The Enabling Power of Graph Coloring Algorithms in Automatic Differentiation and Parallel Processing

Assefaw H. Gebremedhin

Department of Computer Science and Computing Research Institute
Purdue University, IN, USA
agebreme@purdue.edu

Abstract. We illustrate the essential features of Combinatorial Scientific Computing using our work on graph coloring algorithms for sparse derivative computation as an example.

Keywords. Graph coloring, sparse derivative computation, automatic differentiation, parallel computing

1 Combinatorial Scientific Computing

Combinatorial scientific computing (CSC) is founded on the recognition of the enabling power of combinatorial algorithms in scientific and engineering computation and in high-performance computing. The domain of CSC extends beyond traditional scientific computing—the three major branches of which are numerical linear algebra, numerical solution of differential equations, and numerical optimization—to include a range of emerging and rapidly evolving computational and information science disciplines. Examples of the latter include computational biology, chemistry, and physics; computational climate and material sciences; statistical physics; bioinformatics; and large-scale data management and analysis. Orthogonally, CSC problems could also emanate from infrastructural technologies for supporting high-performance computing. Among such technologies are algorithmic tools for data partitioning, load balancing, and task scheduling in parallel computing, and tools for data and iteration reordering in irregular computation.

Despite the apparent disparity in their origins, CSC problems and scenarios are unified by the following common features:

- The overarching goal is often to make computation efficient—by minimizing overall execution time, memory usage, and/or storage space—or to facilitate knowledge discovery or analysis.
- Identifying the most accurate combinatorial abstractions that help achieve this goal is usually a part of the challenge.
- The abstractions are often expressed, with advantage, as graph or hypergraph problems.

- The identified combinatorial problems are typically NP-hard to solve optimally. Thus, fast, often linear-time, approximation (or heuristic) algorithms are the methods of choice.
- The combinatorial algorithms themselves often need to be parallelized, to avoid their being bottlenecks within a larger parallel computation.
- Implementing the algorithms and deploying them via software toolkits is critical.

2 Coloring in Derivative Computation

The talk given at the Dagstuhl Seminar on CSC (09061) attempted to illustrate the aforementioned features of CSC through an example: We considered the enabling role graph coloring models and their algorithms play in efficient computation of sparse derivative matrices via automatic differentiation (AD). We focused on efforts being made on this topic within the SciDAC Institute for Combinatorial Scientific Computing and Petascale Simulations (CSCAPES). Aiming at providing overview than details, we discussed the various coloring models used in sparse Jacobian and Hessian computation, the serial and parallel algorithms developed in CSCAPES for solving the coloring problems, and a case study that demonstrates the efficacy of the coloring techniques in the context of an optimization problem in a Simulated Moving Bed process, a purification technique used in the chemical industry [1,2,3,4,5,6].

Implementations of our serial algorithms for the coloring and related problems in derivative computation are assembled and made publicly available in a package called `ColPack`. Implementations of our parallel coloring algorithms are incorporated into and deployed via the load-balancing toolkit `Zoltan`. `ColPack` has been interfaced with `ADOL-C`, an operator overloading-based AD tool that has recently acquired improved capabilities for automatic detection of sparsity patterns of Jacobians and Hessians (sparsity pattern detection is the first step in derivative matrix computation via coloring-based compression). Further information on `ColPack` and `Zoltan` is available at their respective web sites, which can be accessed via http://www.cscapes.org.

3 Coloring in Parallel Scientific Computing

Graph coloring is also useful for discovering computational subtasks that can be carried out concurrently in a parallel scientific computing (PSC) context. At this Dagstul Seminar, a roundtable discussion that sought to identify some specific applications within PSC where the use of coloring might need to be further investigated was held. Three applications were discussed in some detail: adaptive mesh refinement, preconditioned iterative methods for sparse linear systems (where “multi-coloring” is used), and full sparse tiling for sparse equation solvers. Of these, full sparse tiling and multi-coloring, especially in the blocked case, were considered promising avenues for further investigation.

References

1. Gebremedhin, A., Manne, F., Pothen, A.: What color is your Jacobian? Graph coloring for computing derivatives. SIAM Review **47** (2005) 629–705
2. Gebremedhin, A., Tarafdar, A., Manne, F., Pothen, A.: New acyclic and star coloring algorithms with applications to Hessian computation. SIAM J. Sci. Comput. **29** (2007) 1042–1072
3. Gebremedhin, A., Pothen, A., Tarafdar, A., Walther, A.: Efficient computation of sparse Hessians using coloring and automatic differentiation. INFORMS Journal of Computing **21** (2009) 209–223
4. Gebremedhin, A., Pothen, A., Walther, A.: Exploiting sparsity in Jacobian computation via coloring and automatic differentiation: A case study in a simulated moving bed process. In Bischof, C., Bucker, H., Hovland, P., Naumann, U., Utke, J., eds.: Proceedings of AD2008. Volume 64 of Lecture Notes in Computational Science and Engineering., Springer (2008) 339–349
5. Bozdağ, D., Gebremedhin, A., Manne, F., Boman, E.G., Catalyurek, U.V.: A framework for scalable greedy coloring on distributed-memory parallel computers. Journal of Parallel and Distributed Computing **68** (2008) 515–535
6. Bozdağ, D., Catalyurek, U., Gebremedhin, A., Manne, F., Boman, E.G., Ozgunner, F.: Distributed-memory parallel algorithms for distance-2 coloring and their application to derivative computation. (2009) Submitted to SISC. A preliminary version appeared in LNCS 3726, pp 796–806, 2005.