

OPTIMAL CONTROL

XVII century, Bernoulli (1696): ground shape to let ball go from A to B in minimum time? **Brachistochrone curve**

- How to accelerate car to reach goal in min time/energy with max speed

CLASSIC CONTROL:

- need reference trajectory
- rise time
- overshoot
- settling time
- hard to handle constraints

O.C.:

- no need for ref. traj.
- objective clearly specified through cost function
- can handle constraints

Running cost

$O \subset P$

minimize
 $x(\cdot), u(\cdot)$

T

$$\int_0^T \ell(x(t), u(t), t) dt$$

$$+ \ell_f(x(T))$$

Terminal cost

subject to

$$\dot{x}(t) = f(x(t), u(t), t)$$

Dynamics

$$\forall t \in [0, T]$$

$$x(0) = x_0$$

Initial conditions

$$g(x(t), u(t), t) \leq 0$$

$$\forall t \in [0, T]$$

Path constraints (e.g., actuator limits)

- $x(\cdot), u(\cdot)$ are trajectories $\Rightarrow$ infinite-dimensional objects
- constraints are infinitely many!
- $\neq$ optimization problem

OCP EXAMPLE: PENDULUM

$$\underset{x(\cdot), u(\cdot)}{\text{minimize}} \quad \int_0^T (q(t)^2 + \dot{q}(t)^2 + \tau(t)^2) dt + 10^2 q(T)^2$$

$$\text{subject to} \quad I \ddot{q} = \tau + m g \sin(q) \Rightarrow \underbrace{\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} \dot{q} \\ \bar{I}^{-1}(\tau - m g \sin(q)) \end{bmatrix}}_{f(x, u)} \quad \forall t$$

$$|\tau(t)| \leq \tau^{\max} \quad \forall t \in [0, T]$$

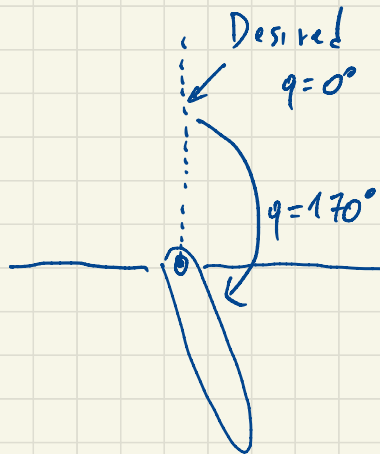
$$\forall t \in [0, T]$$

$$q^{\min} \leq q(t) \leq q^{\max}$$

$$\forall t \in [0, T]$$

$$x = (q, \dot{q})$$

$$u = \tau$$



O.C. METHOD FAMILIES

OPTIMIZE $\Rightarrow$ DISCRETIZE

Continuous Time

DISCRETIZE $\Rightarrow$ OPTIMIZE

Discrete Time

Global

②

Hamilton-Jacobi-Bellman
(HJB)

①

Dynamic Programming
(DP)

Local

③

- Pontryagin Maximum Principle (PMP)
- Calculus of variations
- Indirect method

④

- Direct method

OPTIMAL CONTROL : DIRECT VS INDIRECT METHODS

INDIRECT METHODS: analytically derive necessary optimality conditions $\rightarrow$ solve a boundary-value problem

DIRECT METHODS: first discretize the continuous time problem transforming it in an NLP (Transcription) $\rightarrow$ solve numerically the associated NLP to find a local optimum

- $\rightarrow$ work well in practice: can use off-the-shelf solvers
- $\rightarrow$ flexible: can deal with different constraint types
- $\rightarrow$ $\exists$ multiple ways to do transcription

NON LINEAR PROGRAMMING

$$\min_x f(x)$$

x

$$\text{s.t. } g(x) = 0$$

$$h(x) \leq 0$$

general form of optim. problem

- feasible set: $\Omega \triangleq \{x \mid g(x) = 0, h(x) \leq 0\}$
- assume f, g, h have continuous 1st and 2nd order derivatives
- Let's define optimality conditions in face of constraints

RECAP UNCONSTRAINED CASE

- 1st order **necessary** conditions of optimality
 x^* is local optimum (stationary point) $\Rightarrow \nabla f(x^*) = 0$
- 2nd order **sufficient** conditions of optimality

$\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) > 0 \Rightarrow x^*$ strict local minimizer

$\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) < 0 \Rightarrow x^*$ strict local maximizer

$\nabla^2 f(x^*)$ indefinite $\Rightarrow$ nothing can be said

ONLY EQUALITY CONSTRAINTS

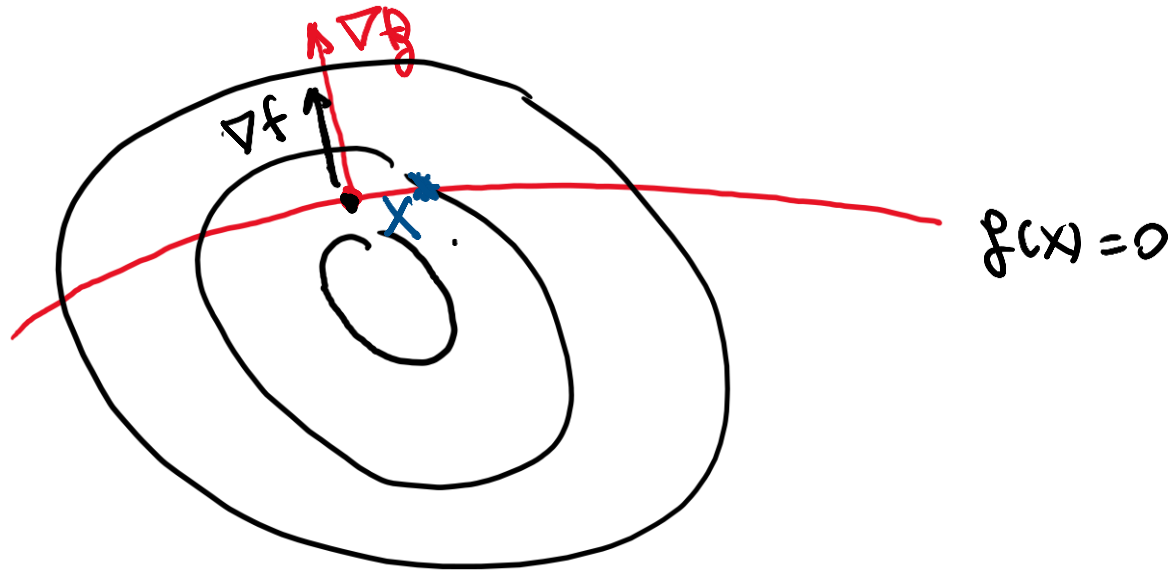
$$\rightarrow d \in \text{Null}(\nabla g(x)^T)$$

- Feasible cone: $F(x) = \{d \mid \nabla g(x)^T d = 0\}$
 - $\rightarrow$ directions $\perp \nabla g$ moving do not affect constraints (i.e. $g(x) = 0$)
- Set of **descent** directions: $D(x) = \{d \mid \nabla f(x)^T d < 0\}$
 - $\rightarrow$ directions along which $f(x)$ decreases
- Local minimum $x^* \Rightarrow D(x^*) \cap F(x^*) = \emptyset$

(1) $\forall d \in F(x) \quad \nabla f(x)^T d \geq 0 \rightarrow$ all directions along constraints increase $f(x)$
($d \in \text{Null}(\nabla g(x)^T)$)

- (1) can be reformulated as :

$$\nabla f(x) \in \text{Range}(\nabla g(x)) \iff \exists \lambda \in \mathbb{R}^m \text{ st } \nabla f(x) = \nabla g(x)\lambda$$



↳ lagrange multiplier

KARUSH-KUHN-TUCKER (KKT) CONDITIONS:

$$\begin{cases} g(x)=0 & \rightarrow \text{feasibility} \\ \nabla f(x) = \nabla g(x)\lambda & \rightarrow \text{local optimum (stationary point)} \end{cases}$$

- we introduce Lagrangian function:

$$\mathcal{L}(x, \lambda) = f(x) + \underbrace{\lambda^T g(x)}$$

~ adding a penalty function when there are constraint violations

- with Lagrangian we roll back to the unconstrained case:

$$\nabla \mathcal{L}(x, \lambda) = 0$$

$$\begin{array}{l} \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{array} \rightarrow \left\{ \begin{array}{l} \nabla f(x) + \nabla g(x) \lambda = 0 \\ g(x) = 0 \end{array} \right.$$

KKT CONDITIONS

↓
system of
non linear equations
↓
solve with Newton
method

INEQUALITY CONSTRAINTS

- Define set of **active constraints** : $A(x) = \{i \mid h_i(x) = 0\}$
(constraints satisfied with 0 margin)

- extended feasible cone:

$$F(x) = \{d \mid \nabla g(x)^T d = 0, \forall h_i(x)^T d \leq 0, i \in A(x)\}$$

KKT CONDITIONS GENERALIZED

$$\begin{cases} f(x) = 0, h(x) \leq 0 \rightarrow \text{feasibility} \\ \nabla f(x)^T d \geq 0 \quad \forall d \in F(x) \rightarrow \text{local stationary point} \end{cases}$$

Farkas's lemma $\rightarrow$ Lagrange multipliers relative to active inequality constraints

$$\begin{cases} \nabla f(x) + \nabla g(x) \lambda + \nabla h_A(x) \mu_A = 0 \\ \mu_A \geq 0 \rightarrow \text{unilaterality} \end{cases}$$

• ISSUE : μ a vector can change dimension

↳ Introduce **complementarity condition**

$$\Leftrightarrow \begin{cases} \nabla f(x) + \nabla g(x) \lambda + \nabla h(x) \mu = 0 \\ \mu \geq 0 \\ \mu^T h(x) = 0 \end{cases}$$

$\rightarrow$ if $h_i(x) = 0 \Rightarrow \mu_i > 0 \rightarrow$ only active play a role
 $\rightarrow$ if $h_i(x) < 0 \Rightarrow \mu_i = 0$

• Redefine The Lagrangian

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x)$$

SUMMARY

$$\min_x f(x)$$

$$\text{s.t. } g(x) = 0 \\ h(x) \leq 0$$

$\Rightarrow$ KKT system

$$\begin{cases} \nabla_x \mathcal{L}(x, \lambda, \mu) = 0 \\ g(x) = 0 \\ \mu(x) \leq 0 \\ \mu \geq 0 \\ \mu^T h(x) = 0 \end{cases}$$

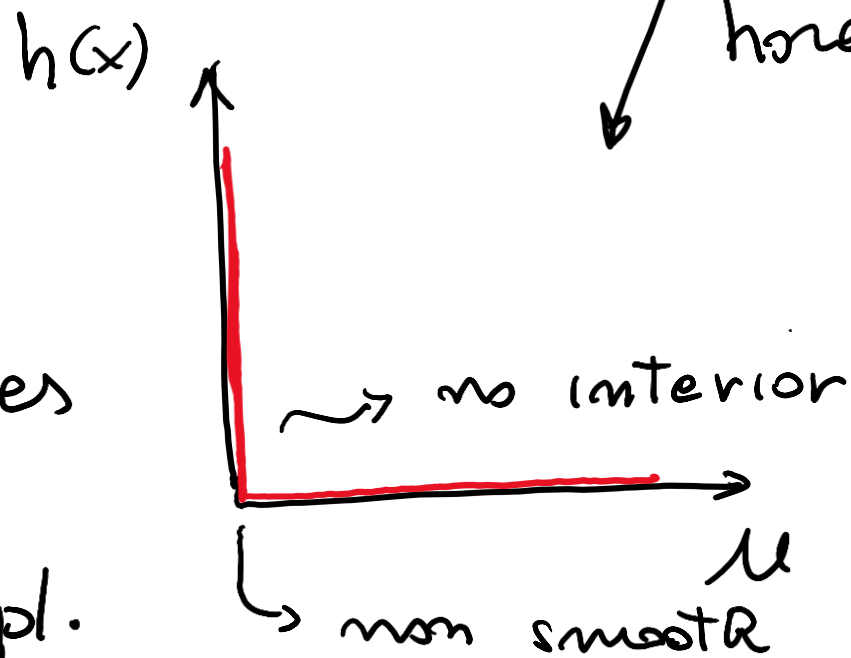
Two approaches:

- Active set methods:

find which constraints
are active $\rightarrow$ Transform
problem with only equalities
 $\rightarrow$ Newton method

- Interior point: relax compl.

constraint $\mu^T h(x) = 0 \Rightarrow \mu^T h(x) \leq \epsilon$



- with only equalities we need to solve a nonlinear system of equations $\rightarrow$ Newton's method

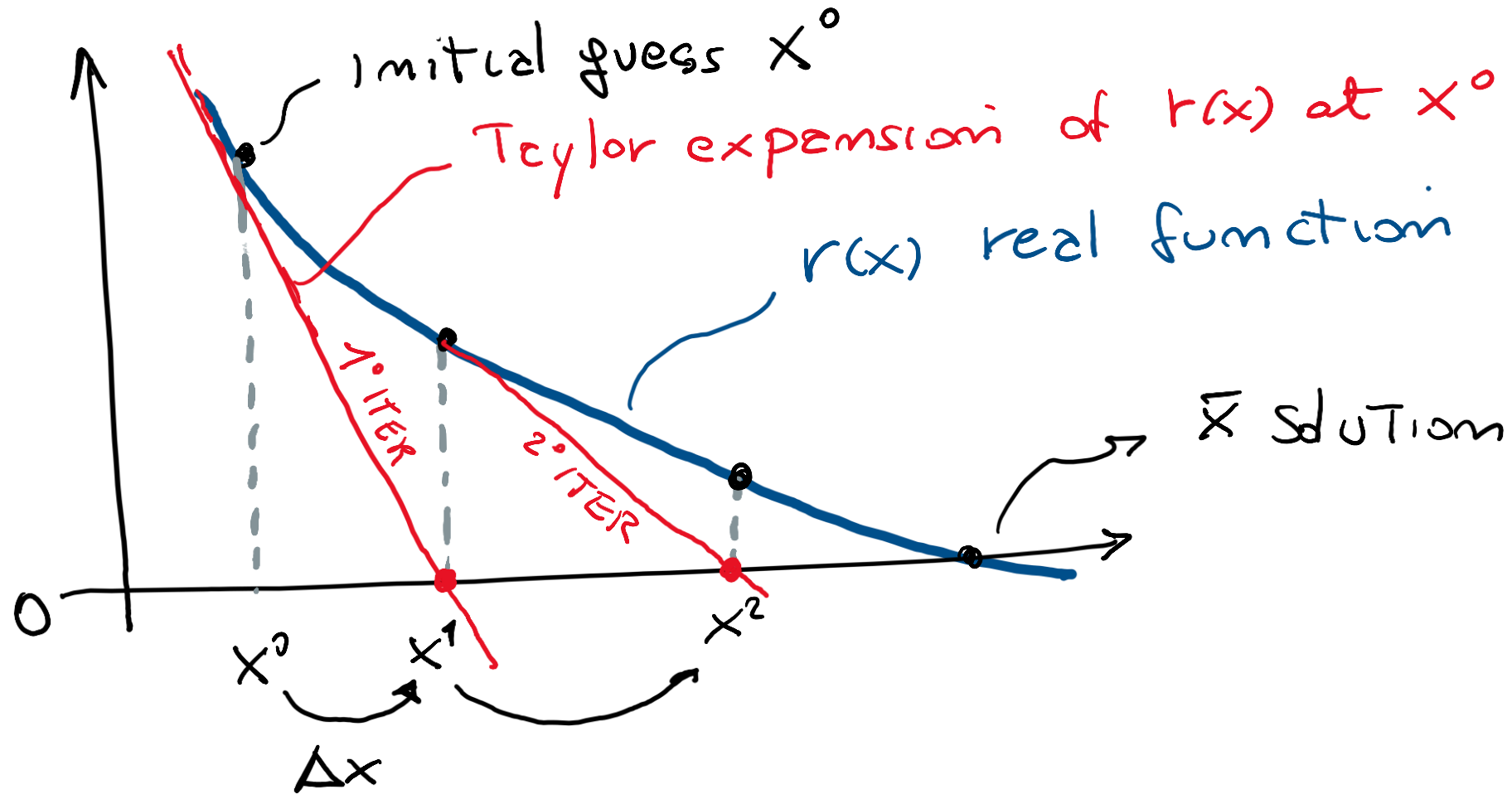
NEWTON METHOD

- $r(x) = 0$ nonlinear system we want to solve
current guess $\bar{x}$
- take a linear approximation of $r(x)$ at $\bar{x}$ (1st order Taylor expansion)

$$r(\bar{x} + \Delta x) \approx r(\bar{x}) + \nabla r(\bar{x})^T \Delta x \rightarrow \text{linear set of equations}$$
- $\Delta x = - \underbrace{\nabla r(\bar{x})^{-T}}_{\text{invertible?}} r(\bar{x}) \rightarrow \text{Newton's step}$
- apply Newton's step to update current guess

$$x^{k+1} = x^k + \Delta x \rightarrow \text{Iterate until } \|r(x)\| \leq \text{threshold}$$

SCALAR EXAMPLE



- Newton method updates The initial guess To The root of The Taylor expansion

SOLVE KKT WITH NEWTON

$$\rightarrow f(x) + \lambda^T g(x)$$

$$\begin{cases} \nabla f + \nabla g \lambda = 0 \\ g(x) = 0 \end{cases} \Leftrightarrow r(x, \lambda) = \begin{bmatrix} \nabla_x \mathcal{L}(x, \lambda) \\ g(x) \end{bmatrix} = 0$$

Local optimi
conditions
for a problem
with only eq.
constraints

- linearize around current guess

$$r(\bar{x} + \Delta x, \bar{\lambda} + \Delta \lambda) \simeq r(\bar{x}, \bar{\lambda}) + \nabla r^T \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = 0$$

$$\begin{bmatrix} \underbrace{\nabla_{xx}^2 \mathcal{L}(\bar{x}, \bar{\lambda})}_{\text{Hessian Symmetric}} & \nabla g(\bar{x}) \\ \underbrace{(\nabla g(\bar{x}))^T}_{\text{KKT MATRIX}} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}) \\ -g(\bar{x}) \end{bmatrix}$$

KKT MATRIX $\rightarrow$ symmetric indefinite ($\emptyset$ eig)

OBSERVATIONS

- KKT system $\rightarrow$ local optimum
 - $\rightarrow$ moving locally $f(x)$ does not improve
 - $\rightarrow$ may be bad globally but in practice is connected with good quality of solution
- Solution depends on the initial guess
- caveats: Newton method does not always work
 - inversion of $\nabla f(x)^T \rightarrow$ Regularization
 - linear approx of $r(x)$ is not the real function
 - $\hookrightarrow$ line search

DESCENT DIRECTION & REGULARIZATION

- For simplicity consider unconstrained problem: $\min_x f(x)$

$$\mathcal{L}(x, \lambda) \Rightarrow f(x)$$

- Direction d is a descent direction $\Leftrightarrow \nabla f(x)^T d < 0$

- Newton step in unconstrained case is:

$$r(x) = 0 \Leftrightarrow \nabla f(x) = 0$$

$$\Delta x = -\nabla r(x)^{-T} r(x) \Leftrightarrow \Delta x = -\underbrace{\nabla^2 f(x)^{-1}}_{\text{Hessian } H} \nabla f(x)$$

- is Δx a descent direction?

$$-\nabla f(x)^T H^{-1} \nabla f(x) < 0?$$

- if $H \succ 0 \Rightarrow \Delta x$ is a descent direction $(H \succ 0 \Rightarrow H^{-1} \succ 0)$
 $\hookrightarrow$ converges to local **minimum** eigs do not change

- in non convex optimization we have no guarantee to

$\Rightarrow$ regularize : $H = \nabla^2 f(x) + \epsilon I$

choose $\epsilon > 0$ so that $H \succ 0$ (e.g. $\epsilon \approx \max(-\epsilon_{\min}, 0)$)

- $H \succ 0 \Rightarrow H$ is always invertible
- regularization modifies Newton step Δx such that its direction point in the opposite direction of the gradient
- why not use gradient descent? $\rightarrow$ we don't know how much to step along $-\nabla f(x)$
- Newton step Thanks to 2^o order info in H tells us also the **step length** in that direction

ALTERNATIVE INTERPRETATION OF NEWTON STEP

- Let's reason about cost $f(x)$ rather than $v(x) = \nabla f(x)$
- instead of minimize $f(x)$, minimize a quadratic (2^o order) approximation of $f(x)$ about $\bar{x}$ (current guess)

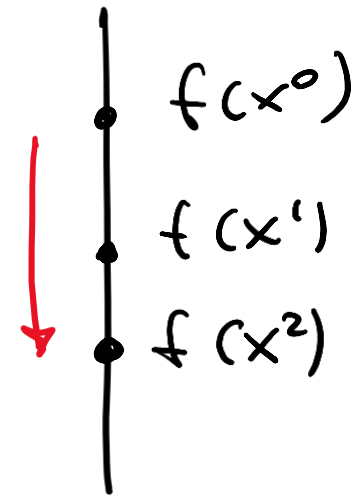
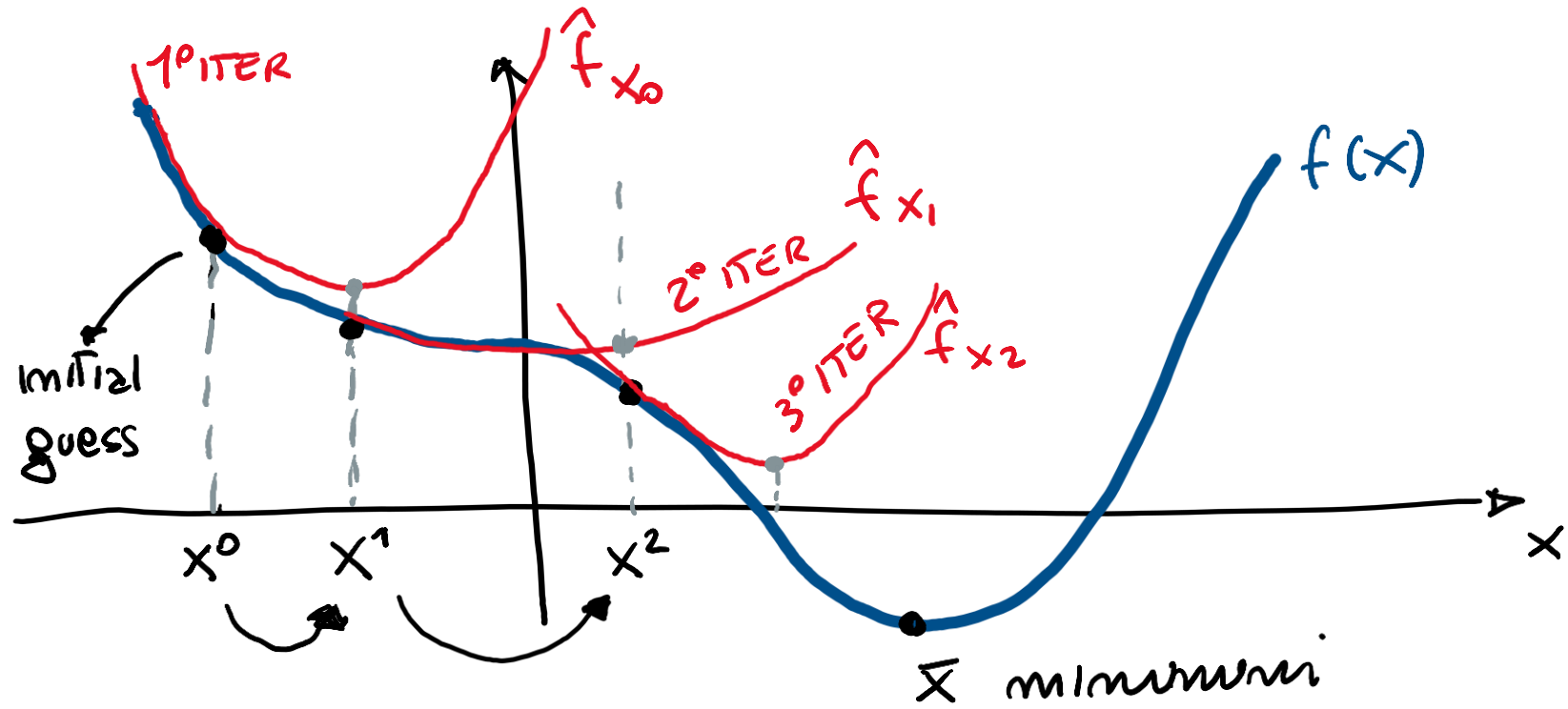
$$(1) \min_{\Delta x} \underbrace{f(\bar{x}) + \nabla f(\bar{x})^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(\bar{x}) \Delta x}_{\hat{f}_{\bar{x}}(\Delta x)}$$

- solving (1) results in The Newton step

$$\nabla f(\bar{x}) + \nabla^2 f(\bar{x}) \Delta x = 0 \Rightarrow \Delta x = -\nabla^2 f(\bar{x})^{-1} \nabla f(\bar{x})$$

- we no longer look for root of $v(x)$ but for the minimum of $\hat{f}_{\bar{x}}(\Delta x)$

GRAPHICAL EXAMPLE (SCALAR CASE)



f decreases
for each Newton
step

- because $\hat{f}(x)$ is an approximation of $f(x)$ it can happen that moving at its minimum does not bring a decrease in $f(x)$

BACKTRACKING LINE SEARCH

- $\hat{f}(x)$ is a local approx of $f(x)$, valid on a neighborhood of the current solution
- $\nabla f(x)^T d < 0$ is a **local** requirement of being a descent direction
- Take a fraction of the Newton step along its direction
To ensure there is a decrease of $f(x)$

$$x^{k+1} = x^k + \alpha \Delta x \quad \text{step length } \alpha \in [0, 1]$$

$\hookrightarrow \alpha = 1$ Standard Newton

- The line search does not change the direction
- how to choose α ?

PSEUDO-CODE

INPUT: x^k

OUTPUT: x^{k+1}

① eval $f(x)$ at current guess: $f = f(x^k)$

② compute Newton step: $\Delta x = -(\nabla^2 f + \epsilon I)^{-1} r$ ($r = \nabla f$)

③ $x_{try} = x^k + \alpha \Delta x$, $\alpha = 1$

reduction = $f - f(x_{try})$

while (reduction < $\gamma \alpha \nabla f^T \Delta x$)

$\alpha = \beta \alpha$

$x_{try} = x^k + \alpha \Delta x$

reduction = $f - f(x_{try})$

④ $x^{k+1} = x_{try}$

Hyperparameters

- $\beta \in [0, 1] \rightarrow \approx 0,5$

- $\gamma \in [0, 1]$

- $\epsilon > 0$

OBSERVATIONS

- we modified Newton method to be applied to optimization
- we are finding a minimum of $f(x)$ not only a root of $r(x) = \nabla f$
- The condition to check for convergence should be on $f(x)$