INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2007; **72**:46–71
Published online 29 January 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1998

On some fluid–structure iterative algorithms using pressure segregation methods. Application to aeroelasticity

Santiago Badia*, † and Ramon Codina

International Center for Numerical Methods in Engineering (*CIMNE*), *Universitat Politècnica de Catalunya*, *Jordi Girona 1-3*, *Edifici C1*, *08034 Barcelona*, *Spain*

SUMMARY

In this paper we suggest some algorithms for the fluid–structure interaction problem stated using a domain decomposition framework. These methods involve stabilized pressure segregation methods for the solution of the fluid problem and fixed point iterative algorithms for the fluid–structure coupling. With one single loop the solution of the coupled system tends to one of the strongly coupled monolithic systems. These coupling algorithms are applied to the aeroelastic simulation of suspension bridges. We assess flexural and torsional frequencies for a given inflow velocity. Increasing this velocity we reach the value for which the flutter phenomenon appears. Copyright © 2007 John Wiley & Sons, Ltd.

Received 10 January 2006; Revised 15 December 2006; Accepted 19 December 2006

KEY WORDS: fluid–structure interaction; pressure segregation; aeroelasticity; bridge aerodynamics

1. INTRODUCTION

1.1. State of the art

The interaction between a fluid and a structure appears in a wide variety of fields. Probably, the most analysed fluid–structure interaction problem is the aeroelastic one (specially for aeronautical applications), for instance in the simulation of the action of a fluid (air) over a structure (such as a wing or a bridge). Recently, an increasing interest in the simulation of hemodynamics has motivated a lot of research on fluid–structure algorithms appropriate for the blood–vessel system.

The implementation of a coupled problem can be done using two different global strategies. The *monolithic* strategy implies the solution of the coupled problems simultaneously. *Partitioned*

*Correspondence to: Santiago Badia, International Center for Numerical Methods in Engineering (CIMNE), Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.
†E-mail: sbadia@cimne.upc.edu

Contract/grant sponsor: Ministerio de Educación y Ciencia of Spain through Plan Nacional I+D+I; contract/grant number: Project ADEL (BIA2003-09078-C02-01)

methods are usually used in order to keep software modularity and to allow the use of the numerical methods developed for every field separately. When using pressure segregation methods for the fluid problem (as in this work) *partitioned* procedures are naturally adapted.

The numerical simulation of the fluid–structure coupled problem is complicated. It does not only inherit the difficulties associated with the fluid and structure simulations, but the coupling of these two systems is also a hard task in many situations. The difficulties arising from this coupled system depend strongly on the physical properties of the case to be simulated. Thus, the choice of an appropriate algorithm that deals well with the coupling varies with the problem to solve. For instance, applications in aeroelasticity and hemodynamics have very different behaviour for the same coupling algorithm. Whereas for aeroelastic problems there is a clear tendency to solve the coupled system using explicit procedures (see [1] and references therein) or implicit Dirichlet–Neumann schemes and Richardson iterations (see [2, 3]), these methods are not appropriate in most hemodynamics applications. In the last case, the use of more involved implicit procedures for the coupling are required in order to reach good convergence. This situation can be explained by the *added mass* effect (see [4]). When the structure density ρ_s is much larger than the fluid density ρ_f (as it happens in aeroelasticity), the classical Dirichlet–Neumann approach is stable, even without relaxation, and exhibits a good convergence ratio. On the contrary, when the fluid and structure densities are of the same order (as in hemodynamics) the *added mass* introduced by the fluid over the structural problem makes the convergence of this coupling algorithm too involved. In some cases, even with relaxation, convergence cannot be reached. And when relaxation is enough, too small relaxation parameters and lots of iterations are needed. In order to obtain convergence, GMRES iterations are used (instead of simpler Richardson iterations) and more involved linearizations, like Newton and quasi-Newton algorithms, have been suggested (see [5, 6]). The relaxation of these methods is a key aspect in order to reach convergence when dealing with these problems, and some possibilities have been used (see [3, 7]).

Herein we want to obtain appropriate algorithms for the simulation of fluid–structure problems using finite element methods. The interpretation of the coupling of the fluid and structure as a domain decomposition (DD) method without overlapping used in [8] is adopted. We apply these algorithms to the aeroelastic analysis of bridges. These algorithms are based on the blend of fixed point algorithms of Dirichlet–Neumann type with pressure segregation methods. We assume a Newtonian and incompressible fluid. The structure, as it is usually done in the analysis of these problems, is considered a rigid body with elastic coefficients in the rigid body motion degrees of freedom (dofs).

1.2. Ingredients of a fluid–structure solver

In this kind of problems the displacement of the structure changes the domain of the fluid. Then, the fluid equations have to be able to deal with moving domains. With this aim we use an *arbitrary Lagrangian Eulerian* (ALE) approach. Some ALE formulations have been analysed in [9–11]. The ALE scheme has an intrinsic error in time that can spoil the accuracy of the fluid solver in fixed domains. For this reason, an appropriate ALE scheme depends on the time accuracy of the fluid solver for fixed domains. The ALE approach involves the movement of the domain (mesh) with appropriate Dirichlet boundary conditions. This movement is defined by a mesh displacement. Different techniques have been proposed for its computation. The most widely used is the harmonic extension of the Dirichlet functions on the boundaries, this methodology being the one adopted in the present work.

DOI: 10.1002/nme

The fluid solver for incompressible flows is a key point of the algorithm because it consumes most of the CPU time. The monolithic treatment of the Navier–Stokes equations is involved (for the system solver) and time consuming. In order to improve the situation, we suggest the use of pressure segregation methods in their fractional step and predictor–corrector forms (see [12, 13]). On the other hand, we use the orthogonal subgrid scale stabilized finite element method (see [14]) for the space discretization, that allows the use of equal velocity–pressure interpolation.

Less attention is paid to the structure solver. The following exposition can be applied to any kind of structural problem, with linear or non-linear material behaviour. Nevertheless, in the application we have considered, the structure is considered to be a rigid body. Thus, the computational cost of the structure is much lower than the computational cost of the fluid.

1.3. Outline of the article

We have organized the present work as follows. In Section 2 we motivate a novel fluid–structure algorithm and justify its application to aeroelasticity. We state every field problem in its continuous level and some notation is introduced in Section 3. In Section 4 we write the interface equation associated to the problem under consideration, using a DD framework. At the fully discrete level, we introduce the fluid solvers and appropriate coupling procedures (Section 5) involving pressure segregation methods. Section 6 is devoted to the application of these methods to the simulation of bridge aerodynamics. Section 7 concludes the paper by drawing some conclusions.

2. MOTIVATION OF THE WORK

The main contribution of this work is the use of an implicit coupling scheme using Dirichlet–Neumann and Richardson iterations for the fluid–structure coupling together with a pressure segregation method that uncouples the solution of velocity components and pressure. With only one loop, the solution of the coupled system tends to the strongly coupled fluid–structure system. In the rest of the section we emphasize the motivation of this novel algorithm and its application to aeroelasticity.

As explained above, the appropriate algorithm for the solution of the coupled system depends on the kind of problem to be solved. In this paper we have in mind aeroelastic problems. Let us draw some features about this sort of applications:

- The fluid solver consumes much more CPU time than the structure solver. For this reason, the number of fluid evaluations has to be minimized in order to reduce the computational cost.
- The convergence of the coupling iterative process is *easy*. As explained in Section 1, this behaviour is associated with the fact that the structure density is much larger than the fluid density.

For these applications explicit procedures and implicit procedures using simple fixed point algorithms are the most usual approaches. The convergence rate of the Dirichlet–Neumann sub-structuring technique (with Richardson iterations) is good, because the added mass effect can be treated explicitly. That, together with the fact that the fixed point algorithm minimizes the number of fluid evaluations per iteration, has motivated its use for aeroelastic applications.

We can say that, for a given time step size, explicit procedures are cheaper than implicit procedures. However, explicit procedures are also less accurate and less stable. Moreover, when using explicit techniques we are restricted to *small enough* time step sizes or otherwise the solution

DOI: 10.1002/nme

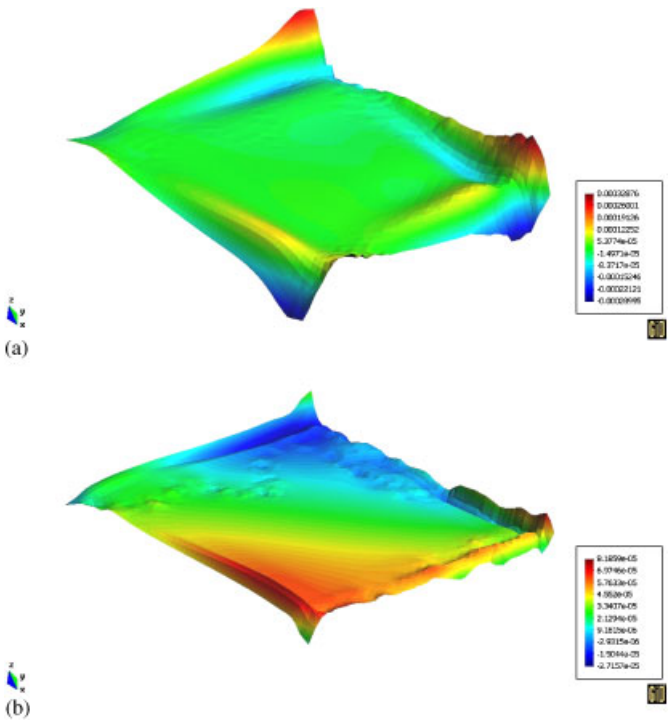

Figure 1. Pressure error plots: (a) without predictor–corrector iterations and (b) with predictor–corrector iterations.

explodes. Implicit procedures allow larger time step sizes, reducing the number of evaluations. Furthermore, the convergence of simple fixed point iterations in aeroelastic applications is easy.

Even though explicit procedures seem to be a good and stable approach for aeronautical application involving compressible flows, when dealing with incompressible flows (for instance in bridge aerodynamics), the stability is a more critical issue. The situation is more dramatic when using second-order schemes (in time), needing more *unstable* extrapolations of displacement and mesh velocity (in order to keep overall accuracy). Thus, in aeroelastic applications, where high-order schemes in time are appealing, due to the highly transient behaviour, high-order implicit procedures are a good choice. At every time step, fluid and structure solutions are *equilibrated*, freeing the simulation of the undesirable numerical instabilities of the explicit coupling.

The drawback of an implicit method is the fact that some evaluations of the fluid are needed per time step value, even though, as pointed out above, larger time step sizes can be used. And the bottle neck of the coupling is the fluid solver. In order to reduce the CPU cost of the implicit approach we have considered pressure segregation (or fractional step type) methods. Doing that, the initial fluid system, with $d+1$ dofs per node is reduced to $d+1$ small systems (with 1 dof per node). It implies a large reduction of the fluid solver computational cost. The main problem related to this splitting is the fact that we are introducing artificial pressure boundary conditions on the interface. This misbehaviour is important in fluid–structure applications, where the pressure on the interface is a key value that drives the coupling. As explained in Section 5 and showed in Figure 1 for a test problem, predictor–corrector methods that consist on iterating over the

DOI: 10.1002/nme

split system are an appealing choice because we can still decouple every velocity component and pressure computations and tend to the monolithic solution, reducing the pressure boundary layer and splitting error.

The last step is to consider the same loop for the implicit coupling and the predictor–corrector fluid solver. This is the candidate algorithm that we will consider in this work, because

- The coupling is treated implicitly, getting equilibrated fluid–structure solutions and can be easily extended to high-order methods (in time). The solution is not affected by numerical instability, basic when evaluating physical instabilities (as flutter).
- The computational cost of the fluid solver is clearly reduced.
- The quality of the pressure on the interface is improved along the iterative process.
- And only with one loop that deal with the fluid solver and coupling, the solution tend to the strongly coupled implicit system, the *reference solution*.

Due to the fact that the solution obtained with this method converges to the monolithic implicit system (apart from the tolerance of the stopping criterion or maximum number of iterations), the convergence behaviour of this iterative procedure in aeroelastic applications is the most important aspect to be considered using numerical experimentation. This algorithm is introduced in Section 5 and used for the evaluation of the flutter velocity of a bridge (involving the simulation of incompressible flows and large Reynolds numbers) in Section 6.

3. THE CONTINUOUS PROBLEM

In this section we introduce the fluid–structure problem at the continuous level. First, we treat some aspects about the problem domain, the definition of its movement and its restriction to the fluid and structure, the domain velocity and the matching conditions that these restrictions satisfy on the interface. After that, we state the governing equations of the fluid and structure problems and suggest how to calculate the domain displacement. We conclude this section with the matching conditions (that is, continuity of some values) that have to be imposed over the interface between the fluid and the solid.

We denote by Ω_t the domain occupied by the heterogeneous mechanical system at a given time $t>0$. This domain is divided into the structure domain Ω_t^{s} and its complement Ω_t^{f} occupied by the fluid. We denote by $\Sigma_t \equiv \partial\Omega_t^{\mathrm{f}} \cap \partial\Omega_t^{\mathrm{s}}$ the fluid–structure interface. Further, $\mathbf{n}_{\mathrm{f}}$ is the outward normal of Ω_t^{f} on Σ_t and $\mathbf{n}_{\mathrm{s}}$ its counterpart for the structure.

The total domain Ω_t is defined at every time instant by a family of mappings $\mathscr{A}_t$

$$\mathscr{A}_t : \Omega_0 \longrightarrow \Omega_t$$

where Ω_0 is the reference domain associated to $t=0$. We stress the fact that $\mathscr{A}_t$ is arbitrary.

Let us introduce some notation. Given a function $f : \Omega_t \times [0, T] \longrightarrow \mathbb{R}$ defined at the current domain we indicate by $\hat{f} = f \circ \mathscr{A}_t$ the corresponding function defined at the initial configuration

$$\hat{f} : \Omega_0 \times [0, T] \longrightarrow \mathbb{R}, \quad \hat{f}(\mathbf{x}_0, t) = f(\mathscr{A}_t(\mathbf{x}_0), t)$$

Furthermore, the time derivatives at the initial configuration are defined as follows:

$$\left.\frac{\partial f}{\partial t}\right|_{\mathbf{x}_0} : \Omega_t \times (0, T) \longrightarrow \mathbb{R}, \quad \left.\frac{\partial f}{\partial t}\right|_{\mathbf{x}_0}(\mathbf{x}, t) = \frac{\partial \hat{f}}{\partial t}(\mathbf{x}_0, t)$$

DOI: 10.1002/nme

We denote by $\mathbf{d}(\mathbf{x}, t)$ the displacement of the domain evaluated at the current configuration. Then, we could write the mapping $\mathscr{A}_t$ as $\mathscr{A}_t(\mathbf{x}_0, t) = \mathbf{x}_0 + \widehat{\mathbf{d}}(\mathbf{x}_0, t)$. We split the domain displacement into its fluid and structure restriction as $\mathbf{d} = \mathscr{R}_{\Omega_t^{\mathrm{s}}}\mathbf{d} + \mathscr{R}_{\Omega_t^{\mathrm{f}}}\mathbf{d} =: \mathbf{d}^{\mathrm{s}} + \mathbf{d}^{\mathrm{f}}$. From the trace theorem we know that

$$\mathbf{d}^{\mathrm{s}}|_{\Sigma_t} = \mathbf{d}^{\mathrm{f}}|_{\Sigma_t} \tag{1}$$

has to be satisfied. Moreover, we define

$$\mathbf{w} = \left.\frac{\partial \mathbf{d}^{\mathrm{f}}}{\partial t}\right|_{\mathbf{x}_0} \tag{2}$$

which is the domain velocity that we will require in order to write the fluid equations in an ALE framework.

In the present work we assume a Newtonian incompressible fluid. We use the ALE formulation in order to write the Navier–Stokes equations on moving domains. In what follows we only consider the boundary conditions on Σ_t. The rest of boundary conditions are essential for the definition of the problem but do not affect the following exposition. For this reason we have omitted them for the sake of clarity. The Navier–Stokes equations that govern the fluid problem read as follows: find a velocity field $\mathbf{u}$ and a pressure field p such that

$$\rho_{\mathrm{f}}\frac{\partial \mathbf{u}}{\partial t} - \mu\Delta\mathbf{u} + \rho_{\mathrm{f}}\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \rho_{\mathrm{f}}\mathbf{f}_{\mathrm{f}} \quad \text{in } \Omega_t^{\mathrm{f}} \times (0, T) \tag{3a}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega_t^{\mathrm{f}} \times (0, T) \tag{3b}$$

where ρ_{f} is the density and μ the viscosity of the fluid. The Cauchy stress tensor for the fluid is $\boldsymbol{\sigma}^{\mathrm{f}} = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$ where $\boldsymbol{\varepsilon}(\mathbf{u}) = (\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}})/2$ is the strain rate tensor and $\mathbf{I}$ the identity matrix. We denote by $\boldsymbol{\sigma}_n^{\mathrm{f}} := \boldsymbol{\sigma}^{\mathrm{f}}|_{\Sigma_t}\cdot\mathbf{n}_{\mathrm{f}}$ the normal stress on Σ_t.

Let us recall the *Reynolds transport formula*. Let $\psi(\mathbf{x}, t)$ be a function defined on Ω_t. Then, for any subdomain $V_t \subseteq \Omega_t$ such that $V_t = \mathscr{A}_t(V_0)$ with $V_0 \subseteq \Omega_0$ it holds that

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_t}\psi(\mathbf{x}, t)\,\mathrm{d}V = \int_{V_t}\left(\left.\frac{\partial\psi}{\partial t}\right|_{\mathbf{x}_0} + \psi\nabla\cdot\mathbf{w}\right)\mathrm{d}V \tag{4}$$

At this point, using expression (4) for the time derivative, we can write the fluid equations (3) in the ALE framework as follows: find a velocity $\mathbf{u}$ and a pressure p such that

$$\rho_{\mathrm{f}}\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{x}_0} - \mu\Delta\mathbf{u} + \rho_{\mathrm{f}}(\mathbf{u}-\mathbf{w})\cdot\nabla\mathbf{u} + \nabla p = \rho_{\mathrm{f}}\mathbf{f}_{\mathrm{f}} \quad \text{in } \Omega_t^{\mathrm{f}} \times (0, T) \tag{5a}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega_t^{\mathrm{f}} \times (0, T) \tag{5b}$$

Remark 1
Formulations (3) and (5) are equivalent at the continuous level.

The structure can easily handle moving domains using a fully Lagrangian framework. For instance, we consider an elastic structure. We denote by $\boldsymbol{\sigma}_n^{\mathrm{s}} := \boldsymbol{\sigma}^{\mathrm{s}}|_{\Sigma_t}\cdot\mathbf{n}_{\mathrm{s}}$ the normal stress on Σ_t. The displacement of the structure domain $\mathbf{d}^{\mathrm{s}}$ is assumed to be equal to the structure displacement.

DOI: 10.1002/nme

The fluid displacement $\mathbf{d}^{\mathrm{f}}$ is arbitrary but has to satisfy condition (1). Thus, we can write $\mathbf{d}^{\mathrm{f}}$ as an arbitrary extension of $\mathbf{d}^{\mathrm{s}}|_{\Sigma_t}$ into Ω_t^{f}, that we denote by $\mathbf{d}^{\mathrm{f}} = \mathrm{Ext}(\mathbf{d}^{\mathrm{s}}|_{\Sigma_t})$. Different choices of the lifting operator $\mathrm{Ext}(\cdot)$ have been proposed in the literature. Herein, we adopt an harmonic extension evaluated at the current domain Ω_t^{f}. In this case, $\mathbf{d}^{\mathrm{f}}$ is solution of the Laplace problem.

At this point, suitable matching conditions have to be applied on the interface Σ_t. These are continuity of normal stresses (due to the action–reaction principle) and velocities (due to the perfect adherence of the fluid to the structure):

$$\mathbf{u} = \frac{\partial \widehat{\mathbf{d}}^{\mathrm{s}}}{\partial t} \quad \text{on } \Sigma_t \times (0, T) \tag{6}$$

$$\boldsymbol{\sigma}_n^{\mathrm{f}} + \boldsymbol{\sigma}_n^{\mathrm{s}} = \mathbf{0} \quad \text{on } \Sigma_t \times (0, T) \tag{7}$$

Then, the fluid–structure coupled problem is completely defined by the fluid problem (5), the structure problem, the fluid domain displacement and the interface matching conditions (1), (6) and (7). For the statement of the weak form of the fluid problem, let us define the following spaces, for a given $t \in (0, T)$:

$$\mathscr{V}(\Omega_t^{\mathrm{f}}) := \{\mathbf{v} : \Omega_t^{\mathrm{f}} \to \mathbb{R}^d, \mathbf{v} = \hat{\mathbf{v}} \circ (\mathscr{A}_t^{\mathrm{f}})^{-1}, \hat{\mathbf{v}} \in (H^1(\Omega_0^{\mathrm{f}}))^d\}$$

$$\mathscr{V}_0(\Omega_t^{\mathrm{f}}) := \{\mathbf{v} \in \mathscr{V}(\Omega_t^{\mathrm{f}}) | \mathbf{v}|_{\Sigma_t} = \mathbf{0}\}$$

$$\mathscr{Q}(\Omega_t^{\mathrm{f}}) := \{q : \Omega_t^{\mathrm{f}} \to \mathbb{R}, q = \hat{q} \circ (\mathscr{A}_t^{\mathrm{f}})^{-1}, \hat{q} \in L^2(\Omega_0^{\mathrm{f}})\}$$

$$\Gamma(\Sigma_t) := \{\boldsymbol{\gamma} : \Sigma_t \to \mathbb{R}^d, \boldsymbol{\gamma} = \hat{\boldsymbol{\gamma}} \circ (\mathscr{A}_t|_{\Sigma_t})^{-1}, \hat{\boldsymbol{\gamma}} \in (H^{1/2}(\Sigma_0))^d\}$$

The notation used here is as follows: $L^2(\omega)$ denotes the space of square integrable functions in a spatial domain ω, $H^1(\omega)$ is the space of functions in $L^2(\omega)$ with first derivatives in $L^2(\omega)$, and $H^{1/2}(\sigma)$ is the space of functions defined on a $d-1$-manifold σ that are the trace of functions in $H^1(\omega)$, with $\sigma \subset \partial\omega$. For functions f and g defined on a d- or $d-1$-manifold, we write $\langle f, g\rangle_\omega := \int_\omega fg \, \mathrm{d}\omega$, omitting the subscript when ω is the domain where the problem under consideration is posed. For σ a $d-1$-manifold and $f \in H^{1/2}(\sigma)$, the space of functions g such that $\langle f, g\rangle_\sigma < \infty$ is denoted by $H^{-1/2}(\sigma)$. Finally, $(\cdot, \cdot)$ denotes the usual L^2 product in the domain where the problem considered is posed.

4. THE DOMAIN DECOMPOSITION APPROACH

In this section we reformulate the fluid–structure problem in a DD framework, as done in [15] and later works [7, 16].

Let us consider the time discretized version of (5) using backward-differencing formulas (BDF) for the time integration at time step $t^{n+1} = (n+1)\delta t$, $\delta t > 0$ being the time step size (assumed constant for simplicity). We denote the BDF-p operator as

$$D_p f^{n+1} = \frac{1}{\delta t \gamma_p} \left(f^{n+1} + \sum_{i=1}^{p} \alpha_p^i f^{n+1-i} \right) \tag{8}$$

DOI: 10.1002/nme

where f is a generic time dependent function, f^n denotes its approximation at t^n, p is the order of accuracy of the scheme and γ_p and α_p^i are the parameters that define the BDF numerical integration (see [17]). The first- and second-order BDF methods are defined as:

$$D_1 f^{n+1} = f^{n+1} - f^n$$

$$D_2 f^{n+1} = \tfrac{3}{2}(f^{n+1} - \tfrac{4}{3} f^n + \tfrac{1}{3} f^{n-1})$$

At a fixed time step $n+1$, let us denote by $\boldsymbol{\lambda}$ the interface variable corresponding to the displacement on the fluid–structure interface, $\mathbf{d}|_{\Sigma_{t^{n+1}}}$. We denote by $\mathsf{FL}_{\delta t}$ the operator that gives the velocity and pressure field at t^{n+1} for a given $\boldsymbol{\lambda}$

$$\mathsf{FL}_{\delta t} : \Gamma(\Sigma_{t^{n+1}}) \to \mathscr{V}(\Omega^{\mathrm{f}}_{t^{n+1}}) \times \mathscr{Q}(\Omega^{\mathrm{f}}_{t^{n+1}})$$

$$\boldsymbol{\lambda} \mapsto (\mathbf{u}^{n+1}, p^{n+1})$$

There are multiple choices for the $\mathsf{FL}_{\delta t}(\boldsymbol{\lambda})$ operator, corresponding to the different possibilities for the time approximation of the incompressible Navier–Stokes equations, such as the monolithic system or the fractional step version at the continuous level in space. Let us start with the monolithic scheme. In this case, $\mathsf{FL}_{\delta t}(\boldsymbol{\lambda}) = (\mathbf{u}^{n+1}, p^{n+1})$ is computed by solving the problem: given $\boldsymbol{\lambda} \in \Gamma(\Sigma_{t^{n+1}})$, find $\mathbf{u}^{n+1} \in \mathscr{V}(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p^{n+1} \in \mathscr{Q}(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\frac{\rho_{\mathrm{f}}}{\delta t}(D_p \mathbf{u}^{n+1}, \mathbf{v}) + \mu(\nabla \mathbf{u}^{n+1}, \nabla \mathbf{v}) + \rho_{\mathrm{f}}((\mathbf{u}^{n+1} - \mathbf{w}^{n+1}) \cdot \nabla \mathbf{u}^{n+1}, \mathbf{v})$$

$$-(p^{n+1}, \nabla \cdot \mathbf{v}) = \rho_{\mathrm{f}} \langle \mathbf{f}_{\mathrm{f}}^{n+1}, \mathbf{v} \rangle \quad \forall v \in \mathscr{V}_0(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{9a}$$

$$(\nabla \cdot \mathbf{u}^{n+1}, q) = 0 \quad \forall q \in \mathscr{Q}(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{9b}$$

$$\mathbf{u}^{n+1} = \frac{1}{\delta t \gamma_p} \left(\boldsymbol{\lambda} + \sum_{i=1}^{p} \alpha_p^i \mathbf{d}^{n+1-i} \right) \quad \text{on } \Sigma_{t^{n+1}} \tag{9c}$$

Borrowing classical concepts from DD methods, we can define the *Steklov–Poincaré interface operator* (see [8]) for the fluid as follows: $\mathscr{S}_{\mathrm{f}}$ is the Dirichlet-to-Neumann map in Ω_t^{f} such that

$$\begin{aligned} \mathscr{S}_{\mathrm{f}} : H^{1/2}(\Sigma_t) &\to H^{-1/2}(\Sigma_t) \\ \boldsymbol{\lambda} &\mapsto \boldsymbol{\sigma}_n^{\mathrm{f}} \end{aligned} \tag{10}$$

This operator consists of solving the fluid problem given a value for the interface variable $\boldsymbol{\lambda}$, that is $\mathsf{FL}_{\delta t}(\boldsymbol{\lambda})$, and recover the normal stress on the interface $\boldsymbol{\sigma}_n^{\mathrm{f}}$. Thus, this is a mapping between the trace of the displacement field $\mathbf{d}$ and the space of normal stresses exerted by the fluid. Obviously, this operator depends on the fluid solver used, $\mathsf{FL}_{\delta t}$.

We point out that the Steklov–Poincaré operator $\mathscr{S}_{\mathrm{f}}$ for the fluid is non-linear. It involves two different non-linearities: one associated to the convective term of the Navier–Stokes equations and a second one due to the fact that the fluid domain $\Omega_t^{\mathrm{f}} \equiv \Omega_t^{\mathrm{f}}(\boldsymbol{\lambda})$ does depend on the interface variable (shape non-linearity). This implies that the superposition of problems cannot be used and thus $\mathscr{S}_{\mathrm{f}}$ has to include also forcing terms and non-homogeneous boundary conditions.

DOI: 10.1002/nme

Analogously for the structure, we define the Steklov–Poincaré operator: $\mathscr{S}_\mathrm{s}$ is the Dirichlet-to-Neumann map in Ω_t^s such that

$$\begin{aligned} \mathscr{S}_\mathrm{s} : H^{1/2}(\Sigma_t) &\to H^{-1/2}(\Sigma_t) \\ \boldsymbol{\lambda} &\mapsto \boldsymbol{\sigma}_n^\mathrm{s} \end{aligned} \tag{11}$$

In this case, $\mathscr{S}_\mathrm{s}$ consists of solving the structure problem using $\boldsymbol{\lambda}$ as Dirichlet boundary condition for $\mathbf{d}^\mathrm{s}$ on Σ_t and extract the value of the normal stress $\boldsymbol{\sigma}_n^\mathrm{s}$ on Σ_t. Therefore, this is a mapping between the trace of the displacement field $\mathbf{d}$ and the space of normal stresses exerted by the structure. Again, this operator is non-linear even for linear constitutive equations (as the elastic case considered) because of the *shape derivative* (the deformation of the structure domain). Let us introduce also $\mathscr{S}_\mathrm{s}^{-1}$, which is the so called *Poincaré–Steklov interface operator*: $\mathscr{S}_\mathrm{s}^{-1}$ is the Neumann-to-Dirichlet map in Ω_t^s such that

$$\begin{aligned} \mathscr{S}_\mathrm{s}^{-1} : H^{-1/2}(\Sigma_t) &\to H^{1/2}(\Sigma_t) \\ \boldsymbol{\sigma}_n^\mathrm{s} &\mapsto \boldsymbol{\lambda} \end{aligned} \tag{12}$$

The operator $\mathscr{S}_\mathrm{s}^{-1}$ consists of solving the structure problem using $\boldsymbol{\sigma}_n^\mathrm{s}$ as Neumann boundary condition on Σ_t and recover $\mathbf{d}^\mathrm{s}$ on the boundary. $\mathscr{S}_\mathrm{s}^{-1}$ will be used for fixed point algorithms.

At this point the interface condition (7) that involves continuity of normal stresses on Σ_t can be easily rewritten as: find $\boldsymbol{\lambda} \in \Gamma(\Sigma_{t^{n+1}})$ such that

$$\mathscr{S}_\mathrm{f}(\boldsymbol{\lambda}) + \mathscr{S}_\mathrm{s}(\boldsymbol{\lambda}) = \mathbf{0} \tag{13}$$

Thus, using the DD approach the initial coupled problem has been reduced to an interface equation.

An alternative form of the interface equation, obtained by applying the inverse of the Steklov–Poincaré operator $\mathscr{S}_\mathrm{s}^{-1}$ in (13), reads as: find $\boldsymbol{\lambda} \in \Gamma(\Sigma_{t^{n+1}})$ such that

$$-\mathscr{S}_\mathrm{s}^{-1}(\mathscr{S}_\mathrm{f}(\boldsymbol{\lambda})) = \boldsymbol{\lambda} \tag{14}$$

This expression motivates the use of the *Dirichlet–Neumann fixed point algorithm* (see [2]). The iterative fixed point procedure can be written as: given $\boldsymbol{\lambda}^k$, with $k \geqslant 0$, find $\boldsymbol{\lambda}^{k+1}$ such that

$$-\mathscr{S}_\mathrm{s}^{-1}(\mathscr{S}_\mathrm{f}(\boldsymbol{\lambda}^k)) = \boldsymbol{\lambda}^{k+1} \tag{15}$$

where $\mathscr{S}_\mathrm{f}(\boldsymbol{\lambda})$ is associated to an appropriate semi-discrete fluid solver $\mathsf{FL}_{\delta t}(\boldsymbol{\lambda})$. The initialization $\boldsymbol{\lambda}^0$ of the iterative process is treated in Section 5. Let us explain this equation: given a value for the interface displacement $\boldsymbol{\lambda}^k$, we solve the fluid problem for this $\boldsymbol{\lambda}^k$ using $\mathsf{FL}_{\delta t}(\boldsymbol{\lambda}^k)$ and recover the normal stresses on the interface $\boldsymbol{\sigma}_n^\mathrm{f}$, that is to say, we compute $\mathscr{S}_\mathrm{f}(\boldsymbol{\lambda}^k)$. Then, we calculate the structure problem with $\boldsymbol{\sigma}_n^\mathrm{s} = \boldsymbol{\sigma}_n^\mathrm{f}$ as boundary condition on the fluid–structure interface. It gives a new value of the interface displacement, that now we call $\boldsymbol{\lambda}^{k+1}$. In this case we solve the *Neumann-to-Dirichlet* Poincaré–Steklov interface operator $-\mathscr{S}_\mathrm{s}^{-1}(\boldsymbol{\sigma}_n^\mathrm{f})$. This procedure is repeated until convergence.

Remark 2

The solution of the fluid problem $\mathsf{FL}_{\delta t}(\boldsymbol{\lambda})$ requires non-linear iterations. Thus, algorithm (15) involves the use of nested iterative loops.

DOI: 10.1002/nme

We are also interested on a linearized version of $\mathscr{S}_\mathrm{f}$. We denote by $\mathsf{FL}_{\delta t}(\mathbf{u}_*^{n+1};\boldsymbol{\lambda})$ the linearized fluid operator that differs from the non-linearized version, i.e. (9), in the fact that the convective term in the momentum equation of the fluid has been replaced by $\mathbf{u}_*^{n+1}\cdot\nabla\mathbf{u}^{n+1}$ with $\mathbf{u}_*^{n+1}$ given. We also denote by $\widetilde{\mathscr{S}}_\mathrm{f}(\mathbf{u}_*^{n+1})$ the linearization of $\mathscr{S}_\mathrm{f}$ *around the point* $\mathbf{u}_*^{n+1}$, that is, involving the solution of the linearized fluid problem with $\mathsf{FL}_{\delta t}(\mathbf{u}_*^{n+1};\boldsymbol{\lambda})$. In the next section we suggest the use of the *semi*-linear interface operator in some cases. We stress the fact that $\widetilde{\mathscr{S}}_\mathrm{f}(\mathbf{u}_*^{n+1})$ is non-linear due to the *shape derivative*.

A different version of the fixed point algorithm (15) is obtained when using the *semi-linearized* version of the interface operator $\mathscr{S}_\mathrm{f}$ for the fluid. In this case the fixed point algorithm reads as follows: given $\boldsymbol{\lambda}^k$ and $\mathbf{u}^{n+1,k}$ with $k>0$, compute $\boldsymbol{\lambda}^{k+1}$ by

$$-\mathscr{S}_\mathrm{s}^{-1}(\widetilde{\mathscr{S}}_\mathrm{f}(\mathbf{u}^{n+1,k};\boldsymbol{\lambda}^k))=\boldsymbol{\lambda}^{k+1} \tag{16}$$

and obtain $\mathbf{u}^{n+1,k+1}$ from $\mathsf{FL}_{\delta t}(\mathbf{u}^{n+1,k};\boldsymbol{\lambda}^k)$. The procedure is repeated until a selected norm of $\mathbf{u}^{n+1,k+1}-\mathbf{u}^{n+1,k}$ and $\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k$ is below a threshold tolerance.

Remark 3
When using the algorithm (16) the same loop deals with the coupling of the fluid and structure systems and the non-linearity of the fluid equations.

Remark 4
The *semi-linearized* fixed point algorithm (16) involves the domain update at each iteration. This situation can be relaxed by using some criterion over $(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k)$ in order to decide to update or to freeze the domain at the current iteration (that is to say, to neglect or not the shape derivative). We can also consider the explicit treatment of the domain, evaluating only one domain movement per time step. This semi-implicit procedure is very appealing because it reduces the computational cost without compromising stability (see [18]). Alternatively, instead of freezing the domain, we can use a transpiration method (cheaper than the movement of the domain), as suggested in [19], in order to accelerate the iterative process.

Alternative forms of the interface equation (13) motivate different iterative algorithms for the coupling (see [5, 16]). Besides the iterative algorithm for the coupling, a relaxation method is advisable in order to improve the convergence properties of all the previous algorithms. The Aitken acceleration method is the most widely used. Different values of the optimal relaxation parameter when using the Aitken technique for the vectorial case have been proposed by Deparis [7].

5. THE DISCRETE PROBLEM

This section is devoted to the fully discretized version of the coupling problem. We are focused on the discretization of the fluid. Three different sorts of methods are considered: monolithic, pressure-correction and predictor–corrector. Every method is introduced and stated. In the applications we consider the stabilized versions of these schemes using orthogonal subgrid scales. However, for the sake of clarity, we omit the stabilization terms in the formulation. We refer the reader to a set of articles that deal with stabilized pressure segregation methods [12, 13, 20, 21]. The use of a stabilized space discretization allows us to use the same low-order finite element space for the interpolation of velocity and pressure. After the exposition of the alternative methods for the

DOI: 10.1002/nme

fluid problem, we state the discrete extension operator used for the calculation of the fluid domain movement. Finally, we suggest some coupling procedures taking into account the fluid solver used. These procedures stated are being used in Section 6.

5.1. The discrete fluid problem

The fully discretized version of the monolithic scheme (9) reads as follows: for $n=0, 1, 2, \ldots$, given $\boldsymbol{\lambda}_h \in \Gamma_h(\Sigma_{t^{n+1}})$ (understood as the displacement on the solid boundary at time step n), find $\mathbf{u}_h^{n+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p_h^{n+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\frac{\rho_{\mathrm{f}}}{\delta t}(D_p\mathbf{u}_h^{n+1}, \mathbf{v}_h) + \mu(\nabla\mathbf{u}_h^{n+1}, \nabla\mathbf{v}_h) + \rho_{\mathrm{f}}((\mathbf{u}_h^{n+1} - \mathbf{w}^{n+1})\cdot\nabla\mathbf{u}_h^{n+1}, \mathbf{v}_h)$$

$$-(p_h^{n+1}, \nabla\cdot\mathbf{v}_h) = \rho_{\mathrm{f}}\langle\mathbf{f}_{\mathrm{f}}, \mathbf{v}_h\rangle \quad \forall\mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega^{\mathrm{f}}_{t^{n+1}})$$

$$(\nabla\cdot\mathbf{u}_h^{n+1}, q_h) = 0 \quad \forall q_h \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$$

$$\mathbf{u}_h^{n+1} = \frac{1}{\delta t\gamma_p}\left(\boldsymbol{\lambda}_h + \sum_{i=1}^{p}\alpha_p^i\mathbf{d}_h^{n+1-i}\right) \quad \text{on } \Sigma_{t^{n+1}}$$

where $\Gamma_h(\Sigma_{t^{n+1}})$, $\mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $\mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ are classical finite element approximation spaces of the functional spaces $\Gamma(\Sigma_{t^{n+1}})$, $\mathscr{V}(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $\mathscr{Q}(\Omega^{\mathrm{f}}_{t^{n+1}})$, respectively.

A substantial reduction of the computational cost can be gained by using splitting techniques. These techniques allow the uncoupled computation of velocity components and pressure. Herein, we consider a pressure correction method. An up-to-date overview of these methods can be found in [22]. The classical approach is to motivate the splitting at the continuous level, as it was initially suggested by Chorin and Temam. Herein, we consider a pressure-correction method obtained at the discrete level (see [20]). Using the discrete approach good Dirichlet boundary conditions can be enforced to the end-of-step velocity whereas only the value of its normal component can be imposed when considering the splitting at the continuous level. This feature is of special interest in fluid–structure applications, where the continuity of velocities must be imposed on the interface. This scheme consists of: given $\boldsymbol{\lambda}_h \in \Gamma_h(\Sigma_{t^{n+1}})$, find $\mathbf{u}_h^{n+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p_h^{n+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ from the following scheme:

1. Find $\hat{\mathbf{u}}_h^{n+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\frac{\rho_{\mathrm{f}}}{\gamma_p\delta t}\left(\hat{\mathbf{u}}_h^{n+1} - \sum_{i=1}^{p}\alpha_p^i\mathbf{u}_h^{n+1-i}\mathbf{v}_h\right) + \mu(\nabla\hat{\mathbf{u}}_h^{n+1}, \nabla\mathbf{v}_h) + \rho_{\mathrm{f}}((\hat{\mathbf{u}}_h^{n+1} - \mathbf{w}^{n+1})\cdot\nabla\hat{\mathbf{u}}_h^{n+1}, \mathbf{v}_h)$$

$$-(\widetilde{p}_h^{n+1}, \nabla\cdot\mathbf{v}_h) = \rho_{\mathrm{f}}\langle\mathbf{f}_{\mathrm{f}}^{n+1}, \mathbf{v}_h\rangle \quad \forall\mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{17a}$$

$$\hat{\mathbf{u}}_h^{n+1} = \frac{1}{\delta t\gamma_p}\left(\boldsymbol{\lambda}_h - \sum_{i=1}^{p}\alpha_p^i\mathbf{d}_h^{n+1-i}\right) \quad \text{on } \Sigma_{t^{n+1}} \tag{17b}$$

2. Find $\mathbf{u}_h^{n+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p_h^{n+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\frac{\rho_{\mathrm{f}}}{\delta t\gamma_p}(\mathbf{u}_h^{n+1} - \hat{\mathbf{u}}_h^{n+1}, \mathbf{v}_h) - (p_h^{n+1} - \widetilde{p}_h^{n+1}, \nabla\cdot\mathbf{v}_h) = 0 \quad \forall\mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{18a}$$

DOI: 10.1002/nme

$$(\nabla \cdot \mathbf{u}_h^{n+1}, q_h) = 0 \quad \forall q_h \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}}), \tag{18b}$$

$$\mathbf{u}_h^{n+1} = \frac{1}{\delta t \gamma_p}\left(\boldsymbol{\lambda}_h + \sum_{i=1}^{p} \alpha_p^i \mathbf{d}_h^{n+1-i}\right) \quad \text{on } \Sigma_{t^{n+1}} \tag{18c}$$

In order to reduce the computational cost, we can split the pressure and end-of-step velocity in (18), using an equivalent version involving a *discrete pressure Poisson equation*:

2.1. Find $p_h^{n+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$-\gamma_p \delta t(\Pi_h(\nabla p_h^{n+1} - \nabla \widetilde{p}_h^{n+1}), \nabla q_h) = \rho_{\mathrm{f}}(\hat{\mathbf{u}}_h^{n+1}, \nabla q_h) \quad \forall q_h \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{19}$$

2.2. Find $\mathbf{u}_h^{n+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\frac{\rho_{\mathrm{f}}}{\delta t \gamma_p}(\mathbf{u}_h^{n+1} - \hat{\mathbf{u}}_h^{n+1}, \mathbf{v}_h) - (p_h^{n+1} - \widetilde{p}_h^{n+1}, \nabla \cdot \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{20a}$$

$$\mathbf{u}_h^{n+1} = \frac{1}{\delta t \gamma_p}\left(\boldsymbol{\lambda}_h + \sum_{i=1}^{p} \alpha_p^i \mathbf{d}_h^{n+1-i}\right) \quad \text{on } \Sigma_{t^{n+1}} \tag{20b}$$

In step 2, $\widetilde{p}_h^{n+1}$ is *an appropriate approximation to* p_h^{n+1} and Π_h is the L^2 projection onto the velocity space. We consider an *incremental* fractional step method when $\widetilde{p}_h^{n+1} = p_h^n$. This method has an splitting error of order $\mathscr{O}(\delta t^2)$. The results are much better than for *total* projection methods, where $\widetilde{p}_h^{n+1} = 0$, without extra computational cost. The system matrix associated to (19) is cumbersome, but can be tackled when using an iterative solver, case in which only matrix–vector products are needed. Furthermore, the use of a closed integration rule for approximating the Gramm (mass) matrix that appears in (19) reduces considerably the computational cost. Anyway, (19) can be approximated by using a more classical Laplacian discretization (see [20]), leading to the following pressure equation: find $p_h^{n+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$-\gamma_p \delta t(\nabla p_h^{n+1} - \nabla \widetilde{p}_h^{n+1}, \nabla q_h) = \rho_{\mathrm{f}}(\hat{\mathbf{u}}_h^{n+1}, \nabla q_h) \quad \forall q_h \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{21}$$

Remark 5

These pressure segregation methods obtained at the discrete level introduce the same artificial boundary condition that we find when we do the splitting at the continuous level (see Figure 1), that is, $\partial p^{n+1}/\partial n = 0$ on the Dirichlet boundary of the velocity. This misbehaviour is particularly important on fluid–structure simulations, due to the fact that the fluid–structure interface is a Dirichlet boundary. Thus, an artificial boundary condition over the pressure is imposed on the interface, where the pressure is integrated for the calculation of the stresses exerted by the fluid. We defend the use of predictor–corrector schemes introduced below in order to improve the pressure accuracy over the interface.

When we use an iterative implicit procedure for the coupling, the fluid problem is evaluated (at least) as many times as coupling iterations. Thus, it is natural to put in the momentum equation $\widetilde{p}_h^{n+1} = p_h^{n+1,k}$, $p_h^{n+1,k}$ the pressure having been obtained at the previous iteration. In fact, if the resulting scheme converges, *the end-of-step velocity* $\mathbf{u}_h^{n+1}$ *converges to the intermediate velocity* $\hat{\mathbf{u}}_h^{n+1}$. Furthermore, $\mathbf{u}_h^{n+1}$ *converges to the solution of the monolithic fluid system*. Thus, we do

DOI: 10.1002/nme

not need to distinguish between $\hat{\mathbf{u}}_h^{n+1}$ and $\mathbf{u}_h^{n+1}$ and (18) can be ignored. The final system to be solved at every coupling iteration is the following: given $\boldsymbol{\lambda}_h^k \in \Gamma_h(\Sigma_{t^{n+1}})$ and $p_h^{n+1,k} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$, find $\mathbf{u}_h^{n+1,k+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p_h^{n+1,k+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\frac{\rho_{\mathrm{f}}}{\delta t}(D_p\mathbf{u}_h^{n+1,k+1}, \mathbf{v}_h) + \mu(\nabla\mathbf{u}_h^{n+1,k+1}, \nabla\mathbf{v}_h) + \rho_{\mathrm{f}}((\mathbf{u}_h^{n+1,k+1} - \mathbf{w}^{n+1}) \cdot \nabla\mathbf{u}_h^{n+1,k+1}, \mathbf{v}_h)$$
$$-(p_h^{n+1,k}, \nabla \cdot \mathbf{v}_h) = \rho_{\mathrm{f}}\langle \mathbf{f}_{\mathrm{s}}, \mathbf{v}_h\rangle \quad \forall \mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{22a}$$

$$-\gamma_p \delta t(\Pi_h(\nabla p_h^{n+1,k+1} - \nabla p_h^{n+1,k}), \nabla q_h) = \rho_{\mathrm{f}}(\mathbf{u}_h^{n+1,k+1}, \nabla q_h) \quad \forall q_h \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{22b}$$

$$\mathbf{u}_h^{n+1,k+1} = \frac{1}{\delta t \gamma_p}\left(\boldsymbol{\lambda}_h + \sum_{i=1}^{p} \alpha_p^i \mathbf{d}_h^{n+1-i}\right) \quad \text{on } \Sigma_{t^{n+1}} \tag{22c}$$

This problem is denoted by $\mathsf{PC}_{\delta t,h}(p_h^{n+1,k}; \boldsymbol{\lambda}_h^k)$. We remark that in the case presented nested loops are needed: an internal loop to deal with the non-linearity of the convective term and an external for the convergence to the monolithic fluid system (for fluid problems) or the *monolithic* coupling system (for fluid–structure problems). Again, there is the possibility to use one loop for everything. In this case, the final system is: given $\boldsymbol{\lambda}_h^k \in \Gamma_h(\Sigma_{t^{n+1}})$, $\mathbf{u}_h^{n+1,k} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p_h^{n+1,k} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$, find $\mathbf{u}_h^{n+1,k+1} \in \mathscr{V}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ and $p_h^{n+1,k+1} \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}})$ such that

$$\rho_{\mathrm{f}}(D_p\mathbf{u}_h^{n+1,k+1}, \mathbf{v}_h) + \mu(\nabla\mathbf{u}_h^{n+1,k+1}, \nabla\mathbf{v}_h) + \rho_{\mathrm{f}}((\mathbf{u}_h^{n+1,k} - \mathbf{w}^{n+1}) \cdot \nabla\mathbf{u}_h^{n+1,k+1}, \mathbf{v}_h)$$
$$-(p_h^{n+1,k}, \nabla \cdot \mathbf{v}_h) = \rho_{\mathrm{f}}\langle \mathbf{f}_{\mathrm{s}}, \mathbf{v}_h\rangle \quad \forall \mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{23a}$$

$$-\gamma_p \delta t(\Pi_h(\nabla p_h^{n+1,k+1} - \nabla p_h^{n+1,k}), \nabla q_h) = \rho_{\mathrm{f}}(\mathbf{u}_h^{n+1,k+1}, \nabla q_h) \quad \forall q_h \in \mathscr{Q}_h(\Omega^{\mathrm{f}}_{t^{n+1}}) \tag{23b}$$

$$\mathbf{u}_h^{n+1,k+1} = \frac{1}{\delta t \gamma_p}\left(\boldsymbol{\lambda}_h + \sum_{i=1}^{p} \alpha_p^i \mathbf{d}_h^{n+1-i}\right) \quad \text{on } \Sigma_{t^{n+1}} \tag{23c}$$

In this case the fluid solver is denoted by $\mathsf{PC}_{\delta t,h}(\mathbf{u}_h^{n+1,k}, p_h^{n+1,k}; \boldsymbol{\lambda}^k)$. Methods (22) and (23) are *predictor–corrector* schemes. These methods have been introduced in [12, 23] without the fluid–structure motivation. In these references the stabilization terms omitted in the present exposition are carefully treated.

In the iterative process of predictor–corrector methods the splitting error is reduced. Let us prove this comment with a numerical test. We solve the Navier–Stokes equations over the unit square with a force term such that the analytical solution of the transient problem is

$$\mathbf{u} = (y, -x)\ \sin(\pi t/10)$$
$$p = x + y$$

The simulation is carried out using scheme (22) during 10 s. The pressure error on the boundary for $t = 10$ is plotted in 1(a), without predictor–corrector iterations. Using predictor–corrector iterations the pressure error, plotted in 1(b), is reduced, particularly on the boundary. The final error of the predictor–corrector solver depends on the tolerance for the external loop and maximum number of iterations allowed.

DOI: 10.1002/nme

5.2. The discrete fluid domain movement

As commented in the previous section, we use a harmonic extension operator on Ω_t^{f} in order to obtain $\mathbf{d}_h^{\mathrm{f}}$. The discrete problem reads as follows: given $\boldsymbol{\lambda}_h^k \in \Gamma_h(\Sigma_{t^{n+1}})$, find $(\mathbf{d}_h^{\mathrm{f}})^{n+1} \in \mathscr{V}_h(\Omega_{t^{n+1}}^{\mathrm{f}})$ such that

$$(\nabla(\mathbf{d}_h^{\mathrm{f}})^{n+1}, \nabla\mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \mathscr{V}_{h,0}(\Omega_{t^{n+1}}^{\mathrm{f}}) \tag{24a}$$

$$(\mathbf{d}_h^{\mathrm{f}})^{n+1} = \boldsymbol{\lambda}_h \tag{24b}$$

We call $(\mathbf{d}_h^{\mathrm{f}})^{n+1} = \mathrm{Ext}_h(\boldsymbol{\lambda}_h)$. The harmonic operator is applied on Ω_t^{f} because it allows to solve this problem using the same mesh that we use to compute the fluid problem.

5.3. Coupling algorithms for the discrete problem

When considering aeroelastic applications fixed point algorithms of Dirichlet–Neumann type are effective. This kind of methods is considered in the following. On the other hand, in order to reduce the computational cost of the fluid solver, pressure segregation methods have been considered. The last task is to match these ingredients in an effective way. We suggest the use of a 'one loop algorithm', where the same loop deals with the fluid solver and coupling. For every coupling iteration $k \geqslant 0$, the problem to be solved is: given $\boldsymbol{\lambda}_h^{n+1,k}$, $\mathbf{u}_h^{n+1,k}$ and $p_h^{n+1,k}$, find $\boldsymbol{\lambda}_h^{n+1,k+1}$ such that

$$\boldsymbol{\lambda}_h^{n+1,k+1} = -\mathscr{S}_{\mathrm{s}}^{-1}(\widetilde{\mathscr{S}}_{\mathrm{f}}(\mathbf{u}_h^{n+1,k}; \boldsymbol{\lambda}_h^{n+1,k})) \tag{25}$$

with $(\mathbf{u}_h^{n+1,k+1}, p_h^{n+1,k+1}) = \mathsf{PC}_{\delta t,h}(\mathbf{u}_h^{n+1,k}, p_h^{n+1,k}; \boldsymbol{\lambda}_h^{n+1,k})$. Thus, in the implicit coupling process, we have to solve (25) until convergence. In this method the same loop deals with the non-linearity of the convective term and the convergence to the monolithic system. Some other alternatives for the treatment of the iterations are possible. For instance, the use of nested loops, one for the coupling and one for the non-linearity. This case is similar to (25) but uses $\mathscr{S}_{\mathrm{f}}(\boldsymbol{\lambda}_h^{n+1,k})$ together with the fluid solver (22). However, for simplicity, we only use (25) in the numerical experimentation. Alternative versions of (25) can be tested for every application in order to identify which is faster.

Scheme (25) can also be considered in an explicit way, performing only 1 iteration per time step. Even though implicit versions have been chosen herein, explicit and semi-implicit (see Remark 4) versions can be useful for some aeroelastic applications.

In any case, the domain has to be initialized for every time step value. Different alternatives have been suggested in the literature. A first-order approximation in time is $\boldsymbol{\lambda}_h^{n+1,0} = \boldsymbol{\lambda}_h^n$. A more accurate second-order approximation that reduces the artificial energy introduced to the system is proposed in [24] for explicit procedures. Unfortunately, when using explicit procedures numerical instabilities occur much earlier with the second-order predictor (see the numerical experimentation in [3]). In this work we have adopted as initialization

$$\boldsymbol{\lambda}^{n+1,0} = -\mathscr{S}_{\mathrm{s}}^{-1}((\boldsymbol{\sigma}_n^{\mathrm{f}})^n) \tag{26}$$

that is, we solve the structure problem at t^{n+1} using as Neumann boundary condition the normal stress $(\boldsymbol{\sigma}_n^{\mathrm{f}})^n$ exerted by the fluid at the previous time step. A second-order method of this type is

$$\widetilde{\boldsymbol{\lambda}}^{n+1} = -\mathscr{S}_{\mathrm{s}}^{-1}(2(\boldsymbol{\sigma}_n^{\mathrm{f}})^n - (\boldsymbol{\sigma}_n^{\mathrm{f}})^{n-1}) \tag{27}$$

DOI: 10.1002/nme

The iterative procedure must be supplemented with an appropriate stopping criterion. Due to the fact that the loop includes predictor–corrector iterations and the fluid–structure coupling, both the convergence of the fluid unknowns and the interface displacement can be taken into account, even though there is clear dependence. The predictor–corrector convergence is evaluated using relative errors

$$\frac{\|\mathbf{u}^{n+1,k+1}-\mathbf{u}^{n+1,k}\|_0}{\|\mathbf{u}^{n+1,k}\|_0} \leqslant \mathrm{tol}_u \tag{28a}$$

$$\frac{\|p^{n+1,k+1}-p^{n+1,k}\|_0}{\|p^{n+1,k}\|_0} \leqslant \mathrm{tol}_p \tag{28b}$$

and for the coupling procedure the convergence of the interface displacement is assessed with

$$\frac{\|\boldsymbol{\lambda}^{n+1,k+1}-\boldsymbol{\lambda}^{n+1,k}\|_0}{\|\boldsymbol{\lambda}^{n+1,k}\|_0} \leqslant \mathrm{tol}_c \tag{29}$$

Remark 6
Along this section we have considered $\mathbf{w}^{n+1}$ independent of the iterative process for the sake of clarity. However, this is not the general case. How to treat this mesh velocity in the iterative algorithm has been pointed out in Remark 4.

Remark 7
The fixed point algorithms suggested in this section are not restricted to matching grids. A simple projector of the interface unknown from the structure interface mesh to the fluid one (or vice versa) would be needed for non-matching grids.

6. APPLICATIONS

6.1. Bridge aerodynamics

Among the different topologies of bridges, *suspension* bridges span the greatest distances. However, the bending moments acting on the deck sections of this sort of bridges are relatively small. Even though the span between piles is very large, the distance between cables, that in fact are working as piles, is small. For this reason these structures are flexible and light. Their very low structural damping make them sensitive to wind actions. While for other topologies the aeroelastic behaviour is not considered important, for suspension bridges it represents a key aspect of the design process.

In this work we study the flutter phenomenon. This dynamic phenomenon is induced by the fluid–structure coupling (the energy transfer). The flutter happens when the damping induced by the fluid to the structure makes the overall structure damping negative. When this aeroelastic phenomenon was not taken into account by the engineers it caused some historical failures of bridges. One of the most important criteria of design is the *flutter limit* velocity (when flutter occurs). An acceptable structure must have a large enough flutter limit velocity. A large gap between the maximum velocity of design and this limit is required.

The flutter analysis has been developed by experimentation in wind tunnels. For instance, the design of the Great Belt bridge (Denmark) involved more than 16 sections (see [25]). In wind tunnels the flutter limit is obtained evaluating the *aeroelastic derivatives*. This methodology, that

DOI: 10.1002/nme

was originated in aeronautics, was extrapolated to bridge aerodynamics by Scanlan and Tomko [26]. This experimentation process is very expensive and time consuming. Further, the Reynolds number of the real problem cannot be reproduced in conventional wind tunnels. The increasing in the capability of computers together with the improvement of numerical methods have motivated in the last decade the use of computer methods for the analysis of bridge aerodynamics.

The present application is devoted to the evaluation of flexural and torsional frequencies of the Great Belt bridge for a given inflow velocity and the direct flutter simulation using the methods introduced in the previous sections. The finite element method together with stabilized predictor–corrector and pressure-correction fluid solvers for the coupling have been used. The ALE framework has allowed to formulate the flow problem in moving domains. Second-order accurate methods (in time) have been considered. No turbulence modelling has been used, due to the fact that the bridge deck is a bluff body, the flow is detached and the influence of the turbulence effects for this case is less important than for the aeroelastic analysis of wings. However, for wide decks, the flow re-attaches at a given point. Nevertheless, we use a stabilized finite element method motivated by a multiscale approach. There is a recent trend among the computational mechanics community to claim that this kind of methods can replace conventional turbulence models (see [27]).

6.2. The bridge model

For the numerical aeroelastic analysis of bridges, the 3D problem is usually reduced to a 2D problem. In fact, this is also the usual procedure for wind tunnel tests. In order to simulate the correct natural frequencies in the fundamental symmetric flexural and torsional modes, spring stiffnesses are applied to the elastic centre of the cross-section. Lumped mass and moment of inertia on the gravity centre have been introduced to simulate the mass and moment of inertia per unit length. Furthermore, the 2D cross-section is considered to be a rigid body.

In order to obtain the equations governing the displacement of the bridge section, Newton's law is formulated on the gravity centre, and the spring force depending on the displacement of the structure is applied to the elastic centre. When the gravity centre and the elastic centre are in different positions, the resulting governing equations are non-linear. However, assuming that the rotation angle is *small*, the equations can be easily linearized. Thus, the linearized ordinary differential equation (ODE) that governs the displacement of the structure reads as follows: find the displacement vector $\mathbf{d}_s \in \mathbb{R}^3$ (for a 2D problem) that contains the translation and rotation of the structure such that

$$\mathbf{M}\ddot{\mathbf{d}}_s + \mathbf{C}\dot{\mathbf{d}}_s + \mathbf{K}\mathbf{d}_s = \mathbf{f} \tag{30}$$

where $\mathbf{M}$ is the *mass* matrix, $\mathbf{C}$ is the *damping* matrix, $\mathbf{K}$ is the *stiffness* matrix and $\mathbf{f}$ is the external force exerted over the structure (including force and moment). This external force depends on the displacement of the structure, that is, $\mathbf{f} = \mathbf{f}(\mathbf{d}_s)$, and thus the problem is non-linear. The *linearized mass* matrix has the following expression:

$$\mathbf{M} = \begin{pmatrix} m & 0 & -s_x \\ 0 & m & s_y \\ -s_x & s_y & I_\theta \end{pmatrix}$$

where m, I_θ, s_x and s_y denote the mass, inertial moment and static moments associated to the elastic centre (per unit length), respectively. The damping coefficients are usually given as a *percentage logarithmic decrement*.

DOI: 10.1002/nme

For the time integration of the ODE (31) we use the unconditional stable *constant-average-acceleration* scheme, also called *trapezoidal rule*, which is described by the following set of equations:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{d}}_{\mathrm{s}}^{n+1} + \mathbf{C}\dot{\mathbf{d}}_{\mathrm{s}}^{n+1} + \mathbf{K}\mathbf{d}_{\mathrm{s}}^{n+1} = \mathbf{f}^{n+1} \\ \mathbf{d}_{\mathrm{s}}^{n+1} = \mathbf{d}_{\mathrm{s}}^{n} + \delta t \dot{\mathbf{d}}_{\mathrm{s}}^{n} + \dfrac{\delta t^2}{4}(\ddot{\mathbf{d}}_{\mathrm{s}}^{n+1} + \ddot{\mathbf{d}}_{\mathrm{s}}^{n}) \\ \dot{\mathbf{d}}_{\mathrm{s}}^{n+1} = \dot{\mathbf{d}}_{\mathrm{s}}^{n} + \dfrac{\delta t}{2}(\ddot{\mathbf{d}}_{\mathrm{s}}^{n+1} + \ddot{\mathbf{d}}_{\mathrm{s}}^{n}) \end{cases}$$

This second-order accurate scheme is particularly appropriate for the case under consideration due to the fact that preserves the energy of the structure, given by

$$E_{\mathrm{s}} = \tfrac{1}{2}\,\dot{\mathbf{d}}_{\mathrm{s}} \cdot \mathbf{M}\dot{\mathbf{d}}_{\mathrm{s}} + \tfrac{1}{2}\,\mathbf{d}_{\mathrm{s}} \cdot \mathbf{K}\mathbf{d}_{\mathrm{s}} \tag{31}$$

which is an important feature when analysing the aeroelastic stability of the structure.

6.3. The coupling model

As justified in Section 2, we consider (25) as fluid–structure solver for the numerical simulation of bridge aeroelasticity because this scheme inherits the good properties (stability and accuracy) of implicit procedures together with a low computational cost (split fluid system and one loop for the predictor–corrector and coupling).

As it is widely known, explicit procedures can lead to undesirable numerical instability, being more instable for higher-order time approximations (see [3, 24]). Due to the fact that we want to assess the stability of the coupling problem, intimately related to the energy transfer between fluid and structure, the use of an implicit procedure that avoids these instabilities is justified. Further, the implicit procedure tends to the solution of the monolithic coupled system, eliminating the splitting error associated to staggered procedures.

Due to the complexity of external flows that appear in aeroelastic applications, and its highly transient behaviour, the use of second-order methods are worth it, and even more when no extra computational cost is introduced. We have used here the BDF-2 scheme, both for the time integration of the momentum equation and for the evaluation of the mesh velocity in the fluid domain. By doing this, and as it is proved in [11] for the convection–diffusion equation, the ALE formulation does not spoil the second-order of accuracy of the fluid solver. The movement of the fluid domain has been computed by solving the discrete problem (24).

We point out that when the structure is considered to be a rigid body, the interface equation (14) has the following *integrated* form

$$-\mathscr{S}_{\mathrm{s}}^{-1}\left(\int_{\Sigma_t} \mathscr{S}_{\mathrm{f}}(\boldsymbol{\lambda})\,\mathrm{d}\Sigma\right) = \boldsymbol{\lambda} \tag{32}$$

where $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{d}_{\mathrm{s}})$ and $\mathscr{S}_{\mathrm{s}}(\boldsymbol{\lambda})$ gives the forces and moments (not the stresses) that cause a displacement $\boldsymbol{\lambda}$. Likewise, $\mathscr{S}_{\mathrm{f}}(\boldsymbol{\lambda})$ contains not only the components of the normal stress exerted by the fluid, but also the moments per unit of area (length if $d = 2$), and therefore the integral of $\mathscr{S}_{\mathrm{f}}(\boldsymbol{\lambda})$ gives the total force and moment exerted by the fluid on Σ_t. In this case we use a fixed point iterative method to solve the non-linear interface problem (32). More precisely, the method used here is the integral version of the iteration scheme stated in (25).

DOI: 10.1002/nme

Table I. Properties of the Great Belt bridge.

Mass per unit length, m (kg/m)	2.27×10^3
Vertical static moment on elastic centre per unit length, s_y (kg m/m)	1.61×10^4
Mass moment of inertia on elastic centre per unit length, I_θ (kg m^2/m)	2.47×10^6
Vertical spring stiffness, k_y (N/m^2)	8.78×10^3
Torsional spring stiffness, k_θ (N m/m^2)	7.21×10^6
Vertical logarithmic damping, l_y (%)	1
Torsional logarithmic damping, l_θ (%)	0.6

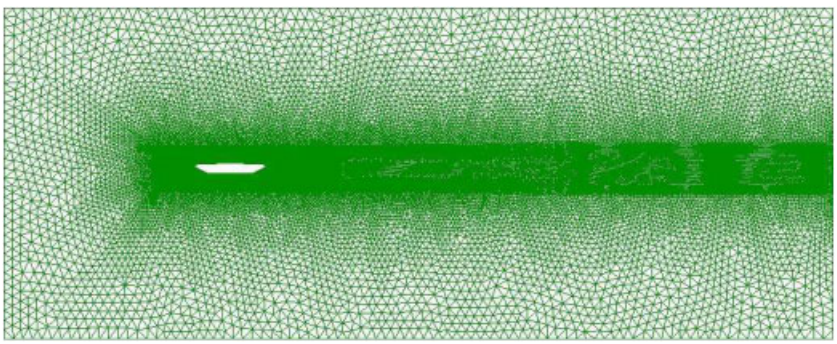

Figure 2. Space domain of analysis and mesh used for the simulation.

Even though fixed point algorithms have a good convergence for aeroelastic applications, we have used the Aitken acceleration technique for scalar equations. We define the residual of the interface equation as

$$\mathbf{r}(\boldsymbol{\lambda}^k) = -\mathscr{S}_\mathrm{s}^{-1}\left(\int_{\Sigma_t} \mathscr{S}_\mathrm{f}(\boldsymbol{\lambda}^k)\,\mathrm{d}\Sigma\right) - \boldsymbol{\lambda}^k$$

Exploiting the fact that the structure is considered to be a rigid body, the relaxation parameter can be obtained from the expression for scalar equations. In this case, we consider the *diagonal relaxation matrix*

$$\boldsymbol{\omega}^k = \begin{pmatrix} \omega_x^k & 0 & 0 \\ 0 & \omega_y^k & 0 \\ 0 & 0 & \omega_\theta^k \end{pmatrix}$$

that verifies

$$\boldsymbol{\omega}^k(\mathbf{r}(\boldsymbol{\lambda}^k) - \mathbf{r}(\boldsymbol{\lambda}^{k-1})) = \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}$$

DOI: 10.1002/nme

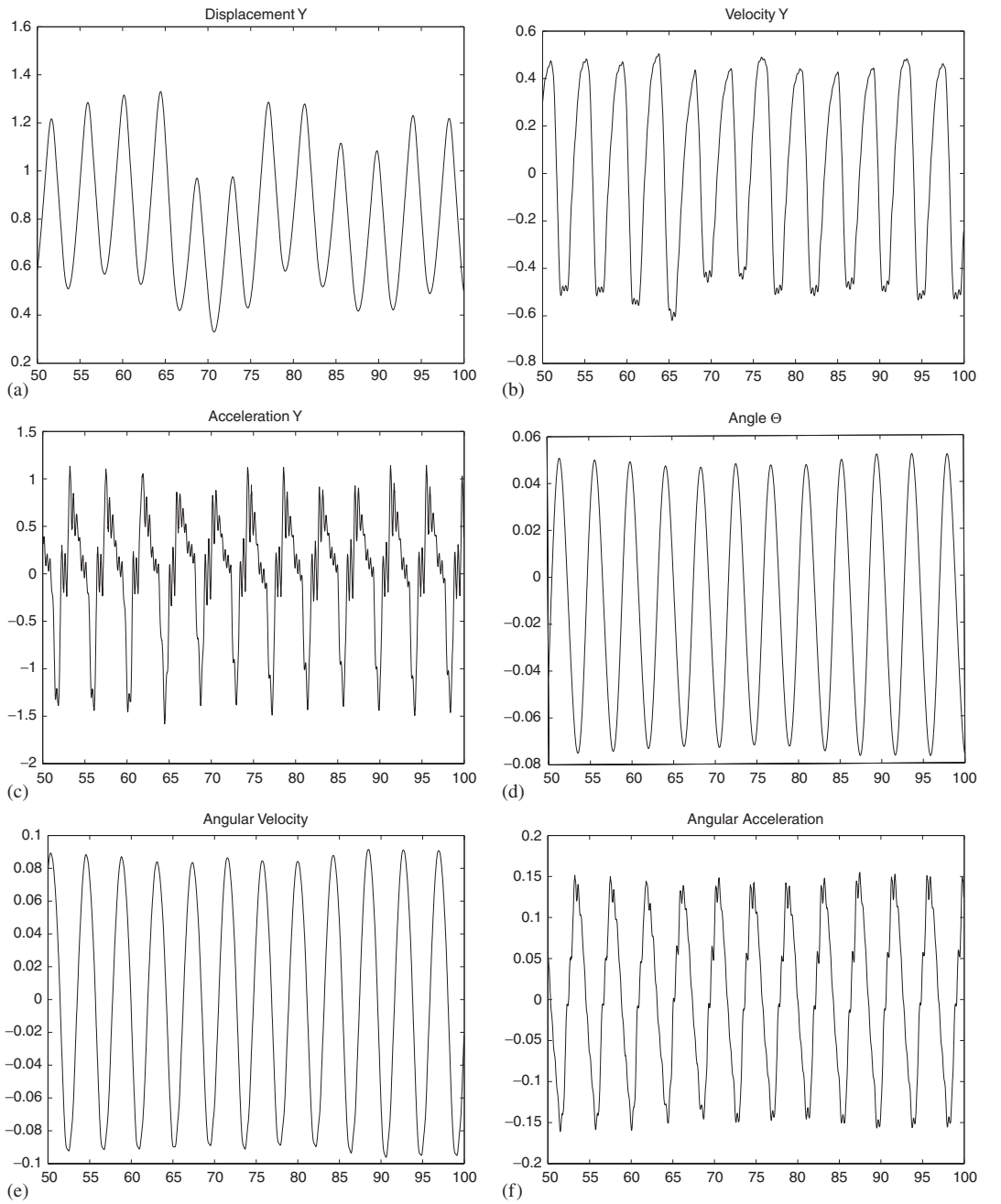

Figure 3. Movement of the bridge for an inflow velocity $\mathbf{u}_{\text{in}} = (50, 0)$ m/s: (a) vertical displacement *vs* time; and (b) vertical velocity *vs* time; (c) vertical acceleration *vs* time; (d) rotation angle *vs* time; (e) angular velocity *vs* time; and (f) angular acceleration *vs* time.

DOI: 10.1002/nme

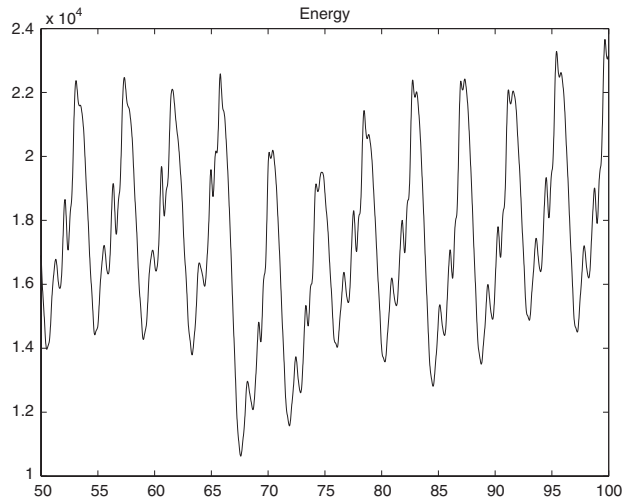

Figure 4. Bridge energy *vs* time for inflow velocity $\mathbf{u}_{\text{in}} = (50, 0)\,\text{m/s}$.

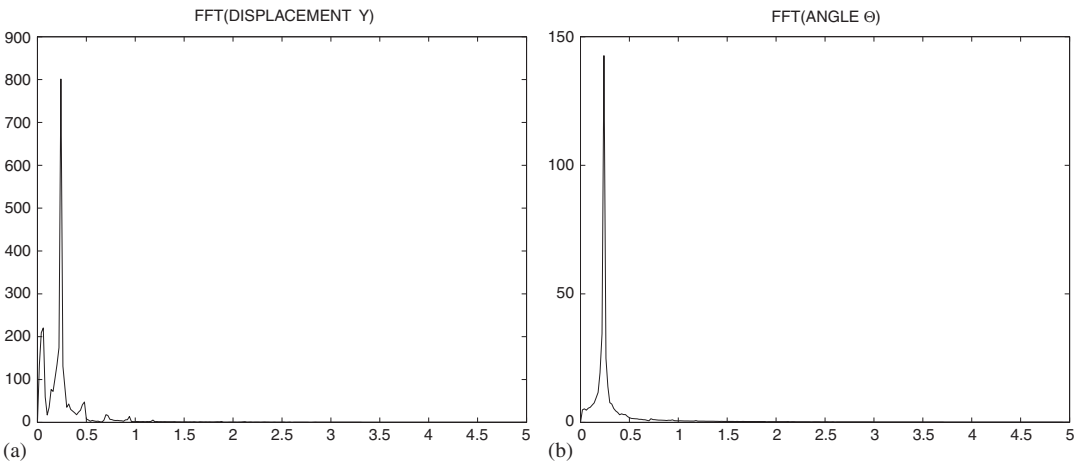

Figure 5. Fourier transform of vertical displacement and rotation angle of the bridge for inflow velocity $\mathbf{u}_{\text{in}} = (50, 0)\,\text{m/s}$: (a) vertical displacement: amplitude *vs* frequency and (b) rotation angle: amplitude *vs* frequency.

The *relaxed* version of a fixed point iteration applied to (32) is

$$\boldsymbol{\lambda}^{k+1} = -\boldsymbol{\omega}^k \mathscr{S}_{\text{s}}^{-1} \left(\int_{\Sigma_t} \mathscr{S}_{\text{f}}(\boldsymbol{\lambda}^k)\, \text{d}\boldsymbol{\Sigma} \right) + (\mathbf{I} - \boldsymbol{\omega}^k)\boldsymbol{\lambda}^k$$

A deep study of relaxation methods in a fluid–structure framework can be found in [7].

The stopping criterion (28)–(29) has been used with a tight tolerance for the fluid–structure coupling and looser for the pressure convergence. The aim is to reach the implicit fluid–structure configuration without too many iterations and unnecessary pressure accuracy.

DOI: 10.1002/nme

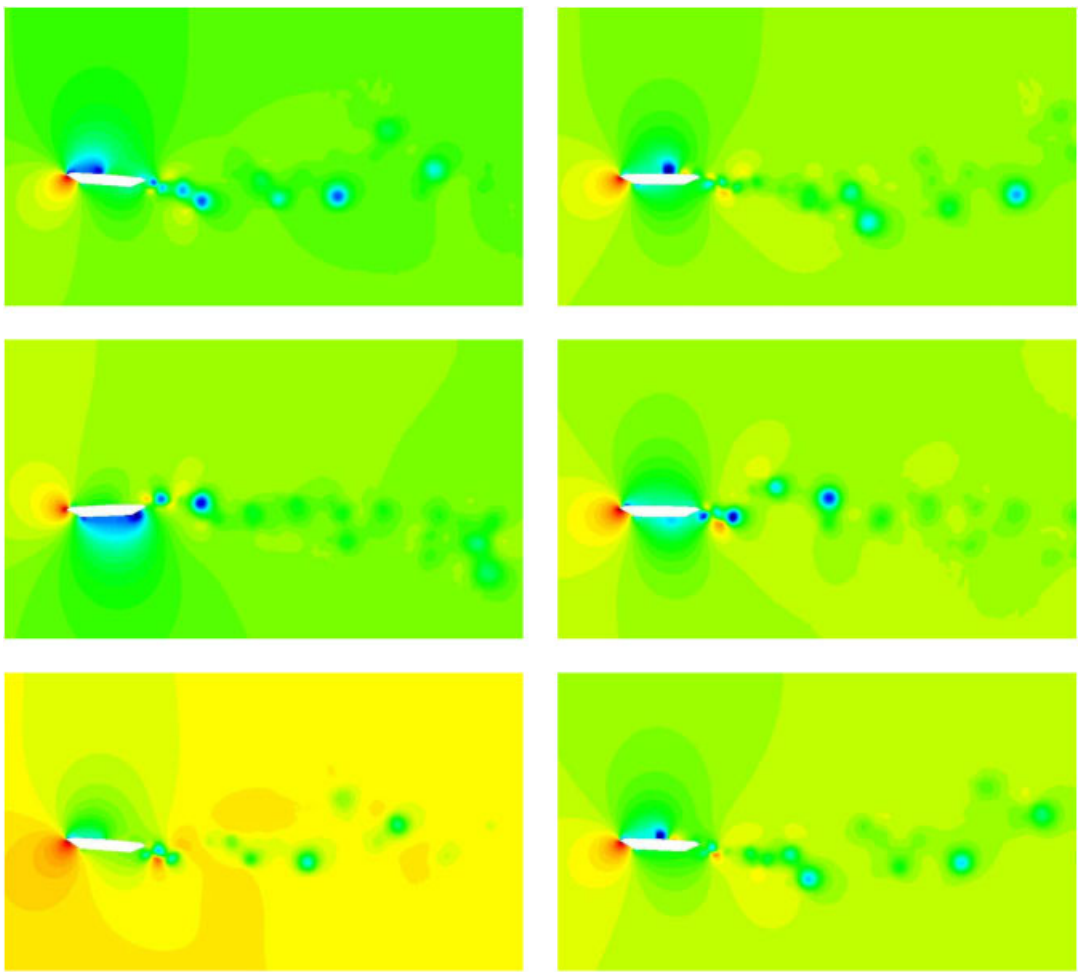

Figure 6. Contours of the pressure at different time steps (increasing time from left to right and from top to bottom) for inflow velocity $\mathbf{u}_{\text{in}} = (50, 0)\,\text{m/s}$.

6.4. Assessment of frequencies and direct flutter simulation

This section is devoted to the numerical simulation of the flutter limit and the assessment of frequencies of the Great Belt bridge (Denmark). The parameters that define the problem have been summarized in Table I and have been extracted from [28]. The problem domain and its finite element discretization is shown in Figure 2. We have used an unstructured mesh of 48 453 linear triangles for this simulation. A time step size of 0.01 s has been considered. The horizontal movement is restricted, as it is usually assumed. We do not know which are the appropriate elastic coefficients when analysing the real sized problem with the real inflow velocity. For this reason we have assumed the elastic coefficients used for the dimensionless approximation analysed by Selvam and Govindaswamy [28]. It has to be taken into account that this assumption affects the obtained results and complicates the comparison to wind tunnel experiments.

DOI: 10.1002/nme

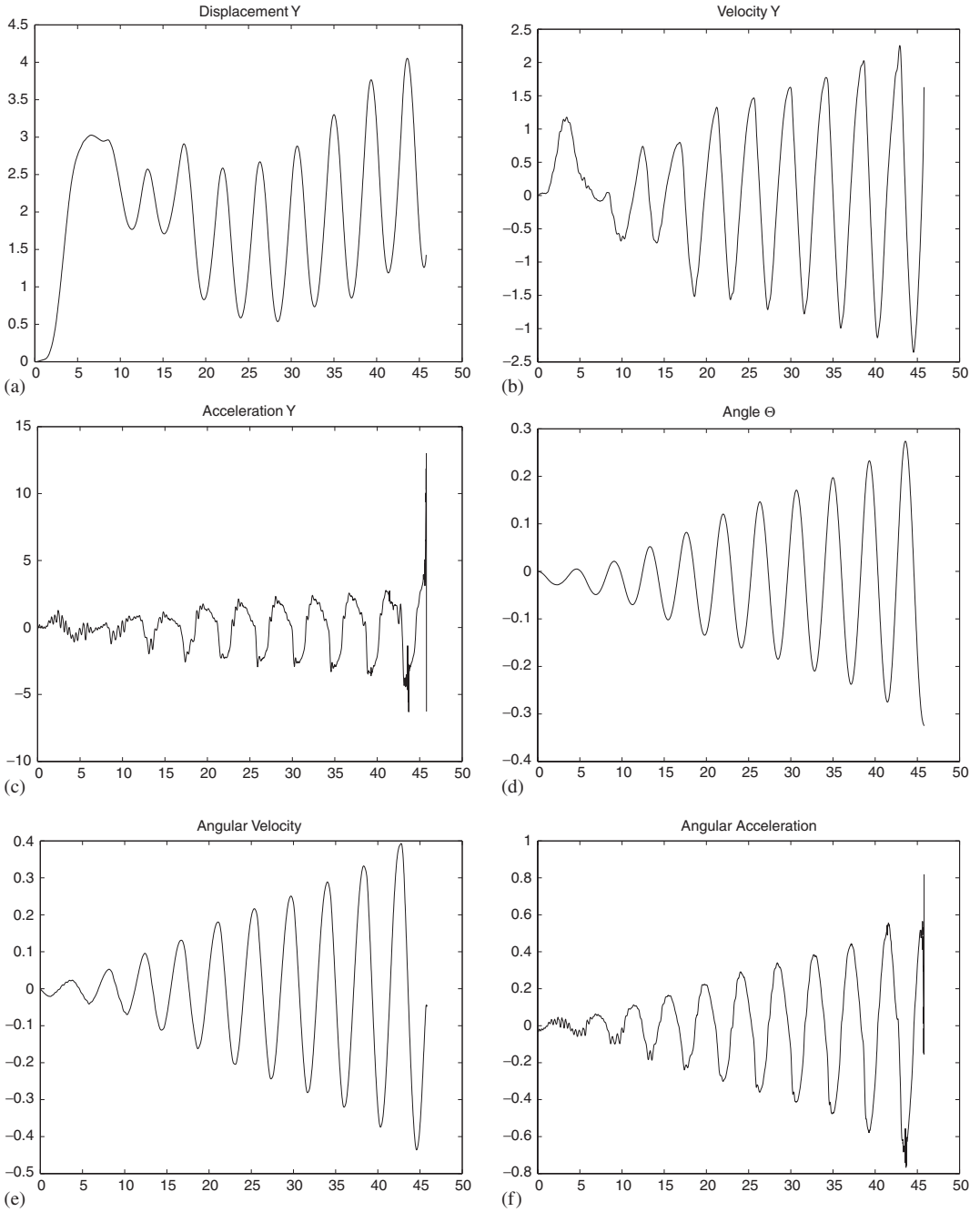

Figure 7. Movement of the bridge for inflow velocity $\mathbf{u}_{\text{in}} = (55, 0)\,\text{m/s}$: (a) vertical displacement *vs* time; (b) vertical velocity *vs* time; (c) vertical acceleration *vs* time; (d) rotation angle *vs* time; (e) angular velocity *vs* time; and (f) angular acceleration *vs* time.

DOI: 10.1002/nme

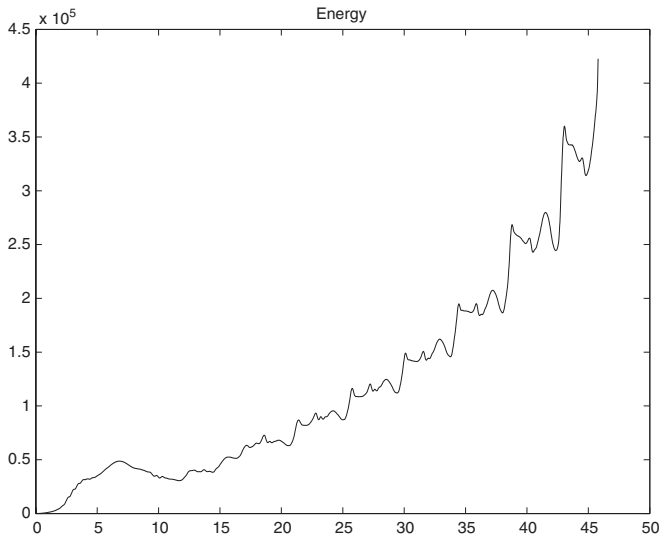

Figure 8. Bridge energy *vs* time for inflow velocity $\mathbf{u}_{\text{in}} = (55, 0)\,\text{m/s}$.

Given an inflow velocity of $\mathbf{u}_{\text{in}} = (50, 0)\,\text{m/s}$, we obtain the temporary response of the bridge. The Reynolds number in this situation is over 10^7. In Figures 3(a), (b) and (c) we show the vertical displacement, velocity and acceleration. Figures 3(d), (e) and (f) show the rotation angle, angular velocity and angular acceleration. We plot the results after some time of computation. In Figure 4 we plot the energy of the structure, defined in (31). These plots prove the stability of the structure.

Using a *Fourier fast transform* we have obtained the frequencies associated to the vertical displacement (flexural frequency) and rotation angle (torsional frequency). We show these results in Figures 5(a) and (b). In both cases a clear dominant frequency governs the movement.

We show pressure contours at different time steps in Figures 6.

The average number of iterations needed for the convergence of the integral version of method (25) to the monolithic system for a given time step is around *4 iterations per time step* for an inflow velocity of 50 m/s.

In a second step, we increase the inflow velocity until we reach the aeroelastic instability. The flutter phenomenon appears for an inflow velocity of 55 m/s. We plot the same values as before in Figures 7 and 8. We easily see in this case that the flutter instability appears for this velocity. In fact, the instability is translational and torsional (see Figures 7(a) and (d)). We plot velocities and accelerations for vertical displacement and rotation angle in Figures 7(b) and (c) and 7(e) and (f). The aerodynamic instability is clearly shown from the increase of the structure energy (Figure 8).

Obviously, the number of iterations needed for the inflow velocity of 55 m/s increases with the structure energy. We end this section with the plots of pressure at different time steps in Figure 9.

6.5. Aeroelastic derivatives using numerical experimentation

A different approach to the direct flutter simulation is the calculation of the aeroelastic derivatives. This is the procedure when using wind tunnel tests. We refer to [26] for an introduction to this methodology.

DOI: 10.1002/nme

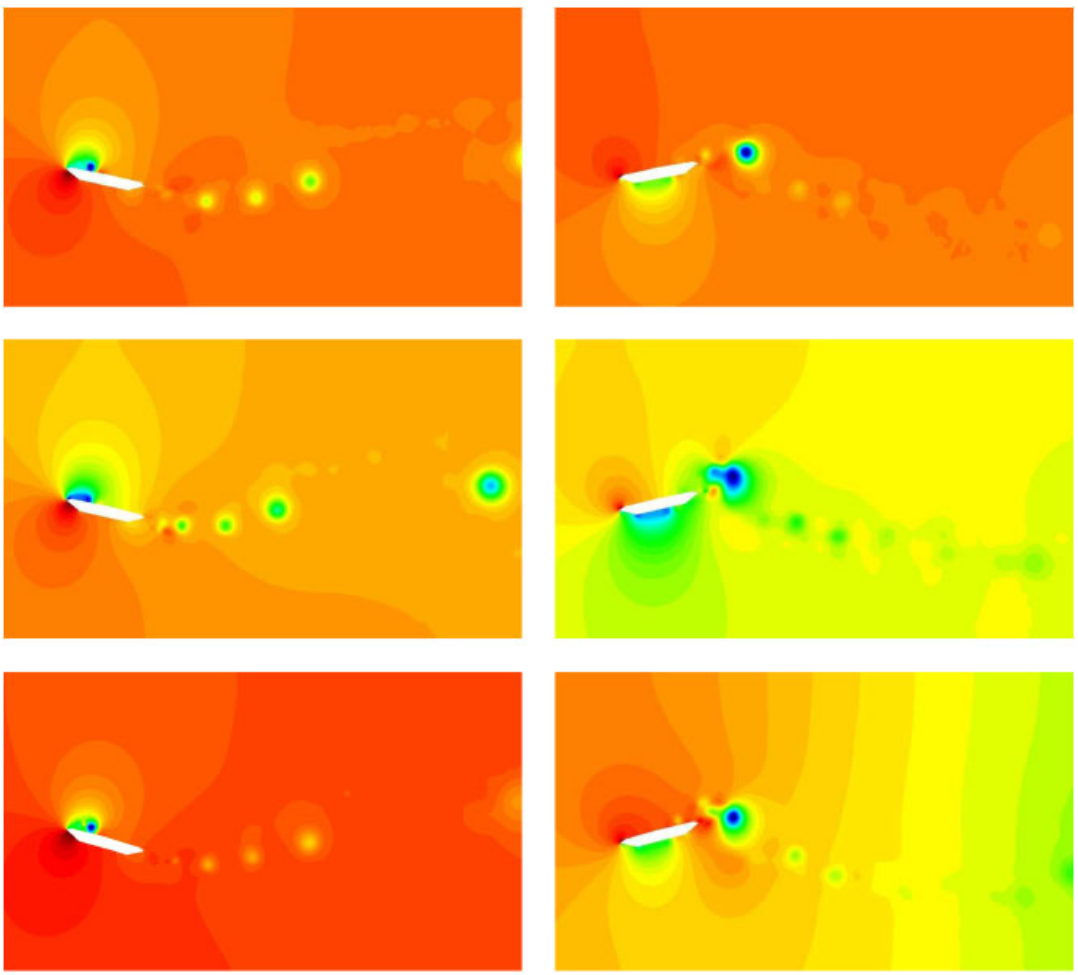

Figure 9. Contours of the pressure at different time steps (increasing time from left to right and from top to bottom) for inflow velocity $\mathbf{u}_{\text{in}} = (55, 0)\,\text{m/s}$.

Rossi [29] has used the same fluid–structure code for the assessment of aeroelastic derivatives, reproducing wind tunnel tests. The forced-vibration and free-vibration methods have been used. For the first one, the bridge movement is fixed, and therefore no fluid–structure coupling procedures apply, simplifying the problem to the solution of the fluid system using the pressure segregation method (17)–(21)–(20). For the free-vibration method scheme (25) has been applied in an explicit fashion (only 1 iteration per time step) for the assessment of aeroelastic derivatives. In both cases, the results obtained are in agreement with the wind tunnel experiments, as expected, because the wind tunnel test is being simulated, the comparison is fair. Furthermore, the problem is less challenging, because it involves adimensionalized problems with lower Reynolds numbers. Finally, no elastic coefficients are needed *a priori*.

DOI: 10.1002/nme

7. CONCLUSIONS

The coupling methods proposed herein for the calculation of the flutter limit have shown a sound behaviour in aeroelastic applications. The *coupling method proposed converges to the monolithic problem*. That is, the coupling process does not introduce any extra error (apart from the tolerance of the stopping criterion). Furthermore, this method shows a good convergence behaviour for this kind of problems and high-order (in time) procedures can be designed without compromising stability.

The other key point is the fact that *the present methods uncouple the velocity and pressure computation*, that implies a high reduction of the computational cost of the fluid problem, the bottle neck of aeroelastic simulations, making implicit procedures more affordable.

We have considered in the applications a second-order stabilized pressure segregation method (predictor–corrector versions) together with a second-order ALE formulation, a second-order structure solver and a coupling iterative procedure that tends to the monolithic system. Thus, *the overall fluid–structure coupling procedure* used *is second-order accurate in time*, an important property for highly transient external flows that appear in aeroelastic applications.

We have applied this methods to the aeroelastic analysis of a bridge deck. The flutter velocity of 55 m/s has been obtained by direct simulation, increasing the inflow velocity until the instability appears. This velocity differs from the 65–70 m/s obtained using the aeroelastic derivatives theory assessed with wind tunnel tests. However, this gap could be expected, since the problems solved are very different. It seems that the elastic coefficient that should be used for the direct analysis of flutter in dimensional form has to be higher than the one used for the scaled problem.

In fact, reproducing the wind tunnel tests using the same fluid–structure code, the numerical results are in exceptional agreement with the wind tunnel results when assessing the aeroelastic derivatives, as reported in [29].

The fluid–structure schemes suggested in the present work are fixed point type algorithms that work well for aeroelastic applications. For hemodynamics applications, where the added mass effect is critical, these kind of methods do not work well. Splitting procedures that are well suited to this kind of applications and subsequent coupling procedures that tend to the monolithic solution are being explored. Promising results have already been presented in [30].

REFERENCES

1. Farhat C, Geuzaine P, Brown G. Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers and Fluids* 2003; **32**:3–29.
2. Cervera M, Codina R, Galindo M. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engineering Computations* 1996; **13**(6):4–30.
3. Mok DP, Wall WA. Partitioned analysis schemes for transient interaction of incompressible flows and nonlinear flexible structures. In *Trends in Computational Structural Mechanics*, Wall WA, Bletzinger KU, Schweizerhof K (eds). CIMNE: Barcelona, Spain, 2001.
4. Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer Methods in Applied Mechanics and Engineering* 2001; **194**(42–44):4506–4527.
5. Fernández MA, Moubachir M. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers and Structures* 2005; **83**(2–3):127–142.
6. Gerbeau JF, Vidrascu M. Quasi-Newton algorithm based on a reduced model for fluid–structure interaction problems in blood flows. *Mathematical Modelling and Numerical Analysis* 2003; **37**(4):631–647.
7. Deparis S. Numerical analysis of axisymmetric flows and methods for fluid–structure interaction arising in blood flow simulation. *Ph.D Thesis*, École Polytechnique Fédérale de Lausanne, 2004.

DOI: 10.1002/nme

8. Quarteroni A, Valli A. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science: Oxford, 1999.
9. Formaggia L Nobile F. A stability analysis for the Arbitrary Lagrangian Eulerian formulation with finite elements. *East-West Journal of Numerical Mathematics* 1999; **7**:105–132.
10. Boffi D, Gastaldi L. Stability and geometric conservation laws for ALE formulation. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:4717–4739.
11. Badia S, Codina R. Analysis of a stabilized finite element approximation of the transient convection–diffusion equation using an ALE framework. *SIAM Journal on Numerical Analysis* 2006; **44**(5):2159–2197.
12. Codina R, Soto O. Approximation of the incompressible Navier–Stokes equations using orthogonal-subscale stabilization and pressure segregation on anisotropic finite element meshes. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1403–1419.
13. Codina R, Badia S. On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2900–2918.
14. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
15. Le Tallec P, Mouro J. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3039–3067.
16. Deparis S, Discacciati M, Quarteroni A. A domain decomposition framework for fluid–structure interaction problems. *Proceedings of ICCFD3*, Toronto, 2004.
17. Hundsdorfer W, Verwer JG. *Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations*. Springer: Berlin, 2003.
18. Nobile F. Numerical approximation of fluid–structure interaction problems with application to haemodynamics. *Ph.D Thesis*, École Polytechnique Fédérale de Lausanne, 2001.
19. Deparis S, Fernández MA, Formaggia L. Acceleration of a fixed point algorithm for fluid–structure interaction using transpiration conditions. *Mathematical Modelling and Numerical Analysis* 2003; **37**(4):601–616.
20. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112–140.
21. Codina R, Badia S. Second order fractional step schemes for the incompressible Navier–Stokes equations. Inherent pressure stability and pressure stabilization. *Proceedings of WCCM VI*, Beijing, China, 2004.
22. Guermond JL, Minev P, Shen J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:6011–6045.
23. Codina R, Folch A. A stabilized finite element predictor–corrector scheme for the incompressible Navier–Stokes equations using a nodal based implementation. *International Journal for Numerical Methods in Fluids* 2004; **44**:483–503.
24. Piperno S, Farhat C. Partitioned procedures for the transient solution of coupled aeroelastic problems—Part II: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3147–3170.
25. Larsen A, Jacobsen AS. Aerodynamic design of the great belt east bridge. In *Aerodynamics of Large Bridges*, Larsen A (ed.). Balkema: Rotterdam, Netherlands.
26. Scanlan RH, Tomko JJ. Airfoil and bridge deck flutter derivatives. *Journal of Engineering Mechanics Division ASCE 97* (*EM6*), 1971; 1717–1737.
27. Hughes TJR, Mazzei L, Jansen KE. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science* 2000; **3**:47–59.
28. Selvam RP, Govindaswamy S. Aeroelastic analysis of bridge girder section using computer modelling. *Technical Report*, University of Arkansas, 2001.
29. Rossi R. Light weight structures: structural analysis and coupling issues. *Ph.D Thesis*, Universitá di Bologna, 2005.
30. Badia S, Quaini A, Quarteroni A. Splitting methods based on algebraic factorization for fluid structure interaction. *SIAM Journal on Scientific Computing*, submitted.

DOI: 10.1002/nme