Algorithms for Nonsmooth Optimization

Frank E. Curtis, Lehigh University

presented at

Center for Optimization and Statistical Learning,
Northwestern University

2 March 2018

Outline

Outline

Nonsmooth optimization

In mathematical optimization, one wants to

- minimize an objective
- subject to constraints

i.e.,

$$\min_{x \in \mathcal{X}} \; f(x)$$

Why *nonsmooth* optimization?

Nonsmooth optimization

In mathematical optimization, one wants to

- minimize an objective
- subject to constraints

i.e.,

$$\min_{x \in \mathcal{X}} \; f(x)$$

Why *nonsmooth* optimization? Nonsmoothness can arise for different reasons:

- physical
- technological
- methodological
- numerical

(Bagirov, Karmitsa, Mäkelä (2014))

Nonsmooth optimization

In mathematical optimization, one wants to

- minimize an objective
- subject to constraints

i.e.,

$$\min_{x \in \mathcal{X}} \; f(x)$$

Why *nonsmooth* optimization? Nonsmoothness can arise for different reasons:

- physical (phenomena can be nonsmooth)
  - phase changes in materials
- technological
- methodological
- numerical

(Bagirov, Karmitsa, Mäkelä (2014))

Nonsmooth optimization

In mathematical optimization, one wants to

- minimize an objective
- subject to constraints

i.e.,

$$\min_{x \in \mathcal{X}} \; f(x)$$

Why *nonsmooth* optimization? Nonsmoothness can arise for different reasons:

- physical (phenomena can be nonsmooth)
  - phase changes in materials
- technological (constraints impose nonsmoothness)
  - obstacles in shape design
- methodological
- numerical

(Bagirov, Karmitsa, Mäkelä (2014))

Nonsmooth optimization

In mathematical optimization, one wants to

- minimize an objective
- subject to constraints

i.e.,

$$\min_{x \in \mathcal{X}} \; f(x)$$

Why *nonsmooth* optimization? Nonsmoothness can arise for different reasons:

- physical (phenomena can be nonsmooth)
  - phase changes in materials
- technological (constraints impose nonsmoothness)
  - obstacles in shape design
- methodological (nonsmoothness introduced by solution method)
  - decompositions; penalty formulations
- numerical

(Bagirov, Karmitsa, Mäkelä (2014))

Nonsmooth optimization

In mathematical optimization, one wants to

- minimize an objective
- subject to constraints

i.e.,

$$\min_{x \in \mathcal{X}} \; f(x)$$

Why *nonsmooth* optimization? Nonsmoothness can arise for different reasons:

- physical (phenomena can be nonsmooth)
  - phase changes in materials
- technological (constraints impose nonsmoothness)
  - obstacles in shape design
- methodological (nonsmoothness introduced by solution method)
  - decompositions; penalty formulations
- numerical (analytically smooth, but practically nonsmooth)
  - "stiff" problems

(Bagirov, Karmitsa, Mäkelä (2014))

Data fitting

$$\min_{x\in\mathbb{R}^n} \ \theta(x)+\psi(x) \quad \text{where, e.g.,} \quad \theta(x)=\|Ax-b\|_2^2$$

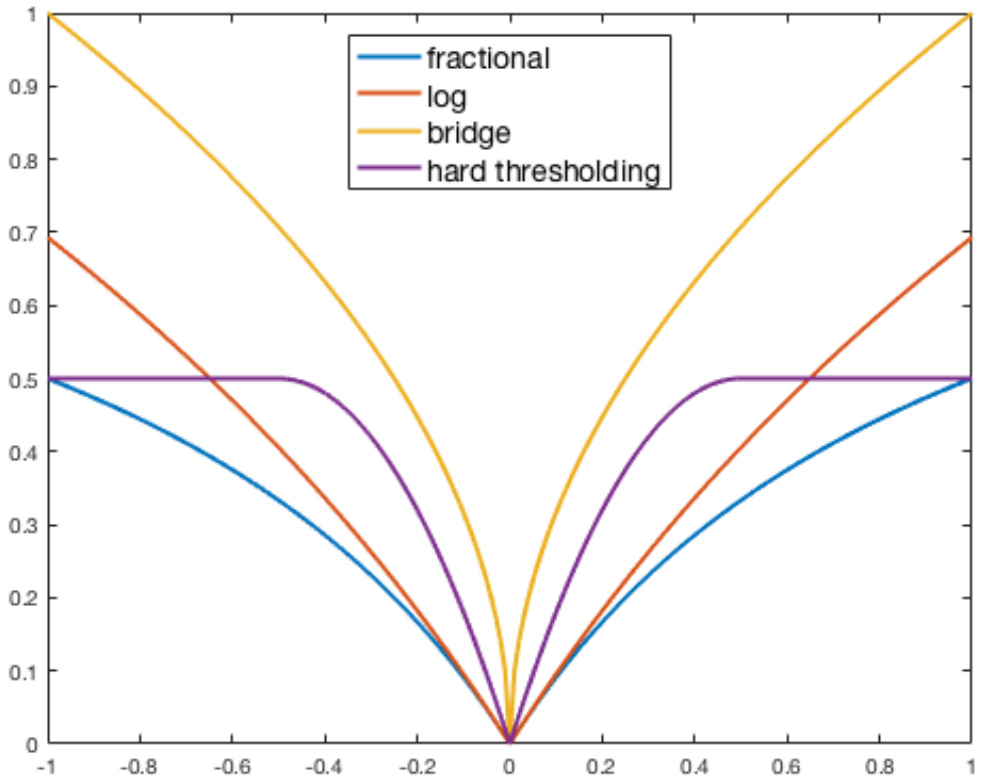
and $\psi(x)=\sum_{i=1}^{n}\phi(x_i)$ with

$$\phi_1(t)=\frac{\alpha|t|}{1+\alpha|t|},$$

$$\phi_2(t)=\log(\alpha|t|+1),$$

$$\phi_3(t)=|t|^q, \quad \text{or}$$

$$\phi_4(t)=\alpha-\frac{(\alpha-t)_+^2}{\alpha}$$



Clusterwise linear regression (CLR)

Given a dataset of pairs $\mathcal{A} := \{(a_i, b_i)\}_{i=1}^{l}$, the goal of CLR is to simultaneously

- partition the dataset into k disjoint clusters, and
- find regression coefficients $\{(x_j, y_j)\}_{j=1}^{k}$ for each cluster

in order to minimize overall error in the fit; e.g.,

$$\min_{\{(x_j,y_j)\}} f_k(\{x_j, y_j\}), \quad \text{where} \quad f_k(\{x_j, y_j\}) = \sum_{i=1}^{l} \min_{j \in \{1,\ldots,k\}} |x_j^T a_i - y_j - b_i|^p.$$

This objective is nonconvex (though it is a difference of convex functions).

Decomposition

Various types of decomposition strategies introduce nonsmoothness.

- Primal decomposition can be used for

$$\min_{(x_1,x_2,y)} \; f_1(x_1,y) + f_2(x_2,y),$$

where y is the complicating/linking variable; equivalent to

$$\min_y \; \phi_1(y) + \phi_2(y) \quad \text{where} \quad \begin{cases} \phi_1(y) := \min\limits_{x_1} \; f_1(x_1,y) \\ \phi_2(y) := \min\limits_{x_2} \; f_2(x_2,y) \end{cases}$$

This master problem may be nonsmooth in y.

Decomposition

Various types of decomposition strategies introduce nonsmoothness.

- Primal decomposition can be used for

$$\min_{(x_1,x_2,y)} \; f_1(x_1,y) + f_2(x_2,y),$$

where y is the complicating/linking variable; equivalent to

$$\min_y \; \phi_1(y) + \phi_2(y) \quad \text{where} \quad \begin{cases} \phi_1(y) := \min\limits_{x_1} \; f_1(x_1,y) \\ \phi_2(y) := \min\limits_{x_2} \; f_2(x_2,y) \end{cases}$$

This master problem may be nonsmooth in y.

- Dual decomposition can be used for same problem, reformulating as

$$\min_{(x_1,x_2,y)} \; f_1(x_1,y_1) + f_2(x_2,y_2) \quad \text{s.t.} \quad y_1 = y_2$$

The Lagrangian is separable, meaning the dual function decomposes:

$$g_1(\lambda) = \inf_{(x_1,y_1)} \; (f_1(x_1,y_1) + \lambda^T y_1)$$

$$g_2(\lambda) = \inf_{(x_2,y_2)} \; (f_2(x_2,y_2) - \lambda^T y_2)$$

Dual problem to maximize $g(\lambda) = g_1(\lambda) + g_2(\lambda)$ may be nonsmooth in λ.

Dual decomposition with constraints

Consider the nearly separable problem

$$\begin{aligned} \min_{(x_1,\ldots,x_m)} \ & \sum_{i=1}^m f_i(x_i) \\ \text{s.t. } & x_i \in \mathcal{X}_i \text{ for all } i \in \{1,\ldots,m\} \\ & \sum_{i=1}^m A_i x_i \le b \ \text{ (e.g., shared resource constraint)} \end{aligned}$$

where the last are complicating/linking constraints; "dualizing" leads to

$$\begin{aligned} g(\lambda) := \min_{(x_1,\ldots,x_m)} \ & \sum_{i=1}^m f_i(x_i) + \lambda^T \left(\sum_{i=1}^m A_i x_i - b \right) \\ \text{s.t. } & x_i \in \mathcal{X}_i \text{ for all } i \in \{1,\ldots,m\}. \end{aligned}$$

Given $\lambda \in \mathbb{R}^m$, the value $g(\lambda)$ comes from solving separable problems; the dual

$$\max_{\lambda \ge 0} \ g(\lambda)$$

is typically nonsmooth (and people often use poor algorithms!).

Control of dynamical systems

Consider the discrete time linear dynamical system:

$$\begin{aligned} y_{k+1} &= Ay_k + Bu_k \quad \text{(state equation)} \\ z_k &= Cy_k \quad \text{(observation equation)} \end{aligned}$$

Supposing we want to "design" a control such that

$$u_k = XCy_k \qquad \text{(where } X \text{ is our variable)}$$

consider the "closed loop system" given by

$$\begin{aligned} y_{k+1} &= Ay_k + Bu_k \\ &= Ay_k + BXCy_k \\ &= (A + BXC)y_k. \end{aligned}$$

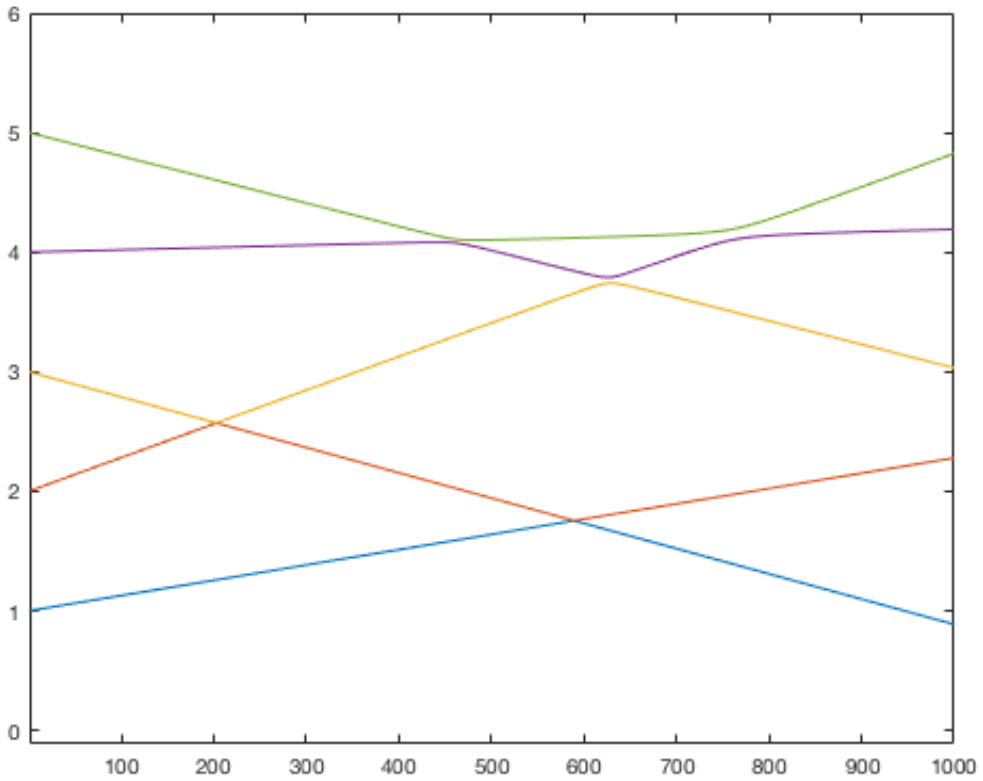
Common objectives are to minimize a stability measure

$$\rho(A + BXC),$$

which are often functions of the eigenvalues of $A + BXC$.

Eigenvalue optimization

Plots of ordered eigenvalues as matrix is perturbed along a given direction:



Other sources of nonsmooth optimization problems

- Lagrangian relaxation
- Composite optimization (e.g., penalty methods for "soft constraints")
- Parametric optimization (e.g., for model predictive control)
- Multilevel optimization

Outline

Derivatives

When I teach an optimization class, I always start with the same question:

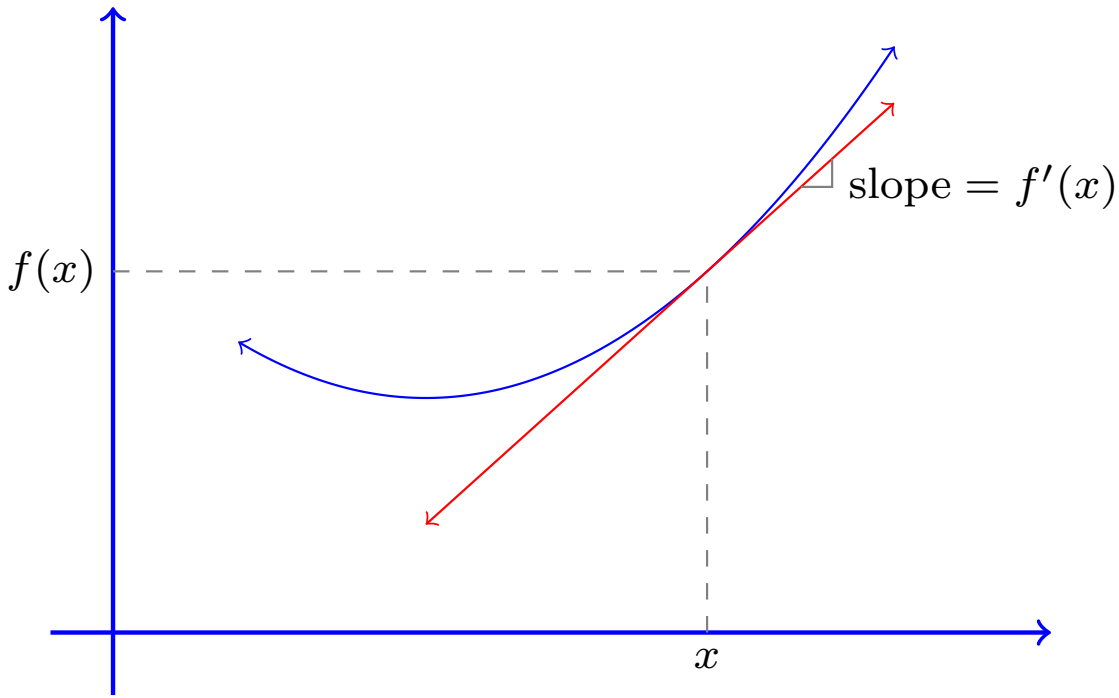
What is a derivative? ($f : \mathbb{R} \to \mathbb{R}$)

Derivatives

When I teach an optimization class, I always start with the same question:

What is a derivative? $(f : \mathbb{R} \to \mathbb{R})$

Answer I get: "slope of the tangent line"



Gradients

Then I ask:

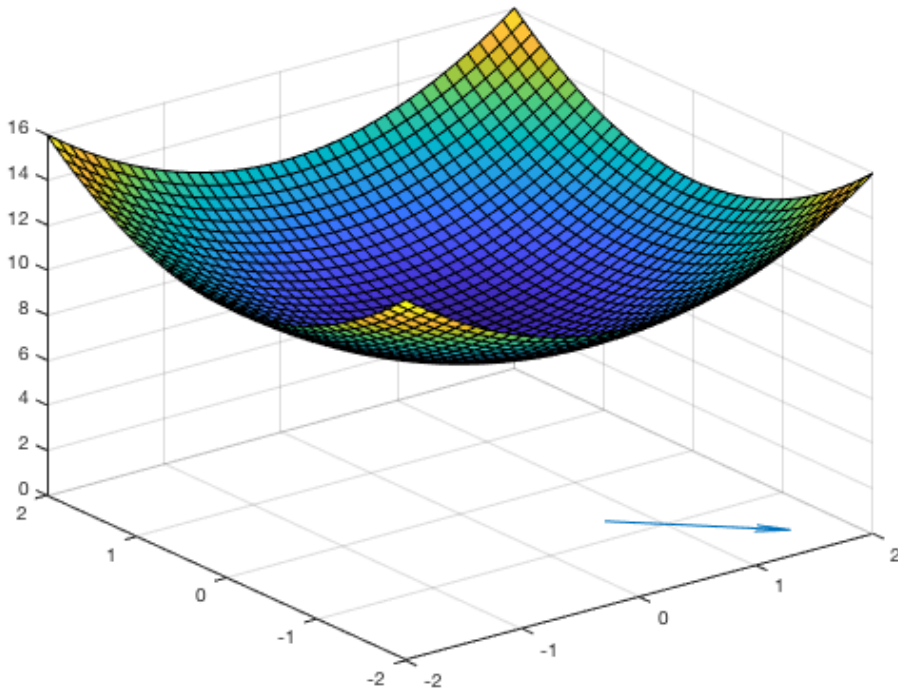
What is a gradient? $(f : \mathbb{R}^n \to \mathbb{R})$

Gradients

Then I ask:

What is a gradient? $(f : \mathbb{R}^n \to \mathbb{R})$

Answer I get: "direction along which the function increases at the fastest rate"



Derivative vs. gradient

So if a derivative is a magnitude (here, a slope), then why does it generalize in multiple dimensions to something that is a direction?

$$(n = 1) \qquad f'(x) = \frac{df}{dx}(x) = \frac{\partial f}{\partial x}(x)$$

$$(n \geq 1) \qquad \nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

What's important? Magnitude? direction?

Derivative vs. gradient

So if a derivative is a magnitude (here, a slope), then why does it generalize in multiple dimensions to something that is a direction?

$$(n = 1) \qquad f'(x) = \frac{df}{dx}(x) = \frac{\partial f}{\partial x}(x)$$

$$(n \geq 1) \qquad \nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

What's important? Magnitude? direction?

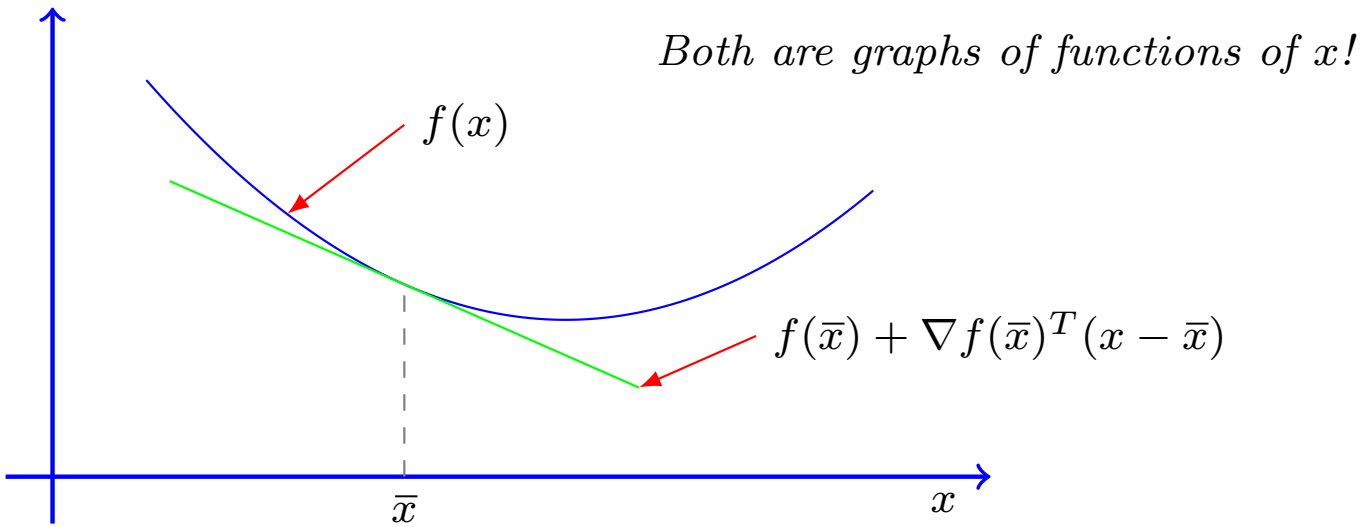
Answer: The gradient is a vector in $\mathbb{R}^n$, which

- has magnitude (e.g., its 2-norm)
- can be viewed as a direction
- and gives us a way to compute *directional derivatives*

Differentiable f

How should we think about the gradient?

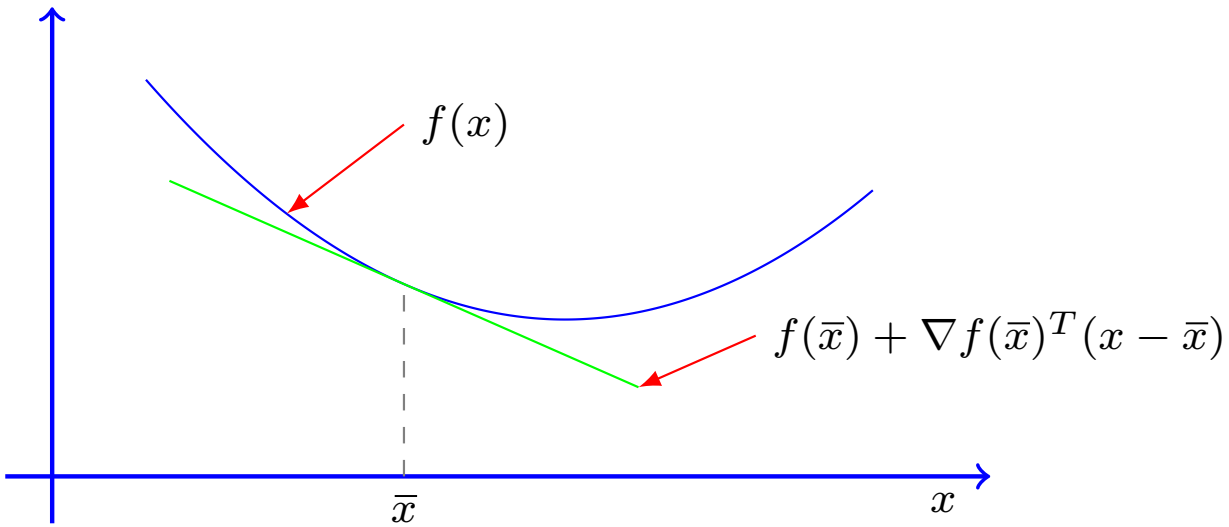
If f is continuously differentiable (i.e., $f \in \mathcal{C}^1$),
then $\nabla f(\bar{x})$ is the unique vector in the linear (Taylor) approximation of f at $\bar{x}$.



Differentiable and convex f

If $f \in \mathcal{C}^1$ is convex, then

$$f(x) \geq f(\overline{x}) + \nabla f(\overline{x})^T (x - \overline{x}) \quad \text{for all} \quad (x, \overline{x}) \in \mathbb{R}^n \times \mathbb{R}^n$$



Graphs and epigraphs

There is another interpretation of a gradient that is also useful. First...

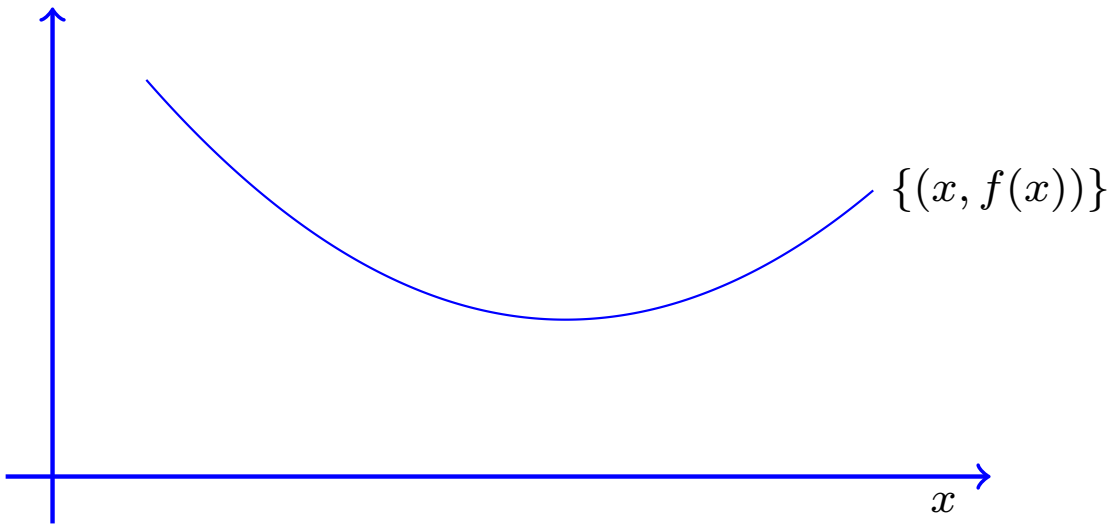
What is a graph?

Graphs and epigraphs

There is another interpretation of a gradient that is also useful. First...

What is a graph?

A *set* of points in $\mathbb{R}^{n+1}$, namely, $\{(x, z) : f(x) = z\}$



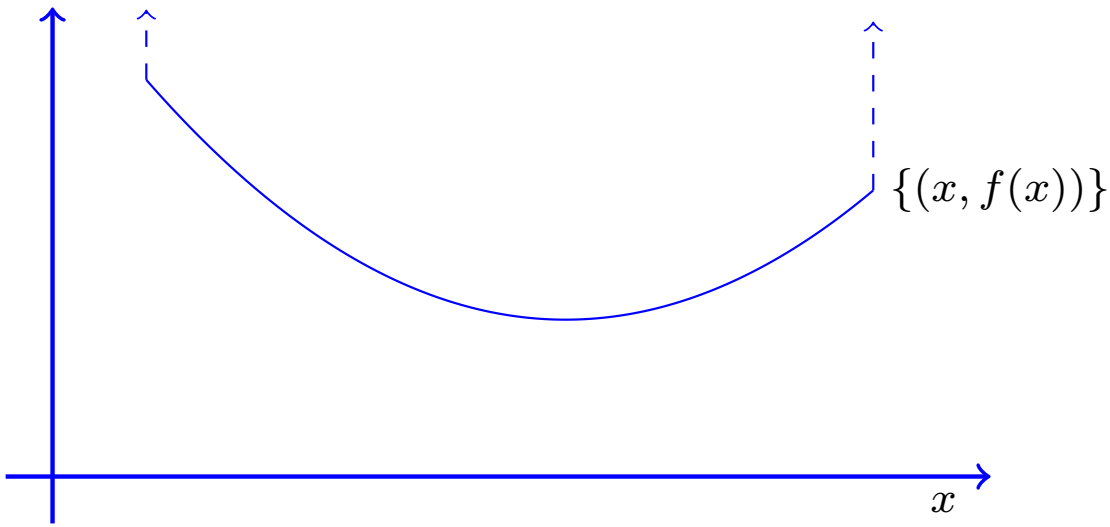
Graphs and epigraphs

There is another interpretation of a gradient that is also useful. First...

What is a graph?

A *set* of points in $\mathbb{R}^{n+1}$, namely, $\{(x, z) : f(x) = z\}$

A related quantity, another *set*, is the **epigraph**: $\{(x, z) : f(x) \leq z\}$



Differentiable and convex f

If $f \in \mathcal{C}^1$ is convex, then, for all $(x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n$,

$$
\begin{aligned}
& f(x) \geq f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) \\
\iff \quad & f(x) - \nabla f(\bar{x})^T x \geq f(\bar{x}) - \nabla f(\bar{x})^T \bar{x} \\
\iff \quad & \begin{bmatrix} -\nabla f(\bar{x}) \\ 1 \end{bmatrix}^T \begin{bmatrix} x \\ f(x) \end{bmatrix} \geq \begin{bmatrix} -\nabla f(\bar{x}) \\ 1 \end{bmatrix}^T \begin{bmatrix} \bar{x} \\ f(\bar{x}) \end{bmatrix}
\end{aligned}
$$

Differentiable and convex f

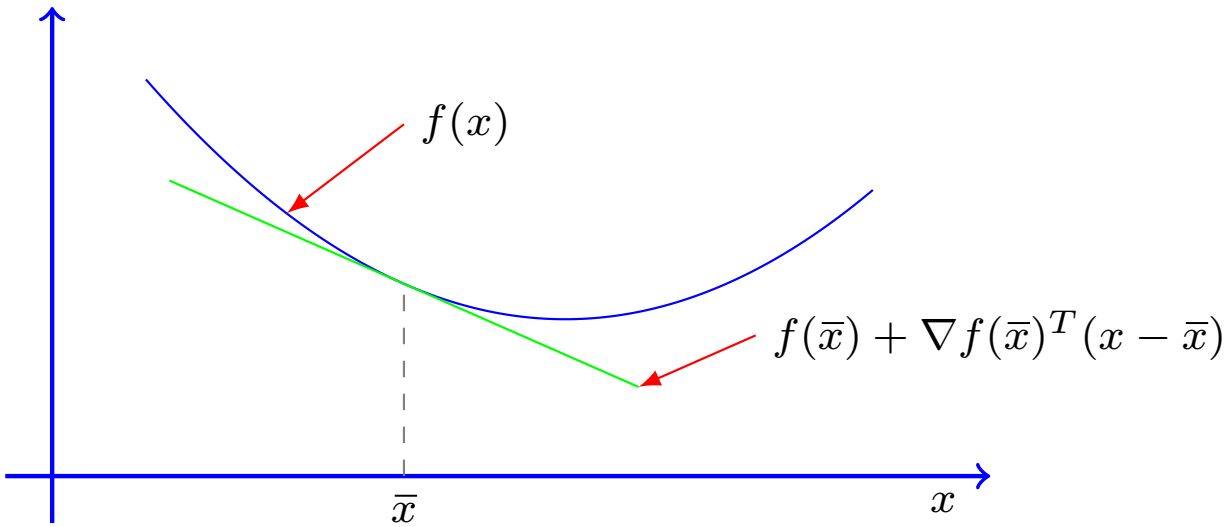
If $f \in \mathcal{C}^1$ is convex, then, for all $(x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n$,

$$\begin{aligned} & f(x) \geq f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) \\ \iff \quad & f(x) - \nabla f(\bar{x})^T x \geq f(\bar{x}) - \nabla f(\bar{x})^T \bar{x} \\ \iff \quad & \begin{bmatrix} -\nabla f(\bar{x}) \\ 1 \end{bmatrix}^T \begin{bmatrix} x \\ f(x) \end{bmatrix} \geq \begin{bmatrix} -\nabla f(\bar{x}) \\ 1 \end{bmatrix}^T \begin{bmatrix} \bar{x} \\ f(\bar{x}) \end{bmatrix} \end{aligned}$$

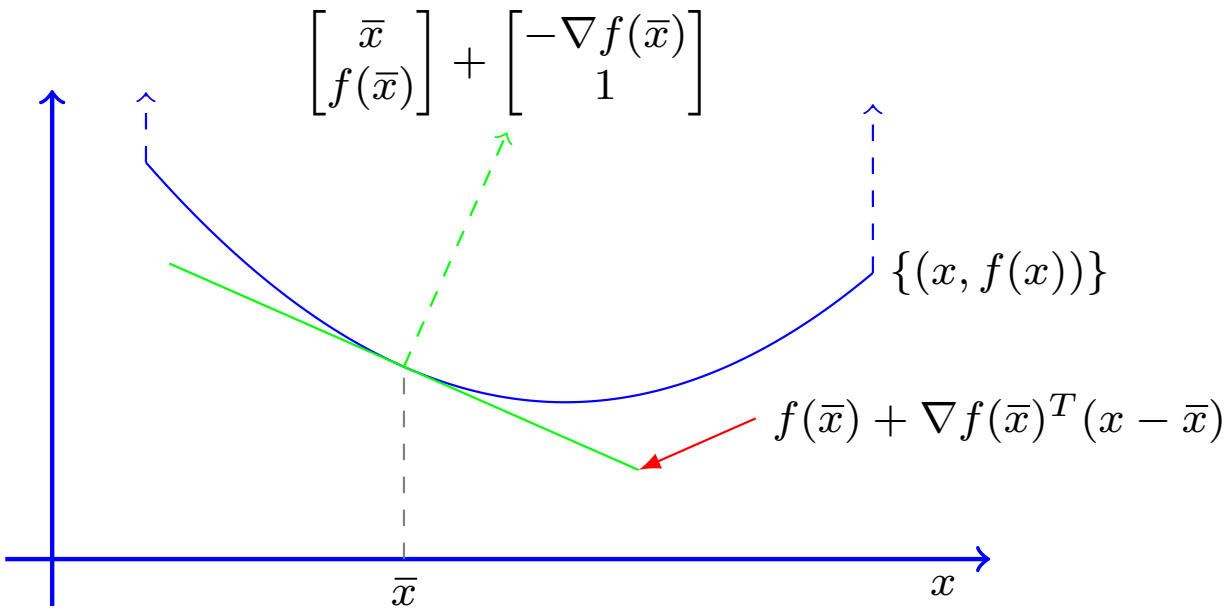
Note: Given $\bar{x}$, the vector $\begin{bmatrix} -\nabla f(\bar{x}) \\ 1 \end{bmatrix}$ is given,

so the inequality above involves a *linear function* over $\mathbb{R}^{n+1}$ and says the value at any point $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ in the graph is at least the value at $\begin{bmatrix} \bar{x} \\ f(\bar{x}) \end{bmatrix}$

Linearization



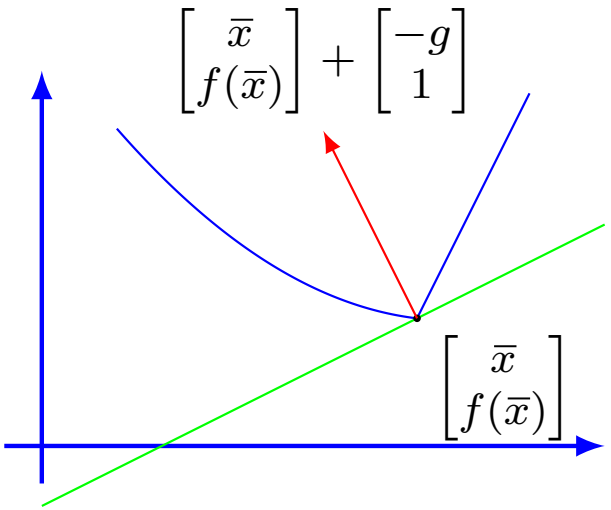
Linearization and supporting hyperplane for epigraph



Subgradients (convex f)

Why was that useful?

We can generalize this idea when the function is not differentiable somewhere.



A vector $g \in \mathbb{R}^n$ is a subgradient of a convex $f : \mathbb{R}^n \to \mathbb{R}$ at $\bar{x} \in \mathbb{R}^n$ if

$$f(x) \geq f(\bar{x}) + g^T(x - \bar{x})$$

$$\iff \quad \begin{bmatrix} -g \\ 1 \end{bmatrix}^T \begin{bmatrix} x \\ f(x) \end{bmatrix} \geq \begin{bmatrix} -g \\ 1 \end{bmatrix}^T \begin{bmatrix} \bar{x} \\ f(\bar{x}) \end{bmatrix}$$

Subdifferentials

Theorem

If f is convex and differentiable at x, then $\nabla f(x)$ is its unique subgradient at x.

But in general,
the set of all subgradients for a convex f at x is the *subdifferential* of f at x:

$$\partial f(x) := \{g \in \mathbb{R}^n : g \text{ is a subgradient of } f \text{ at } x\}.$$

From the definition, it is easily seen that

$$x_* \text{ is a minimizer of } f \text{ if and only if } 0 \in \partial f(x_*)$$

What about nonconvex, nonsmooth?

We need to generalize the idea of a subgradient further.

- Directional derivatives
- Subgradients
- Subdifferentials

Let's return to this after we discuss some algorithms...

Outline

A fundamental iteration

Thinking of $-\nabla f(x_k)$, we have a vector that

- directs us in a direction of descent, and
- vanishes as we approach a minimizer

A fundamental iteration

Thinking of $-\nabla f(x_k)$, we have a vector that

- directs us in a direction of descent, and
- vanishes as we approach a minimizer

Algorithm : Gradient Descent

1: Choose an initial point $x_0 \in \mathbb{R}^n$ and stepsize $\alpha \in (0, 1/L]$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: **if** $\|\nabla f(x_k)\| \approx 0$, **then return** x_k
4: **else** set

$$x_{k+1} \leftarrow x_k - \alpha \nabla f(x_k)$$

I call this a fundamental iteration.

A fundamental iteration

Thinking of $-\nabla f(x_k)$, we have a vector that

- directs us in a direction of descent, and
- vanishes as we approach a minimizer

Algorithm : Gradient Descent

1: Choose an initial point $x_0 \in \mathbb{R}^n$ and stepsize $\alpha \in (0, 1/L]$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: **if** $\|\nabla f(x_k)\| \approx 0$, **then return** x_k
4: **else** set

$$x_{k+1} \leftarrow x_k - \alpha \nabla f(x_k)$$

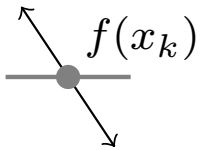
I call this a fundamental iteration.

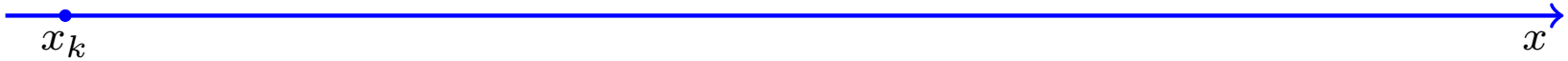
Here, we suppose ∇f is Lipschitz continuous, i.e., there exists $L \geq 0$ such that

$$\|\nabla f(\bar{x}) - \nabla f(x)\|_2 \leq L\|\bar{x} - x\|_2 \quad \text{for all} \quad (x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n$$

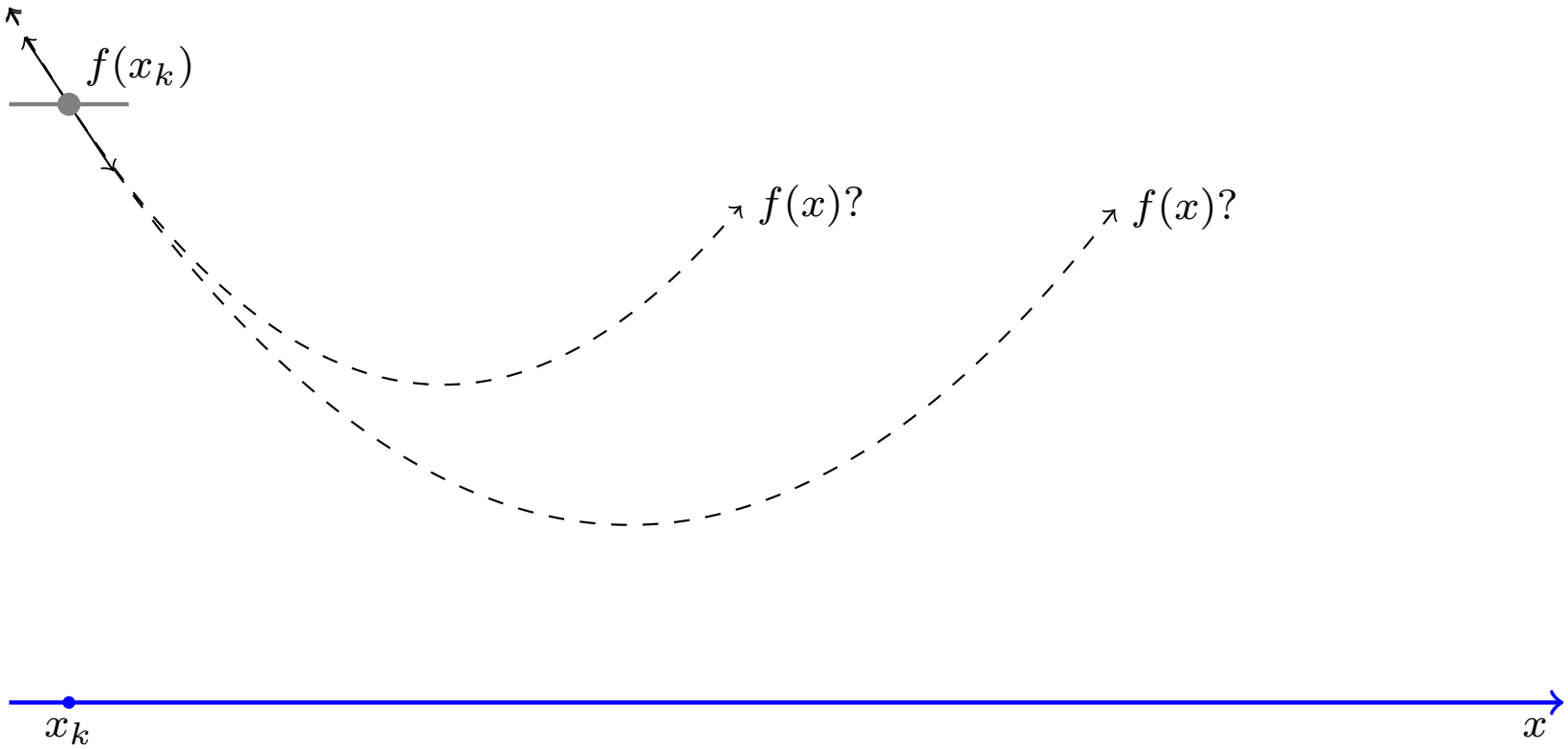
$$\implies \quad f(x) \leq f(\bar{x}) + \nabla f(\bar{x})^T(x - \bar{x}) + \tfrac{1}{2}L\|x - \bar{x}\|_2^2 \quad \text{for all} \quad (x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n.$$

Convergence of gradient descent

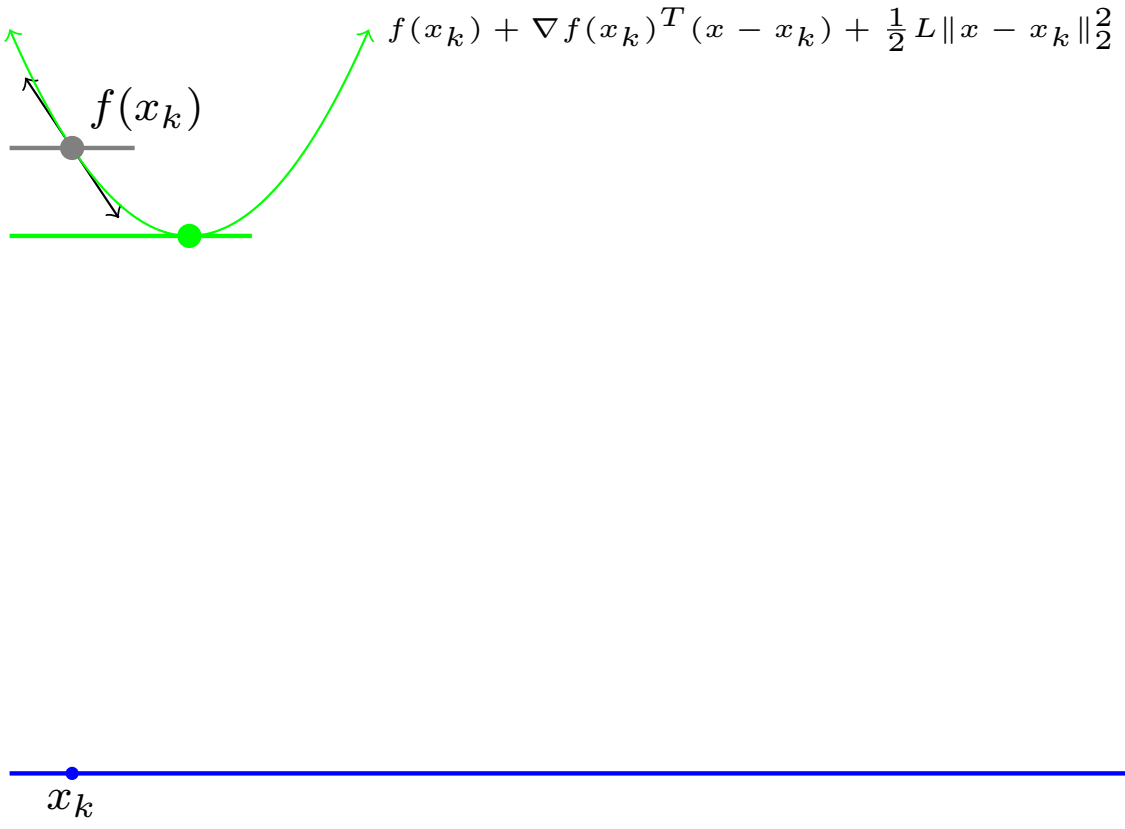




Convergence of gradient descent



Convergence of gradient descent



Gradient descent for f

Theorem

If ∇f is Lipschitz continuous with constant $L > 0$ and $\alpha \in (0, 1/L]$, then

$$\sum_{j=0}^{\infty} \|\nabla f(x_j)\|_2^2 < \infty \quad \textit{which implies} \quad \{\nabla f(x_j)\} \to 0.$$

Gradient descent for f

Theorem

If ∇f is Lipschitz continuous with constant $L > 0$ and $\alpha \in (0, 1/L]$, then

$$\sum_{j=0}^{\infty} \|\nabla f(x_j)\|_2^2 < \infty \quad \textit{which implies} \quad \{\nabla f(x_j)\} \to 0.$$

Proof.

Let $k \in \mathbb{N}$ and recall that $x_{k+1} - x_k = -\alpha \nabla f(x_k)$. Then, since $\alpha \in (0, 1/L]$,

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \tfrac{1}{2} L \|x_{k+1} - x_k\|_2^2 \\ &= f(x_k) - \alpha \|\nabla f(x_k)\|_2^2 + \tfrac{1}{2} \alpha^2 L \|\nabla f(x_k)\|_2^2 \\ &= f(x_k) - \alpha (1 - \tfrac{1}{2} \alpha L) \|\nabla f(x_k)\|_2^2 \\ &\leq f(x_k) - \tfrac{1}{2} \alpha \|\nabla f(x_k)\|_2^2. \end{aligned}$$

Thus, summing over $j \in \{0, \ldots, k\}$, one finds

$$\infty > f(x_0) - f_{\inf} \geq f(x_0) - f(x_{k+1}) \geq \tfrac{1}{2} \alpha \textstyle\sum_{j=0}^{k} \|\nabla f(x_j)\|_2^2.$$

Strong convexity

Now suppose that f is c-strongly convex, which means that

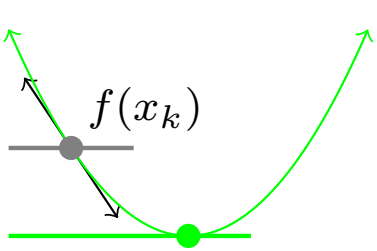
$$f(\bar{x}) \geq f(x) + \nabla f(x)^T(\bar{x} - x) + \tfrac{1}{2}c\|\bar{x} - x\|_2^2 \quad \text{for all} \quad (x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n.$$

Important consequences of this are that

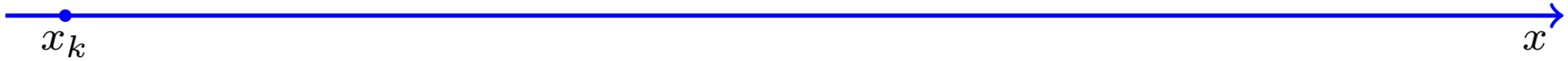
- f has a unique global minimizer, call it x_* with $f_* := f(x_*)$, and
- the gradient norm grows with the optimality error in that

$$2c(f(x) - f_*) \leq \|\nabla f(x)\|_2^2 \quad \text{for all} \quad x \in \mathbb{R}^n.$$

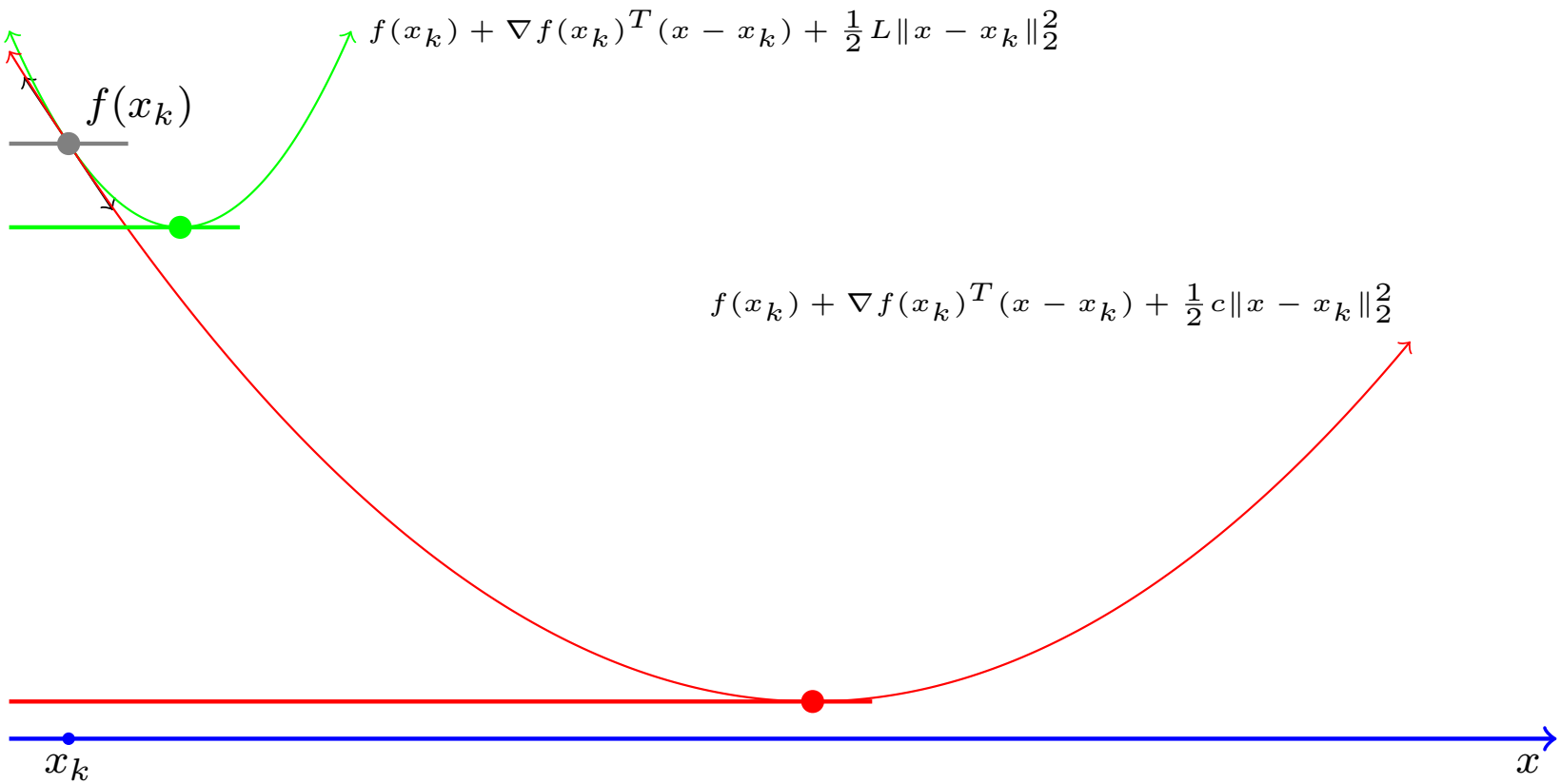
Strong convexity, lower bound



$$f(x_k) + \nabla f(x_k)^T (x - x_k) + \tfrac{1}{2} L \|x - x_k\|_2^2$$



Strong convexity, lower bound



Gradient descent for strongly convex f

Theorem

If ∇f is Lipschitz with $L > 0$, f is c-strongly convex, and $\alpha \in (0, 1/L]$, then

$$f(x_{j+1}) - f_* \leq (1 - \alpha c)^{j+1}(f(x_0) - f_*) \quad \text{for all} \quad j \in \mathbb{N}.$$

Gradient descent for strongly convex f

Theorem

If ∇f is Lipschitz with $L > 0$, f is c-strongly convex, and $\alpha \in (0, 1/L]$, then

$$f(x_{j+1}) - f_* \leq (1 - \alpha c)^{j+1}(f(x_0) - f_*) \quad \text{for all} \quad j \in \mathbb{N}.$$

Proof.

Let $k \in \mathbb{N}$. Following the previous proof, one finds

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - \tfrac{1}{2}\alpha \|\nabla f(x_k)\|_2^2 \\ &\leq f(x_k) - \alpha c(f(x_k) - f_*). \end{aligned}$$

Subtracting f_* from both sides, one finds

$$f(x_{k+1}) - f_* \leq (1 - \alpha c)(f(x_k) - f_*).$$

Applying the result repeatedly over $j \in \{0, \ldots, k\}$ yields the result.

A fundamental iteration when f is nonsmooth?

What is a fundamental iteration for nonsmooth optimization?

A fundamental iteration when f is nonsmooth?

What is a fundamental iteration for nonsmooth optimization?

Steepest descent!

For convex f, the *directional derivative* of f at x along s is

$$f'(x;s) = \max_{g \in \partial f(x)} g^T s$$

Along which direction is f decreasing at the fastest rate?

A fundamental iteration when f is nonsmooth?

What is a fundamental iteration for nonsmooth optimization?

Steepest descent!

For convex f, the *directional derivative* of f at x along s is

$$f'(x;s) = \max_{g \in \partial f(x)} g^T s$$

Along which direction is f decreasing at the fastest rate?

The solution of an optimization problem!

$$\begin{aligned}
\min_{\|s\|_2 \le 1} f'(x;s) &= \min_{\|s\|_2 \le 1} \max_{g \in \partial f(x)} g^T s \\
&= \max_{g \in \partial f(x)} \min_{\|s\|_2 \le 1} g^T s \quad \text{(von Neumann minimax theorem)} \\
&= \max_{g \in \partial f(x)} (-\|g\|_2) \\
&= -\min_{g \in \partial f(x)} \|g\|_2 \quad \Longrightarrow \text{(need minimum norm subgradient)}
\end{aligned}$$

Main challenge

But, typically, we can only access $g \in \partial f(x)$, not all of $\partial f(x)$

I would argue:

no practical fundamental iteration for general nonsmooth optimization

(no computable descent direction that vanishes near a minimizer)

What are our options?

Main challenge

But, typically, we can only access $g \in \partial f(x)$, not all of $\partial f(x)$

I would argue:

no practical fundamental iteration for general nonsmooth optimization

(no computable descent direction that vanishes near a minimizer)

What are our options?

There are a few ways to design a convergent algorithm:

- algorithmically (e.g., subgradient method)
- iteratively (e.g., cutting plane / bundle methods)
- randomly (e.g., gradient sampling)

Subgradient method

Algorithm : Subgradient method (not descent)

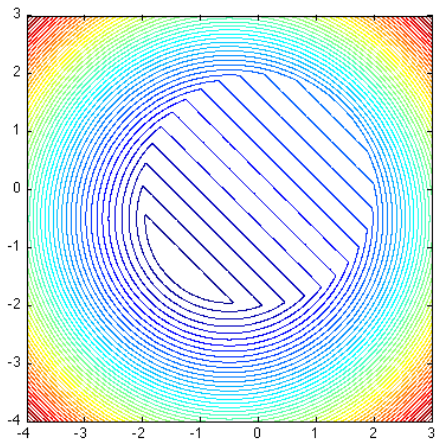
1: Choose an initial point $x_0 \in \mathbb{R}^n$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: **if** a termination condition is satisfied, **then return** x_k
4: **else** compute $g_k \in \partial f(x_k)$, choose $\alpha_k \in \mathbb{R}_{>0}$, and set

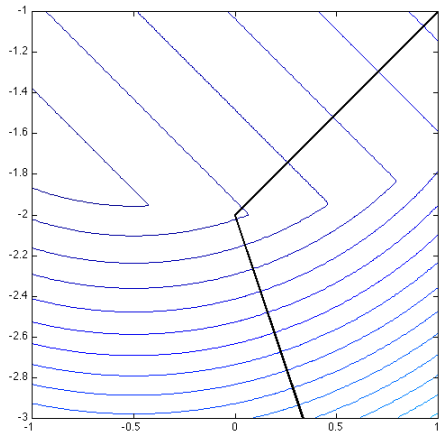
$$x_{k+1} \leftarrow x_k - \alpha_k g_k$$

Why not "subgradient descent"?

Consider

$$\min_{x\in\mathbb{R}^2} \; f(x), \quad \text{where} \quad f(x_1,x_2) := x_1 + x_2 + \max\{0, x_1^2 + x_2^2 - 4\}.$$





At $x = (0,-2)$, we have

$$\partial f(x) = \operatorname{conv}\left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-3\end{bmatrix}\right\}, \quad \text{but} \quad -\begin{bmatrix}1\\1\end{bmatrix} \quad \text{and} \quad -\begin{bmatrix}1\\-3\end{bmatrix}$$

are both directions of ascent for f from x!

Decreasing the distance to a solution

The objective f is not the only measure of progress.

- Given an arbitrary subgradient g_k for f at x_k, we have

$$f(x) \geq f(x_k) + g_k^T(x - x_k) \quad \text{for all} \quad x \in \mathbb{R}^n, \tag{1}$$

  which means that all points with an objective value lower than $f(x_k)$ lie in

$$\mathcal{H}_k := \{x \in \mathbb{R}^n : g_k^T(x - x_k) \leq 0\}$$

- Thus, a small step along $-g_k$ should decrease the distance to a solution
- (Convexity is crucial for this idea)

"Algorithmic convergence"

Theorem

If f has a minimizer, $\|g_k\|_2 \leq G \in \mathbb{R}_{>0}$ for all $k \in \mathbb{N}$, and the stepsizes satisfy

$$\sum_{k=1}^{\infty} \alpha_k = \infty \quad \textit{and} \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty, \tag{2}$$

then

$$\lim_{k \to \infty} \left\{ \min_{j \in \{0,\ldots,k\}} f_j \right\} = f_*.$$

- An example sequence satisfying (2) is $\alpha_k = \alpha/k$ for $k = 1, 2, \ldots$

Proof, $\lim_{k\to\infty}\left\{\min_{j\in\{0,\ldots,k\}} f_j\right\} = f_*$, part 1.

Let $k \in \mathbb{N}$. By (1), the iterates satisfy

$$\begin{aligned}\|x_{k+1} - x_*\|_2^2 &= \|x_k - \alpha_k g_k - x_*\|_2^2 \\ &= \|x_k - x_*\|_2^2 - 2\alpha_k g_k^T(x_k - x_*) + \alpha_k^2\|g_k\|_2^2 \\ &\leq \|x_k - x_*\|_2^2 - 2\alpha_k(f_k - f_*) + \alpha_k^2\|g_k\|_2^2.\end{aligned}$$

Applying this inequality recursively, we have

$$0 \leq \|x_{k+1} - x_*\|_2^2 \leq \|x_0 - x_*\|_2^2 - 2\sum_{j=0}^{k}\alpha_j(f_j - f_*) + \sum_{j=0}^{k}\alpha_j^2\|g_j\|_2^2,$$

which implies that

$$\begin{aligned}2\sum_{j=0}^{k}\alpha_j(f_j - f_*) &\leq \|x_0 - x_*\|_2^2 + \sum_{j=1}^{k}\alpha_j^2\|g_j\|_2^2 \\ \Rightarrow \min_{j\in\{0,\ldots,k\}} f_j - f_* &\leq \frac{\|x_0 - x_*\|_2^2 + G^2\sum_{j=1}^{k}\alpha_j^2}{2\sum_{j=0}^{k}\alpha_j}. \qquad (3)\end{aligned}$$

Proof, $\lim_{k\to\infty}\left\{\min_{j\in\{0,\ldots,k\}} f_j\right\} = f_*$, part 2.

Now consider an arbitrary scalar $\epsilon > 0$. By (2), there exists a nonnegative integer K such that, for all $k > K$,

$$\alpha_k \leq \frac{\epsilon}{G^2} \quad \text{and} \quad \sum_{j=0}^{k} \alpha_j \geq \frac{1}{\epsilon}\left(\|x_0 - x_*\|_2^2 + G^2 \sum_{j=0}^{K} \alpha_j^2\right).$$

Then, by (3), it follows that for all $k > K$ we have

$$\begin{aligned}
\min_{j\in\{0,\ldots,k\}} f_j - f_* &\leq \frac{\|x_0 - x_*\|_2^2 + G^2 \sum_{j=0}^{K} \alpha_j^2}{2\sum_{j=0}^{k} \alpha_j} + \frac{G^2 \sum_{j=K+1}^{k} \alpha_j^2}{2\sum_{j=0}^{K} \alpha_j + 2\sum_{j=K+1}^{k} \alpha_j} \\
&\leq \frac{\|x_0 - x_*\|_2^2 + G^2 \sum_{j=0}^{K} \alpha_j^2}{\frac{2}{\epsilon}\left(\|x_0 - x_*\|_2^2 + G^2 \sum_{j=0}^{K} \alpha_j^2\right)} + \frac{G^2 \sum_{j=K+1}^{k} \frac{\epsilon}{G^2}\alpha_j}{2\sum_{j=K+1}^{k} \alpha_j} \\
&= \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.
\end{aligned}$$

The result follows since $\epsilon > 0$ was chosen arbitrarily.

Cutting plane method

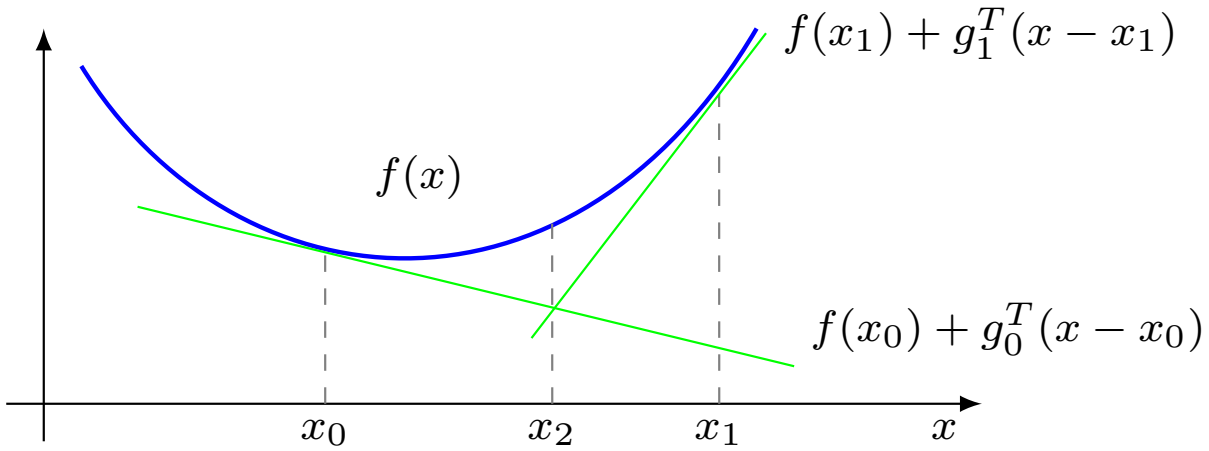
Subgradient methods lose previously computed information in every iteration.

Cutting plane method

Subgradient methods lose previously computed information in every iteration.

- Suppose, after a sequence of iterates, we have the affine underestimators

$$f_i(x) = f(x_i) + g_i^T(x - x_i) \text{ for all } i \in \{0, \ldots, k\}.$$



- At iteration k, we can compute the next iterate by solving the master problem

$$x_{k+1} \leftarrow \arg\min_{x \in \mathcal{X}} \hat{f}_k(x), \quad \text{where} \quad \hat{f}_k(x) := \max_{i \in \{1,\ldots,k\}} (f(x_i) + g_i^T(x - x_i)).$$

Cutting plane method convergence

The iterates of the cutting plane method yield lower bounds of the optimal value:

$$v_{k+1} := \min_{x \in \mathcal{X}} \hat{f}_k(x) \leq \min_{x \in \mathcal{X}} f(x) =: f_*.$$

Therefore, if $v_{k+1} = f(x_{k+1})$, then we terminate since $f(x_{k+1}) = f_*$.

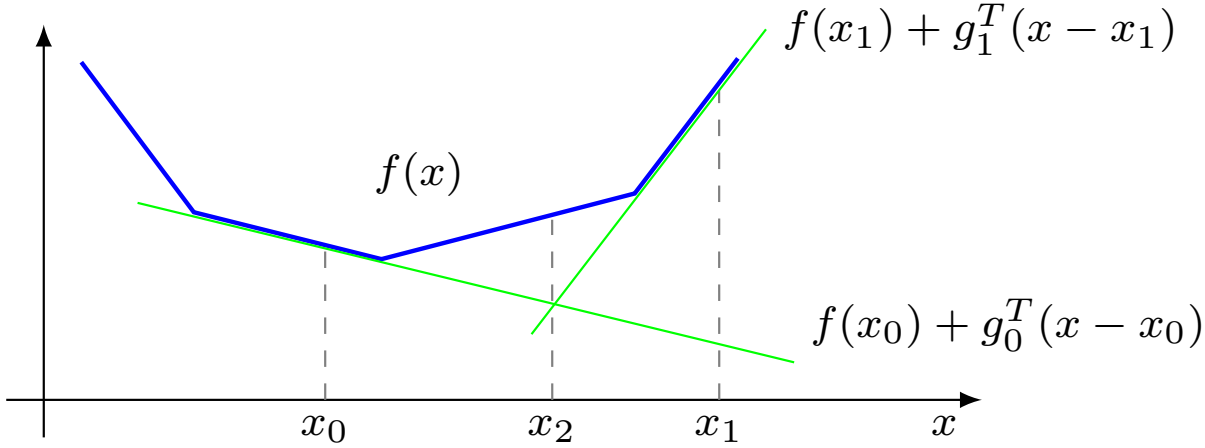
Cutting plane method convergence

The iterates of the cutting plane method yield lower bounds of the optimal value:

$$v_{k+1} := \min_{x \in \mathcal{X}} \hat{f}_k(x) \leq \min_{x \in \mathcal{X}} f(x) =: f_*.$$

Therefore, if $v_{k+1} = f(x_{k+1})$, then we terminate since $f(x_{k+1}) = f_*$.

- If f is piecewise linear, then convergence occurs in finitely many iterations!



However, in general, we have the following theorem.

Theorem

The cutting plane method yields $\{x_k\}$ satisfying $\{f(x_k)\} \to f_*$.

Bundle method

A bundle method attempts to combine the practical advantages of a cutting plane method with the theoretical strengths of a proximal point method.

- Given x_k, consider the regularized master problem

$$\min_{x\in\mathbb{R}^n} \left(\hat{f}_k(x) + \frac{\gamma}{2}\|x - x_k\|_2^2\right), \quad \text{where} \quad \hat{f}_k(x) := \max_{i\in\mathcal{I}_k}(f(x_i) + g_i^T(x - x_i)).$$

  Here, $\mathcal{I}_k \subseteq \{1, \ldots, k-1\}$ indicates a subset of previous iterations.

- This problem is equivalent to the quadratic optimization problem

$$\begin{aligned} \min_{(x,v)\in\mathbb{R}^n\times\mathbb{R}} \quad & v + \frac{\gamma}{2}\|x - x_k\|_2^2 \\ \text{s.t.} \quad & f(x_i) + g_i^T(x - x_i) \le v \quad \text{for all} \quad i \in \mathcal{I}_k. \end{aligned}$$

- Only move to a "new" point when a sufficient decrease is obtained.

Convergence rate analyses are limited; $\mathcal{O}(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ for strongly convex f

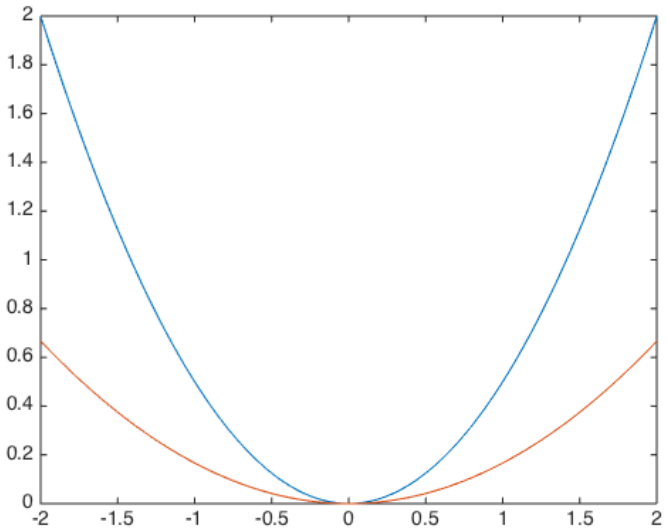
Bundle method convergence

Analysis makes use of the Moreau-Yosida regularization function

$$f_\gamma(\bar{x}) = \min_{x \in \mathbb{R}^n} \left(f(x) + \tfrac{1}{2}\gamma \|x - \bar{x}\|_2^2 \right).$$

Theorem

If x_k is not a minimizer, then $f_\gamma(x_k) < f(x_k)$.

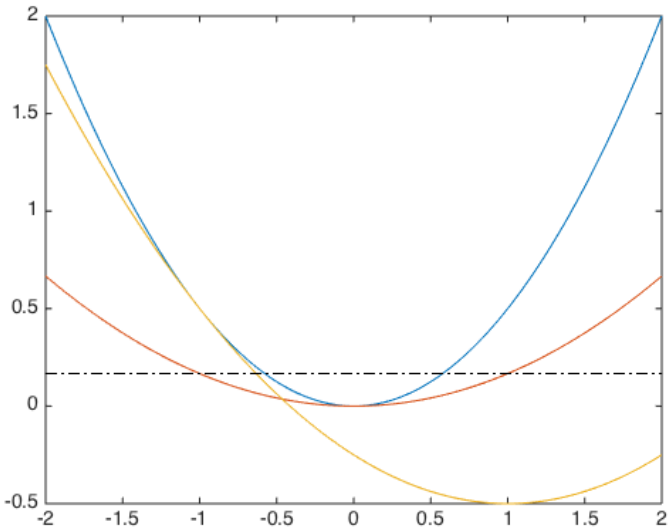


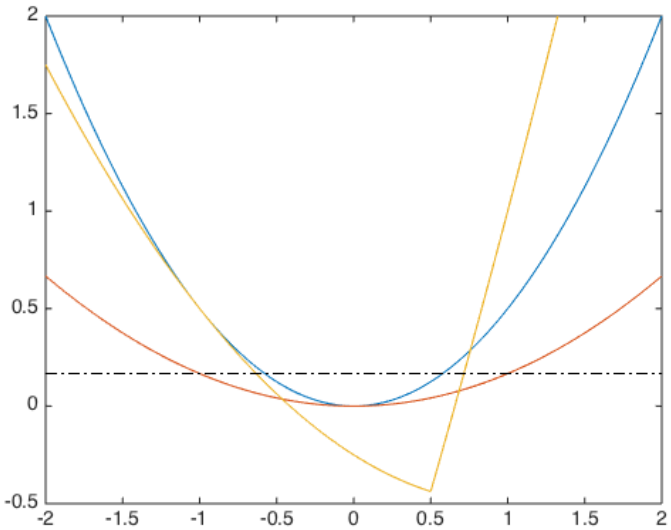
Bundle method convergence

Theorem

For all $(k, j) \in \mathbb{N} \times \mathbb{N}$ *in a bundle method,*

$$v_{k,j} + \tfrac{1}{2}\gamma\|x_{k,j} - x_k\|_2^2 \leq f_\gamma(x_k) < f(x_k).$$





Outline

Clarke subdifferential

What if f is nonconvex and nonsmooth? What are subgradients?

We still need *some* structure; we assume

- f is locally Lipschitz and
- f is differentiable on a full measure set $\mathcal{D}$

Clarke subdifferential

What if f is nonconvex and nonsmooth? What are subgradients?

We still need *some* structure; we assume

- f is locally Lipschitz and
- f is differentiable on a full measure set $\mathcal{D}$

The Clarke subdifferential of f at x is

$$\partial f(x) = \operatorname{conv}\left\{ \lim_{j\to\infty} \nabla f(x_j) : x_j \to x \text{ and } x_j \in \mathcal{D} \right\},$$

i.e., convex hull of limits of gradients of f at points in $\mathcal{D}$ converging to x

Clarke subdifferential

What if f is nonconvex and nonsmooth? What are subgradients?

We still need *some* structure; we assume

- f is locally Lipschitz and
- f is differentiable on a full measure set $\mathcal{D}$

The Clarke subdifferential of f at x is

$$\partial f(x) = \operatorname{conv}\left\{\lim_{j\to\infty} \nabla f(x_j) : x_j \to x \text{ and } x_j \in \mathcal{D}\right\},$$

i.e., convex hull of limits of gradients of f at points in $\mathcal{D}$ converging to x

Theorem

If f is continuously differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$

Differentiable, but nonsmooth

Theorem

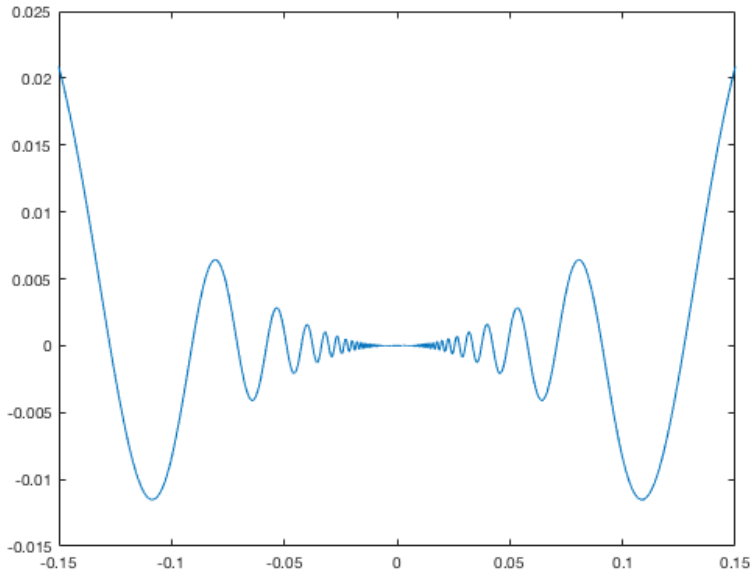
If f is differentiable at x, then $\{\nabla f(x)\} \subseteq \partial f(x)$ (not necessarily equal)

Considering

$$f(x) = \begin{cases} x^2 \cos(\frac{1}{x}) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

one finds that

$$f'(0) = 0$$
$$\text{yet} \quad [-1, 1] \subseteq \partial f(0)$$



Clarke ϵ-subdifferential

As before, we typically cannot compute $\partial f(x)$.

It is approximated by the Clarke ϵ-subdifferential, namely,

$$\partial_\epsilon f(x) = \text{conv}\{\partial f(\mathbb{B}(x, \epsilon))\},$$

which in turn can be approximated as in

$$\partial_\epsilon f(x) \approx \text{conv}\{\nabla f(x_k), \nabla f(x_{k,1}), \ldots, \nabla f(x_{k,m})\},$$
$$\text{where } \{x_{k,1}, \ldots, x_{k,m}\} \subset \mathbb{B}(x_k, \epsilon).$$

Clarke ϵ-subdifferential and gradient sampling

As before, we typically cannot compute $\partial f(x)$.

It is approximated by the Clarke ϵ-subdifferential, namely,

$$\partial_\epsilon f(x) = \text{conv}\{\partial f(\mathbb{B}(x, \epsilon))\},$$

which in turn can be approximated as in

$$\partial_\epsilon f(x) \approx \text{conv}\{\nabla f(x_k), \nabla f(x_{k,1}), \ldots, \nabla f(x_{k,m})\},$$
$$\text{where } \{x_{k,1}, \ldots, x_{k,m}\} \subset \mathbb{B}(x_k, \epsilon).$$

In *gradient sampling*, we compute the minimum norm element in

$$\text{conv}\{\nabla f(x_k), \nabla f(x_{k,1}), \ldots, \nabla f(x_{k,m})\},$$

which is equivalent to solving

$$\min_{(x,v)\in\mathbb{R}^n\times\mathbb{R}} \; v + \|x - x_k\|_2^2$$
$$\text{s.t. } f(x_k) + \nabla f(x_{k,i})^T(x - x_k) \le v \;\text{ for all }\; i \in \{1, \ldots, m\}$$

Outline

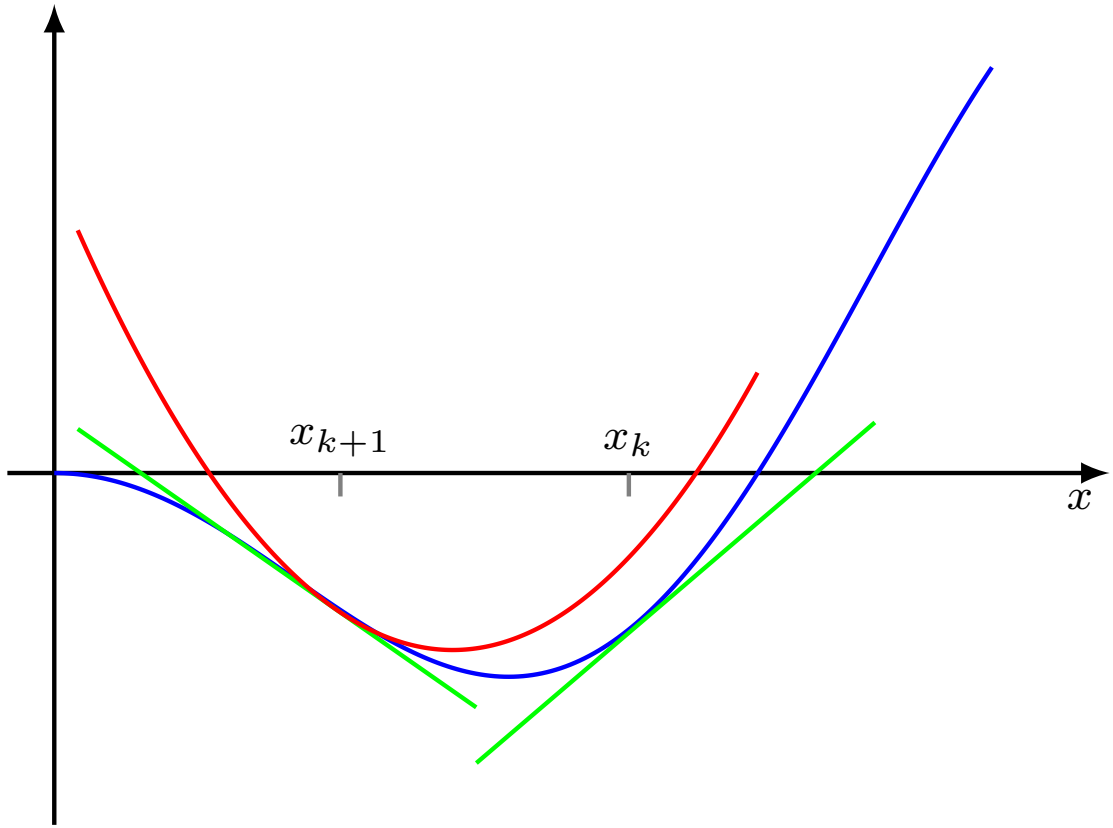
Popular and effective method

Despite all I've talked about, a very effective method: **BFGS**

Popular and effective method

Despite all I've talked about, a very effective method: **BFGS**

Approximate second-order information with gradient displacements:



Secant equation $H_k y_k = s_k$ to match gradient of f at x_k, where

$$s_k := x_{k+1} - x_k \quad \text{and} \quad y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$$

BFGS-type updates

Inverse Hessian and Hessian approximation updating formulas ($s_k^T v_k > 0$):

$$W_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T W_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}$$

$$H_{k+1} \leftarrow \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right)^T H_k \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right) + \frac{v_k v_k^T}{s_k^T v_k}$$

With an appropriate technique for choosing v_k, we attain

- *self-correcting* properties for $\{H_k\}$ and $\{W_k\}$
- (inverse) Hessian approximations that can be used in other algorithms

Subproblems in nonsmooth optimization algorithms

With sets of points, scalars, and (sub)gradients

$$\{x_{k,j}\}_{j=1}^m, \quad \{f_{k,j}\}_{j=1}^m, \quad \{g_{k,j}\}_{j=1}^m,$$

nonsmooth optimization methods involve the primal subproblem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & \left(\max_{j\in\{1,\ldots,m\}} \{f_{k,j} + g_{k,j}^T(x - x_{k,j})\} + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \right) \\ \text{s.t. } & \|x - x_k\| \leq \delta_k, \end{aligned} \tag{P}$$

but, with $G_k \leftarrow [g_{k,1} \ \cdots \ g_{k,m}]$, it is typically more efficient to solve the dual

$$\begin{aligned} \sup_{(\omega,\gamma)\in\mathbb{R}^m_+\times\mathbb{R}^n} \ & -\tfrac{1}{2}(G_k\omega + \gamma)^T W_k (G_k\omega + \gamma) + b_k^T\omega - \delta_k\|\gamma\|_* \\ \text{s.t. } & \mathbb{1}_m^T\omega = 1. \end{aligned} \tag{D}$$

The primal solution can then be recovered by

$$x_k^* \leftarrow x_k - W_k \underbrace{(G_k\omega_k + \gamma_k)}_{\tilde{g}_k}.$$

Algorithm Self-Correcting Variable-Metric Alg. for Nonsmooth Opt.

1: Choose $x_1 \in \mathbb{R}^n$.
2: Choose a symmetric positive definite $W_1 \in \mathbb{R}^{n \times n}$.
3: Choose $\alpha \in (0, 1)$
4: **for** $k = 1, 2, \ldots$ **do**
5: Solve (P)–(D) such that setting

$$\begin{aligned} G_k &\leftarrow \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix}, \\ s_k &\leftarrow -W_k(G_k\omega_k + \gamma_k), \\ \text{and} \quad x_{k+1} &\leftarrow x_k + s_k \end{aligned}$$

6: yields

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2}\alpha(G_k\omega_k + \gamma_k)^T W_k(G_k\omega_k + \gamma_k).$$

7: Choose v_k (details omitted, but very simple)
8: Set

$$W_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T W_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}.$$

Instances of the framework

Cutting plane / bundle methods

- Points added incrementally until sufficient decrease obtained
- Finite number of additions until accepted step

Gradient sampling methods

- Points added randomly / incrementally until sufficient decrease obtained
- Sufficient number of iterations with "good" steps

In any case: convergence guarantees require $\{W_k\}$ to be uniformly positive definite and bounded *on a sufficient number of accepted steps*

C++ implementation: NonOpt

BFGS w/ weak Wolfe line search							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+2.26e-07	450	1017	452	451
mxhilb	Stepsize	+3.13e-03	+9.26e-02	101	1886	113	102
chained lq	Stepsize	+5.00e-02	-6.93e+01	205	4754	207	206
chained cb3 1	Stepsize	+1.00e-01	+9.80e+01	347	7469	348	348
chained cb3 2	Stepsize	+1.00e-01	+9.80e+01	64	1496	69	65
active faces	Stepsize	+2.50e-02	+2.22e-16	24	672	27	25
brown function 2	Stepsize	+1.00e-01	+2.04e-05	395	17259	396	396
chained mifflin 2	Stepsize	+5.00e-02	-3.47e+01	476	10808	508	477
chained crescent 1	Stepsize	+1.00e-01	+2.18e-01	74	2278	91	75
chained crescent 2	Stepsize	+1.00e-01	+5.86e-02	313	7585	334	314

Bundle method with self-correcting properties							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+1.04e-06	193	441	635	440
mxhilb	Stationary	+9.77e-05	+2.25e-05	39	338	351	137
chained lq	Stationary	+9.77e-05	-6.93e+01	29	374	398	366
chained cb3 1	Stationary	+9.77e-05	+9.80e+01	50	1038	1069	1017
chained cb3 2	Stationary	+9.77e-05	+9.80e+01	29	174	204	173
active faces	Stationary	+9.77e-05	+2.09e-02	17	387	165	32
brown function 2	Stationary	+9.77e-05	+2.49e-03	232	10094	9674	9438
chained mifflin 2	Stationary	+9.77e-05	-3.48e+01	393	24410	19493	18924
chained crescent 1	Stationary	+9.77e-05	+2.73e-04	30	66	92	59
chained crescent 2	Stationary	+9.77e-05	+4.36e-05	137	6679	6140	5997

Thanks!

`NonOpt` coming soon...

- Andreas could finish in a day...
- ...what has taken me 6 months on sabbatical, so
- it'll be done when he has a free day ;-)

Thanks for listening!