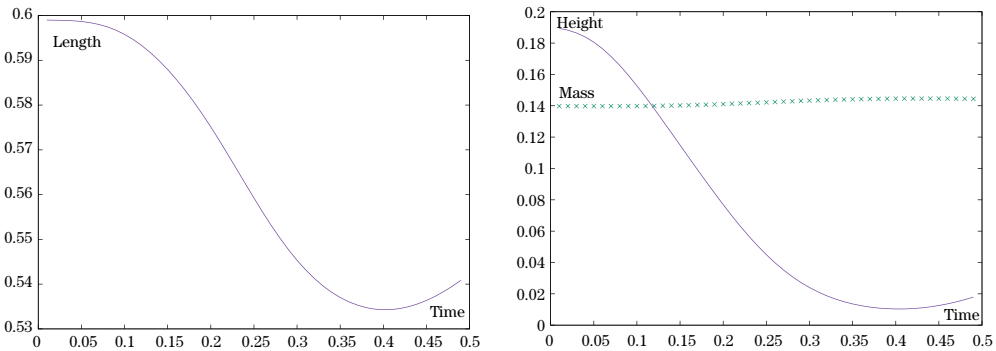

Figure 4 FLUSTRUK-FSI-2* test of [13]. Position of the upper right corner of the flag versus time: x vs t on the left and y vs t on the right. In addition on the right the mass multiplied by 20 is plotted at each time step.

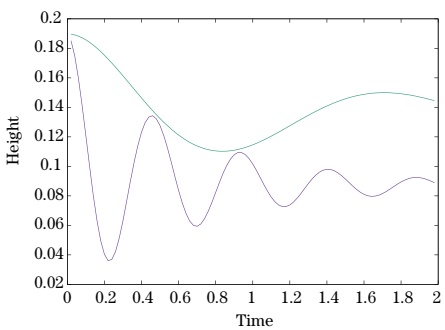

Figure 5 FLUSTRUK-FSI-2*: $\mu^s = 2.5\,10^6$; comparison between a small value of $\rho^s = 2$ (top curve), and a high value $\rho^s = 2\,10^6$ (lower curve).

5.1.2 Flow past a cylinder with a thick flag attached

This test is known as FLUSTRUK-FSI-3 in [13]. The geometry is the same as above but now $\overline{U} = 2$, $\mu = 210^6$ and $\rho^s = \rho^f$. After some time a Karman-Vortex alley develops and the flag beats accordingly. Results are shown in Figures 6–7; the first one displays a snapshot of the velocity vector norms and the second the y-coordinate versus time of the top right corner of the flag.

These numerical results compare reasonably well with those of [13]. The frequency is 5s^{-1} compared to 5.04 and the maximum amplitude 0.031 compared to 0.032. However the results

are still somewhat sensitive to the time step size, the number of iterations in the nonlinear solver and the mesh size. An extensive convergence analysis needs to be made to assert that the precision of these simulations is better than 10%.

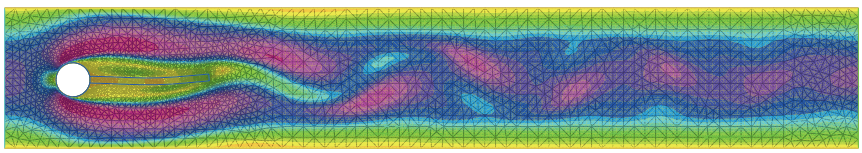

Figure 6 FLUSTRUK-FSI-3 Test. Color map based on the norm of the fluid and solid velocity vectors.

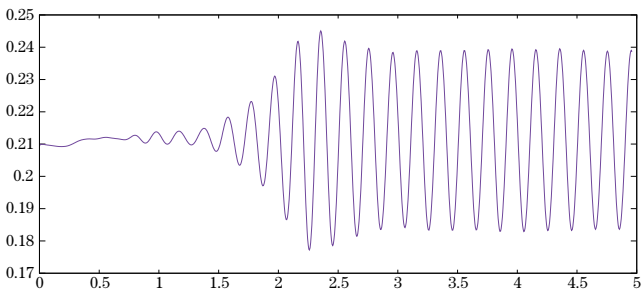

Figure 7 FLUSTRUK-FSI-3 Test. Vertical position of the upper right tip of the flag versus time shown up to t=5.

6 Conclusion

A fully Eulerian fluid-structure formulation has been presented for compressible materials with large displacements, discretised by an implicit first order Euler Scheme and the $P^2 - P^1$ or stabilised $P^1 - P^1$ elements. An energy estimate has been obtained which guarantees the stability of the scheme so long as the motion of the vertices does not flip-over a triangle. The method has been implemented with FreeFem++ (see [18]). It is reasonably robust when the vertices in the structure are moved by their velocities and the fluid is remeshed with an automatic Delaunay mesh generator. The method is first order in time and therefore somewhat too diffusive for delicate tests. It is being extended to 3D and to second order in time discretisation.

7 Appendix

7.1 The Freefem++ Script

We include this script for readers wishing to try the program. First download and install FreeFEM++ and then run this program.

```
// NBPROC 1
load "MUMPS"
// Turek-Dunne FSI Test
load "Curvature"
load "isoline"
verbosity=0;

int m=4;

int la1=10, la2=11;
real U0=0, cx0 = 0.2, cy0 = 0.2; // center of cyl.
real r=0.05, H=0.41, L=2.5; // radius of cylinder, size of domain
real ll=0.35, h2=0.01;//flagella length and half thickness
real la=asin(h2/r), x0=sqrt(r*r-h2*h2);

border fr1(t=0,L){x=t; y=0; label=1;}
border fr2(t=0,H){x=L; y=t; label=2;}
border fr3(t=L,0){x=t; y=H; label=1;}
border fr4(t=H,0){x=0; y=t; label=3;}
border fr5(t=la,2*pi-la){x=cx0+r*cos(-t); y=cy0+r*sin(-t); label=4;}
border br1(t=-la,la){x=cx0+r*cos(-t); y=cy0+r*sin(-t); label=4;}
border br2(t=0,ll){x=cx0+x0+t; y=cy0-h2;label=la1;}
//border br3(t=-pi/2,pi/2){x=cx0+x0+ll+h2*cos(t); y=cy0+h2*sin(t); label=la1;}
border br3(t=-h2,h2){x=x0+cx0+ll; y=cy0+t;label=la1;}
border br4(t=ll,0){x=cx0+x0+t; y=cy0+h2;label=la1;}
func FixBord = fr1(20*m)+fr2(4*m)+fr3(20*m)+fr4(4*m)+fr5(12*m);
//plot(FixBord + br1(m)+br2(6*m)+br3(m)+br4(6*m));
mesh th=buildmesh(FixBord + br1(m)+br2(12*m)+br3(m)+br4(12*m));
// plot(th,wait=1);

// variables have been divided by rhof (which is normally 1000)
// Dunne free fall test
 real Ubar=0, mu=3*0.5e3, rhos=20, gravity=9.81, T=2; int NN=100;
// Turek-Rannacher Karman Vortex text
// real Ubar=2, mu=2e3, rhos=1, gravity=0, T=4; int NN=600;

real rhof=1, nu=1./1000, penal=1e-6, dt=T/NN;
real sigma = 0.4, E = 2*mu*(1+sigma)/rhos,
lambda = E*sigma/(1+sigma)/(1-2*sigma);

int nsl=1;
real lga1;
real[int,int] SLa1(3,nsl);
```

```
lga1=extractborder(th,la1,SLa1);
border a11(t=0,SLa1.m-1){ P.x=SLa1(0,t); P.y=SLa1(1,t); label=la1;}
th=buildmesh( a11((SLa1.m-1))+br1(m)+FixBord,fixeborder=1);
//plot(th,wait=1);
int[int] rr=[0,2]; th=change(th,region=rr);
int fluid=th(0.01,0.01).region, beam=th(x0+cx0+0.1,cy0).region;
// cout<<fluid<<"  "<<beam<<endl;
mesh ths=trunc(th,region==beam);// plot(ths,wait=1);
mesh thf=trunc(th,region==fluid);// plot(thf,wait=1);
mesh thsold=ths;

int nbA;
for(int nbA=0;nbA<ths.nv;nbA++)
if((ths(nbA).x==(x0+cx0+ll)) && (ths(nbA).y==(cy0+h2)))
cout<<"nbA="<<nbA<<" x= "<<ths(nbA).x<<" y= "<<ths(nbA).y<<endl;

fespace V2h(th,P2);
fespace V2hsold(thsold,P2);
fespace Vh(th,P1);
fespace V2hs(ths,P2);
fespace V1h(ths,P1);
fespace Wh(th,[P2,P2,P1]);

Vh p,ph,pp,pph;
V2h uu=0,vv=0,u,v,uh,vh;
V2h uold=0, vold=0, uaux;
V2hs us=0,vs=0, d1=0,d2=0, usold=0,vsold=U0;
// used to keep data on an old mesh
V2hsold dd1,dd2, uusold=0,vvsold=-vsold, uus,vvs;
V1h Jh,tgammah,gammah,ah,bh;
// V2h lambda=0;  // semi-smooth Newton multiplyer

macro div(u,v) ( dx(u)+dy(v) ) // EOM
macro DD(u,v)  [[2*dx(u),div(v,u)],[div(v,u),2*dy(v)]] // EOM
macro Grad(u,v)[[dx(u),dy(u)],[dx(v),dy(v)]] // EOM
macro det(u,v) (dx(u)*dy(v)-dx(v)*dy(u)) //EOM
macro normg(u,v) (dx(u)^2+dy(v)^2+dx(v)^2+dy(u)^2) //EOM
macro J(d1,d2) (1./(1-div(d1,d2)+det(d1,d2))) //EOM
macro tgamma(d1,d2) (2-2*div(d1,d2)+normg(d1,d2)) //EOM

varf GStokesl([u,v,p],[uh,vh,ph]) =
int2d(th,beam)( rhos/Jh*([u,v]'*[uh,vh]/dt
+dt*bh*trace(DD(uh,vh)*(DD(u,v)  -Grad(u,v)*Grad(d1,d2)'
```

```
- Grad(d1,d2)*Grad(u,v)'))
+ dt*lambda*div(u,v)*div(uh,vh)) + penal*p*ph)
+ int2d(th,fluid)(rhof*[u,v]'*[uh,vh]/dt- div(uh,vh)*p
- div(u,v)*ph + penal*p*ph
+ nu/2*trace(DD(uh,vh)'*DD(u,v)) )
// + int1d(th,la1,la2,qfe=qf2pE) (tgv*((lambda + ckunisch*(y-toler))<=0)*v*vh)
+ on(1,4, u=0,v=0) +  on(3,u=Ubar*y*(H-y)*6/H/H,v=0);

varf RHS([u,v,p],[uh,vh,ph]) =
 int2d(th,fluid)(rhof*[convect([uu,vv],-dt,uold),convect([uu,vv],-dt,vold)]'
 *[uh,vh]/dt - rhof*gravity*vh)
 + int2d(th,beam) ( rhos/Jh*([usold,vsold]'*[uh,vh]/dt
 -gravity*vh*(1-0*rhof/rhos) - bh*trace(DD(uh,vh)*( DD(d1,d2)
 - Grad(d1,d2)*Grad(d1,d2)') )-div(uh,vh)*(ah+lambda*div(d1,d2) )  ) )
//+ int1d(th,la1,la2,qfe=qf2pE)
//(tgv*((lambda +ckunisch*(y-toler))<=0)*abs(toler-y)/dt*vh)
+ on(1,4, u=0,v=0) +  on(3,u=Ubar*y*(H-y)*6/H/H,v=0);

Wh [w1,w2,wp];
/*
varf Residual([w1,w2,wp,wpp],[uh,vh,ph,pph]) =
 int2d(th,beam) ( -gravity*vh*rhos - c1*trace(DD(uh,vh)*(DD(d1,d2)
  - Grad(d1,d2)*Grad(d1,d2)'))  //f-Au
  + rhos*[usold,vsold]'*[uh,vh]/dt )
+  int2d(th,fluid)(rhof*[convect([uu,vv],-dt,uold),convect([uu,vv],-dt,vold)]'
*[uh,vh]/dt)
-   int2d(th,beam)( rhos*[u,v]'*[uh,vh]/dt - div(uh,vh)*pp  - div(u,v)*pph
  + penal*pp*pph+ penal*p*ph
+dt*c1*trace(DD(uh,vh)*(DD(u,v)  -Grad(u,v)*Grad(d1,d2)'
- Grad(d1,d2)*Grad(u,v)')))
-  int2d(th,fluid)(rhof*[u,v]'*[uh,vh]/dt- div(uh,vh)*p -div(u,v)*ph
+ penal*p*ph + penal*pp*pph
+ nu/2*trace(DD(uh,vh)'*DD(u,v)));
*/
ofstream ff("tail"+m+".txt");
real t0=0, MMCL=-100, MCL=-100,maxCL=-100,minCL=100;
// Computation time loop
{
for(int n=1;n<NN;n++){
    Jh=J(d1,d2);                // plot(ths,Jh,fill=true,value=1,wait=1);
    tgammah = tgamma(d1,d2);
    gammah = tgammah*Jh^2;
    ah=lambda*(tgammah-1)*(gammah/2-1) + mu*(gammah-Jh^2-1)*tgammah
```

```
       -lambda*div(d1,d2) ;
    bh=(lambda*(gammah/2-1)+mu*(gammah-1))/2;
//  plot(Jh,value=1,wait=1);
    thsold = ths;
    uus=us; vvs=vs; dd1=d1; dd2=d2; uusold=usold; vvsold=vsold;
lambda=0;
    for(int j=0;j<2; j++){      // finds Omega_{n+1} implicitly
{
matrix Al=GStokesl(Wh,Wh,solver=sparsesolver);
real[int] bl=RHS(0,Wh);
[w1,w2,wp]=[0,0,0];
w1[]=Al^-1*bl;
u=w1;v=w2; p=wp;  us=u; vs=v; uu=u; vv=v;
}
 uus[]=us[]; vvs[]=vs[];
     ths = movemesh(thsold,[x+uus*dt,y+vvs*dt]);
     lga1=extractborder(ths,la1,SLa1);
     thf=buildmesh( a11(-(SLa1.m-1))+FixBord,fixeborder=1);
     th=ths+thf;     // plot(th,wait=1);
th=change(th,region=rr);
d1=0; d2=0; usold=0; vsold=0;
// copy old values in d1,d2 defined with new ths (d(X^n))
d1[]=dd1[];  d2[]=dd2[];  usold[]=uusold[]; vsold[]=vvsold[];
    }
    usold[]=uus[]; vsold[]=vvs[];
uold=u; vold=v;
real xm=int2d(ths)(x), ym=int2d(ths)(y),
    um=int2d(ths)(uus), vm=int2d(ths)(vvs), nm=int2d(ths)(x^2+y^2);
    real yum=int2d(ths)(y*uus), xvm=int2d(ths)(x*vvs), onem=int2d(ths)(1.);
    real omega=-(yum-xvm-ym*um/onem+xm*vm/onem) / (nm-(ym^2+xm^2)/onem);
uus=uus+omega*y-(um+omega*ym)/onem;
    vvs=vvs-omega*x-(vm-omega*xm)/onem;
d1[]=dd1[]+dt*uus[]; // update array computed on old mesh
d2[]=dd2[]+dt*vvs[];
uaux= sqrt(uold^2+vold^2);
int n1=n+1000;
real CL=ths(nbA).y;
if(minCL>CL) minCL=CL;
if(MMCL<MCL && CL<MCL && MCL>0.2 && MMCL>0.2) {
cout<<"tf= "<<n*dt-t0<<" minCL= "<<minCL<<" maxCL= "<<MCL<<" amplitude= "
    <<(MCL-minCL)/2<<endl;
MCL=-100; MMCL=-100;
t0=n*dt; minCL=10;
```

```
}
MMCL=MCL; MCL=CL;
ff<<n*dt<<"  "<<ths(nbA).x<<" "<<ths(nbA).y<<"  "<< int2d(ths)(Jh)<<endl;
func box=[[-0.1,-0.1],[1,0.5]];
 plot(th,bb=box,[us,vs],p,fill=0,value=0,wait=0,cmm="t="+n*dt,
 ps="geom"+n+".ps");
 }
}
```

Acknowledgement The author thanks Frédéric Hecht for very valuable discussions and comments.

References

[1] Antman, S. S., Nonlinear Problems of Elasticity, (2nd ed.), Applied Mathematical Sciences, **107**, Springer-Verlag, New York, 2005.

[2] Bathe, K. J., Finite Element Procedures, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[3] Bathe, K. J., Ramm, E. and Wilson, E. L., Finite element formulations for large deformation dynamic analysis, *Int. J. Numer. Methods Eng.*, **9**(2), 1975, 353–386.

[4] Boffi, D., Brezzi, F. and Fortin, M., Mixed Finite Element Methods and Applications, Computational Mathematics, Heidelberg, **44**, Springer-Verlag, Berlin, 2013.

[5] Boffi, D., Cavallini, N. and Gastaldi, L., The finite element immersed boundary method with distributed Lagrange multiplier, *SIAM J. Numer. Anal.*, **53**(6), 2015, 2584–2604.

[6] Boulakia, M., Existence of weak solutions for the motion of an elastic structure in an incompressible viscous fluid, *C. R. Math. Acad. Sci. Paris*, **336**(12), 2003, 985–990.

[7] Bukaca, M., Canic, S., Glowinski, R., et al., Fluid-structure interaction in blood flow capturing non-zero longitudinal structure displacement, *Journal of Computational Physics*, **235**, 2013, 515–541.

[8] Chiang, C.-Y., Pironneau, O., Sheu, T. and Thiriet, M., Numerical study of a 3D Eulerian monolithic formulation for fluid-structure-interaction, *Fluids*, 2017.

[9] Ciarlet, P. G., Mathematical Elasticity, I., Three-dimensional Elasticity, North Holland, Amsterdam, 1988.

[10] Cottet, G. H., Maitre, E. and Milcent, T., Eulerian formulation and level set models for incompressible fluid-structure interaction, *M2AN Math. Model. Numer. Anal.*, **42**(3), 2008, 471–492.

[11] Coupez, T., Silva, L. and Hachem, E., Implicit Boundary and Adaptive Anisotropic Meshes, New challenges in Grid Generation and Adaptivity for Scientific Computing, S. Peretto and L. Formaggia (eds.), **5**, Springer-Verlag, Cham, 2015.

[12] Coutand, D. and Shkoller, S., Motion of an elastic solid inside an incompressible viscous fluid, *Arch. Ration. Mech. Anal.*, **176**(1), 2005, 25–102.

[13] Dunne, T., Adaptive finite element approximation of fluid-structure interaction based on an Eulerian variational formulation, ECCOMAS CFD, 2006, Wesseling, P., Oñate, E. and Périaux, J. (eds.), Elsevier, TU Delft, The Netherlands, 2006.

[14] Dunne, T., An Eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation, *Int. J. Numer. Meth. Fluids*, **51**, 2006, 1017–1039.

[15] Dunne, Th. and Rannacher, R. Adaptive Finite Element Approximation of Fluid-Structure Interaction Based on an Eulerian Variational Formulation, Fluid-Structure Interaction: Modelling, Simulation, Optimization, Bungartz, H-J. and Schaefer, M. (eds.), Lecture Notes in Computational Science and Engineering, **53**, Springer-Verlag, Berlin, 2006, 110–146.

[16] Fernandez, M. A., Mullaert, J. and Vidrascu, M., Explicit Robin-Neumann schemes for the coupling of incompressible fluids with thin-walled structures, *Comp. Methods in Applied Mech. and Engg.*, **267**, 2013, 566–593.

[17] Formaggia, L., Quarteroni, A. and Veneziani, A., Alessandro Multiscale Models of the Vascular System, Cardiovasuclar Mathematics, Springer-Verlag, Italia, Milan, 2009, 395–446.

[18] Hecht, F., New development in FreeFem++, *J. Numer. Math.*, **20**, 2012, 251–265, http:// www.FreeFem.org.

[19] Hecht, F. and Pironneau, O., An energy stable monolithic Eulerian fluid-structure finite element method, *International Journal for Numerical Methods in Fluids*, **85**(7), 2017, 430–446.

[20] Change Heil, Matthias to Heil, M., Solvers for large-displacement fluid structure interaction problems: Segregated versus monolithic approaches, *Comput. Mech.*, **43**, 2008, 91–101.

[21] Hron, J. and Turek, S., A monolithic fem solver for an ALE formulation of fluid-structure interaction with configuration for numerical benchmarking, European Conference on Computational Fluid Dynamics ECCOMAS CFD, 2006, Wesseling, P., Onate, E. and Periaux, J. (eds.), TU Delft, The Netherlands, 2006.

[22] Léger, S., Méthode lagrangienne actualisée pour des problèmes hyperélastiques en très grandes déformations, Thèse de Doctorat, Université Laval, 2014 (in France).

[23] Le Tallec, P. and Hauret, P., Energy conservation in Fluid-Structure Interactions, Numerical Methods for Scientific Computing, Variational Problems And Applications, Neittanmaki, P., Kuznetsov, Y. and Pironneau, O. (eds.), CIMNE, Barcelona, 2003.

[24] Le Tallec, P. and Mouro, J., Fluid structure interaction with large structural displacements, *Comp. Meth. Appl. Mech. Eng.*, **190**(24–25), 2001, 3039–3068.

[25] Liu, J., A second-order changing-connectivity ALE scheme and its application to FSI with large convection of fluids and near-contact of structures, *Journal of Computational Physics*, **304**, 2016, 380–423.

[26] Liu, I-Shih, Cipolatti, R. and Rincon, M. A., Incremental Linear Approximation for Finite Elasticity, Proc. ICNAAM, Wiley, 2006.

[27] Marsden, J. and Hughes, T. J. R., Mathematical Foundations of Elasticity, Dover Publications, New York, 1994.

[28] Nobile, F. and Vergara, C., An effective fluid-structure interaction formulation for vascular dynamics by generalized Robin conditions, *SIAM J. Sci. Comp.*, **30**(2), 2008, 731–763.

[29] Peskin, C. S., The immersed boundary method, *Acta Numerica*, **11**, 2002, 479–517.

[30] Pironneau, O., Numerical Study of a Monolithic Fluid-Structure Formulation, Variational Analysis and Aerospace Engineering, **116**, Springer-Verlag, Cham, 2016.

[31] Rannacher, R. and Richter, T., An Adaptive Finite Element Method for Fluid-Structure Interaction Problems Based on a Fully Eulerian Formulation, Lecture Notes in Computational Science and Engineering, **73**, Springer-Verlag, Heidelberg, 2010.

[32] Raymond, J.-P. and Vanninathan, M., A fluid-structure model coupling the Navier-Stokes equations and the Lamé system, *J. Math. Pures Appl.*, **102**, 2014, 546–596.

[33] Richter, Th. and Wick, Th., Finite elements for fluid-structure interaction in ALE and fully Eulerian coordinates, *Comput. Methods Appl. Mech. Engrg.*, **199**, 2010, 2633–2642.

[34] Wang, Y. X., The Accurate and Efficient Numerical Simulation of General Fluid Structure Interaction: A Unified Finite Element Method, Proc. Conf. on FSI problems, IMS-NUS, Singapore, 2016.