The automatic differentiation tool TAF

Ralf Giering and Thomas Kaminski

***Fast*Opt**

Web: http://www.FastOpt.com

EUROMECH, Erlangen-Nuremberg, February 2003

A few facts about *Fast*Opt

- Founded in February 2000 at Hamburg
- By Ralf Giering and Thomas Kaminski
- Two kinds of business:
  - Develop and provide tools (TAF) for Automatic Differentiation (AD)
  - Carry out Consulting Projects with focus on AD, Inverse Modelling, Optimisation...
- 14 years of Experience in AD

Overview

- Intro FastOpt
- Automatic Differentiation
- TAF description
- Peformance of TAF generated code
- Selected applications:
  - Aerodynamics
  - Oceanography
  - Meteorology
- Summary

Automatic Differentiation

- Function defined by a numerical program
- Differentiation of function according to chain rule
- Local Jacobian matrices of individual statements are easy to derive
- Derivative of function is product of local Jacobians
- Can be evaluated in forward or reverse mode
- AD yields exact derivative
- Task can be done automatically
- Source to source transformation or operator overloading

Example forward mode

$$Y = 4*x + \sin(u)$$

$$\begin{pmatrix} u' \\ x' \\ y' \end{pmatrix}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \cos(u) & 4 & 0 \end{pmatrix} * \begin{pmatrix} u' \\ x' \\ y' \end{pmatrix}_1$$

$$g\_y = 4*g\_x + \cos(u)*g\_u$$

Required variables

- local Jacobian matrix

  y = v * w

- control flow

  if (a < 0) then

- index variables

  a(i+4) = b(j)

Example reverse mode

Y = 4*x + sin(u)

$$\begin{pmatrix} u' \\ x' \\ y' \end{pmatrix}_1 = \begin{pmatrix} 1 & 0 & \cos(u) \\ 0 & 1 & 4 \\ 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} u' \\ x' \\ y' \end{pmatrix}_2$$

adu = adu + ady*cos(u)
adx = adx + ady*4
ady = 0

Reverse mode

- Reverse control flow (goto, etc.)
- Provide required variables in opposite order of original computation
- Taping or recomputation of required variables
- Efficient adjoint code requires sophisticated analysis and detailed transformations

TAF: Transformation of Algorithms in Fortran

- Source-to-source translator for Fortran-77 to 95
- Commercial successor of TAMC
- Forward and reverse mode (1st derivatives):
  Tangent linear and adjoint models
- Scalar and vector mode
- Efficient Hessian (2nd derivative) code
  by applying TAF twice (e.g. forward over reverse)
- Command line program with many options
- TAF-Directives are Fortran comments
- Extensive and complex code analyses
  (similar to optimising compilers)
- Generated code is structured and well readable

TAF
More features

- Generation of flexible storing/reading scheme for required variables triggered by TAF init and store directives
- Generation of checkpointing scheme triggered by combination of TAF init and store directives
- Generation of efficient storing/reading scheme (Christianson) for adjoints of 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
- Basic support for MPI and OpenMP

TAF
Ongoing Development

- TAF is constantly being adapted to new Fortran standards
  - Fortran 2000
  - OpenMP 2
- TAF code analyses are constantly being extended
- TAF algorithms are constantly being improved and adapted to the needs of the users
- FastOpt is giving support for TAF users
- FastOpt is offering consulting for AD and further projects

some larger TAF Derivatives

Model (Who)	Lines	Lang	TLM	ADM	Ckp	HES
NASA/NCAR (w. Todling & Lin)	87'000	F90	2,7	6,8	2 lev	-
MOM3 (Galanti & Tziperman)	50'000	F77	Yes	4,6	2 lev	-
MITGCM (ECCO Consortium)	100'000	F77	1,8	5,5	3 lev	11.0/1
BETHY (w. Knorr, Rayner, Scholze)	5'400	F90	1,5	3,6	2 lev	12.5/5
Nav.-Stokes-Solver (Hinze, Slawig)	450	F77	-	2,0	steady	-
NSC2KE	2'500	F77	2,4	3,4	steady	9.8/1
HB_AIRFOIL (Thomas & Hall)	8'000	F90	-	3,0		-

- **Lines: total number of Fortran lines without comments**
- **Numbers for TLM and ADM give CPU time for (function + gradient) relative to forward 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)**

Performance compared to hand coded adjoints

Code	Hand	TAF/TAMC	relativ
EPT (MINPACK-2)	1,5	1,9	26%
GL1 (MINPACK-2)	1,5	1,7	13%
GL2 (MINPACK-2)	2,1	1,3	-40%
MSA (MINPACK-2)	1,75	1,6	-9%
PJB (MINPACK-2)	1,75	2,2	+25%
SSC (MINPACK-2)	1,18	1,13	-5%
Nav.-Stokes (H & S)	1,9	1,3	-30%

=> Performance of TAF generated adjoints is comparable to that of hand written adjoints

TAMC Adjoints of Large Models

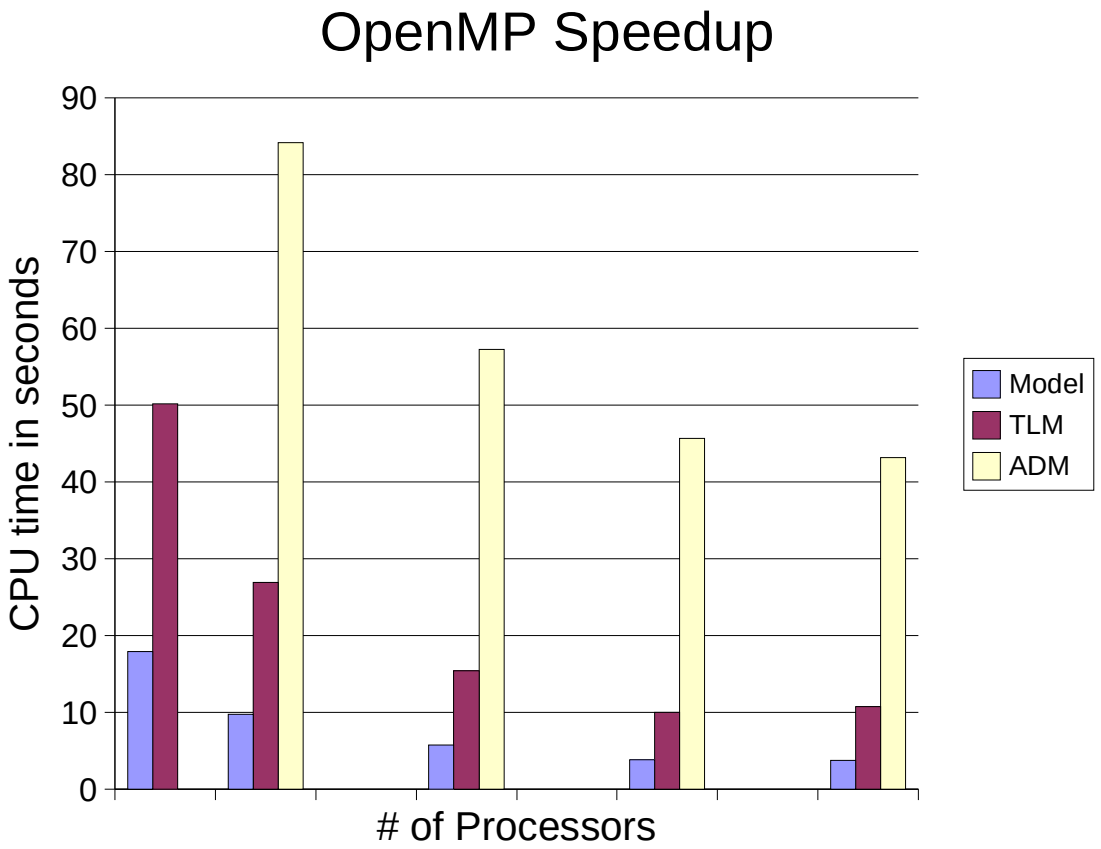
Model (Who)	TLM	ADM	Checkpointing
PUMA (Blessing)	2,0	2,8	
MM5 (Nehrkorn et al.)	Yes	Yes	?
TM2 (Kaminski et al.)		3,4	2 level
TM3 (Roedenbeck et al.)		Yes	?
HCM (Eckert)	Yes	3,7	2 level
MOM3 (Galanti & Tziperman)		4,0	2 level
MITGCM (ECCO Consortium)	1,8	4,5	2 level
HOPE (Oldenborg et al.)		5-6	3 level
WAM (Hersbach)		3,7	2 level

- Numbers for TLM and ADM give CPU time for (function + gradient) relative to forward model
- 2 (3) level checkpointing costs 1 (2) additional model runs

Atmospheric Data Assimilation
at Data Assimilation Office (Todling, Lin...)

- DAO finite Volume GCM: Model of General Atmospheric Circulation
- Hydrodynamics kernel by Lin, Lin/Rood
- ~ 87'000 lines of Fortran 90 (without comments)
- Uses message passing interface (MPI-1,MPI-2) as well as OpenMP for Parallelisation
- Tangent Linear and Adjoint generated by TAF
  Only hand written code for adjoint MPI wrappers
  OpenMP handled by TAF
- Adjoint uses 2 level checkpointing
- To be used by DAO for Data Assimilation (Retrospective Analysis System), Sensitivity Studies, Singular Vector Detection ...

Atmospheric Data Assimilation
OpenMP Speedup for NASA-DAO GCM

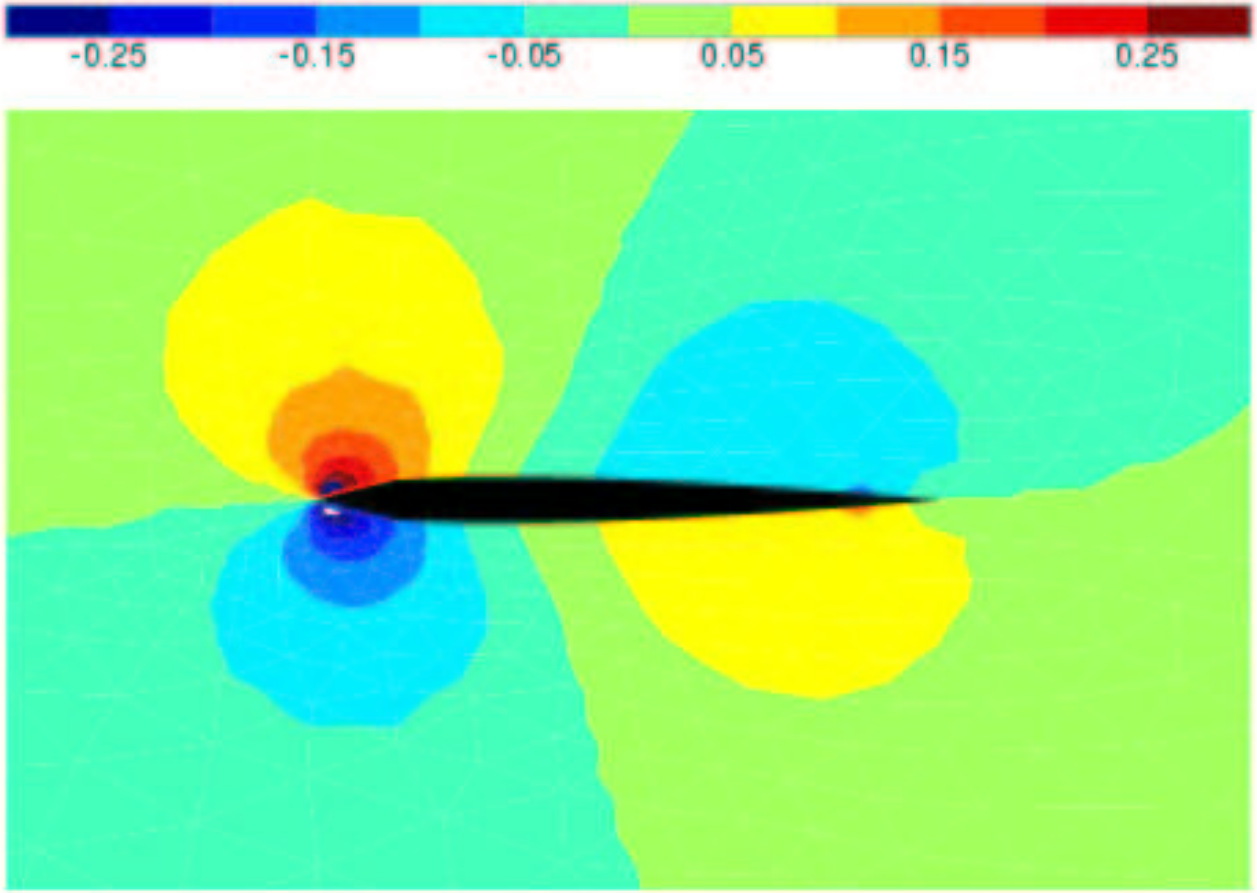


Aerodynamics
with T. Slawig (TU Berlin)

- Model: NSC2KE by Bijan Mohamadi (1994)
- Mixed finite volume-finite element Galerkin CFD model
- 2 dimensional on unstructured grid
- Euler Part with Roe-, Osher-, or Kinematic solver
- K-epsilon turbulence model
- 4^{th} order Runge Kutta
- 2500 lines of Fortran 77 code without comments

Aerodynamics

Sensitivity of lift to vertical velocity



Ocean Data Assimilation
ECCO Consortium

- MIT-GCM: Primitive Equation Model of General Oceanic Circulation
- Various Configurations ~ 100'000 lines of Fortran 77 (without comments)
- Uses MPI and OpenMP for parallelisation
- Tangent Linear, Adjoint, Hessian code generated by TAF
  Only hand written code for communication wrappers
- Adjoint uses 2 or 3 level checkpointing
- Is used for Variational Data Assimilation, Uncertainty Analysis, Kalman Filter and Sensitivity Studies

Summary

- TAF has been applied successfully to various kinds of Fortran codes
- FastOpt has worked a number of projects on
  - Sensitivity Anaysis
  - Optimisation
- For more info check http://www.FastOpt.com