

## PARTITIONED SOLUTION OF THE UNSTEADY ADJOINT EQUATIONS FOR THE ONE-DIMENSIONAL FLOW IN A FLEXIBLE TUBE

J. Degroote<sup>1</sup>, M. Hojjat<sup>2</sup>, E. Stavropoulou<sup>2</sup>, R. Wüchner<sup>2</sup>, K.-U. Bletzinger<sup>2</sup>

<sup>1</sup> Department of Flow, Heat and Combustion Mechanics, Ghent University  
(Joris.Degroote@UGent.be)

<sup>2</sup> Chair of Structural Analysis, Technische Universität München  
({hojjat,stavropoulou,wuechner,kub}@tum.de)

**Abstract.** *For gradient-based optimization, the gradient of the objective function needs to be calculated repeatedly. If the number of design variables is high, this gradient can be obtained efficiently from adjoint equations. This research focuses on the gradient calculation for an objective function which involves a fluid-structure interaction (FSI) simulation. The interaction can be calculated in a partitioned way by coupling a flow solver with a structural solver. In this work, quasi-Newton coupling iterations with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS) are employed for the state equations as well as for their adjoint equations. The problem at hand is the unsteady, one-dimensional flow of an incompressible, inviscid fluid in an elastic tube. Special attention has been given to the interface variables which are exchanged between the adjoint flow and structural solver, to avoid the communication of system matrices between them.*

**Keywords:** *Adjoint, Fluid-structure interaction, partitioned, quasi-Newton.*

### 1. INTRODUCTION

In this paper, unsteady adjoint equations are derived and solved for an elastic tube through which an incompressible fluid flows. The model for the tube wall is a one-dimensional generalised string model and the stiffness of each tube segment is determined by a separate parameter. The cost function measures the difference between the actual displacement of the tube wall and a prescribed displacement.

Several possibilities exist for the calculation of the cost function's gradient with respect to the stiffness parameters. It can be calculated using finite differences, the direct method and the adjoint method [1, 29]. The adjoint method is chosen because its computational cost does not significantly depend on the number of parameters, which is advantageous for a high number of parameters. The difference between the discrete and continuous adjoint formulation is that the discretisation occurs respectively before and after the derivation of the adjoint

equations. Here, the discrete adjoint formulation is selected to obtain the exact gradient of the discrete equations.

The fluid-structure interaction in the model can be dealt with in two ways. In the monolithic approach, the flow equations and structural equations are solved simultaneously [15, 17, 12]. Conversely, the flow equations and structural equations are solved separately in the partitioned approach. To obtain software modularity, the partitioned approach is selected for this research. In weakly (or explicitly, staggered) coupled partitioned simulations, the flow equations and the structural equations are solved once (or a limited, fixed number of times) per time step [10, 8]. As a result, the equilibrium of force and velocity on the fluid-structure interface is not guaranteed. In strongly (or implicitly) coupled partitioned simulations, coupling iterations are performed between the flow equations and the structural equations in each time step. Consequently, the equilibrium conditions on the interface are satisfied (up to an error proportional to the convergence tolerance of the coupling iterations). The weakly coupled methods are suitable for compressible aeroelastic simulations, while strong coupling should be used for simulations with incompressible fluids [35].

Several strongly coupled partitioned techniques treat the flow solver and the structural solver as black boxes, i.e. programmes which calculate an output for a given input, but whose internal algorithms can neither be accessed nor modified. These methods include Gauss-Seidel iterations, Gauss-Seidel iterations with Aitken relaxation [25, 39, 18], interface GMRES [24] and interface quasi-Newton (IQN-ILS) iterations [5, 16]. However, Gauss-Seidel iterations are often unstable if the ratio of the fluid density to the structure density is high, among other reasons [2, 4]. In comparisons consisting of several test cases, IQN-ILS requires fewer coupling iterations per time step than Aitken relaxation or Interface GMRES [5], and the computational cost ratio of a partitioned simulation with IQN-ILS to a monolithic simulation ranges from 0.56 to 3.16 [7]. Therefore, IQN-ILS is selected as coupling algorithm.

In this work, both the forward and the adjoint fluid-structure interaction equations are solved in a partitioned way using the IQN-ILS coupling algorithm. A gradient for a spatially distributed parameter calculated using adjoint equations has already been combined with optimisation of fluid-structure interaction by several authors. For example, it has been applied for the aerostructural shape optimisation of an aeroplane [20, 21] and its wings [22, 23, 9, 11]. However, these are steady problems which can be solved using Gauss-Seidel iterations, while Gauss-Seidel iterations are unstable for the unsteady model at hand. In [27, 28], the adjoint of the steady Euler equations is calculated and the resulting gradient is inserted in an unsteady linear aeroelastic model with four equations to derive a controller for flutter reduction.

Unsteady adjoint solvers have been developed for the optimisation of fluids and structures alone. For example, an adjoint solver of the unsteady Euler equations has been used for the optimisation of blast mitigation devices [34]. Also adjoint solvers of the unsteady Navier-Stokes equations exist [31], even for moving and deforming grids [26]. Checkpointing is often applied to reduce the memory requirements of unsteady adjoint simulations [14, 32, 38], but this is not necessary in this case.

The remainder of this article is organised as follows. The equations and solution pro-

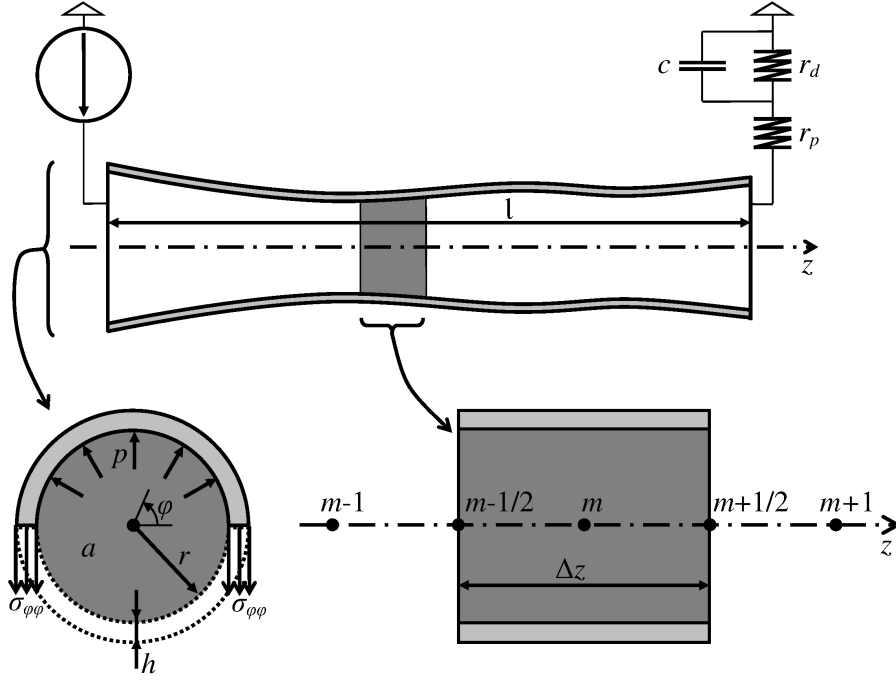


Figure 1. The model for the flow in a straight, flexible tube with details of the cross-section and a control volume used in the discretisation of the governing equations. Also the prescribed velocity at the inlet and the Windkessel model at the outlet are depicted.

cedure are described in Section 2 for the forward problem and in Section 3 for the adjoint problem. Section 4 presents the results, followed by the conclusions in Section 5.

## 2. MODEL

### 2.1. Continuous equations

The unsteady flow of an incompressible, inviscid fluid in a straight, flexible tube is modelled (see Figure 1). The model is one-dimensional in an axisymmetric  $(r, \phi, z)$  coordinate system. The non-overlapping domains of the fluid and the structure are indicated as  $\Omega_f$  and  $\Omega_s$ , respectively. The common boundary of these domains is the fluid-structure interface  $\Gamma$ , defined as

$$\Gamma = \partial\Omega_f \cap \partial\Omega_s, \quad (1)$$

with  $\partial\Omega$  indicating the boundary of domain  $\Omega$ .

The governing equations for the flow are the conservation of mass and momentum, formulated as

$$\frac{\partial a}{\partial t} + \frac{\partial au}{\partial z} = 0 \quad (2a)$$

$$\frac{\partial au}{\partial t} + \frac{\partial au^2}{\partial z} + \frac{1}{\rho_f} \left( \frac{\partial ap}{\partial z} - p \frac{\partial a}{\partial z} \right) = 0, \quad (2b)$$

with  $a$  the cross-sectional area of the tube,  $t$  the time,  $u$  the axial velocity and  $p$  the pressure.

For the structure, a so-called generalised string model is applied. This model is derived from the linear elasticity theory for a cylindrical tube with small thickness under the assumption of membrane deformations [30, 13]. It disregards the axial and circumferential displacement of the tube wall. The governing equation for the structure is given by

$$\rho_s h \frac{\partial^2 r}{\partial t^2} - \kappa G h \frac{\partial^2 r}{\partial z^2} + \frac{E h}{1 - \nu^2} \frac{r - r_o}{r_o^2} - \gamma \frac{\partial^3 r}{\partial^2 z \partial t} = p, \quad (3)$$

with  $r$  the inner radius ( $a = \pi r^2$ ),  $\rho_s$  the structural density and  $h$  the thickness of the tube wall. The other coefficients are the shear modulus  $G$ , the Young modulus  $E$  and the Poisson coefficient  $\nu$ . The viscoelastic term is further omitted ( $\gamma = 0$ ), which is a common simplification [13]. The Timoshenko shear correction factor  $\kappa$  is calculated from the Poisson coefficient [3]

$$\kappa = \frac{2(1 + \nu)}{4 + 3\nu}. \quad (4)$$

## 2.2. Discrete equations

The tube with length  $\ell$  is discretised in space using  $m_e$  segments with length  $\Delta z$ . The pressure and velocity are stored in the cell centres. All terms in the flow equations are discretised using a central scheme, except for first-order upwind discretisation of the convective term in the momentum equation. The time discretisation is first-order backward Euler with time step size  $\Delta t$ . The superscript  $n - 1$  indicates the previous time level ( $t = (n - 1)\Delta t$ ), while the superscript  $n$  for the current time level ( $t = n\Delta t$ ) is omitted.

To obtain structural equations with only first derivatives in time, the radial velocity of the tube wall is introduced ( $v = \partial r / \partial t$ ). Although relatively uncommon, backward Euler time discretisation is applied for the structure as well, to avoid difficulties due to different time discretisation of the flow equations and the structural equations [36]. Moreover, the spatial discretisation is performed using a finite difference scheme instead of the typical finite elements [19].

The elasticity modulus  $E_m$  of each segment is determined by the corresponding parameter  $s_m$  which varies from -1 to 1.

$$E_m = E_o \left( 1 + \frac{1}{2} s_m \right) \quad (5)$$

This definition of the elasticity modulus ensures that it can vary over a realistic range. The goal is to calculate the sensitivity of the cost function with respect to the value of the parameters  $s_m$  ( $m \in \{1, \dots, m_e\}$ ) which appear in this equation.

The discrete equations are subsequently linearised with respect to the reference values  $r_o$ ,  $p_o$ ,  $u_o$  and  $v_o$ . From this point on,  $r$ ,  $p$ ,  $u$  and  $v$  denote perturbations with respect to these reference values. Moreover,  $p_o$ ,  $u_o$  and  $v_o$  are set to zero to simplify the resulting equations. It has been demonstrated in previous research [6] that this particular choice of the reference values results in a model with the same numerical behaviour as the model with all nonlinear terms. Also physical properties such as wave propagation are preserved by this linearisation and reference.

### 2.3. Boundary conditions

As the segments of the tube are indicated with subscripts 1 to  $m_e$ , the inlet (left-hand side) is indicated with a subscript 0 and the outlet (right-hand side) with a subscript  $m_e + 1$ . The flow rate at the inlet is prescribed as a function of the time  $t$  and the pressure is calculated using linear extrapolation. At the outlet of the tube, the velocity is obtained from a linear extrapolation and a Windkessel model relates this velocity with the outlet pressure [37]. The parameter  $s_{m_e+1}$  determines the stiffness of this Windkessel model. The capacitor  $c$  represents the compliance, while the resistors  $r_p$  and  $r_d$  model the viscous resistance. The value of  $c$  is modified by the parameter  $s_{m_e+1}$  which also varies from -1 to 1.

$$c = c_o \left( 1 + \frac{1}{2} s_{m_e+1} \right)^{-1} \quad (6)$$

For the structure, a zero-curvature boundary condition

$$\frac{\partial r}{\partial z} = 0 \quad (7)$$

is applied at both the inlet and the outlet.

### 2.4. Matrix notation

The linearised equations for time step  $n$  can be written in the following block-matrix format

$$\left[ \begin{array}{c|cc} M_f & 0 & 0 \\ \hline 0 & C_f & 0 \\ 0 & 0 & M_s \end{array} \right] \left[ \begin{array}{c} \mathbf{x}_{f\Omega}^n \\ \mathbf{x}_{f\Gamma}^n \\ \mathbf{x}_{s\Gamma}^n \\ \mathbf{x}_{s\Omega}^n \end{array} \right] = \left[ \begin{array}{c} \mathbf{b}_f^n \\ \mathbf{b}_s^n \end{array} \right] + \left[ \begin{array}{c|cc} N_f & 0 & 0 \\ \hline 0 & D_s & 0 \\ 0 & 0 & N_s \end{array} \right] \left[ \begin{array}{c} \mathbf{x}_{f\Omega}^{n-1} \\ \mathbf{x}_{f\Gamma}^{n-1} \\ \mathbf{x}_{s\Gamma}^{n-1} \\ \mathbf{x}_{s\Omega}^{n-1} \end{array} \right]. \quad (8)$$

In this equation,  $M_f$  and  $M_s$  are the system matrices of the flow solver and the structural solver, respectively. The interaction between the fluid and the structure is captured by the off-diagonal blocks  $C_f$  and  $C_s$ . The matrix  $C_f$  describes how the residual of the flow equations changes due to a displacement of the fluid-structure interface, whereas the matrix  $C_s$  converts the pressure on the interface into a contribution (nodal forces) to the residual of the structural equations. The state vector  $\mathbf{x}$  is divided into a fluid part and a structure part, indicated with the subscripts  $f$  and  $s$ , respectively. Both  $\mathbf{x}_f$  and  $\mathbf{x}_s$  are subdivided once more using the subscripts  $\Gamma$  and  $\Omega$ . The subscript  $\Gamma$  refers to variables on the fluid-structure interface, while the subscript  $\Omega$  refers to variables that only appear in either the flow equations or the structural equations. The right-hand side consists of time-dependent vectors  $\mathbf{b}_f$  and  $\mathbf{b}_s$  which do not depend on the state  $\mathbf{x}$ , and a contribution which depends on the state in the previous time step  $\mathbf{x}^{n-1}$ .

The dimensions of  $\mathbf{x}_{f\Omega}$ ,  $\mathbf{x}_{f\Gamma}$ ,  $\mathbf{x}_{s\Gamma}$  and  $\mathbf{x}_{s\Omega}$  are respectively given by  $m_{f\Omega} \times 1$ ,  $m_{f\Gamma} \times 1$ ,  $m_{s\Gamma} \times 1$  and  $m_{s\Omega} \times 1$ . The dimensions of the matrices in Equation 8 follow from the dimension of these vectors. For this specific case, the state vectors are given by

$$\mathbf{x}_{f\Omega} = [u_0 \quad p_0 \quad u_{m_e+1} \quad p_{m_e+1} \quad u_1 \quad u_2 \quad \cdots \quad u_{m_e}]^T \quad (9a)$$

$$\mathbf{x}_{f\Gamma} = [p_1 \quad p_2 \quad \cdots \quad p_{m_e}]^T \quad (9b)$$

$$\mathbf{x}_{s\Gamma} = [r_1 \quad r_2 \quad \cdots \quad r_{m_e}]^T \quad (9c)$$

$$\mathbf{x}_{s\Omega} = [v_1 \quad v_2 \quad \cdots \quad v_{m_e}]^T, \quad (9d)$$

with the superscript T indicating a transpose. The inlet and outlet model for the flow as described in Section 2.3 are included in  $m_{f\Omega}$ . The dimensions mentioned above can thus be written as a function of the number of tube segments  $m_e$ .

$$m_{f\Omega} = 6 + m_e \quad (10a)$$

$$m_{f\Gamma} = m_{s\Gamma} = m_{s\Omega} = m_e \quad (10b)$$

Equation 8 can be abbreviated as

$$\mathbf{A}\mathbf{x}^n = \mathbf{b}^n + \mathbf{B}\mathbf{x}^{n-1}, \quad (11)$$

with

$$\mathbf{A} = \left[ \begin{array}{c|cc} \mathbf{M}_f & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{C}_f & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{M}_s \end{array} \right] \quad \text{and} \quad \mathbf{B} = \left[ \begin{array}{c|cc} \mathbf{N}_f & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{D}_s & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{N}_s \end{array} \right]. \quad (12)$$

The initial state  $\mathbf{x}^0$  is set to zero. For the specific case that is analysed in this work, the block  $\mathbf{D}_s$  in this general equation is filled with zeros. Due to the linearisation, the matrices  $\mathbf{A}$  and  $\mathbf{B}$  are independent of  $\mathbf{x}$  and therefore they remain constant during the simulation. The residual of the governing equations for time step  $n$  is then given by

$$\mathbf{r}^n(\mathbf{x}^n, \mathbf{x}^{n-1}) = \mathbf{0}, \quad (13)$$

with

$$\mathbf{r}^n(\mathbf{x}^n, \mathbf{x}^{n-1}) = \mathbf{A}\mathbf{x}^n - \mathbf{b}^n - \mathbf{B}\mathbf{x}^{n-1}. \quad (14)$$

## 2.5. Coupling iterations

Equation 8 is solved in a partitioned way, which signifies that the flow equations and the structural equations are solved separately. Consequently, coupling iterations need to be performed between the flow equations and the structural equations to obtain the solution of the coupled problem. At convergence of the coupling iterations, the solution is identical to what a monolithic solver would calculate (up to an error proportional to the convergence tolerance of the coupling iterations). In every coupling iteration in time step  $n$ , the flow equations and the structural equations are solved with given values of  $\mathbf{x}_{s\Gamma}$  and  $\mathbf{x}_{f\Gamma}$ , respectively. In the following sections, the superscript  $k$  indicates coupling iteration  $k$  in time step  $n$ .

In the first coupling iteration, an extrapolation ( $\mathcal{E}$ ) based on previous time steps is performed to obtain the input for the flow solver. If no reuse from previous time steps is applied in the least-squares model, then a relaxation with factor  $\omega$  determines the input for the flow solver in the second coupling iteration. At the beginning of coupling iteration  $k$ , the coupling algorithm calculates  $\mathbf{x}_{s\Gamma}^k$ . The flow equations are then given by

$$\left[ \begin{array}{c} \mathbf{M}_f \end{array} \right] \left[ \begin{array}{c} \mathbf{x}_{f\Omega}^k \\ \mathbf{x}_{f\Gamma}^k \end{array} \right] = \left[ \begin{array}{c} \mathbf{b}_f \end{array} \right] + \left[ \begin{array}{c} \mathbf{N}_f \end{array} \right] \left[ \begin{array}{c} \mathbf{x}_{f\Omega}^{n-1} \\ \mathbf{x}_{f\Gamma}^{n-1} \end{array} \right] + \left[ \begin{array}{c} \mathbf{0} \\ \mathbf{D}_f \mathbf{x}_{s\Gamma}^{n-1} \end{array} \right] - \left[ \begin{array}{c} \mathbf{0} \\ \mathbf{C}_f \mathbf{x}_{s\Gamma}^k \end{array} \right]. \quad (15)$$

Once  $\mathbf{x}_{f\Omega}^k$  and  $\mathbf{x}_{f\Gamma}^k$  have been calculated,  $\mathbf{x}_{f\Gamma}^k$  is given to the structural solver. This calculation is further referred to as  $\mathbf{x}_{f\Gamma}^k = \mathcal{F}(\mathbf{x}_{s\Gamma}^k)$ .

The structural solver subsequently solves the structural equations

$$\begin{bmatrix} \mathbf{M}_s \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{s\Gamma}^k \\ \mathbf{x}_{s\Omega}^k \end{bmatrix} = \begin{bmatrix} \mathbf{b}_s \end{bmatrix} + \begin{bmatrix} \mathbf{N}_s \end{bmatrix} \begin{bmatrix} \mathbf{x}_{s\Gamma}^{n-1} \\ \mathbf{x}_{s\Omega}^{n-1} \end{bmatrix} + \begin{bmatrix} \mathbf{D}_s \mathbf{x}_{f\Gamma}^{n-1} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{C}_s \mathbf{x}_{f\Gamma}^k \\ \mathbf{0} \end{bmatrix} \quad (16)$$

for  $\tilde{\mathbf{x}}_{s\Gamma}^k$  and  $\mathbf{x}_{s\Omega}^k$ . The tilde is used to distinguish between the motion of the fluid-structure interface calculated by the coupling algorithm at the beginning of the coupling iteration and that calculated by the structural solver at the end. This calculation is further referred to as  $\tilde{\mathbf{x}}_{s\Gamma}^k = \mathcal{S}(\mathbf{x}_{f\Gamma}^k)$ .

The most trivial coupling algorithm is Gauss-Seidel iteration, which means that the motion of the fluid-structure interface calculated by the structural solver at the end of the previous coupling iteration is applied by the flow solver at the beginning of the next coupling iteration.

$$\mathbf{x}_{s\Gamma}^k = \tilde{\mathbf{x}}_{s\Gamma}^{k-1} \quad (17)$$

However, it is well-understood that this coupling algorithm is unstable for fluid-structure interaction problems with an incompressible fluid and comparable density of the fluid and the structure [2, 6, 4].

Therefore, the interface quasi-Newton coupling algorithm with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS) is applied [5]. This coupling algorithm treats both the flow solver and the structural solver as black boxes. Using the interface displacement applied in the flow solver ( $\mathbf{x}_{s\Gamma}^k$ ) and calculated by the structural solver ( $\tilde{\mathbf{x}}_{s\Gamma}^k$ ) in all coupling iterations, the coupling algorithm constructs an approximation (indicated with a hat) for the inverse of the Jacobian of the interface residual

$$\mathbf{r}_{s\Gamma}^k = \tilde{\mathbf{x}}_{s\Gamma}^k - \mathbf{x}_{s\Gamma}^k \quad (18)$$

with respect to the interface displacement  $\mathbf{x}_{s\Gamma}^k$ . This results in the following update at the beginning of each coupling iteration

$$\mathbf{x}_{s\Gamma}^k = \mathbf{x}_{s\Gamma}^{k-1} + \Delta \mathbf{x}_{s\Gamma}^k \quad (19a)$$

$$\Delta \mathbf{x}_{s\Gamma}^k = \left( \widehat{\frac{\partial \mathbf{r}_{s\Gamma}}{\partial \mathbf{x}_{s\Gamma}}} \right)^{-1} \Delta \mathbf{r}_{s\Gamma}^k, \quad (19b)$$

with  $\Delta \mathbf{x}_{s\Gamma}^k = \mathbf{x}_{s\Gamma}^k - \mathbf{x}_{s\Gamma}^{k-1}$  and  $\Delta \mathbf{r}_{s\Gamma}^k = \mathbf{r}_{s\Gamma}^k - \mathbf{r}_{s\Gamma}^{k-1} = -\mathbf{r}_{s\Gamma}^{k-1}$  because the goal is to find  $\mathbf{x}_{s\Gamma}^k$  so that  $\mathbf{r}_{s\Gamma}^k = \mathbf{0}$ .

Because only the product of the approximation for the inverse of the Jacobian with a vector is required, this matrix does not have to be constructed explicitly. In a matrix-free implementation of IQN-ILS, the computational cost and the memory requirements of the coupling algorithm scale linearly with the number of degrees of freedom in the interface displacement ( $m_{s\Gamma}$ ). The least-squares model which predicts  $\Delta \tilde{\mathbf{x}}_{s\Gamma}^{k-1}$  as a function of  $-\mathbf{r}_{s\Gamma}^{k-1}$  is further represented by  $\mathcal{M}^k$ .

Coupling iterations are performed until  $\|\mathbf{r}_{s\Gamma}^k\|_2 < \epsilon_c \|\mathbf{r}_{s\Gamma}^1\|_2$ , in which  $\epsilon_c$  denotes the relative convergence criterion of the coupling iterations. The complete procedure for the forward simulation is described in Algorithm 1.

---

**Algorithm 1** The IQN-ILS coupling algorithm for the forward simulation.

---

```

1: for  $n = 1, \dots, n_e$  do
2:   for  $k = 1, \dots, k_e$  do
3:     if  $k = 1$  then
4:        $\mathbf{x}_{s\Gamma}^k = \mathcal{E}^n(\mathbf{x}_{s\Gamma}^{n-1}, \mathbf{x}_{s\Gamma}^{n-2}, \mathbf{x}_{s\Gamma}^{n-3})$ 
5:     else if  $k = 2$  and no reuse then
6:        $\mathbf{x}_{s\Gamma}^k = \mathbf{x}_{s\Gamma}^{k-1} + \omega \mathbf{r}_{s\Gamma}^{k-1}$ 
7:     else
8:        $\Delta \mathbf{x}_{s\Gamma}^k = \mathcal{M}^k(-\mathbf{r}_{s\Gamma}^{k-1}) + \mathbf{r}_{s\Gamma}^{k-1}$ 
9:        $\mathbf{x}_{s\Gamma}^k = \mathbf{x}_{s\Gamma}^{k-1} + \Delta \mathbf{x}_{s\Gamma}^k$ 
10:    end if
11:     $\mathbf{x}_{f\Gamma}^k = \mathcal{F}(\mathbf{x}_{s\Gamma}^k)$  Equation 15
12:     $\tilde{\mathbf{x}}_{s\Gamma}^k = \mathcal{S}(\mathbf{x}_{f\Gamma}^k)$  Equation 16
13:     $\mathbf{r}_{s\Gamma}^k = \tilde{\mathbf{x}}_{s\Gamma}^k - \mathbf{x}_{s\Gamma}^k$  Equation 18
14:    if  $k > 2$  and  $\|\mathbf{r}_{s\Gamma}^k\|_2 < \epsilon_c \|\mathbf{r}_{s\Gamma}^1\|_2$  then
15:       $k = k_e + 1$ 
16:    end if
17:  end for
18: end for

```

---

### 3. SENSITIVITY ANALYSIS

#### 3.1. Cost function

The dimensionless least-squares cost function  $j$  is defined using a normalised difference between a prescribed displacement and a simulation of the model described above. It is given by

$$j(\mathbf{s}, \mathbf{x}) = \frac{(\bar{\mathbf{x}}_{s\Gamma} - \bar{\mathbf{x}}_{s\Gamma}^{ref})^T (\bar{\mathbf{x}}_{s\Gamma} - \bar{\mathbf{x}}_{s\Gamma}^{ref})}{m_e n_e (\max \bar{\mathbf{x}}_{s\Gamma}^{ref} - \min \bar{\mathbf{x}}_{s\Gamma}^{ref})^2}, \quad (20)$$

with the superscript *ref* referring to the prescribed displacement (or reference). The vector  $\mathbf{s}$  contains the  $m_e + 1$  parameters which are defined in Equation 5 and Equation 6. The vector  $\mathbf{x}$  is the combination of the state vectors of all time steps

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^{n_e} \end{bmatrix}. \quad (21)$$

The vector  $\bar{\mathbf{x}}_{s\Gamma}$  is identical to  $\mathbf{x}$ , but all entries which do not correspond to  $\mathbf{x}_{s\Gamma}^n$  have been set to zero, giving

$$\bar{\mathbf{x}}_{s\Gamma} = \begin{bmatrix} \bar{\mathbf{x}}_{s\Gamma}^1 \\ \bar{\mathbf{x}}_{s\Gamma}^2 \\ \vdots \\ \bar{\mathbf{x}}_{s\Gamma}^{n_e} \end{bmatrix}, \quad \text{with} \quad \bar{\mathbf{x}}_{s\Gamma}^n = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{x}_{s\Gamma}^n \\ \mathbf{0} \end{bmatrix} \cdot \begin{matrix} \} m_{f\Omega} \times 1 \\ \} m_{f\Gamma} \times 1 \\ \} m_{s\Gamma} \times 1 \\ \} m_{s\Omega} \times 1 \end{matrix} \quad (22)$$



So, the cost function is a sum over all time steps and all tube segments of the squared difference between the radius in the simulation and in the measurement.

### 3.2. Minimisation problem

With the definition of the cost function in Equation 20, a minimisation problem can be formulated as

$$\min_{\mathbf{s}, \mathbf{x}} j(\mathbf{s}, \mathbf{x}) \quad (23)$$

with the governing equations as constraints

$$\mathbf{r}(\mathbf{s}, \mathbf{x}) = \mathbf{0}. \quad (24)$$

The parameter vector  $\mathbf{s}$  and the state vector  $\mathbf{x}$  are defined in Equations 5, 6 and 21. Here,  $\mathbf{r}$  is the combination of the residual of the governing equations in all time steps

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}^1 \\ \mathbf{r}^2 \\ \vdots \\ \mathbf{r}^{n_e} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^1(\mathbf{x}^1, \mathbf{x}^0) \\ \mathbf{r}^2(\mathbf{x}^2, \mathbf{x}^1) \\ \vdots \\ \mathbf{r}^{n_e}(\mathbf{x}^{n_e}, \mathbf{x}^{n_e-1}) \end{bmatrix}. \quad (25)$$

### 3.3. Adjoint equations

As the state vector depends on the parameters, the gradient of the cost function  $j(\mathbf{s}, \mathbf{x}) = j(\mathbf{s}, \mathbf{x}(\mathbf{s}))$  requires application of the chain rule. The total derivatives of  $j$  with respect to the parameters are

$$\frac{dj}{d\mathbf{s}} = \frac{\partial j}{\partial \mathbf{s}} + \frac{\partial j}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{s}}. \quad (26)$$

The partial derivatives in this equation can be calculated quickly and easily from Equation 20, as the state  $\mathbf{x}$  remains constant for  $\partial j / \partial \mathbf{s}$  and the parameters  $\mathbf{s}$  remain constant for  $\partial j / \partial \mathbf{x}$ . By contrast, the total derivative  $d\mathbf{x} / d\mathbf{s}$  requires the solution of the unsteady fluid-structure interaction problem and it is thus time-consuming to calculate.

As the governing equations of the fluid-structure interaction problem always need to be satisfied,  $\mathbf{r}$  should always be equal to  $\mathbf{0}$ . Consequently, also the total derivative  $d\mathbf{r} / d\mathbf{s}$  should be equal to  $\mathbf{0}$ . By applying the chain rule, this total derivative is given by

$$\frac{d\mathbf{r}}{d\mathbf{s}} = \frac{\partial \mathbf{r}}{\partial \mathbf{s}} + \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{s}} = \mathbf{0}. \quad (27)$$

Rewriting the previous equation as

$$\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{s}} = -\frac{\partial \mathbf{r}}{\partial \mathbf{s}} \quad (28)$$

results in a system that can be solved for the total derivative  $d\mathbf{x} / d\mathbf{s}$ . The result is subsequently substituted in Equation 26, yielding

$$\frac{dj}{d\mathbf{s}} = \frac{\partial j}{\partial \mathbf{s}} - \underbrace{\frac{\partial j}{\partial \mathbf{x}} \left( \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{s}}}_{-\mathbf{a}^T} \quad (29)$$

In this equation, the matrix inversion is a symbolic notation for the solution of a large system, which combines all equations of all time steps of the unsteady fluid-structure interaction problem. Obviously, this matrix is neither constructed explicitly nor inverted or factorised due to excessive memory requirements. As explained above, the time steps are calculated consecutively with a partitioned solution technique.

Depending on how the factors are grouped in the last term of Equation 29, two methods can be distinguished. The direct method solves Equation 28 for  $\mathrm{d}\mathbf{x}/\mathrm{d}s$  and substitutes the result in Equation 29. By contrast, the adjoint method first calculates the product of the first two factors in the last term of Equation 29.

$$\frac{\partial j}{\partial \mathbf{x}} \left( \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right)^{-1} = -\mathbf{a}^T \quad (30)$$

The vector  $\mathbf{a}$  is the so-called adjoint (or dual) state, which is the solution of the adjoint equation

$$\left( \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right)^T \mathbf{a} = - \left( \frac{\partial j}{\partial \mathbf{x}} \right)^T. \quad (31)$$

The sought-after gradient  $\mathrm{d}j/\mathrm{d}s$  is finally calculated as

$$\frac{\mathrm{d}j}{\mathrm{d}s} = \frac{\partial j}{\partial s} + \mathbf{a}^T \frac{\partial \mathbf{r}}{\partial s}. \quad (32)$$

The computational cost of solving Equation 31 is independent of the number of parameters, so the adjoint method is suitable for a large number of parameters. However, there are as many right-hand sides as objective functions. As for the direct method, the matrix in Equation 31 is too large to be factorised and stored in the case of unsteady fluid-structure interaction. So, the adjoint equations need to be solved for every objective function. Alternatively, the adjoint method can be derived by introducing Lagrange multipliers, for example as in [33].

From the explanation above, it is clear that the choice between the direct and the adjoint method depends on the number of parameters and objective functions. The direct method requires the solution of a problem comparable to the governing equations for every parameter, while its cost is almost independent of the number of objective functions. The opposite is true for the adjoint method. If central (respectively forward) finite differences are used for the calculation of the gradient, the governing equations need to be solved twice (respectively once) for every parameter and for every objective function. In this specific case, there is only one objective function and there are many parameters, so the adjoint method is selected.

In this case, the cost function (Equation 20) does not explicitly depend on the parameters  $\mathbf{s}$ , so

$$\frac{\partial j}{\partial \mathbf{s}} = \mathbf{0}. \quad (33)$$

Conversely, the residual  $\mathbf{r}$  (Equation 14) depends on the parameters  $\mathbf{s}$ , giving

$$\frac{\partial \mathbf{r}}{\partial \mathbf{s}} = \begin{bmatrix} \frac{\partial \mathbf{A}}{\partial \mathbf{s}} \mathbf{x}^1 - \left( \frac{\partial \mathbf{b}}{\partial \mathbf{s}} \right)^1 - \frac{\partial \mathbf{B}}{\partial \mathbf{s}} \mathbf{x}^0 \\ \frac{\partial \mathbf{A}}{\partial \mathbf{s}} \mathbf{x}^2 - \left( \frac{\partial \mathbf{b}}{\partial \mathbf{s}} \right)^2 - \frac{\partial \mathbf{B}}{\partial \mathbf{s}} \mathbf{x}^1 \\ \vdots \\ \frac{\partial \mathbf{A}}{\partial \mathbf{s}} \mathbf{x}^{n_e} - \left( \frac{\partial \mathbf{b}}{\partial \mathbf{s}} \right)^{n_e} - \frac{\partial \mathbf{B}}{\partial \mathbf{s}} \mathbf{x}^{n_e-1} \end{bmatrix}. \quad (34)$$

The only non-zero contributions to  $\partial \mathbf{r} / \partial \mathbf{s}$  are due to the terms that contain the elasticity modulus  $E_m$  (Equation 5) and the compliance of the Windkessel model  $c$  (Equation 6).

$$\frac{\partial E_m}{\partial s_m} = \frac{E_o}{2} \quad (35a)$$

$$\frac{\partial c}{\partial s_{m_e+1}} = \frac{-c_o}{\left(1 + \frac{1}{2}s_{m_e+1}\right)^2} \frac{1}{2} \quad (35b)$$

In the adjoint equation (Equation 31), both  $(\partial j / \partial \mathbf{x})^T$  and  $(\partial \mathbf{r} / \partial \mathbf{x})^T$  are required. The former is calculated analytically, giving

$$\left(\frac{\partial j}{\partial \mathbf{x}}\right)^T = \frac{2 \left(\bar{\mathbf{x}}_{s\Gamma} - \bar{\mathbf{x}}_{s\Gamma}^{ref}\right)}{m_e n_e \left(\max \bar{\mathbf{x}}_{s\Gamma}^{ref} - \min \bar{\mathbf{x}}_{s\Gamma}^{ref}\right)^2}. \quad (36)$$

The calculation of  $\partial \mathbf{r} / \partial \mathbf{x}$  and its transpose are more involved. This matrix is not constructed or stored, but the solution to Equation 31 is calculated using time steps. Using Equation 14,  $\partial \mathbf{r} / \partial \mathbf{x}$  is given by

$$\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{r}^1}{\partial \mathbf{x}^1} & \frac{\partial \mathbf{r}^1}{\partial \mathbf{x}^2} & \cdots & \frac{\partial \mathbf{r}^1}{\partial \mathbf{x}^{n_e}} \\ \frac{\partial \mathbf{r}^2}{\partial \mathbf{x}^1} & \frac{\partial \mathbf{r}^2}{\partial \mathbf{x}^2} & & \frac{\partial \mathbf{r}^2}{\partial \mathbf{x}^{n_e}} \\ \vdots & & \ddots & \\ \frac{\partial \mathbf{r}^{n_e}}{\partial \mathbf{x}^1} & \frac{\partial \mathbf{r}^{n_e}}{\partial \mathbf{x}^2} & & \frac{\partial \mathbf{r}^{n_e}}{\partial \mathbf{x}^{n_e}} \end{bmatrix} \quad (37a)$$

$$= \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\mathbf{B} & \mathbf{A} & \mathbf{0} & & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{B} & \mathbf{A} & & \mathbf{0} & \mathbf{0} \\ \vdots & & & \ddots & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & -\mathbf{B} & \mathbf{A} \end{bmatrix} \quad (37b)$$

Because  $\partial \mathbf{r} / \partial \mathbf{x}$  is lower bidiagonal, the forward simulation consists of forward time steps. By contrast, its transpose is upper bidiagonal, which is the reason for the backward time steps in the adjoint simulation.

$$\left(\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right)^T = \begin{bmatrix} \mathbf{A}^T & -\mathbf{B}^T & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^T & -\mathbf{B}^T & & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^T & & \mathbf{0} & \mathbf{0} \\ \vdots & & & \ddots & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{A}^T & -\mathbf{B}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{0} & \mathbf{A}^T \end{bmatrix} \quad (38)$$

### 3.4. Matrix notation

Considering Equation 38, the block-matrix format of a time step for the solution of the adjoint equation (Equation 31) is similar to Equation 8.

$$\left[ \begin{array}{c|cc} \mathbf{M}_f^T & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{C}_f^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_s^T \end{array} \right] \left[ \begin{array}{c} \mathbf{a}_{f\Omega}^n \\ \mathbf{a}_{f\Gamma}^n \\ \mathbf{a}_{s\Gamma}^n \\ \mathbf{a}_{s\Omega}^n \end{array} \right] = - \left[ \begin{array}{c} \left(\frac{\partial j}{\partial \mathbf{x}_f^n}\right)^T \\ \left(\frac{\partial j}{\partial \mathbf{x}_s^n}\right)^T \end{array} \right] + \left[ \begin{array}{c|cc} \mathbf{N}_f^T & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{D}_f^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N}_s^T \end{array} \right] \left[ \begin{array}{c} \mathbf{a}_{f\Omega}^{n+1} \\ \mathbf{a}_{f\Gamma}^{n+1} \\ \mathbf{a}_{s\Gamma}^{n+1} \\ \mathbf{a}_{s\Omega}^{n+1} \end{array} \right] \quad (39)$$

This equation is written in a general form. The structure is self-adjoint which allows for a simplification ( $M_s^T = M_s$ ), but this is not used here to keep the formulation general. In abbreviated form, the previous equation yields

$$A^T a^n = - \left( \frac{\partial j}{\partial x^n} \right)^T + B^T a^{n+1}, \quad (40)$$

which is similar to Equation 11, except for the backward time steps ( $n \in \{n_e, \dots, 1\}$ ). The initial adjoint state  $a^{n_e+1}$  is again set to zero.

### 3.5. Coupling iterations

Equation 39 is solved in a partitioned way as well, by performing coupling iterations between the adjoint flow equations and the adjoint structural equations until the solution of the coupled adjoint problem has been found. The superscript  $k$  again indicates coupling iteration  $k$  in time step  $n$ .

The adjoint flow equations in coupling iteration  $k$  are given by

$$\begin{bmatrix} M_f^T \end{bmatrix} \begin{bmatrix} a_{f\Omega}^k \\ a_{f\Gamma}^k \end{bmatrix} = - \begin{bmatrix} \left( \frac{\partial j}{\partial x_f^n} \right)^T \end{bmatrix} + \begin{bmatrix} N_f^T \end{bmatrix} \begin{bmatrix} a_{s\Gamma}^{n+1} \\ a_{s\Omega}^{n+1} \end{bmatrix} + \begin{bmatrix} 0 \\ D_s^T a_{s\Gamma}^{n+1} \end{bmatrix} - \begin{bmatrix} 0 \\ C_s^T a_{s\Gamma}^k \end{bmatrix}, \quad (41)$$

which are solved for  $a_{f\Omega}^k$  and  $a_{f\Gamma}^k$ . This equation contains  $C_s$  and  $D_s$ , which belong to the structural solver. The exchange of matrices between the flow solver and the structural solver is highly unwanted in the partitioned approach. Therefore,  $C_s^T a_{s\Gamma}^k$  and  $D_s^T a_{s\Gamma}^{n+1}$  are given to the flow solver, instead of  $a_{s\Gamma}^k$  and  $a_{s\Gamma}^{n+1}$ . While  $C_s^T a_{s\Gamma}^k$  is provided to the flow solver in every coupling iteration,  $D_s^T a_{s\Gamma}^{n+1}$  is only exchanged once at the beginning of every time step. In the special case that  $C_s$  and  $D_s$  are identical, giving  $D_s^T a_{s\Gamma}^{n+1}$  to the flow solver is not required because the flow solver can store the information from  $t^{n+1}$ .

The structural solver subsequently solves the adjoint structural equations

$$\begin{bmatrix} M_s^T \end{bmatrix} \begin{bmatrix} \tilde{a}_{s\Gamma}^k \\ a_{s\Omega}^k \end{bmatrix} = - \begin{bmatrix} \left( \frac{\partial j}{\partial x_s^n} \right)^T \end{bmatrix} + \begin{bmatrix} N_s^T \end{bmatrix} \begin{bmatrix} a_{s\Gamma}^{n+1} \\ a_{s\Omega}^{n+1} \end{bmatrix} + \begin{bmatrix} D_f^T a_{f\Gamma}^{n+1} \\ 0 \end{bmatrix} - \begin{bmatrix} C_f^T a_{f\Gamma}^k \\ 0 \end{bmatrix} \quad (42)$$

for  $\tilde{a}_{s\Gamma}^k$  and  $a_{s\Omega}^k$ . The tilde is used to distinguish between the adjoint state calculated by the coupling algorithm at the beginning of the coupling iteration and that calculated by the structural solver at the end. This equation contains  $C_f$  and  $D_f$ , which belong to the flow solver. To avoid the exchange of matrices,  $C_f^T a_{f\Gamma}^k$  and  $D_f^T a_{f\Gamma}^{n+1}$  are provided to the structural solver. While the former is given in every coupling iteration, the latter is only given once at the beginning of the time step.

In conclusion, the flow solver calculates  $a_f^k$  for a given  $C_s^T a_{s\Gamma}^k$  and gives  $C_f^T a_{f\Gamma}^k$  to the structural solver, whereas the structural solver calculates  $a_s^k$  for a given  $C_f^T a_{f\Gamma}^k$  and returns  $C_s^T \tilde{a}_{s\Gamma}^k$ . These calculations are further referred to as  $C_f^T a_{f\Gamma}^k = \mathcal{F}^T(C_s^T a_{s\Gamma}^k)$  and  $C_s^T \tilde{a}_{s\Gamma}^k = \mathcal{S}^T(C_f^T a_{f\Gamma}^k)$ .

The adjoint flow equations (Equation 41) and structural equations (Equation 42) are coupled using the IQN-ILS algorithm, similarly to the forward equations. The partitioned adjoint simulation is described in Algorithm 2. The differences with the partitioned forward

simulation are as follows. First, the vectors  $\mathbf{D}_f^T \mathbf{a}_{f\Gamma}^{n+1}$  and  $\mathbf{D}_s^T \mathbf{a}_{s\Gamma}^{n+1}$  are exchanged between the flow solver and the structural solver at the beginning of every time step (lines 2-3). Then, the extrapolation ( $\mathcal{E}$ ) is adapted in a straightforward way to the backward time steps (line 6). Moreover, the interface residual is defined as

$$\mathbf{r}_{s\Gamma}^k = \mathbf{C}_s^T \tilde{\mathbf{a}}_{s\Gamma}^k - \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k \quad (43)$$

in the adjoint simulation. Finally,  $\tilde{\mathbf{x}}_{s\Gamma}^i$  is replaced by  $\mathbf{C}_s^T \tilde{\mathbf{a}}_{s\Gamma}^i$  in the least-squares model ( $\mathcal{M}$  on line 10).

---

**Algorithm 2** The IQN-ILS coupling algorithm for the adjoint simulation.

---

```

1: for  $n = n_e, \dots, 1$  do
2:   send  $\mathbf{D}_f^T \mathbf{a}_{f\Gamma}^{n+1}$  from flow solver to structural solver
3:   send  $\mathbf{D}_s^T \mathbf{a}_{s\Gamma}^{n+1}$  from structural solver to flow solver
4:   for  $k = 1, \dots, k_e$  do
5:     if  $k = 1$  then
6:        $\mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k = \mathcal{E}^n(\mathbf{C}_s^T \mathbf{a}_{s\Gamma}^{n+1}, \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^{n+2}, \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^{n+3})$ 
7:     else if  $k = 2$  and no reuse then
8:        $\mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k = \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^{k-1} + \omega \mathbf{r}_{s\Gamma}^{k-1}$ 
9:     else
10:       $\Delta \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k = \mathcal{M}^k(-\mathbf{r}_{s\Gamma}^{k-1}) + \mathbf{r}_{s\Gamma}^{k-1}$ 
11:       $\mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k = \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^{k-1} + \Delta \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k$ 
12:    end if
13:     $\mathbf{C}_f^T \mathbf{a}_{f\Gamma}^k = \mathcal{F}^T(\mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k)$  Equation 41
14:     $\mathbf{C}_s^T \tilde{\mathbf{a}}_{s\Gamma}^k = \mathcal{S}^T(\mathbf{C}_f^T \mathbf{a}_{f\Gamma}^k)$  Equation 42
15:     $\mathbf{r}_{s\Gamma}^k = \mathbf{C}_s^T \tilde{\mathbf{a}}_{s\Gamma}^k - \mathbf{C}_s^T \mathbf{a}_{s\Gamma}^k$  Equation 43
16:    if  $k > 2$  and  $\|\mathbf{r}_{s\Gamma}^k\|_2 < \epsilon_c \|\mathbf{r}_{s\Gamma}^1\|_2$  then
17:       $k = k_e + 1$ 
18:    end if
19:  end for
20: end for

```

---

## 4. RESULTS

For the results, the tube is discretised in  $m_e = 100$  segments. The parameters of the fluid-structure interaction model and the Windkessel model are listed in Table 1.

Figure 2 depicts the radius at the middle of the tube during one period for the minimal ( $s = -1$ ), nominal ( $s = 0$ ) and maximal ( $s = 1$ ) values of all parameters. The differences in the other variables and at other locations are of the same order of magnitude. The cost function's gradient for  $s = 0$  is shown in this figure, with the reference in Equation 20 calculated using a sinusoidal variation of the tube's stiffness.

The adjoint calculation of the cost function's gradient is verified by comparing it to central finite differences

$$\left( \frac{dj}{ds_m} \right)^{fd} = \frac{j(s + \Delta s_m) - j(s - \Delta s_m)}{2\Delta s_m}, \quad (44)$$

Table 1. The parameters of the fluid-structure interaction model and the Windkessel model [37].

$\rho_f$	1060 kg/m <sup>3</sup>	$\rho_s$	1000 kg/m <sup>3</sup>
$r_o$	3·10 <sup>-3</sup> m	$E_o$	4·10 <sup>5</sup> Pa
$h$	3·10 <sup>-4</sup> m	$G$	4·10 <sup>5</sup> Pa
$\ell$	0.126 m	$\nu$	0.5
$c_o$	6.35·10 <sup>-10</sup> m <sup>3</sup> /Pa	$r_d$	1.768·10 <sup>9</sup> Pa·s/m <sup>3</sup>
$r_p$	2.834·10 <sup>8</sup> Pa·s/m <sup>3</sup>	$t_b$	1 s

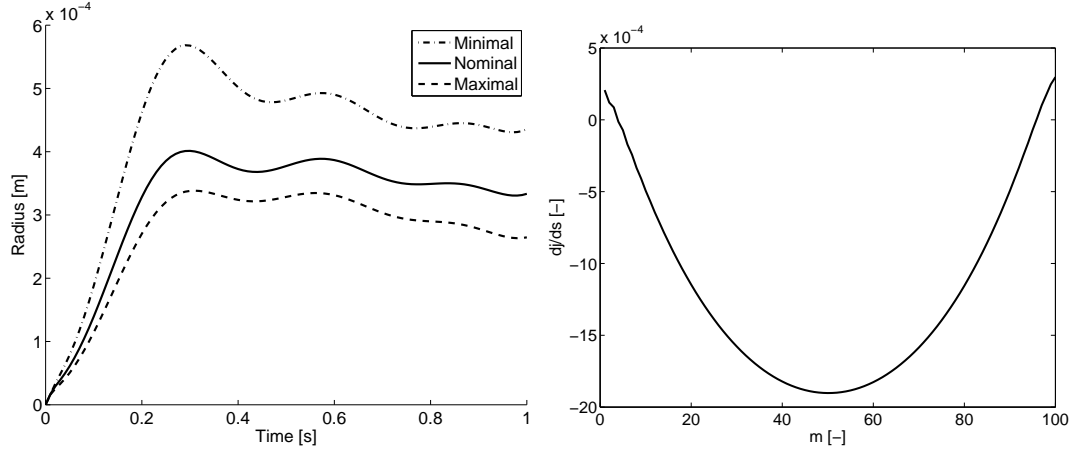


Figure 2. (left) The radius linearised with respect to  $r_o$  at the middle of the tube ( $z = \ell/2$ ) as a function of time for minimal ( $s = -1$ ), nominal ( $s = 0$ ) and maximal values ( $s = 1$ ) of all parameters. (right) The cost function's gradient with respect to the stiffness of the tube segments for  $s = 0$  if the reference is calculated with a sinusoidal variation of the stiffness.

with  $s_m$  and  $\Delta s_m$  respectively the value and the perturbation of element  $m$  in vector  $s$ . This comparison is performed for different values of  $m$  and  $s$ , with  $\Delta s_m = 10^{-3}$  and  $10^{-4}$ .

Table 2 lists the values from this comparison between the adjoint calculation and the finite difference calculation with  $\Delta s_m = 10^{-4}$ . All parameters in the reference calculation (required for the cost function calculation in Equation 20) have been set to 1. For the forward and the adjoint calculation, the elements of the vector  $s$  have consecutively been set to -1, 0 and 1. Elements  $m \in \{1, 10, 101\}$  of the gradient are analysed. The difference between both gradients is at least 5 orders of magnitude smaller than the absolute value of the gradient.

## 5. CONCLUSIONS

In this work, the cost function is calculated with an unsteady fluid-structure interaction simulation and its gradient is obtained from an unsteady adjoint simulation. Both the forward and the adjoint simulation are partitioned with a quasi-Newton coupling algorithm (IQN-ILS). The exchange of operators between the flow solver and the structural solver is avoided by a particular choice of the variables that are exchanged between the solvers.

Table 2. The verification of the gradient calculation by means of a comparison with finite differences using step size  $\Delta s_m = 10^{-4}$ .

$m$	$s$	$dj/ds_m$	$(dj/ds_m)^{fd}$	$ dj/ds_m - (dj/ds_m)^{fd} $
1	-1	-9.4184248e-03	-9.4184185e-03	6.3051349e-09
1	0	-1.2726096e-03	-1.2726133e-03	3.6851994e-09
1	1	0.0000000e+00	-1.0685503e-12	1.0685503e-12
10	-1	-1.0022075e-02	-1.0022066e-02	9.3869615e-09
10	0	-1.3523483e-03	-1.3523465e-03	1.8668861e-09
10	1	0.0000000e+00	-4.2920656e-13	4.2920656e-13
101	-1	4.8240110e-01	4.8240119e-01	8.5818958e-08
101	0	7.0555794e-02	7.0555806e-02	1.1679346e-08
101	1	0.0000000e+00	-4.8473439e-11	4.8473439e-11

## Acknowledgements

Joris Degroote gratefully acknowledges funding by a post-doctoral fellowship of the Research Foundation - Flanders (FWO).

## References

- [1] R. Balasubramanian and J.C. Newman. Discrete direct and adjoint sensitivity analysis for arbitrary mach number flows. *International Journal for Numerical Methods in Engineering*, 66(2):297–318, 2006.
- [2] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42–44):4506–4527, 2005.
- [3] G. Cowper. The shear coefficient in Timoshenko’s beam theory. *Journal of Applied Mechanics*, 33(2):335–340, 1966.
- [4] J. Degroote, S. Annerel, and J. Vierendeels. Stability analysis of Gauss-Seidel iterations in a partitioned simulation of fluid-structure interaction. *Computers & Structures*, 88(5–6):263–270, 2010.
- [5] J. Degroote, K.-J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers & Structures*, 87(11–12):793–801, 2009.
- [6] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in FSI applications. *Computers & Structures*, 86(23–24):2224–2234, 2008.

- [7] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. Performance of partitioned procedures in fluid-structure interaction. *Computers & Structures*, 88(7–8):446–457, 2010.
- [8] C. Farhat, K.G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17–18):1973–2001, 2006.
- [9] A. Fazzolari, N. Gauger, and J. Brezillon. Efficient aerodynamic shape optimization in MDO context. *Journal of Computational and Applied Mathematics*, 203(2):548–560, 2007.
- [10] C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24–25):3247–3270, 2001.
- [11] N. Gauger and A. Fazzolari. *MegaDesign and MegaOpt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design, Results of the closing symposium of the MegaDesign and MegaOpt projects, Braunschweig, Germany, 23 - 24 May, 2007*, volume 107 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, chapter Adjoint Methods for Coupled CFD-CSM Optimization, pages 237–246. Springer-Verlag, Berlin Heidelberg, 2009.
- [12] M. Gee, U. Küttler, and W.A. Wall. Truly monolithic algebraic multigrid for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 85(8):987–1016, 2011.
- [13] J.-F. Gerbeau and M. Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(4):631–648, 2003.
- [14] A. Griewank and A. Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, 2000.
- [15] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1–2):1–23, 2004.
- [16] M. Hojjat, E. Stavropoulou, T. Gallinger, U. Israel, R. Wüchner, and K.-U. Bletzinger. Fluid-structure interaction in the context of shape optimization and computational wind engineering. In H.-J. Bungartz, M. Mehl, and M. Schäfer, editors, *Fluid-Structure Interaction II: Modelling, Simulation, Optimization*, Lecture Notes in Computational Science and Engineering, pages 351–381. Springer, Berlin Heidelberg, 2010.



- [17] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23–26):2087–2104, 2004.
- [18] U. Küttler and W.A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.
- [19] F. Li and Z. Sun. A finite difference scheme for solving the Timoshenko beam equations with boundary feedback. *Journal of Computational and Applied Mathematics*, 200(2):606–627, 2007.
- [20] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, 2004.
- [21] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6(1):33–62, 2005.
- [22] K. Maute, M. Nikbay, and C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *American Institute of Aeronautics and Astronautics Journal*, 39(11):2051–2061, 2001.
- [23] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional nonlinear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, 2003.
- [24] C. Michler, E.H. van Brummelen, and R. de Borst. An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 47(10-11):1189–1195, 2005.
- [25] D.P. Mok, W.A. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. In K.-J. Bathe, editor, *Computational Fluid and Solid Mechanics*, pages 1325–1328. Elsevier, 2001.
- [26] E. Nielsen, B. Diskin, and N. Yamaleev. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. In *19<sup>th</sup> AIAA Computational Fluid Dynamics Conference*, pages 1–22, San Antonio, TX, USA, 22–25 June 2009. AIAA 2009-3802.
- [27] K. Palaniappan, P. Sahu, J. Alonso, and A. Jameson. Active flutter control using an adjoint method. In *44<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit*, pages 1–11, Reno, NV, USA, 9–12 January 2006. AIAA 2006-844.
- [28] K. Palaniappan, P. Sahu, J. Alonso, and A. Jameson. Design of adjoint based laws for wing flutter control. In *47<sup>th</sup> AIAA Aerospace Sciences Meeting*, pages 1–20, Orlando, FL, USA, 5–8 January 2009. AIAA 2009-148.

- [29] D.I. Papadimitriou and K.C. Giannakoglou. Direct, adjoint and mixed approaches for the computation of hessian in airfoil design problems. *International Journal for Numerical Methods in Fluids*, 56(10):1929–1943, 2008.
- [30] A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: problems, models and methods. *Computing and Visualization in Science*, 2(4):163–197, 2000.
- [31] M.P. Rumpfkeil and D.W. Zingg. The optimal control of unsteady flows with a discrete adjoint method. *Optimization and Engineering*, 11(1):5–22, 2010.
- [32] J. Sternberg and A. Griewank. *Automatic Differentiation: Applications, Theory, and Implementations*, volume 50 of *Lecture Notes in Computational Science and Engineering*, chapter Reduction of Storage Requirement by Checkpointing for Time-Dependent Optimal Control Problems in ODEs, pages 99–110. Springer-Verlag, Berlin Heidelberg, 2006.
- [33] A. Stück, F. Camelli, and R. Löhner. Adjoint-based design of passive and active shock mitigation devices. In *48<sup>th</sup> AIAA Aerospace Sciences Meeting*, pages 1–30, Orlando, FL, USA, 4–7 January 2010. AIAA 2010-1430.
- [34] A. Stück, F. Camelli, and R. Löhner. Adjoint-based design of shock mitigation devices. *International Journal for Numerical Methods in Fluids*, 64(4):443–472, 2010.
- [35] E.H. van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *Journal of Applied Mechanics*, 76(2):021206–1–7, 2009.
- [36] J. Vierendeels, K. Dumont, E. Dick, and P.R. Verdonck. Analysis and stabilization of fluid-structure interaction algorithm for rigid-body motion. *AIAA Journal*, 43(12):2549–2557, 2005.
- [37] I. Vignon-Clementel, C. Figueroa, K. Jansen, and C. Taylor. Outflow boundary conditions for 3D simulations of non-periodic blood flow and pressure fields in deformable arteries. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(5):625–640, 2010.
- [38] Q. Wang, P. Moin, and G. Iaccarino. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. *SIAM Journal on Scientific Computing*, 31(4):2549–2567, 2009.
- [39] R. Wüchner, A. Kupzok, and K.-U. Bletzinger. A framework for stabilized partitioned analysis of thin membrane-wind interaction. *International Journal for Numerical Methods in Fluids*, 54(6–8):945–963, 2007.