AD and Combinatorics: An Accidental Partnership?

Alex Pothen

Department of Computer Sciences and Computing Research Institute
Purdue University
CSCAPES Institute
www.cs.purdue.edu/people/faculty/apothen

Joint work with
Assefaw Gebremedhin (Old Dominion) and Andrea Walther (Dresden)
Fredrik Manne (Bergen); Arijit Tarafdar (Microsoft)
Doruk Bozdag and Umit Catalyurek (Ohio State); Erik Boman (Sandia)

AD 2008, Bonn

The AD and CSC Communities

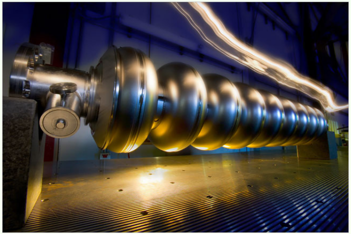
- The computational graph (Bauer, 1974) opens the door for many combinatorial problems in AD, and yet researchers outside the AD community are not familar with them.
- The AD research community has organized since 1991, and the combinatorial scientific computing (CSC) community has organized since 2004.
- Hovland, Naumann, Walther, Utke, Lyons, and others from AD have been interacting with the CSC community since its inception.
- This talk tells a part of this story, and looks forward to even more fruitful interactions.

Overview

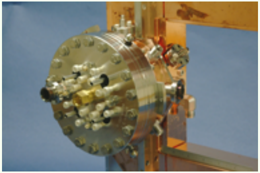
- An Example of the Partnership:
  The CSCAPES Institute
- Describing the Partnership:
  Combinatorics (Coloring) in AD
- Strengthening the Partnership:
  A Way Forward

Outline

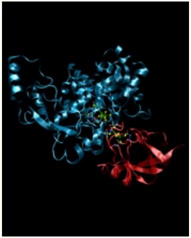
1. The CSCAPES Institute
2. Combinatorics in AD
    - Coloring Models
    - Sequential algorithms
    - Case studies
    - Parallel algorithms
    - Summary
3. The Way Forward



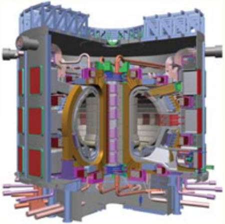
Accelerator Design



RF Gun, SLAC



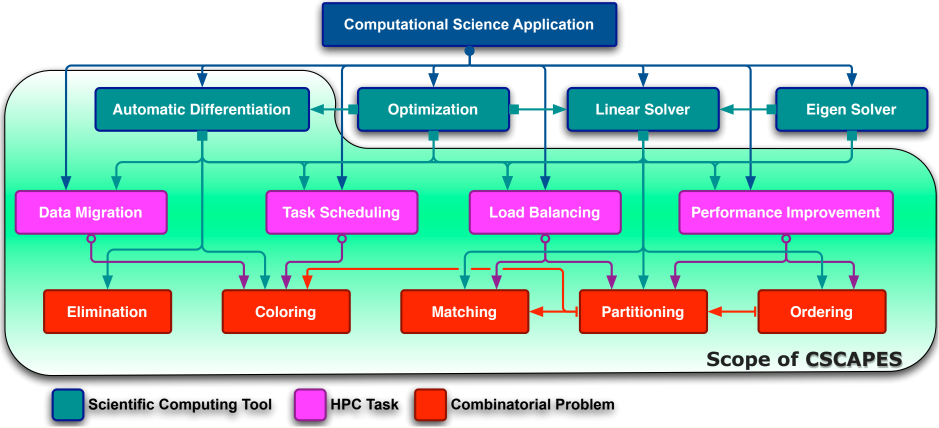
Green Energy



ITER

Scientific Discovery thru' Advanced Computing II

- Office of Science, U.S. Dept. of Energy, 2006-2011
- Sciences
  - Fusion, Accelerator Design, Cosmology, Quantum Chromodynamics, Climate, Groundwater, Materials, Life Sciences
- Enabling Technologies
  - Applied Mathematics, Computer Science, Visualization, Data Management
- ET teams collaborating with Science Application teams to harness petascale computing in simulations
- $ 90 Million per year, 30+ projects, 70+ institutions
- Involve computation at the beginning of a scientific research project, rather than as an afterthought.



Research; Dissemination; Education; Outreach

CSCAPES Institute: Personnel

- Purdue (Old Dominion): Alex Pothen, Assefaw Gebremedhin, Florin Dobrian, Mahantesh Halappanavar, Brandon Hill, Min Huang, Duc Nguyen
- Sandia: Erik Boman, Karen Devine, Bruce Hendrickson, Cederic Chevalier, Michael Wolfe
- Argonne: Paul Hovland, Boyana Norris, Jean Utke, Ilya Safro, Andrew Lyons
- Ohio State: Umit Catalyurek, Doruk Bozdag
- Colorado State: Michelle Mills Strout
- Other Collaborators: Andrea Walther, Uwe Naumann, Fredrik Manne, Yoshi Kawajiri, Larry Biegler

Outline

Combinatorics in AD

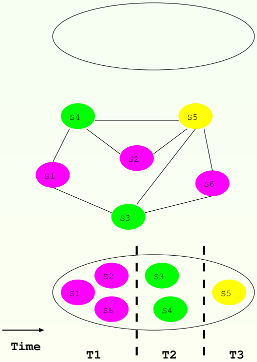
- Computational graph (Bauer, 1974)
- Graph transformations
  - Vertex, edge, face elimination (Griewank, Forth, Pryce, Tadjouddine, Naumann, Utke...)
- Graph coloring (Coleman, More, Steihaug, Hossain, Verma, ...)
- Checkpoint placement in reverse mode (Griewank, Walther, ...)
- ...

Coloring in parallel processing

- A distance-1 coloring of $G = (V, E)$ is
  - a mapping $\phi : V \rightarrow \{1, 2, \ldots, q\}$ s.t. $\phi(u) \neq \phi(v)$ whenever $(u, v) \in E$
  - a partitioning of V into q independent sets

  The objective is to minimize q

- Distance-1 coloring is used to discover concurrency in parallel scientific computing. Examples:
  - iterative methods for sparse linear systems (Jones & Plassmann, 94)
  - adaptive mesh refinement
  - preconditioners (Saad, 96; Hysom & Pothen, 01)
  - eigenvalue computation (Manne, 98)
  - sparse tiling (Strout et al, 02)



Coloring in automatic differentiation: context

Procedure SPARSECOMPUTE($F : R^n \rightarrow R^m$)

S1. Determine the sparsity structure of the derivative (first or second) matrix $A \in R^{m \times n}$ of the function F

S2. Obtain a seed matrix $S \in \{0, 1\}^{n \times q}$ with the smallest q

S3. Compute the numerical values of the entries of the compressed matrix $B = AS \in R^{m \times q}$

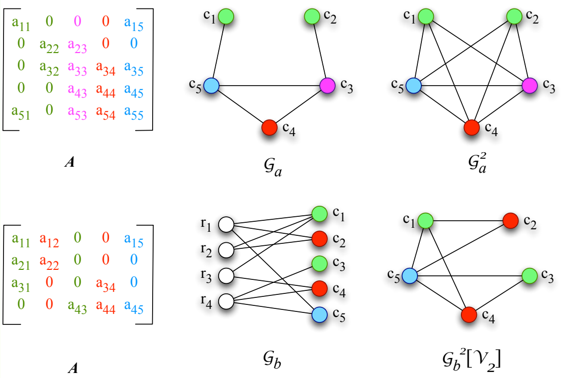
S4. Recover the numerical values of the entries of A from B

The seed matrix S partitions the columns of A:

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

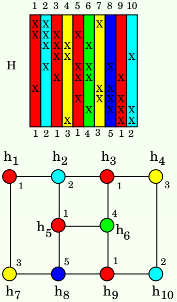
It is obtained using an appropriate coloring on the graph of A.

An archetypal model for direct methods



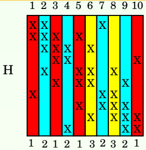
Structurally orthogonal partition of matrix A equivalent to:

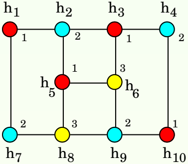
- Distance-2 coloring of the adjacency graph $G_a(A) = (V, E)$ when A is symmetric (McCormick, 1983)
- Partial distance-2 coloring of the bipartite graph $G_b(A) = (V_1, V_2, E)$ when A is nonsymmetric (GMP, 2005)
- Distance-1 coloring of the appropriate square graph (Coleman and Moré, 1983)



$$\begin{pmatrix} h_{11} & h_{12} & h_{17} & 0 & 0 \\ h_{21}+h_{23}+h_{25} & h_{22} & 0 & 0 & 0 \\ h_{33} & h_{32} & h_{34} & h_{36} & 0 \\ h_{43} & h_{4,10} & h_{44} & 0 & 0 \\ h_{55} & h_{52} & 0 & h_{56} & h_{58} \\ h_{63}+h_{65}+h_{69} & 0 & 0 & h_{66} & 0 \\ h_{71} & 0 & h_{77} & 0 & h_{78} \\ h_{85}+h_{89} & 0 & 0 & 0 & h_{88} \\ h_{99} & h_{9,10} & 0 & h_{96} & h_{98} \\ h_{10,9} & h_{10,10} & h_{10,4} & 0 & 0 \end{pmatrix}$$

- Symmetrically orthogonal partition: whenever $h_{ij} \neq 0$
  - h_j only column in a group with nonzero at row i or
  - h_i only column in a group with nonzero at row j
- Star coloring: a vertex coloring ϕ of $G_a(H)$ s.t.
  - ϕ is a distance-1 coloring and
  - every path on 4 vertices (P_4) uses at least 3 colors
- SymOP equivalent to star coloring (Coleman and Moré, 84)





$$\begin{pmatrix} h_{11} & h_{12}+h_{17} & 0 \\ h_{21}+h_{23}+h_{25} & h_{22} & 0 \\ h_{33} & h_{32}+h_{34} & h_{36} \\ h_{43}+h_{4,10} & h_{44} & 0 \\ h_{55} & h_{52} & h_{56}+h_{58} \\ h_{63}+h_{65} & h_{69} & h_{66} \\ h_{71} & h_{77} & h_{78} \\ h_{85} & h_{87}+h_{89} & h_{88} \\ h_{9,10} & h_{99} & h_{96}+h_{98} \\ h_{10,10} & h_{10,4}+h_{10,9} & 0 \end{pmatrix}$$

- Substitutable partition: whenever $h_{ij} \neq 0$
  - h_j in a group where all nonzeros in row i are ordered before h_{ij} or
  - h_i in a group where all nonzeros in row j are ordered before h_{ij}
- Acyclic coloring: a vertex coloring ϕ of $G_a(H)$ s.t.
  - ϕ is a distance-1 coloring and
  - every cycle uses at least 3 colors
- Substitutable partition equivalent to acyclic coloring (Coleman and Cai, 86)

General sparsity pattern:

	unidirectional partition	bidirectional partition	
Jacobian	distance-2 coloring	star bicoloring	Direct
Hessian	star coloring (restricted star coloring)	NA	Direct
Jacobian	NA	acyclic bicoloring	Substitution
Hessian	acyclic coloring (triangular coloring)	NA	Substitution

Nonsym A $\quad G_b(A) = (V_1, V_2, E)$

Sym A $\quad G(A) = (V, E)$

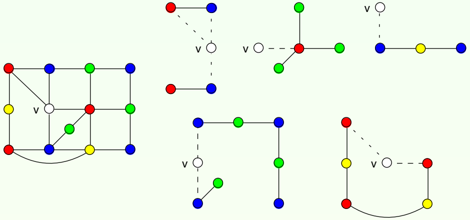
Regular sparsity pattern (discretization of structured grids):

- Formula-based coloring (Goldfarb and Toint, 1984)
- Hierarchical coloring (Hovland, 2007)

Outline

Complexity and Algorithms

- Distance-k, star, and acyclic coloring are NP-hard (they are also hard to approximate)
- A greedy heuristic usually gives a good solution
  GREEDY($G = (V, E)$)
  Let $v_1, v_2, \ldots, v_n$ be an ordering of V
  for $i = 1$ to n **do**
  determine colors forbidden to v_i
  assign v_i the smallest permissible color
  end-for
- For distance-k coloring, GREEDY can be implemented to run in $O(n\bar{d}_k)$ time, where $\bar{d}_k$ is the average degree-k
- We have developed $O(n\bar{d}_2)$-time heuristic algorithms for star and acyclic coloring
  Key idea: exploit the structure of two-colored induced subgraphs



Algorithm (Input: $G = (V, E)$):
for each $v \in V$

1. Choose color for v
   - forbid colors used by neighbors $N(v)$ of v
   - forbid colors leading to two-colored cycles
     - $\forall$ tree T incident on v, if v adj to ≥ 2 vertices of *same* color, forbid the other color in T
2. Update collection of two-colored trees (merge if necessary)

Time: $O(|V|\overline{d}_2 \cdot \alpha)$ Space: $O(|E|)$

Theorem: For every chordal graph $G = (V, E)$

$$\omega(G) = \chi_1(G) = \chi_a(G) \leq \beta(G) + 1$$

$$\chi_s(G) \leq \chi_2(G) = \omega(G^2) \leq \min\{2\beta(G) + 1, |V|\}$$

All inequalities become equalities when G is a band graph.

Experimental results:

ρ, $\overline{\rho}$	10, 10.98		20, 20.99	
	star	acyclic	star	acyclic
banded	11	6	21	11
random	21 – 24	9 – 11	50 – 56	18 – 19

Observed results for banded matrices are *optimal*.

Procedure SparseCompute($F : R^n \to R^m$)

S1. Determine the sparsity structure of the derivative (first or second) matrix $A \in R^{m \times n}$ of the function F

S2. Obtain a seed matrix $S \in \{0,1\}^{n \times q}$ with the smallest q

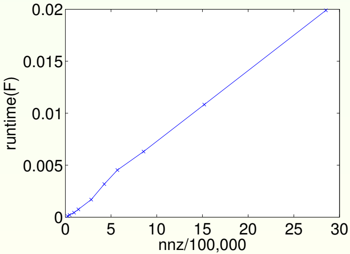
S3. Compute the numerical values of the entries of the compressed matrix $B = AS \in R^{m \times q}$

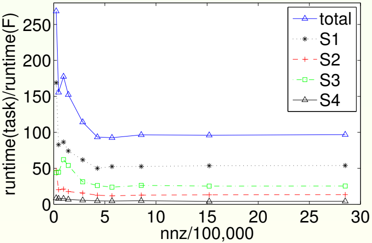
S4. Recover the numerical values of the entries of A from B

S1: Sparsity Detection
S2: Coloring
S3: Computation of Compressed Matrices
S4: Recovery of derivative matrix elements

Experiments using ADOL-C

- Efficacy of the four-step scheme tested in two case studies
  1. Jacobian computation in a Simulated Moving Bed process (chromatography) Walther AD08 talk
  2. Hessian computation in an optimal electric power flow problem
- Experiments showed
  - technique enabled cheap Jacobian/Hessian computation where dense computation is infeasible
  - observed results for each step matched analytical results





Outline

Parallelizing greedy coloring

- Desired task: parallelize Greedy such that
  - speedup is $\Theta(p)$
  - number of colors used is roughly same as in serial
- A difficult task since Greedy is inherently sequential
- For D1 coloring, several approaches based on Luby's parallel algorithm for maximal independent set exist
- Some drawbacks:
  - no actual parallel implementation
  - many more colors than a serial implementation
  - poor parallel speedup on unstructured graphs

(Relaxed) Partitioning applied to greedy coloring

Basic features of the algorithm:

- exploits features of data distribution
  - distinguishes between interior and boundary vertices
- proceeds in rounds, each having two phases:
  - tentative coloring
  - conflict detection
- tentative coloring phase organized in supersteps
  - each processor communicates only after coloring a subset of its assigned vertices using currently available information (infrequent, coarse-grain communication)
- randomization used in resolving conflicts

A Framework for Parallel Distance-1 Coloring

Framework($G = (V, E), s$)
Partition V into $V_1, V_2, \ldots, V_p$ using a graph partitioner
(Processor P_i owns (and colors) V_i, and stores edges E_i incident on V_i.)
On each processor P_i, $i \in I = \{1, \ldots, p\}$
  while uncolored vertices remain **do**   (rounds)
    Partition uncolored vertices into subsets of size s
      (supersteps for tentative coloring)
    for each superstep **do**
      Tentatively color vertices in the superstep
      Send colors of boundary vtxs to relevant processors
      Receive color information from relevant processors
    Wait until all incoming messages are received
    Detect conflicts for boundary vertices
    Uncolor incorrectly colored boundary vertices

Specializations of Framework

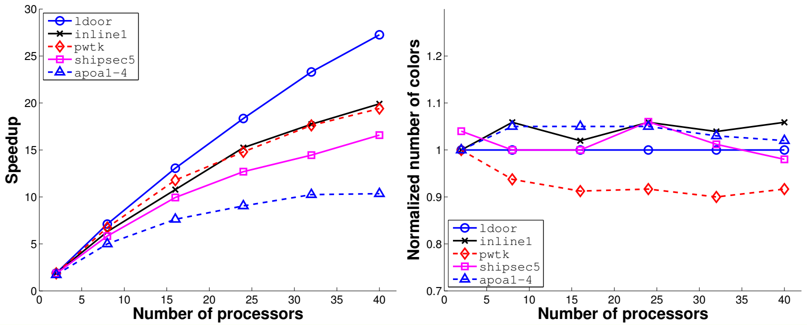
Framework can be specialized along several axes:

1. Color selection strategies:
   First Fit: search for smallest color starts at 1 on each processor
   Staggered FF: search for smallest color starts from different "bases"
2. Coloring order:
   interior vertices can be colored before, after, or interleaved with boundary vertices
3. Local vertex ordering:
   vertices on each processor can be ordered using various degree-based techniques
4. Supersteps:
   can be run synchronously or asynchronously
5. Inter-processor communication:
   can be customized or broadcast-based

Lessons learned from experiments

Good parameter configuration for
large-size (millions of edges) graphs:

- moderately unstructured graphs
  (e.g. a typical application graph):
  1. a superstep size s in the order of 1000
  2. asynchronous supersteps
  3. a coloring order in which interior vertices appear either strictly before or strictly after boundary vertices
  4. First Fit color choice strategy
  5. customized inter-processor communication
- highly unstructured (e.g. random) graphs:
  - s in the order of 100
  - items 2 to 4 same as for moderately unstructured graphs
  - broadcast-based communication



Algorithm FBAC on Itanium 2 cluster.

Summary

- Current accomplishments:
  - Designed and implemented new sequential algorithms for distance-k, star, acyclic, and other coloring problems.
  - C++ implementations and ordering functions assembled in a package called ColPack.
  - Integrated parts of ColPack with the AD tool ADOL-C.
  - Developed parallel algorithms for distance-1, distance-2, and restricted star coloring, available through the Zoltan package.
- Planned activities:
  - Integrate coloring software with tools in OpenAD.
  - Develop parallel coloring algorithms for tera- and peta-scale computation.
  - Collaborate with chemical engineers to solve coupled PDEs with cyclic boundary conditions from liquid chromatographic and gaseous separations.

Outline

1. The CSCAPES Institute
2. Combinatorics in AD
   - Coloring Models
   - Sequential algorithms
   - Case studies
   - Parallel algorithms
   - Summary
3. The Way Forward

More about CSCAPES, SciDAC

- www.cscapes.org, www.scidac.org
- CSCAPES article in SciDAC Review, Fall 2007, pp. 26–35.
- CSCAPES Institute seminars using remote conferencing systems
- SIAM Workshop on CSC 2009, Oct. 29-31, 2009 (Monterey Beach CA)
- Oberwolfach Workshop on CSC, Feb. 2009
- Annual SciDAC conferences, SciDAC Review (online magazine)
- DEISA: Distributed European Infrastructure for Supercomputer Applications.

More about AD and CSC Communities

- Need to increase participation in our communities. Both have steep learning curves as price of admission. Planned survey by Gebremdhin and Naumann on Combinatorics in AD.
- Strengthen existing collaborations in CSCAPES Institute and elsewhere.
- Develop robust AD software efficient for many core architectures.
- Collaborate with researchers working on tera- and peta-scale simulations: intensive computations, large data sets, deep memory hierarchies, and non-uniform communication costs: opportunities for combinatorial algorithms.

Further reading

Hendrickson and Pothen,
CSC: The enabling power of discrete algorithms in computational science.
LNCS, 4395, pp. 260-280, 2007.

Gebremedhin, Manne and Pothen.
SIAM Review 47(4):629–705, 2005.

Gebremedhin, Tarafdar, Manne and Pothen.
New Acyclic and Star Coloring Algorithms with Application to Computing Hessians.
SIAM J. Sci. Comput. 29:1042–1072, 2007.

Gebremedhin, Pothen and Walther.
Exploiting Sparsity in Jacobian Computation via Coloring and AD:
A Case Study in a Simulated Moving Bed Process.
Fifth International Conference on AD, Bonn, Germany, Aug 2008, to appear. 12 pp.

Gebremedhin, Pothen, Tarafdar and Walther.
Efficient Computation of Sparse Hessians using Coloring and AD.
INFORMS Journal on Computing, to appear. 30 pp.

Bozdag, Gebremedhin, Manne, Boman and Catalyurek.
A Framework for Scalable Greedy Coloring on Distributed-memory Parallel Computers.

J. Parallel Distrib. Comput. 68(4):515–535, 2008.