Toward a Discrete Adjoint Model of ACE+

Z. Dastouri[1], J. Lotz[2] and U. Naumann[2]

[1] LuFG Informatik 12: Software and Tools for Computational Engineering (STCE)
RWTH Aachen University, DE 52056 Aachen
dastouri@stce.rwth-aachen.de

[2] STCE – RWTH Aachen University
{lotz, naumann}@stce.rwth-aachen.de

Key Words: *Algorithmic Differentiation, Operator Overloading, Adjoint, CFD-ACE+.*

CFD-ACE+[1] is a general computational fluid dynamics and multiphysics solver for a wide range of physics disciplines that is developed by ESI group[2]. The partial differential equations involved can be solved in multidimensional (zero- to three-dimensional), and in steady or transient form. CFD-ACE+ is offered as a base package that includes flow, heat transfer and turbulence and the source code is written in Fortran.

Sensitivity capability of CFD code is highly desirable for design optimization, error estimation and parameter studies. Adjoint method is a common approach for the gradient computation when the problem posseses a large number of design variables. This method can be broken down into two categories, continuous and discrete. In the continuous version, the adjoint equations are derived analyticaly from the primal equation, and are then discretized. This is in contrast to the discrete approach that generates the adjoints directly from the primal discretization. Tangent as well as adjoint sensitivity of CFD code can be generated by Algorithmic Differentiation (AD) [1]. The method is based on the application of the chain rule of differentiation to each operation in the program flow. The derivatives given by the chain rule can be propagated forward (forward mode) or backward (reverse mode).

There are two main methods for implementing algorithmic differentiation: by source code transformation (s-t) or by using derived data types and operator overloading (o-o). In o-o AD the code segments and arguments of the primal code are stored inside a memory structure called tape during the forward run of the primal. In reverse mode the stored values on the tape are interpreted to get the resulting derivatives. While in s-t approach the code is parsed at a compile time and the actual code is differentiated.

In this paper, we establish a discrete adjoint version of ACE+ using operator overloading dco/fortran [2]. The results of adjoint version of o-o and s-t approaches are previously obtained for a simpler, but very similar structure CFD code called GPDE[3]. GPDE is a pressure based Navier–Stokes equations CFD solver that uses a same discretization type (faced-based incompressible SIMPLE scheme [3,4]) as ACE+, but has a more limited volume (fewer robustness/accuracy enhancing additions, simpler iterative solver). Our focus in this talk is to describe a methodical approach for application AD tool on a legacy large size flow solver such as ACE+, its complexities and challenges in comparison of applying AD tool on a medium size CFD solver. We intend to extend the application of o-o tool on

1 http://www.esi-group.com/products/Fluid-Dynamics/cfd-ace
2 https://www.esi-group.com/
3 http://cfdpack.net/

ACE+ to unsteady flows by using a hybrid (source transformation / overloading) discrete adjoint approach.

ACKNOWLEDGMENT
The presented research is supported by the project "About Flow", funded by the European Commission under FP7-PEOPLE-2012-ITN-317006. See http://aboutflow.sems.qmul.ac.uk

REFERENCES

[1] A.Griewank, A.Walther, Evaluating Derivatives–Principles and Techniques of Algorithmic Differentiation, Second Edition, SIAM, 2008

[2] U. Naumann. The Art of Differentiating Computer Programs. SIAM, 2011.

[3] J.H.Ferziger, M.Peric. Computational Methods for Fluid Dynamics. Springer, 2002.

[4] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal for Heat Mass Transfer,* 15:1787-1806, 1972.