

AD2CompEng - Automatic Differentiation and Adjoint Applied to Computational Engineering Progress Meeting 14th June 2004

Shaun Forth

S.A.Forth@rmcs.cranfield.ac.uk

Engineering Systems
Cranfield University (Shrivenham Campus)
RMCS Shrivenham
SWINDON

Agenda (times are approximate)

- 10:15 Arrival and coffee
- 10:30 Introduction to project - Shaun Forth
- 10:45 Adjoint differentiation of beam3d - Mohamed Tadjouddine
- 11:30 Use of beam3d adjoint in optimisation - Andy Keane
- 12:15 Review of Beam3d (with buffet lunch)
- 13:15 Review of AD with Hydra - Sergio Campobasso/David Radford
- Requirements for Higher Derivatives - Mike Giles
- 14:00 Planning of Cranfield Hydra AD work
- 15:00 Short Report on Merlin/IMPNS Work with Ning Qin - Shaun Forth
- 15:30 AOB, date of next meeting and close

Attendees (and Apologies)

Cranfield University : Shaun Forth, Mohamed Tadjouddine, John Pryce, (Mark Savill).

Southampton University : Andy Keane.

Oxford University : Mike Giles, Sergio Campobasso, Mihai Duta.

Sheffield University : (Ning Qin), Alan LeMoigne.

Cambridge University : David Radford.

University of Hertfordshire : Bruce Christianson.

ROLLS-ROYCE : Leigh Lapworth.

AIRBUS UK : Stefano Tursi.

BAE SYSTEMS : David Standingford.

AD2CompEng Project

- Proposal written by Forth, Keane, Giles and Qin [FGQK01].
- Letters of support from ROLLS-ROYCE and AIRBUS
- Submitted 1st November 2001
- Grant awarded 22nd March 2002 - earliest start date 1st October 2002
- Recruitment difficult, Dia Zeidan appointed March 2003.
- Dia Zeidan resigned November 2003.
- Mohamed Tadjouddine starts work on project December 2003.

Project Objectives [FGQK01]

1. To carefully compare and hence demonstrate the superiority of AD tools over hand-coding for engineering adjoint codes in terms of accuracy, ease of use and efficiency. Amount of required AD expert intervention to be assessed.
2. Using (1) facilitate novel engineering design optimisation strategies using hybrid evolutionary/gradient based methods.
3. Use AD tools to produce adjoint code for a space-marched flow solver. This is highly involved and demanding ... more akin to a time-dependent problem. Careful use of the AD tools and expert intervention will be required to keep memory and run-time requirements to a minimum via a check-pointing strategy.
4. Show superiority (robustness/efficiency) of AD over finite-differencing for Jacobian-vector products within a Newton solver for a turbulent Navier-Stokes flow solver.
5. Education of a new AD application specialist and feedback lessons learned from above into AD tool development.

Phase 1: Eng. Design (Andy Keane)

- Structural optimisation problem: objective function given by BEAM3D code, optimisation currently by time-consuming GA.
- Adjoint (gradient) code required for Meta-Lamarckian (hybrid gradient-descent/GA) optimisation.
- BEAM3D interesting features: complex valued arithmetic; large, dense linear solve (LAPACK); parallel loop over forcing frequencies.
- First adjoint solver developed by MT using ADIFOR 3.0 starting Jan 2004 and delivered May 2004 - in 5 months
- Only 1 month > plan, but 9 months behind schedule.
- ADIFOR work presented at European Conference on Computational Optimisation, Dresden, March 2004.
- Submitted to AD2004 conference (Chicago July).
- TAF generated gradients by mid-June.

Phase 2: Aerospace Design (Mike Giles)

- Generate HYDRA adjoint via AD Tools (TapEnADe, TAF, ADIFOR 3) and compare performance (with hand-coding).
- Performance issues - flux adjoints via recomputation, parallel loops (residual accumulation), iterative-incremental techniques, suppressing differentiation in convergence acceleration techniques.
- Recently - Cusdin/Müller [CM03b, CM03a]: replace `abs` with branches, tape reduction by branch canonicalisation and interface contraction (flux into subroutine), inlining taping routines, local arrays for taping, eliminating common sub-expressions, unrolling small loops into single statements, pure differentiation, Runge-Kutta fixed-point solvers: AD adjoints **can** be as fast as hand-coding.
- Do we need a modified research plan?

Phase 3: Supersonic Adjoint (Ning Qin)

- Original proposal - use check-pointing (storage of intermediates) to reduce taping/recomputation in space-marched flow solver adjoint and apply to supersonic aircraft design.
- Now of less interest to Ning.
- Suggested looking at active control for time-dependent problems - still will need check-pointing?

Phase 4: CFD Newton solvers (Ning Qin)

- Original proposal - use AD for large-scale Newton solvers in IMPNS space-marching solver.
- Now of less interest to Ning.
- Suggested looking at using AD to generate Newton solver from explicit unstructured mesh solver?

References

- [CM03a] P. Cusdin and J.-D. Müller. Deriving linear and adjoint codes for CFD using automatic differentiation. *Submitted to AIAA Journal*, 2003.
- [CM03b] P. Cusdin and J.-D. Müller. Improving the performance of code generated by automatic differentiation. School of Aeronautical Engineering Report QUB-SAE-03-04, Queen's University Belfast, School of Aeronautical Engineering, Faculty of Engineering, Queen's University, Belfast, Sep. 2003. Submitted to *Optimization Methods and Software*.
- [FGQK01] Shaun A. Forth, Michael B. Giles, Ning Qin, and Andrew J. Keane. AD2CompEng - automatic differentiation and adjoints applied to computational engineering. Case for Support in EPSRC Proposal, Oct. 2001.