



David M. Gay: Background and Initial AD Work

David M. Gay
Optimization and Uncertainty Estimation Dept.

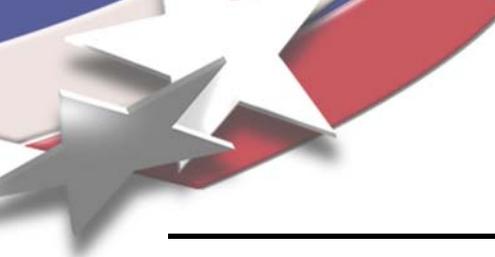
<http://endo.sandia.gov/DAKOTA>
<http://www.cs.sandia.gov/departments/9211/index.htm>





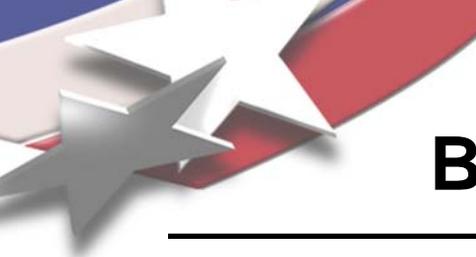
Outline

- Background
 - College (Mich., Freiburg), grad school (Cornell)
 - Boston area (NBER, MIT/CCREMS)
 - NL2SOL, intervals
 - Bell Labs
 - PORT
 - f2c
 - AMPL and AD
- Now at Sandia
 - More AD
 - Global opt. (more intervals)



Background

- Undergrad at U. of Michigan
 - Math major (some physics, computer science)
 - Summer programmer – first LP exposure
 - Senior year in Germany (Freiburg im Breisgau)
- Grad studies at Cornell
 - Numerical analysis in Computer Science Dept.
 - Student of John Dennis
 - “Disproved” Scolnik’s polynomial-time LP alg
 - Thesis: *Brown’s method and some generalizations, with applications to minimization problems.*



Background – Austin, Boston

- UT Austin (1 year)
- NBER, MIT/CCREMS
 - Cambridge MA (6 years)
 - NL2SOL
 - intervals
 - IEEE floating-point standard
 - focus on portability (from Virginia Klema, EISPACK)
 - married Tanner



Background – Bell Labs

- Bell Labs (20 or 22 years, starting Nov. 1981)
 - PORT library (helped with optimization chapter)
 - Chemical equilibrium with Ken McAfee et al.
 - Karmarkar hoopla
 - Fortran-to-C converter f2c
 - Binary \leftrightarrow Decimal conversions
 - AMPL
 - work with Bob Fourer, Brian Kernighan
 - Early C++ users
 - AD, initially first derivatives
 - then Hessians (using partially sep. structure)



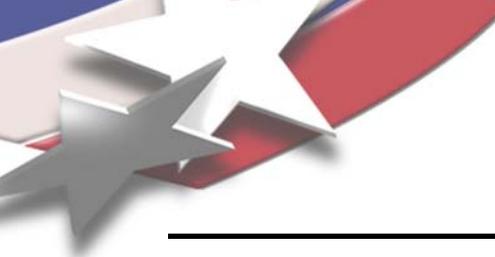
AMPL Lang. for Math. Programming

State finite-dimensional optimization problems...

$$\begin{array}{l} \text{Minimize } f(x) \\ \text{s.t. } L \preceq c(x) \preceq u \end{array}$$

in conventional algebraic notation. Easily

- manipulate,
- solve (separately),
- examine results,
- solve related problems (new data),
- iterate.



AMPL problem classes

- Conventional math. programming problems
 - Linear, nonlinear
 - Continuous or integer variables
- Complementarity constraints
$$f(x) \geq 0 @ g(x) \geq 0$$
- Logic-programming constructs (new)
- Stochastic programming (forthcoming).



Impact of AMPL

- Used in various courses worldwide
- Facilitates research
 - Methods using explicit Hessians (LOQO, Knitro,...)
 - Grad students
- Heavily used in NEOS
- Numerous commercial users
- Used at Sandia
 - Water contamination modeling
 - Bob Carr, Vitus Leung, Jean-Paul Watson
 - Stefan Chakerian (internal NEOS)



AD – Automatic Differentiation

Use *chain rule* to *numerically* combine partials of each operation in expression graph.

Two variants, *forward* and *backward*.

Forward AD recurs partials at each operation.

- Great for one variable
 - Jacobian-vector products in Premo & Sierra
 - (Roscoe Bartlett, Bart van Bloemen Waanders)
 - High-order ODEs
- Slow for n variables: $O(n)$ for $f(x) \rightarrow O(n^2)$ for $s f(x)$



Backward AD for Optimization

For smooth optimization,

- * always need first derivatives
- * sometimes need second.

Backward AD gives function and gradient (first derivatives) in time proportional to function — at cost of memory.



Backward AD details (1)

Two ingredients:

1. Linearity...

$$f(x) = f_1(x) + f_2(x) + \dots + f_k(x)$$

$$\Rightarrow f'(x) = f'_1(x) + f'_2(x) + \dots + f'_k(x)$$



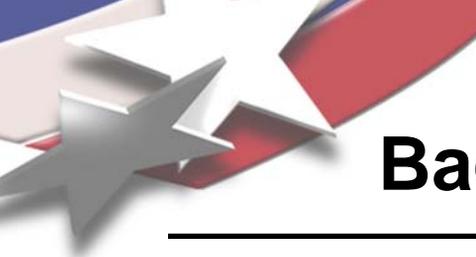
Backward AD details (2)

2. Chain Rule

$$x = o(y, z), \quad f = f(x)$$

$$\Rightarrow \frac{\partial f}{\partial y} = f'(x) \frac{\partial o}{\partial y}$$

$$\frac{\partial f}{\partial z} = f'(x) \frac{\partial o}{\partial z}$$

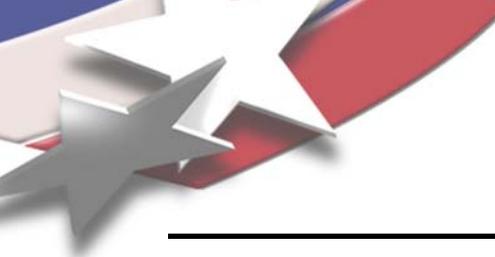


Backward AD: sums of products

Result: if x_i contributes directly to x_j for $j \in S$, then x_j contributes

$\frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$ to $\frac{\partial f}{\partial x_i}$, and

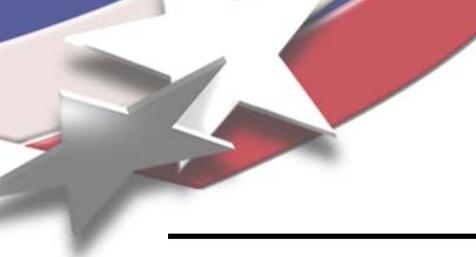
$$\frac{\partial f}{\partial x_i} = \sum_{j \in S} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}.$$



AD: back propagation

To accumulate all $\frac{\partial f}{\partial x_i}$, visiting operations in reverse order suffices. Example:

```
void ADcontext::Gradcomp()
{
    Derp *d = Derp::LastDerp;
    d->b->aval = 1;
    for(; d; d = d->next)
        d->c->aval += *d->a * d->b->aval;
}
```



Backward AD performance

Example: grid improvement objective (Pat Knupp)

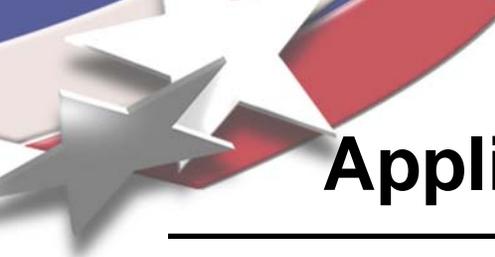
$$f = \frac{3\det(AW^{-1})^{2/3}}{\|AW^{-1}\|_F^2}$$

on a fast Linux PC:

Custom backward AD via C++ operator overloading give $f(x)$ and $s f(x)$ in about 10 × time for $f(x)$ alone.

Optimizing the AD → 1.3 × time for $f(x)$ alone.

For f^3 , it's 40 × and 2.3 ×.

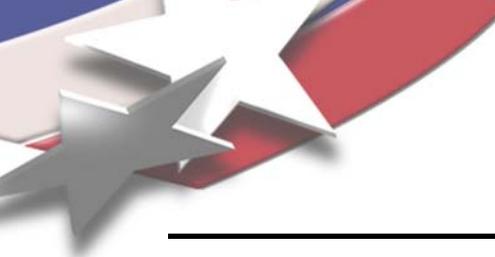


Application to large-scale problems

Accurate, efficient first derivatives widely useful, but getting them is *intrusive*.

Evidence from Sierra/Premo: could

- intrude gradient at each node and
- manually accumulate overall gradient.



Global Optimization

Many heuristics

- Simulated annealing
- Genetic algorithms
- Tabu search
- Stochastic multistart

but only ***branch and bound*** is rigorous.

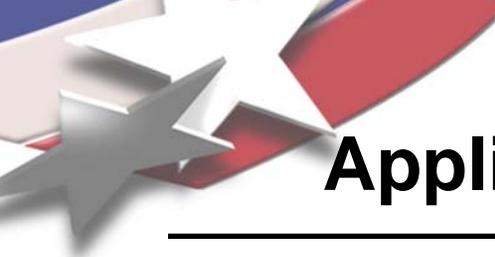
PICO does linear branch & bound; with Bill Hart (etc.), hope to add nonlinear bounding – a rich subject.



Bounding for global optimization

Techniques include

- Convex underestimating functions
- Interval bounds
 - to exclude regions
- Interval sufficiency conditions
 - guarantee a unique solution in a region



Applications of Global Optimization

Nonconvex problems abound

- Nuclear safety (polynomial, linear constraints)
- Protein folding
- Data fitting
- Sensor placement
- Inverse problems



Outside professional activities

- Treasurer of the *Mathematical Programming Society*®
- INFORMS Computing Society prize committee
- Program committee for 2004 AD Conference