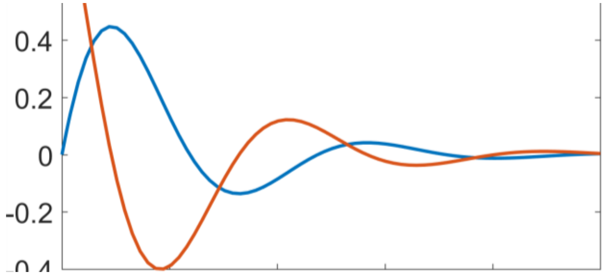
Parameter Estimation

Ahmet Sacan

Parameter Estimation

- Simulation:
  - Can generate predicted values
  - Predictions are dependent on parameters

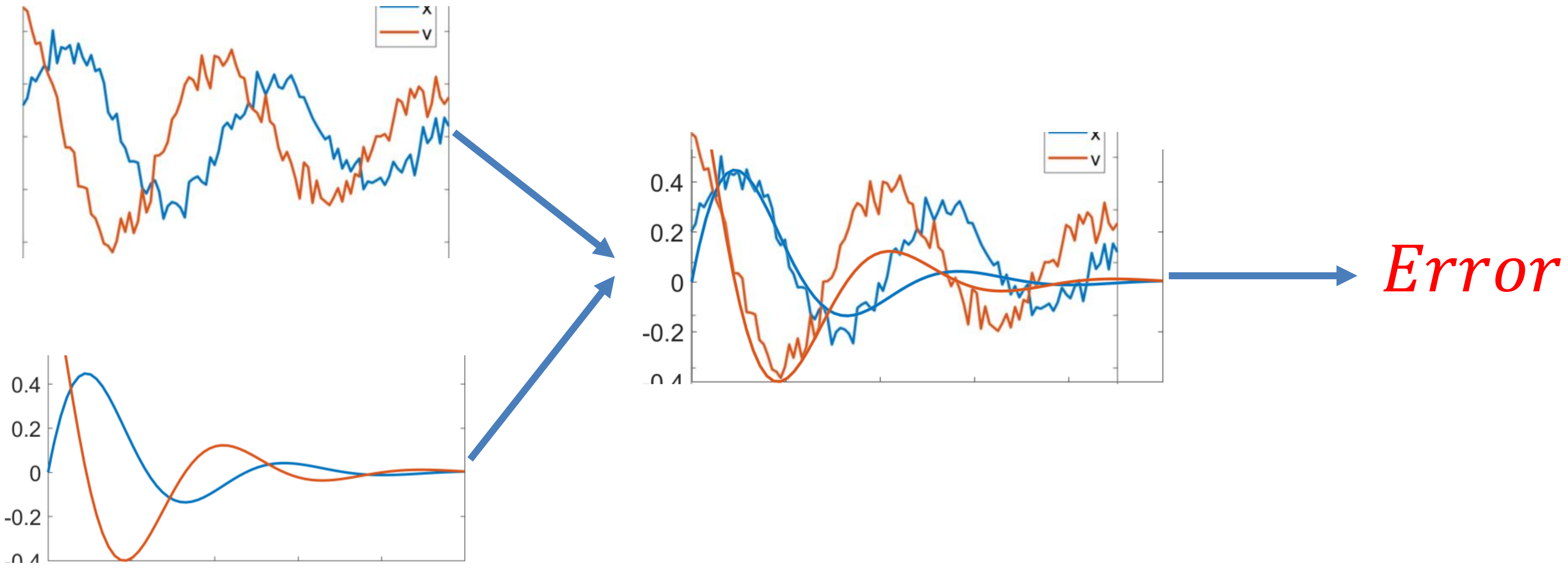


- Experimental Data:
  - "Real-world" measurements
  - Noisy
  - May be condition-dependent

- Parameter Estimation:
  - Reconcile simulation & experimental data
  - Optimize model parameters so the simulation can replicate experimental data (as best as possible)

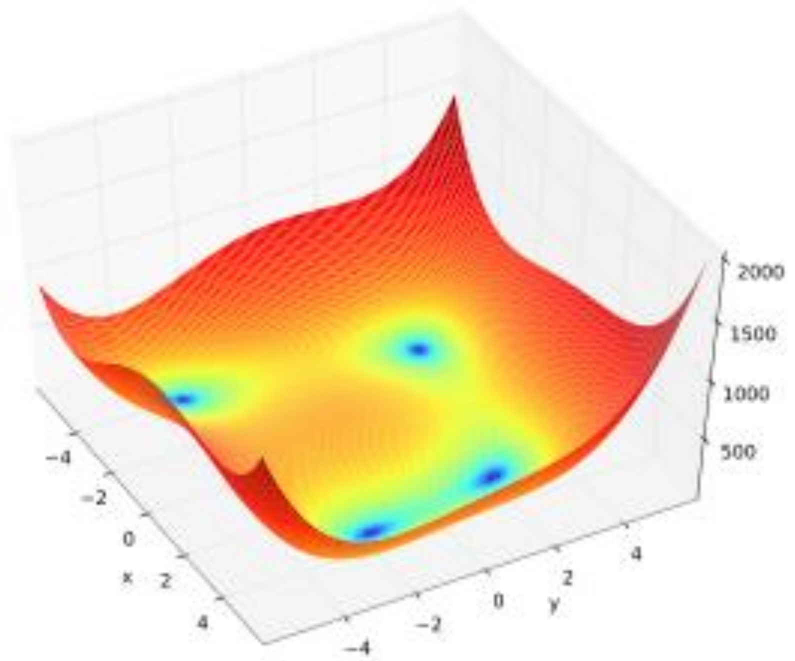
Error/Evaluation/Objective Function

- Define an error function that (for given parameters, θ) quantifies how well simulation values match experimental data
  - $eval(\theta) \rightarrow \color{red}{error}$

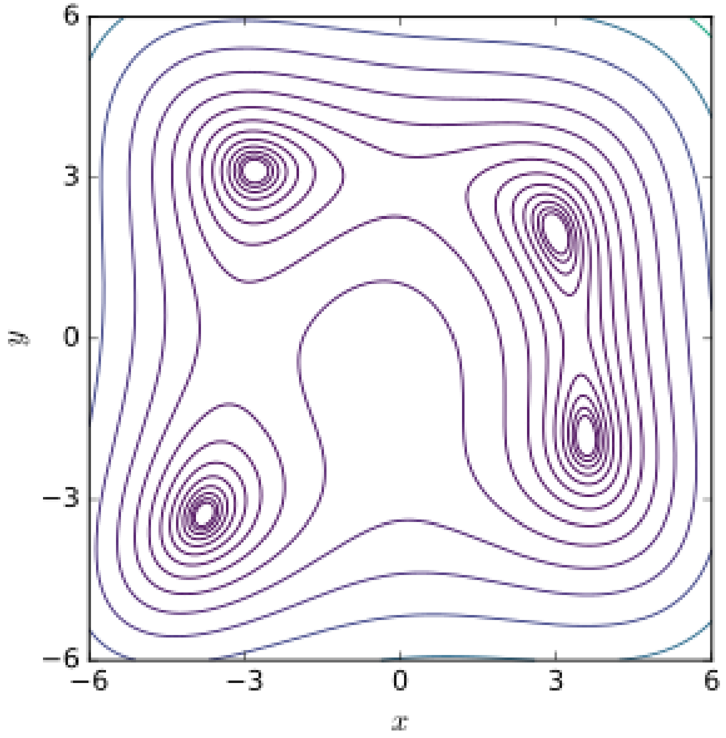


Error Function Example

- Let θ be 2-dimensional, i.e., we can vary two parameters in the model.
- The error function can be visualized in 3D (surface) or in 2D (contour plot, level sets).



https://en.wikipedia.org/wiki/Himmelblau%27s_function



Function Optimization

- Find best θ that results in smallest error.
- If the minimum points of the error function could be calculated analytically, we would have the correct/exact answer.
- But for most problems, closed form solution not available.
- We cannot try all possible parameters (infinitely many!)

Function Optimization Ideas

- Try many different random values of θ, pick the best one.
- Try many equally spaced values of θ, pick the best one. (e.g., meshgrid)

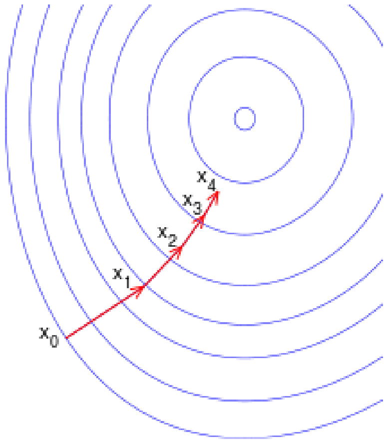
- Start with a random initial θ, and repeatedly improve θ to give a smaller error.
  - How to improve?

Line Search Methods

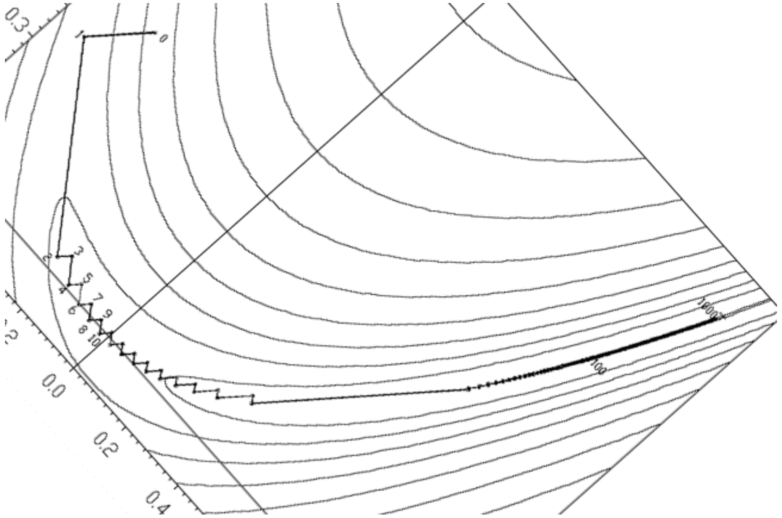
- Start with an initial guess x_0
- Repeat:
  - Find a descent direction along which error function is reduced
    - Methods: Gradient Descent, Newton's method, Quasi-Newton method
  - Choose a step size and move by the step size along that direction.

Gradient Descent

- aka Steepest Descent
- $x_{n+1} = x_n - \eta \boldsymbol{\nabla f(x_n)}$
- Sometimes ∇f is available. When it's not, approximate from data



https://en.wikipedia.org/wiki/Gradient_descent



$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$

Demo: fminunc()

- fminunc: Unconstrained Nonlinear Optimization
- Objective function:
  - $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$

Demo: fmincon()

- fminunc: Constrained Nonlinear Optimization
- Objective function:
    - $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$
- Constraint:
    - $x_1 + 2x_2 \leq 1$
  - Rewrite constraint as: $Ax \leq b$, A=[1,2], b=1.

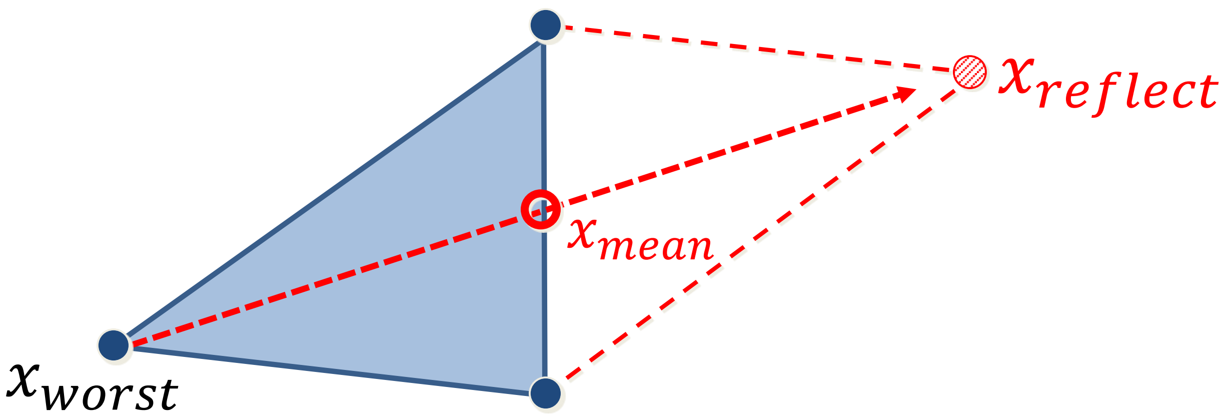
Fmincon exercise

Nelder-Mead Simplex Method

- Keep track of n+1 points in n dimensions
  - e.g., a triangle in 2D.
- At each iteration, do one of:
  - reflect, expand, contract, shrink

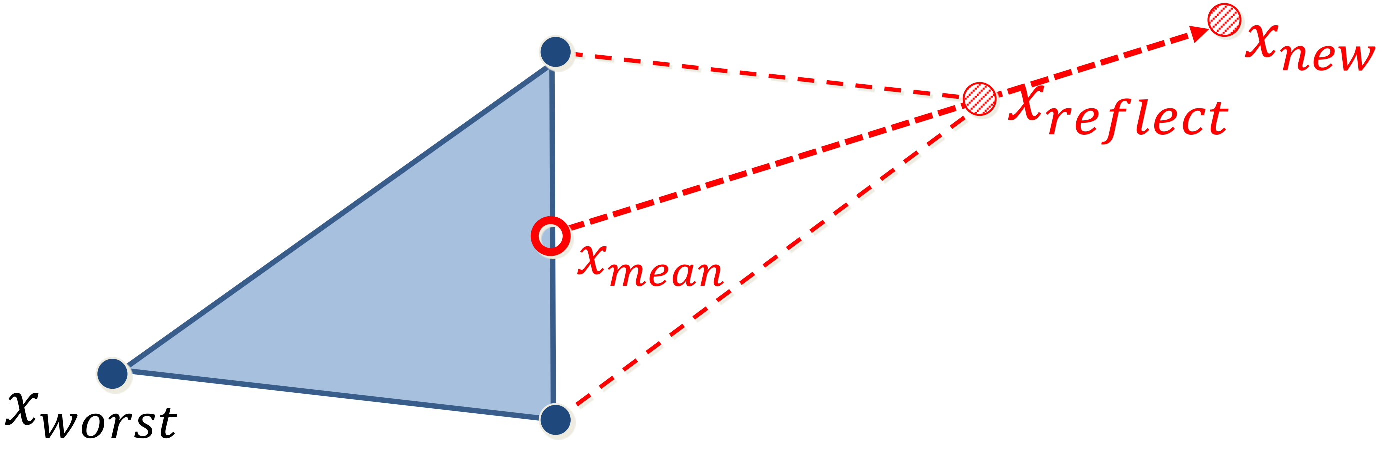
Nelder-Mead

- Basic operation: **Reflect**
  - Calculate f at each point
  - identify the x_{worst}, and its f_{worst}
  - x_m: average of remaining points.
  - Reflect: $x_{reflect} = x_{mean} + R(x_{mean} - x_{worst})$



Is $f_{reflect}$ better than f_{best} ? → **Expand & done.**

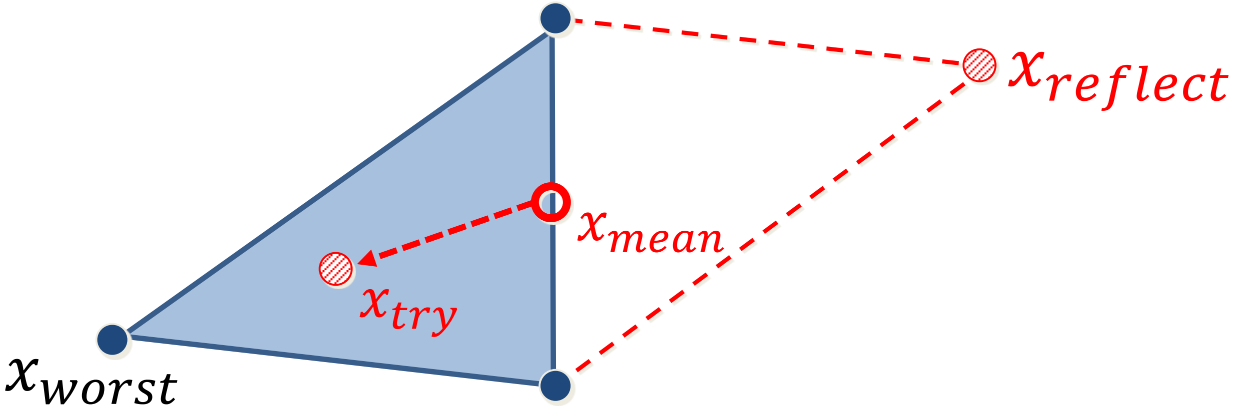
$$x_{new} = x_{mean} + E(x_{reflect} - x_{mean})$$



Is $f_{reflect}$ worse than f_{worst}? → **Contract Inside:**

$$x_{try} = x_{mean} - C(x_{mean} - x_{worst})$$

Shrink if still worse than f_{worst}.

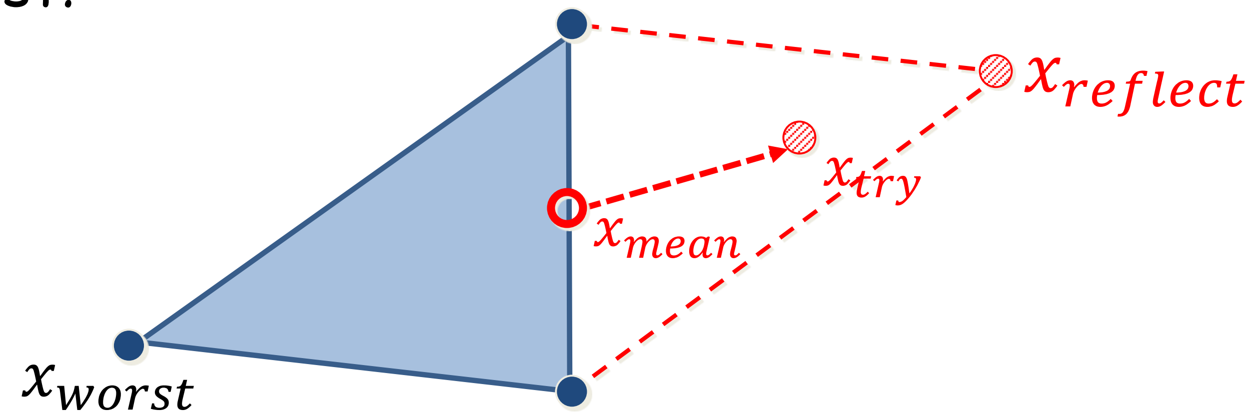


Is $f_{reflect}$ better than f_{worst} but worse than second worst?

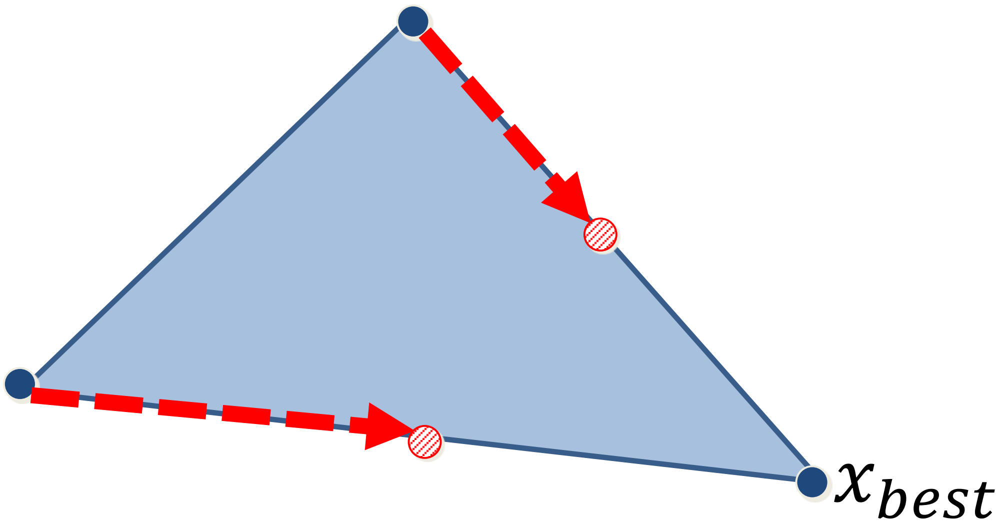
Contract Inside:

$$x_{try} = x_{mean} + C(x_{reflect} - x_{mean})$$

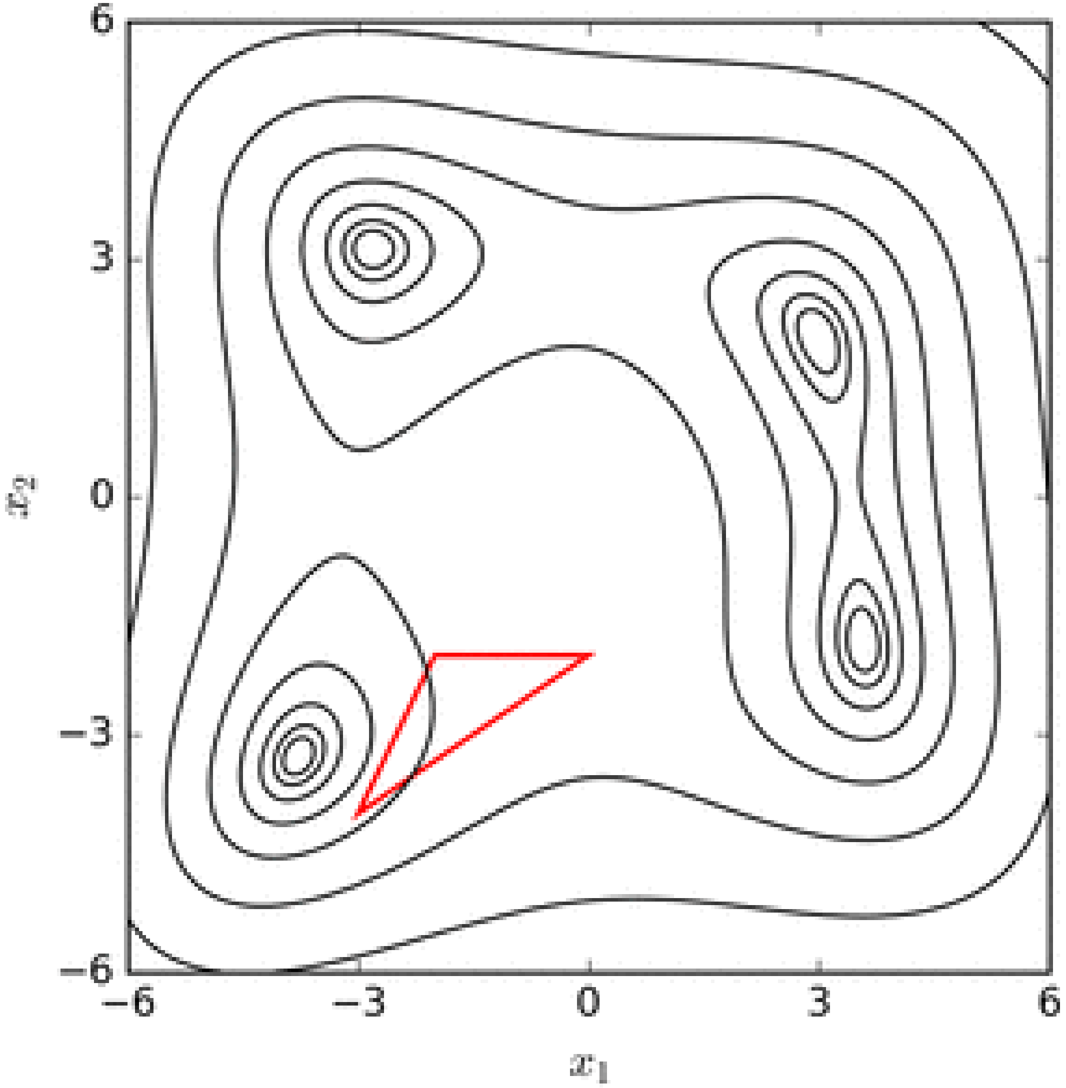
Shrink if worse than $f_{reflect}$.



Shrink: $x_i = x_i + S(x_i - x_{best})$

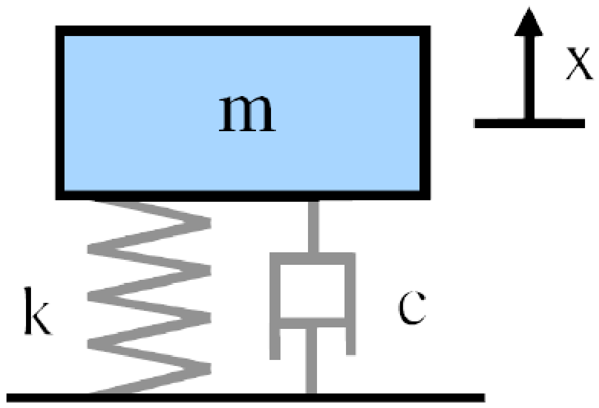


Nelder-Mead



Demo: Mass-Spring-Damper

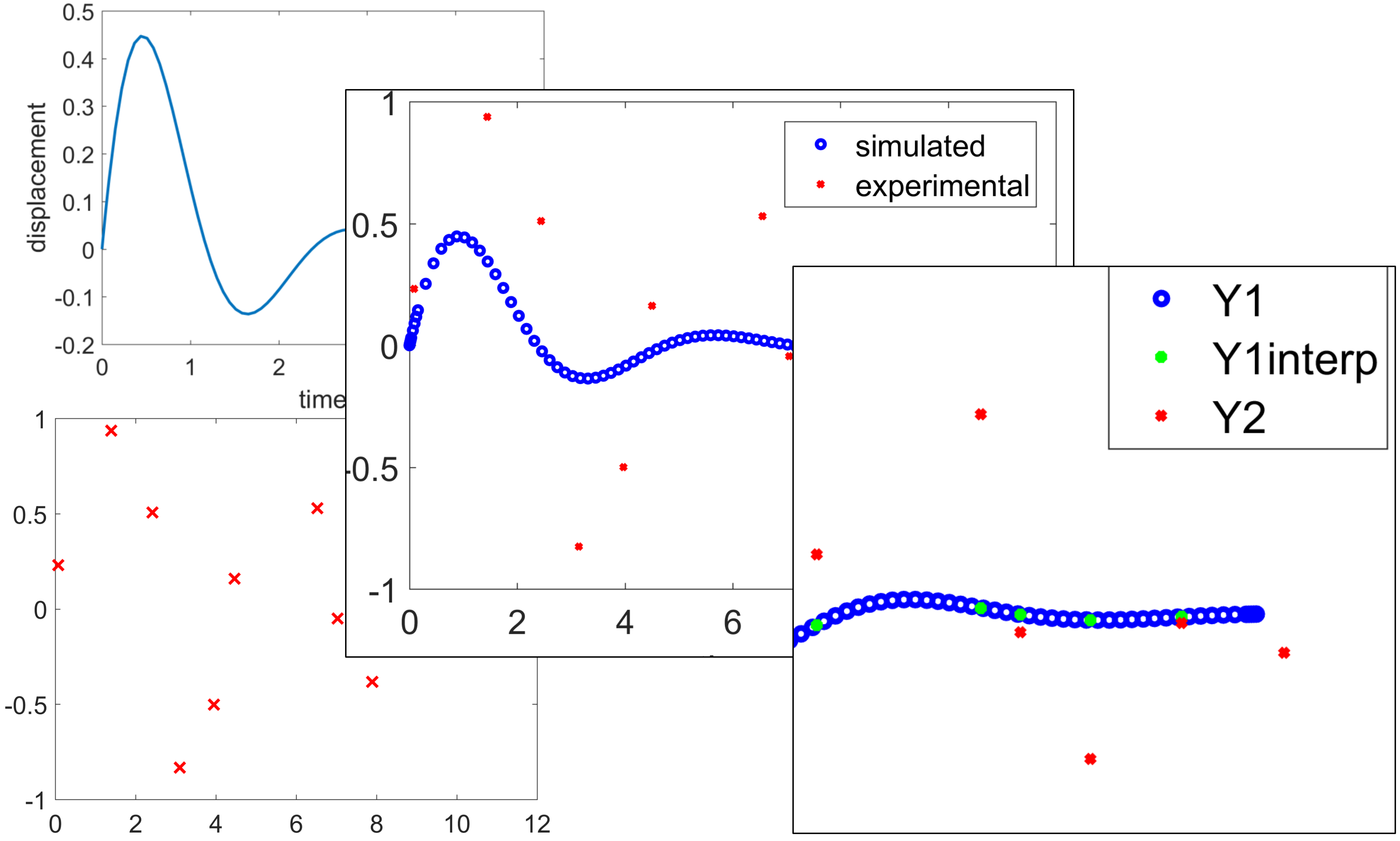
- Integrator1: $\dot{x} \quad \rightarrow \quad x$
- Integrator2: $\dot{v} \quad \rightarrow \quad v = \dot{x}$
- Implement the equation:
- $\dot{v} = -\frac{k}{m}x - \frac{c}{m}\dot{x}$



Mass-Spring-Damper - Steps

- Get experimental data (Extract from figure if numerical data not available)
- Decide which parameters of the model to vary
- Write an evaluation function that returns the error for a given parameter vector
  - Simulate the system for that parameter vector
  - Compare the simulated data with experimental data to calculate error.
- Use one of the optimization functions to find the best parameters & its error.

Comparing with real data



Have a Simulink or Simbiology Model to Optimize?

- No problem.
- Just write an objective function that takes in parameters, runs Simulink/Simbiology model with those parameters and returns the error.

- Demo: massspringdamper_demo

Genetic Algorithms