Effective Programming Practices for Economists
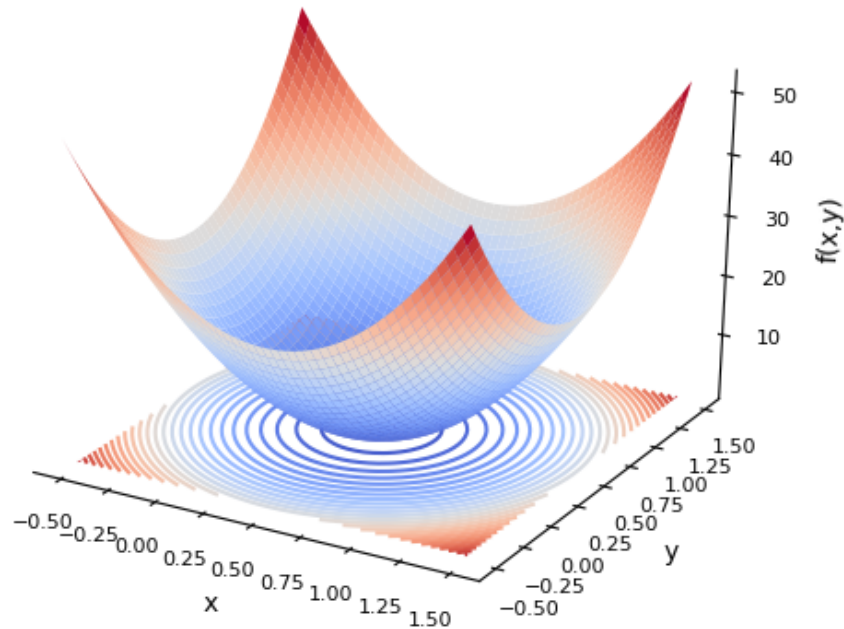
Numerical Optimization

Introduction to numerical optimization

Janoś Gabler and Hans-Martin von Gaudecker

Example problem

- **Criterion** $f(a, b) = a^2 + b^2$

- Parameters a, b

- Want: $a^*, b^* = \mathrm{argmin} f(a, b)$

- Possible extensions:

  - Constraints

  - Bounds (a.k.a. box constraints)

- Optimum at $a^* = 0, b^* = 0$,
  $f(a^*, b^*) = 0$

Applications of numerical optimization

- Probit, many Logit models, …

- Fitting machine learning models

- Estimating structural models

  - Maximum likelihood

  - Method of simulated moments

- Calculating optimal policies from a structural model

- Solving utility maximization problems

What is an optimization algorithm

- **Our definition**: A function that takes a criterion function and start parameters and returns a solution, possibly after a long time

- There are many different optimizers

- Picking the right one can make a huge difference but is hard

- Use a mix of theory and experimentation to get there

Libraries for optimization

- There are many optimization libraries in Python

- All are a bit different

- We will use optimagic to access all of them with a unified interface

- Optimagic development started in Bonn's Open Source Economics group!

- Using optimagic

  - Overview
  - Mechanics of minimising/maximising a function
  - Picking algorithms
  - Visualizing histories

- Intuition behind different algorithms

  - Example used throughout

  - Grid search methods

  - Derivative-based methods

    - Line search
    - Trust region

  - Derivative-free methods

    - Direct search
    - Trust region