Efficient Derivative code for data assimilation systems provided by automatic differentiation

R. Giering[1], T. Kaminski[1], and M. Voßbeck[1]

[1]*FastOpt* (http//FastOpt.com)

FastOpt

- Founded in February 2000 at Hamburg
- By Ralf Giering and Thomas Kaminski
- One more colleague as of July 2003: Michael Voßbeck
- Two kinds of business:
  -> Develop and provide tools for Automatic Differentiation (AD)/ Adjoint coding
  -> Consulting projects with focus on AD, Inverse Modelling, Data Assimilation
- 16 years of experience in Adjoint coding

TAF

- Compiler tool for Fortran 77-95: successor of TAMC
- Tangent linear and adjoint code from code of underlying model
- Generated code well readable
- Hessian code by applying TAF twice

TAF-Advanced

- Generation of flexible storing/reading scheme for required variables triggered by TAF init and store directives
- Generation of simple checkpointing scheme (Griewank 1992) triggered by combination of TAF init and store directives
- Generation of efficient adjoint (Christianson, 1994, 1998) for converging iterations triggered by TAF loop directive
- TAF flow directives for black-box routines, or to include user provided derivative code (exploit self-adjointness, MPI wrappers, etc...)
- Automatic Sparsity Detection
- Can generate adjoint with restart capability after interruptions (divided adjoint)

Difficulties

- Need TAF directives from user for efficient adjoint and Hessian code
  -> storing vs. recomputation
  -> indicate parallel loops
- Need to avoid complex control flows
  -> simple ones o.k.
  -> need to recode some nested GOTO, CYCLE, ENTRY constructs
- Few restrictions on coding standard
  -> operator overloading
- Need to avoid complex sequences of dynamic memory allocation
- TAF analysis handles array as a whole
  -> incomplete array assignments can cause problems
- Support for parallelisation still basic
  -> Handles the main OpenMP directives
  -> Hand coding of MPI-wrappers required

Performance Examples

Model (ADM reference)	Lines	Lang	TLM	ADM	Ckp	HES
MOM3 (Galanti et al. 02)	50'000	F77	Yes	4,6	2 lev	-
MITGCM (Marotzke et al. 99, Heimbach et al. 02)	100'000	F77	1,8	5,5	3 lev	11.0/1
BETHY (Kaminski et al. 03, Scholze 03)	5'400	F90	1,5	3,6	2 lev	12.5/5
Nav.-Stokes-Solver (Hinze and Slawig 03)	450	F77	-	2,0	steady	-
NSC2KE (Giering et al. 05)	2'500	F77	2,4	3,4	steady	9.8/1
HB_AIRFOIL (Thomas et al. 03)	8'000	F90	-	3,0		-
ARPS (Yang et al. 04) in progress	40'000	F90	2	11,0	2 lev	-

- Lines: total number of Fortran lines without comments
- Numbers for TLM and ADM give CPU time for (model + gradient) rel. to model
- HES format: CPU time for Hessian * n vectors rel. t. forw. model/ n
- 2 (3) level checkpointing costs 1 (2) additional model run(s)

TAC++

- Compiler tool for C(++)
- Same approach as TAF, similar performance:

Model	#lines of code	FUNC [s]	TLM / FUNC	ADM/ FUNC
Roeflux (Cusdin and Muelle	140	6,2E-7	3,3	3,9
2streams (Pinty et al.)	56	4,0E-7	2,5	2,7
TAU-ij (Heinrich et al.)	129	3,0E-3	--	2,6
GasNetOpt (Steinbach)	25	2,4E-7	2,5	2,7

- Work in progress; functionality will be extended "model by model"
- More details in Vossbeck et al. (2004)

TLM/ADM of fvGCM

Joint work with R. Todling, R. Errico, R. Gelaro, and N. Winslow (GMAO)
TLM/ADM generation automated one-click procedure (no modifi. of generated code)

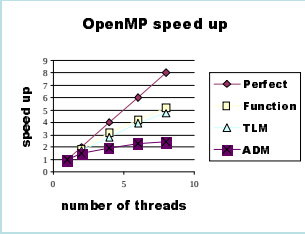
- Storing required variables: 41 TAF init directives and 75 TAF store directives
- 204 TAF flow directives for generation of specific call sequences (e.g. for FFT)
- 11 TAF loop directives to indicate parallel loops
- Complex control flows simplified at two places
- Initialisation split off the main model code
- OpenMP handled by TAF
- ADM MPI wrappers (+2 TLM MPI wrappers) hand-written, interfaces generated by TAF flow directives

More details in Todling et al. (2003) and Giering et al. (2005)

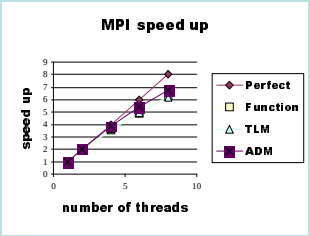
fvGCM Performance

Platform/Setup	TLM	ADM	ADM-noopt
Linux Intel 4	1,5	7.0	-
SGI OpenMP-1 / 8 threads	1,5	10,8	20,6
SGI MPI-1 / 8 processors	1,5	3,9	12,6

- Numbers for TLM and ADM give CPU time for (model + gradient) rel. to model
- Due to bug of SGI-Fortran compiler, flags for full optimisation yields ADM-errors of a few %
- ADM-noopt uses reduced optimisation level for ADM



- Generating code for OpenMP-2 avoid many costly critical sections (verified on Linux; generated code for OpenMP-2 does not run on SGI)



References

- Christianson, B., Reverse Accumulation and Attractive Fixed Points, Optimization Methods and Software 3(4), 311-326, 1994.
- Christianson, B., Reverse Accumulation and Iterative Functions, Optimization Methods and Software 9(4), 307-322, 1998.
- Galanti, E., E. Tziperman, M. Harrison, A. Rosati, R. 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.
- Giering, R., and T. Kaminski, Recipes for Adjoint Code Construction, *ACM Trans. Math. Software, 24(4)*: 437-474, 1998.
- Giering, R., T. Kaminski, and T. Slawig. Applying TAF to a Navier-Stokes solver that simulates an Euler flow around an airfoil. to appear in Future Generation Computer Systems, 2005.
- Giering, R., T. Kaminski, R. Todling, R. Errico, R. Gelaro, and N. Winslow. Generating tangent linear and adjoint versions of NASA/GMAO's Fortran-90 global weather forecast model. In M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, editors, accepted by AD 2004, Lecture Notes in Computational Science and Engineering, Berlin, 2005. Springer.
- Hinze, M. and T. Slawig. Adjoint Gradients Compared To Gradients From Algorithmic Differentiation In Instantaneous Control Of The Navier-Stokes Equations. Optimization Methods & Software, 18(3):299-315, 2003.
- A. Griewank, A. .Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, Optimization Methods and Software 1 (1992), 35 - 54.
- Heimbach, P., C. Hill, and R. Giering. Automatic generation of efficient adjoint code for a parallel Navier-Stokes solver. In P. M. A. Sloot, C. J. K. Tan, J. J. Dongarra, and A. G. Hoekstra, editors, Computational Science - ICCS 2002, Proceedings of the International Conference on Computational Science, Amsterdam, The Netherlands, April 21-24, 2002. Part II, volume 2330 of Lecture Notes in Computer Science, pages 1019-1028, Berlin, 2002. Springer.
- Kaminski, T., R. Giering, M. Scholze, P. Rayner and W. Knorr, An example of an automatic differentiation-based modelling system . *In L. Gavrilova,V. Kumar, P. L'Ecuyer, and C. J. K. Tan, editors,* Computational Science - ICCSA 2003, International Conference Montreal, Canada, May 2003, Proceedings, Part II, volume 2668 of Lecture Notes in Computer Science, pages 95-104, Springer, Berlin, 2003.
- Marotzke, J., R. Giering, Q. K. Zhang, D. Stammer, C. N. Hill, and T. Lee. Construction of the adjoint MIT ocean general circulation model and application to atlantic heat transport sensitivity. J. Geophys. Res., 104:29,529 - 29,548, 1999.
- Scholze, M., Model studies on the response of the terrestrial carbon cycle to climate change and variability *PhD thesis, Max-Planck-Institut für Meteorologie., Hamburg, Germany*, 2003.
- Thomas, J.P., K. C. Hall, and E. H. Dowell. A discrete adjoint approach for modeling unsteady aerodynamic design sensitivities. 41st AIAA Aerospace Sciences Meeting, Reno, Nevada, 2003. Voßbeck, M., R. Giering, and T. Kaminski. Towards a tool for forward and reverse mode source to source transformation in C++. Extended Abstract for AD2004: The 4th International Conference on Automatic Differentiation, Chicago, Il, USA, July 19-23, 2004.
- Todling, R., R. Giering, Kaminski, T., Y. Zhu, and J. Guo, Retrospective Data Assimilation For GEOS-4, *Geophysical Research Abstracts, Vol. 5, 11354, 2003 European Geophysical Society*, 2003.
- Xiao, Y., M. Xue, and J. Gao. Development of Adjoint for a Complex Atmospheric Model, the ARPS, using TAF. In M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, editors, accepted by AD 2004, Lecture Notes in Computational Science and Engineering, Berlin, 2005. Springer.