Nested Methods and a Superior Alternative

Kenneth Judd
Hoover Institution

January 29, 2016

Constrained Optimization by Nesting

- Maximization: Suppose that we want to solve

$$\max f(x, Y(x))$$

where $Y(x)$ is the solution to some other numerical problem described by $g(x, y) = 0$. Nested approach has two layers

  - Inner loop: compute $Y(x)$, using your own code - BAD idea!!!!
  - Outer loop: for each x, compute $f(x, Y(x))$ in an unconstrained optimization algorithm written by user - BAD idea!!!

- General experience: Nested methods are slow and unreliable

  - Bad code for inner loop tempts you to use loose stopping rule for inner loop, which requires very loose stopping rule for outer loop (Rust used tight stopping rule for inner loop; BLP and others use sloppy stopping rules)
  - You need to compute $Y(x)$ for each value of x used in the outer loop
    - Finite difference methods used for outer loop derivatives
    - Slow methods lead to inferior econometrics: no bootstrapping, use "computationally light" estimators
    - End up with a job at a HYPMS university.

Math Programming to the Rescue!

- Suppose that you want to solve

$$\max f(x, Y(x))$$

where $Y(x)$ is the solution to some other numerical problem.

$$0 = g(x, y)$$

  - Reformulate problem as

$$\max_{x,y} f(x, y)$$
$$\text{s.t. } 0 = g(x, y)$$

  - Advantages
    - You can use a solver written by professionals.
    - No need for you to write code to compute $Y(x)$
    - You can set tight stopping rules for all variables, y and x.
    - You can try several solvers to find the one that works best
    - Never have to worry about getting offers from HYPMS cartel
  - Disadvantage: Memory cost is too large IF you don't use good solvers that (a) exploit sparseness, and (b) use automatic differentiation.
- Lesson: Learn some math so that you can get the computer to do the hard work.