TC2: Optimization for Machine Learning

Master of Science in AI and Master of Science in Data Science

@ UPSaclay

2025/2026.

Faïcel Chamroukhi

université PARIS-SACLAY

SystemX INSTITUT DE RECHERCHE TECHNOLOGIQUE

chamroukhi.com

TC2: Optimization for Machine Learning

Objective: Study the mathematical and computational constructions and properties of key optimization algorithms in different situations with use case illustrations in machine learning.

Programme

- week1: Introductions and background (convexity, differentiability, optimality conditions, convergence rates …)
- week2: Continuous optimization formulation and basic concepts; Optimization concepts over multivariate continuous spaces; Convexity and Differentiability; Gradient and Hessian concepts;
- week3: continuous optimizaiton (Gradient descent methods) : Mathetamatical construction of descent methods, Gradient Descent, Descent Directions, Convergence, rates; Step-size tuning and Line Search; Accelerations
- week4:
  - Intermediate exam (contrôle continu) (Written, during 45', accounts for 40% of the final grade)
  - Continuous optimization (Second order methods : Newton methods including Quasi-Newton, secant, IRLS)
- week5: constrained optimizaiton: Equality and Inequality constraints, Duality/Lagrangian, KKT optimality conditions;
- week6: Stochastic, Non-convex optimization (Stochastic Gradient, The EM Algorithm, SEM)
- Week7: Final exam (Written, during 90', accounts for 60% of the final grade)

Schedule

- **When & where:** Thursday afternoon (1pm—5/7pm), from November 06 till December 18, in PUIO Building
  - Details on : https://sites.google.com/view/mastersagenda/ai
- All sessions should be attended in person unless you're informed otherwise about an exceptional online session
- Little coding sessions might be organized during the courses for illustrations, a priori on Matlab (or Python)

Contact : Inquires about the course to be sent to <faicel DOT chamroukhi AT universite-paris-saclay DOT fr>

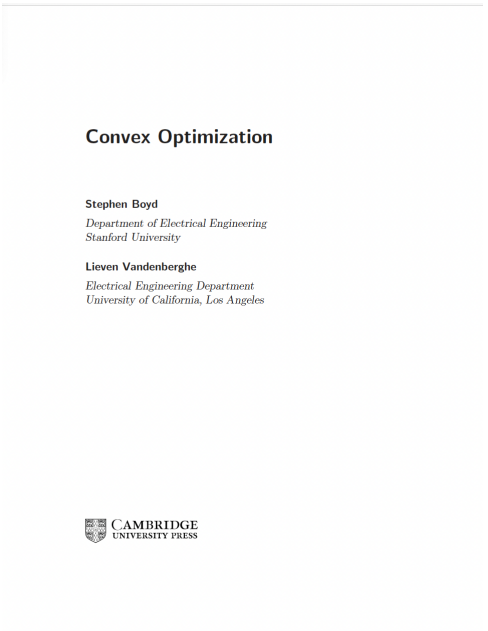
week 1
November 06. 2025

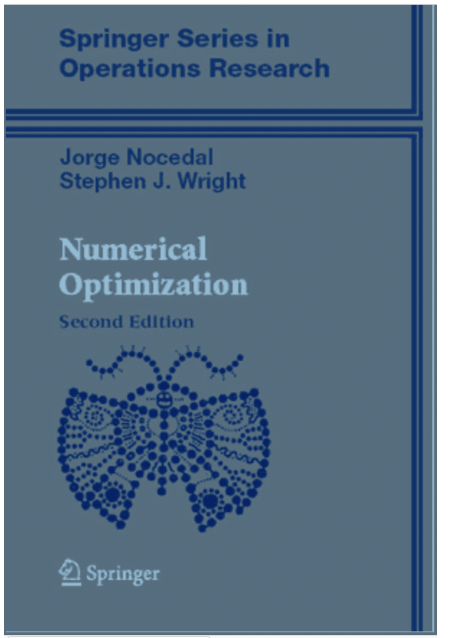
Outline

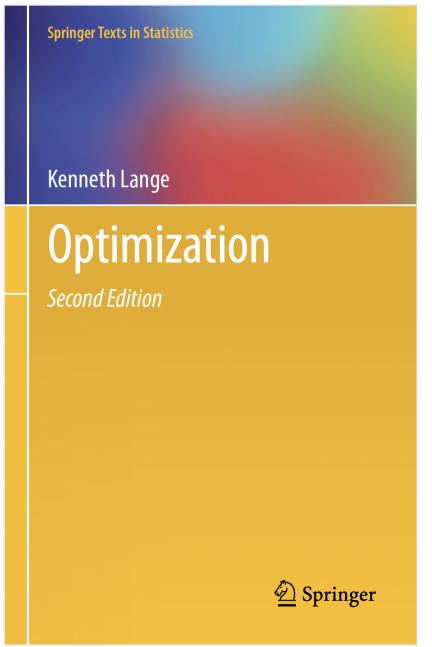
Optimization

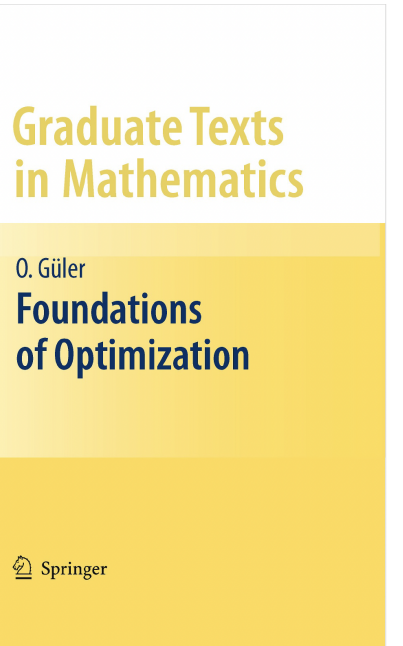
université PARIS-SACLAY

- Optimization is the science of selecting the **best element** from a set of available alternatives by optimizing (maximizing or minimizing) and **objective function**, while satisfying any given constraints. **constraints**.
- It is foundational to many fields including operational research, economics, engineering, and **machine learning**.
- Some monographs about Optimization :









Application of Optimization in Transportation

- **Problem** : Minimize transportation costs by optimizing routes and loads across a network of suppliers and customers.
- **Objective** : Minimize total distance or cost while ensuring demand and supply constraints are met.

From: Optimization of heterogeneous vehicle logistics scheduling with multi-objectives and multi-centers

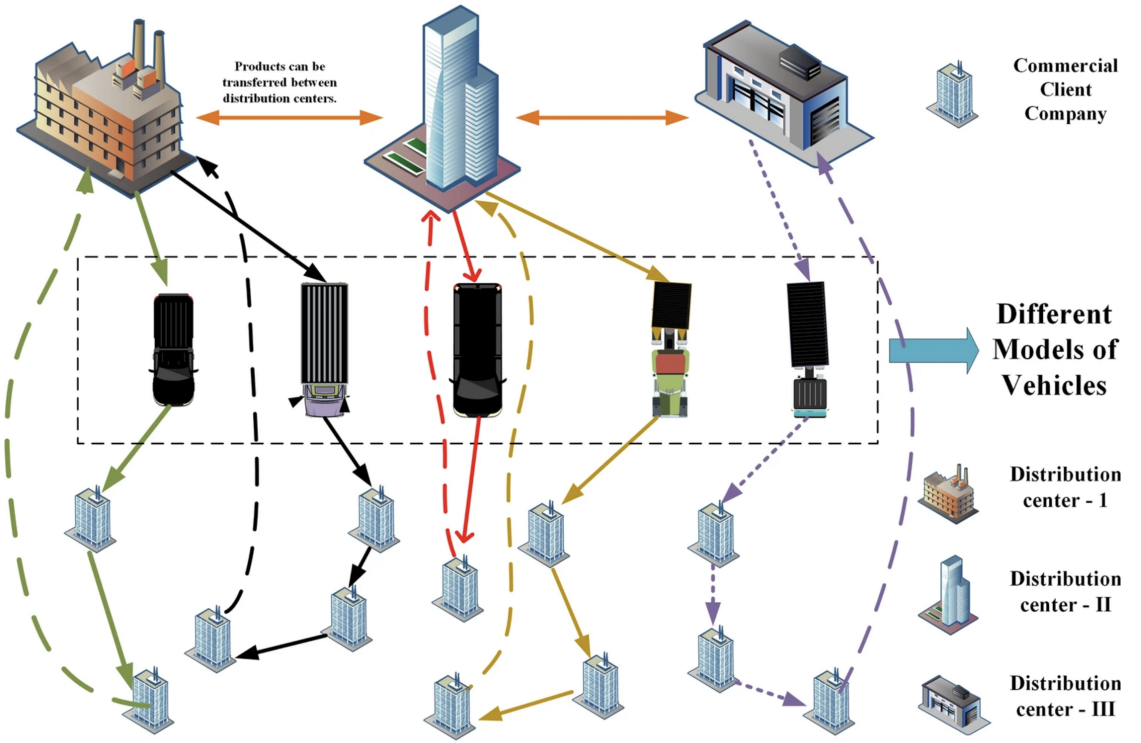


FIGURE – Example of a transportation optimization

Application of Optimization in Economics

université PARIS-SACLAY

- **Problem** : Portfolio Optimization : Allocate assets in a portfolio to maximize expected return while minimizing risk.
- **Objective** : Find the optimal weights for different assets to balance risk and return.

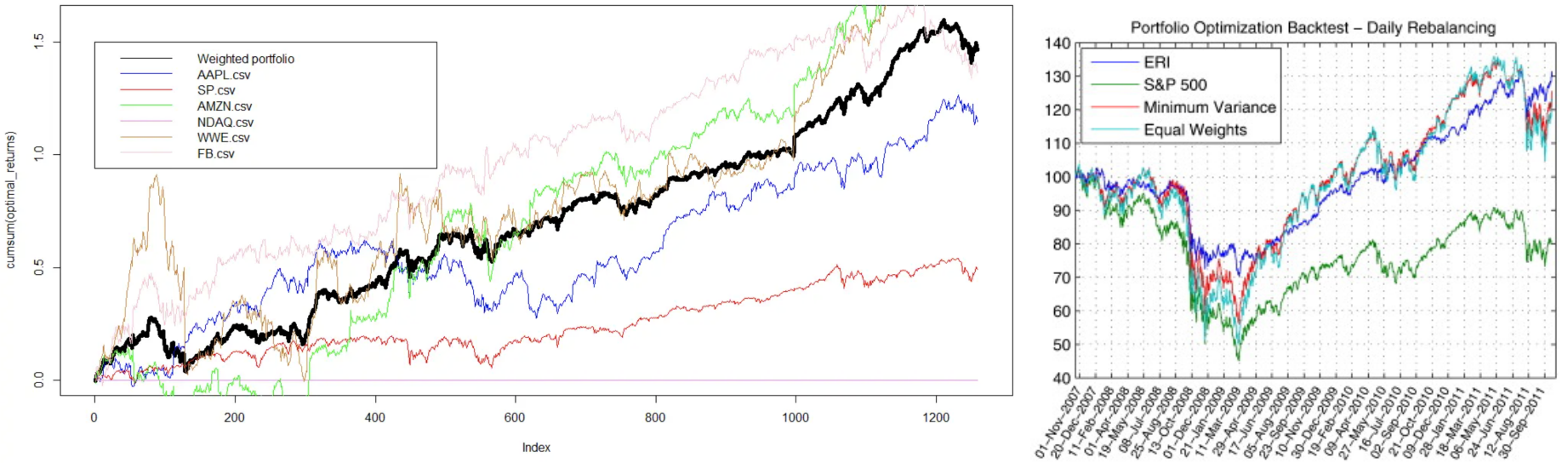


FIGURE – Example of Portfolio Optimization showing optimal risk-return trade-offs

Figure : G Mainik et al. Journal of Empirical Finance, Volume 32, June 2015, Pages 115-134

- **Problem** : Design structures (e.g., bridges, mechanic, aircraft components) to minimize weight while maintaining strength and stability.
- **Objective** : Reduce material usage and cost while ensuring the structure can withstand specified loads. eg. by Finite Element Analysis (FEA) and topology optimization.

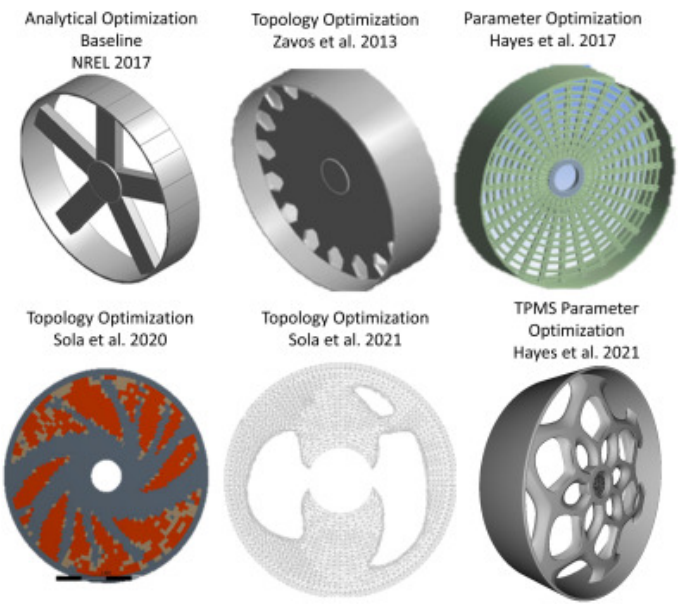


FIGURE – Topology optimization of a mechanical component

- **Problem** : Hyperparameter Optimization : Tune hyperparameters (e.g., learning rate, number of layers) to improve model performance.
- **Objective** : Maximize model accuracy or minimize loss by selecting optimal hyperparameters. by Grid / random search, or Bayesian optimization.

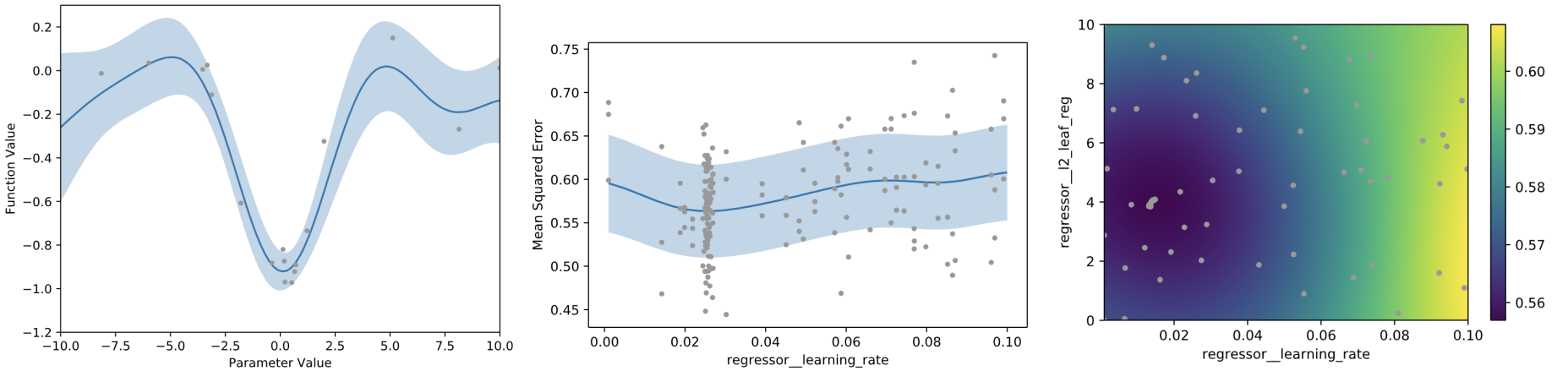


FIGURE – Example of hyperparameter (in regression) tuning in ML with Bayesian Optimization

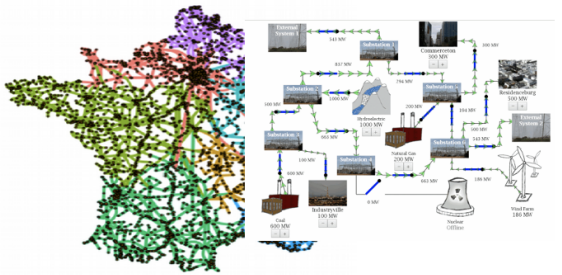
université PARIS-SACLAY

PPTX Slide on Optim for real-world applications in Engineering / Industry

Related to the **design and supervision of complex (physical) systems**

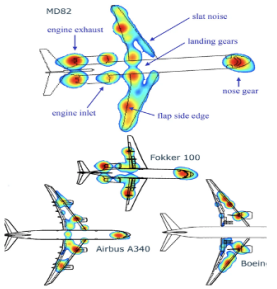
- Covering various fields in physics (mechanics, fluid dynamics, aerodynamics, electromagnetism ...)
- In a wide variety of Applications in industry, in particular in numerical **simulation**

Electricity (power grids)



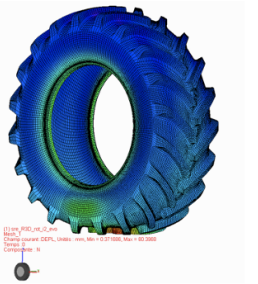
Picture from Marot, A., et al. (2018).

Aerodynamics



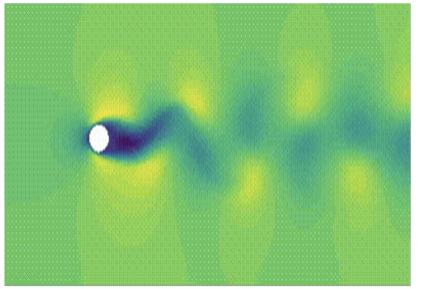
Picture form Merino-Martínez *et al. CEAS Aeronautical* Journal (2019).

Solid Mechanics pneumatics



From HSA – SystemX

Fluid Flows/Dynamics



E. Menier (PhD, LSIN/SystemX, 2024)

Optimization

université PARIS-SACLAY

Key components of an optimization problem include :

- **Objective Function** : The function $f(x)$ we aim to maximize or minimize, e.g., a prediction error in machine learning.
- **Constraints** : Conditions that solutions must satisfy, such as :
  - **Inequality constraints** : $g(x) \leq 0$
  - **Equality constraints** : $h(x) = 0$
- **Feasible Set** : The set of all possible solutions that satisfy the constraints $S = \{x \in \text{Dom} f \mid g_i(x) \leq 0 \; \forall i, \; h_j(x) = 0 \; \forall j\}$.
- **Optimal Solution** : The solution $x^\star$ that maximizes or minimizes the objective function within the feasible set.

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & f(x) \\
& \text{subject to} \\
& h_j(x) = 0, \quad \forall j = 1, \ldots, \ell \\
& g_i(x) \leq 0, \quad \forall i = 1, \ldots, q. \\
x^\star \quad & = \arg\min_{x \in \mathbb{R}^n} f(x)
\end{aligned}
$$

Optimization in Machine Learning

université PARIS-SACLAY

- x represents the **parameters** in a model.
- Constraints impose requirements on model parameters (e.g., nonnegativity, Sparsity, Boundedness, Sum-to-One..).
- The objective $f(x)$ is often expressed as a sum of two terms : eg. Minimization of the penalized empirical risk, given $\lambda > 0$

$$\min_{x \in \mathbb{R}^n} \underbrace{L(x) + \lambda \Omega(x)}_{f(x)}$$

$f(x)$ is a composite function comprising a loss term $L(x)$ and a regularization $\lambda\Omega(x)$:

  - $L(x)$: A prediction error (or loss) on some observed data.
  - $\Omega(x)$: A regularization term that penalizes model complexity
  - λ : the regularization parameter / coefficient (an hyperparameter) that controls the strength of the regularization term $\Omega(x)$ relative to the loss term $L(x)$ in the optimization objective.

[Discussion on Likelihood/posterior vs Loss/cost function]

Some challenging points in Optimization

université PARIS-SACLAY

$$\begin{cases} x^\star = \arg\min_{x\in\mathbb{R}^n} f(x) \\ \text{with } h_j(x^\star) = 0,\ \forall j = 1,\dots,\ell \\ \text{and } g_i(x^\star) \leq 0,\ \forall i = 1,\dots,q. \end{cases}$$

- **Well-posedness** of the problem (existence, uniqueness (solvability and determinacy), and stability to perturbations)
- **Optimality** conditions (characterization of the solution $x^\star$, eg. KKT or Lagrange multipliers for constrained problems)
- **Computation** of $x^\star$ (algorithmic and numerical considerations)

Two fundamental properties :

- **Convexity** : ensures global optimality.
- **Differentiability** : enables use of eg. gradient-based methods.

Optimization Issues

université PARIS-SACLAY

- **Global vs. Local Optima** : A *global optimum* is the absolute best solution, whereas a *local optimum* is only the best within a specified neighborhood of the domain of f. Many algorithms may get trapped in local optima, especially in non-convex problems.
- **Uniqueness** : A unique solution is ideal, but non-convex problems often have multiple optima, requiring extra consideration for solution selection.
- **Multimodality** : Functions with multiple peaks and valleys (multimodal) pose challenges, as algorithms like gradient descent might converge to a local rather than a global optimum.
- **Convexity** : Convex functions ensure that any local minimum is also the global minimum, simplifying optimization. Non-convex functions lack this property and are more complex to optimize.
- **Differentiability** : A differentiable function is one with a smooth, continuous slope, making it suitable for gradient-based optimization. Points where a function is not differentiable like sharp corners or sudden changes, require alternatives, such as subgradient techniques, to handle these irregularities.

Optimization Issues

université PARIS-SACLAY

Graphic illustrations/explanations TBD on the board :
global vs local optima
uniqueness
multimodality
convexity
differentiability
etc

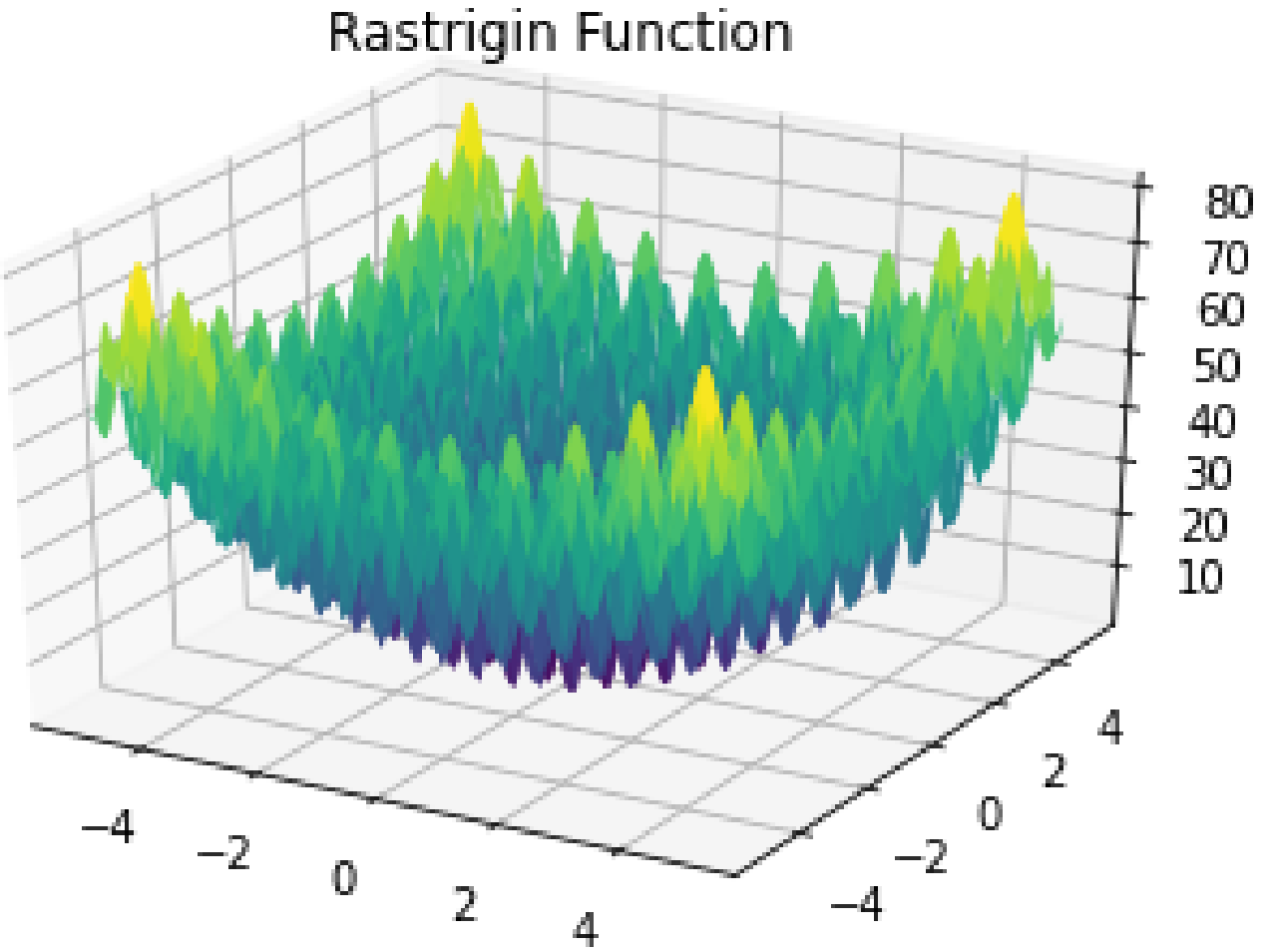
Optimization issues I

université PARIS-SACLAY

- curse of **Dimensionality** : When the number of variables is **large**, the complexity of the problem increases significantly, making it harder to visualize and solve.
- **Non-separability** : There are **dependencies between the optimization variables**, so that they cannot be optimized independently, complicating the optimization.
- **Ill-posedness** : An ill-posed problem is one where the solution does not depend continuously on the data, meaning small changes in input can lead to large variations in the output, or the solution might not even exist or be unique. Ill-posedness can lead to instability in optimization, especially in numerical methods.
- **Ill-conditioning** : The function may have a **poorly scaled gradient**, meaning small changes in certain directions lead to disproportionately large changes in others. This imbalance can cause slow or unstable convergence during optimization, as it makes it difficult for gradient-based methods to find a stable path to the optimum.
- **Ruggedness** : The function may be **non-smooth, discontinuous, multimodal, and/or noisy**. Such characteristics introduce multiple local minima and irregularities, making it challenging for optimization algorithms to find the global minimum or to converge.

$$f(x) = An + \sum_{i=1}^{n} \left(x_i^2 - A\cos(2\pi x_i) \right), \qquad \text{with} A = 10; n = 2$$

global minimum in closed form (0,0), but can be tricky for an algorithm



Types of Optimization Problems

université PARIS-SACLAY

- **Types of Optimization Problems**
  - Linear vs. Non-linear
  - Convex vs. Non-convex
  - Constrained vs. Unconstrained
  - Discrete vs. Continuous
  - Deterministic vs Stochastic

Optimization in Machine Learning

université PARIS-SACLAY

$$\begin{cases} x^\star = \arg\min_{x\in\mathbb{R}^n} f(x) \\ \text{with } h_j(x) = 0,\ \forall j = 1,\dots,\ell \\ \text{and } g_i(x) \leq 0,\ \forall i = 1,\dots,q. \end{cases}$$

- x represents the **parameters** in a model.
- Constraints impose requirements on model parameters (e.g., nonnegativity, Sparsity, Boundedness, Sum-to-One..).
- The objective $f(x)$ is often expressed as a sum of two terms : $\min_{x\in\mathbb{R}^n} L(x) + \lambda\Omega(x)$
  - A prediction error (or loss) on some observed data.
  - A regularization term that penalizes model complexity.

  - **NB** Optimization is crucial in ML for core tasks like efficient model parameter training, regularization, and hyperparamter tuning.

[Likelihood/posterior vs Loss/cost function]

- OLS : given X, Y,

$$\min_{w \in \mathbb{R}^d} \underbrace{\|Xw - Y\|^2}_{f(w)}$$

- LASSO : given X, Y, and $\lambda > 0$

$$\min_{w \in \mathbb{R}^p} \underbrace{\|Xw - Y\|^2 + \lambda \|w\|_1}_{f(w)}$$

- Minimization of the penalized empirical risk, given $\lambda > 0$

$$\min_{w \in \mathbb{R}^d} \underbrace{L(w) + \lambda \Omega(w)}_{f(w)}$$

where L is the empirical risk (data loss) and Ω is the penalization.

- Optimization without constraints

$$\min_{w \in \mathbb{R}^d} f(w)$$

where $f : \mathbb{R}^d \to \mathbb{R}$.

Optim in ML example

université PARIS-SACLAY

- **Example :** Minimizing the Mean Squared Error (MSE) in linear regression :

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (y_i - X_i \beta)^2$$

- This is a continuous, unconstrained, convex optimization problem with a closed-form solution.

(proofs later as exercices)

Examples of optimization problems in machine learning :

- **Linear Programming (LP)**
  - **Formulation** : Optimizes a linear objective subject to linear constraints.

$$\min_x \{c^T x \mid Ax \leq b\}$$

  - **Application Examples** :
  - Least Absolute Deviations (LAD) regression, which minimizes the absolute deviations between predicted and observed values, $|Ax - b|$.

  - Linear soft margin SVM : minimize $\sum_i \xi_i$ subject to the linear constraints $y_i(w^T x_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i, where ξ_i are the slack variables allowing for some misclassifications to balance margin maximization and error minimization.

Examples of optimization problems in machine learning :

- **Quadratic Programming (QP)**
  - **Formulation** : Optimizes a quadratic objective function subject to linear constraints.

$$\min_x \left\{ \frac{1}{2} x^T Q x + c^T x \mid Ax \leq b \right\}$$

  - **Application Examples** :
  - LASSO regression minimizing $\frac{1}{2}\|Xw - y\|^2$ subject to the constraint $\|w\|_1 \leq t$, where t is a parameter controlling the sparsity of the solution.

  - Hard-margin SVM : minimize $\frac{1}{2} w^T w$ subject to the constraint $y_i(w^T x_i + b) \geq 1$ for all i, which ensures correct classification with maximum margin.

Examples of optimization problems in machine learning :

- **Linear Programming (LP)**
  - **Formulation :** Minimize a linear objective subject to linear inequality constraints :

$$\min_{x \in \mathbb{R}^d} \; c^\top x \quad \text{subject to} \quad Ax \leq b$$

  - **Application Examples :**

    LAD Regression : Minimizes the ℓ_1-norm of the residuals :

$$\min_x \; \|Ax - b\|_1$$

    Soft-margin SVM : Minimizes total slack :

$$\min_{w,b,\xi} \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0$$

Examples of optimization problems in machine learning :

- **Quadratic Programming (QP)**
  - **Formulation :** Minimize a quadratic objective with linear inequality constraints :

$$\min_{x \in \mathbb{R}^d} \ \frac{1}{2} x^\top Q x + c^\top x \quad \text{subject to} \quad Ax \leq b$$

  - **Application Examples :**

    LASSO (constrained form) :

$$\min_{w} \ \frac{1}{2} \|Xw - y\|_2^2 \quad \text{s.t.} \quad \|w\|_1 \leq t$$

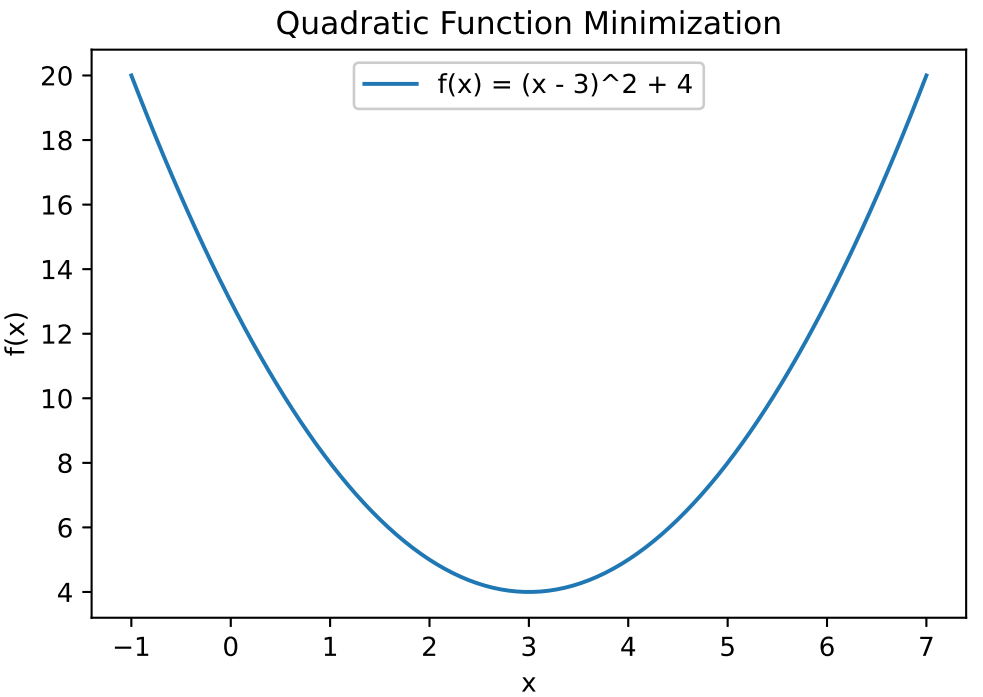
    Hard-margin SVM :

$$\min_{w,b} \ \frac{1}{2} \|w\|_2^2 \quad \text{s.t.} \quad y_i (w^\top x_i + b) \geq 1$$

Introduction to Optimization Concepts

université PARIS-SACLAY

- An **Optimization Problem** aims to minimize $f(x)$ (objective function) with variables $x \in \mathbb{R}^n$.
- **Example :** Minimize $f(x) = (x-3)^2 + 4$.
  $\hookrightarrow$ **Solution :** $f'(x) = 2(x-3) \rightarrow x = 3$ (minimum)



Introduction to Optimization Concepts

université PARIS-SACLAY

Two fundamental properties :

Convexity

Differentiability

[Done On the board why these properties matter]

- Convexity simplifies finding global optima.
- Differentiability is crucial for gradient-based methods.

- in ML point of view

- in Optim point of view

Convexity : Sets and Functions

université PARIS-SACLAY

Convex Sets and Functions :

- A set $S \subset \mathbb{R}^n$ is convex if, for any $x, y \in S$ and $\lambda \in [0, 1]$,

$$\lambda x + (1 - \lambda)y \in S.$$

  $\hookrightarrow$ This means that the line segment between any two points in the set lies entirely within the set.

- A function f is convex if for any x, y and $\lambda \in [0, 1]$,

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

  $\hookrightarrow$ This implies that the line segment connecting any two points on the graph of the function lies above the graph.

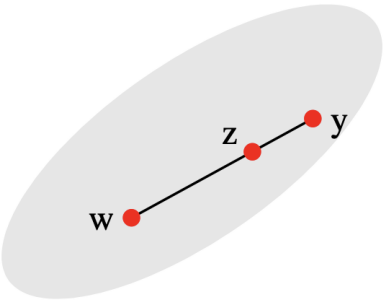
- A twice-differentiable [We'll see differentiability] function of a single variable is convex if and only if its second derivative is nonnegative on its entire domain

- **Importance of Convexity**
  - Guarantees that any local minimum is a global minimum.
  - Simplifies optimization.

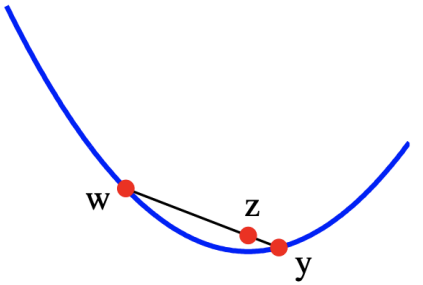
Convexity

université PARIS-SACLAY

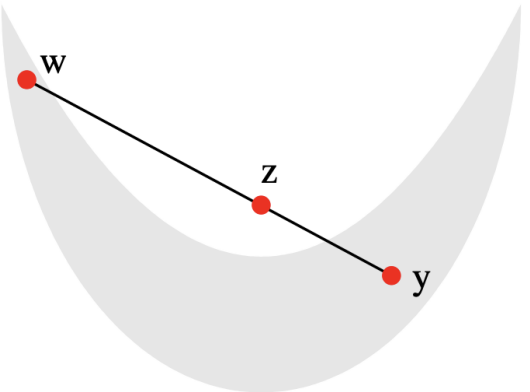
Example of a Convex Set



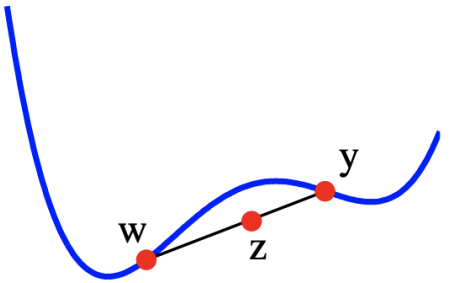
Example of a Convex Function



Example of a Non-Convex Set



Example of a Non-Convex Function

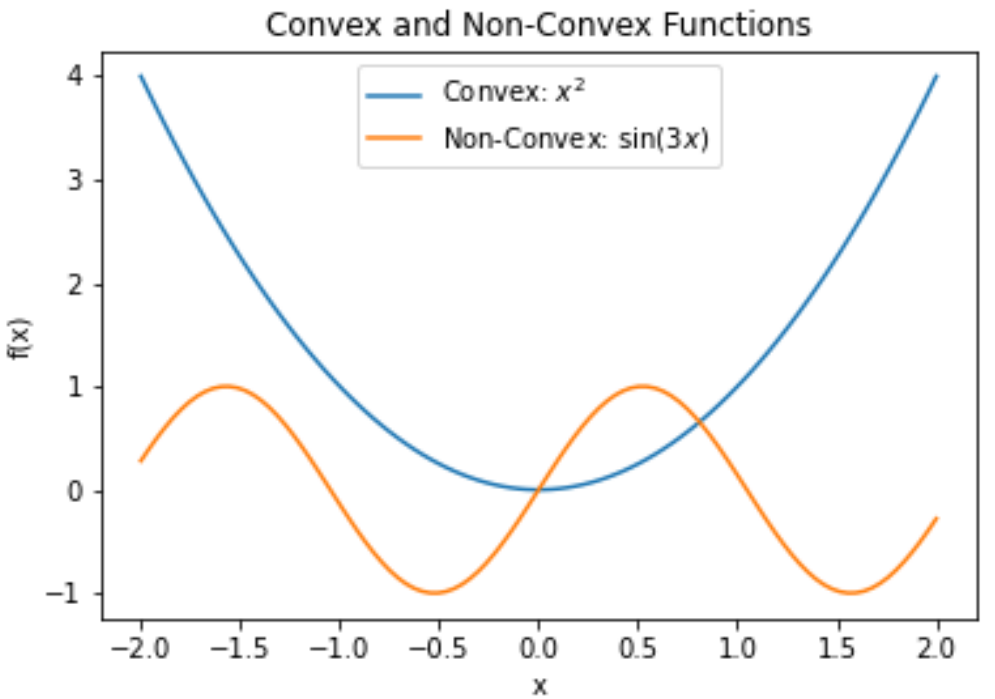


Convexity : Examples

Example :

- $f(x) = x^2$ is convex because $f''(x) = 2 > 0$.
- $f(x) = \sin(3x)$ is non-convex since it has regions of positive and negative curvature. [Exercice :)]

Proof : For $f(x) = x^2$, $f'(x) = 2x, \quad f''(x) = 2 > 0.$



Examples of Convex Functions

université PARIS-SACLAY

- Quadratic functions : $f(x) = ax^2 + bx + c$ (for $a > 0$)
- Exponential functions : $f(x) = e^x$
- Norms : $f(x) = \|x\|_2$ (common in ML regularization)

Properties

université PARIS-SACLAY

- **Operations that Preserve Convexity**
  - Non-negative weighted sum : If f_1 and f_2 are convex, then $\alpha f_1 + \beta f_2$ (with $\alpha, \beta \geq 0$) is convex.
  - Composition rules : If f is convex and increasing, and g is convex, then $f(g(x))$ is also convex.
- **Jensen's Inequality**
  - If f is a convex function and X is a random variable, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$.
- **Implications for Optimization**
  - Convexity ensures any local minimum is also a global minimum.
  - Makes it easier to design algorithms with guaranteed convergence.

Differentiability in $\mathbb{R}$ (n=1)

université PARIS-SACLAY

Definition : Let $f : \mathbb{R} \to \mathbb{R}$. We say f is differentiable at $x \in \mathbb{R}$ if

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \text{ exists, } h \in \mathbb{R}.$$

Notation :

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

The derivative corresponds to the slope of the tangent at x.

cf. the graph in a future slide

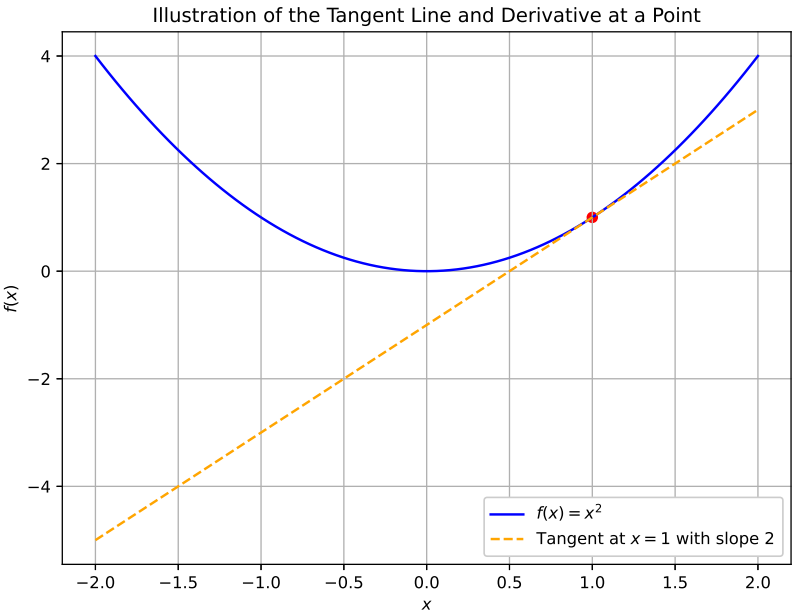
Derivative as the slope of the tangent I

The derivative of a function $f(x)$ at a point $x = a$ represents the slope of the tangent to the function at that point.

- For a function $f(x) = x^2$, the derivative is $f'(x) = 2x$.
- At a specific point $x = 1$:
  - The function value is $f(1) = 1^2 = 1$.
  - The derivative at this point is $f'(1) = 2$, which is the slope of the tangent line at $x = 1$.
- Tangent line : The tangent line at $x = 1$ can be written as :

$$\text{Tangent at } x = 1 : \quad y = f'(1) \cdot (x - 1) + f(1)$$

Derivative as the slope of the tangent II



- Visualization :
  - The curve $f(x) = x^2$ is plotted in blue.
  - The tangent line at $x = 1$, shown in orange, has a slope equal to the derivative at that point, $f'(1) = 2$.
  - This tangent line touches the curve only at $x = 1$, showing that the derivative represents the rate of change of the function at that point.

Differentiability

université PARIS-SACLAY

- **Differentiable Functions**
  - A function is differentiable if it has a derivative at each point.
  - Gradient and Hessian provide first- and second-order information.
- **Importance in Optimization**
  - Gradient descent relies on gradients to locate minima or maxima.
  - Hessian provides curvature information for Newton's method.

Differentiability and Gradient Conceptst

université PARIS-SACLAY

- For a differentiable function $f(x)$, the gradient $\nabla f(x)$ provides the direction of steepest ascent.
- Moving in the direction of $-\nabla f(x)$ minimizes the function.

Example : $f(x) = (x-3)^2 + 4$

$$\nabla f(x) = 2(x-3).$$

Gradient Descent Update : $x_{t+1} = x_t - \alpha \nabla f(x_t)$.

Optimality Conditions

université PARIS-SACLAY

- **First-Order Necessary Conditions**
  - For unconstrained problems, if f is differentiable and x^* is a local minimum, then :
$$\nabla f(x^*) = 0$$
  - **Example :** Solving $\nabla f(x) = 0$ to find stationary points in a quadratic function : $f(x) = ax^2 + bx + c$
- **Second-Order Necessary and Sufficient Conditions**
  - If x^* is a local minimum and f is twice differentiable, the second derivative is nonnegative, or for vector functions, the Hessian $H(x^*)$ is positive semidefinite ($H(x^\star) \succeq 0$, i.e all eigenvalues ≥ 0). (necessary).
  - If the Hessian is positive definite (all eigenvalues > 0), then x^* is a strict local minimum (sufficient for strict minimum).

Optimality conditions for constrained problems (KKT) to be studied later

- **Karush-Kuhn-Tucker (KKT) Conditions** : For a constrained optimization problem, the KKT conditions provide necessary conditions for optimality, combining Lagrange multipliers to handle constraints.

Convergence Rates

université PARIS-SACLAY

- **Definition and Importance**
  - Convergence rate describes how quickly an optimization algorithm approaches the optimal solution.
  - Important for comparing algorithms : faster convergence means fewer iterations and faster solutions.

Linear Convergence

université PARIS-SACLAY

- **Linear convergence** means that the error decreases by a constant fraction with each iteration.
- Formally :

$$\|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|$$

- The constant c represents the rate at which the error decreases ; a smaller c results in faster convergence.
- For linear convergence, c must satisfy $0 < c < 1$.
- **Example** : Gradient descent on strongly convex functions exhibits linear convergence, where each step moves steadily but not as rapidly as quadratic convergence.

Quadratic Convergence

université PARIS-SACLAY

- **Quadratic convergence** means that the error decreases by the square of the previous error in each iteration, resulting in very rapid convergence near the solution.
- Formally :

$$\|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2$$

- For quadratic convergence, c must be a positive constant, $c > 0$.
- This condition ensures that as k increases, the error $\|x^{(k)} - x^*\|$ reduces very quickly if the iterates are sufficiently close to the solution x^*.
- **Example** : Algorithms like Newton's method achieve quadratic convergence near the solution under conditions of sufficient smoothness and proximity , ie. when the iterates are close to the solution.

Convergence rates

université PARIS-SACLAY

Sublinear Convergence : The error decreases more slowly than linear convergence, approaching zero at a diminishing rate.

- Often observed in non-smooth or non-convex problems where faster rates are difficult to achieve.
- Common rates include $1/k$ or $1/\sqrt{k}$, where k is the iteration number.
- Common in Subgradient methods

Convergence rates

université PARIS-SACLAY

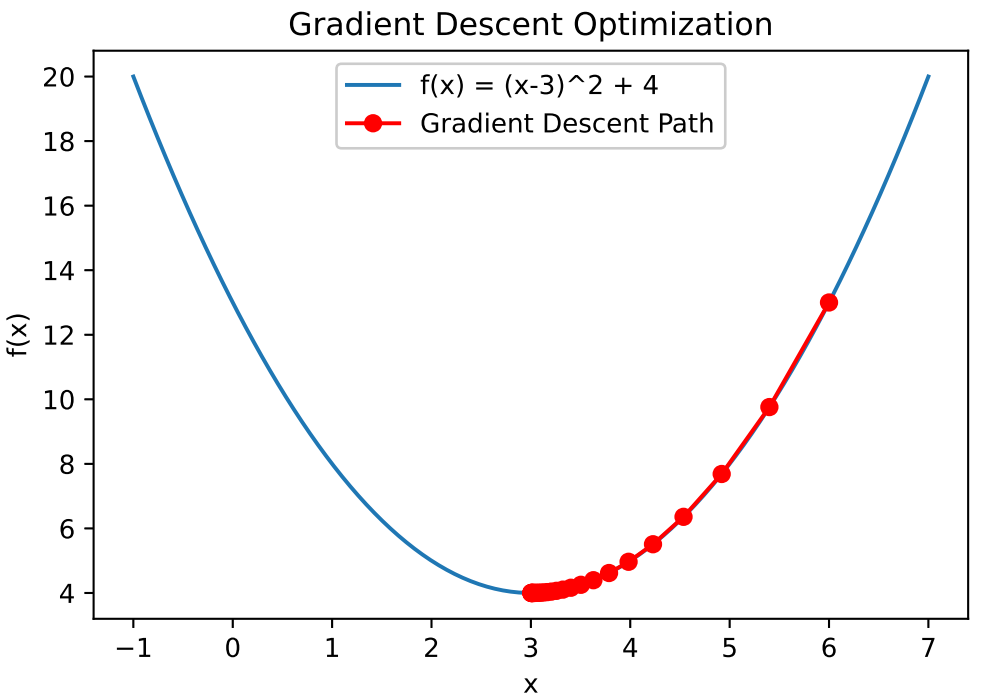
- **Factors Affecting Convergence Rates**
  - Properties of the function : smoothness, convexity, and Lipschitz continuity.
  - Choice of algorithm : Different algorithms like gradient descent or Newton's method have inherent convergence rates.
- **Trade-offs in Convergence Rates**
  - Faster-converging algorithms (e.g., Newton's method) may require more computation per iteration.
  - Balancing iteration cost with speed of convergence is essential in practical applications.

Convergence Rates

Types of Convergence :

- **Linear Convergence :** Error decreases proportionally per iteration.
- **Quadratic Convergence :** Error decreases quadratically, eg. in Newton's method.

Example : Gradient Descent on $f(x) = (x-3)^2$



Example : Linear vs. Quadratic Convergence

[Practical work : will be seen also in Python]

Objective : To demonstrate the difference in convergence rates between two optimization methods :

- Gradient Descent : Exhibits linear convergence.
- Newton's Method : Exhibits quadratic convergence.

Problem Setup :

- We consider the function $f(x) = x^2$, which has a minimum at $x = 0$.
- Both methods start from $x = 10$ and perform 10 iterations, with a learning rate $\alpha = 0.1$.

Goals :

- Compare the convergence paths of each method.
- Plot the error convergence rates to visualize the difference between linear and quadratic convergence.

Gradient Descent and Newton's Method

université PARIS-SACLAY

[Practical work : will be seen also in Python]

Example : minimize $f(x) = x^2$

Function and Derivatives :

- $f(x) = x^2$: The function to minimize.
- $f'(x) = 2x$: The first derivative, used in both methods.
- $f''(x) = 2$: The second derivative, required for Newtons method.

Gradient Descent (Linear Convergence) :

- Update rule : $x_{t+1} = x_t - \alpha f'(x_t) = x_t - 2\alpha x_t$.
- We start from $x_0 = 10$ and take 10 steps with a learning rate $\alpha = 0.1$.

Newton's Method (Quadratic Convergence) :

- Update rule : $x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)} = x_t - \frac{2}{2}x_t$.
- We use the same starting point and number of steps for comparison.

[Practical work]

Convergence Path Plot :

- For each method, we plot the function $f(x) = x^2$ and overlay the path taken by each optimization algorithm.
- This shows how each method approaches the minimum at $x = 0$.

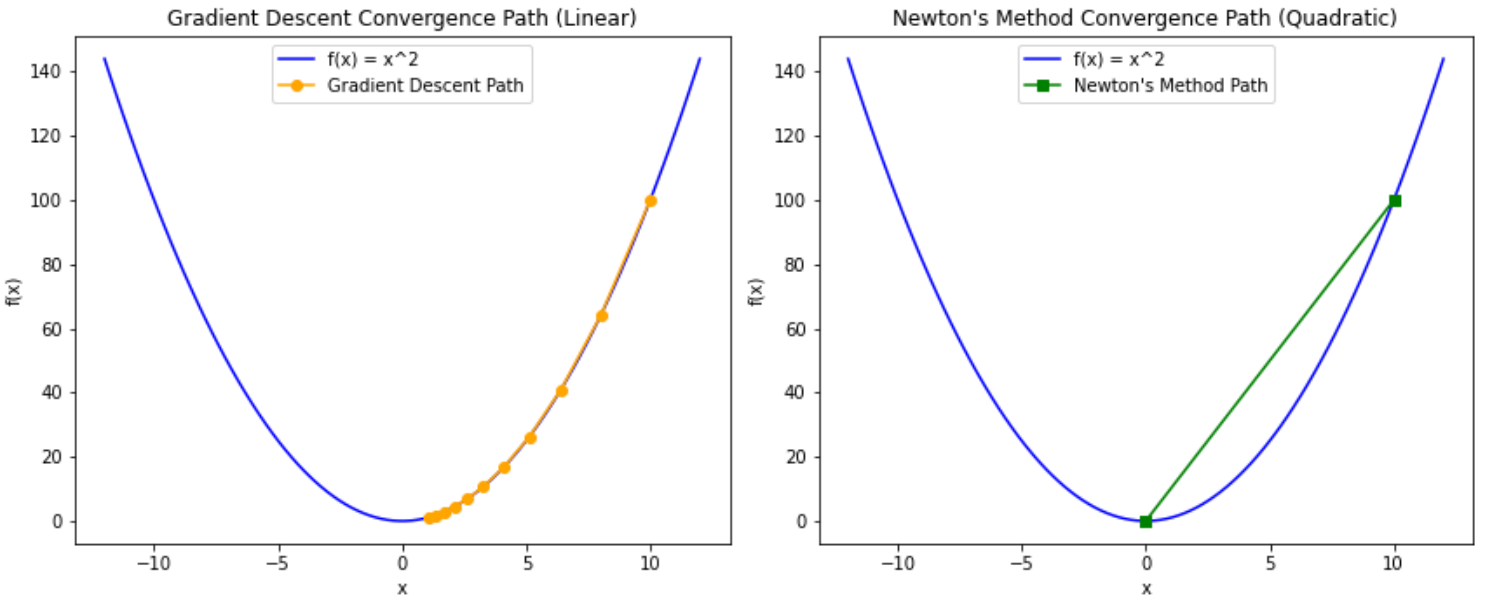


FIGURE – Convergence paths for Gradient Descent (left) and Newton's Method (right).

Any remark here for this visualisation :) ? [No convergece for GD after 10 iterations, ...]

[Practical work]

Convergence Path Plot :

- For each method, we plot the function $f(x) = x^2$ and overlay the path taken by each optimization algorithm.
- This shows how each method approaches the minimum at $x = 0$.

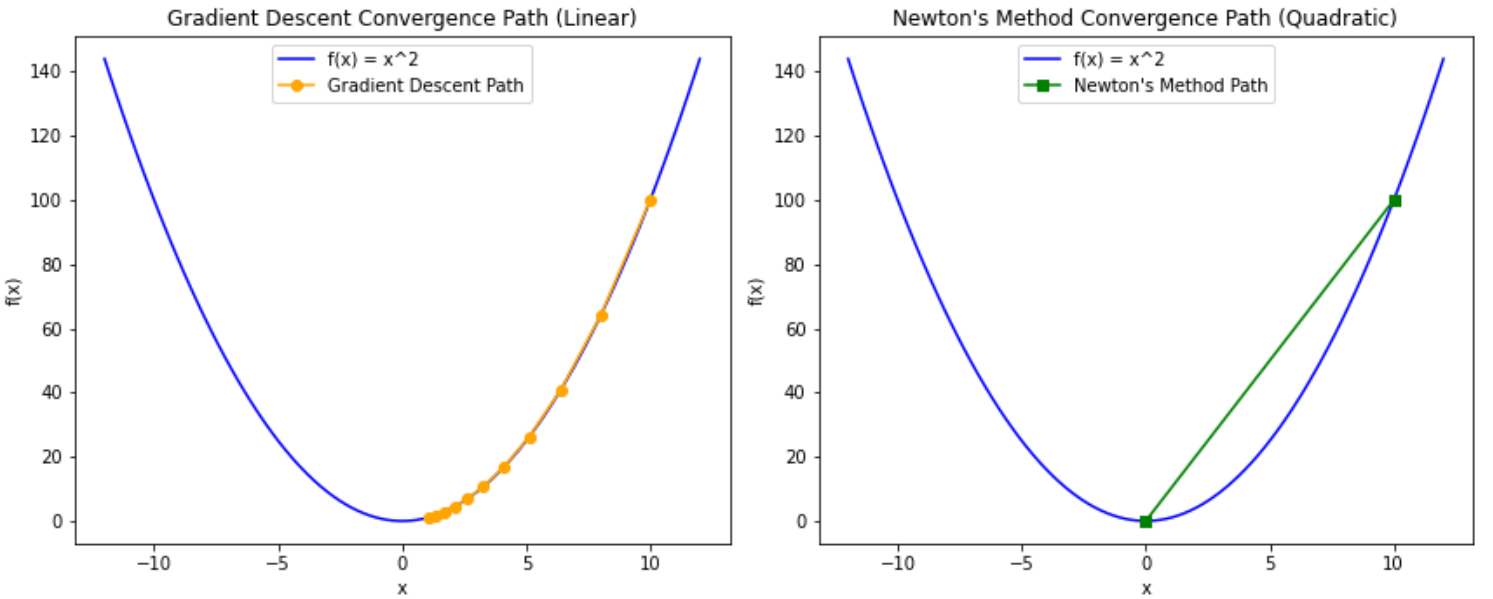


FIGURE – Convergence paths for Gradient Descent (left) and Newton's Method (right).

Any remark here for this visualisation :) ? [No convergece for GD after 10 iterations, ...]

Error convergence rates

[Practical work]

- We calculate the error (distance to the minimum) at each iteration for both methods.
- Errors are plotted on a log scale to clearly demonstrate the difference in convergence rates (linear vs. quadratic).

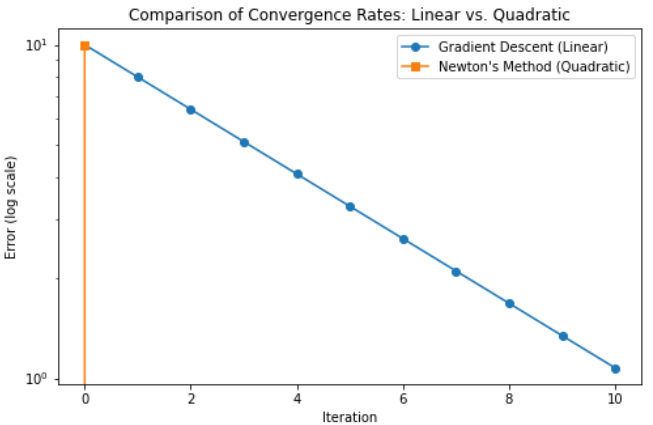


FIGURE – Error convergence rates for Gradient Descent and Newton's Method for $f(x) = x^2$.

Error Convergence Rates

[Practical work]

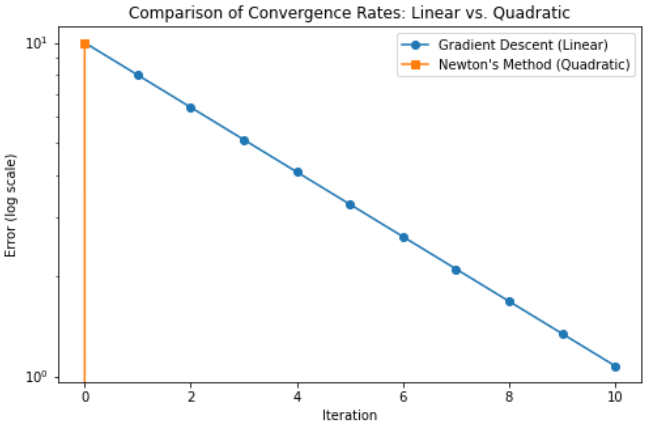


FIGURE – Error convergence rates for Gradient Descent and Newton's Method for $f(x) = x^2$.

- **Gradient Descent (Linear Convergence) :** Appears as a straight line with a moderate slope on the log scale, representing a steady reduction in error per iteration.
- **Newtons Method (Quadratic Convergence) :** Shows a sharp drop-off, indicating rapid error reduction and demonstrating quadratic convergence.
- Newton's method's quadratic convergence highlights its efficiency for well-behaved functions, but it requires second derivatives and is more computationally expensive.

Exercises [Practical Work]

université PARIS-SACLAY

Exercise 1 : Identify Convexity

- $f(x) = e^x$ is convex since $f''(x) = e^x > 0$.
- $f(x) = -x^2$ is non-convex since $f''(x) = -2 < 0$.

Exercise 2 : Gradient Computations

- For $f(x, y) = x^2 + y^2$, $\nabla f(x, y) = (2x, 2y)$.

Exercise 3 : Optimality Conditions

- For $f(x) = x^3 - 3x$, solve $f'(x) = 3x^2 - 3 = 0$ to get $x = \pm 1$.
- Check $f''(x) = 6x$ to classify critical points.

Summary

université PARIS-SACLAY

- Convexity simplifies finding global optima.
- Differentiability is crucial for gradient-based methods.
- Optimality conditions guide validity of solutions.
- Convergence rates impact algorithm efficiency.