Shape Sensitivity Analysis for Coupled Fluid-Solid Interaction Problems

Koorosh Gobal

November 28, 2015

Contents

List of Figures

List of Tables

Abstract

In this paper, a robust continuum sensitivity formulation for the shape sensitivity analysis of weakly coupled aero-structural systems is derived. In this method, the solid boundaries are modelled using the immersed boundary approach. This simplifies the grid generation for the complex and deforming geometries since the computational mesh does not need to conform to the boundaries. The sensitivity analysis consists of differentiating the continuum form of the governing equations where the effect of the solid boundaries are modelled as additional forcing terms in these equations. By differentiating the governing equations, it is possible to reuse the operators utilized for solving the governing equations. Therefore, there is no need to develop new solvers for the solution of sensitivity response. This methodology is applied to different demonstration problems including flow over a cylinder and a simplified aeroelastic model of a wing. The wing structure is modelled as a beam with the lifting surface mounted at the tip where the load is transfered to the structure through the mounting point. The sensitivity results with this approach compares well with the complex step method results. Moreover, it is shown that the methodology is capable of handing complex shapes with high Reynolds numbers.

Chapter 1

Introduction

1.1 Motivation

Fluid-structure interaction (FSI) problems play important role in many scientific and engineering fields, such as automotive, aerospace, and biomedical industry. Despite the wide application, a comprehensive study of FSI systems still remains a challenge due to their strong nonlinearity and multidisciplinary nature. For most FSI problems, analytical solutions to are not available, and physical experiments are limited in scope. Therefore, to get more insight in the physics involved in the complex interaction between fluids and solids, numerical simulations are used. The numerical solutions are conducted based on Computational Fluid Dynamics (CFD) models for the flow and Finite Element Analysis (FEA) for the structural response. Nevertheless, the prohibitive amount of computations has been one of the major issues in the design optimization of such coupled multidisciplinary systems. The other bottle neck is generating an appropriate computational domain that represents the fluid and solid regions. The effort and time required to take a geometry from a CAD package, clean up the model, and generate a mesh is usually a large portion of the overall human time required for the simulation. This cannot be automated for complex and moving domains. The Immersed Boundary (IB) method, reduces the amount of time needed for the fluid flow simulations and provides fast results by directly addressing the challenges associated with this issue.

Due to the large amount of computations involved in the FSI simulation, the gradient based methods are the best candidates for design optimization of such problems. Sensitivity analysis is the integral part of gradient based methods. Although there are analytical techniques for efficient and accurate sensitivity calculation, they have not yet implemented in commercial CFD packages. Therefore, most gradient optimization techniques relay on finite difference method for sensitivity calculation when solving FSI problem that are prone to errors.

The motivation for the research proposed in this document is in two areas.

First, we want to have sensitivity analysis capabilities that can treat the solvers as black-box. This means that we can solve both the governing equations and sensitivity response using the same code. Second, a robust analysis technique for the coupled FSI system based on IB method is formulated. The current approach of IB is not suited for the sensitivity analysis due to the discontinuities in its formulation. This will be explained in more details in the following Chapters.

1.2 Literature Review

1.2.1 Sensitivity Analysis

Sensitivity analysis consists of computing the derivatives of solution of the governing equations, i.e. displacement, velocity, or pressure, with respect to one of several independent design variables, i.e. shape of boundaries or size of elements. There various applications for sensitivity information such as improving the accuracy of surrogate models as in gradient enhanced Kriging [1] or uncertainty quantification [2]. However, our main motivation is the use of this information in gradient-based optimization. The calculation of gradients is often the most expensive step in the optimization cycle therefore, using efficient methods for accurate calculation of sensitivities are vital to the optimization process. As shown in Figure 1.1, methods for sensitivity calculation can be put into three methods: i) numerical, ii) analytical, and iii) automatic differentiation.

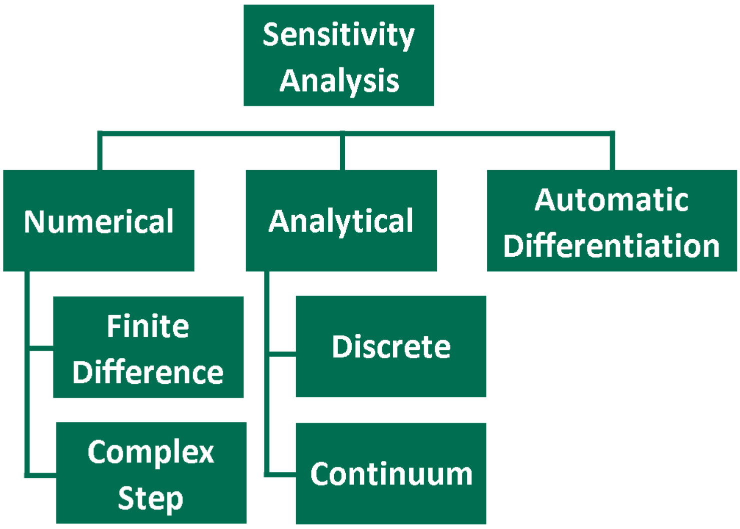


Figure 1.1: Sensitivity calculation techniques.

The Finite Difference (FD) method is probably the easiest method to implement for calculating the sensitivity of a variable. The fact that they can be implemented even when a given computational model is a black box makes most gradient based optimzation algorithms perform finite differences by default when the used does not provide the required gradients. However, the computational cost associated with finite difference for large systems can become very large. For a system with n number of design variables, the analysis needs to be done $n+1$ time to calculate the design sensitivities. Furthermore, to ensure the accuracy, convergence study needs to be done for selecting the appropriate step size for finite difference. The inaccuracy of finite differencing could result in convergence difficulties and inaccurate optimum results. On the other hand, Complex Step (CS) method avoids the loss of precision in finite differences approximation of sensitivities by employing complex arithmetic [3]. The complex step derivative is calculated as shown in Equation (1.1).

$$\mathcal{F}'\left(u;b\right)=\frac{\mathrm{Im}\left[\mathcal{F}\left(u;b+ih\right)\right]}{h} \tag{1.1}$$

This means that we perturb the design variable by an imaginary value of ih and then look at the imaginary portion of the resulting response. Using the complex step method, we can choose a small step size for h without loosing accuracy. However, many commercial packages such as ANSYS or Nastran cannot handle complex arithmetic which makes the implementation of complex step method infeasible. Moreover, the high cost of finite difference is still associated with the complex method as well.

Automatic differentiation (AD) is based on the systematic application of the differentiation to computer programs [4]. In the AD approach, the chain rule of differentiation is applied to every line in the program. This assumes that the computer program consists of a sequence of explicit functions that act successively on some variables. Therefore, by differentiating each of these functions and applying the chain rule, it is possible to calculate the sensitivities.

There has been many research on utilizing AD for optimization. Bischof et al. used AD for calculating the sensitivities using a CFD solver. They used ADIFOR for differentiating the source code of their CFD code (TLNS3D) which later used for calculating the sensitivity of a transonic flow to change in the boundary conditions. Hascout et al., also used AD for a sonic boom reduction under a supersonic aircraft. However, in all of these works, it is needed to have access to the source code and modify the solver extensively to calculate the sensitivities. This make the use of this method for general purpose codes infeasible since the source code is usually not available.

The short comings of numerical and AD techniques, fuels the research for more sophisticated methods for sensitivity calculation which are generally known as analytical methods. Formulation of the analytical sensitivities requires derivation of analytic sensitivity equations. These are obtained by differentiating the governing equations with respect to design variables such as shape of the boundaries. Analytical methods can be further categorized based on how the sensitivity equations derived and solved. Typically the continuum equations are solved using an approximate method which discretizes the governing equations. The Discrete Sensitivity Analysis (DSA) technique, differentiates the discretized governing equations with respect to design variables to get the analytical sensitivity equations [5]. This system of equations is later solved for analytical sensitivities.

The discrete sensitivity analysis has been historically the method of choice to calculate the high-accurate sensitivity when the details of analysis are available [6]. This method has been adopted by the structural optimization community, and been applied to fluid-solid interaction problems as well. Reuther et al., used the discrete method for aerodynamic shape optimization of a complex aircraft configuration [7]. They used Euler flow as the aerodynamics theory where they optimized different configurations for transonic and supersonic regimes. However, they only focused on the flow. Martins et al., developed an adjoint method for sensitivity analysis for an aero-structural aircraft design framework where the sensitivities were computed using a coupled adjoint approach. The framework was used on a supersonic business jet to calculate the sensitivity of drag with respect to the shape (OML) of the aircraft. In their work, the discretization details of the solver needs to be known to calculate the sensitivities. As a matter of fact, source code modification is essential in all the other related research in the area as well [8, 9, 10].

As mentioned before, sensitivity calculation is the essential part of gradient based optimization. However, many general purpose solvers do not have this capability. This is even more prominent in the fluid mechanics community where none of the available commercial software packages such as ANASYS Fluent or CFX has analytical sensitivity analysis capabilities. In optimization loop, they relay on finite difference as a method for sensitivity calculation. As mentioned in the previous paragraph, for the discrete sensitivity methods, intimate knowledge and access to the source codes of these software is needed to calculate the sensitivities. However, the source code is usually inaccessible and very complex. Therefore, there is a great interest in sensitivity calculation techniques which require minimum knowledge of the analysis code. This can be achieved using the sensitivity formulation that operate on the governing equations before they are discretized. These method are commonly known as Continuum Sensitivity Methods [11].

The Continuum Sensitivity Analysis (CSA), involves solving a set of partial differential equations named the continuum sensitivity equations (CSEs) to get the analytical sensitivities. When deriving the CSEs, the governing equations can either be differentiated in local or total form. The difference between local and total differentiation depends on the Lagrangian or Eulerian representation of the governing equations. In the general case of continuum mechanics, displacement and velocity are vector valued functions. I any application, we have the choice of writing these vectors as functions of the position of material particles before deformation. This is called the Lagrangian description of motion and is really helpful for visualizing the deformations. This technique is mostly adopted in solid mechanics where we track the material points as the deformations are usually assumed to be small. However, in the fluid flow problems, since it is generally hard to identify a reference configuration and the deformations are large, it is preferable to write the displacement and velocities as functions of the deformed position of the particles. These quantities are now defined for a particular point in space that does not move with the particles. This is called the Eulerian description of motion. As CSA was matured over the year, the computational fluids community adopted the local CSEs, since its formulation is consistent with the governing equations [12, 13]. The structural optimization community adopted the total formulation for the sensitivities since its formulation was consistent with the Lagrangian formulation of structural mechanics. Nevertheless, the total and local formulations for the sensitivities can be converted to each other. This will be explained in more details in next chapter.

Overally, the sensitivity analysis in general is more matured is the concept of structural optimization than for problems of fluid dynamics. Nevertheless, neither of these disciplines do not typically employ CSA for the sensitivity calculation. Aurora and Haug [14], followed by Dems and Mroz [15], were among the first to develop the concept of CSA for structural optimization. They modeled the sensitivities as functionals therefore they were able to convert the sensitivity integrals over the entire domain to the integrals over the boundaries. Although using the approach it is possible to reduce the cost of the simulation, accurate values of functionals are required at the boundaries. This is not always achievable especially for finite element analysis where the solution accuracy drops near the boundaries. The lack of applicability of CSA in structural optimization is due to the complicated definition of the boundary conditions and maturity of discrete methods for the sensitivity calculation.

The first application of CSA in optimization problems for fluid dynamics is the work by Borggaard and Burns [16] for shape sensitivity analysis of inviscid supersonic flows over rigid bodies. Stanley and Stewart [17] applied CSA in a fluid mechanics discipline with a goal for aerodynamic design. Pelletier and Etienne have applied CSA to numerous fluid-structure interaction (FSI) problems [18] focused mainly on sensitivities of fluid flow parameters near the structure. Liu and Canfield have employed CSA for shape optimization of nonlinear structures subject to an aeroelastic gust response [19]. They used the finite element method to solve the potential flow around an airfoil and applied CSA to find the airfoil pressure coefficient sensitivity with respect to the maximum camber.

In almost all of the research done on sensitivity calculation of flow field to shape design variable, body conformal grids were used. The conforming mesh methods consider the interface conditions as physical boundary conditions, which treat the interface location as part of the solution and requires meshes that conform to the interface. Using this approach it is possible to represent the solid boundary shape with good accuracy. However, due to the coupling of fluid mesh topology and solid boundary shape, with the movement and/or deformation of the solid structure, re-meshing (or mesh-updating) is needed. Although conforming mesh methods have been widely used in many FSI problems, they are cumbersome, if not impossible, to apply to problems with large deformations [20]. The other shortcoming of body-conformal grids, is the effect of mesh deformation on the sensitivity analysis. Since the computational domain is affected by change in the shape of the boundaries, it is required to calculate the mesh sensitivities as well [21]. This adds to the computational cost of calculating sensitivities.

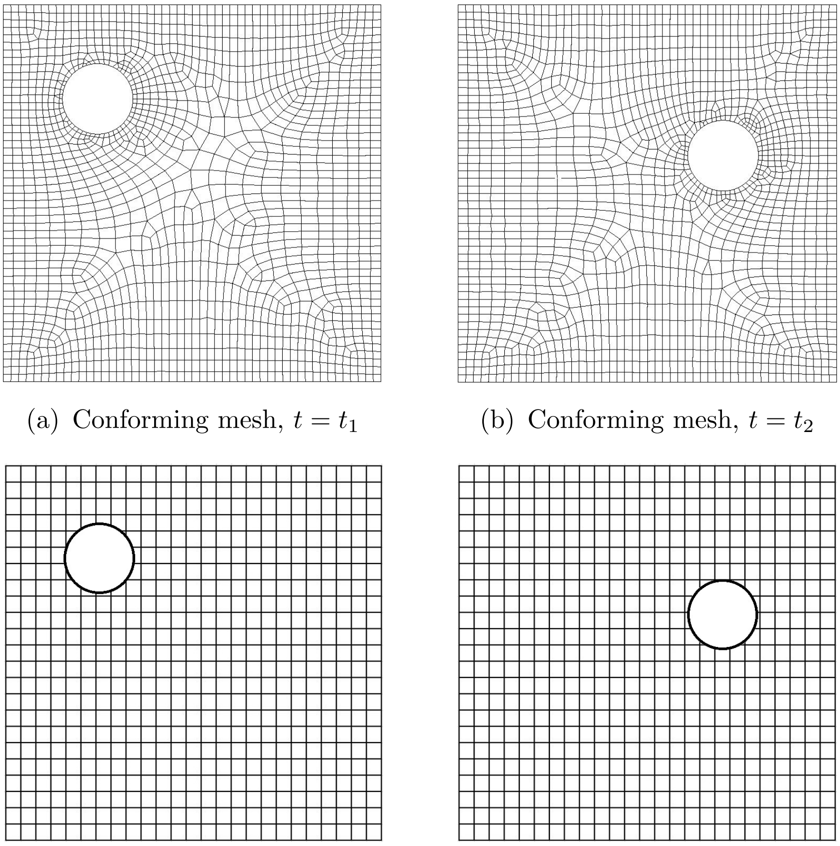
The shortcoming of a robust grid generation and the additional cost of calculating mesh sensitivities motivated an important research effort to develop a method that does not require fluid domain mesh modification for the optimization iterations. On of the possible candidates to achieve this goal, is the use of Immersed Boundary (IB) methods to decouple the fluid mesh from the shape of solid domain.

1.2.2 Immersed Boundary Method

Traditionally the simulation methods for flow over complex bodies are based on a body-fitted multi-block or unstructured grid methods. However, in the last decade another class of techniques, called immersed boundary methods, have been introduced to model the flow around solid boundaries. The term immersed boundary is first used by Peskin [22] to simulate the blood flow through heart valves. What distinguishes this method from the other methods of representing the solid boundaries is that the flow is solved on the Cartesian grid that does not conform to the solid boundaries. Therefore, the mesh generation is greatly simplified and in case of moving/deforming boundaries, there is no need to update the mesh during the simulation. A separate formulation is used to impose the effect of the boundaries on the solution of the equations.

Consider the simulation of flow past a circular cylinder as shown in Figure 1.2. Generating structured or unstructured grids is achieved in two steps. First, a surface grid covering the boundaries is generated. This is then used as a boundary condition to generate a grid in the volume occupied by the fluid. The differential form of the governing equations is then transformed to a curvilinear coordinate system aligned with the grid lines [23]. If a finite-volume technique is employed, then the integral form of the governing equations is discretized and the geometrical information regarding the grid is incorporated directly into the discretization. If an unstructured grid is employed, then either a finite-volume or a finite-element methodology can be used.

Non-body conformal Cartesian grid can also be utilized for this simulation as shown in Figure 1.2. In this approach the IB would still be represented through some means such as a surface grid, but the Cartesian volume grid would be generated with no regard to this surface grid. Thus, the solid boundary would



(a) Conforming mesh, $t = t_1$

(b) Conforming mesh, $t = t_2$

(c) Nonconforming mesh, $t = t_1$

(d) Nonconforming mesh, $t = t_2$

Figure 1.2: Example of conforming and nonconforming meshes.

cut through this Cartesian volume grid. Because the grid does not conform to the solid boundary, incorporating the boundary conditions would require modifying the equations in the vicinity of the boundary. Assuming that such a procedure is available, the governing equations would then be discretized using a finite-difference, finite-volume, or a finite-element technique without resorting to coordinate transformation or complex discretization operators.

Depending on how the effect of solid boundaries are imposed, IB methods can be divided into four categories: i) Continuous forcing, ii) discrete forcing, iii) ghost cell, and iv) cut-cell method.

Continuous Forcing Method

This is the originaly used by Peskin [22] and later developed by others researchers [24, 25, 26]. In this approach, the boundary configuration is described by a curve $x(s,t)$ (Lagrangian nodes) where the location of each point on this boundary is governing by its equations of motion. The forces that the curve $x(s,t)$ exerts on the fluid is calculated buy the constitutive law of the curve $x(s,t)$ that relates the displacements calculated by the equation of motion to stress values. These stress values are transferred to the Navier-Stokes (NS) (Eulerian nodes) for the fluid by means of a Dirac delta function. Practical implimentation of this method resets in representing the Dirac delta function as a discrete function that has the same properties. This method is applied to variety of problems, including Cardiac blood flow [27], animal locomotion [28], multiphase flows [29], and particle sedimentation [30].

(TALK ABOUT VIRTUAL BOUNDARY METHOD)

Bibliography

[1] Han, Z.-H., Görtz, S., and Zimmermann, R., “Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function,” *Aerospace Science and Technology*, Vol. 25, No. 1, 2013, pp. 177–189.

[2] Pettit, C. L., “Uncertainty quantification in aeroelasticity: recent results and research challenges,” *Journal of Aircraft*, Vol. 41, No. 5, 2004, pp. 1217–1229.

[3] Martins, J. R., Sturdza, P., and Alonso, J. J., “The complex-step derivative approximation,” *ACM Transactions on Mathematical Software (TOMS)*, Vol. 29, No. 3, 2003, pp. 245–262.

[4] Naumann, U., *The art of differentiating computer programs: an introduction to algorithmic differentiation*, Vol. 24, Siam, 2012.

[5] Choi, K. K. and Kim, N.-H., *Structural sensitivity analysis and optimization 1: linear systems*, Springer Science & Business Media, 2006.

[6] Arora, J. S. and Haug, E. J., “Methods of design sensitivity analysis in structural optimization,” *AIAA journal*, Vol. 17, No. 9, 1979, pp. 970–974.

[7] Reuther, J. J., Jameson, A., Alonso, J. J., Rimllnger, M. J., and Saunders, D., “Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2,” *Journal of aircraft*, Vol. 36, No. 1, 1999, pp. 61–74.

[8] Gamboa, P., Vale, J., P. Lau, F., and Suleman, A., “Optimization of a morphing wing based on coupled aerodynamic and structural constraints,” *AIAA journal*, Vol. 47, No. 9, 2009, pp. 2087–2104.

[9] Pandya, M. J. and Baysal, O., “Gradient-based aerodynamic shape optimization using alternating direction implicit method,” *Journal of aircraft*, Vol. 34, No. 3, 1997, pp. 346–352.

[10] Kim, H.-J., Sasaki, D., Obayashi, S., and Nakahashi, K., "Aerodynamic optimization of supersonic transport wing using unstructured adjoint method," *AIAA journal*, Vol. 39, No. 6, 2001, pp. 1011–1020.

[11] Haftka, R. T. and Adelman, H. M., "Recent developments in structural sensitivity analysis," *Structural optimization*, Vol. 1, No. 3, 1989, pp. 137–151.

[12] Borggaard, J. and Burns, J., "A PDE sensitivity equation method for optimal aerodynamic design," *Journal of Computational Physics*, Vol. 136, No. 2, 1997, pp. 366–384.

[13] Hristova, H., Etienne, S., Pelletier, D., and Borggaard, J., "A continuous sensitivity equation method for time-dependent incompressible laminar flows," *International Journal for numerical methods in fluids*, Vol. 50, No. 7, 2006, pp. 817–844.

[14] Arora, J. and Haug, E., "Methods of Design Sensitivity Analysis in Structural Optimization," *AIAA Journal*, Vol. 17, No. 9, 1979, pp. 970–974.

[15] Dems, K. and Mroz, Z., "Variational approach to first- and second-order sensitivity analysis of elastic structures," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 4, 1985, pp. 637–661.

[16] Borggaard, J. and Burns, J., *A sensitivity equation approach to shape optimization in fluid flows*, Springer, 1995.

[17] Stanley, L. G. D. and Stewart, D. L., *Design Sensitivity Analysis: Computational Issues of Sensitivity Equation Methods*, Frontiers in applied mathematics, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2002.

[18] Etienne, S. and Pelletier, D., "A general approach to sensitivity analysis of fluid–structure interactions," *Journal of Fluids and Structures*, Vol. 21, No. 2, 2005, pp. 169–186.

[19] Liu, S. and Canfield, R. A., "Equivalence of continuum and discrete analytic sensitivity methods for nonlinear differential equations," *Structural and Multidisciplinary Optimization*, Vol. 48, No. 6, 2013, pp. 1173–1188.

[20] Sahin, M. and Mohseni, K., "An arbitrary Lagrangian–Eulerian formulation for the numerical simulation of flow patterns generated by the hydromedusa Aequorea victoria," *Journal of Computational Physics*, Vol. 228, No. 12, 2009, pp. 4588–4605.

[21] Liu, S. and Canfield, R. A., "Boundary velocity method for continuum shape sensitivity of nonlinear fluidstructure interaction problems," *Journal of Fluids and Structures*, Vol. 40, No. 0, 2013, pp. 284 – 301.

[22] Peskin, C. S., "Numerical analysis of blood flow in the heart," *Journal of computational physics*, Vol. 25, No. 3, 1977, pp. 220–252.

[23] Anderson, J. D. and Wendt, J., *Computational fluid dynamics*, Vol. 206, Springer, 1995.

[24] Saiki, E. and Biringen, S., "Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method," *Journal of Computational Physics*, Vol. 123, No. 2, 1996, pp. 450–465.

[25] Zhu, L. and Peskin, C. S., "Interaction of two flapping filaments in a flowing soap film," *Physics of Fluids (1994-present)*, Vol. 15, No. 7, 2003, pp. 1954–1960.

[26] Beyer, R. P. and LeVeque, R. J., "Analysis of a one-dimensional model for the immersed boundary method," *SIAM Journal on Numerical Analysis*, Vol. 29, No. 2, 1992, pp. 332–364.

[27] Peskin, C. S. and McQueen, D. M., "A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid," *Journal of Computational Physics*, Vol. 81, No. 2, 1989, pp. 372–405.

[28] Fauci, L. J. and Peskin, C. S., "A computational model of aquatic animal locomotion," *Journal of Computational Physics*, Vol. 77, No. 1, 1988, pp. 85–108.

[29] Kempe, T., Lennartz, M., Schwarz, S., and Fröhlich, J., "Imposing the free-slip condition with a continuous forcing immersed boundary method," *Journal of Computational Physics*, Vol. 282, 2015, pp. 183–209.

[30] Uhlmann, M., "An immersed boundary method with direct forcing for the simulation of particulate flows," *Journal of Computational Physics*, Vol. 209, No. 2, 2005, pp. 448–476.