11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20–25, 2014, Barcelona, Spain

EQUATIONAL DIFFERENTIATION OF INCOMPRESSIBLE FLOW SOLVERS

Guillaume Pierrot

ESI Group, 99 rue des Solets, Parc d'Affaires SILIC, 94513 Rungis CEDEX
guillaume.pierrot@esi-group.com

Key words: *Equational Differentiation, Algorithmic Differentiation, incompressible Navier-Stokes, Adjoint System.*

This paper formally introduces the concept of Equational Differentiation as an automatic methodology to derive discrete adjoint equations associated to a numerical solver. The Equational Differentiation (ED), although related to, differs from Algorithmic Differentiation (AD) in the purpose. Indeed, whereas AD intends to derive the code itself, ED focuses on deriving the discrete equations that are ultimately solved by the solver.

This new paradigm has two major implications. First of all, it completely disconnects the primal and the adjoint solvers, letting a total freedom on the way we are to solve the adjoint equations, thus deciding whether or not to re-use the primal solver methodology. Of course one may argue, that provided the primal solver has converged enough, it is somehow implied that the Jacobian matrix of the fixed-point iterations has eigenvalues of magnitude lower than one (see [6] for a discussion on this topic), making the transposed Jacobian a suitable preconditioner for the adjoint system. But on the one hand it does not imply that it is optimal, so if we are able to derive a better preconditioning strategy for the adjoint, why not using it? And on the other hand, real-life simulations (e.g. industrial aerodynamics cases) tend to exhibit limit cycles with approximately converged solutions and eigenvalues magnitude of the Jacobian matrix of the order of one. As a consequence the adjoint solver, when preconditioned by the transposed Jacobian, may very well fail to converge satisfactorily, even from an engineering point of view. Of course the fact that ED allows for a total freedom in the solving of the adjoint equations does not tell what to do with such a freedom and how to solve the system efficiently. Some preconditioning strategies will be discussed here in the context of discrete adjoint equations originating from a commercial incompressible Navier-Stokes Finite Volume solver.

The second implication is that it considerably mitigates some of the drawbacks of classical AD. First of all it is far less intrusive an approach. Indeed, in the ED methodology we do not care how the primal solver actually solves the equations provided it does it satisfactorily, hence allowing for the needed information exchange to occur at a shallow level wrt the numerical solver, typically user subroutines may be used. Second, the huge memory consumption typical of AD reverse mode is drastically reduced due to the fact that ED does not propagate over iterations, only one (pseudo-) iteration being involved in the differentiation process.

Related works include work from Nielsen & al. [1], where ED is applied through complex-variable method to a compressible flow solver, although not formally presented as a separated methodology, as well as work by Christianson [4], where a methodology is proposed to apply AD only to the final step, once steady-state has been reached, thus guaranteeing the accuracy of the computed adjoint state. In this approach however, the methodology used for solving the adjoint equations is necessarily the transposed one of the primal solver. In [5] however the same author proposed a generic methodology to compute the gradient with AD, letting the freedom on how to solve the adjoint equations, but without describing a general methodology for deriving it and still relying on an intrusive AD approach.

REFERENCES

[1] Nielsen E J., Kleb W L., Efficient construction of discrete adjoint operators on unstructured grids by using complex variables. AIAA Paper 20050324, *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005-0324.

[2] Schmidt S., Schulz V., Shape derivative for general objective functions and the incompressible Navier-Stokes equations. *Control and Cybernetic*, Vol. **39**, 2010.

[3] Jones D, Mller J-D., Christakopoulos F., Preparation and assembly of discrete adjoint CFD codes.*Computers and Fluids*, Vol. **46**, 2011, 282-286.

[4] Christianson B., Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, Vol. **3**, 1994, 311-326.

[5] Christianson B., Reverse accumulation and implicit functions. *Optimization Methods and Software*, Vol. **9**, 1998, 307-322.

[6] Gilbert J-C., Automatic Differentiation and iterative processes. *Optimization Methods and Software*, Vol. **1**, 1991, 13-21.