Numerical Nonlinear Optimization Part II

Andreas Wächter

Center for Nonlinear Studies

June 29, 2020

Goal of this Lecture Mini-Series

- Accessible to broad audience.
  - Assume basic knowledge of multi-dimensional calculus.
- Give overview of practical optimization algorithms for nonlinear constrained optimization.
  - Includes theoretical characterization of optima.
- Concentrate on intuition of algorithms and theoretical concepts.
  - No complicated proofs.
  - Some “cheating” (ignoring some subtleties).
- 90 min reserved, but roughly targeting 75 min.
- I will make slides available after the lectures.

Outline

Last week:

- Optimality conditions for unconstrained optimization.
- Three basic unconstrained optimization algorithms.

Today:

- Line search and trust region methods.
- Optimality conditions for constrained optimization.

Summary of Last Lecture

$$\boxed{\min_{x \in \mathbb{R}^n} \ f(x)}$$

- Look for local minima.
- Main theoretical tool: Taylor expansions.

$$\boxed{f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_k) d}$$

- Necessary optimality conditions:

$$\boxed{\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \text{ is positive semi-definite}}$$

- Sufficient optimality conditions:

$$\boxed{\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \text{ is positive definite}}$$

Unified Algorithm Framework

- Quadratic model of objective at iterate x_k:

$$\boxed{q_k(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d}$$

- Different choices of B_k result in different method.

Given: Stopping tolerance $\epsilon > 0$.

1: Choose x_0 and set $k \leftarrow 0$.
2: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
3: Compute or update B_k.
4: Minimize $q_k(x_k + d)$ to get step d_k. $(d_k = -B_k^{-1} \nabla f(x_k))$
5: Take step $x_{k+1} = x_k + d_k$.
6: Increase iteration counter $k \leftarrow k + 1$.
7: **end while**

Comparison of Steps (1)

Gradient method:

- $B_k = \frac{1}{\alpha} I$
- $d_k = -\alpha \nabla f(x_K)$.
- Global linear convergence rate for appropriate step size α.
- Does not require second derivatives.

Newton's method:

- $B_k = \nabla^2 f(x_k)$
- Local quadratic convergence rate.
- Requires computation of $\nabla^2 f(x_k)$.
- Needs special attention when $\nabla^2 f(x_k)$ is indefinite.
  - In that case, $q_k(x_k + d)$ does not have a minimizer.

Comparison of Steps (2)

Quasi-Newton methods:

- B_k is Hessian approximation.
- Updated in each iteration by a formula (e.g., BFGS).
- Local super-linear convergence rate (in theory under somewhat strong assumptions, but often in practice).
- Does not require second derivatives.

Our Algorithm So Far

Given: Stopping tolerance $\epsilon > 0$.

Choose x_0 and set $k \leftarrow 0$.
while $\|\nabla f(x_k)\| > \epsilon$ **do**
    Compute or update B_k.
    Minimize $q_k(x_k + d)$ to get step d_k.

    Take step $x_{k+1} = x_k + d_k$.
    Increase iteration counter $k \leftarrow k + 1$.
end while

Concerns:

- Sometimes, this basic algorithm fails to converge.
- The iterates might cycle or diverge.

Our Algorithm So Far

Given: Stopping tolerance $\epsilon > 0$.

```
Choose x_0 and set k ← 0.
while ‖∇f(x_k)‖ > ε do
    Compute or update B_k.
    Minimize q_k(x_k + d) to get step d_k.

    Take step x_{k+1} = x_k + d_k.
    Increase iteration counter k ← k + 1.
end while
```

Concerns:

- Sometimes, this basic algorithm fails to converge.
- The iterates might cycle or diverge.
- One remedy: Take a shorter step.

Our Algorithm So Far

Given: Stopping tolerance $\epsilon > 0$.

Choose x_0 and set $k \leftarrow 0$.
while $\|\nabla f(x_k)\| > \epsilon$ **do**
    Compute or update B_k.
    Minimize $q_k(x_k + d)$ to get step d_k.
    Choose step size $\alpha_k > 0$.
    Take step $x_{k+1} = x_k + \alpha_k \cdot d_k$.
    Increase iteration counter $k \leftarrow k + 1$.
end while

Concerns:

- Sometimes, this basic algorithm fails to converge.
- The iterates might cycle or diverge.
- One remedy: Take a shorter step.

Line Search

$$\boxed{x_{k+1} = x_k + \alpha_k \cdot d_k}$$

- Introduce a step size $\alpha_k > 0$.
- Choose α_k so that objective is improved:

$$\boxed{f(x_k + \alpha_k \cdot d_k) < f(x_k)}$$

- Called line search because it looks for a new iterate along the line

$$\{x_k + \alpha \cdot d_k : \ \alpha > 0\}$$

- We could seek minimizer

$$\min_{\alpha>0} \ f(x_k + \alpha \cdot d_k)$$

but that is usually very computationally expensive.

Backtracking Line Search

Given: Stopping tolerance $\epsilon > 0$.

```
Choose x_0 and set k ← 0.
while ‖∇f(x_k)‖ > ε do
    Compute or update B_k.
    Minimize q_k(x_k + d) to get step d_k.
    Set α_k ← 1.
    while f(x_k + α_k · d_k) ≥ f(x_k) do
        Set α_k ← (1/2) α_k.
    end while
    Take step x_{k+1} = x_k + α_k · d_k.
    Increase iteration counter k ← k + 1.
end while
```

Descent Direction

$$\boxed{f(x_k + \alpha_k \cdot d_k) < f(x_k)}$$

- To make sure such $\alpha_k > 0$ exists, d_k should be descent direction.

$$f(x_k + \alpha_k \cdot d_k) \qquad\qquad\qquad\qquad < f(x_k)$$

Descent Direction

$$\boxed{f(x_k + \alpha_k \cdot d_k) < f(x_k)}$$

- To make sure such $\alpha_k > 0$ exists, d_k should be descent direction.

$$f(x_k + \alpha_k \cdot d_k) \approx f(x_k) + \alpha_k \nabla f(x_k)^T d_k < f(x_k)$$

- So, we need

$$\boxed{\nabla f(x_k)^T d_k < 0.}$$

- Then, for sufficiently small α_k, the step is accepted.

Ensuring Descent Directions

- How can we guarantee that d_k is a descent direction?
- Recall step calculation: Solve $\boxed{B_k d_k = -\nabla f(x^k)}$.
- We want

$$0 < -\nabla f(x_k)^T d_k = d_k^T B_k d_k$$

- So, d_k is a descent direction if B_k is positive definite.
  - This is also the condition that ensures q_k has minimizer!

- We would not think about this if we just apply Newton's method to "$\nabla f(x) = 0$".

Gradient method:	$B_k = \frac{1}{\alpha} I$	✓
BFGS method:	B_k positive definite	✓
Newton's method:	$B_k = \nabla^2 f(x_k)$	**?**

Descent Directions for Newton's Method

- If f is not convex, $B_k = \nabla^2 f(x_k)$ might not be positive definite.
- In that case, we need to modify B_k.
- One option: Use
$$\boxed{B_k = \nabla^2 f(x_k) + \lambda \cdot I}$$
with some regularization parameter $\lambda \geq 0$.
- If λ sufficiently large, B_k is positive definite.
- Could compute most negative eigenvalue of B_k, but that is costly.
- Cheap strategy: Try increasingly larger values of λ.

Simple Strategy to Compute Regularization Parameter λ

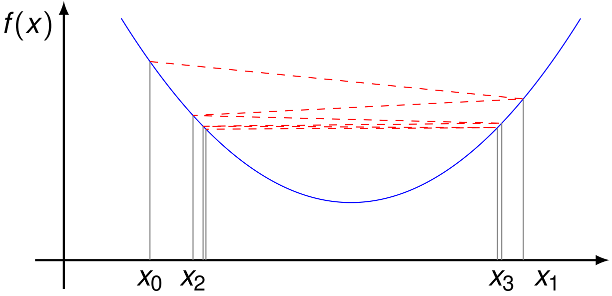
Given: x_k and parameters $\lambda_{\text{small}} > 0$, $\kappa > 1$.

Set $\lambda \leftarrow 0$.
repeat
    Set $B_k \leftarrow \nabla^2 f(x_k) + \lambda \cdot I$.
    Try to compute Cholesky factorization

$$\boxed{B_k = L_k^T L_k} \qquad\qquad (L_k \text{ lower triangular})$$

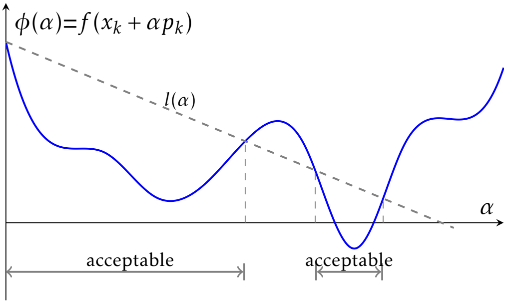
    if successful **then**
        Solve $L_k^T v = -\nabla f(x_k)$ and $L_k d_k = v$ to get d_k.
    else
        Set $\lambda \leftarrow \max\{\lambda_{\text{small}}, \kappa \cdot \lambda\}$
    end if
until d_k has been computed

Need Sufficient Decrease



- In our algorithm, we asked for "$f(x_k + \alpha_k \cdot d_k) < f(x_k)$."
- However, that is not enough to guarantee convergence.
- Need to make sure α_k provides <u>sufficient</u> decrease in f.

Armijo Condition



- Relaxed tangent: $\ell(\alpha) = f(x_k) + \alpha \cdot \eta \nabla f(x_k)^T d_k$
- Armijo condition:

$$\boxed{f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \cdot \eta \nabla f(x_k)^T d_k}$$

- With this, can prove global convergence under mild assumptions:
  - "Every limit point of $\{x_k\}$ is a stationary point."

Alternative Strategy: Trust Region

$$\boxed{q_k(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d}$$

- This is a local model of $f(x)$ around x_k.
- We should “trust” it only for a limited range.

- Compute step from trust-region subproblem:

$$\min_{d \in \mathbb{R}^n} \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d$$

- Trust-region radius $\Delta_k > 0$ expresses how far we trust the model.
- Δ_k is updated from iteration to iteration.

Alternative Strategy: Trust Region

$$\boxed{q_k(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d}$$

- This is a local model of $f(x)$ around x_k.
- We should "trust" it only for a limited range.

- Compute step from trust-region subproblem:

$$\min_{d \in \mathbb{R}^n} \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d$$

- Trust-region radius $\Delta_k > 0$ expresses how far we trust the model.
- Δ_k is updated from iteration to iteration.

Alternative Strategy: Trust Region

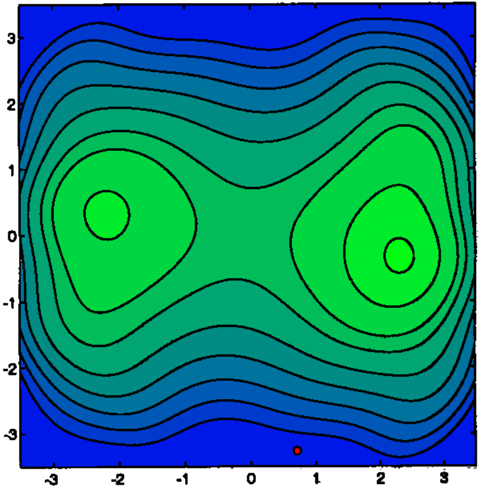
$$\boxed{q_k(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_k) d}$$

- This is a local model of $f(x)$ around x_k.
- We should "trust" it only for a limited range.
- Compute step from trust-region subproblem:

$$\begin{array}{ll} \min\limits_{d \in \mathbb{R}^n} & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \\ \text{s.t.} & \|d_k\| \le \Delta_k \end{array}$$

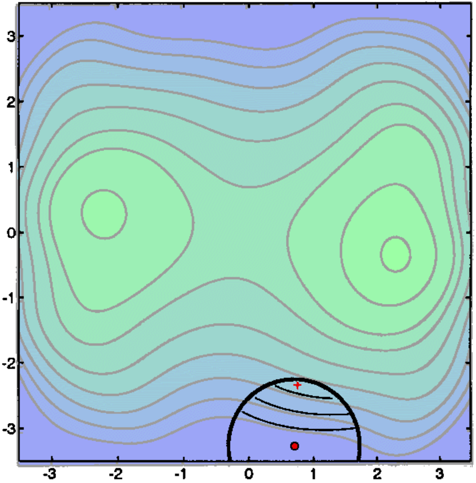
- Trust-region radius $\Delta_k > 0$ expresses how far we trust the model.
- Δ_k is updated from iteration to iteration.

Trust-Region Method Example Problem



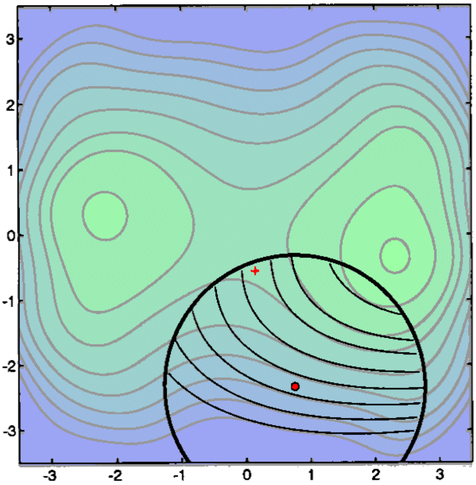
(From Frank Vanden Berghen's website)

Trust-Region Method Example Iteration 1



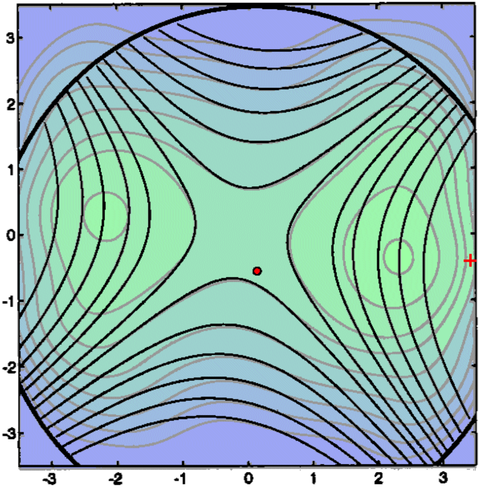
(From Frank Vanden Berghen's website)

Trust-Region Method Example Iteration 2



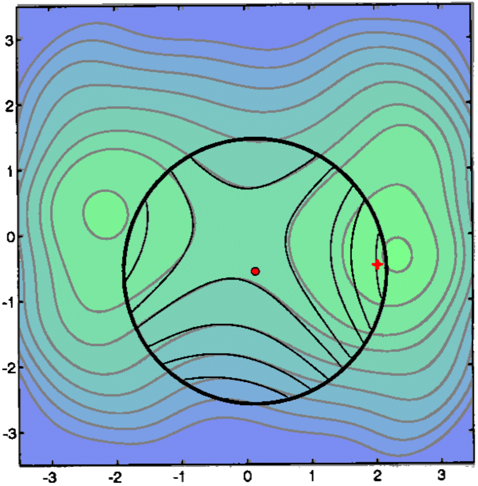
(From Frank Vanden Berghen's website)

Trust-Region Method Example Iteration 3



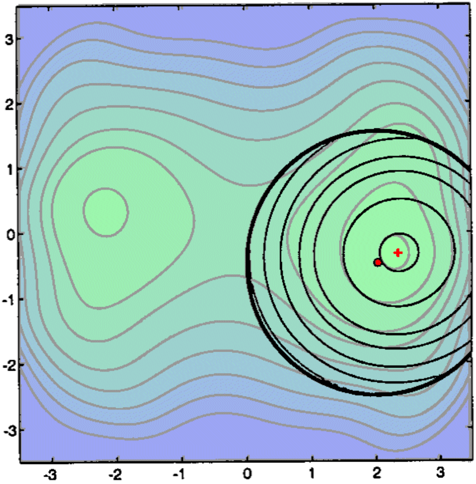
(From Frank Vanden Berghen’s website)

Trust-Region Method Example Iteration 4



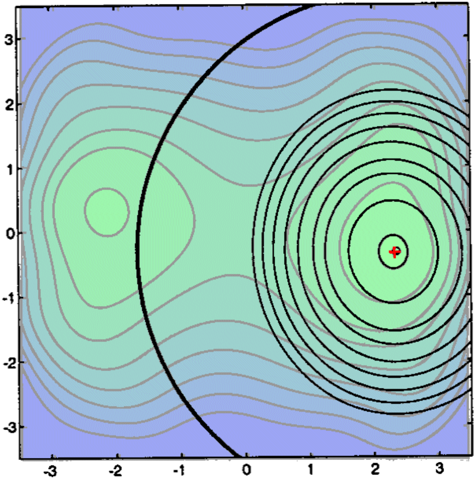
(From Frank Vanden Berghen's website)

Trust-Region Method Example Iteration 5



(From Frank Vanden Berghen's website)

Trust-Region Method Example Iteration 6



(From Frank Vanden Berghen's website)

Trust-Region Update

- Idea:
  - Increase trust region if $q_k(x_k + d_k)$ agrees well with $f(x_k + d_k)$.
  - Decrease trust region if $q_k(x_k + d_k)$ is very different from $f(x_k + d_k)$.
- How can we measure quality of model agreement?
  - Predicted reduction: $\text{pred}_k = q_k(x_k) - q_k(x_k + d_k) > 0$
  - Actual reduction: $\text{ared}_k = f(x_k) - f(x_k + d_k)$
  - Agreement ratio: $\rho_k = \frac{\text{ared}_k}{\text{pred}_k}$
- Ideally: $\rho_k \approx 1$.
- Good agreement: $\rho_k \geq \eta_{\text{good}}$ with $\eta_{\text{good}} \in (0, 1)$.
- Bad agreement: $\rho_k \leq \eta_{\text{bad}}$ with $\eta_{\text{bad}} \in (0, \eta_{\text{good}}]$.

A Basic Trust-Region Algorithm

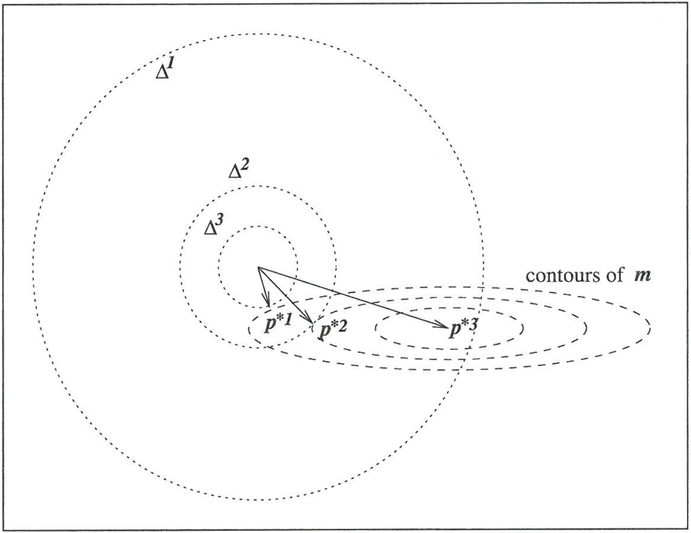
Given: Parameter $\epsilon > 0$, $0 < \eta_{bad} \leq \eta_{good} < 1$.

Choose $x_0 \in \mathbb{R}^n$, $\Delta_0 > 0$. Set $k \leftarrow 0$.
while $\|\nabla f(x_k)\| > \epsilon$ **do**
    Compute or update B_k.
    Solve trust-region subproblem with radius Δ_k to get d_k.
    Set $\text{pred}_k = q_k(x_k) - q_k(x_k + d_k)$, $\text{ared}_k = f(x_k) - f(x_k + d_k)$.
    Compute $\rho_k = \text{ared}_k / \text{pred}_k$.
    if $\rho_k \geq \eta_{\text{good}}$ **then**
        Set $x_{k+1} = x_k + d_k$ and $\Delta_{k+1} = 2\Delta_k$.
    else if $\rho_k > \eta_{\text{bad}}$ **then**
        Set $x_{k+1} = x_k + d_k$ and $\Delta_{k+1} = \Delta_k$.
    else
        Set $x_{k+1} = x_k$ and $\Delta_{k+1} = \frac{1}{2}\Delta_k$.
    end if
    Increase $k \leftarrow k + 1$.
end while

Trust-Region Algorithm Discussion

- Handles indefinite $B_k = \nabla^2 f(x_k)$ in a natural manner.
- We have $\rho_k \to 1$ as $\Delta_k \to 0$.
  - So, a new iterate will eventually be accepted.
- The trial points lie on a curved path, not a line.
- As $\Delta_k \to 0$, trial step approaches gradient direction.
- Convergence can still be achieved if trust-region subproblem is solved inaccurately, e.g., for large problems.
- Can prove global convergence under mild assumptions:
  - "Every limit point of $\{x_k\}$ is a stationary point."

Path of Trust Region Trial Points



Unconstrained Optimization Recap

- We saw three types of step computations d_k:
  - Gradient method
  - Newton's method
  - Quasi-Newton methods
- We saw two strategies to guarantee global convergence:
  - Line search
  - Trust region
- For large-scale problems:
  - Use sparse matrix factorization techniques.
  - Use iterative linear solvers, e.g., conjugate gradients.
  - Limited-memory BFGS (L-BFGS).

Constrained Nonlinear Optimization Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t.} \ & c_E(x) = 0 \\ & c_I(x) \leq 0 \end{aligned}$$

$$\begin{aligned} f &: \mathbb{R}^n \longrightarrow \mathbb{R} \\ c_E &: \mathbb{R}^n \longrightarrow \mathbb{R}^{n_E} \\ c_I &: \mathbb{R}^n \longrightarrow \mathbb{R}^{n_I} \end{aligned}$$

- We assume that all functions are twice continuously differentiable.
- A point $x \in \mathbb{R}^n$ satisfying all constraints, i.e.,

$$\begin{aligned} c_E(x) &= 0 \\ c_I(x) &\leq 0 \end{aligned}$$

  is called feasible.
- Let $\Omega \subset \mathbb{R}^n$ be the set of all feasible point.
- Often called "Nonlinear Program" (NLP).

Types of Minimizers

$$\boxed{\min_{x\in\mathbb{R}^n} f(x)} \qquad \text{(NLP)}$$

- A point $x^* \in \mathbb{R}^n$ is a global minimizer of (NLP) if $f(x) \geq f(x^*)$ for all $x \in \mathbb{R}^n$.
- A point $x^* \in \mathbb{R}^n$ is a local minimizer of (NLP), if $f(x) \geq f(x^*)$ for all $x \in N_\epsilon(x^*)$ for some $\epsilon > 0$.

Types of Minimizers

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & f(x) \\ \text{s.t. } & c_E(x) = 0 \\ & c_I(x) \le 0 \end{aligned} \qquad \text{(NLP)}$$

- A point $x^* \in \Omega$ is a global minimizer of (NLP) if $f(x) \geq f(x^*)$ for all $x \in \Omega$.
- A point $x^* \in \Omega$ is a local minimizer of (NLP), if $f(x) \geq f(x^*)$ for all $x \in N_\epsilon(x^*) \cup \Omega$ for some $\epsilon > 0$.
- Again, the methods we will discuss try to find local minimizers.

Special Case: Convex Problems

Definition (Convex Set)

A set S is convex, if for all $x, y \in S$ and all $\lambda \in [0, 1]$ we have

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

Proposition

If f is convex and Ω is convex, then every local minimizer is a global minimizer.

Proposition

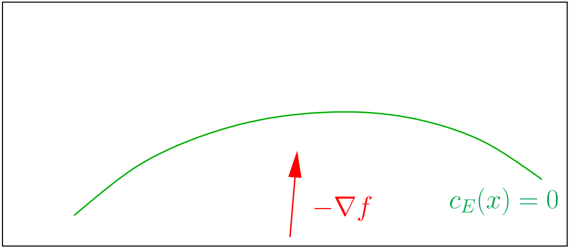
If all c_E are affine and all c_I are convex, then Ω is convex.

Examples:

- Linear Programs, Second-Order Cone Programs, Semi-Definite Programs.

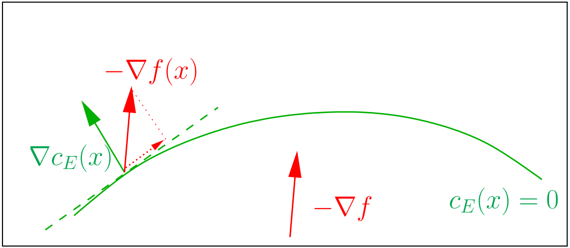
Optimality Conditions: Equality Constraints

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$



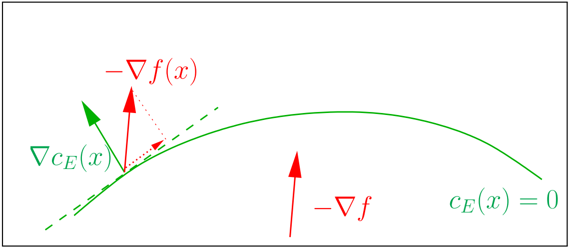
Optimality Conditions: Equality Constraints

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & c_E(x) = 0 \end{array}$$



Optimality Conditions: Equality Constraints

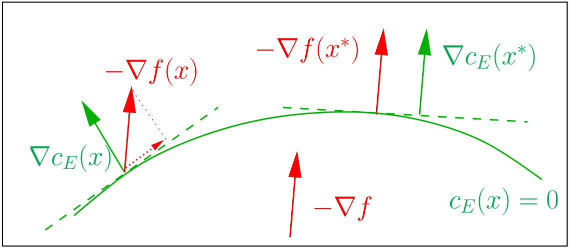
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$



- Moving along projection of $-\nabla f(x)$ onto tangent space of feasible set decreases objective.

Optimality Conditions: Equality Constraints

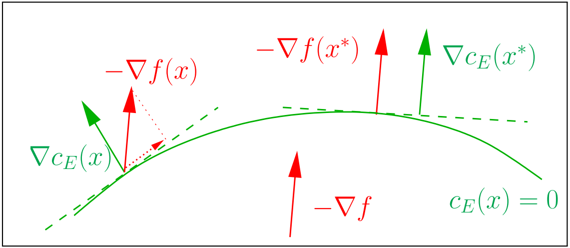
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$



- Moving along projection of $-\nabla f(x)$ onto tangent space of feasible set decreases objective.

Optimality Conditions: Equality Constraints

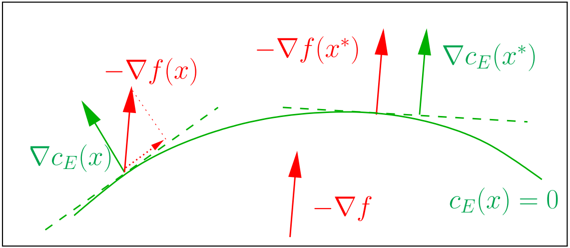
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$



- Moving along projection of $-\nabla f(x)$ onto tangent space of feasible set decreases objective.
- At local minimum, projection of $-\nabla f(x)$ must be zero.

Optimality Conditions: Equality Constraints

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & c_E(x) = 0 \end{array}$$

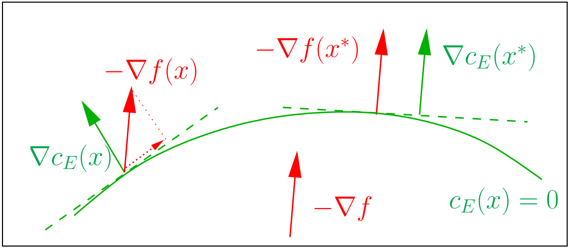


- Moving along projection of $-\nabla f(x)$ onto tangent space of feasible set decreases objective.
- At local minimum, projection of $-\nabla f(x)$ must be zero.
- For this, $-\nabla f(x^*)$ must be linear combination of constraint gradient:

$$\boxed{-\nabla f(x^*) = \nabla c_E(x^*)\,\lambda_E} \qquad \lambda_E \in \mathbb{R}$$

Optimality Conditions: Equality Constraints

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$

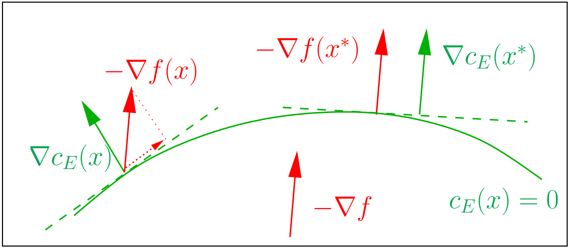


- Moving along projection of $-\nabla f(x)$ onto tangent space of feasible set decreases objective.
- At local minimum, projection of $-\nabla f(x)$ must be zero.
- For this, $-\nabla f(x^*)$ must be linear combination of constraint gradients:

$$\boxed{-\nabla f(x^*) = \sum_{j=1}^{n_E} \nabla c_{E,j}(x^*)\, \lambda_{E,j}} \qquad \lambda_E \in \mathbb{R}^{n_E}$$

Optimality Conditions: Equality Constraints

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & c_E(x) = 0 \end{array}$$



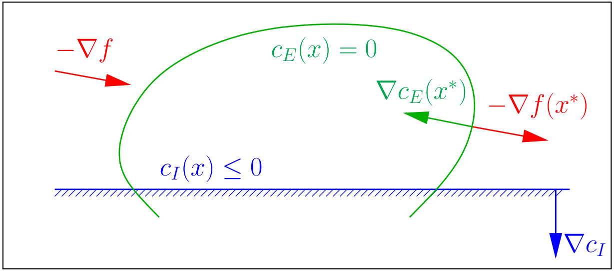
- Moving along projection of $-\nabla f(x)$ onto tangent space of feasible set decreases objective.
- At local minimum, projection of $-\nabla f(x)$ must be zero.
- For this, $-\nabla f(x^*)$ must be linear combination of constraint gradients:

$$\boxed{-\nabla f(x^*) = \sum_{j=1}^{n_E} \nabla c_{E,j}(x^*)\,\lambda_{E,j} = \nabla c_E(x^*)\,\lambda_E} \qquad \lambda_E \in \mathbb{R}^{n_E}$$

- Notation: Columns of $\nabla c_E(x^*)$ are the constraints gradients.

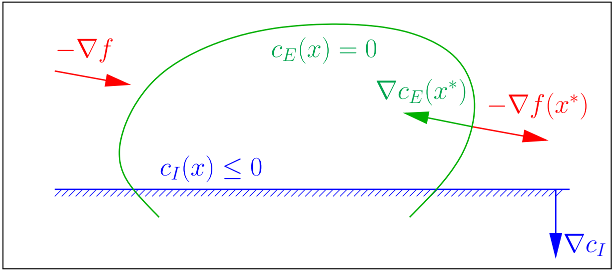
Optimality Conditions: Inequality Constraints

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \leq 0$$



Optimality Conditions: Inequality Constraints

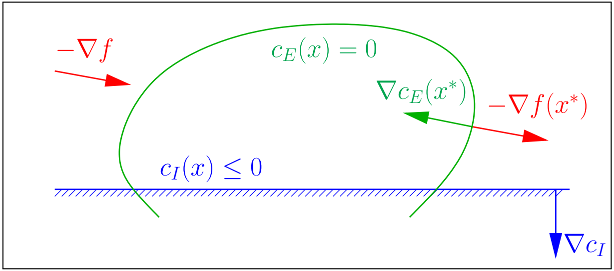
$$\min_{x\in\mathbb{R}^n} \; f(x)$$
$$\text{s.t.} \;\; c_E(x) = 0$$
$$c_I(x) \le 0$$



- First local minimum:
  - Inequality constraint is inactive (not binding), it might as well not be there.

Optimality Conditions: Inequality Constraints

$$\min_{x \in \mathbb{R}^n} \; f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \le 0$$

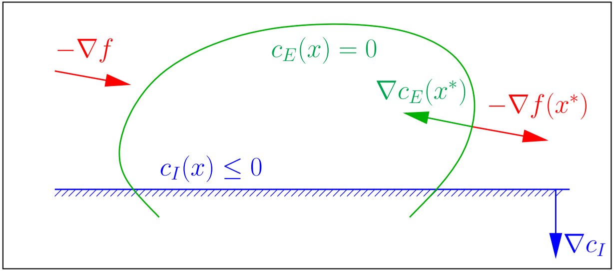


- First local minimum:
  - Inequality constraint is inactive (not binding), it might as well not be there.
- Same relationship as before:

$$\boxed{-\nabla f(x^*) = \nabla c_E(x^*) \cdot \lambda_E} \qquad \lambda_E \in \mathbb{R}$$

Optimality Conditions: Inequality Constraints

$$\min_{x\in\mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \le 0$$

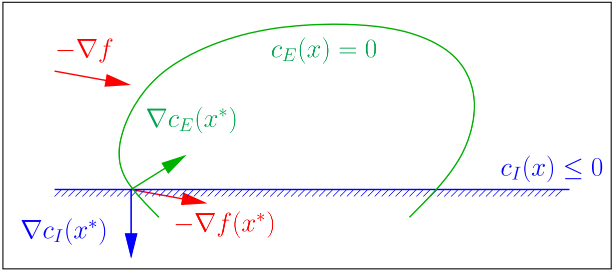


- First local minimum:
  - Inequality constraint is inactive (not binding), it might as well not be there.
- Same relationship as before:

$$\boxed{-\nabla f(x^*) = \nabla c_E(x^*)\cdot\lambda_E + \nabla c_I(x^*)\cdot\lambda_I} \qquad \lambda_E \in \mathbb{R},\ \lambda_I = 0$$

Optimality Conditions: Inequality Constraints

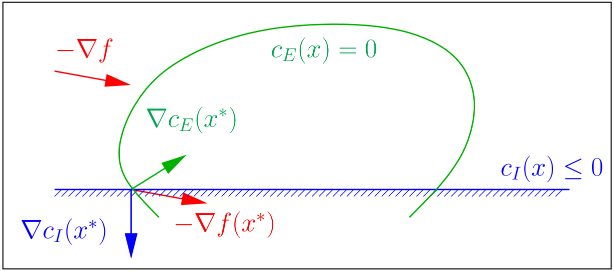
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \leq 0$$



- Second local minimum:
  - Inequality constraint is active.

Optimality Conditions: Inequality Constraints

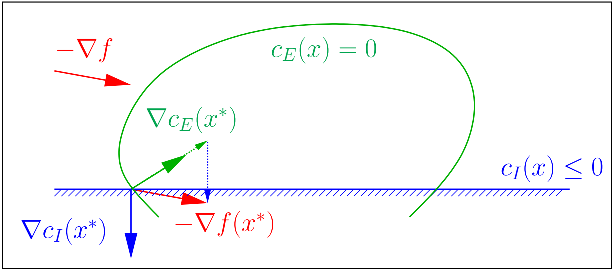
$$\min_{x\in\mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \leq 0$$



- Second local minimum:
  - Inequality constraint is active.
- Projection of $-\nabla f(x^*)$ onto tangent space of "$c_E(x) = 0$" points into direction that violates "$c_I(x) \leq 0$".

Optimality Conditions: Inequality Constraints

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \leq 0$$

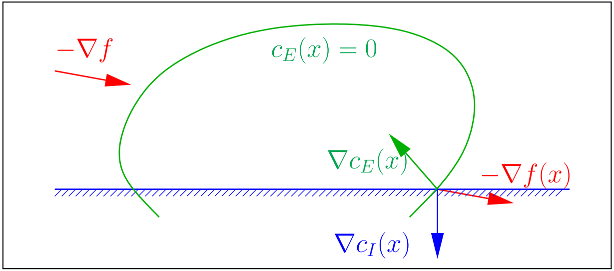


- Second local minimum:
  - Inequality constraint is active.
- Projection of $-\nabla f(x^*)$ onto tangent space of "$c_E(x) = 0$" points into direction that violates "$c_I(x) \leq 0$".

$$\boxed{-\nabla f(x^*) = \nabla c_E(x^*) \cdot \lambda_E + \nabla c_I(x^*) \cdot \lambda_I} \qquad \lambda_E \in \mathbb{R}, \ \lambda_I \geq 0$$

Optimality Conditions: Inequality Constraints

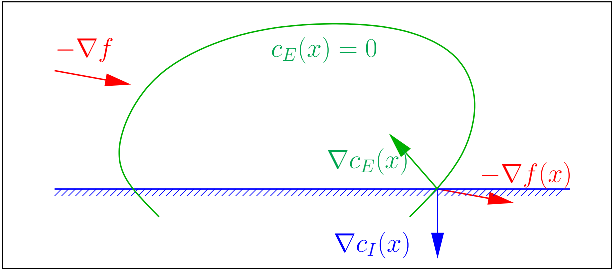
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \le 0$$



- Another point where inequality is active.

Optimality Conditions: Inequality Constraints

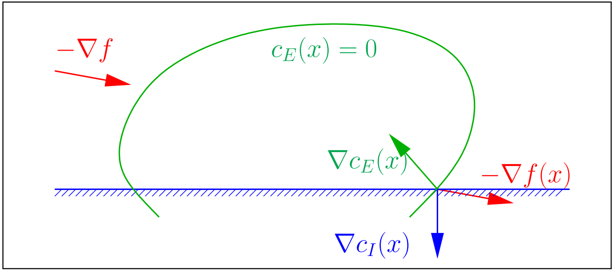
$$\min_{x\in\mathbb{R}^n} \ f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \le 0$$



- Another point where inequality is active.
- Projection of $-\nabla f(x)$ onto tangent space of "$c_E(x) = 0$" points into direction that satisfies "$c_I(x) \le 0$".

Optimality Conditions: Inequality Constraints

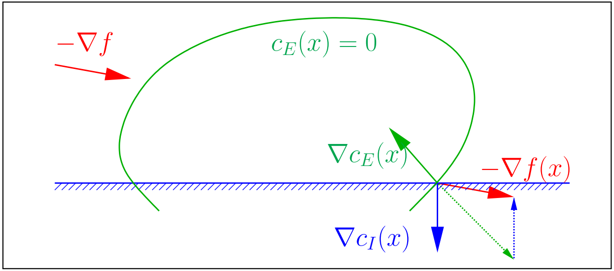
$$\min_{x \in \mathbb{R}^n} \; f(x)$$
$$s.t. \quad c_E(x) = 0$$
$$c_I(x) \leq 0$$



- Another point where inequality is active.
- Projection of $-\nabla f(x)$ onto tangent space of "$c_E(x) = 0$" points into direction that satisfies "$c_I(x) \leq 0$".
  - Can move into this direction and improve objective.

Optimality Conditions: Inequality Constraints

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ s.t. \quad & c_E(x) = 0 \\ & c_I(x) \le 0 \end{aligned}$$



- Another point where inequality is active.
- Projection of $-\nabla f(x)$ onto tangent space of "$c_E(x) = 0$" points into direction that satisfies "$c_I(x) \le 0$".
  - Can move into this direction and improve objective.

$$\boxed{-\nabla f(x) = \nabla c_E(x) \cdot \lambda_E + \nabla c_I(x) \cdot \lambda_I} \qquad \lambda_E \in \mathbb{R},\ \lambda_I < 0$$

Summary of Conditions

- Projection of $-\nabla f(x^*)$ onto the right tangent space must be zero:

$$\boxed{\nabla f(x^*) + \nabla c_E(x^*)\lambda_E + \nabla c_I(x^*)\lambda_I = 0}$$

  for some Lagrangian multipliers $\lambda_E \in \mathbb{R}^{n_E}$ and $\lambda_I \in \mathbb{R}^{n_I}$.
  - There is no direction that decreases objective and stays feasible.
- Releasing active inequality does not make it possible to improve objective:

$$\boxed{\lambda_I \geq 0}$$

- Only active constraints can contribute to the (local) optimality conditions:

$$\boxed{c_{I,j}(x^*) \cdot \lambda^*_{I,j} = 0} \quad \text{for all } j = 1, \ldots, n_I$$

  - If constraint is not active, multiplier must be zero.
  - This is called complementarity condition.
  - "At least one of $c_{I,j}(x^*)$ and $\lambda^*_{I,j}$ has to be zero."