Advanced Scientific Computing Research
FY 2006 Accomplishment

Mathematical Programming without Derivatives – Bringing Optimization to the Masses

Tamara G. Kolda[*] and Joshua D. Griffin, Sandia National Laboratories

Summary

Optimization is often daunting to engineers and scientists because it involves complicated derivative calculations and the software is hard to use. We have developed robust derivative-free methods and software that are easy to use, run efficiently in parallel (even when the scientist's code is not parallel), and are mathematically proven to converge.

Our focus is on solving real-world optimization problems from science and engineering – the types of problems that are based on complicated (perhaps even stochastic) simulations that are expensive and do not have any derivative information.

For example, we are currently collaborating on a problem in nuclear safety studies where we want to determine the worst-case damage if a component is dropped. Each "function evaluation" requires meshing, a parallel simulation (that takes 1-15 hours on ten processors of an advanced parallel supercomputer), and analysis of the simulation output to determine the final objective value, i.e., the damage.

Few optimization methods or codes can solve such problems and fewer still can do so efficiently. We have designed a parallel and *asynchronous* derivative-free optimization method that reliably and robustly solves these types of optimization problems. Allowing different simulations to be run independently and asynchronously is the key to our success. Much time is wasted on supercomputers for synchronization, but our methods fix these problems by more efficiently making use of available resources. Over and over again, our methods have proven to be faster than synchronous implementations; see Figure 1.

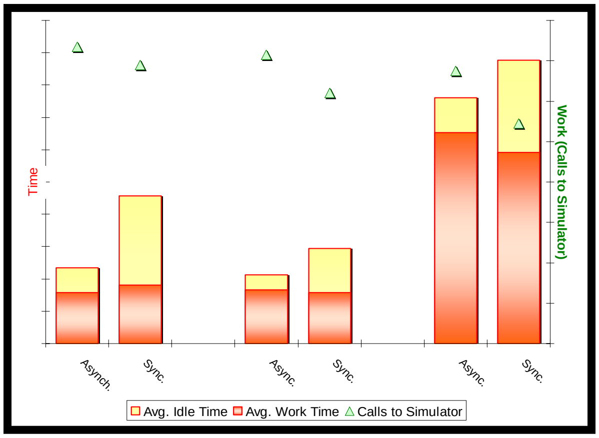

Figure 1. Numerical results on 3 problems; our asynchronous method (left) is faster than synchronous (right). Orange bars show actual work time and yellow shows idle time. Green triangles indicate calls to the simulator; the asynchronous method makes more calls but is still faster due to its efficiency.

This year, we have extended our methods to handle linear and nonlinear constraints. This is a major advancement and means that more complicated nonlinear programming problems can now be solved. Moreover, our convergence theory proves that these

---

[*] (925) 294-4769, tgkolda@sandia.gov

methods converge to a constrained stationary point – i.e., a true mathematical solution that is as good as would be achieved using derivative-based methods (if that were an option).

Because these methods are easy to use and robust, they are proving useful in a wide variety of scientific domains. Published examples of the use of APPSPACK include parameter estimation for thermal design, microfluidics channel design, forging process design for structure strength, transmembrane protein structure prediction, optimal pump placement and rate calculation for groundwater flow, fitting statistical models for image processing, and optimal control of a fed-batch fermentation process. Our codes have even found their way into the computer and social sciences, having recently been used for parameter tuning for support vector machines and in social dynamics modeling.

Our methods are implemented in the code APPSPACK, which is freely downloadable from http://software.sandia.gov/appspack under the terms of the GNU L-GPL license. Version 5.0 was released on June 30, 2006 and now supports linear equality and inequality constraints. APPSPACK runs in parallel using MPI (see Figure 2), and it interfaces with the simulation via file I/O, meaning that the simulation can be written any language.

In the past year, one paper appeared in the *SIAM Journal on Optimization*, another was accepted for future publication in the same journal, a third was accepted to the *ACM Transactions on Mathematical Software*, and a fourth was submitted to the *SIAM Journal on Scientific Computing*. We are active in the community, presenting invited lectures at scientific meetings.

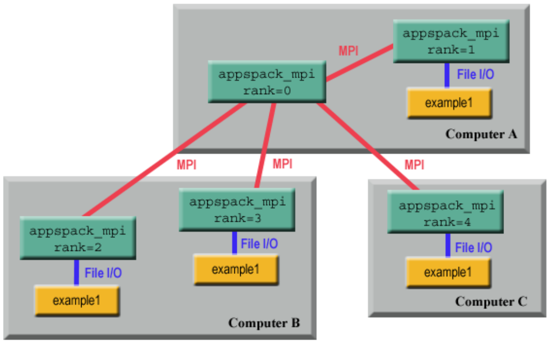

Figure 2: An illustration of APPSPACK running in parallel on 3 processors.

We will continue to work on the development of methods to handle problems with nonlinear constraints. This involves solving a sequence of linearly-constrained subproblems. We are developing convergence theory as well as implementing and testing the methods. We will also further our examination of why these methods behave so well in the presence of noise even though the convergence theory does not account for it. To do this, we are developing new convergence theory that explains this good behavior and offers some clues on to how to best select algorithmic parameters in the presences of noise.

Our work with applications has been aided by our colleagues at Sandia National Laboratories. Our theoretical work has been in collaboration with Prof. Virginia Torczon and Prof. R. Michael Lewis at the College of William and Mary in Virginia, as well as Prof. Richard Byrd from the University of Colorado at Boulder. The development of the APPSPACK software has also been partly supported by the NNSA.

For further information on this subject contact:

Dr. Anil Deane, Program Manager
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
deane@mics.doe.gov