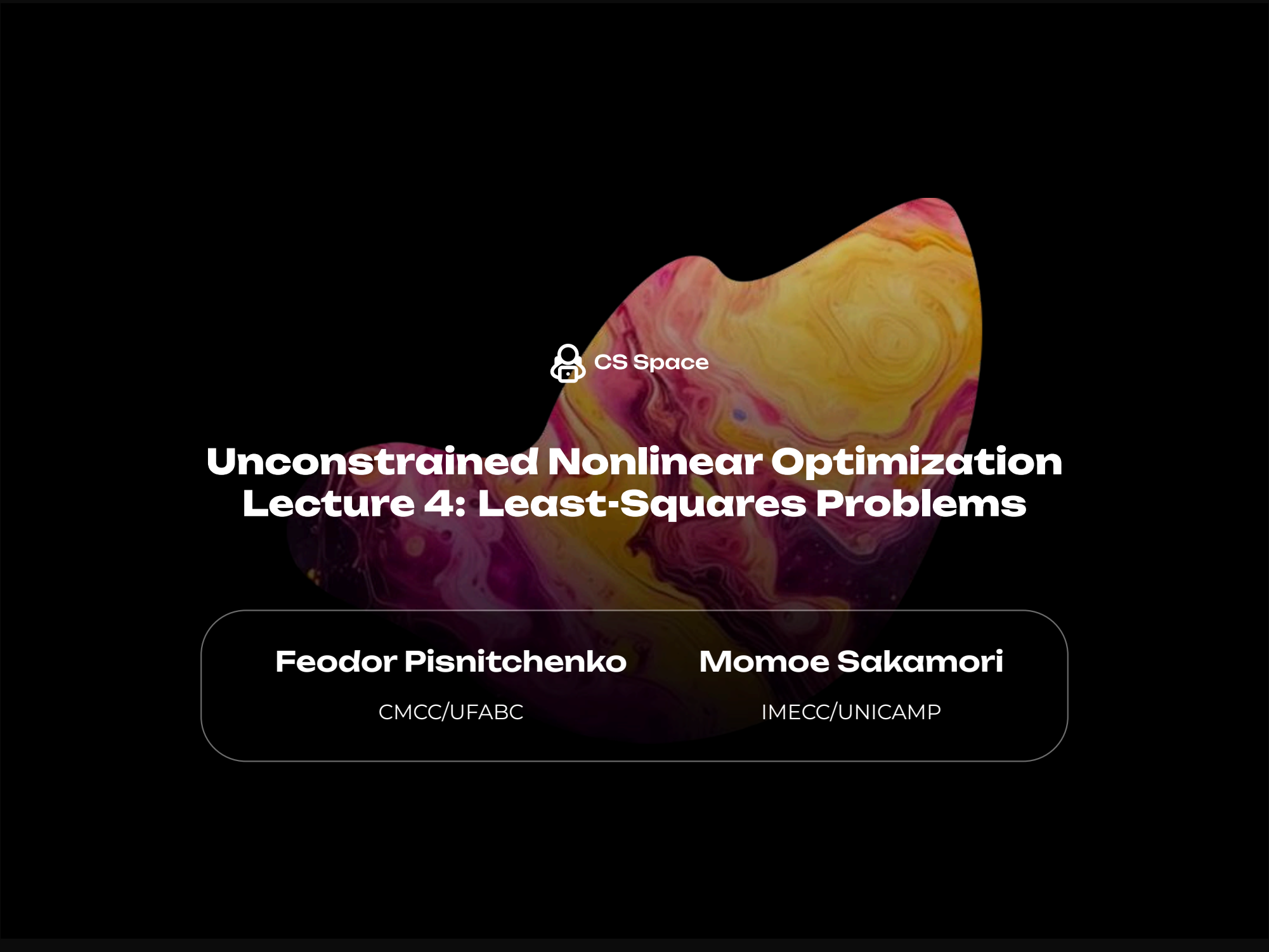


Unconstrained Nonlinear Optimization Lecture 4: Least-Squares Problems

Feodor Pisnitchenko

CMCC/UFABC

Momoe Sakamori

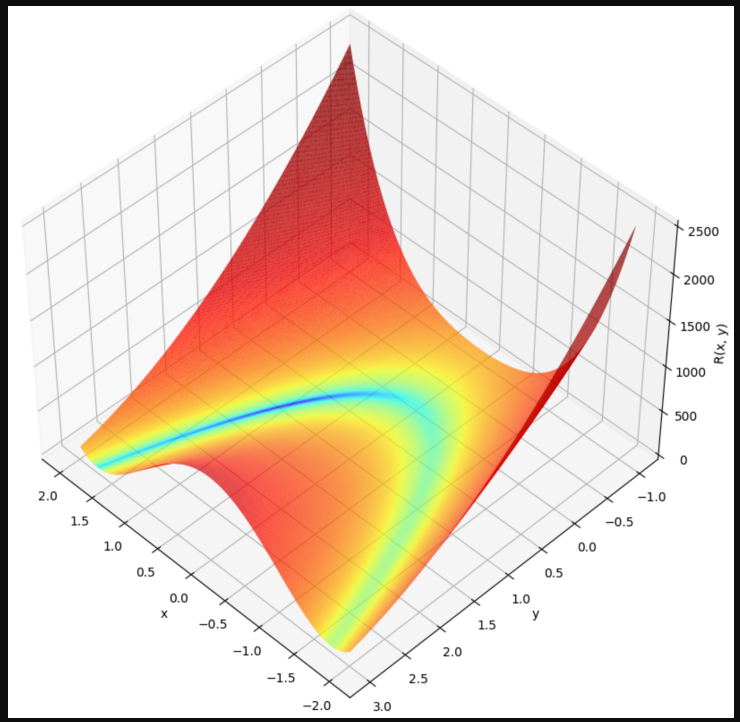
IMECC/UNICAMP

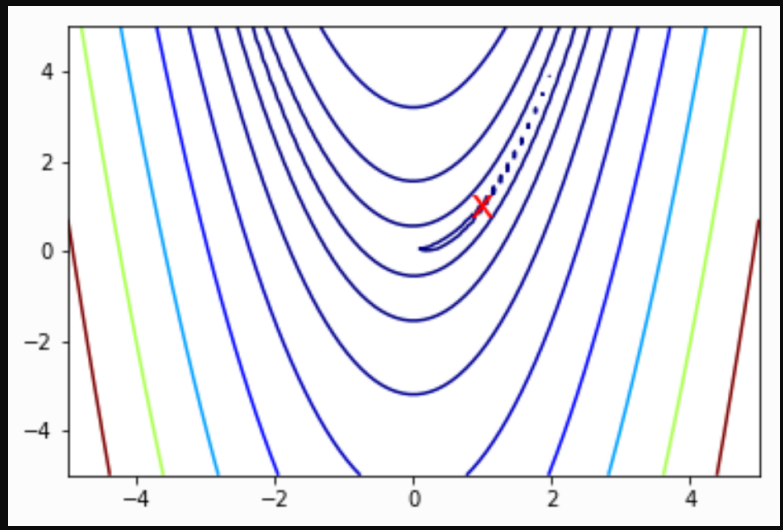
Rosenbrock Function

- The Rosenbrock function, often referred to as the *banana function* due to its distinctive shape, is defined as:

$$F(x, y) = (a - x)^2 + b(y - x^2)^2$$

- It has a global minimum at the point (a, a^2).
- Commonly used parameter values are $a = 1$ and $b = 100$.

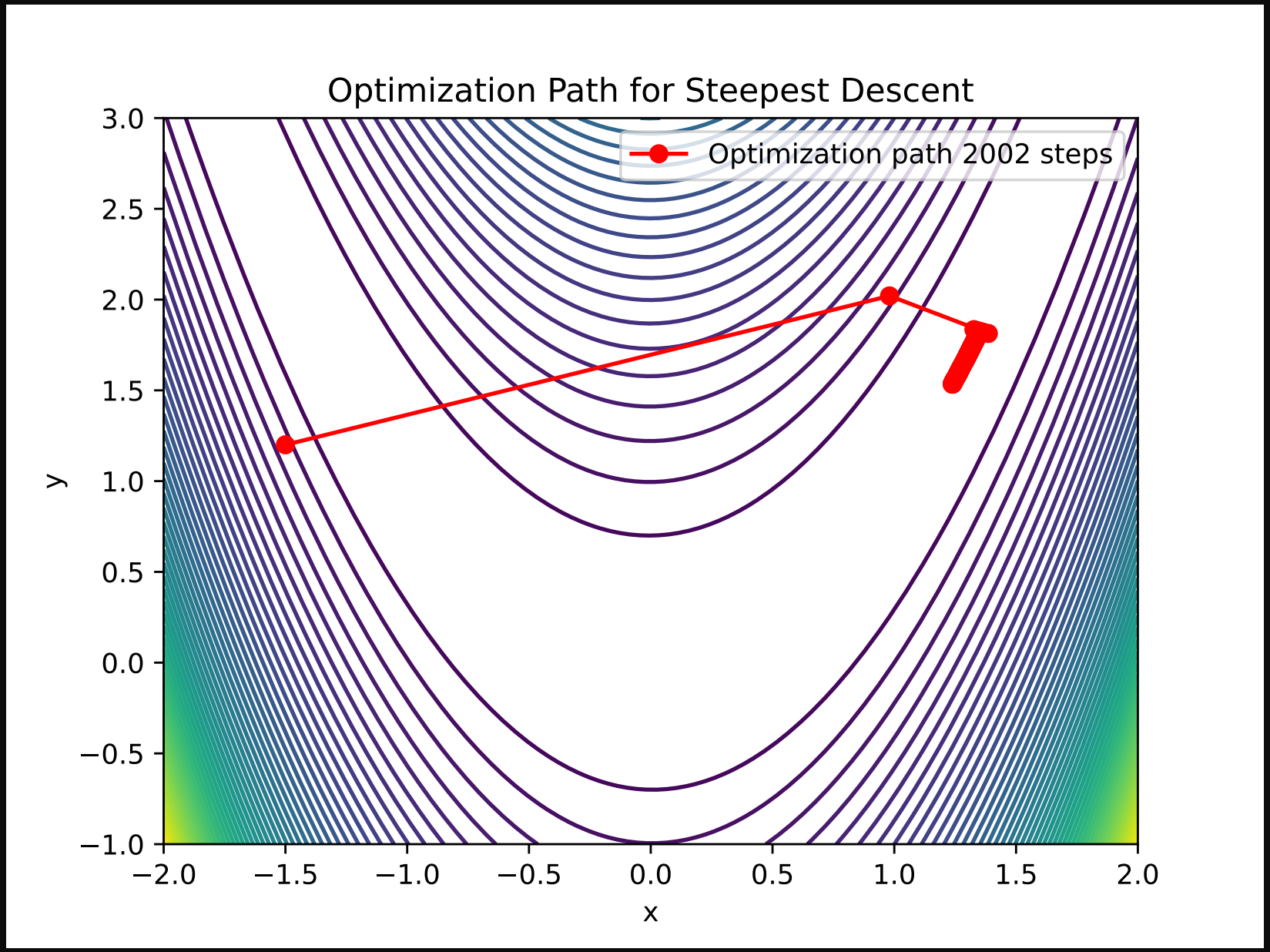




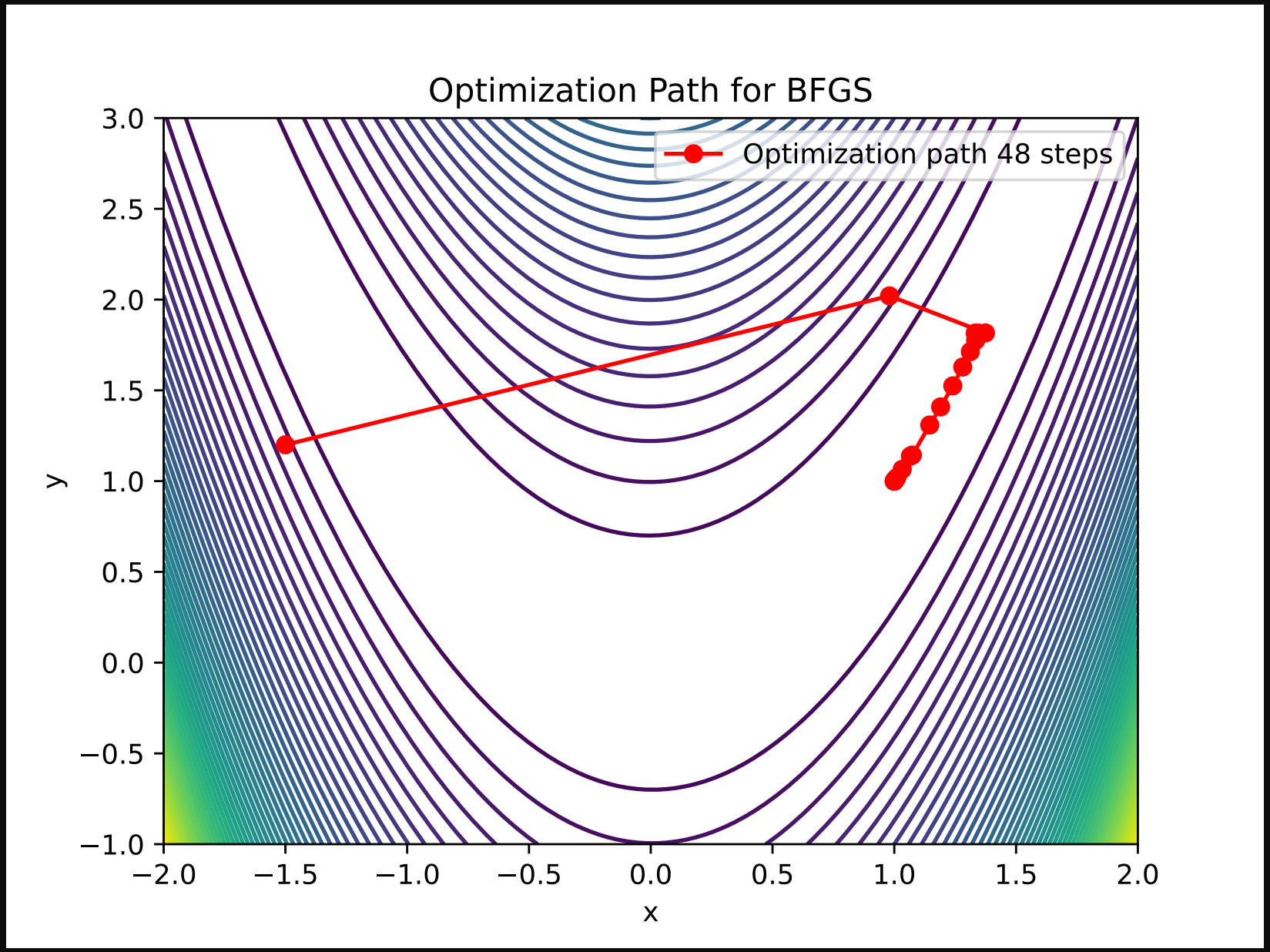
Line Search with Backtracking

For implementation details and code, visit:
`https://github.com/feodorp/nlp`

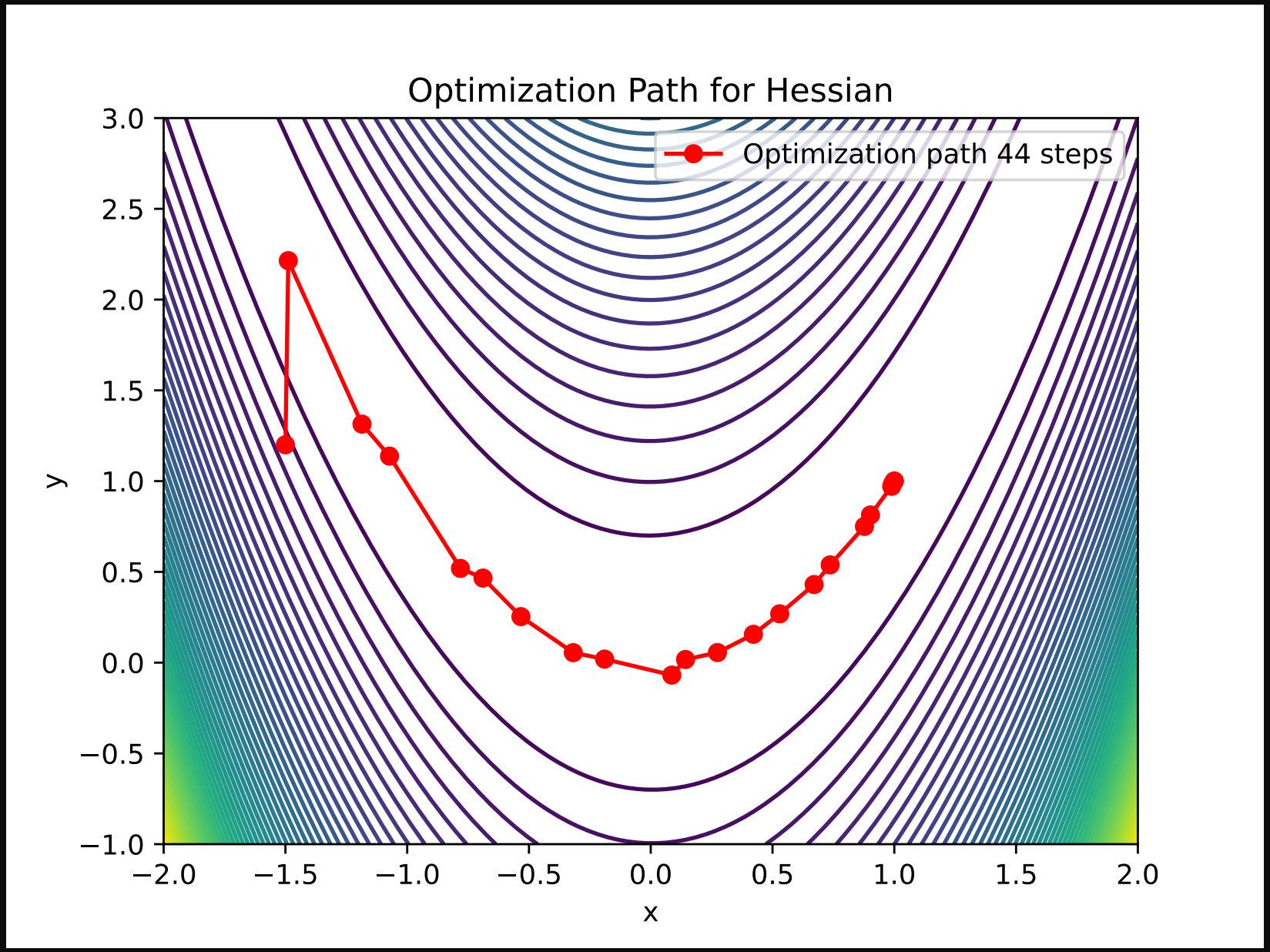
Line Search with Backtracking: Steepest Descent



Line Search with Backtracking: BFGS



Line Search with Backtracking: Modified Hessian



Least-Squares Problem Formulation

In least-squares optimization problems, the objective function F is typically formulated as:

$$F(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x),$$

where:

- Each residual function $r_j : \mathbb{R}^n \to \mathbb{R}$ is smooth (i.e., continuously differentiable).
- We assume $m \geq n$, meaning there are at least as many residuals as variables.
- The Euclidean norm is denoted by $\|\cdot\|$.

Objective Function

By defining the residual vector $r(x) = \left[r_1(x), \ldots, r_m(x)\right]^T$, we can express:

$$F(x) = \frac{1}{2}\|r(x)\|^2,$$

$$\nabla F(x) = \sum_{j=1}^{m} r_j(x)\nabla r_j(x),$$

$$\nabla^2 F(x) = \sum_{j=1}^{m} \nabla r_j(x)\nabla r_j(x)^T + \sum_{j=1}^{m} r_j(x)\nabla^2 r_j(x).$$

Jacobian Matrix of Residuals

The Jacobian matrix of the residuals, denoted $J(x)$, is given by:

$$J(x) = \begin{bmatrix} \frac{\partial r_1(x)}{\partial x_1} & \cdots & \frac{\partial r_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m(x)}{\partial x_1} & \cdots & \frac{\partial r_m(x)}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla r_1(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix},$$

where each row $\nabla r_j(x)^T$ represents the gradient of the residual function $r_j(x)$.

Objective Function (Jacobian Form)

Using the Jacobian, we can rewrite the objective function and its derivatives as:

$$F(x) = \frac{1}{2} r(x)^T r(x),$$
$$\nabla F(x) = J(x)^T r(x),$$
$$\nabla^2 F(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \approx J(x)^T J(x).$$

The approximation $\nabla^2 F(x) \approx J(x)^T J(x)$ is often used because:

1. Near the solution, each residual $r_j(x)$ may be nearly affine, making $\nabla^2 r_j(x)$ small.
2. Alternatively, the residuals $r_j(x)$ themselves may be small in magnitude.

Algorithms for Non-linear Least-Squares Problem

In this presentation, we will focus on two key algorithms for solving non-linear least-squares problems:

- The **Gauss-Newton Method**
- The **Levenberg-Marquardt Method**

Gauss-Newton Method

In the standard Newton's method, the search direction p_k^N is found by solving:

$$\nabla^2 F(x_k) p_k^N = -\nabla F(x_k)$$

However, for least-squares problems, the true Hessian is:

$$\nabla^2 F(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x)$$

In the Gauss-Newton method, we approximate the Hessian by neglecting the second term, using:

$$\nabla^2 F(x_k) \approx J_k^T J_k$$

Thus, the Gauss-Newton direction p_k^{GN} is obtained by solving:

Gauss-Newton Direction

$$J_k^T J_k p_k^{\text{GN}} = -J_k^T r_k$$

Gauss-Newton Method

- If J_k has full rank and $\nabla F_k \neq 0$, then p_k^{GN} is a descent direction because:

$$(p_k^{\text{GN}})^T \nabla F_k = (p_k^{\text{GN}})^T J_k^T r_k = -(p_k^{\text{GN}})^T J_k^T J_k p_k^{\text{GN}} = -\|J_k p_k^{\text{GN}}\|^2 < 0$$

  unless $J_k p_k^{\text{GN}} = 0$, implying $\nabla F_k = J_k^T r_k = 0$, meaning x_k is a stationary point.

- The Gauss-Newton direction p_k^{GN} minimizes the linear least-squares problem:

$$\min_p \frac{1}{2}\|J_k p + r_k\|^2$$

- Thus, p_k^{GN} can be computed using standard linear least-squares solvers.

Gauss-Newton Method

- The Gauss-Newton direction is derived from a linear approximation of the residual vector:

$$r(x_k + p) \approx r_k + J_k p$$

Thus, minimizing the quadratic model:

$$F(x_k + p) \approx \frac{1}{2}\|J_k p + r_k\|^2$$

leads to the Gauss-Newton step.

- In practice, a line search is performed along p_k^{GN} to choose a step length α_k that satisfies conditions such as the Armijo and Wolfe conditions to ensure sufficient decrease and curvature.

Gauss-Newton Method

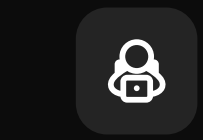
Algorithm 4: Gauss-Newton Method

1. Initialize with x_0.
2. **Repeat:**
    a. Compute the Gauss-Newton direction p_k by solving:

$$J(x_k)^T J(x_k) p_k = -\nabla F(x_k)$$

    b. Perform a line search along p_k to find a suitable step length α_k.
    c. Update the iterate: $x_{k+1} = x_k + \alpha_k p_k$
    d. Increment k.
3. **Until** convergence.

Convergence of the Gauss-Newton Method



Theorem 1

Under the following assumptions:

- Each residual function r_j is Lipschitz continuously differentiable in a neighborhood $\mathcal{N}$ of the bounded level set $\mathcal{L} = \{x \mid F(x) \leq F(x_0)\}$, where x_0 is the initial point.
- The Jacobian matrices $J(x)$ satisfy a uniform full-rank condition on $\mathcal{N}$, meaning there exists $\gamma > 0$ such that $\|J(x)z\| \geq \gamma\|z\|$ for all $x \in \mathcal{N}$ and $z \in \mathbb{R}^n$.

Then, for the sequence $\{x_k\}$ generated by the Gauss-Newton method with step lengths α_k satisfying the Wolfe conditions, we have:

$$\lim_{k\to\infty} J_k^T r_k = 0$$

Convergence of the Gauss-Newton Method

- The convergence rate of the Gauss-Newton method can be analyzed similarly to Newton's method.
- Assume x_k is close to a solution x^*, and the conditions of Theorem 1 hold.

Then:

$$x_k + p_k^{\text{GN}} - x^* = x_k - x^* - [J_k^T J_k]^{-1} \nabla F(x_k)$$

Since $\nabla F(x_k) = J_k^T r_k$ and $\nabla F(x^*) = 0$ at the solution, we can write:

$$x_k + p_k^{\text{GN}} - x^* = [J_k^T J_k]^{-1} \left[J_k^T J_k (x_k - x^*) - \nabla F(x_k) \right]$$

Recall that the true Hessian is:

$$\nabla^2 F(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) = M(x) + H(x)$$

where $M(x) = J(x)^T J(x)$ and $H(x) = \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x)$.

Convergence of the Gauss-Newton Method

Using Taylor's theorem, the gradient can be expressed as:

$$\nabla F(x_k) - \nabla F(x^*) = \int_0^1 \nabla^2 F(x^* + t(x_k - x^*))(x_k - x^*)\, dt$$

Since $\nabla F(x^*) = 0$, this becomes:

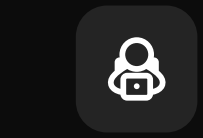
$$\nabla F(x_k) = \int_0^1 [M(x^* + t(x_k - x^*)) + H(x^* + t(x_k - x^*))](x_k - x^*)\, dt$$

Assuming $H(x)$ is Lipschitz continuous near x^*, we have:

$$\|x_{k+1} - x^*\| \leq \|[M(x^*)]^{-1} H(x^*)\| \|x_k - x^*\| + O(\|x_k - x^*\|^2)$$

where $x_{k+1} = x_k + p_k^{\text{GN}}$.

Convergence of the Gauss-Newton Method



- If $\|[J(x^*)^T J(x^*)]^{-1} H(x^*)\| < 1$, the Gauss-Newton method converges linearly with a rate depending on this norm. If $H(x^*) = 0$, convergence is quadratic.
- For large-scale problems where n and m are large, and especially if $J(x)$ is sparse, solving the linear system exactly at each iteration can be computationally expensive.
- In such cases, inexact variants of the Gauss-Newton method can be employed, similar to inexact Newton methods, by approximately solving the system with $J_k^T J_k$ in place of the true Hessian.

The Gauss-Newton Method

- **Advantages:**
  1. Achieves local quadratic convergence for zero-residual problems (where $r(x^*) = 0$).
  2. Exhibits fast local linear convergence for problems that are mildly nonlinear with small residuals.
  3. Solves linear least-squares problems in a single iteration.
- **Disadvantages:**
  1. May converge slowly (linearly) for problems that are significantly nonlinear or have large residuals.
  2. May fail to converge locally for highly nonlinear problems or those with very large residuals.
  3. The method is not well-defined if the Jacobian $J(x_k)$ does not have full column rank.
  4. Global convergence is not guaranteed without additional strategies (e.g., line search or trust region).

Levenberg-Marquardt Method

- The Gauss-Newton method lacks a built-in mechanism to control step size, often relying on a line search along the computed direction. Additionally, if the Jacobian $J(x_k)$ is rank-deficient or ill-conditioned, the Gauss-Newton step may be undefined or poorly determined. To address these issues, alternative methods like the Levenberg-Marquardt algorithm are employed.

Levenberg-Marquardt Method

Proposed by Levenberg and refined by Marquardt, the Levenberg-Marquardt method introduces a damping parameter $\lambda > 0$ into the quadratic model:

$$m(p) = \frac{1}{2}\|J(x)p + r(x)\|^2 + \frac{1}{2}\lambda\|p\|^2$$

This can be rewritten as:

$$m(p) = \frac{1}{2}\|r(x)\|^2 + p^T J(x)^T r(x) + \frac{1}{2}p^T [J(x)^T J(x) + \lambda I]p$$

The gradient of $m(p)$ is:

$$\nabla m(p) = J(x)^T r(x) + [J(x)^T J(x) + \lambda I]p$$

Since $J(x)^T J(x) + \lambda I$ is positive definite for $\lambda > 0$, the quadratic model $m(p)$ is strictly convex, ensuring a unique global minimizer.

Levenberg-Marquardt Method

The Levenberg-Marquardt direction p^{LM} is computed by solving the linear system:

$$[J(x)^T J(x) + \lambda I]p^{\text{LM}} = -J(x)^T r(x)$$

- Here, $\lambda > 0$ is known as the damping parameter or the Levenberg-Marquardt parameter.

Levenberg-Marquardt Method

To verify that p^{LM} is a descent direction:

$$\begin{aligned}(p^{\text{LM}})^T \nabla F(x) &= (p^{\text{LM}})^T J(x)^T r(x) \\ &= -(p^{\text{LM}})^T [J(x)^T J(x) + \lambda I] p^{\text{LM}} \\ &= -\|J(x) p^{\text{LM}}\|^2 - \lambda \|p^{\text{LM}}\|^2 < 0\end{aligned}$$

since $\lambda > 0$ and $p^{\text{LM}} \neq 0$ (unless $\nabla F(x) = 0$).

Levenberg-Marquardt Method

Proposition

Let $\psi(\lambda) = \|[J(x)^T J(x) + \lambda I]^{-1} b\|^2$ for $\lambda \geq 0$, where $b \in \mathbb{R}^n \setminus \{0\}$. Then $\psi(\lambda)$ is a decreasing and continuous function with $\lim_{\lambda \to \infty} \psi(\lambda) = 0$.

Levenberg-Marquardt Method

- **Proof Idea:** Using the spectral decomposition, let $J(x)^T J(x) = U \Lambda U^T$, where $\Lambda = \text{diag}(\mu_1, \ldots, \mu_n)$ with $\mu_i \geq 0$, and U is orthogonal. Then:

$$\psi(\lambda) = \| [J(x)^T J(x) + \lambda I]^{-1} b \|^2 = \sum_{i=1}^{n} \frac{v_i^2}{(\mu_i + \lambda)^2},$$

  where $v = U^T b$. Since each term decreases as λ increases, so does $\psi(\lambda)$.

- For $b = \nabla F(x) = J(x)^T r(x)$, this implies that $\| p^{\text{LM}} \|$ decreases as λ increases.

Levenberg-Marquardt Method

- The damping parameter λ controls the step size:
  - As $\lambda \to \infty$, $p^{\text{LM}} \to -\frac{1}{\lambda}\nabla F(x)$, aligning with the steepest descent direction but with a small step size.
  - As $\lambda \to 0$, $p^{\text{LM}} \to -[J(x)^T J(x)]^{-1} J(x)^T r(x)$, which is the Gauss-Newton direction.
- Thus, λ allows interpolation between the Gauss-Newton step and a short step in the steepest descent direction.

Update λ

- To adjust λ adaptively, we use the gain ratio:

$$\rho_k = \frac{F(x_k) - F(x_k + p_k^{\text{LM}})}{m_k(0) - m_k(p_k^{\text{LM}})}$$

  which measures how well the model $m_k(p)$ predicts the actual decrease in F.

- Then, update λ as:

$$\lambda_{k+1} = \begin{cases} 2\lambda_k & \text{if } \rho_k < \frac{1}{4} \\ \frac{\lambda_k}{3} & \text{if } \rho_k > \frac{3}{4} \\ \lambda_k & \text{otherwise} \end{cases}$$

Update λ

- **Interpretation of ρ_k:**
  - If $\rho_k > \frac{3}{4}$, the model m_k approximates F well, so we decrease $\lambda_{k+1} = \frac{\lambda_k}{3}$ to allow a larger step.
  - If $\rho_k < \frac{1}{4}$, the model is a poor approximation, so we increase $\lambda_{k+1} = 2\lambda_k$ to take a smaller step.
  - If $\frac{1}{4} \leq \rho_k \leq \frac{3}{4}$, the approximation is acceptable, so we keep $\lambda_{k+1} = \lambda_k$.

Update λ

- To avoid abrupt changes in λ when ρ_k is near $\frac{1}{4}$ or $\frac{3}{4}$, Nielsen proposed a continuous update formula:

$$\lambda_{k+1} = \begin{cases} \lambda_k \max\left\{\frac{1}{3}, 1-(2\rho_k - 1)^3\right\}, & \text{if } \rho_k > 0 \\ \lambda_k v_k, & \text{otherwise} \end{cases}$$

  with $v_{k+1} = 2v_k$ if $\rho_k \leq 0$, starting from some initial v_0.

- This provides a smoother adjustment of λ, potentially improving convergence.

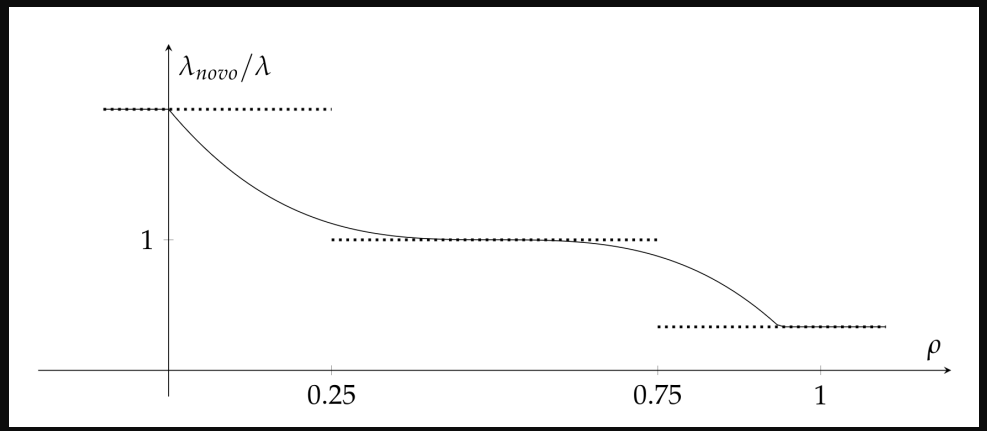


Figure: The λ update: first approach (dotted line) and Nielsen's approach (solid line).

Step Acceptance

Given $\eta \in [0, \frac{1}{4})$:

- **Accept** the step if $\rho_k > \eta$, then $x_{k+1} = x_k + p_k^{\text{LM}}$.
- **Reject** the step if $\rho_k \leq \eta$, and adjust λ accordingly.

LM Algorithm

Algorithm 5: Levenberg-Marquardt Method

1. Initialize x_0, $\eta \in [0, \frac{1}{4})$, $\lambda_0 > 0$, and set $k = 0$.
2. **Repeat:**
   a. Compute p_k by solving:

$$[J(x_k)^T J(x_k) + \lambda_k I]p_k = -J(x_k)^T r(x_k)$$

   b. Evaluate the gain ratio $\rho_k = \frac{F(x_k) - F(x_k + p_k)}{m_k(0) - m_k(p_k)}$.
   c. Update λ_{k+1} based on ρ_k (as previously described).
   d. **If** $\rho_k > \eta$:
      - Set $x_{k+1} = x_k + p_k$
   e. **Else:**
      - Set $x_{k+1} = x_k$
   f. Increment k.
3. **Until** convergence.

Levenberg-Marquardt Method

- Various modifications to the Levenberg-Marquardt method have been proposed to enhance its performance, such as incorporating second-order correction terms to better approximate the Hessian.

Rosenbrock Function

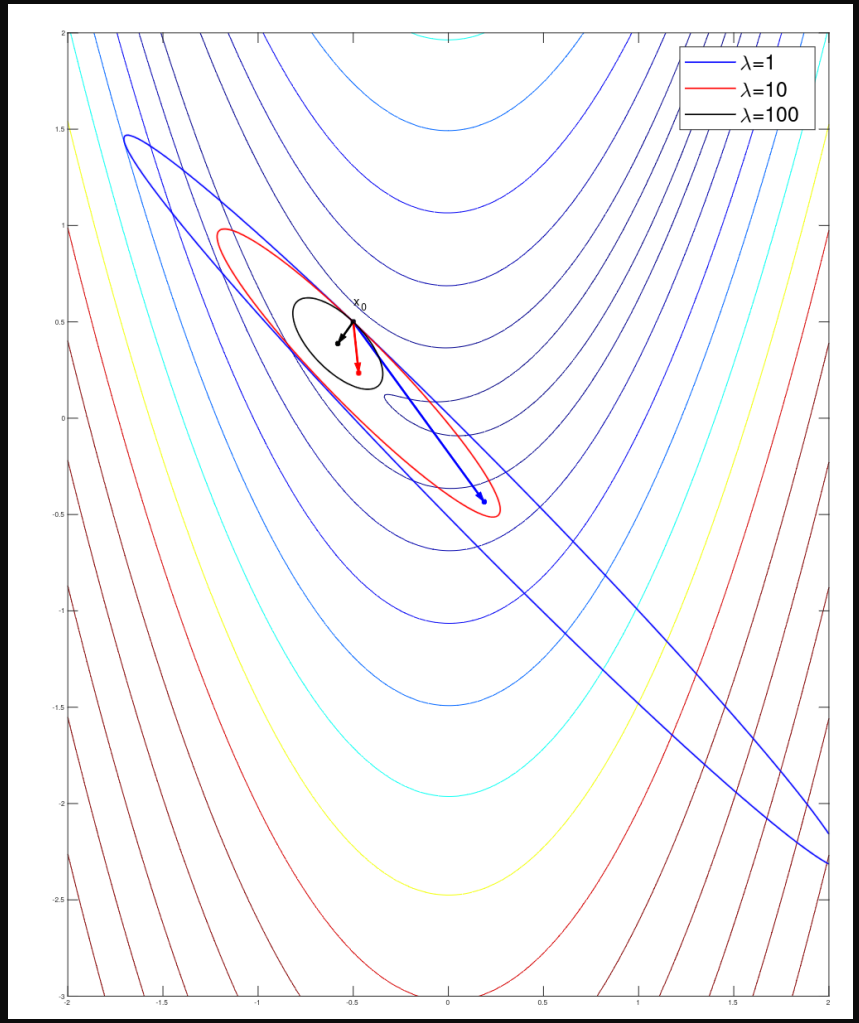


Figure: Illustration of Levenberg-Marquardt steps for the Rosenbrock function with varying damping parameters λ.

A Trust Region Method

Lemma

The Levenberg-Marquardt direction p^{LM} solves the trust-region subproblem:

$$\min_{p} \frac{1}{2}\|Jp + r\|^2 \quad \text{subject to} \quad \|p\| \leq \Delta$$

if and only if there exists $\lambda \geq 0$ such that:

$$(J^T J + \lambda I)p^{\text{LM}} = -J^T r$$

and

$$\lambda(\Delta - \|p^{\text{LM}}\|) = 0$$

with $\|p^{\text{LM}}\| \leq \Delta$.

Rosenbrock Function

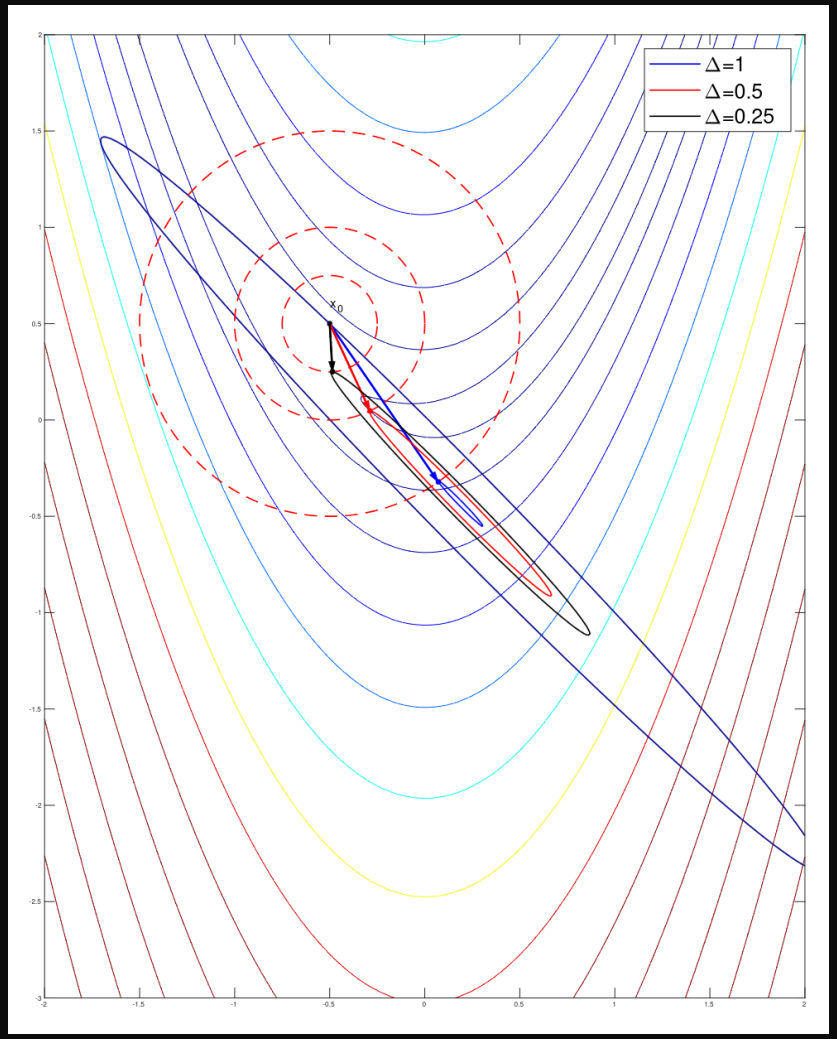
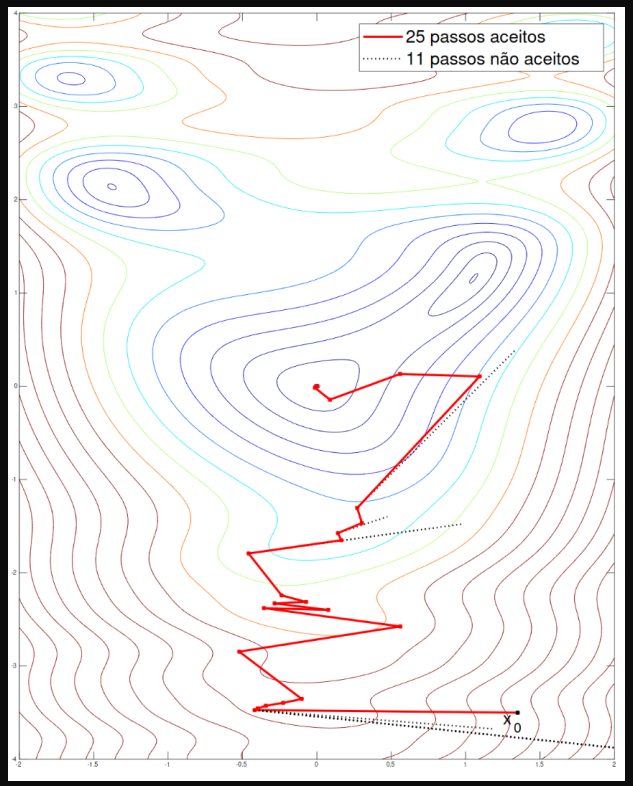


Figure: Optimization steps minimizing the quadratic model within trust regions of different sizes for the Rosenbrock function.

Example 1

Example: Application of the Levenberg-Marquardt method:

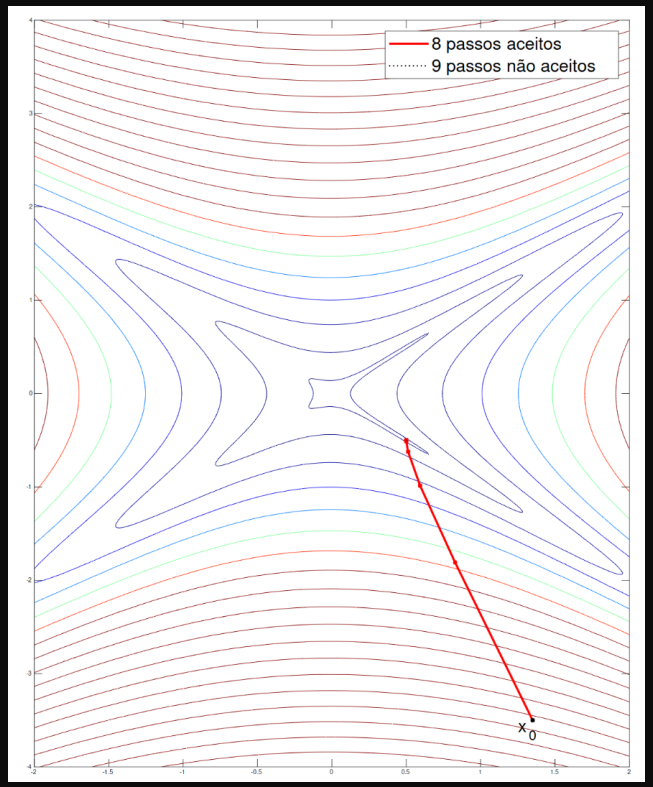
$$F(x,y) = (1-x^4)^2 + 100(y-x^2)^2 + 100(\sin(y^2)-x)^2$$



Example 2

Example: Application of the Levenberg-Marquardt method:

$$F(x, y) = (2y^2 - x)^2 + 100(y^2 - x^2)^2$$



Example 3

Example: Application of the Levenberg-Marquardt method:

$$F(x, y) = (1 - \sin(x))^2 + 100(\cos(y) - x^2)^2$$