Research Topic: Automatic Differentiation in Matlab

Dr Shaun A. Forth
Applied Mathematics & Scientific Computing
Engineering Systems & Management Department
Cranfield University
DCMT Shrivenham
Swindon SN6 8LA, UK
email: S.A.Forth@cranfield.ac.uk

October 19, 2009

1 Abstract

Automatic Differentiation (AD) concerns the accurate and efficient differentiation of functions defined by computer programs [Gri00]. AD is used in a wide variety of applications including weather forecasting [GKT+05], climate studies [GTH+02], aerodynamic optimisation [BGHK94], and is being integrated into packages for solving ODEs [SKF05] and optimisation.

Over the past 9 years Dr Shaun Forth has developed an AD tool named MAD (see [For06] and www.matlabAD.com) for the automatic differentiation of computer codes written in Matlab. Cranfield University has a vacancy for a suitably qualified, able and enthusiastic student to undertake undergraduate summer intern work for up to 3 months in the field of automatic differentiation in Matlab or applications thereof with Dr Shaun Forth. Possible areas of research include:

- Sparsity detection - robustly finding which entries of a functions Jacobian or Hessian are non-zero [GK06, Wal08].
- Improving the efficiency of reverse mode AD [Gri00].
- Extended Jacobian-based approaches to AD [FTPR04].
- Sensitivity analysis of differential equation solvers such as Kunge-Kutta methods.
- Applications of AD to scientific and engineering problems.

The area to be studied will depend on the experience, abilities and interests of the successful applicant.

2 Pre-Requisites

Applicants will be expected:

- To be studying for a degree in a mathematical, scientific or engineering discipline.
- To have some experience in writing computer programs for scientific or engineering analysis.
- To have good Matlab programming skills.

- Depending on the exact project to which they are assigned, applicants might also benefit from experience of
  - Object-oriented programming.
  - Mathematics/numerical methods for linear algebra, differential equations, finite-elements, optimization.

3 Questions for Applicants

If you wish to be considered for this positions then please submit response to all of the question in Sect. 3.1 and **ONE** of the questions in Sect. 3.2, together with any Matlab code developed, to Dr Forth compressing any folders using WinZip or gzip if necessary. Dr Forth will be looking for evidence of good Matlab software development skills and mathematical insight in your response - you are not expected to have experience of AD. Please feel free to attach **relevant** additional supporting documents, e.g., reports/papers relating to mathematical or software tasks you have previously performed.

3.1 COMPULSORY QUESTION: A Simple Implementation of Forward Mode AD in Matlab

The class folder `@fwdmode` found with this document contains a simple implementation of the so-called forward mode of automatic differentiation. The m-file `SimpleExample.m` shows how this class may be used to calculate the derivative of the function $y = x * x + 3 * x + 2$ for $x = 3$ where $*$ means multiplication.

1. How are the multiplication and addition operations differentiated?
2. Why is the derivative component of x set to be 1?
3. Can you extend the `fwdmode` class to differentiate,
$$y = 3 * x * x - 2 * x - 5,$$
or
$$y = 3x^2 * sin(x).$$
4. What other operations/operations can you add to the `fwdmode` class so it can differentiate more general functions? Make sure to document your testing of these functions.

3.2 ANSWER ONE of the Following Additional Questions

Submit answers to **ONE** of the following topic areas:

1. The m-file `SecondExample.m` shows how the `fwdmode` class can be used to calculate second derivatives. Explain how this is done and test for a greater range of functions.
2. Extend the `fwdmode` class to cope with `if` statements leading to branching, i.e., to automatically cope with code of the form:

```
y=3*x*x-2*x-5
if y>2
  z=3*y;
else
  z=-y;
end
```

3. Describe how the accurate and efficient use of derivatives via AD has, or might, improve accuracy or robustness of algorithms for your particular area of interest in scientific computing. You might like to look for and review a relevant paper using AD in this area.

4. What is the reverse, or adjoint mode of automatic differentiation? How does this compute the gradient of a function for a cost of only a small multiple of the number of floating point operations in the original code - no matter how many variables the function depends on?

References

[BGHK94] Christian H. Bischof, Larry Green, Ken Haigler, and T. Knauff. Calculation of sensitivity derivatives for aircraft design using automatic differentiation. In *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 94-4261*, pages 73–84. American Institute of Aeronautics and Astronautics, 1994. Also appeared as Argonne National Laboratory, Mathematics and Computer Science Division, Preprint MCS-P419-0294.

[For06] Shaun A. Forth. An efficient overloaded implementation of forward mode automatic differentiation in MATLAB. *ACM Trans. Math. Softw.*, 32(2):195–222, June 2006. DOI: `http://doi.acm.org/10.1145/1141885.1141888`.

[FTPR04] Shaun A. Forth, Mohamed Tadjouddine, John D. Pryce, and John K. Reid. Jacobian code generated by source transformation and vertex elimination can be as efficient as hand-coding. *ACM Trans. Math. Softw.*, 30(3):266 – 299, Sep. 2004. DOI: `http://doi.acm.org/10.1145/1024074.1024076`.

[GK06] Ralf Giering and Thomas Kaminski. Automatic sparsity detection implemented as a source-to-source transformation. In Vassil N. Alexandrov, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Computational Science – ICCS 2006*, volume 3994 of *Lecture Notes in Computer Science*, pages 591–598, Heidelberg, 2006. Springer. `dx.doi.org/10.1007/11758549_81`.

[GKT+05] Ralf Giering, Thomas Kaminski, Ricardo Todling, Ronald Errico, Ronald Gelaro, and Nathan Winslow. Tangent linear and adjoint versions of NASA/GMAO's Fortran 90 global weather forecast model. volume 50 of *Lecture Notes in Computational Science and Engineering*, pages 273–282. Springer, New York, NY, 2005.

[Gri00] Andreas Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Applied Mathematics. SIAM, Philadelphia, PA, 2000.

[GTH+02] E. Galanti, E. Tziperman, M. Harrison, A. Rosati, Ralf Giering, and Z. Sirkes. The equatorial thermocline outcropping — a seasonal control on the tropical pacific ocean-atmosphere instability. *Journal of Climate*, 15(19):2721–2739, 2002.

[SKF05] L. F. Shampine, Robert Ketzscher, and Shaun A. Forth. Using AD to solve BVPs in MATLAB. *ACM Trans. Math. Softw.*, 31(1):79–94, March 2005. DOI: `http://doi.acm.org/10.1145/1055531.1055535`.

[Wal08] Andrea Walther. Computing sparse Hessians with automatic differentiation. *Transaction on Mathematical Software*, 34:Article 3, 2008.