The Optimal Computation of Hessians in Optimization

Assefaw H. Gebremedhin, Alex Pothen, Arijit Tarafdar and Andrea Walther

Old Dominion University, Norfolk VA USA
and Technical University of Dresden, Germany

June 16th 2006

Overview of Talk

1. Computing derivatives using FD and AD
2. Exploiting sparsity
3. Direct and Substitution recovery methods and Star and Acyclic coloring formulations
4. New algorithms and results on Colors and Runtimes
5. Results for Hessian computation in ADOL-C
6. Conclusions

One approach for computing derivatives: Approximation via Finite Differencing

$F : \mathbb{R}^n \to \mathbb{R}^m$, $A = F'(x)$

$$Ae_j = \frac{\partial}{\partial x_j} F(x) \approx \frac{1}{h}[F(x + he_j) - F(x)], \quad 1 \leq j \leq n \ *$$

(e_j is the jth coordinate vector)

$f : \mathbb{R}^n \to \mathbb{R}$, $A = \nabla^2 f$
can be estimated by applying * on ∇f (if available).

Note that computing the entire Jacobian/Hessian in this fashion requires $n + 1$ function evaluations.

A better approach for computing derivatives: Automatic Differentiation

- AD is a technology for transforming
  source code for computing a function into
  source code for computing its derivative
- Relies on systematic application of the *chain rule*
  - a function is decomposed into a sequence of arithmetic operations and intrinsic functions (variables classified as *independent*, *intermediate* and *dependent*)
- Unlike Finite Differencing, incurs *no truncation error*

Automatic Differentiation (cont'd)

- Two basic modes
  - Forward
    - idea: choose an input variable, calculate sensitivity of every intermediate wrt that input.
    - computes the Jacobian-vector product $F'(x) \cdot s$ at a cost of at most $5 \cdot OPS(F(x))$.
  - Reverse
    - idea: choose an output variable, calculate sensitivity of that output wrt each intermediate.
    - computes the vector-Jacobian product $s \cdot F'(x)$ at a cost of at most $5 \cdot OPS(F(x))$, independent of number of inputs.
- Using the *second-order adjoint* mode, which is a combination of the forward and the reverse modes, the Hessian-vector product $\nabla^2 f(x) \cdot s$ can be computed at a cost of at most $10 \cdot OPS(f(x))$.

Sparse derivative matrix computation

Given: a vector function F (or a scalar function f)
Goal: compute the derivative matrix $A = F'$ (or $\nabla^2 f$) *efficiently*

Approach: employ the following four-step strategy

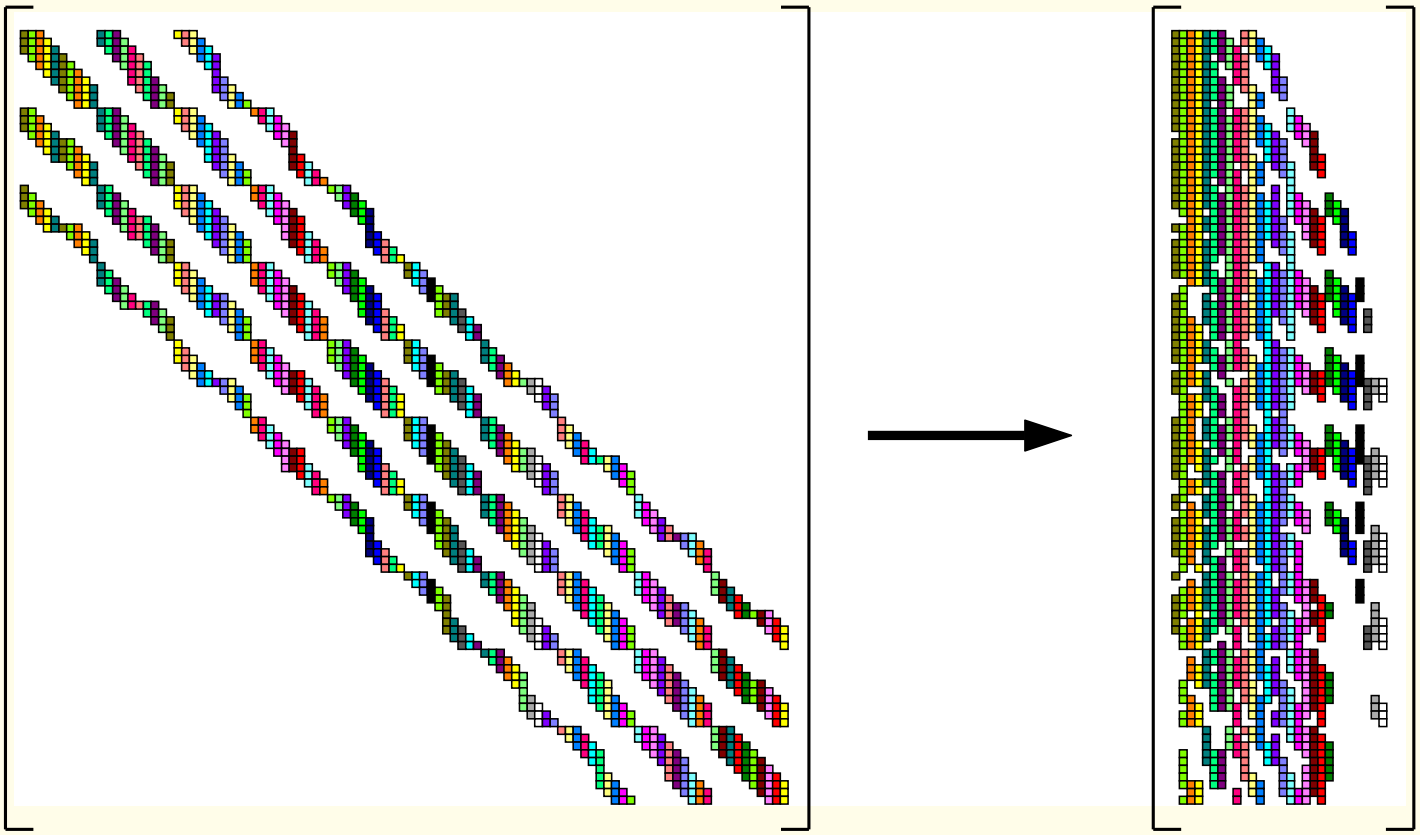
1. Compute the sparsity pattern of $A \in \mathbb{R}^{m \times n}$
2. Compute a seed matrix $S \in \{0,1\}^{n \times p}$ with the smallest p
3. Compute the compressed matrix $B = AS$
4. Recover the nonzeros of A from B

The matrix S *partitions* the columns of A:

$$s_{jk} = \begin{cases} 1 & \text{if column } a_j \text{ belongs to group } k, \\ 0 & \text{otherwise.} \end{cases}$$

$$\sum_{k=1}^{p} s_{jk} = 1 \qquad \forall j : 1 \le j \le n.$$

A matrix and its compressed representation



Computation of seed matrix: problem variations

Matrix types

- nonsymmetric
- symmetric

Partition types

- unidirectional
- bidirectional

Recovery schemes

- direct
- substitution

Required matrix entries

- all
- some

Overview of coloring formulations

	1d partition	2d partition	
Jacobian	distance-2 coloring	star bicoloring	Direct
Hessian	star coloring	—	Direct
Jacobian	—	acyclic bicoloring	Substitution
Hessian	acyclic coloring	—	Substitution

Nonsym A	$G_b(A) = (V_1, V_2, E)$
Sym A	$G(A) = (V, E)$

Compressing the Hessian for direct recovery

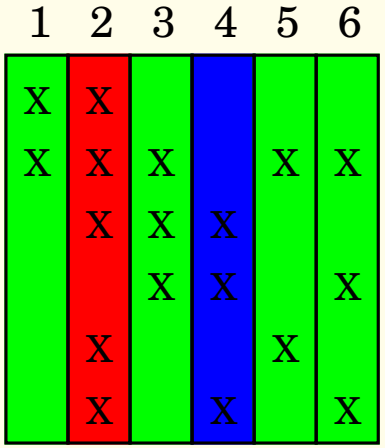
Symmetrically orthogonal partition

whenever $a_{ij} \neq 0$

- a_j only col in a group with nonzero at row i, or
- a_i only col in a group with nonzero at row j.

Formulated by

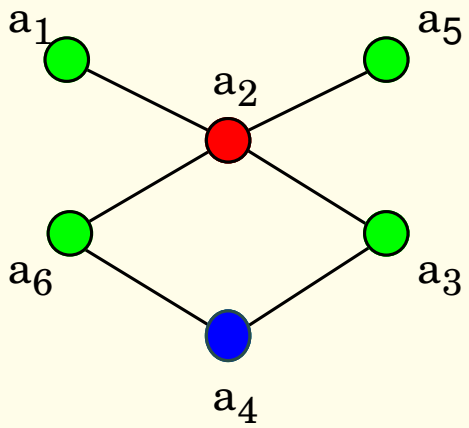
Powell and Toint.



Star coloring

A vertex coloring ϕ of $G(A)$ s.t.

- ϕ is a distance-1 coloring, and
- every P_4 uses ≥ 3 colors.



Equivalence established by

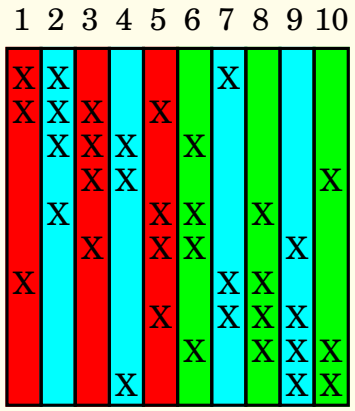
Coleman and Moré (84).

Compressing the Hessian for indirect recovery

Substitutable partition

whenever $a_{ij} \neq 0$

- a_j in a group where all nonzeros in row i are ordered before a_{ij}, or
- a_i in a group where all nonzeros in row j are ordered before a_{ij}.



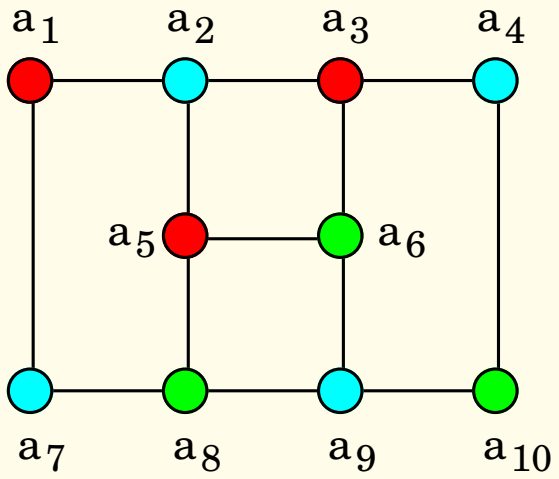
Formulated by

Powell and Toint.

Acyclic coloring

A vertex coloring ϕ of $G(A)$ s.t.

- ϕ is a distance-1 coloring, and
- every cycle uses ≥ 3 colors.



Equivalence established by

Coleman and Cai (86).

Experimental results: Overview

- Algorithms compared
  - greedy algorithms for distance-k coloring (D1 and D2)
  - two earlier algorithms for star coloring (NS, GMP; RS, Powell and Toint)
  - an earlier algorithm for triangular coloring (T-sl, Coleman and More)
  - the new star (S) and acyclic (A) coloring algorithms

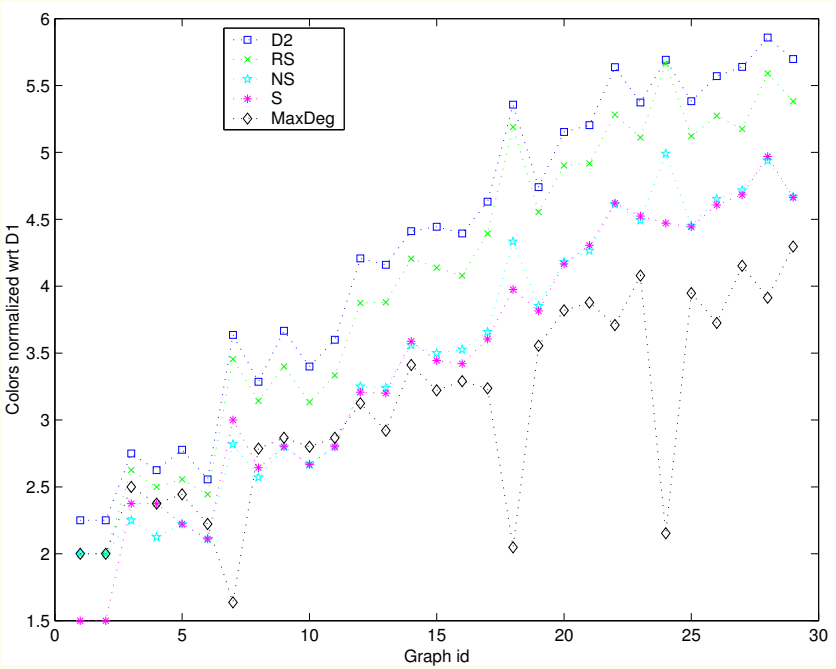
	$\|V\|$ in 1000	$\|E\|$ in 1000	MaxDeg	MinDeg	AvgDeg
range	10 – 150	50 – 17,000	8 – 860	0 – 230	3 – 600
sum	1,500	88,000	6,400	800	4,200

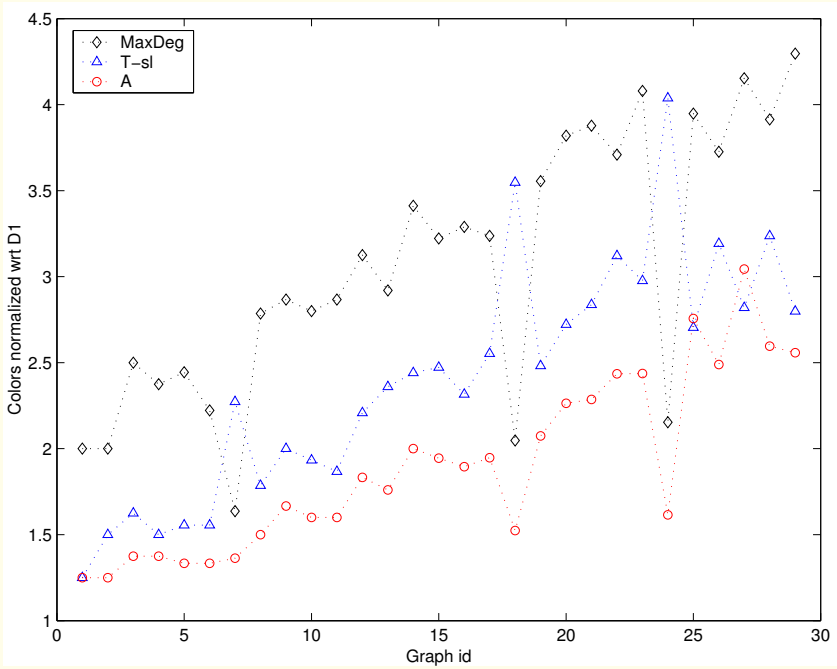
Table: Summary of size and density of test graphs (total: 29).

	D2	RS	NS	S	T-sl	A	D1
colors	9,240	8,749	7,636	7,558	5,065	4,110	1,757
time (min)	28.2	34.4	930	162	12.4	32.5	0.04

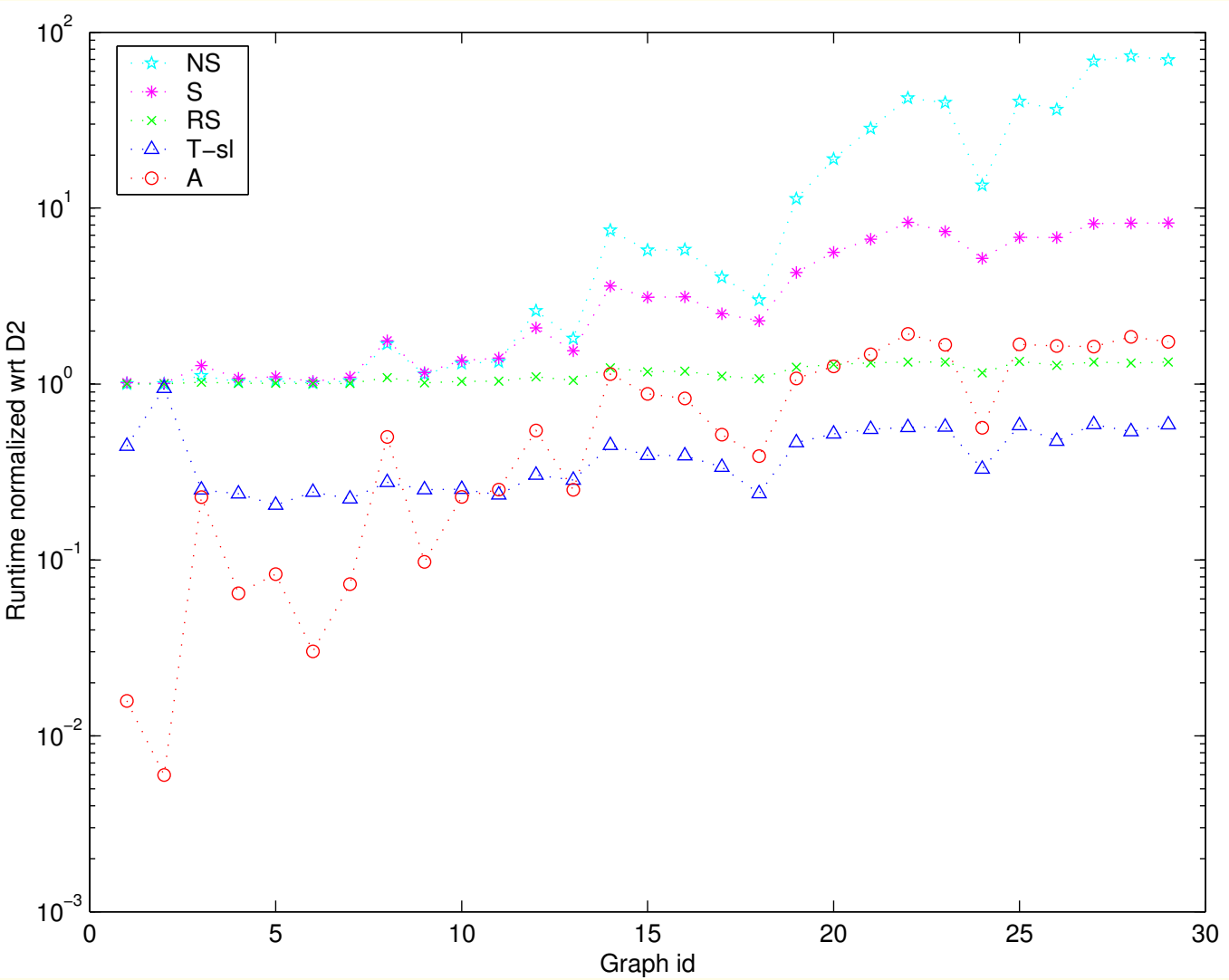
Table: Total number of colors and runtime, summed over all test cases.

Experimental results in more detail: Colors





Experimental results in more detail: Runtimes



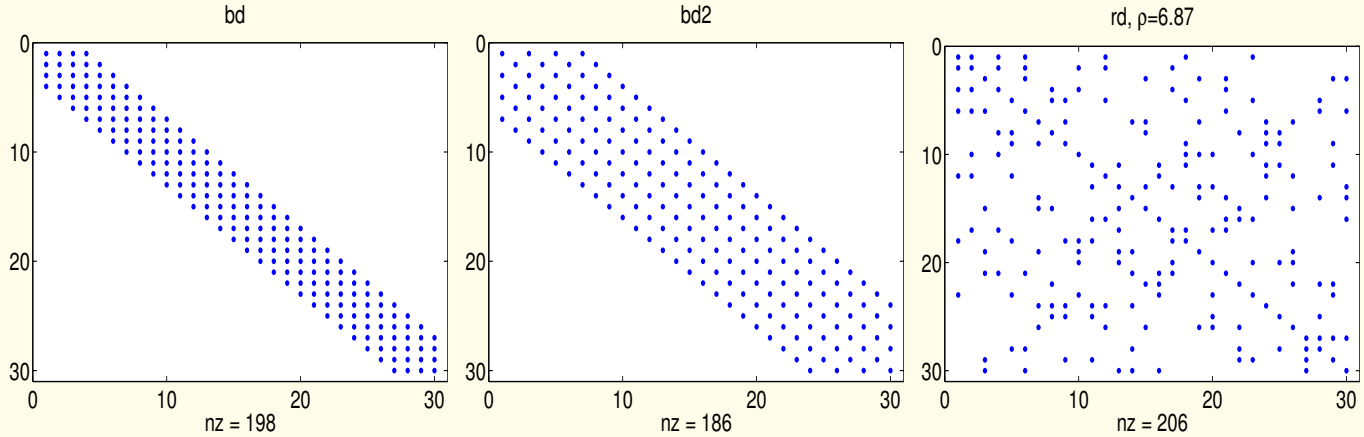
Experiments using ADOL-C

Test function:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{with} \quad f(x) = x^t C x + a^t x,$$

$$C \in \mathbb{R}^{n \times n}, a = (10, \ldots, 10)^t \in \mathbb{R}^n$$

Structure and size of the Hessian C:

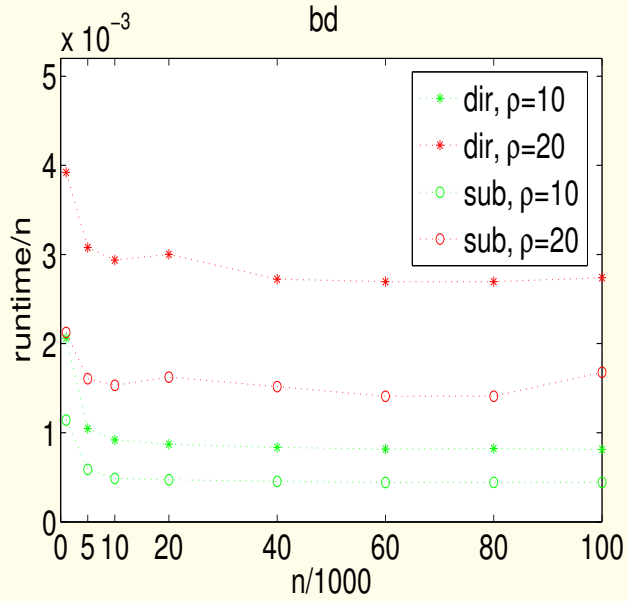


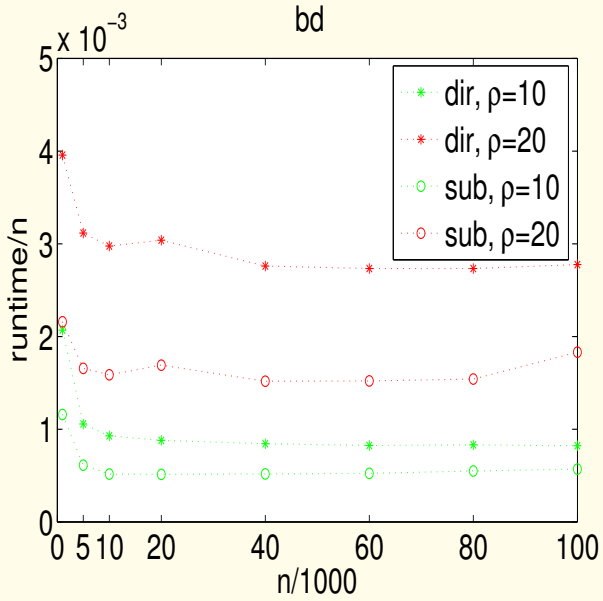
bd1	banded matrix of bandwidth $\rho + 1$ (nnz/row $= \rho + 1$)
bd2	banded matrix of bandwidth $2\rho + 1$ (nnz/row $= \rho + 1$)
rd	random matrix with average nnz/row $\overline{\rho} \approx \rho + 1$

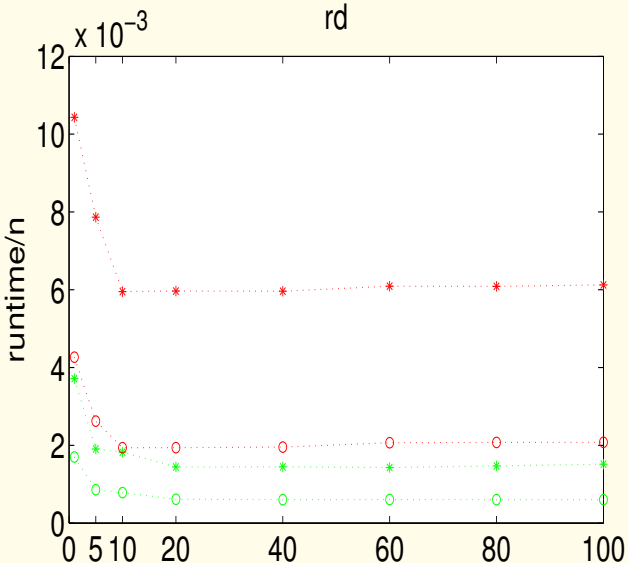
$\rho \in \{10, 20\}$, $\overline{\rho} \in \{10.98, 20.99\}$

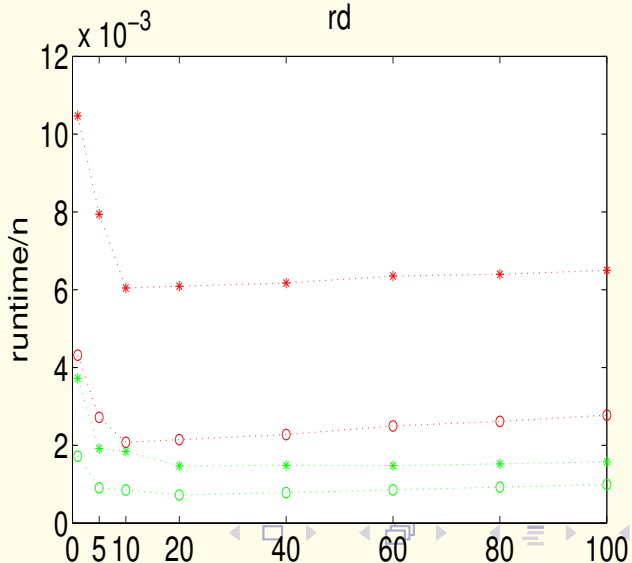
$n/1000 \in \{1, 5, 10, 20, 40, 60, 80, 100\}$

ADOL-C Times for 100K problem: Compr Hess and Tot









ADOL-C: Absolute Runtimes

Problem	Method	Color	Compr. Hess.	Recovery	Total
Banded	Direct	3	274	0.4	278
Banded	Subs.	12	168	3	183
Random	Direct	37	612	0.5	650
Random	Subs.	24	207	45	277

Table: Times for various steps in Hessian computation.

Conclusions

- First practical algorithm and implementation for acyclic coloring (Hessian computation with a substitution method)
- New efficient algorithm for star coloring (Hessian computation with a direct method)
- Codes for computing seed matrix (coloring), compressed Hessian, and recovery of Hessian elements incorporated into ADOL-C.
- A substitution method enables faster computation of Hessians than with a direct method.

References

Gebremedhin, Manne and Pothen.

Graph Coloring for Computing Derivatives.

SIAM Review, Vol 47, No 4, pp. 629–705, 2005.

Gebremedhin, Tarafdar, Manne and Pothen.

New Acyclic and Star Coloring Algorithms.

Submitted to *SIAM Journal on Scientific Computing*.

Gebremedhin, Pothen, Tarafdar and Walther.

Efficient Computation of Sparse Hessians using ADOL-C.

In Prep.