Computers and Fluids 142 (2017) 109–117

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Computational analysis of flow-driven string dynamics in turbomachinery

Kenji Takizawa [a], Tayfun E. Tezduyar [b,*], Hitoshi Hattori [a]

[a] Department of Modern Mechanical Engineering, Waseda University, 1-6-1 Nishi-Waseda, Shinjuku-ku, Tokyo 169-8050, Japan
[b] Mechanical Engineering, Rice University, 6100 Main Street, Houston, TX 77005, USA

ARTICLE INFO

Article history:
Received 22 December 2015
Accepted 23 February 2016
Available online 18 March 2016

Keywords:
Turbomachinery
Fan
String dynamics
Space–Time Variational Multiscale method
ST-VMS
ST Slip Interface method
ST-SI
Isogeometric Analysis
IGA
Higher-order functions

ABSTRACT

We focus on computational analysis of flow-driven string dynamics. The objective is to understand how the strings carried by a fluid interact with the solid surfaces present and get stuck on or around those surfaces. Our target application is turbomachinery, such as understanding how strings get stuck on or around the blades of a fan. The components of the method we developed for this purpose are the Space–Time Variational Multiscale (ST-VMS) and ST Slip Interface (ST-SI) methods for the fluid dynamics, and a one-way-dependence model and the Isogeometric Analysis (IGA) for the string dynamics. The ST-VMS method is the core computational technology and it also has the features of a turbulence model. The ST-SI method allows in a consistent fashion slip at the interface between the mesh covering a spinning solid surface and the mesh covering the rest of the domain, and with this, we maintain high-resolution representation of the boundary layers near spinning solid surfaces such as fan blades. With the one-way-dependence model, we compute the influence of the flow on the string dynamics, while avoiding the formidable task of computing the influence of the string on the flow, which we expect to be small. The IGA for the string dynamics gives us not only a higher-order method and smoothness in the structure shape, but also smoothness in the fluid dynamics forces calculated on the string. To demonstrate how the method can be used in computational analysis of flow-driven string dynamics, we present the pilot computations we carried out, for a duct with cylindrical obstacles and for a ventilating fan.

1. Introduction

In turbomachinery, objects carried by the fluid can sometimes have a negative effect on the rotor. For example, a piece of string carried by the fluid can get stuck on or around the blades of a fan, possibly hindering the rotor motion. In this article, we focus on computational analysis of flow-driven string dynamics. Our objective is to enable a better understanding of how the strings carried by a fluid interact with the solid surfaces present and get stuck on or around those surfaces. The main components of the method we developed for this purpose are the Space–Time Variational Multiscale (ST-VMS) and ST Slip Interface (ST-SI) methods for the fluid dynamics, and a one-way-dependence model and the Isogeometric Analysis (IGA) for the string dynamics.

Because a string is a very thin object, its influence on the flow will be very small. With the one-way-dependence model, we compute the influence of the flow on the string dynamics, while avoiding the formidable task of computing the influence of the string on the flow. The one-way-dependence model has been used in other contexts of computational engineering analysis. The examples we are familiar with are calculating the aerodynamic forces acting on the suspension lines of spacecraft parachutes [1–3] and calculating the forces acting on the particles in particle-laden flows [4,5]. In the first example the suspension lines are assumed to have no influence on the flow, and in the second example the particles are assumed to have no influence on the flow. In our case here, we first compute the flow field and store the time-dependent flow data, and then compute several patterns of string dynamics.

In computing the flow field, we use the ST-VMS method [6,7]. This is the VMS version of the Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [8–10]. The DSD/SST method is a moving-mesh method. The VMS components of the ST-VMS method are from the residual-based VMS (RBVMS) method given in [11–14]. The ALE-VMS method [15,16] is the VMS version of the Arbitrary Lagrangian–Eulerian (ALE) finite element method [17]. The ALE-VMS method was first presented in [18]. The RBVMS and ALE-VMS methods have been used successfully for different classes of problems (see, for example, [3,16,18–39]). The stabilization components of the original DSD/SST method are

* Corresponding author. Tel.: +1 7133486051.
E-mail addresses: tezduyar@tafsm.org, tezduyar@gmail.com (T.E. Tezduyar).

http://dx.doi.org/10.1016/j.compfluid.2016.02.019

the Streamline-Upwind/Petrov-Galerkin (SUPG) [40] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [8] stabilizations, and for that the method is now also called “ST-SUPS” (the acronym was coined in [3]). The ST-VMS method is essentially an “augmented” version of the ST-SUPS method, with two additional stabilization terms beyond the three that the ST-SUPS method has (see [3,6,7]). The classes of problems the ST-VMS method has been successfully applied to include wind-turbine aerodynamics [3,24,31,41–46], flapping-wing aerodynamics [3,44,45,47–53], cardiovascular fluid mechanics [44,45,51,54–58], spacecraft aerodynamics [59,60], thermo-fluid analysis of ground vehicles and their tires [61], and flow analysis for the turbine part of a turbocharger [62].

One of the desirable features of the ST methods is being able to use higher-order basis functions in time, including the non-uniform rational B-spline (NURBS) basis functions. There are now many publications reporting successful use of NURBS as spatial basis functions (for examples of use in conjunction with the RB-VMS and ALE-VMS methods, see again [3,16,18–39]). The ST-VMS method (and also the ST-SUPS method), when combined with higher-order NURBS basis functions in time, provides a more accurate representation of the motion of the solid surfaces, and a mesh motion consistent with that. It also provides more efficiency in temporal representation of the motion and deformation of the volume meshes, and better efficiency in remeshing. These desirable features have enabled computational analysis in flapping-wing aerodynamics [3,44,45,47–53], separation aerodynamics of spacecraft [59], wind-turbine aerodynamics [31,43–46], and thermo-fluid mechanics of ground vehicles and their tires [61]. The ST framework and NURBS in time also enable, with the “ST-C” method, extracting a continuous representation from the computed data and, in large-scale computations, efficient data compression [61,63].

With the moving-mesh methods, by moving the fluid mechanics mesh to follow a fluid–solid interface, we can control the mesh resolution near the interface, have high-resolution representation of the boundary layers, and obtain accurate solutions in such critical flow regions. We want to be able to do that also in flow problems with a spinning solid surface, such as the rotor of a fan. We want the mesh covering the spinning solid surface to spin with it so that we maintain the high-resolution representation of the boundary layers. That requires something special at the interface between the spinning mesh and the rest of the mesh. That was first accomplished in the ST framework with the Shear–Slip Mesh Update Method (SSMUM) [64–66], which was introduced in [64,65] and named “SSMUM” in [66]. Later it was accomplished also with the ST/NURBS Mesh Update Method (STNMUM), which was introduced in [47–49] and named in [43]. The STNMUM is more general than the SSMUM and simpler to use. It was successfully used in [43] in ST-VMS computation of flow past a wind-turbine rotor, with the tower included in the model. In the STNMUM, NURBS basis functions are used for the temporal representation of the spinning motion, mesh motion and also in remeshing. The rotor motion is represented by quadratic temporal NURBS basis functions, with sufficient number of temporal patches for a full rotation. With that, we can represent the circular paths associated with the rotor motion exactly. With an added “secondary mapping” [3,6,7,47], we can also specify a constant angular velocity corresponding to the invariant speeds along those paths.

The ST-SI method is the most recent ST method where the mesh covering a spinning solid surface spins with it and maintains the high-resolution representation of the boundary layers. The starting point in the development of the ST-SI method was the version of the ALE-VMS method designed for computations with “sliding interfaces” [27,67]. This ALE-VMS version has been used successfully in a number of computations with spinning solid surfaces [27,29,32,33,67]. In the ST-SI method, interface terms similar to those in the ALE-VMS version are added to the ST-VMS formulation to account for the compatibility conditions for the velocity and stress. While having high-resolution representation of the boundary layers near a spinning solid surface, by using NURBS functions in temporal representation of the spinning motion, the ST-SI method has exact representation of the circular paths associated with the spinning.

As we compute the flow field, we store the computed time-dependent data with a special data compression method based on the ST-C method [63]. With the ST-C method, we can represent the data with fewer temporal control points, resulting in reduced computer storage cost. In one of the two ST-C versions introduced in [63], the continuous representation is extracted by projection from a solution already computed. Because we use a successive-projection technique (SPT), with a small number of temporal NURBS basis functions at each projection, the extraction can take place as the original solution is being computed, without the need to first complete the computation and store all that data. This version was named “ST-C-SPT” in [63]. In the work reported in this article, the large time-history data from the flow field computation is stored using the ST-C-SPT method.

The IGA is used in the string dynamics with higher-order basis functions. This gives us a higher-order method and smoothness in the structure shape. It also gives us smoothness in the fluid dynamics forces calculated on the string. Furthermore, although the bending effect is small for thin filaments considered in this work, it was shown in [68] that the use of higher-order smooth basis functions enables one to incorporate bending action into the string formulation without introducing rotational degrees of freedom. These bending terms, although not considered in this work, can provide additional numerical stabilization for the string.

We describe the ST-VMS and ST-SI methods in Section 2. In Section 3, we present the pilot computations we carried out, for a duct with cylindrical obstacles and for a ventilating fan. The concluding remarks are given in Section 4.

2. Method

This section is mostly a condensed version of the related parts of the methods presented in [46]. The flow field is computed with the ST-VMS method:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&- \int_{(P_n)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \mathrm{d}Q \\
&+ \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h \right) + \nabla q^h \right] \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC} \nabla \cdot \mathbf{w}^h \rho r_\mathrm{C}(\mathbf{u}^h) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \tau_\mathrm{SUPS} \mathbf{w}^h \cdot \left(\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \nabla \mathbf{u}^h \right) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}^2}{\rho} \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \left(\nabla \mathbf{w}^h \right) \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q = 0, \qquad (1)
\end{aligned}$$

where

$$\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h), \qquad (2)$$

$$r_\mathrm{C}(\mathbf{u}^h) = \nabla \cdot \mathbf{u}^h \qquad (3)$$