Model robustness and error metrics

Valeriya Naumova

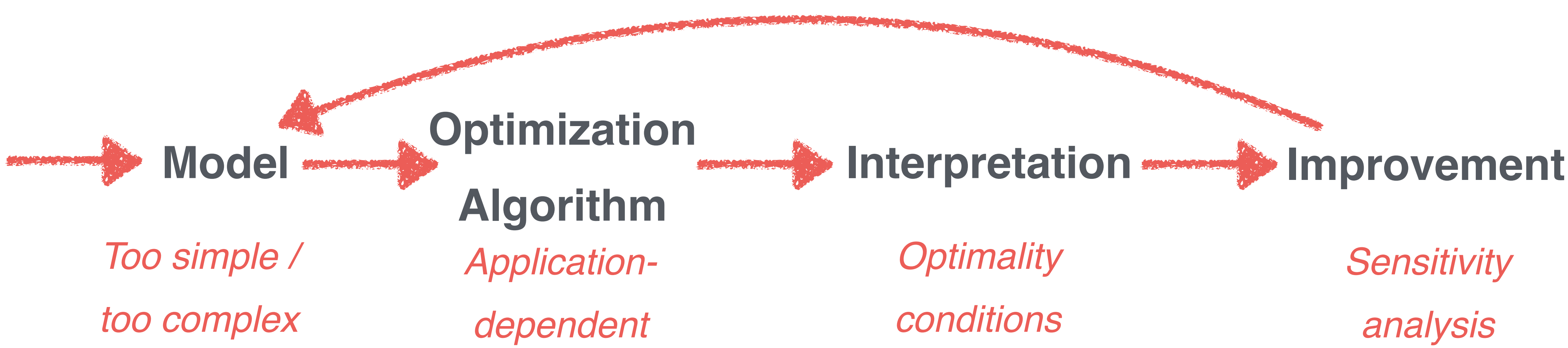
Simula Research Laboratory AS

Simula Summer School 2016

Introduction

“fascinating blend of theory and computation,
heuristics and rigour”
R. Fletcher, 2000

- **Optimization** is an important tool in decision science and in the analysis of physical systems.
- **Key ingredients**:
  - **objective**: quantitative measure of the performance of the system under study (time, potential energy, etc.);
  - **variables** or **unknowns;**
  - **constraints.**



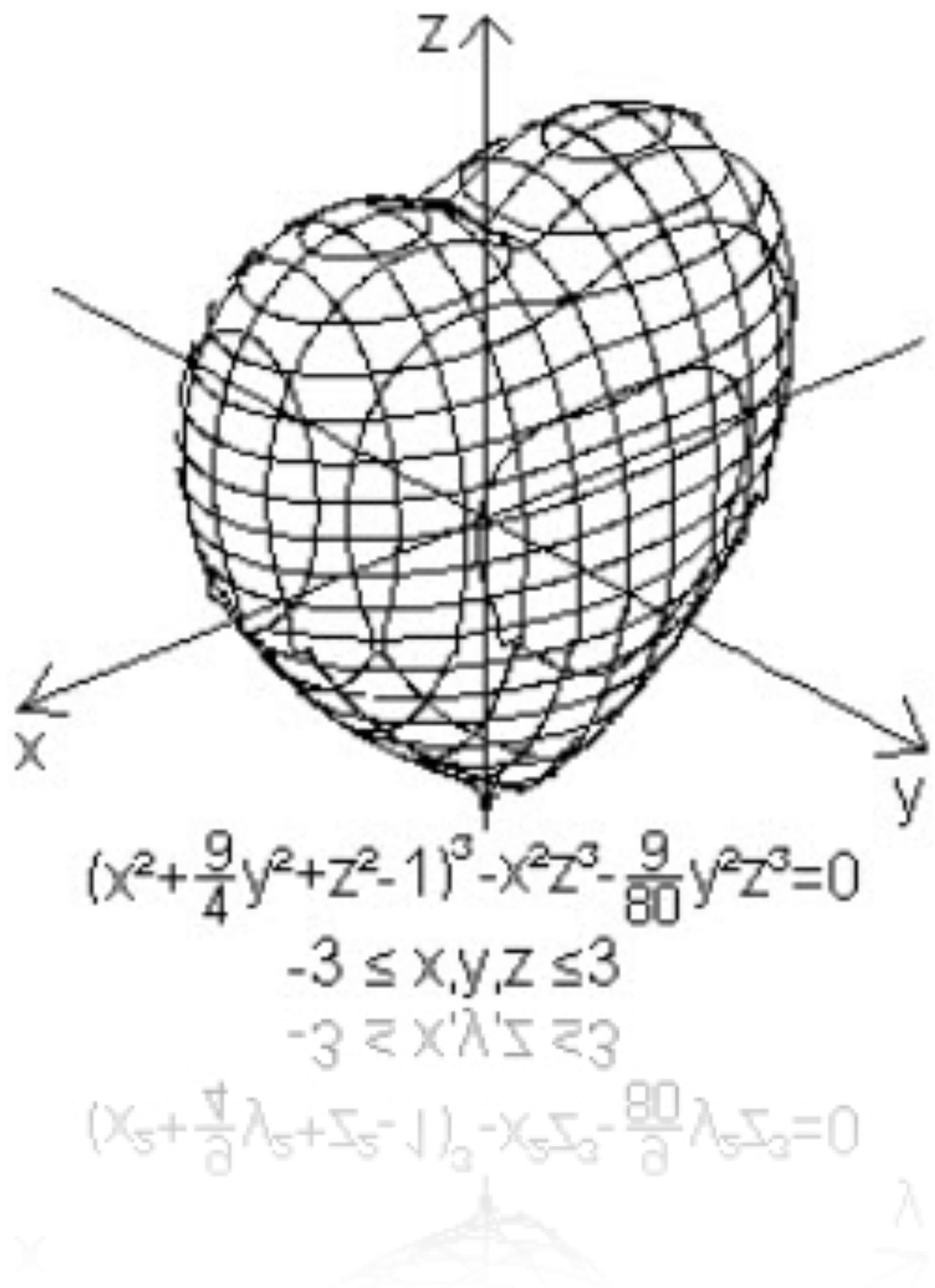
Introduction. Goals

- Understand fundamentals of optimization and various aspects of the process (modeling, optimality conditions, interpretation);
- Gain understanding of the state-of-the-art optimization algorithms; capabilities and limitations;
- Understand basic ideas from uncertainty quantification;
- Overview state-of-the-art techniques for uncertainty quantification.

References

1. J. Nocedal. *Numerical Optimization.* Springer, Second Ed, 2006.
2. R. Fletcher. *Practical Methods of Optimization*. Wiley, 2000.
3. R. Rockafellar. *Convex Analysis*. Princeton University Press, 1972.
4. D. Cacuci. *Sensitivity and Uncertainty Analysis: Theory I.* Chapman & Hall, 2006.
5. Wikipedia and Web.

Outline



Fundamentals of optimization

- Mathematical Formulation
- Examples
- Notation and definition
- Optimization algorithms

Derivative-based methods

- Line search methods
- Trust-region methods
- Conjugate gradient methods

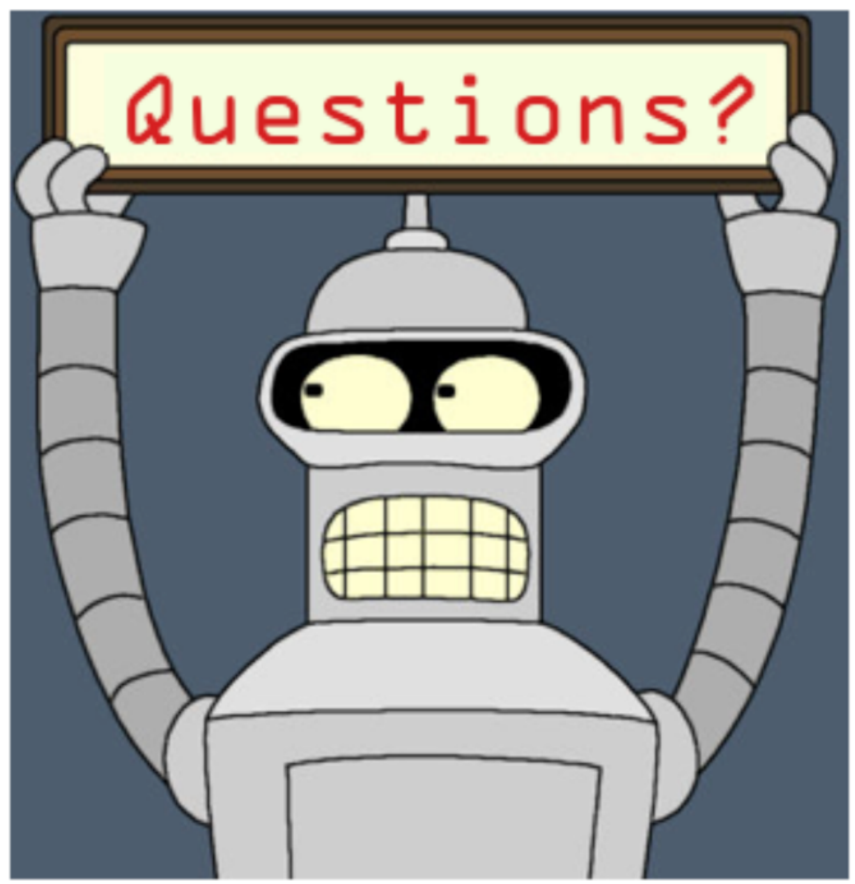
Derivative-free optimization

- Finite differences and noise
- Model-based methods

Uncertainty quantification

- Why uncertainty quantification?
- Definitions
- Computations under uncertainty

The course contains many ideas and (quite) a bit of math, questions help prevent sleeping...



Notation and Definition

Optimization Problem

$$\min_{x \in \mathbb{R}^n} f_0(x) \quad \text{subject to} \quad \begin{array}{l} f_i(x) = 0, i \in \mathcal{E} \\ f_i(x) \geq 0, i \in \mathcal{I} \end{array}$$

$x = (x_1, \ldots, x_n)$: optimization variables;

$f_0 : \mathbb{R}^n \to \mathbb{R}$: objective function;

$f_i : \mathbb{R}^n \to \mathbb{R}, \quad i = 1, \ldots, m$: constraint functions

$\mathcal{I}$ and $\mathcal{E}$ are sets of indices for equality and inequality constraints, respectively

$$\max f = -\min -f$$

Feasible region:

set of points satisfying all constraints

Optimal solution:

x^* has smallest value of f_0 among all vectors that satisfy the constraint

Notation and Definition: Examples

Data Fitting

- **variables**: model parameters
- **constraints**: prior information, parameter limits
- **objective**: measure of misfit or prediction error

Portfolio Optimization

- **variables**: amount invested in different assets
- **constraints**: budget, max/min investment per asset, minimum return
- **objective**: overall risk or return variance

Notation and Definition: Examples

The Transportation Problem

A pharma company has *2 factories F_1 and F_2* and a dozen retail outlets (pharmacies) R_1, R_2,… R_{12}.

Each factory F_i can produce a_i quantity of certain antiarrhythmic pills each week; a_i called the capacity of the plant.

Each retail outlet R_j has a known *weekly demand* of b_j quantity of the product. The cost of shipping of one quantity of the pill from factory F_i to retail outlet R_j is c_{ij}.

Problem: determine how much of the product to ship from each factory to each outlet so as to satisfy all the requirements and minimize costs.

→ **Linear programming problem**
(objective function and the constraints are all linear functions)

Notation and Definition

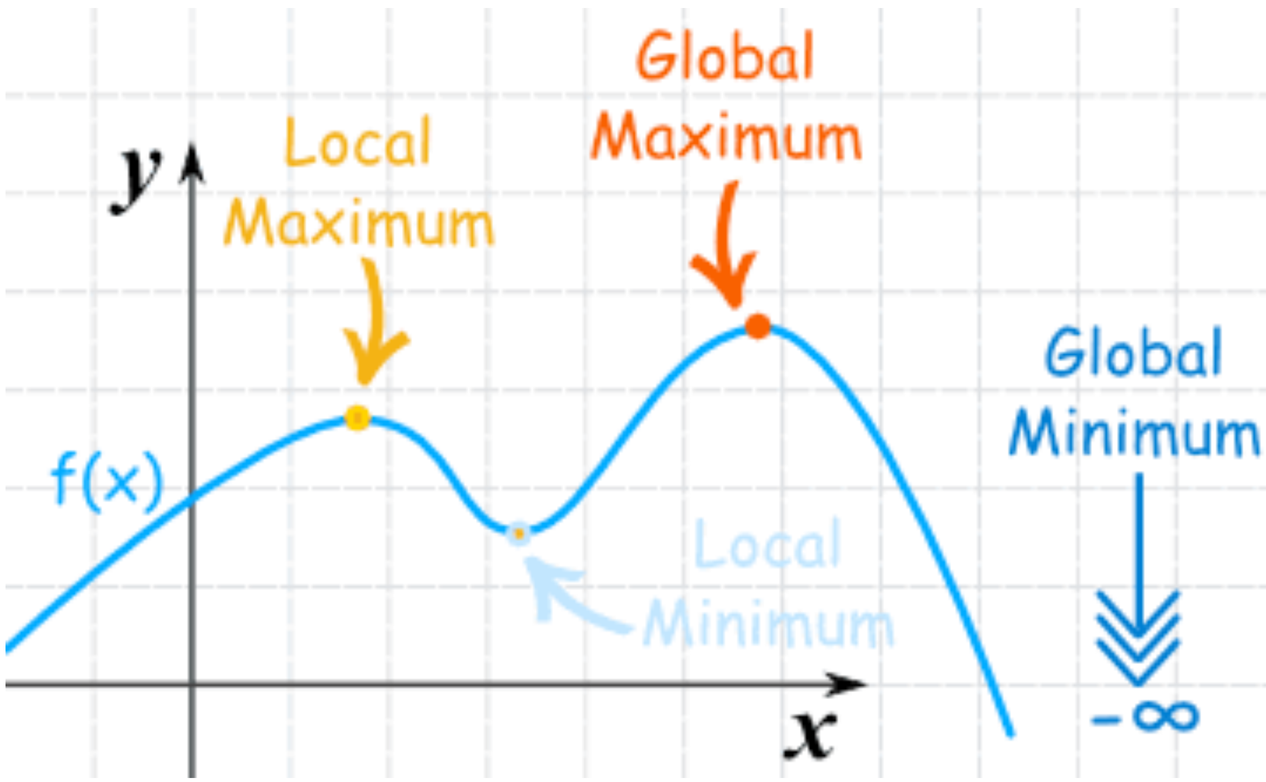
Optimization Types and Main Concepts

- **Continuous versus Discrete Optimization (**integer programming problems**)**
- **Constrained and Unconstrained Optimization**
  - **Unconstrained:** $\mathcal{I} = \mathcal{E} = \emptyset$
  - **Constrained:** Linear programming / nonlinear programming problems
- **Global and Local Optimization**
- **Stochastic and Deterministic Optimization**
  - **Stochastic:** optimize the expected performance of the model
  - **Deterministic:** model is completely known
- **Convexity**
  - Objective function is convex / equality constraints are linear / inequality constraints are concave
- **Optimization Algorithms (iterative)**
  - Robustness / Efficiency / Accuracy

Notation and Definition

Unconstrained Optimization. What is a solution?

$$\min_x f(x)$$



Global minimizer:

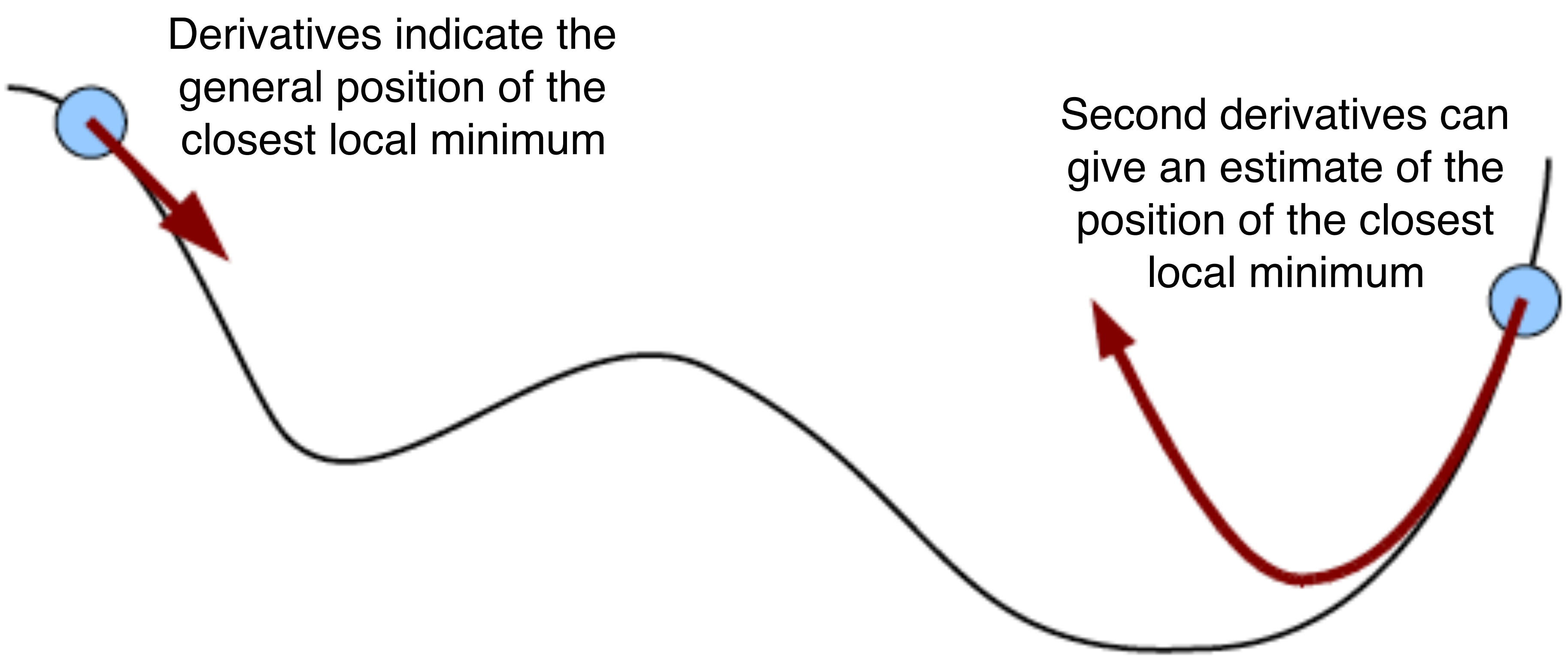
$$f(x^*) \leq f(x) \quad \forall x \in \mathbf{R}^n$$

Local minimizer:

$$f(x^*) \leq f(x^* + \epsilon)$$
$$-\delta \leq \epsilon \leq \delta, \quad \delta > 0$$

Notation and Definition

Differentiability



No such local cues without derivatives

- Derivatives may not exist.
- Derivatives may be too costly to compute.

Notation and Definition. What is a solution?

Recognizing a Local Minimum

Taylor's Theorem

$$f(x^*) = f(x) + \nabla f(x)^T (x - x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x)(x - x^*)$$

First-Order Necessary Conditions

x^* is a local minimizer and f is cont. diff. in an open neighbourhood of x^*

$$\Longrightarrow \nabla f(x^*) = 0.$$

Stationary point: $\nabla f(x^*) = 0.$

Second-Order Necessary Conditions

x^* is a local minimizer, f is twice cont.diff. in an open neighbourhood of x^*

$$\Longrightarrow \nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \text{ is psd}$$

Reminder: B is pd if $p^T B p > 0 \quad \forall p \neq 0$
B is psd if $p^T B p \geq 0 \quad \forall p$

Notation and Definition. What is a solution?

Characterizing a Local Minimum

Second-Order Sufficient Conditions

$\nabla^2 f$ is cont. in an open neighbourhood of x^*

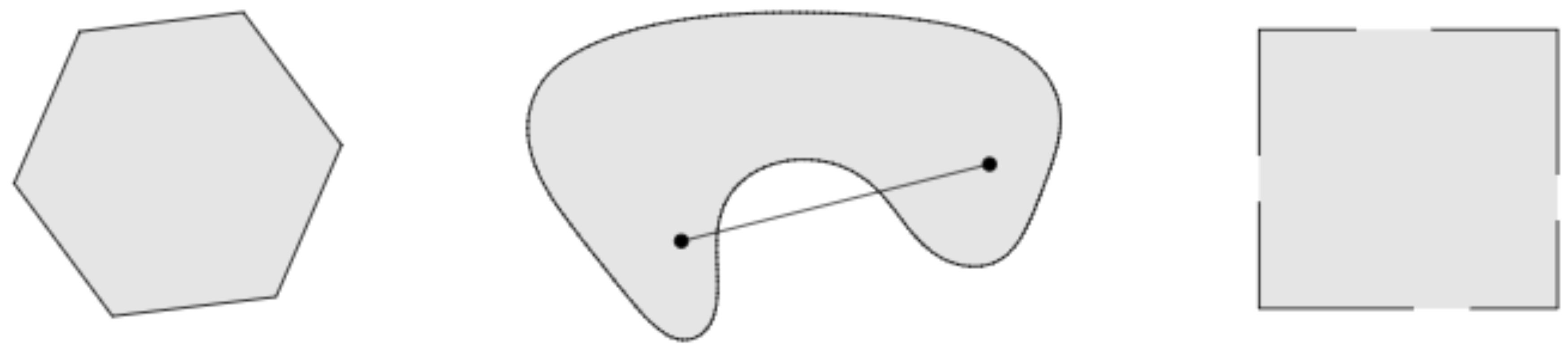
$\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is pd

$\Longrightarrow x^*$ is a strict local minimizer of f

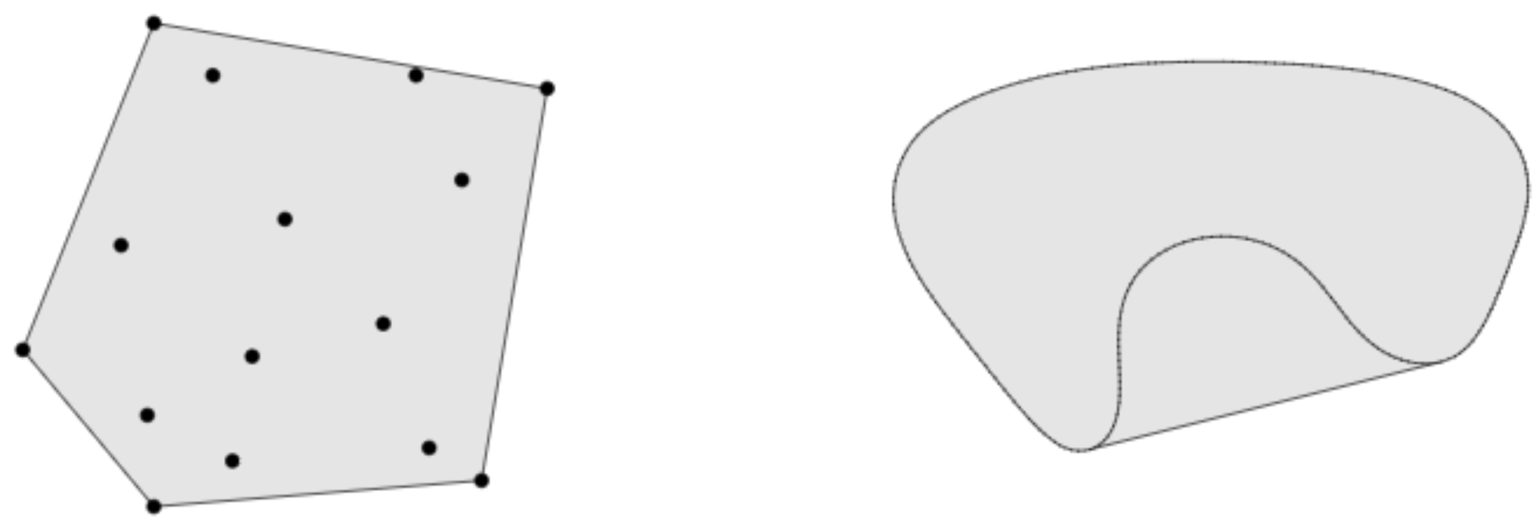
Example: $f(x) = x^4$ has a strict local minimum at $x^* = 0$.

Notation and Definition

Convexity: Sets and Functions



one convex, two nonconvex sets

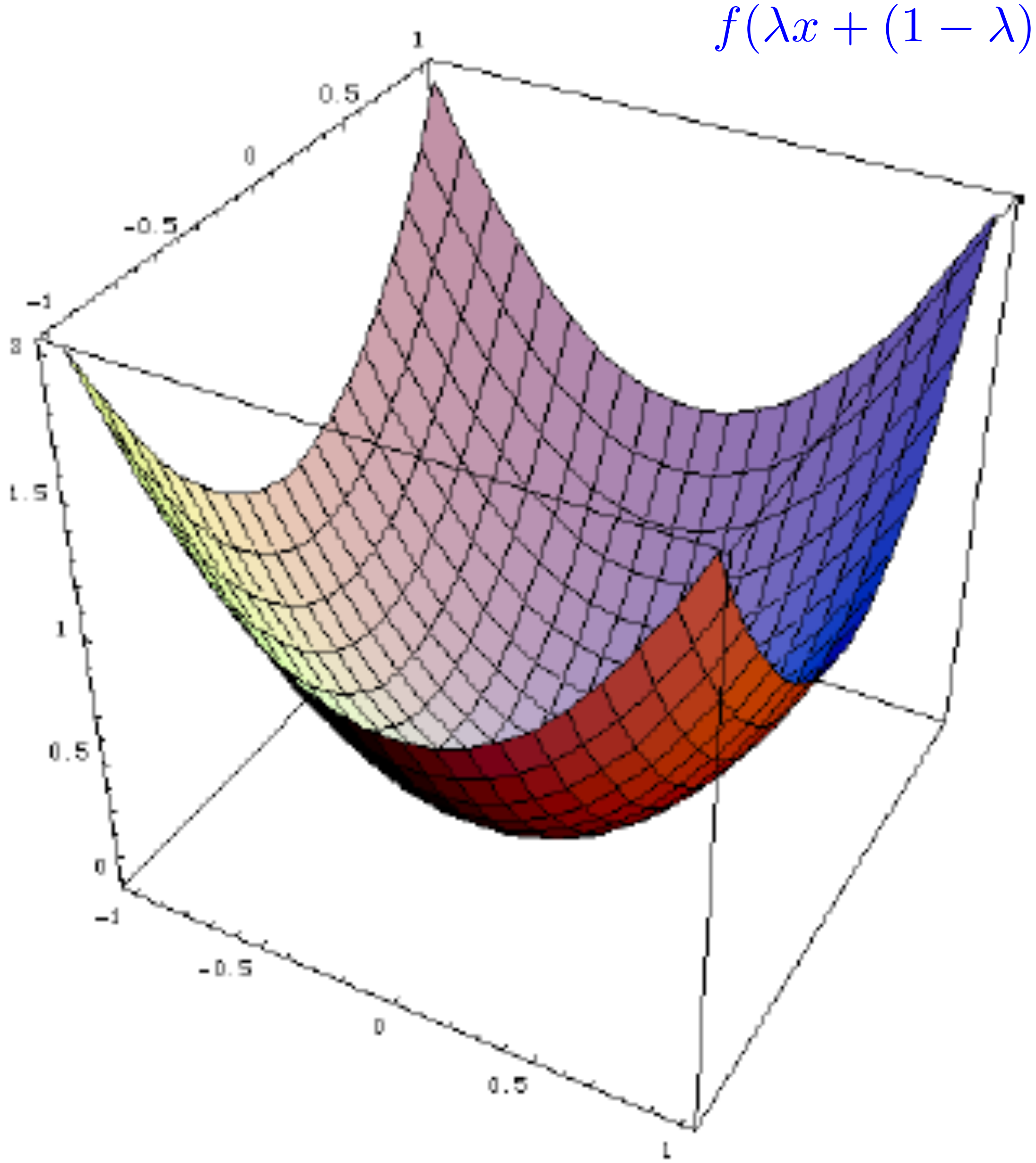


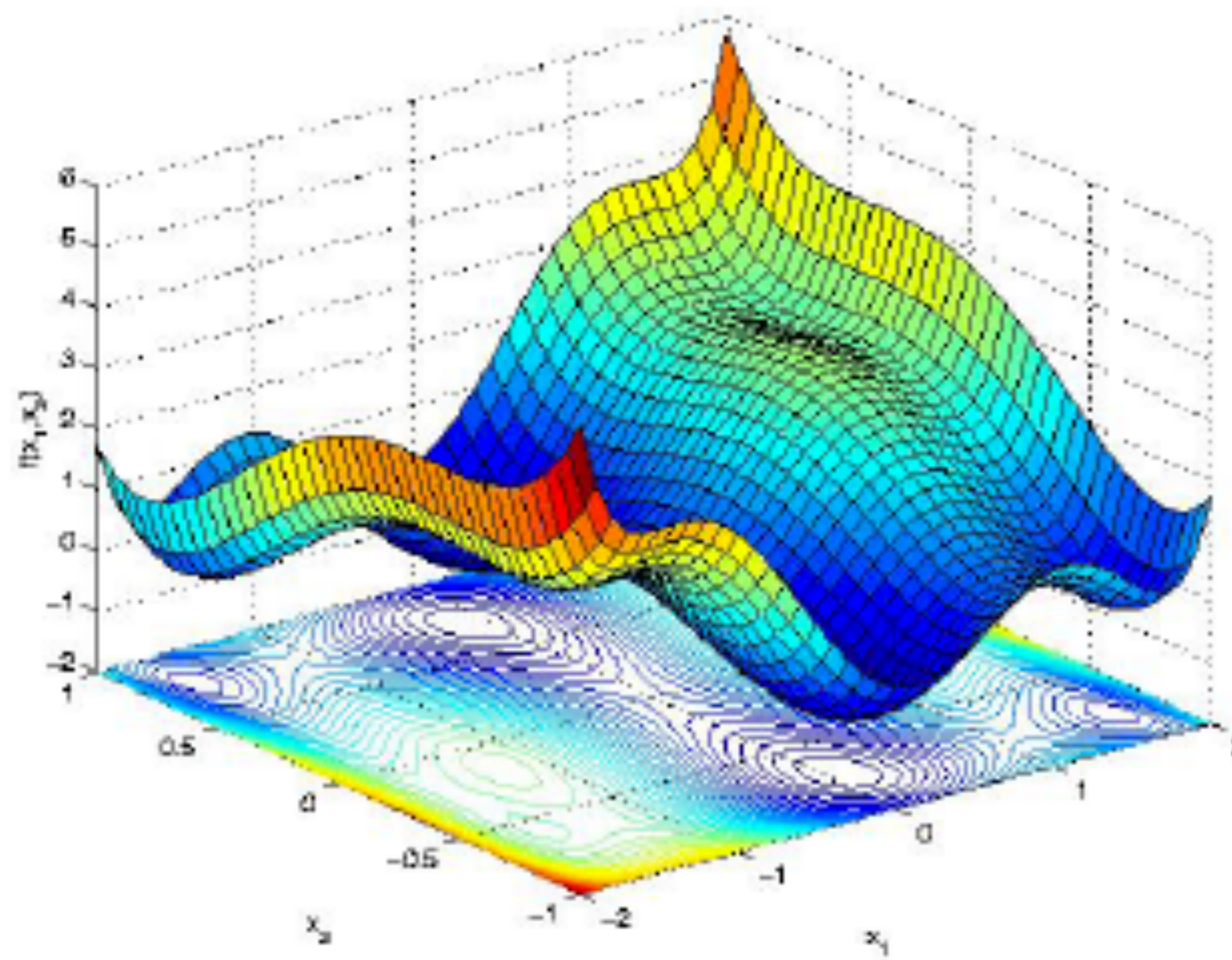
Convex hull: set of all convex combinations of points in the set

Notation and Definition

Convexity: Functions

$$\forall x, y, \quad \forall 0 \leq \lambda \leq 1,$$
$$f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y),$$





Local = Global

Stationary Point $\overset{\text{differentiable}}{=}$ Global

Local minima, saddle points, plateaux, etc

Optimization algorithms are easy to use. They always return the same solution.

Optimization algorithms usually finds local minima. Result depends on subtle details.

Notation and Definition: Examples

The Transportation Problem

A pharma company has *2 factories* F_1 *and* F_2 and a dozen retail outlets (pharmacies) $R_1, R_2, \ldots R_{12}$.

Each factory F_i can produce a_i quantity of certain antiarrhythmic pills each week; a_i called the capacity of the plant.

Each retail outlet R_j has a known *weekly demand* of b_j quantity of the product. The cost of shipping of one quantity of the pill from factory F_i to retail outlet R_j is c_{ij}.

Problem: determine how much of the product to ship from each factory to each outlet so as to satisfy all the requirements and minimize costs.

Have a break and fun in funding a solution!

Notation and Definition: Examples

The Transportation Problem: Solution

The Decision Variables:

The Objective function:

The Constraints:

Formulation:

Notation and Definition: Examples

The Transportation Problem: Solution

$$
\begin{aligned}
\min \quad & \sum_{ij} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{12} x_{ij} \le a_i, \quad i = 1, 2 \\
& \sum_{i=1}^{2} x_{ij} \ge b_j, \quad j = 1, \ldots, 12 \\
& x_{ij} \ge 0, \quad i = 1, 2, \quad j = 1, \ldots, 12
\end{aligned}
$$

Derivative-based algorithms

Overview

- Variety of a powerful collection of algorithms for unconstrained optimization of smooth functions.
- **Ingredients:**
  - **Input**: Starting point x_0
  - Algorithms generates a sequence of iterates $\{x_k\}_{k=0}^{\infty}$
  - **Update**: $x_k \rightarrow x_{k+1}$
    - *Line Search:*

      choose a direction p_k and search along this direction from the current iterate x_k for a new iterate with a lower function value.

      $$\min_\alpha f(x_k + \alpha p_k)$$

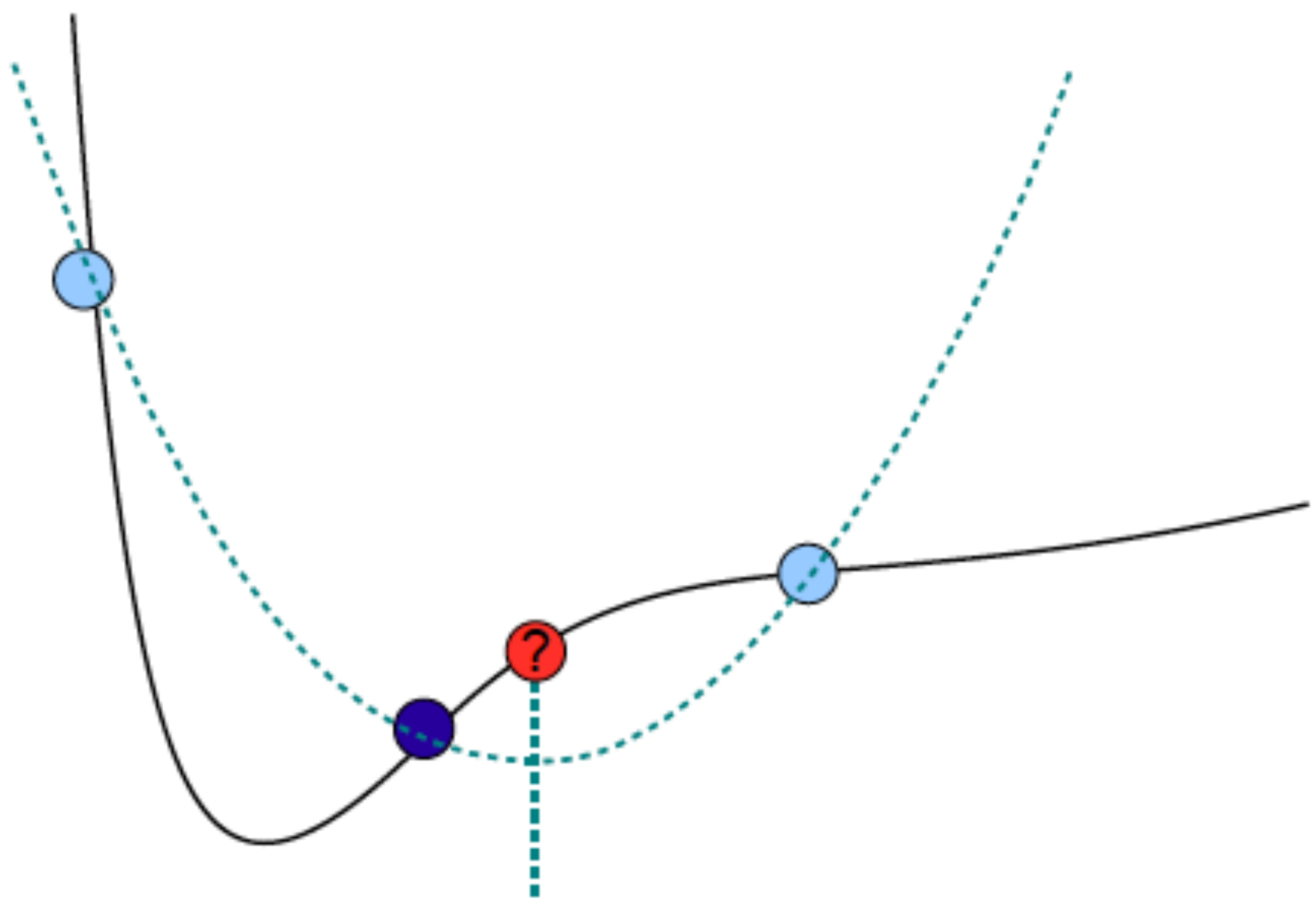
    - *Trust Region*:

      using information about f, construct a model function m_k whose behavior near the current point x_k is similar to that of the actual objective function f.

      $$\min_p m_k(x_k + p_k), \text{ where } x_k + p_k \text{ inside the trust region}$$

  - **Termination**: no more progress / accuracy is reached

Line Search



– Fitting a parabola sometimes gives much better guess.

$$x_{k+1} = x_k + \alpha p_k$$

Line Search

Brent Algorithm for Line Search

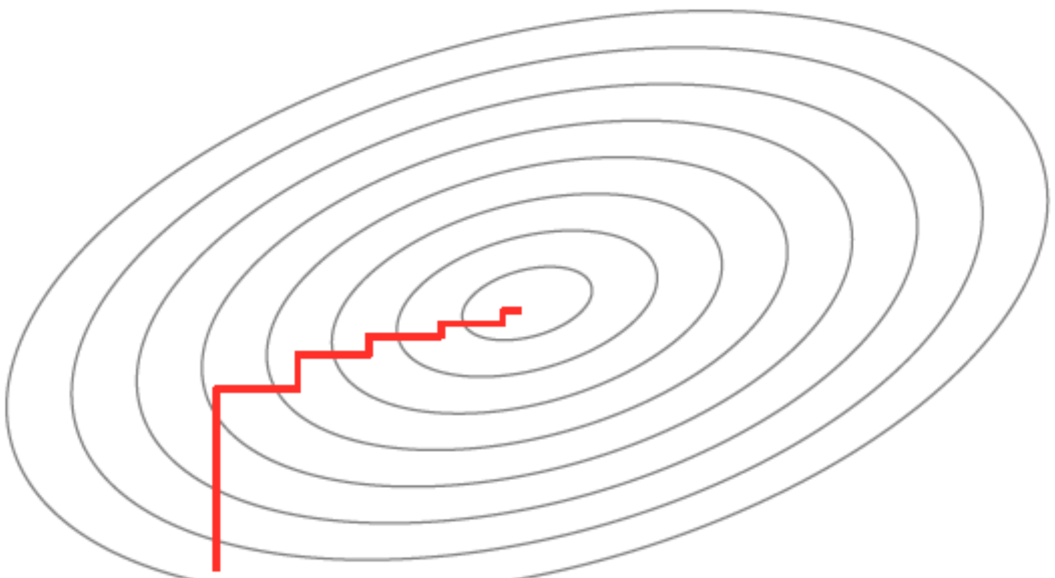
- Alternate **golden section** and **parabolic interpolation.**
- No more than twice slower than golden section.
- No more than twice slower than parabolic section.
- In practice, almost as good as the **best of the two.**

Variants with derivatives

- Improvements if we can compute function value and its first derivative together
- Improvements if we can compute function value and its first / second derivatives together

Line Search

Coordinate Descent (Derivative-Free) Method

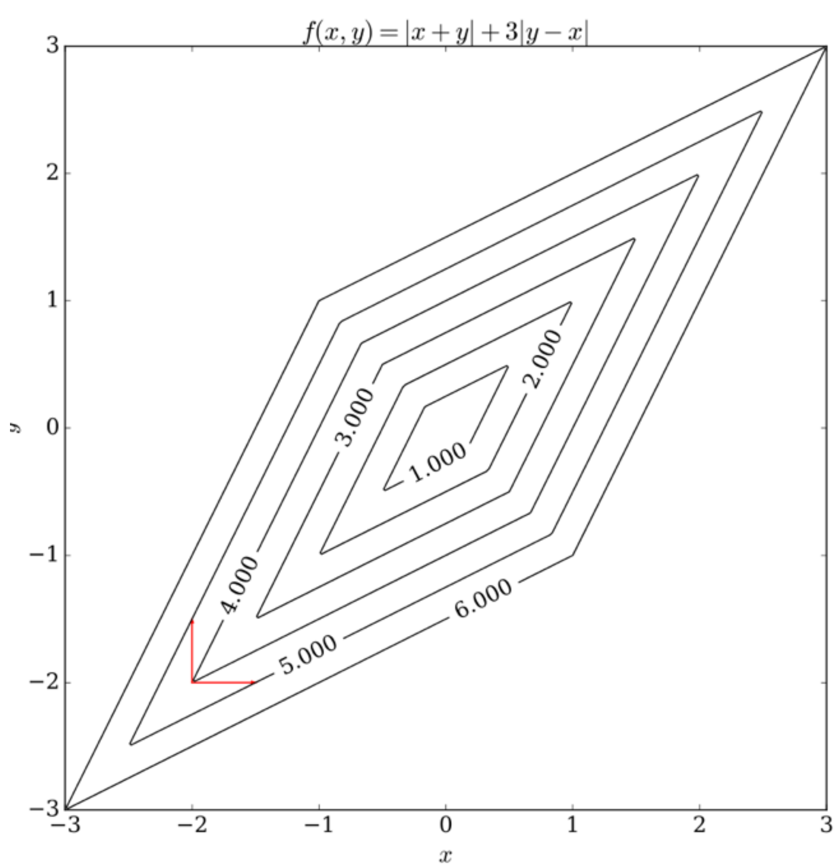


Perform successive line searches along the axes.

Tends to zig-zag

Pros:

- Simple;
- Competitive to other methods (esp. in machine learning);
- No derivative needed.



Limitations:

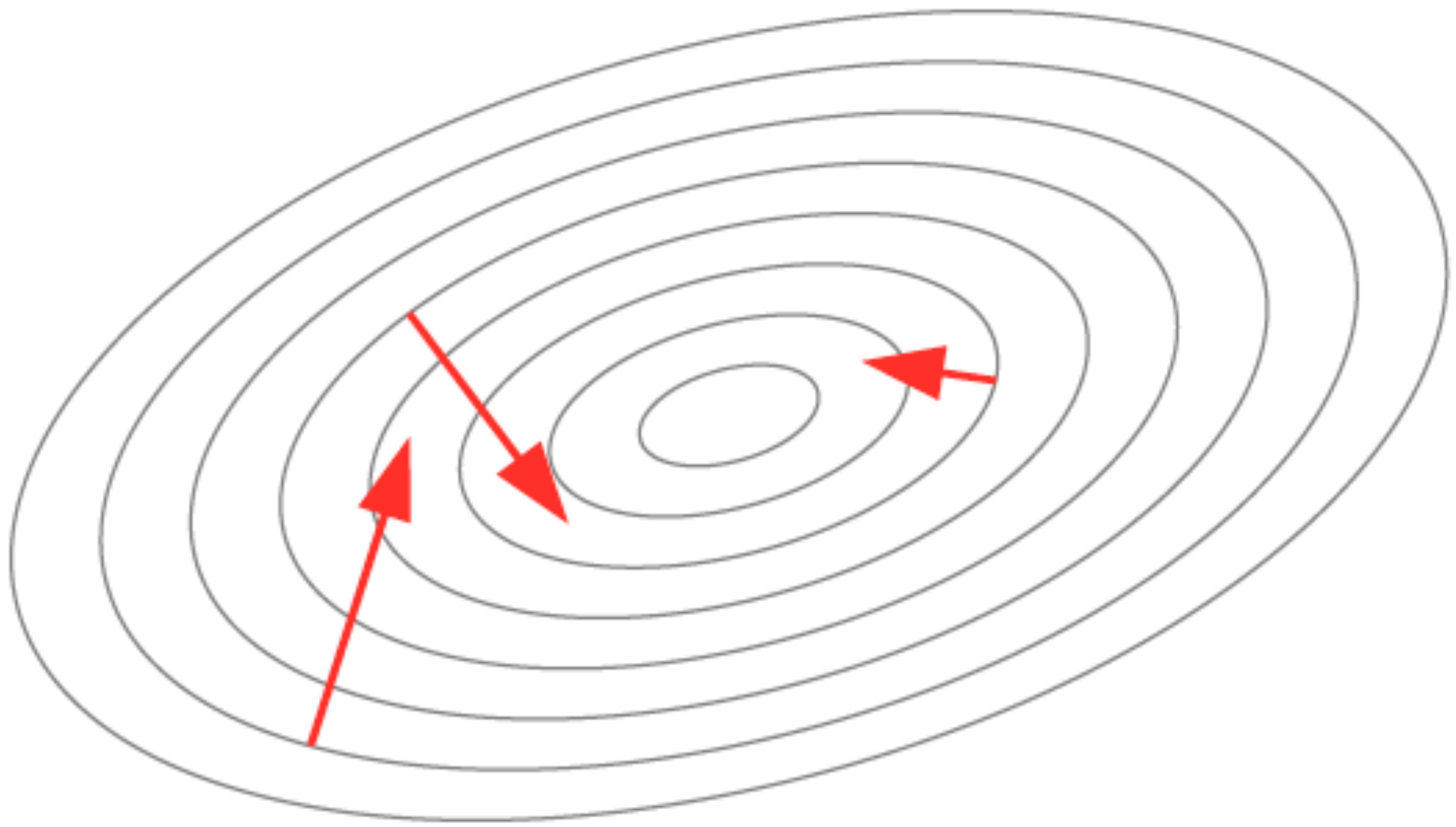
- Does not work for non-smooth functions.

May get stuck at a non-stationary point

If a function is cont. differentiable,

Line Search

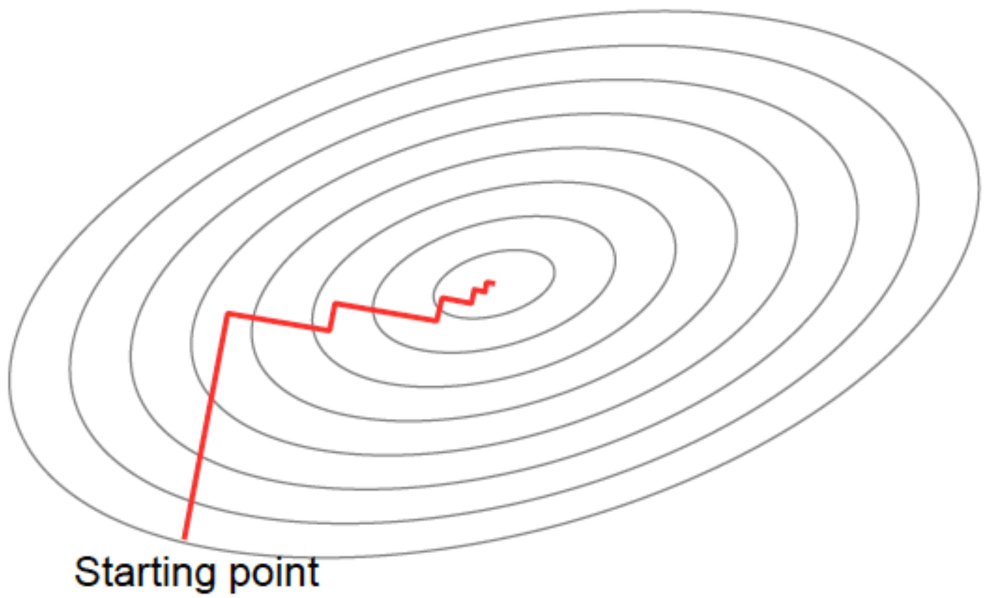
Gradient



The gradient $\frac{\partial f}{\partial \omega} = \left(\frac{\partial f}{\partial \omega_1}, \ldots, \frac{\partial f}{\partial \omega_d} \right)$ gives the steepest descent direction.

Line Search

Steepest Descent Method



Perform successive line searches along the gradient direction (*see Taylor's theorem*).

$$p_k = -\nabla f_k$$

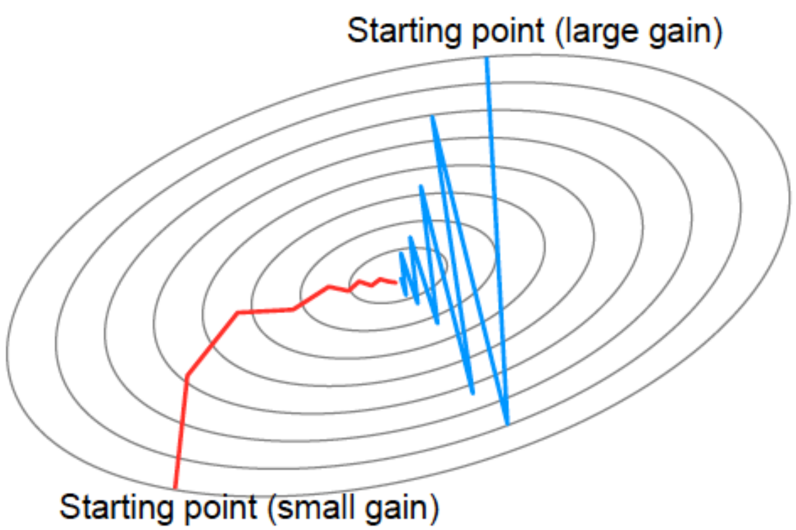
Pros:

- Calculation of the gradient only;
- Beneficial if computing the gradients is cheap enough.

Limitations:

- Can be expensive;
- Slow on difficult problems.

Gradient + line search



$$x_{k+1} = x_k - \gamma \nabla f(x)$$

Line Search

Hessian Matrix

$$H(\omega) = \begin{pmatrix} \frac{\partial^2 f}{\partial \omega_1 \partial \omega_1} & \cdots & \frac{\partial^2 f}{\partial \omega_1 \partial \omega_d} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial \omega_d \partial \omega_1} & \cdots & \frac{\partial^2 f}{\partial \omega_d \partial \omega_d} \end{pmatrix}$$

Newton Direction

Taylor expansion:

$$f(x_k + p) \approx f(x_k) + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p = m_k(p)$$

$$\text{Assuming } \nabla^2 f_k \text{ is pd, then } p_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k$$

Line Search

Newton Method

$$x_{k+1} = x_k - (\nabla^2 f_k)^{-1} \nabla f_k$$

- Can be used in a line search method when $\nabla^2 f_k$ is **pd**

  (guarantee that Newton direction is a descent direction)
- Beware when Hessian is not positive definite!

  (the Newton direction may not even be defined)
- Very few iterations needed when Hessian is positive definite! Quadratic convergence!

Limitations:

- Computation and storage of $\nabla^2 f_k$ can be quiet costly.

Quasi-Newton Method

- Methods that avoid the drawbacks of Newton, i.e., do not **require computation** of the Hessian but use an approximation instead.
- But behave like Newton during the final convergence.

$$x_{k+1} = x_k - B_k^{-1} \nabla f_k$$

Line Search

(Nonlinear) Conjugate Gradient Method

$$x_{k+1} = x_k - \nabla f_k + \beta_k p_{k-1}$$

$\beta_k \in \mathbb{R}$ ensures that p_k and p_{k-1} are *conjugate*.

- Methods were originally designed to solve systems of linear equations.

$$Ax = b \quad \Longleftrightarrow \quad \min \phi(x) = \frac{1}{2} x^T A x - b^T x.$$

A is an $n \times n$ symmetric pd matrix.

Pros:

- More effective than the steepest descent direction.
- No matrix storage. Almost as simple as the steepest descent methods.
- Adapted to solve nonlinear optimization problems.

Limitations:

- Do not attain fast convergence rates as Newton or Quasi-Newton.
- Sensitive to roundoff errors.

Reminder: two non-zero vectors u and v are conjugate (wrt A) if $u^T A v = 0$.

Line Search

Summary for Derivative-based Optimization

- **Line Search:** fix direction, choose distance.

  For most algorithms $p_k = -B_k^{-1}\nabla f_k$, where B_k is symmetric nonsingular

  steepest descent: $B_k = I$;

  Newton's method: $B_k = \nabla^2 f(x_k)$;

  Quasi-Newton method: $B_k \approx \nabla^2 f(x_k)$.

- **Trust Region:** find maximum distance, choose direction and actual distance.

$$x_{k+1} = x_k + p_k$$

p_k is approx. minimizer of model m_k of f in trust region.

p_k does not produce sufficient decrease $\to$ region too big, think it and try again.

Step lengths? Convergence? Rate of Convergence?

Derivative-free optimization

Overview. Why?

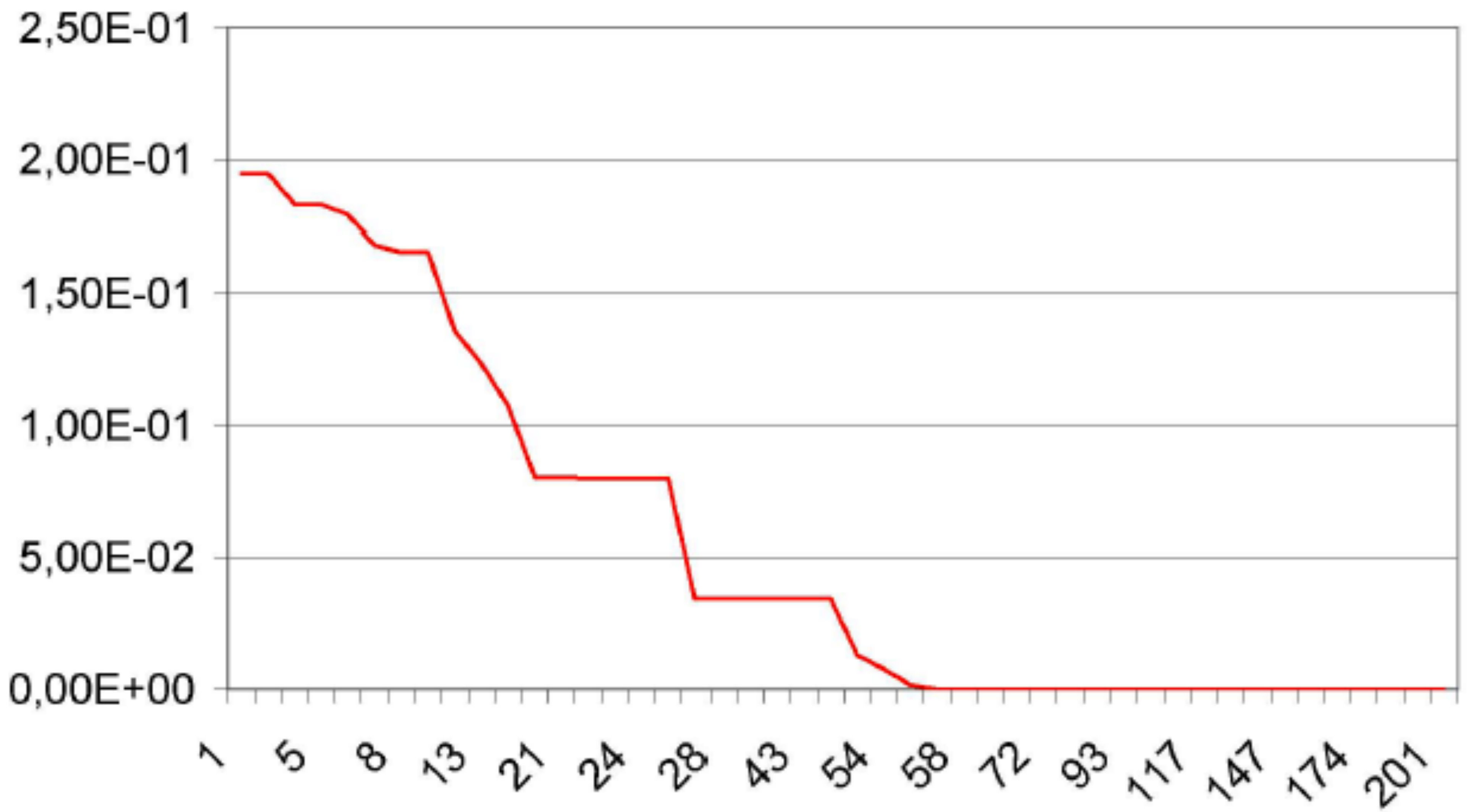
- Evaluating ∇f in practice is sometimes impossible:
  - the function can be the results of a simulation / experiment;
  - coding of gradient may be time-consuming or impractical.
- **Approaches:**
  - approximate gradient and possibly Hessian using finite differences, then apply derivative-based method. **BUT**:
    - Number of function evaluations may be excessive.
    - Unreliable with noise.
  - use function values at a set of sample points, instead of the gradient approximation, and determine a new iterate by a different means. **BUT**:
    - less developed / less efficient.
    - effective only for small problems.
    - difficult to use with general methods.

Recommendation: derivative-based > finite-difference based > derivative-free

Derivative-free optimization

Limitations

In DFO convergence / stopping is typically slow (per function evaluation)



Newton Method:
Quadratic convergence
First + Second order derivatives

Quasi-Newton Method:
Superlinear convergence
First order derivatives

Derivative-Free Optimization

Overview

- Derivative-free optimization (DFO) algorithms sample function values to determine the new iterate.
  - *Model-based methods:* build a linear or quadratic model of f by interpolating f at a set of samples and use it with trust-region. Slow convergence rate and very costly steps.
  - *Coordinate descent*: minimize successively along each variable. If it does converge, its rate of convergence is often much slower than that of steepest descent. Very simple and convenient sometimes.
  - *Pattern-search methods:* the iterate carries a set of directions that is possibly updated based on the values of f along them. Generalizes coordinate descent to a richer direction set.
  - *Conjugate directions* built using the parallel subspace property: computing the new conjugate direction requires n line searches (CG requires only one).
  - *Finite-difference approximations* to the gradient degrade significantly with noise in f.

Uncertainty Quantification (UQ)

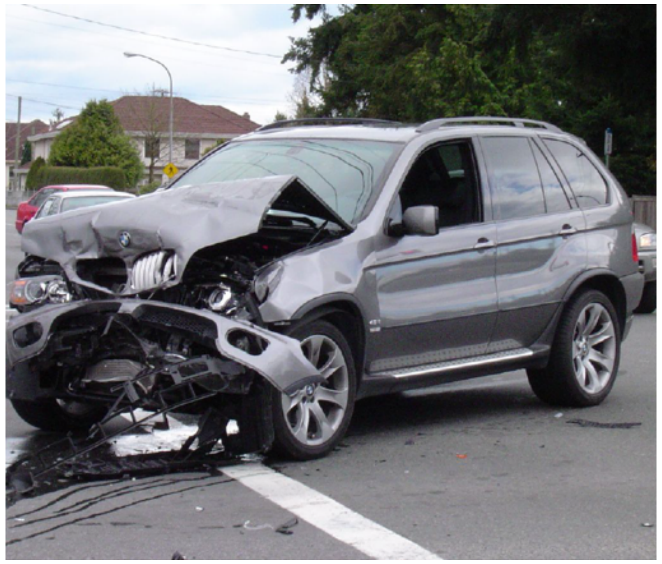
Definition from Wikipedia

Uncertainty quantification (UQ) is the science of quantitative characterization and reduction of uncertainties in applications. It tries to determine how likely certain outcomes are if some aspects of the system are not exactly known.

An example would be to predict the acceleration of a human body in a head-on crash with another car: even if we exactly knew the speed, small differences in the manufacturing of individual cars, how tightly every bolt has been tightened, etc, will lead to different results that can only be predicted in a statistical sense. [...]

Uncertainty Quantification

Why UQ? Decision Making



UQ is critical in identifying the confidence in an outcome.

Provides basis for certification in high-consequence decisions.

UQ is a fundamental component of model validation.

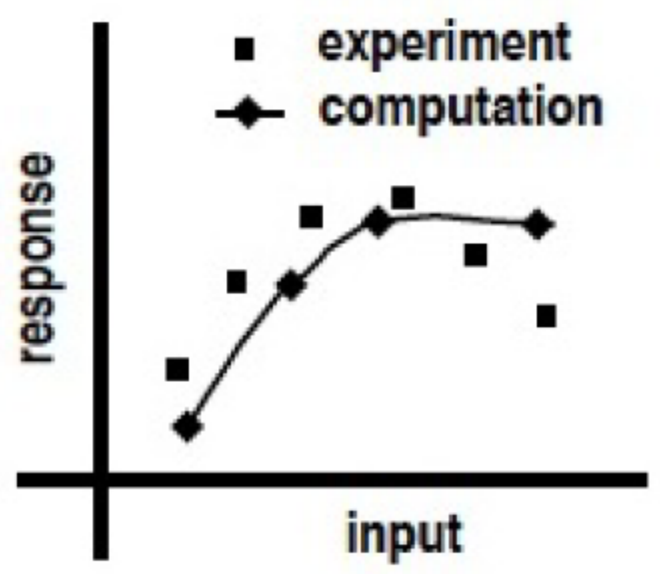
Required to identify the effect limited knowledge in inputs of the simulations

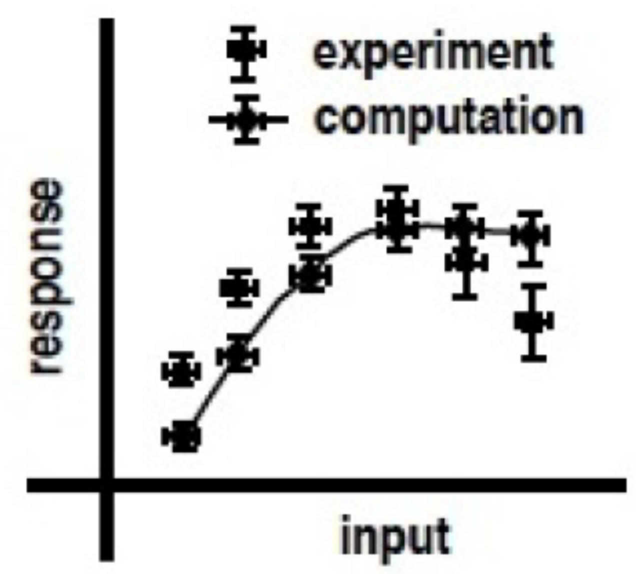
Uncertainty Quantification

Why UQ?

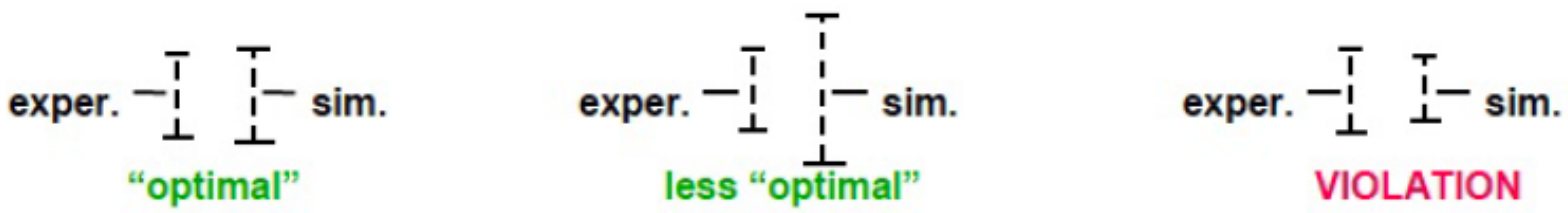
In spite of the wide spread use of simulations it remains difficult to provide objective confidence levels

One of the objective of UQ is to add error bars





… But also the precise notion of validated model



Definitions and notations

"As we know there are known knowns.There are things we know we know. We also know there are known unknowns.That is to say, we know there are some things we do not know. But there are also unknown unknowns,The ones we don't know we don't know."

D. Rumsfeld, 2002, Department of Defense news briefing

The American Institute for Aeronautics and Astronautics (AIAA) has developed the *"Guide for the Verification and Validation (V&V) of Computational Fluid Dynamics Simulations"* (1998)

- **Verification**: The process of determining that a model implementation accurately represents the developer's conceptual description of the model.

  "are we solving the equations correctly?" – it is an exercise in mathematics

- **Validation**: The process of determining the degree to which a model is an accurate representation of the real world for the intended uses of the model.

  "are we solving the correct equations?" – it is an exercise in physics

Definitions and notations

According to the *"Guide for the Verification and Validation (V&V) of Computational Fluid Dynamics Simulations"* (1998)

- **Errors** as recognisable deficiencies of the models or the algorithms employed
- **Uncertainties**: as a potential deficiency that is due to lack of knowledge.

  - ***The definitions are not very precise***
  - ***Do not clearly distinguish between the mathematics and the physics.***
  - ***What is the relation to V&V?***

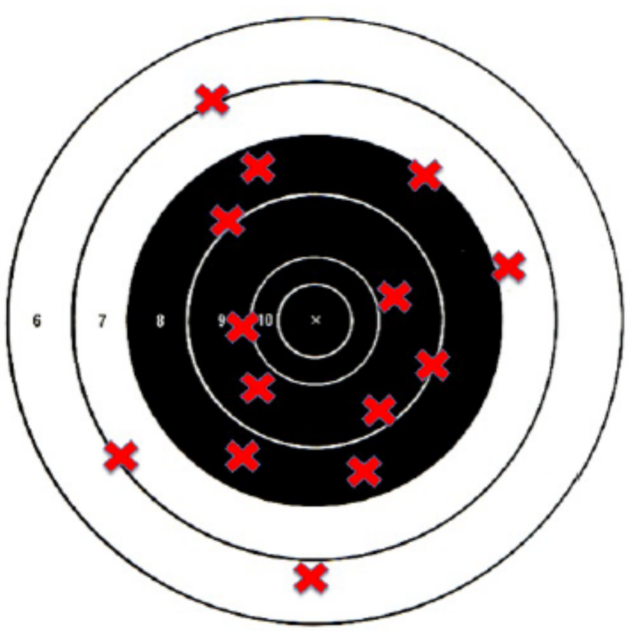
Definitions and notations

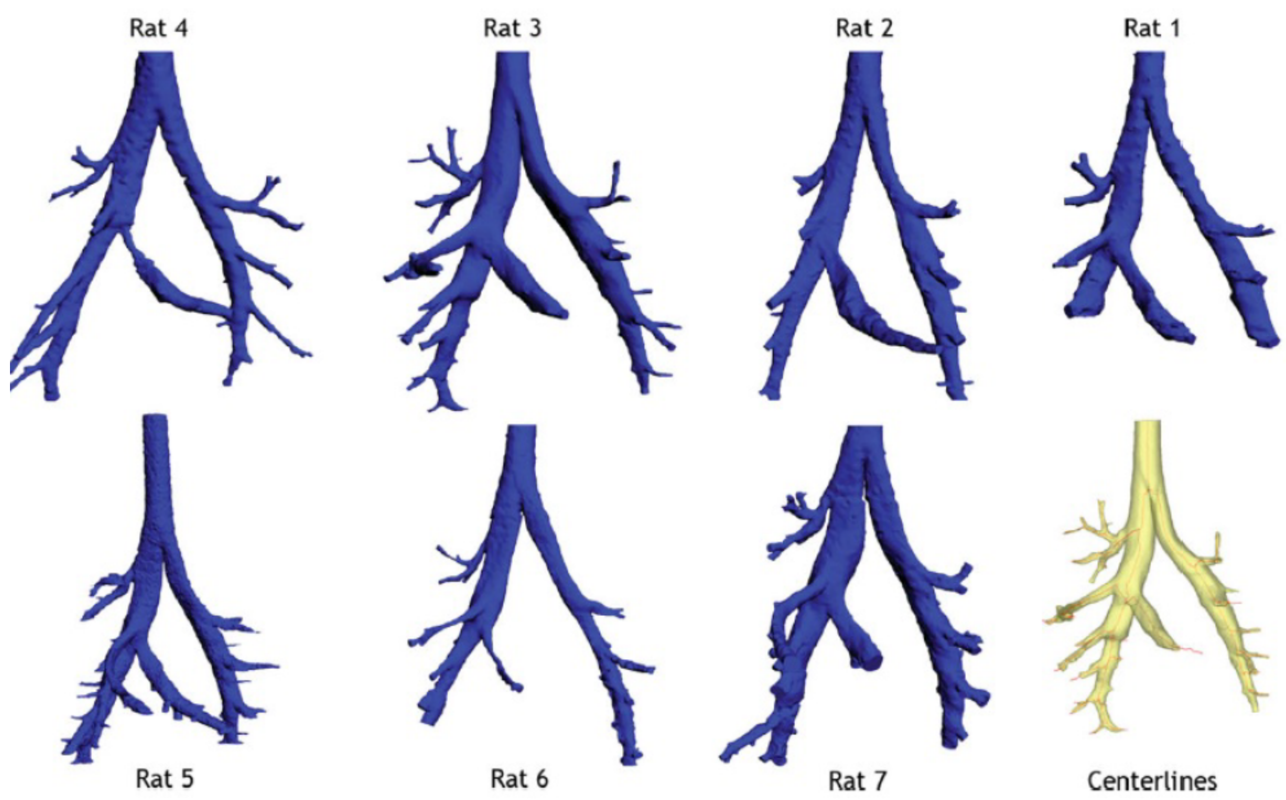
- **What are errors?** errors are associated to the translation of a mathematical formulation into a numerical algorithm and a computational code.
  - roundoff, limited convergence of iterative algorithms;
  - implementation mistakes (bugs);
  - is the mathematics…
- **What are uncertainties**? uncertainties are associated to the specification of the input physical parameters required for performing the analysis.
  - is the physics….

Definitions and notations

Uncertainties

- **Aleatory:** it is the physical variability present in the system or its environment.
  - It is not strictly due to a lack of knowledge and cannot be reduced (also referred to as variability, stochastic uncertainty or **irreducible uncertainty**);
  - **It is naturally defined in a probabilistic framework;**
  - Examples are: material properties, operating conditions manufacturing tolerances, etc.
  - In mathematical modelling it is also studied as **noise**.

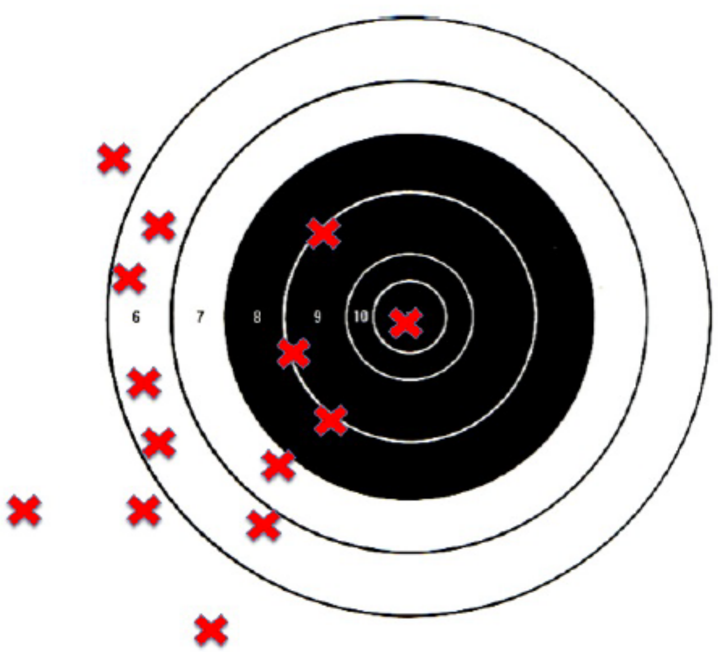


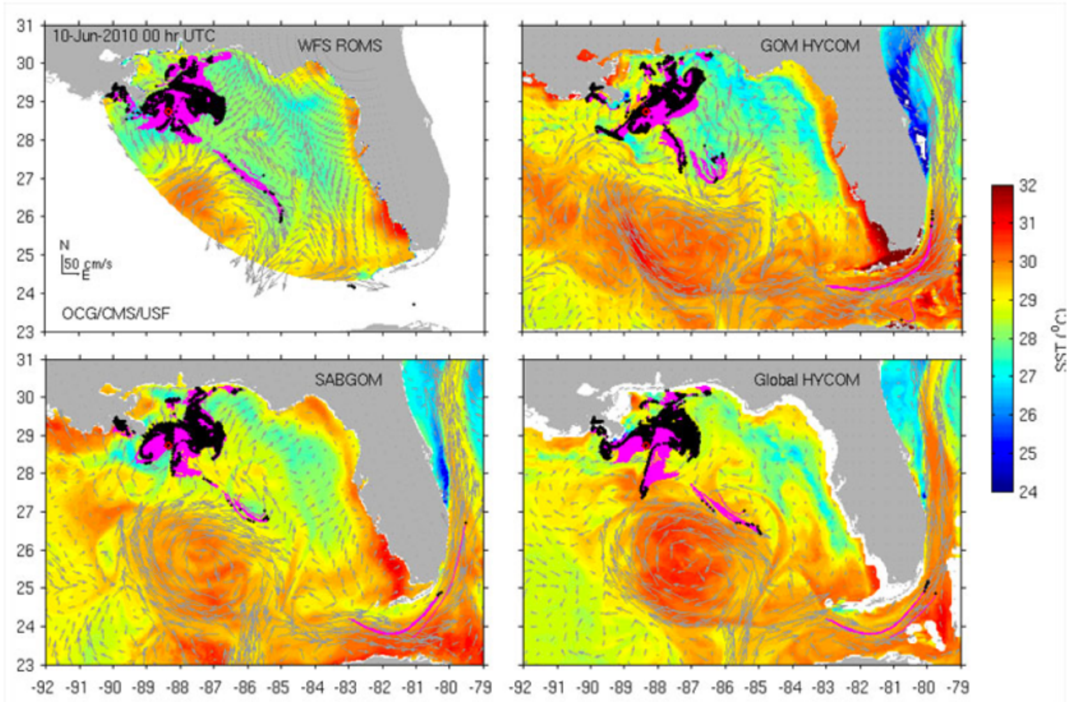


Definitions and notations

Uncertainties

- **Epistemic:** it is a potential deficiency that is due to a lack of knowledge
  - It can arise from assumptions introduced in the derivation of the mathematical model (it is also called **reducible uncertainty** or incertitude);
  - Examples are: turbulence model assumptions or surrogate chemical models;
  - **It is NOT naturally defined in a probabilistic framework;**
  - Can lead to strong **bias** of the predictions.





Deep water horizon oil tracking forecast

Definitions and notations

Summary: Not all uncertainties created equal..

- **Uncertainties relate to the physics of the problem** of interest! not to the errors in the mathematical description/solution…

- Reducible vs. Irreducible Uncertainty
  - **Epistemic uncertainty can be reduced** by increasing our knowledge, e.g. performing more experimental investigations and/or developing new physical models.
  - **Aleatory uncertainty cannot be reduced** as it arises naturally from observations of the system. Additional experiments can only be used to better characterize the variability.

- **Sensitivity vs Uncertainty Analysis**
  - **Sensitivity analysis** investigates the connection between inputs and outputs of a (computational) model
  - **Uncertainty analysis** aims at identifying the overall output uncertainty in a given system.

Computations under Uncertainty
= Predictive Simulations

"The significant problems we face cannot be solved at the same level of thinking we were at when we created them."

A. Einstein

Consider a generic computational model in high dimensions

How do we handle the uncertainties?

1. **Uncertainty definition:** characterize uncertainties in the inputs
2. **Uncertainty propagation:** perform simulations accounting for the identified uncertainties
3. **Certification**: establish acceptance criteria for predictions

Uncertainty Definition

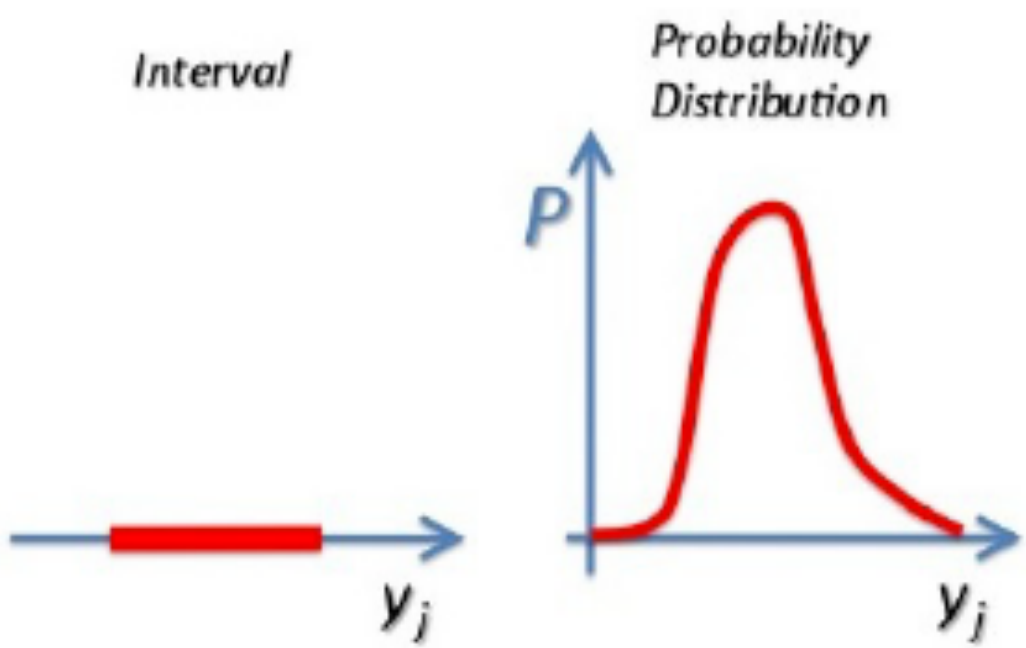
The objective is characterize uncertainties in simulation inputs, based on **available information**

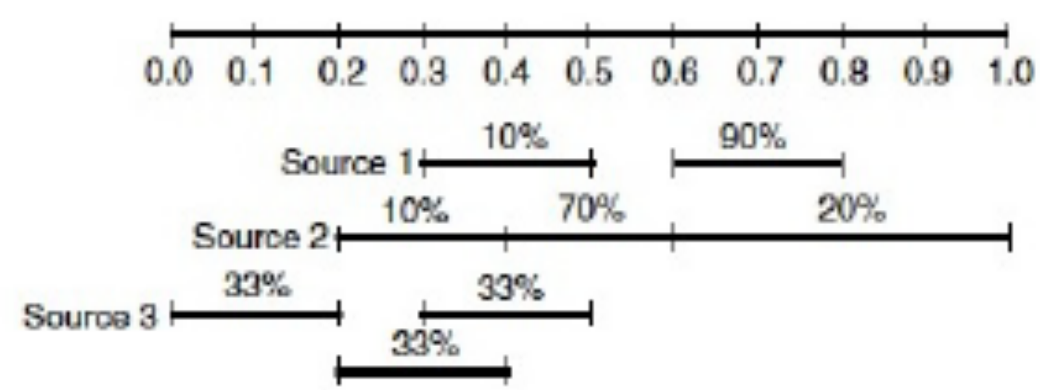
- **Direct methods:** Experimental observations / Theoretical arguments / Expert opinions, etc.
- **Inverse methods (Inference, Calibration):** determination of the statistical input parameters that represent observed data using a computational model

Uncertainty Definition

Identification of all the (d) explicit and hidden parameters of the mathematical/ computational model **y**

Characterization of the associated level of knowledge





The mathematical framework for propagating uncertainties is dependent on the data representation chosen

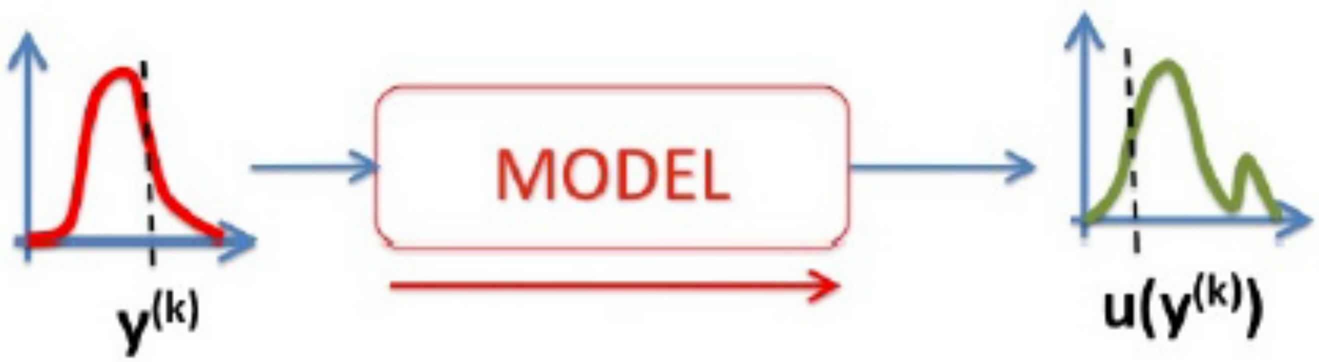
Uncertainty Propagation

Perform simulations accounting for the uncertainty represented as **randomness**

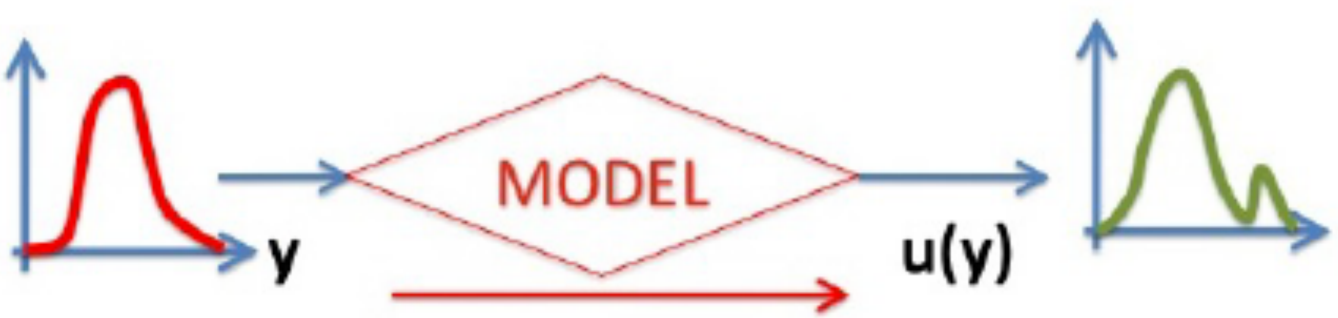
- Define an abstract probability space $(\Omega, \mathcal{A}, \mathcal{P})$
- Introduce uncertain input as **random quantities** $y(\omega), \quad \omega \in \Omega$
- The original problem becomes **stochastic** with solution $u(\omega) = u(y(\omega))$

Uncertainty Propagation

Nonintrusive methods only require (multiple) solutions of the **original** (deterministic) model



Intrusive methods require the formulation and solution of a **stochastic** version of the original problem



Uncertainty Propagation

Nonintrusive methods only require (multiple) solutions of the **original** (deterministic) model

+ Simple extension of the "conventional" simulation paradigm

+ Embarrassingly parallel: solutions are independent

+ Conceptually very simple

Intrusive methods require the formulation and solution of a **stochastic** version of the original problem

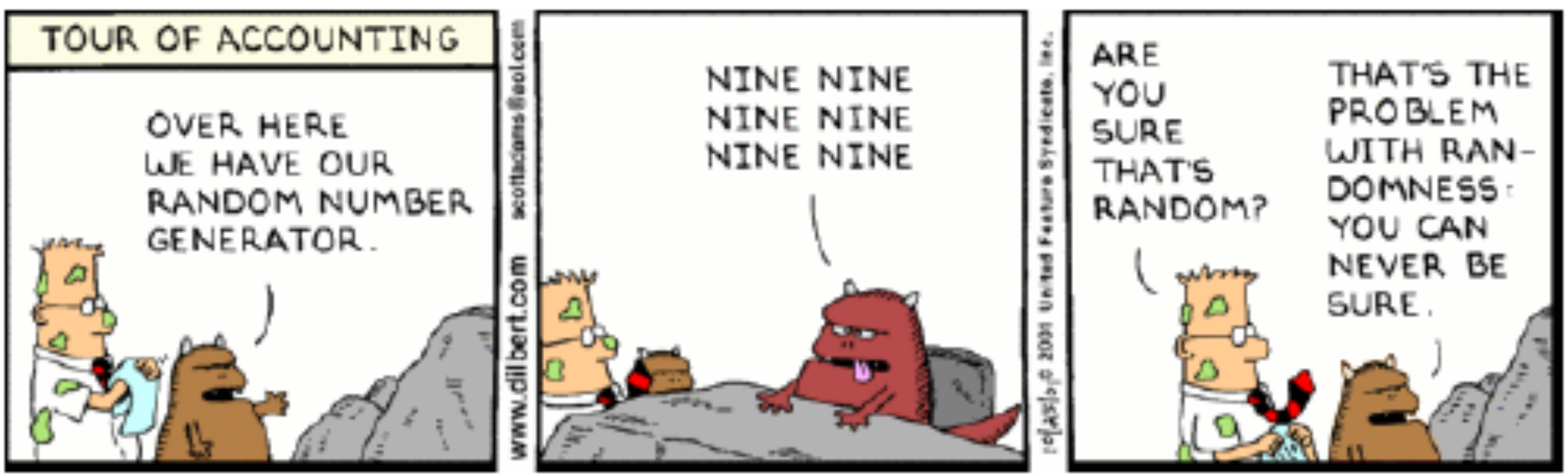
+ Exploit the mathematical structure of the problem

+ Leverage theoretical & algorithmic advancements

(Probabilistic) Uncertainty Propagation

Uncertainty = Randomness

- Sampling Methods: Monte Carlo, Quasi Monte Carlo, Lati Hypercube, etc.
- Intrusive Methods: Polynomial Chaos, Adjoints, etc.
- Non-Intrusive Methods: Stochastic Collocation, Response Surface, etc.
- Optimization Methods



Certification

Certification and Validation