# Bound Constrained Optimization

## GIAN Short Course on Optimization: Applications, Algorithms, and Computation

Sven Leyffer

Argonne National Laboratory

September 12-24, 2016

# Outline

# Introduction to Bound Constraints

Motivation for Bound-Constrained Optimization

- Practical problems involve variables that must satisfy bounds
  e.g. pressure, temperature, ...
- General optimization requires bound-constrained subproblems
  e.g. trust-region subproblem

**Bound-Constrained Optimization**

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to } l \leq x \leq u$$

where $f : \mathbb{R}^n \to \mathbb{R}$ twice continuously differentiable,
and bounds $l, u \in \mathbb{R}^n$ can be infinite.

- Review optimality conditions ... preview KKT conditions
- Introduce gradient-projection methods for large-scale problems

# Outline

# Optimality Conditions for Bound-Constraints

Consider

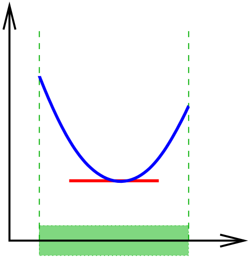
$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to } l \leq x \leq u$$

Look at components $x_i$ to derive optimality conditions (3 cases)

Case I: $l_i < x_i < u_i$ Inactive bounds

Unconstrained Case:

Recall stationarity: $\frac{\partial f}{\partial x_i} = 0$
... i.e. zero gradient in $x_i$



# Optimality Conditions for Bound-Constraints

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to } l \le x \le u$$

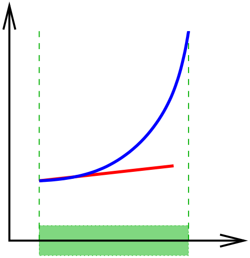
Look at components $x_i$ to derive optimality conditions (3 cases)

Case II: $l_i = x_i$ Lower bound active

Slope of $f$ in direction $e_i$ should be $\geq 0$
... otherwise reduce $f$ by moving away from $l_i$

Lower Bound: $\frac{\partial f}{\partial x_i} \geq 0$ and $x_i = l_i$

... $e_i$ is $i^{th}$ unit vector



# Optimality Conditions for Bound-Constraints

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to } l \le x \le u$$

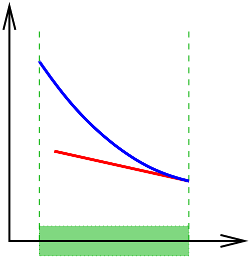
Look at components $x_i$ to derive optimality conditions (3 cases)

Case III: $x_i = u_i$ Upper bound active

Slope of $f$ in direction $-e_i$ should be $\geq 0$
... otherwise reduce $f$ by moving away from $u_i$

Upper Bound: $\frac{\partial f}{\partial x_i} \leq 0$ and $x_i = u_i$

... $e_i$ is $i^{th}$ unit vector.



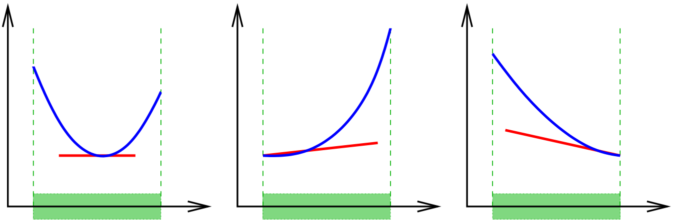
# Optimality Conditions for Bound-Constraints

Optimality conditions are related to sign condition on multipliers in KKT conditions (tomorrow's lecture)

## Theorem (Optimality Conditions for Bound Constraints)

*Let $f(x)$ be continuously differentiable. If $x^*$ local minimizer of*

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\textit{minimize}} & f(x) \\ \textit{subject to} & l \leq x \leq u \end{array} \quad \textit{then} \quad \frac{\partial f}{\partial x_i}(x^*) \begin{cases} \geq 0, & \textit{if } x_i^* = l_i \\ = 0, & \textit{if } l_i < x_i^* < u_i \\ \leq 0, & \textit{if } x_i^* = u_i. \end{cases}$$



# Projection Operator for Bound Constraints

Projection operator, $P_{[l,u]}(x)$, projects $x$ into box, $[l, u]$:

$$\left[P_{[l,u]}(x)\right]_i := \begin{cases} l_i, & \text{if } x_i \leq l_i \\ x_i, & \text{if } l_i < x_i < u_i \\ u_i, & \text{if } x_i \geq u_i. \end{cases}$$

We can restate first-order conditions equivalently as follows.

**Corollary (First-Order Conditions for Bound Constraints)**

*Let $f(x)$ be continuously differentiable. If $x^*$ local minimizer, then*

$$x^* = P_{[l,u]}\left(x^* - \nabla f(x^*)\right).$$

**Proof.** See Exercise this afternoon.

# Active Sets

Active sets play important role in general constrained optimization.

**Definition (Active Set)**

Set of *active constraints*: constraints that hold with equality at $\hat{x}$:

$$\mathcal{A}(\hat{x}) := \{i : l_i = \hat{x}_i\} \cup \{-i : u_i = \hat{x}_i\},$$

Convention: positive $i$ for lower, negative $i$ for upper bounds

- Sign convention is not needed, if at most one bound finite
- Sign convention mimics sign of gradient at stationary point

Next derive active-set algorithm for quadratics, then generalize it.

# Outline

# Bound-Constrained Quadratic Optimization

Bound constrained quadratic program (QP)

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; q(x) = b^T x + \frac{1}{2} x^T G x \quad \text{subject to } l \leq x \leq u$$

where $b \in \mathbb{R}^n$, and $G \in \mathbb{R}^{n \times n}$ is symmetric

- Do not assume $G$ positive definite ... seek local minimum
- Instead assume all bounds finite, $l > -\infty$ and $u < \infty$

$\Rightarrow$ stationary point exists ... unbounded case handled easily.

## Main Idea Algorithm

- Take projected-gradient step to identify (optimal) face
- Perform local optimization on face of hyper cube

Projected-gradient along steepest descend $\Rightarrow$ convergence

# Projected-Gradient Step

Bound constrained quadratic program (QP)

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; q(x) = b^T x + \frac{1}{2} x^T G x \quad \text{subject to } l \le x \le u$$

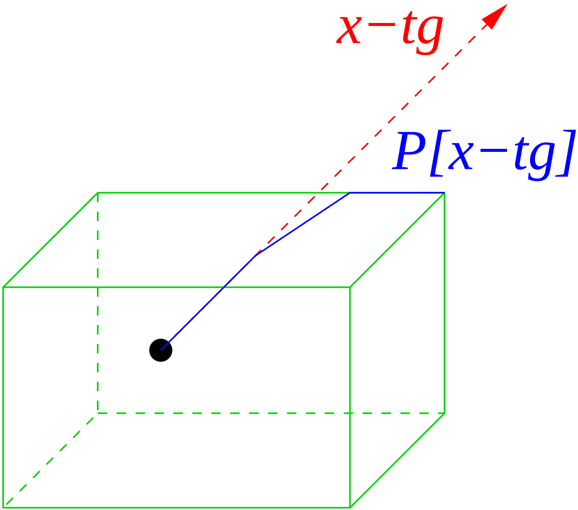
Given feasible point, x, with $l \le x \le u$, and gradient, $g = Gx + b$,
... consider piecewise linear path parameterized in $t$:

$$x(t) := P_{[l,u]}\left(x - tg\right),$$

Goal: Find first minimizers of $q(x)$ along this path
$\Leftrightarrow$ find first minimizer of $q(x(t))$

- Construct analytic description of piecewise linear path, $x(t)$
- Find first minimizer along this path

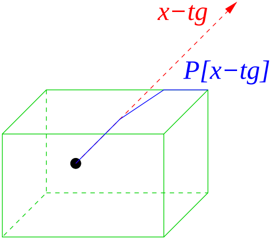
# Projected Gradient Path



# Construction of Projected Gradient Path

$\hat{t}_i$: value of $t$ when component $i$ reaches bound in direction $-g$:

$$\hat{t}_i = \begin{cases} (x_i - u_i)/g_i & \text{if } g_i < 0, \text{ and } u_i < \infty \\ (x_i - l_i)/g_i & \text{if } g_i > 0, \text{ and } l_i > -\infty \\ \infty & \text{otherwise.} \end{cases}$$




NB: if $g_i = 0$, then $x_i$ unchanged, i.e. $\hat{t}_i = \infty$

To describe path $x(t)$, must identify breakpoints along $x(t)$:

$$x_i(t) = \begin{cases} x_i - t g_i & \text{if } t \leq \hat{t}_i \\ x_i - \hat{t}_i g_i & \text{if } t \geq \hat{t}_i, \end{cases}$$

i.e. once component *i*s its bound at $\hat{t}_i$ it does not change

Identify breakpoints of $x(t)$ by ordering the $\hat{t}_i$ increasingly
$\Rightarrow$ get sequence, $0 < t_1 < t_2 < t_3 \ldots$

# Construction of Projected Gradient Path

Intervals $[0, t_1], [t_1, t_2], [t_2, t_3], \ldots$ correspond to segments of $x(t)$

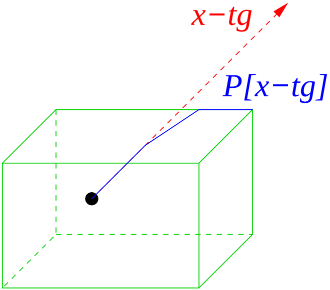
Expression of the $j^{th}$ segment, $[t_{j-1}, t_j]$

$$x(t) = x(t_{j-1}) + \delta s^{(j-1)},$$

where stepsize $\delta$ and direction $s^{(j-1)}$ are:

$$\delta = t - t_{j-1}, \quad \delta \in [0, t_j - t_{j-1}],$$

$$s_i^{(j-1)} = \begin{cases} -g_i \text{ if } t_{j-1} \le \hat{t}_i \\ 0 \quad \text{otherwise.} \end{cases}$$



# Construction of Projected Gradient Path

Obtain explicit expression for $q(x)$ in segment $t \in [t_{j-1}, t_j]$:

$$q(x(t)) = b^T \left(x(t_{j-1}) + \delta s^{j-1}\right) \\ \quad + \tfrac{1}{2}\left(x(t_{j-1}) + \delta s^{j-1}\right)^T G \left(x(t_{j-1}) + \delta s^{j-1}\right),$$

which is a 1D quadratic in $\delta$ and can be written as

$$q(\delta) = q(x(t)) = f_{j-1} + f'_{j-1}\delta + \tfrac{1}{2}\delta^2 f''_{j-1}, \quad \text{for } \delta \in [0, t_j - t_{j-1}],$$

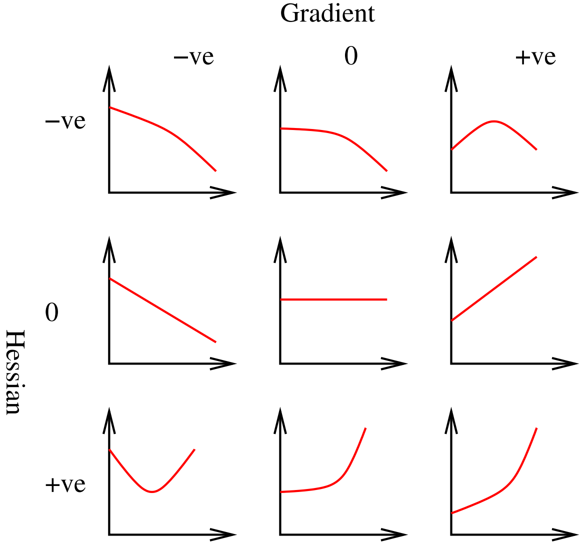
with coefficients given by

$$\begin{aligned} f_{j-1} &= b^T x(t_{j-1}) + \tfrac{1}{2} x(t_{j-1})^T G x(t_{j-1}) \\ f'_{j-1} &= b^T s^{(j-1)} + x(t_{j-1})^T G s^{(j-1)} \\ f''_{j-1} &= s^{(j-1)^T} G s^{(j-1)}. \end{aligned}$$

To find minimum of $q(x(t)) = q(\delta)$ in $[0, t_j - t_{j-1}]$ differentiate
... minimizer then depends on the signs of $f'_{j-1}$ and $f''_{j-1}$

# Finding Minimum on Projected Gradient Path

Nine cases for min of $q(x(t)) = q(\delta)$ in $[0, t_j - t_{j-1}]$



# Construction of Projected Gradient Path

Nine cases for minimization of $q(\delta) = f'_{j-1}\delta + \frac{1}{2}\delta^2 f''_{j-1}$

| | $f'_{j-1} < 0$ | $f'_{j-1} = 0$ | $f'_{j-1} > 0$ |
|---|---|---|---|
| $f''_{j-1} < 0$ | $\delta = t_j - t_{j-1}$ | $\delta = t_j - t_{j-1}$ | $\delta = 0$ |
| $f''_{j-1} = 0$ | $\delta = t_j - t_{j-1}$ | $\delta = t_j - t_{j-1}$ | $\delta = 0$ |
| $f''_{j-1} > 0$ | $\delta = \min\left(\frac{-f'_{j-1}}{f''_{j-1}}, t_j - t_{j-1}\right)$ | $\delta = 0$ | $\delta = 0$ |

Optimal $\delta$ either on boundary or in interior.

Algorithm for First Minimizer of $q(x(t))$

1. Examine intervals $[0, t_1], [t_1, t_2], [t_2, t_3], \ldots$
2. Stop in interval $j$ , where the optimum, $\delta^* < t_j - t_{j-1}$

Optimum is $t^* = t_{j-1} + \delta^*$, and the Cauchy point is $x_C = x(t^*)$

# First Minimizer Along Projected Gradient Path

Given initial point, $x$, and direction, $g$.

Compute breakpoints $\hat{t}_i$, and set $j = 1$.

Get $t_0 := 0 < t_1 < t_2 < \ldots$ ordering $\hat{t}_i$, remove duplicates/zeros.

**repeat**

Compute $f'_{j-1}, f''_{j-1}$, and find $\delta^*$ from above table.

**if** $\delta^* < t_j - t_{j-1}$ **then**

Set $t^* = t_{j-1} + \delta^*$ found.

**end**

Set $j = j + 1$.

**until** $t^*$ *found*;

Return $t^*$ and $x(t^*)$.

# Subspace Optimization

Bound constrained QP

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; q(x) = b^T x + \tfrac{1}{2} x^T G x \quad \text{subject to } l \le x \le u$$

First minimizer along projected-gradient path:

$$\underset{t}{\text{minimize}} \; q(x(t)), \quad \text{where} \quad x(t) := P_{[l,u]}\left(x - tg\right),$$

gives Cauchy point, $x_c$ & candidate active set ... explore subspace

## Cauchy Point Active Set, $\mathcal{A}(x_C)$, Subproblem

$$\begin{aligned}
&\underset{x}{\text{minimize}} \;\; q(x) = \tfrac{1}{2} x^T G x + b^T x + c \\
&\text{subject to } x_i = l_i,\ \forall\ i \in \mathcal{A}(x_C) \qquad x_i = u_i,\ \forall\ -i \in \mathcal{A}(x_C) \\
&\qquad\qquad\quad l_i \le x_i \le u_i,\ \forall\ \pm i \notin \mathcal{A}(x_C).
\end{aligned}$$

Extend conjugate-gradient algorithm $\Rightarrow$ good for large problems

# Quadratic Projected-Gradient Projection Algorithm

Bound constrained QP

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; q(x) = b^T x + \frac{1}{2} x^T G x \quad \text{subject to } l \le x \le u$$

**Quadratic Projected-Gradient Projection Algorithm**
Given $l \le x^{(0)} \le u$, set $k = 0$.
**repeat**

Define the path $x^{(k)}(t) := P_{[l,u]}\left(x^{(k)} - t g^{(k)}\right)$.

Get Cauchy point, $x_C^{(k)}$: find first minimizer $q(x^{(k)}(t))$

Active set, $\mathcal{A}(x_C^{(k)})$, set up subspace optimization problem.

Approximately solve subspace optimization for $l \le x^{(k+1)} \le u$.

Set $k = k + 1$.

**until** $x^{(k)}$ *is (local) optimum*;

# Quadratic Projected-Gradient Projection Algorithm

- Algorithm requires feasible starting point, $l \leq x^{(0)} \leq u$
  - If starting point, $\hat{x}$ infeasible, then project: $x^{(0)} = P_{[l,u]}(\hat{x})$
- Use conjugate-gradient method to solve subproblem approx.
  - Stop, when we reach bound
  - Check for negative curvature ... go to bound

**Theorem (Finite Active-Set Identification)**

*Assume solution, $x^*$, is strictly complementary, i.e.*

$$x_i^* = l_i \;\Rightarrow\; \frac{\partial f}{\partial x_i}(x^*) > 0 \quad \textit{and} \quad x_i^* = u_i \;\Rightarrow\; \frac{\partial f}{\partial x_i}(x^*) < 0$$

*then identify optimal active set, $\mathcal{A}(x^*)$, after finite number of projected gradient steps*

... hence terminate finitely for a quadratic

# Outline

# Bound-Constrained Nonlinear Optimization

Now consider bound-constrained optimization:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to } l \leq x \leq u$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ twice continuously differentiable,
and bounds $l, u \in \mathbb{R}^n$ can be infinite.

How can we generalize projected-gradient to nonlinear $f(x)$?

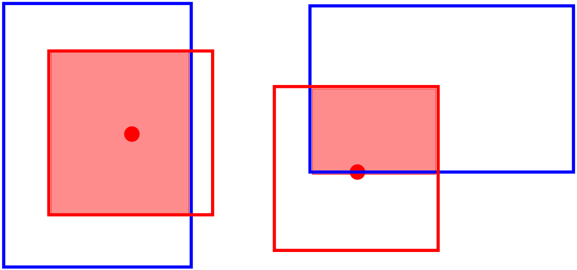
- Use Cauchy-point (steepest descend) idea to get convergence.
- Perform subspace optimization of a quadratic model
  ... measure progress with respect to $f(x)$
- Embed in trust-region or line-search framework
  ... here show TR framework

# Bound-Constrained Nonlinear Optimization

Intersection of $\ell_\infty$ trust-region with bounds is simple:

$$l_i^{(k)} = \max\left(l_i, x_i^{(k)} - \Delta_k\right)$$

$$u_i^{(k)} = \min\left(u_i, x_i^{(k)} + \Delta_k\right)$$



In following, assume that bounds $l, u$ are already $l^{(k)}, u^{(k)}$

# General Projected-Gradient Algorithm

Start by describing how we obtain a new point:

**Algorithm** $s$ **= StepComputation(**$x^{(k)}$**)**

1. Define path $x^{(k)}(t) := P_{[l,u]}\left(x^{(k)} - t\hat{g}^{(k)}\right)$.
2. Form quadratic model, $q_k(s)$, of $f(x)$ around $x^{(k)}$.
3. Get Cauchy point, $x_C^{(k)}$: first minimizer of $q_k(s^{(k)}(t))$
4. Get active set, $\mathcal{A}(x_C^{(k)})$, set up subspace optimization.
5. Approx. minimize $q_k(s)$ over inactive variables such that $l \leq x^{(k)} + s \leq u$.

# General Projected-Gradient Algorithm

Given $l \leq x^{(0)} \leq u$, set $\Delta_0 = 1$, and $k = 0$.

**repeat**

Obtain step $s$: $l \leq x^{(k)} + s \leq u$ with Cauchy property.

Compute $r_k = \frac{f^{(k)} - f(x^{(k)} + s^{(k)})}{f^{(k)} - q_k(s^{(k)})} = \frac{\text{act. reductn.}}{\text{pred. reductn.}}$.

**if** $r_k \geq \eta_v$ *very successful step* **then**

Accept $x^{(k+1)} := x^{(k)} + s^{(k)}$, increase $\Delta_{k+1} := \gamma_i \Delta_k$.

**else if** $r_k \geq \eta_v$ *successful step* **then**

Accept $x^{(k+1)} := x^{(k)} + s^{(k)}$, set $\Delta_{k+1} := \Delta_k$.

**else if** $r_k < \eta_v$ *unsuccessful step* **then**

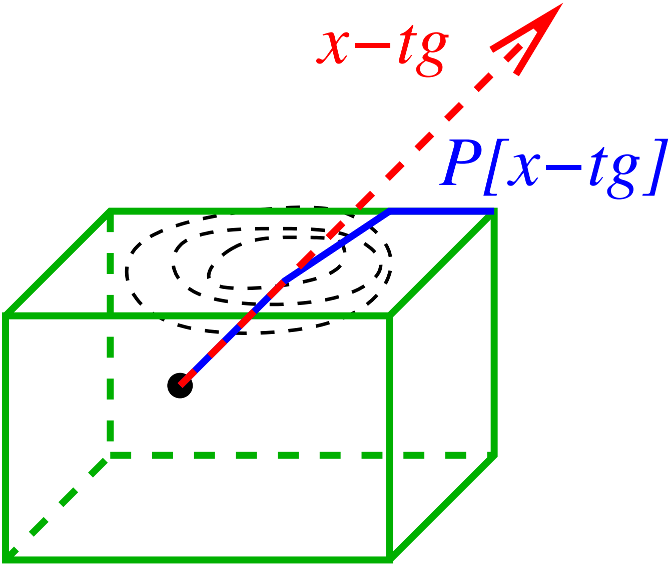
Reject step $x^{(k+1)} := x^{(k)}$, reduce $\Delta_{k+1} := \gamma_d \Delta_k$.

**end**

Set $k = k + 1$.

**until** $x^{(k)}$ *is (local) optimum*;

# Illustration of Projected-Gradient Algorithm



# Conclusions & Summary

Presented bound constrained optimization

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \quad \text{subject to } l \le x \le u$$

Introduces concept of active sets

Derived projected gradient method with subspace optimization

- Computes min along piecewise linear path: Cauchy point
- Uses conjugate gradients to minimize in subspace