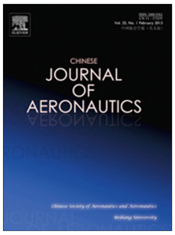
Chinese Journal of Aeronautics, (2017), **xxx**(xx): xxx–xxx

CSAA

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com



Automatic differentiation based discrete adjoint method for aerodynamic design optimization on unstructured meshes

Gao Yisheng, Wu Yizhao *, **Xia Jian**

College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Received 4 July 2016; revised 1 September 2016; accepted 30 November 2016

KEYWORDS

Automatic differentiation (AD);
Discrete adjoint;
Navier-Stokes equations;
Optimization;
Unstructured meshes

Abstract A systematic methodology for formulating, implementing, solving and verifying discrete adjoint of the compressible Reynolds-averaged Navier-Stokes (RANS) equations for aerodynamic design optimization on unstructured meshes is proposed. First, a general adjoint formulation is constructed for the entire optimization problem, including parameterization, mesh deformation and flow equations, which is followed by detailed formulations of matrix-vector products arising in the adjoint model. According to this formulation, procedural components of implementing the required matrix-vector products are generated by means of automatic differentiation (AD) in a structured and modular manner. Furthermore, a duality-preserving iterative algorithm is employed to solve flow adjoint equations arising in the adjoint model, ensuring identical convergence rates for the tangent and the adjoint models. A three-step strategy is adopted to verify the adjoint computation. The proposed method has several remarkable features: the use of AD techniques avoids tedious and error-prone manual derivation and code programming; duality is strictly preserved so that consistent and highly accurate discrete sensitivities can be obtained; and comparable efficiency to hand-coded implementation can be achieved. Upon the current discrete adjoint method, a gradient-based optimization framework has been developed and applied to a drag reduction problem.

1. Introduction

The advances in computational fluid dynamics (CFD) and high performance computers have enabled the coupling of CFD and numerical optimization techniques to effectively determine the optimum aerodynamic shape of complex configuration. Among several optimization algorithms for aerodynamic design problems, adjoint method[1] is now widely used since it allows one to compute the gradient (or derivatives)

---

* Corresponding author.
E-mail addresses: gaoyisheng@nuaa.edu.cn (G. Yisheng), wyzao@nuaa.edu.cn (W. Yizhao), jxia@nuaa.edu.cn (X. Jian).

Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER **Production and hosting by Elsevier**

http://dx.doi.org/10.1016/j.cja.2017.01.009
1000-9361

Please cite this article in press as: Yisheng G et al. Automatic differentiation based discrete adjoint method for aerodynamic design optimization on unstructured meshes, *Chin J Aeronaut* (2017), http://dx.doi.org/10.1016/j.cja.2017.01.009

of an objective function efficiently by solving an additional linear problem that is of the same order of magnitude of a single flow solution and essentially independent of the number of design variables. Two types of adjoint approaches have been established: continuous[2–5] and discrete.[6–10] In the continuous adjoint approach, the adjoint equations and the corresponding boundary conditions are first derived by the linearization of the governing partial differential equations (PDEs) and then discretized. This approach allows the flexible discretization of the derived PDEs, reducing the CPU and memory overheads. On the other hand, the discrete adjoint is constructed by reversing the above order of the linearization and the discretization. The advantage of the discrete adjoint approach is that one can obtain the exact discrete gradient of the objective function and verify it with great precision using the complex-step method[11] or dual number automatic differentiation (DNAD) method.[12] The discrete version is also conceptually straightforward. As will be illustrated in the following section, in finite dimensional vector space endowed with the Euclidean inner product where the discretized flow equations are defined, adjoint operator is equivalent to the transpose of the matrix form of the linear operator. Thus, in concept, the formulation of the discrete adjoint only involves transpose operation, which is much simpler than the continuous counterpart.

However, the implementation of the discrete adjoint method remains a challenging task due to the difficulty associated with the exact linearization of the underlying sophisticated numerical schemes and turbulence models. Hand-coded implementation is tedious and error-prone, resulting in lengthy code development time. Moreover, some approximations or simplifications are often made in hand-coded implementation, such as the neglect of the differentiation of artificial dissipation or the assumption of constant eddy-viscosity. These approximations or simplifications lead to inaccurate discrete gradient of the objective function, and may in turn affect the optimization process.[13] One promising way to address this implementation issue is the use of automatic differentiation (AD).[14] If the source code of the original flow solver is provided, AD tools can generate codes capable of calculating the required derivatives in an automatic manner. AD tools usually offer two different modes of operation: the forward (or tangent) mode and the reverse (or adjoint) mode. In the forward mode, the derivatives are propagated in the same direction of the original flow solver, while in the reverse mode the propagation of the derivatives is reversed. The reverse mode of AD technique is analogous to the discrete adjoint and the computational cost is independent of the number of input variables, thereby very suitable for the construction of the discrete adjoint code. Unfortunately, a ‘black-box’ application of the reverse mode to the entire flow solver will usually result in very inefficient codes with excessive CPU and memory overheads because of the high computational cost and storage of intermediate variables in the reversed propagation of the derivatives. To tackle this problem, a strategy of selective application of AD tools to the original CFD code has been promoted and adopted in both structured and unstructured solvers.[15–19] AD tools are piecewise applied to the original solver, and the derived codes are manually assembled in an appropriate reverse order, eliminating unnecessary computation and memory consumption which occur in ‘black-box’ application of AD tools. But the existing AD-assisted discrete adjoint solvers on unstructured grids are still not as efficient as hand-coded version.

In this paper, the main objective is to establish a systematic methodology for the development of accurate and efficient discrete adjoint solver on unstructured meshes, including adjoint formulation, detailed implementation, solution of adjoint equations and verification. First, a general adjoint formulation proposed by Mavriplis[20] is adopted to construct the adjoint model of the entire optimization problem, including parameterization, mesh deformation and flow equations. And then detailed formulations of matrix-vector products arising in the adjoint model are presented in a form suitable for the subsequent application of AD tools. Based on these formulations, procedural components of implementing the matrix-vector products are generated by using the reverse mode of AD tools in a structured and modular manner. Upon these procedural components, a duality-preserving iterative algorithm,[21,22] which is exact adjoint of the iterative algorithm used in the original flow problem, is developed for the iterative solution of the flow adjoint equations. In order to verify the discrete adjoint solver, a three-step strategy is adopted: first the complex-step method is applied to the entire optimization problem; the tangent model is then verified with the result of the complex-step method; finally the adjoint computation is verified by checking the duality between the tangent and adjoint models.

The proposed methodology for developing discrete adjoint solver offers several advantages:

(1) Reduced development effort. Compared to hand-coded adjoint implementation, the use of AD tools in the proposed method avoids tedious and error-prone manual derivation and programming of detailed computational procedures for the adjoint model, substantially reducing the development difficulty and time.
(2) Consistent and accurate gradient. Since duality is strictly preserved in adjoint implementation and the duality-preserving iterative algorithm is developed for the solution of flow adjoint equations, the gradient of the objective function calculated by the adjoint model is consistent with the one obtained by the tangent model or other exact numerical differentiation methods in the sense of machine precision, not only for the final converged result, but also for intermediate values throughout each iteration.
(3) High efficiency. For AD-assisted discrete adjoint solver on structured grids, the transposed Jacobian arising in the adjoint model can be calculated once and explicitly stored to attain good performance.[16,18] However, this storage is often unaffordable in the case of unstructured grids due to much larger neighbors-to-neighbors stencil, especially for 3D problems. Therefore, using AD tools to generate codes of computing matrix-vector products in the adjoint model without explicit storage is preferred. For the existing AD-assisted adjoint solvers on unstructured meshes, the runtime of the flow adjoint computation is usually 2–5 times that of the flow solution, owing to the computation and storage of immediate variables.[15,17] Recently, Albring et al. used C++ Expression Template technique to efficiently implement the reverse mode of AD for developing a discrete adjoint solver within the open-source multi-physics framework SU2,[23] obtaining a runtime ratio equal to 1.17 for inviscid flow.[24] In the current work, based on the structured