Numerical Optimization: Introduction

Qiang Zhu

University of Nevada Las Vegas

May 12, 2020

Overview

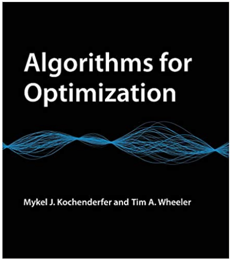
Syllabus

We have two goals:

- Learn Julia programming
- Understand the optimization methods

Subjects to be covered

- Julia programming
- Local Optimization
  - Derivatives and Gradients
  - Bracketing
  - First/second-Order optimization
  - Gradient free methods
  - Stochastic methods
- Global Optimization
- Sampling Plans
- Surrogate Optimization
- Expression Optimization

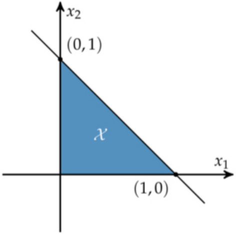


Meeting twice a week (90 minutes each time).

- review of homework (20-30 mins)
- lecture (30-50 mins)
- coding (20-30 mins)

Why optimization

$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & f(x_1, x_2) \\ \text{subject to} \quad & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_1 + x_2 \leq 1 \end{aligned}$$



A typical optimization problem is to

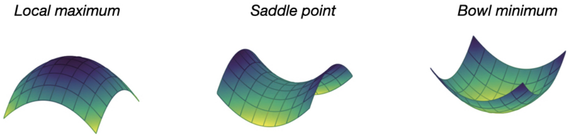
$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$

A design point (x) can be represented as a vector of values corresponding to different design variables.

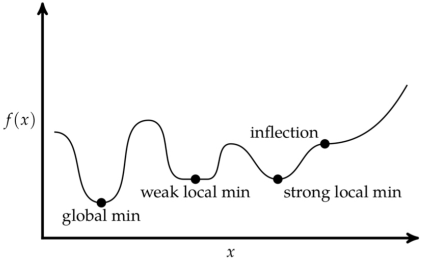
A necessary condition? for $f(x)$ reaches the minimum is that $f'(x) = 0$.

Optimization is hard!

- $f`(x) = 0$ is not a sufficient condition.



- There exist many points where $f`(x) = 0$.

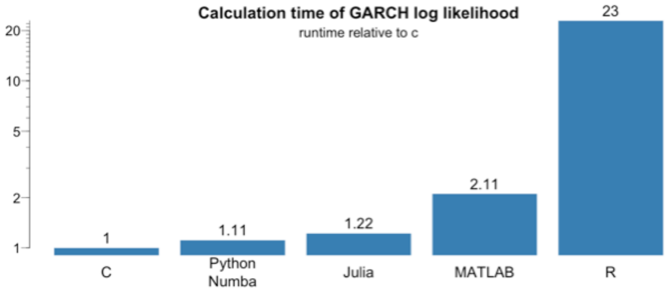


- $f(x)$ and $f`(x)$ are hard to evaluate.

Why Julia?

Run Julia at https://www.juliabox.com

- Math-friendly
- Looks like Python
- Runs like C/Fortran
- Growing ecosystem



Summary

- Optimization in engineering is the process of finding the best system design subject to a set of constraints.
- Optimization can be transformed to a math problem but it is sometimes hard to solve
- We will extensively using the Julia language to learn how to solve the optimization numerically

Homework

In Julia, write the following trial functions, make the contour plots and analyze their minima behavior.

- Booth's function

$$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

- Barnin function

$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t)cos(x_1) + s$$

  where $a = 1, b = 5.1/(4\pi^2), c = 5/\pi, r = 6, s = 10, t = 1/8\pi$.

- Rosenbrock's Banana function

$$f(x_1, x_2) = (a - x_1)^2 + b(x_2 - x_1^2)^2$$

  where $a = 1, b = 5$.

More functions can be found at https://en.wikipedia.org/wiki/Test_functions_for_optimization