Effective discrete adjoint OpenFOAM for volume and surface sensitivities

Arindam Sen* and Markus Towara* and Uwe Naumann†

Software and Tools for Computational Engineering, RWTH Aachen University, 52074, Germany

A discrete adjoint version of OpenFOAM is presented here. This version can be used to accurately compute sensitivities, i.e derivatives of suitable CFD objective functions such as drag, lift, pressure loss etc. with respect to desired design variables such as porosity or mesh points which can then be utilized for the purpose of gradient based design optimization.

I. Introduction

Adjoint methods for gradient based optimization are now widely used in CFD applications. Open source CFD tools such as OpenFOAM are used by a vast user base in both industry and academia for various applications. The objective of this work is to develop a discrete adjoint version of OpenFoam using Algorithmic Differentiation[1] that helps to compute efficient and accurate (up to machine precision) sensitivities, i.e. gradients of specific objective functions such as drag, lift, pressure loss etc. with respect to desired design variables[2].[3] This discrete adjoint version is obtained by overloading all the basic mathematical operations using a custom data type with the aid of operator overloading tool dco/c++.[4] Compared to continuous adjoints, this framework is flexible and easily adaptable. Also, because of the highly objected-oriented nature of the source code (in C++), source transformation tools like Tapenade[5] are not suitable. A black box adjoint usually incurs an unaffordable memory footprint. To overcome this limitation, several standard improvement techniques like binomial checkpointing,[6] symbolic differentiation of the embedded linear iterative solver,[7] reverse accumulation for steady state problems[8] were implemented. This framework is also extended to compute higher order adjoints. A discrete adjoint version of Foam-extend, a fork of OpenFOAM with additional functionalities, was also obtained. For incompressible steady-state problems, the adjoint of the coupled implicit solver was used to gain further performance improvement.[9] This framework was then tested on small to medium scale problems for a range of applications like ducted flows, external aerodynamics and conjugate heat transfer with qualitative validation against the continuous adjoint implementation.[10] The performance, results and challenges are subsequently discussed.

II. Discrete adjoint OpenFOAM

The optimization problem can be represented as $J : I\!R^n \rightarrow I\!R^m$, $J(\alpha) \rightarrow min!$, where n denotes the number of input variables (design space) and m denotes the number of output variables (dimension of the objective function). For the purpose of topology or shape optimization, $m = 1$ or $O(1)$ and $n \gg m$. For such a problem, the cost of obtaining the derivative, $\frac{\partial J}{\partial \alpha}$ using finite differences is $n * Cost(J)$ whereas that using adjoint is $m * Cost(J)$, thus becoming the obvious method of preference.

The basic method for obtaining a black box adjoint of a decoupled incompressible solver in OpenFOAM called *simpleFoam* using dco/c++ has previously been discussed.[2] This framework has then been extented to an unsteady solver and a fully-coupled implicit solver based on Foam-extend[3].[9]

A black box adjoint has limited usefulness in the context of relevant CFD problems pertaining to high memory footprint. Extension of this framework has been achieved by application of standard improvement techniques such as checkpointing and symbolic differentiation of the embedded linear iterative solver to obtain significant performance improvement[3].[9] Additionally for steady state problems, the application of reverse accumulation allows us to adjoin the last non-linear iteration step repeatedly by changing the inputs once the forward evaluations have converged contractively to a fixed point. This method allows

*PhD Candidate, Software and Tools for Computational Engineering, RWTH Aachen University, Germany

†Professor, Software and Tools for Computational Engineering, RWTH Aachen University, Germany

the adjoint solution to usually converge faster and does not need additional memory for storing the checkpoints.

This framework is consistently applicable for a variation of flow characteristics like laminar/turbulent, steady/unsteady, compressible/incompressible and hence has a major advantage over it's continuous counterpart which typically involves tedious mathematical derivations for different flow physics.

III. Results

The framework described above is tested on small to medium scale CFD applications. Fig 1. shows the volume sensitivities of an airduct design domain (Courtesy: Volkswagen AG), i.e derivatives of pressure loss between the inlet and outlet with respect to the porosity term α. The design domain consists of 5 Mio. volume cells and the primal solution is obtained using *simpleFoam* for a Reynolds number 10. Fig. 2 shows the surface sensitivities of a 3D Onera M6, i.e derivatives of drag force on wing with respect to mesh points at a Mach number of 0.84.

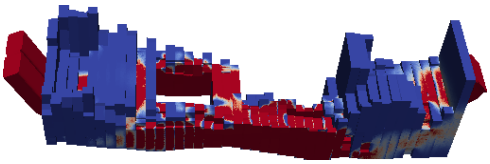

Figure 1. Positive volume sensitivities of airduct design domain. Blue color depicting region where optimization could potentially remove material

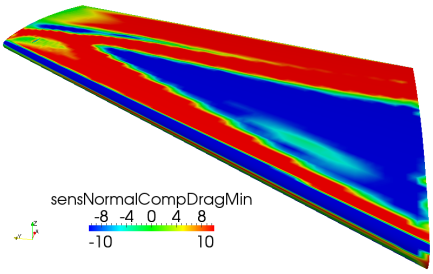

Figure 2. Surface sensitivities of ONERA M6. For drag minimization, Red color: inward displacement, Blue Color: outward displacement

References

[1]U. Naumann. The Art of Differentiating Computer Programs. An Introduction to Algorithmic Differentiation. , Chapter 1 and 2. SIAM, 2012

[2]M.Towara and U.Naumann. A discrete adjoint model for OpenFOAM. Procedia Computer Science , 18(0):429 438, 2013. 2013 International Conference on Computational Science

[3]A.Sen, M.Towara, and U.Naumann. A discrete adjoint version of an unsteady incompressible solver for OpenFOAM using algorithmic differentiation. WCCM XI ECCM V ECFD VI, Barcelona, 2014

[4]J. Lotz, K. Leppkes, and U. Naumann. dco/c++ -derivative code by overloading in C++. Aachener Informatik Berichte (AIB-2011-06), 2011

[5]L. Hascot and V. Pascual. The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. ACM Transactions On Mathematical Software , 39(3), 2013

[6]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):1945, March 2000.

[7]M. B. Giles. Collected matrix derivative results for forward and reverse mode algorithmic differentiation. In Advances in Automatic Differentiation , pages 3544. Springer, 2008

[8]B.Christianson. Reverse accumulation and attractive fixed points. Optimization methods and software, Vol. 3, No. 4, 1994, p. 311-326

[9]A. Sen, M. Towara, and U. Naumann, Sensitivity computation for ducted flows using adjoint of implicit pressure-velocity coupled solver based on Foam, EUROGEN 2015, 14-16 September 2015, Glasgow, UK .

[10]C. Othmer,E. de Villiers and H.G. Weller, Implementation of a continuous adjoint for topology optimization of ducted flows, AIAAA, Miami, 2007.