Introduction to Optimization

Lecture 4: Continuous Optimization II (Gradient-based Optimization)

October 7, 2016

TC2 - Optimisation

Université Paris-Saclay

Anne Auger
INRIA Saclay – Ile-de-France

Dimo Brockhoff
INRIA Saclay – Ile-de-France

Course Overview

1	Fri, 16.9.2016	Introduction to Optimization
	Wed, 21.9.2016 Thu, 22.9.2016	groups defined via wiki everybody went (actively!) through the Getting Started part of github.com/numbbo/coco
2 3 4	Fri, 23.9.2016 Fri, 30.9.2016 Fri, 7.10.2016	Lecture: Benchmarking; final adjustments of groups everybody can run and postprocess the example experiment (~1h for final questions/help during the lecture) Continuous Optimization I Today's lecture: Lecture: Continuous Optimization II
	Mon, 10.10.2016	deadline for intermediate wiki report: what has been done and what remains to be done?
5 6	Fri, 14.10.2016 Mon, 17.10.2016	Lecture Lecture
	Tue, 18.10.2016 Fri, 21.10.2016	deadline for submitting data sets deadline for paper submission
 7	 Fri, 4.11.2016	vacation Final lecture
	7.-11.11.2016	oral presentations (individual time slots)
	14 - 18.11.2016	Exam (exact date to be confirmed)

All deadlines:
23:59pm Paris time

Constrained Optimization

Equality Constraint

Objective:

Generalize the necessary condition of $\nabla f(x) = 0$ at the optima of f *when f is in $\mathcal{C}^1$, i.e. is differentiable and its differential is continuous*

Theorem:

Be U an open set of $(E, \| \quad \|)$, and $f: U \to \mathbb{R}$, $g: U \to \mathbb{R}$ in $\mathcal{C}^1$.

Let $a \in E$ satisfy

$$\begin{cases} f(a) = \inf \{ f(x) \mid x \in \mathbb{R}^n, g(x) = 0 \} \\ g(a) = 0 \end{cases}$$

i.e. a is optimum of the problem

If $\nabla g(a) \neq 0$, then there exists a constant $\lambda \in \mathbb{R}$ called *Lagrange multiplier*, such that

$$\underbrace{\nabla f(a) + \lambda \nabla g(a) = 0}_{} \quad \text{Euler} - \text{Lagrange equation}$$

i.e. gradients of f and g in a are colinear

Geometrical Interpretation Using an Example

Exercise:

Consider the problem

$$\inf \ \{ f(x,y) \mid (x,y) \in \mathbb{R}^2, g(x,y) = 0\}$$

$f(x,y) = y - x^2 \qquad g(x,y) = x^2 + y^2 - 1 = 0$

1) Plot the level sets of f, plot $g = 0$
2) Compute ∇f and ∇g
3) Find the solutions with $\nabla f + \lambda \nabla g = 0$

   equation solving with 3 unknowns (x, y, λ)
4) Plot the solutions of 3) on top of the level set graph of 1)

Interpretation of Euler-Lagrange Equation

Intuitive way to retrieve the Euler-Lagrange equation:

- In a local minimum a of a constrained problem, the hypersurfaces (or level sets) $f = f(a)$ and $g = 0$ are necessarily tangent (otherwise we could decrease f by moving along $g = 0$).
- Since the gradients $\nabla f(a)$ and $\nabla g(a)$ are orthogonal to the level sets $f = f(a)$ and $g = 0$, it follows that $\nabla f(a)$ and $\nabla g(a)$ are colinear.

Generalization to More than One Constraint

Theorem

- Assume $f : U \to \mathbb{R}$ and $g_k : U \to \mathbb{R}$ $(1 \leq k \leq p)$ are $\mathcal{C}^1$.
- Let a be such that
$$\begin{cases} f(a) = \inf \{ f(x) \mid x \in \mathbb{R}^n, \quad g_k(x) = 0, \quad 1 \leq k \leq p \} \\ g_k(a) = 0 \ \text{ for all } 1 \leq k \leq p \end{cases}$$
- If $\left(\nabla g_k(a) \right)_{1 \leq k \leq p}$ are linearly independent, then there exist p real constants $(\lambda_k)_{1 \leq k \leq p}$ such that

$$\nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0$$

Lagrange multiplier

The Lagrangian

- Define the Lagrangian on $\mathbb{R}^n \times \mathbb{R}^p$ as

$$\mathcal{L}(x, \{\lambda_k\}) = f(x) + \sum_{k=1}^{p} \lambda_k g_k(x)$$

- To find optimal solutions, we can solve the optimality system

$$\begin{cases} \text{Find } (x, \{\lambda_k\}) \in \mathbb{R}^n \times \mathbb{R}^p \text{ such that } \nabla f(x) + \sum_{k=1}^{p} \lambda_k \nabla g_k(x) = 0 \\ g_k(x) = 0 \ \text{ for all } 1 \le k \le p \end{cases}$$

$$\Leftrightarrow \begin{cases} \text{Find } (x, \{\lambda_k\}) \in \mathbb{R}^n \times \mathbb{R}^p \text{ such that } \nabla_x \mathcal{L}(x, \{\lambda_k\}) = 0 \\ \nabla_{\lambda_k} \mathcal{L}(x, \{\lambda_k\})(x) = 0 \ \text{ for all } 1 \le k \le p \end{cases}$$

Inequality Constraint: Definitions

Let $\mathcal{U} = \{x \in \mathbb{R}^n \mid g_k(x) = 0 \text{ (for } k \in E),\ g_k(x) \leq 0 \text{ (for } k \in I)\}$.

Definition:

The points in $\mathbb{R}^n$ that satisfy the constraints are also called *feasible* points.

Definition:

Let $a \in \mathcal{U}$, we say that the constraint $g_k(x) \leq 0$ (for $k \in I$) is *active* in a if $g_k(a) = 0$.

Inequality Constraint: Karush-Kuhn-Tucker Theorem

Theorem (Karush-Kuhn-Tucker, KKT):

Let U be an open set of $(E, || \, ||)$ and $f: U \to \mathbb{R}$, $g_k: U \to \mathbb{R}$, all $\mathcal{C}^1$

Furthermore, let $a \in U$ satisfy

$$\begin{cases} f(a) = \inf(f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0 \; (\text{for } k \in E), g_k(x) \leq 0 \; (\text{for } k \in \mathrm{I}) \\ g_k(a) = 0 \; (\text{for } k \in E) \\ g_k(a) \leq 0 \; (\text{for } k \in I) \end{cases}$$

Let I_a^0 be the set of constraints that are active in a. Assume that $\left(\nabla g_k(a)\right)_{k \in E \cup I_a^0}$ are linearly independent.

Then there exist $(\lambda_k)_{1 \leq k \leq p}$ that satisfy

$$\begin{cases} \nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0 \\ g_k(a) = 0 \; (\text{for } k \in E) \\ g_k(a) \leq 0 \; (\text{for } k \in I) \\ \lambda_k \geq 0 \; (\text{for } k \in I_a^0) \\ \lambda_k g_k(a) = 0 \; (\text{for } k \in E \cup I) \end{cases}$$

Theorem (Karush-Kuhn-Tucker, KKT):

Let U be an open set of $(E, || \, ||)$ and $f: U \to \mathbb{R}$, $g_k: U \to \mathbb{R}$, all $\mathcal{C}^1$

Furthermore, let $a \in U$ satisfy

$$\begin{cases} f(a) = \inf(f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0 \;\; (\text{for } k \in E), g_k(x) \leq 0 \; (\text{for } k \in \mathrm{I}) \\ g_k(a) = 0 \; (\text{for } k \in E) \\ g_k(a) \leq 0 \; (\text{for } k \in I) \end{cases}$$

Let I_a^0 be the set of constraints that are active in a. Assume that $\left(\nabla g_k(a)\right)_{k \in E \cup I_a^0}$ are linearly independent.

Then there exist $(\lambda_k)_{1 \leq k \leq p}$ that satisfy

$$\begin{cases} \nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0 \\ g_k(a) = 0 \; (\text{for } k \in E) \\ g_k(a) \leq 0 \; (\text{for } k \in I) \\ \lambda_k \geq 0 \; (\text{for } k \in I_a^0) \\ \lambda_k g_k(a) = 0 \; (\text{for } k \in E \cup I) \end{cases}$$

either active constraint or $\lambda_k = 0$

Descent Methods

Descent Methods

General principle

1. choose an initial point $\boldsymbol{x}_0$, set $t = 1$
2. while not happy
   - choose a descent direction $\boldsymbol{d}_t \neq 0$
   - line search:
     - choose a step size $\sigma_t > 0$
     - set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \sigma_t \boldsymbol{d}_t$
   - set $t = t + 1$

Remaining questions

- how to choose $\boldsymbol{d}_t$?
- how to choose σ_t?

Gradient Descent

Rationale: $\boldsymbol{d}_t = -\nabla f(\boldsymbol{x}_t)$ is a descent direction

indeed for f differentiable

$$f\big(x - \sigma\nabla f(x)\big) = f(x) - \sigma||\nabla f(x)|\big|^2 + o(\sigma||\nabla f(x)||)$$

$$< f(x) \text{ for } \sigma \text{ small enough}$$

Step-size

- optimal step-size: $\sigma_t = \underset{\sigma}{\operatorname{argmin}} f(\boldsymbol{x}_t - \sigma\nabla f(\boldsymbol{x}_t))$
- **Line Search:** total or partial optimization w.r.t. σ
  Total is however often too "expensive" (needs to be performed at each iteration step)
  Partial optimization: execute a limited number of trial steps until a loose approximation of the optimum is found. Typical rule for partial optimization: Armijo rule (see next slides)

Stopping criteria:

norm of gradient smaller than ϵ

The Armijo-Goldstein Rule

Choosing the step size:

- Only to decrease f-value not enough to converge (quickly)
- Want to have a reasonably large decrease in f

Armijo-Goldstein rule:

- also known as backtracking line search
- starts with a (too) large estimate of σ and reduces it until f is reduced enough
- what is enough?
  - assuming a linear f e.g. $m_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k)$
  - expected decrease if step of σ_k is done in direction $\boldsymbol{d}$: $\sigma_k \nabla f(x_k)^T \boldsymbol{d}$
  - actual decrease: $f(x_k) - f(x_k + \sigma_k \boldsymbol{d})$
  - stop if actual decrease is at least constant times expected decrease (constant typically chosen in [0, 1])

The Armijo-Goldstein Rule

The Actual Algorithm:

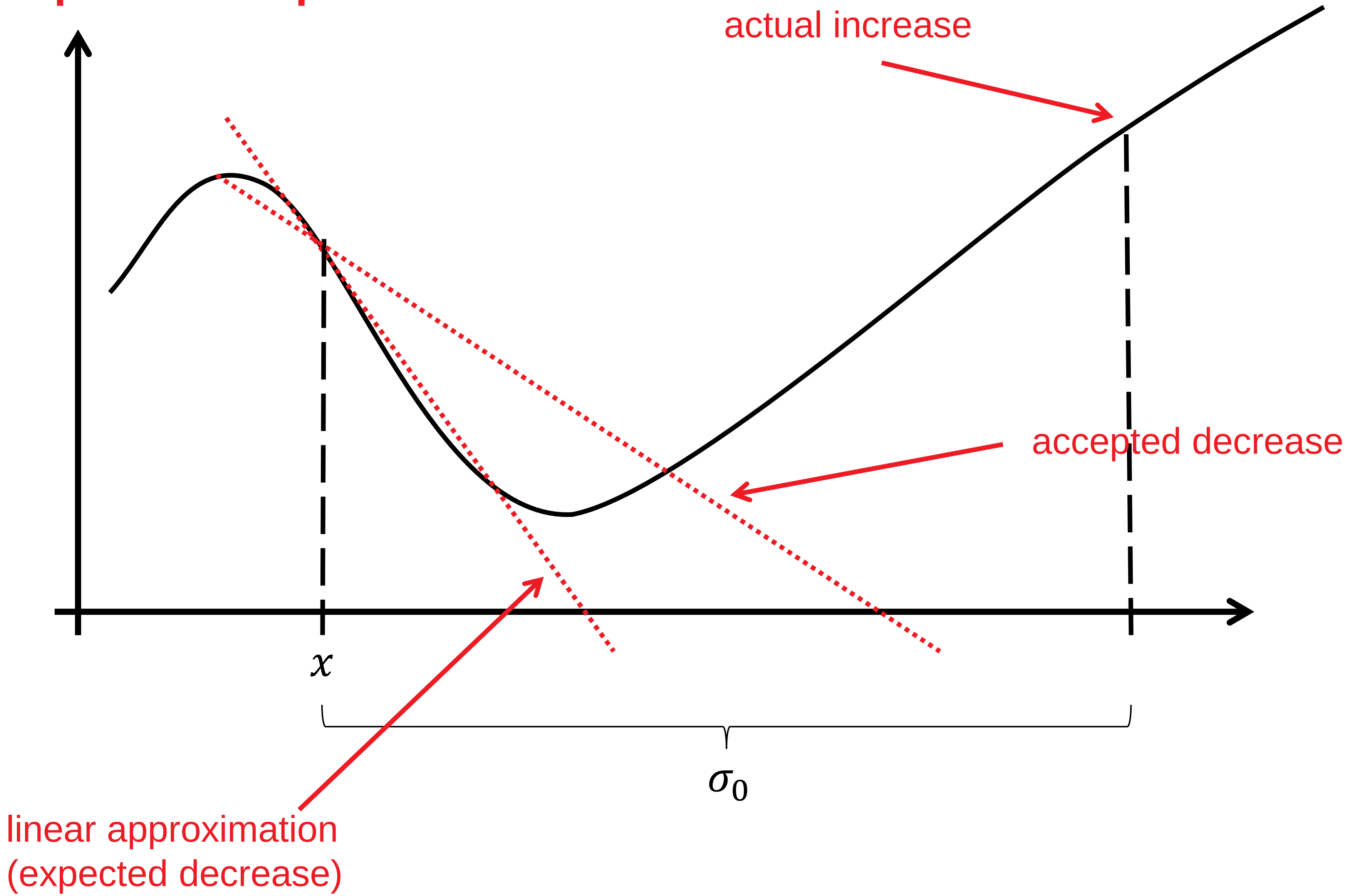
Input: descent direction $\mathbf{d}$, point $\mathbf{x}$, objective function $f(\mathbf{x})$ and its gradient $\nabla f(\mathbf{x})$, parameters $\sigma_0 = 10$, $\theta \in [0, 1]$ and $\beta \in (0, 1)$
Output: step-size σ

Initialize σ: $\sigma \leftarrow \sigma_0$
while $f(\mathbf{x} + \sigma\mathbf{d}) > f(\mathbf{x}) + \theta\sigma\nabla f(\mathbf{x})^T\mathbf{d}$ **do**
    $\sigma \leftarrow \beta\sigma$
end while

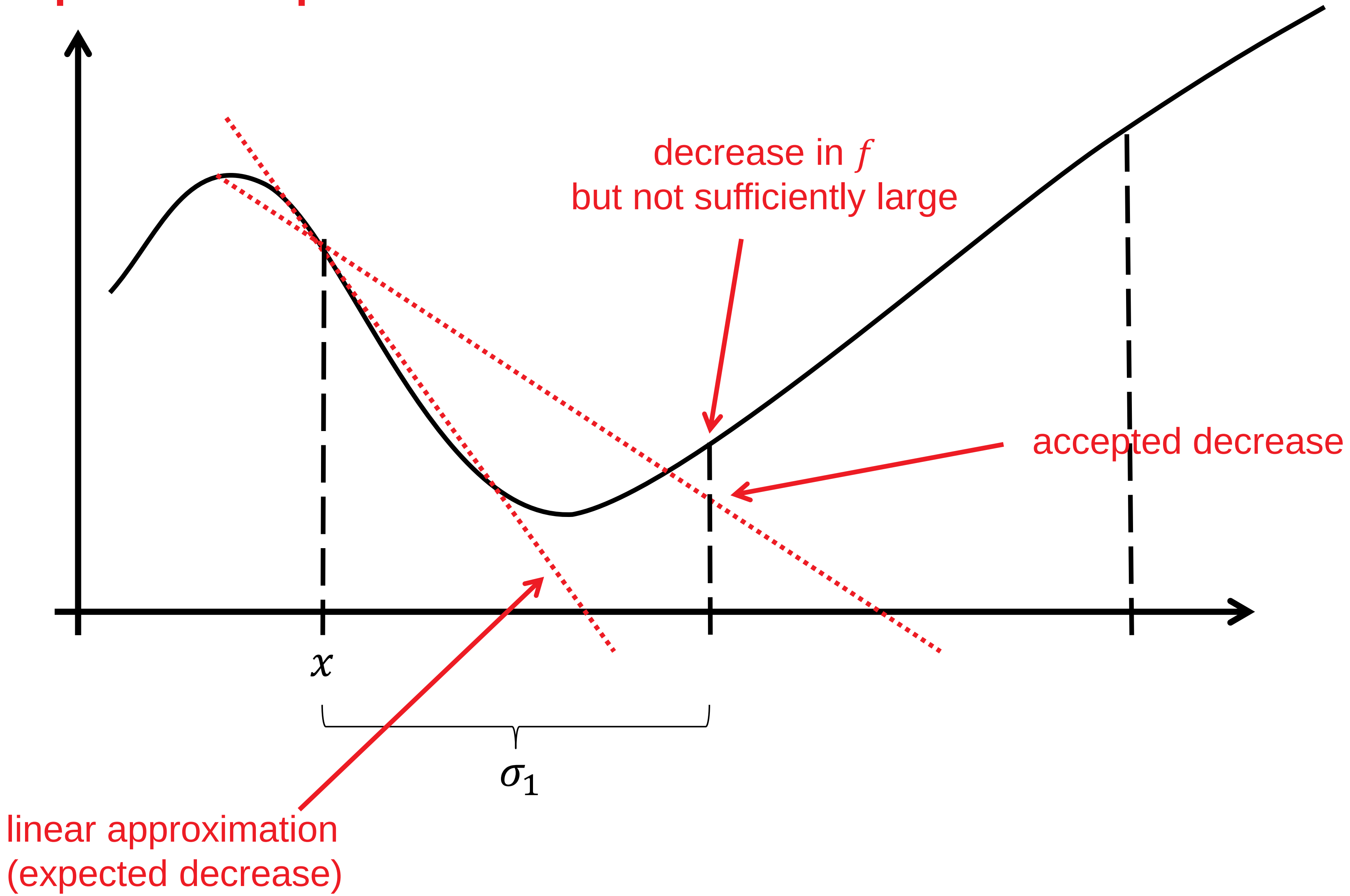
Armijo, in his original publication chose $\beta = \theta = 0.5$.
Choosing $\theta = 0$ means the algorithm accepts any decrease.

The Armijo-Goldstein Rule

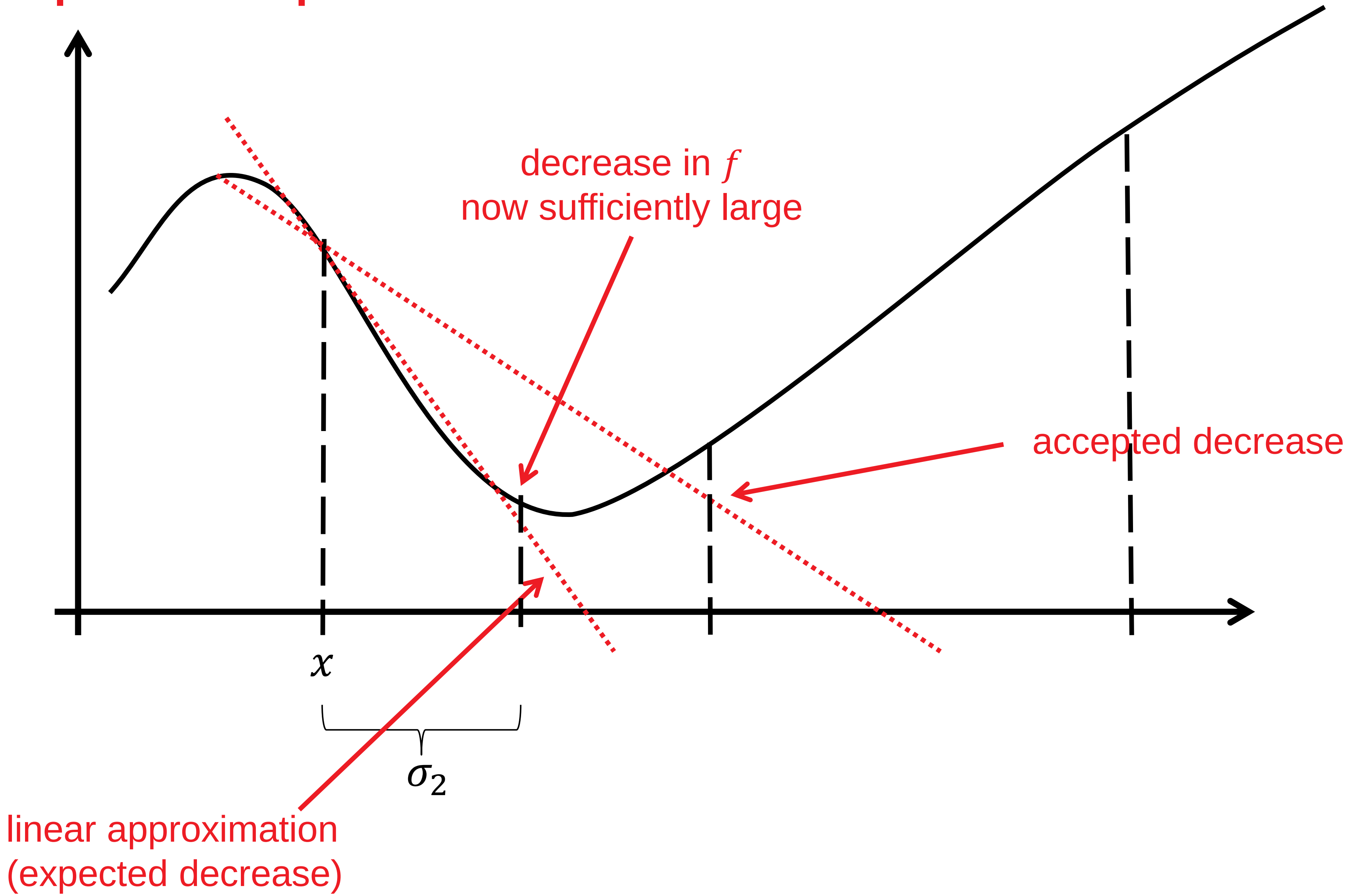
Graphical Interpretation



Graphical Interpretation



Graphical Interpretation



Gradient Descent: Simple Theoretical Analysis

Assume f is twice continuously differentiable, convex and that $\mu I_d \preccurlyeq \nabla^2 f(x) \preccurlyeq L I_d$ with $\mu > 0$ holds, assume a fixed step-size $\sigma_t = \frac{1}{L}$

Note: $A \preccurlyeq B$ means $x^T A x \leq x^T B x$ for all x

$$x_{t+1} - x^* = x_t - x^* - \sigma_t \nabla^2 f(y_t)(x_t - x^*) \text{ for some } y_t \in [x_t, x^*]$$

$$x_{t+1} - x^* = \left(I_d - \frac{1}{L} \nabla^2 f(y_t) \right)(x_t - x^*)$$

$$\text{Hence } ||x_{t+1} - x^*||^2 \leq |||I_d - \tfrac{1}{L} \nabla^2 f(y_t)|||^2 \; ||x_t - x^*||^2$$

$$\leq \left(1 - \frac{\mu}{L}\right)^2 ||x_t - x^*||^2$$

Linear convergence: $||x_{t+1} - x^*|| \leq \left(1 - \frac{\mu}{L}\right) ||x_t - x^*||$

algorithm slower and slower with increasing condition number

Non-convex setting: convergence towards stationary point

Newton Algorithm

Newton Method

- descent direction: $-[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ [so-called Newton direction]
- The Newton direction:
  - minimizes the best (locally) quadratic approximation of f:
    $\tilde{f}(x + \Delta x) = f(x) + \nabla f(x)^T \Delta x + \frac{1}{2} (\Delta x)^T \nabla^2 f(x) \Delta \mathrm{x}$
  - points towards the optimum on $f(x) = (x - x^*)^T A (x - x^*)$
- however, Hessian matrix is expensive to compute in general and its inversion is also not easy

quadratic convergence

Affine Invariance

Affine Invariance: same behavior on $f(x)$ and $f(Ax + b)$ for $A \in \mathrm{GLn}(\mathbb{R})$

- Newton method is affine invariant

  see **http://users.ece.utexas.edu/~cmcaram/EE381V_2012F/Lecture_6_Scribe_Notes.final.pdf**

- same convergence rate on all convex-quadratic functions
- Gradient method not affine invariant

Quasi-Newton Method: BFGS

$x_{t+1} = x_t - \sigma_t H_t \nabla f(x_t)$ where H_t is an approximation of the inverse Hessian

Key idea of Quasi Newton:

successive iterates x_t, x_{t+1} and gradients $\nabla f(x_t)$, $\nabla f(x_{t+1})$ yield second order information

$$q_t \approx \nabla^2 f(x_{t+1}) p_t$$

where $p_t = x_{t+1} - x_t$ and $q_t = \nabla f(x_{t+1}) - \nabla f(x_t)$

Most popular implementation of this idea: Broyden-Fletcher-Goldfarb-Shanno (BFGS)

- default in MATLAB's **`fminunc`** and python's **`scipy.optimize.minimize`**