Why Optimization Is Faster than Solving Systems of Equations: A Qualitative Explanation

Siyu Deng[1], Bimal K C[2], and Vladik Kreinovich[1,2]
[1]Department of Computer Science
[2]Computational Science Program
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
sdeng@miners.utep.edu, bkc@miners.utep.edu,
vladik@utep.edu

Formulation of the problem. At first glance, these two problems can be reduced to each other, so their computational complexity should be comparable:

- optimization $f \to \min$ can be reduced to solving a system of equations obtained by equating all partial derivatives to 0: $\dfrac{\partial f}{\partial x_i} = 0$; and
- solving a system of equations $f_1(x_1, \ldots, x_n) = 0, \ldots, f_n(x_1, \ldots, x_n) = 0$ is equivalent to minimizing the sum $\sum\limits_{i=1}^{n} (f_i(x_1, \ldots, x_n))^2$.

However, empirically, optimization is faster; see, e.g., [1]. How can we explain this?

Possible explanation. In general, the more inputs we have, the more computation time the problem requires. For both optimization problem and the problem of solving a system of equation, the inputs are functions. For our two problems:

- To describe an optimization problem, we need to describe only one function $f(x_1, \ldots, x_n)$ – which is minimized (or maximized).
- On the other hand, to describe a system of n equations with n unknowns, we need to describe n functions $f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)$.

So, not surprisingly, optimization problems are, in general, faster to solve.

References

[1] R. L. Muhanna, R. L. Mullen, and M. V. Rama Rao, "Nonlinear Interval Finite Elements for Beams", *Proceedingds of the Second International Conference on Vulnerability and Risk Analysis and Management (ICVRAM) and the Sixth International Symposium on Uncertainty, Modeling, and Analysis (ISUMA)*, Liverpool, UK, July 13–16, 2014.