ROB-GY 6323
reinforcement learning and optimal control for robotics

Lecture 4
Nonlinear Optimal Control

Course material

All necessary material will be posted on **Brightspace**

Code will be posted on the **Github** site of the class

https://github.com/righetti/optlearningcontrol

Discussions/Forum with Slack

Course Assistant
Armand Jordana
aj2988@nyu.edu
Office hours Monday 1pm to 2pm
Rogers Hall 515

Reminder

- **Homework 1 is due on Sep 28, 2024 11:59 PM**

- **Homework 2 out soon**

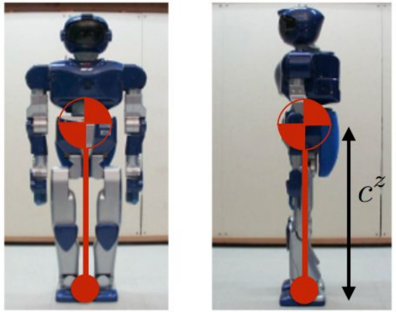
Recap on QP

$$\min_x \frac{1}{2} x^T P x + q^T x$$

$$\text{subject to} \quad Ax = b$$

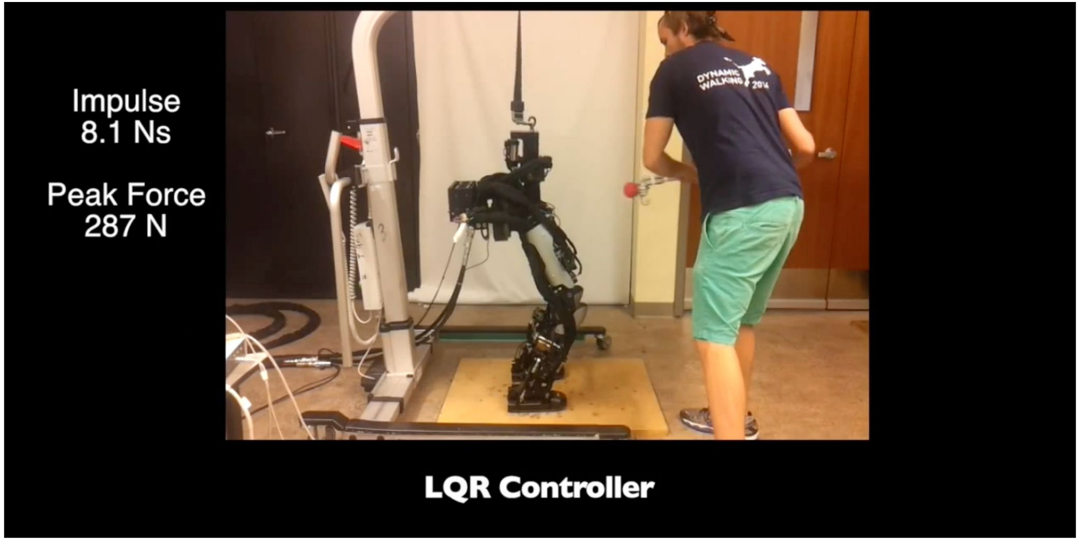
$$Gx \leq h$$

$$\ddot{\mathbf{c}}^{x,y} = \frac{g}{c^z}(\mathbf{c}^{x,y} - \underbrace{\mathbf{p}^{x,y}}_{\text{CoP}})$$

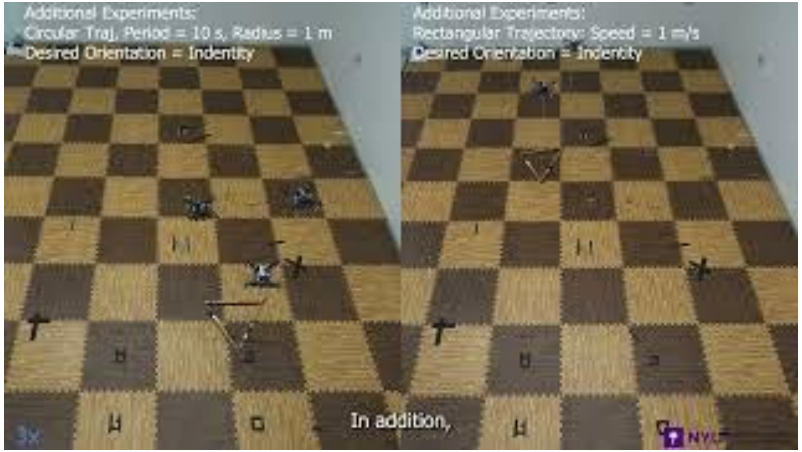


LIPM

Recap on QP



We need nonlinear formulations



Structure of an optimal control problem

$$\min_{x_1,\ldots,x_T,u_0,\ldots,u_{T-1}} \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$$

$$\begin{aligned} \text{subject to } x_{t+1} &= f(x_t, x_t) \\ h_t(x_t, u_t) &\leq 0 \\ h_T(x_T) &\leq 0 \end{aligned}$$

Describe the cost function

Unconstrained optimization

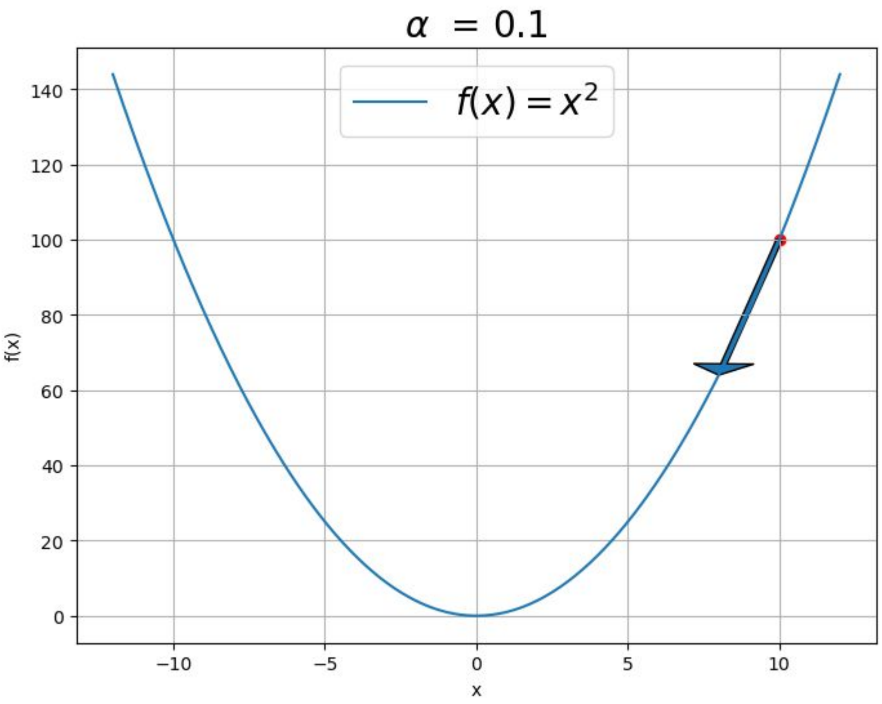
$$\min_{x} f(x)$$

- Cannot rely on closed form solution.

e.g. $f(x) = \exp(\cos(x))(\sin(\cos(x)) + 1)^3$

- Need a (local) iterative method.

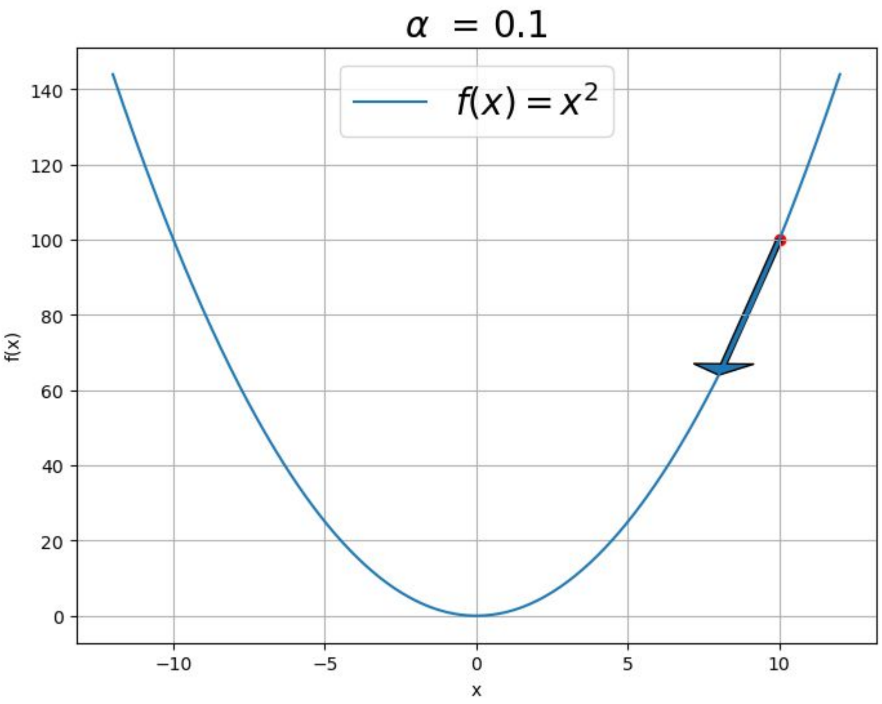
What is a descent direction?



$$x \leftarrow x + p$$

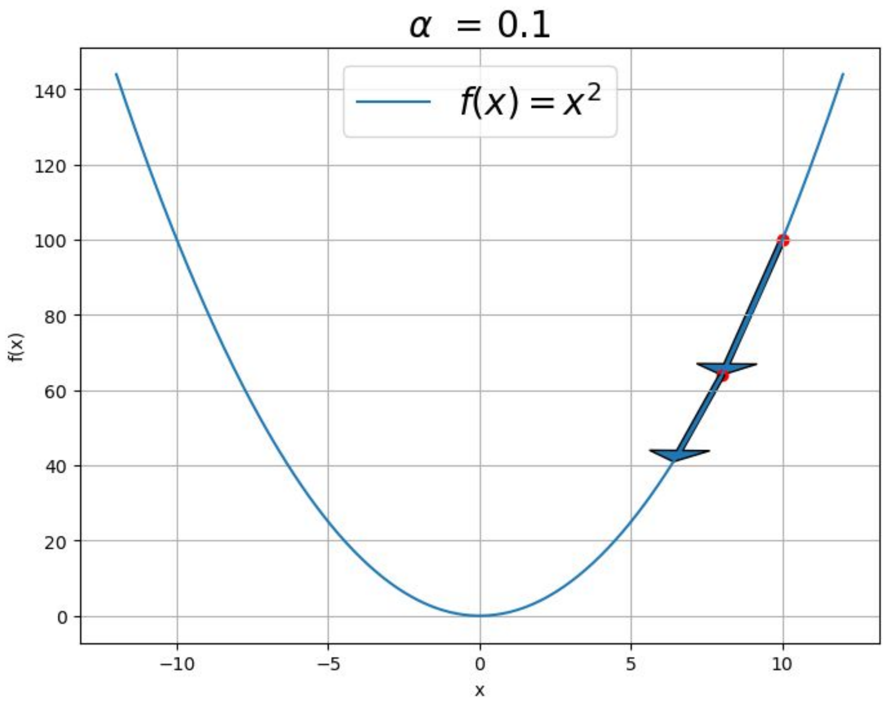
$$p^T \nabla f < 0$$

Gradient descent



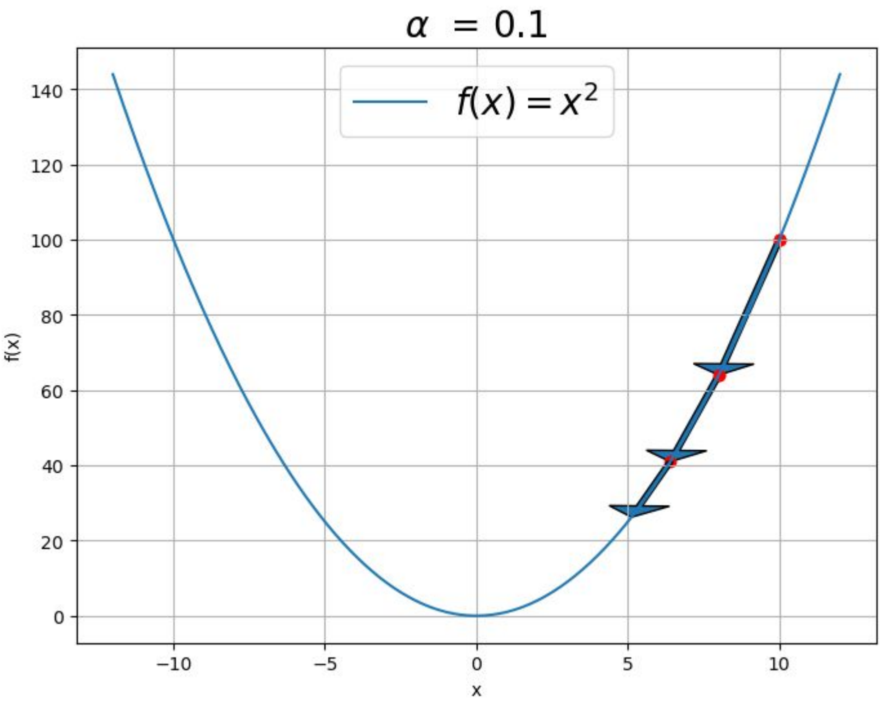
$$x \leftarrow x - \alpha \nabla f$$

Gradient descent



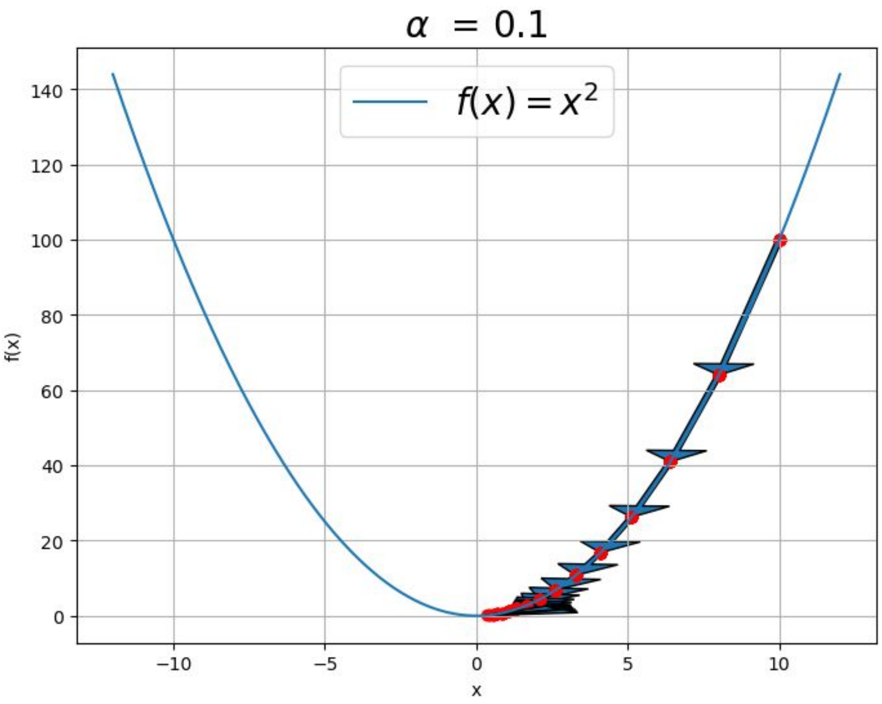
$$x \leftarrow x - \alpha \nabla f$$

Gradient descent



$$x \leftarrow x - \alpha \nabla f$$

Gradient descent

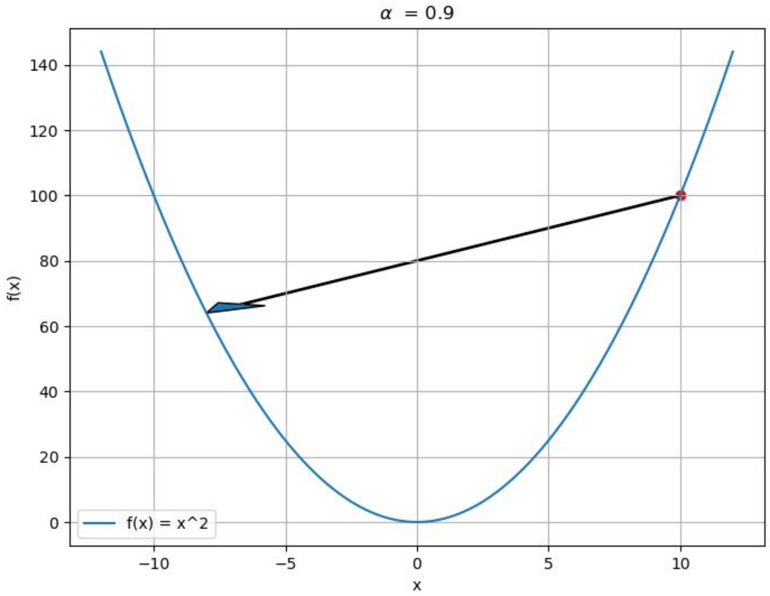


$$x \leftarrow x - \alpha \nabla f$$

What if we increase α?

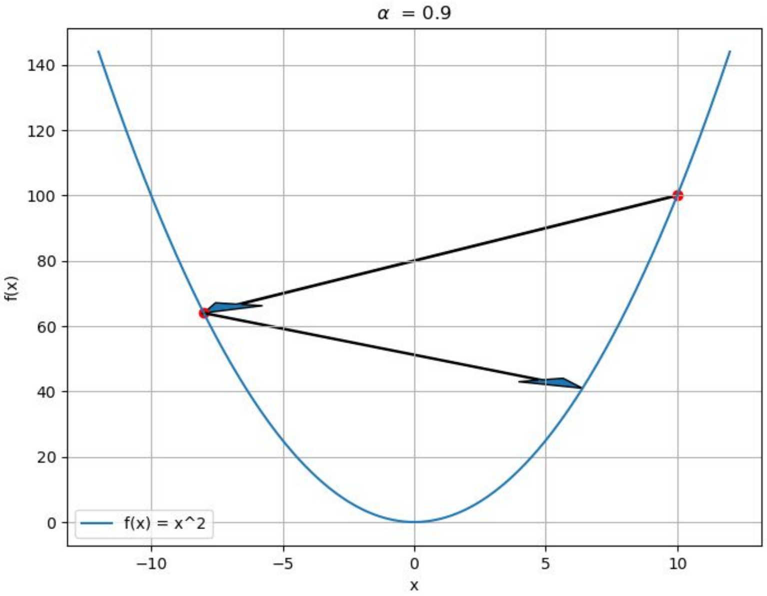
$$x \leftarrow x - \alpha \nabla f$$

What if we increase α?



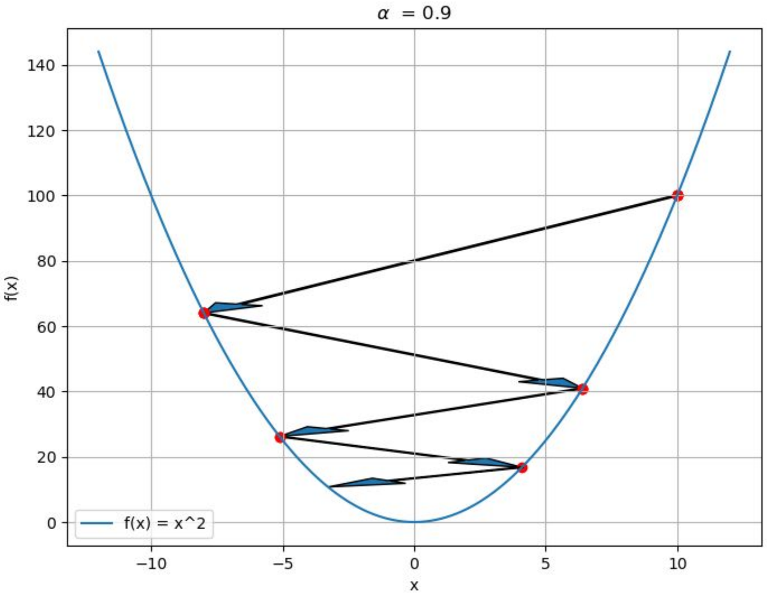
$$x \leftarrow x - \alpha \nabla f$$

What if we increase α?



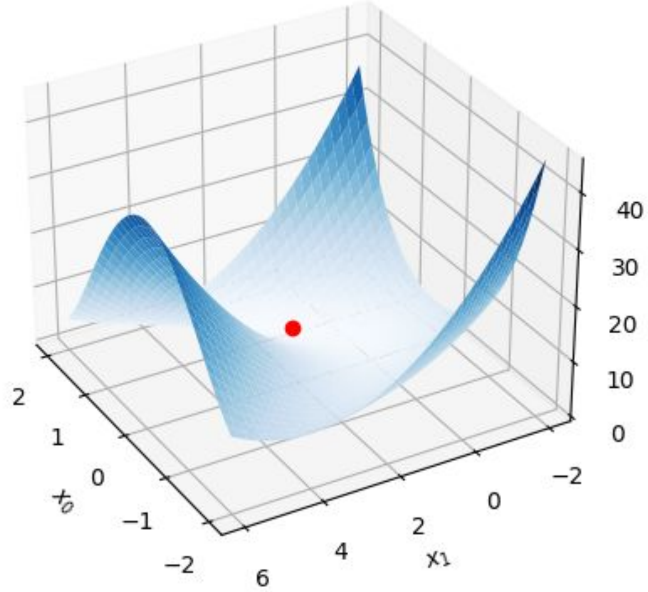
$$x \leftarrow x - \alpha \nabla f$$

What if we increase α?



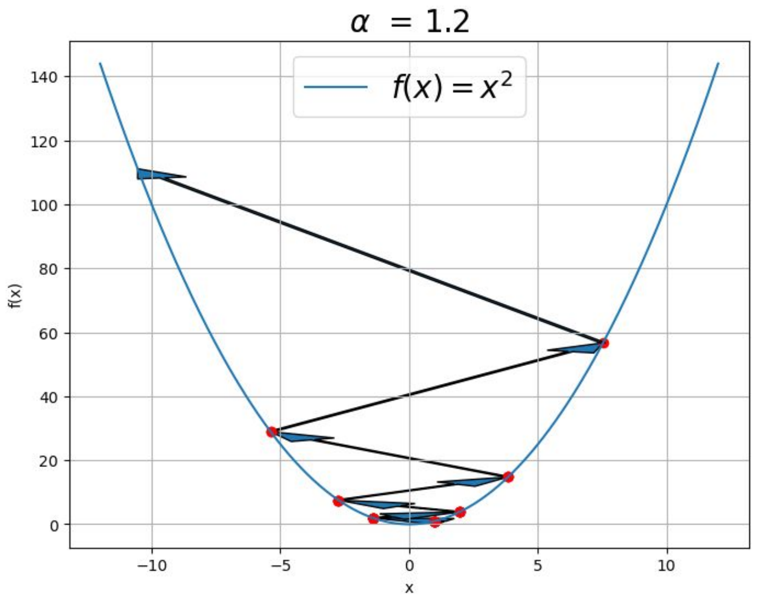
$$x \leftarrow x - \alpha \nabla f$$

The Rosenbrock function



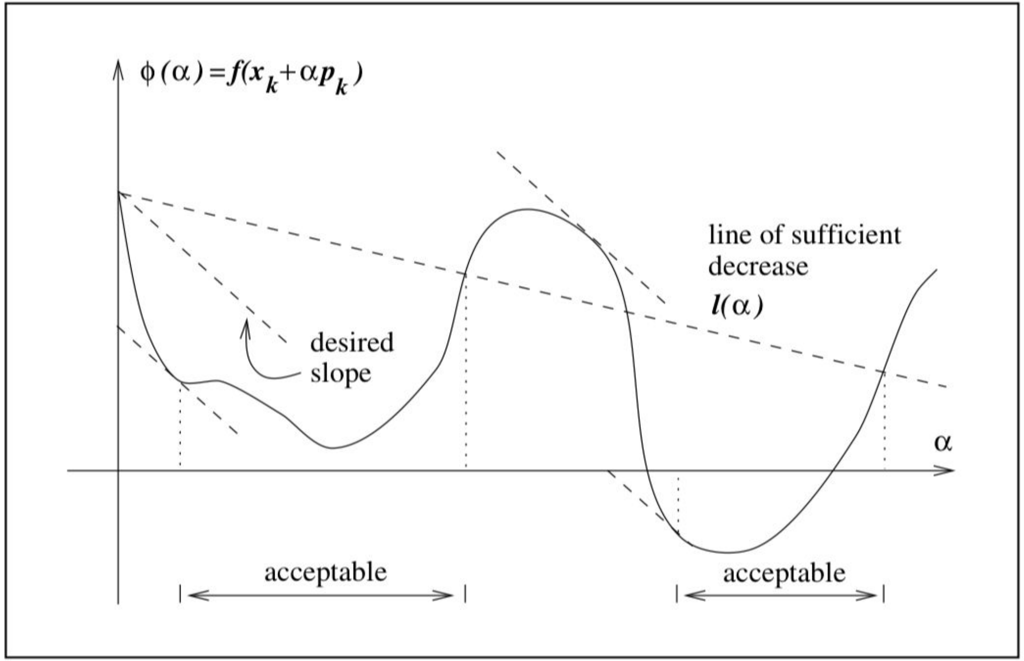
$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

What if α is too large?



We need an automated way to chose α

Line search



$\phi(\alpha) = f(x_k + \alpha p_k)$
line of sufficient
decrease
$l(\alpha)$
desired
slope
α
acceptable
acceptable

In practice

Algorithm 3.1 (Backtracking Line Search).

Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;

repeat until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$

$\alpha \leftarrow \rho\alpha$;

end (repeat)

Terminate with $\alpha_k = \alpha$.

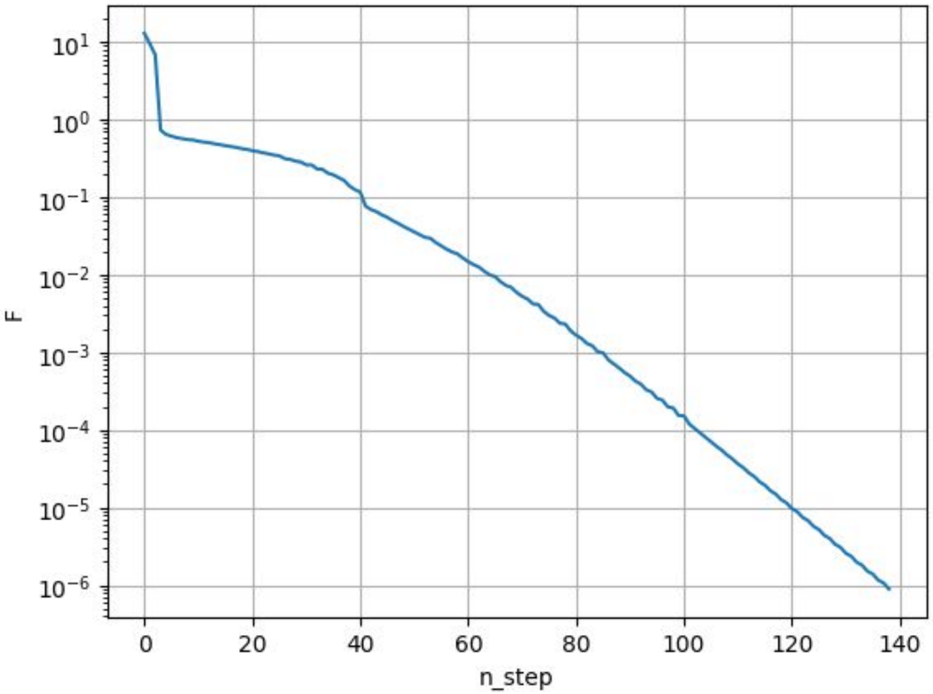
Theorem

Assuming the gradient is Lipschitz continuous then, the iterates of gradient descent converge to a stationary point:

$$\lim_{k \to \infty} \|\nabla f_k\| = 0.$$

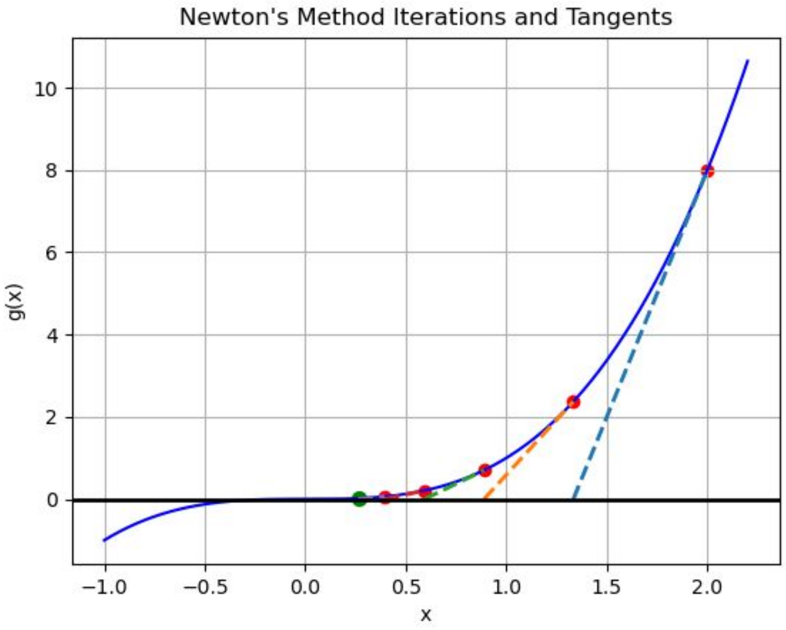
- Does not guarantee to find a minima
- Will always be local

Convergence rate of gradient descent: Linear



Newton method

$$\text{Goal: Find } x \text{ such that } g(x) = 0$$



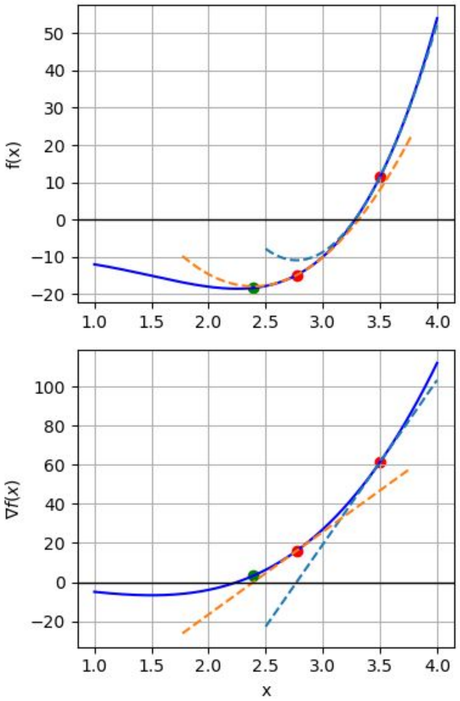
Application

Derive an algorithm to approximate $\sqrt{2}$

Newton method for optimization

$$\min_x f(x)$$

Let's find the zero of $\nabla f(x)$



Newton method for optimization

$$x_{k+1} = x_k + \alpha_k p_k$$
$$p_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

Under what condition is this a descent direction?

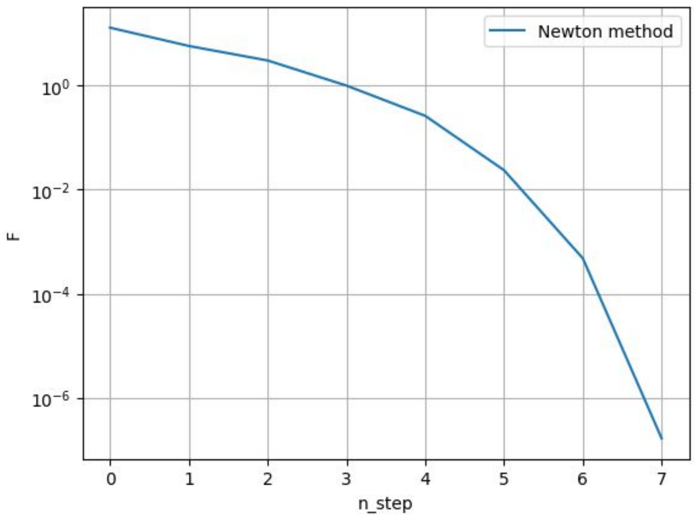
$$\nabla^2 f(x_k) \succ 0$$

Newton method on a quadratic function.

$$f(x) = x^T Q x + q^T x$$

- What is the minimum of f?
- Derive a Newton step

Convergence rate: Quadratic



Nonlinear constrained optimization

$$\min_x f(x) = 0$$
$$\text{subject to } g(x) = 0$$

Recall the KKT condition:

$$\nabla f(x) + \lambda \nabla g(x) = 0$$
$$g(x) = 0$$

Sequential Quadratic Programming (SQP): Apply Newton's method on the KKT

Sequential Quadratic Programming (SQP)

$$x_{k+1} = x_k + p_k$$

where:

$$\min_p \; p^T \nabla^2 f(x_k) p + p^T \nabla f(x_k)$$

$$\text{subject to } \nabla g(x_k)^T p + g(x_k) = 0$$

Example

$$\min_x f(x) = 0$$
$$\text{subject to } g(x) = 0$$

$$f(x) = x^T Q x + q^T x$$
$$g(x) = Ax + b$$

- How to find the solution?

- Derive an SQP

Sequential Quadratic Programming (SQP)

$$\min_x f(x)$$
$$\text{subject to } g(x) = 0$$

Step 1: Find a direction

$$\min_p \; p^T \nabla^2_{xx} \mathcal{L}(x_k) p + p^T \nabla f(x_k)$$
$$\text{subject to } \nabla g(x_k)^T p + g(x_k) = 0$$

Step 2: Find a step length α_k with a line search

Merit function

$$\phi(x) = f(x) + \mu||g(x)||_1$$

Sufficient decrease condition

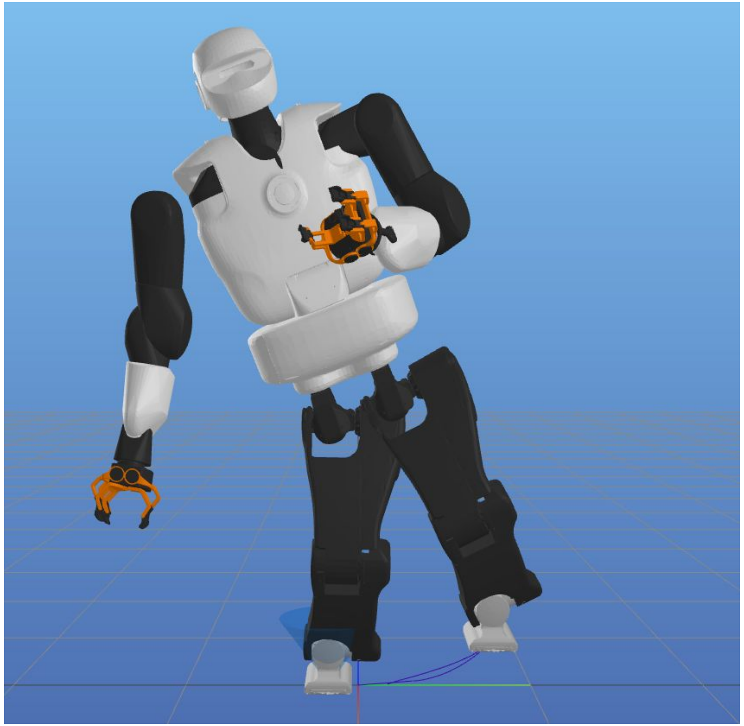
$$\phi(x_k + \alpha_k p_k) \leq \phi(x_k) + \eta\alpha_k(\nabla f_k^T p_k - \mu||g(x_k)||_1)$$

Nonlinear Optimal control

$$\min_{x_1,\ldots,x_T,u_0,\ldots,u_{T-1}} \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$$

$$\text{subject to } x_{t+1} = f(x_t, x_t)$$

Example: Taichi robot



How can we avoid inverting a matrix of size T?

Proposition 2. *By applying the Thomas algorithm, we recover the well-known Riccati recursions. Specifically, the* ***backward pass*** *can be done by initializing $V_T = Q_T$ and $v_T = q_T$, and then by applying the following equations:*

$$
\begin{aligned}
h_k &= r_k + B_k^T(v_{k+1} + V_{k+1}\gamma_{k+1}) \qquad\qquad (9)\\
G_k &= S_k^T + B_n^T V_{k+1} A_k \qquad\qquad K_k = -H_k^{-1} G_k\\
H_k &= R_k + B_k^T V_{k+1} B_k \qquad\qquad k_k = -H_k^{-1} h_k\\
V_k &= Q_k + A_k^T V_{k+1} A_k - K_k^T H_k K_k\\
v_k &= q_k + K_k^T r_k + (A_k + K_k B_k)^T (v_{k+1} + V_{k+1}\gamma_{k+1})
\end{aligned}
$$

Then, the ***forward pass*** *initializes $\Delta x_0 = 0$ and unrolls the linearized dynamics:*

$$\Delta x_{k+1} = (A_k + B_k K_k)\Delta x_k + B_k k_k + \gamma_{k+1} \tag{10a}$$

$$\Delta u_k = K_k \Delta x_k + k_k \tag{10b}$$

$$\lambda_k = V_k \Delta x_k + v_k \tag{10c}$$

Extension to inequality

$$\min_{x_1,\ldots,x_T,u_0,\ldots,u_{T-1}} \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$$

$$\begin{aligned}\text{subject to } x_{t+1} &= f(x_t, x_t)\\ h_t(x_t, u_t) &\leq 0\\ h_T(x_T) &\leq 0\end{aligned}$$

Extension to inequality

$$\min_x f(x)$$
$$\text{subject to } g(x) = 0$$
$$h(x) \leq 0$$

Step 1: Find a direction

$$\min_p \; p^T \nabla^2_{xx} \mathcal{L}(x_k) p + p^T \nabla f(x_k)$$
$$\text{subject to } \nabla g(x_k)^T p + g(x_k) = 0$$
$$\nabla h(x_k)^T p + h(x_k) \leq 0$$

Step 2: Find a step length α_k with a merit function and line search

To solve the inequality QP,

You can use any Black box QP solver and implement your own SQP.

Solvers

Solver	Keyword	Algorithm	API	License	Warm-start
Clarabel	`clarabel`	Interior point	Sparse	Apache-2.0	✗
CVXOPT	`cvxopt`	Interior point	Dense	GPL-3.0	✓
DAQP	`daqp`	Active set	Dense	MIT	✗
ECOS	`ecos`	Interior point	Sparse	GPL-3.0	✗
Gurobi	`gurobi`	Interior point	Sparse	Commercial	✗
HiGHS	`highs`	Active set	Sparse	MIT	✗
HPIPM	`hpipm`	Interior point	Dense	BSD-2-Clause	✓
MOSEK	`mosek`	Interior point	Sparse	Commercial	✓
NPPro	`nppro`	Active set	Dense	Commercial	✓
OSQP	`osqp`	Augmented Lagrangian	Sparse	Apache-2.0	✓
PIQP	`piqp`	Proximal interior point	Dense & Sparse	BSD-2-Clause	✗
ProxQP	`proxqp`	Augmented Lagrangian	Dense & Sparse	BSD-2-Clause	✓
QPALM	`qpalm`	Augmented Lagrangian	Sparse	LGPL-3.0	✓
qpax	`qpax`	Interior point	Dense	MIT	✗
qpOASES	`qpoases`	Active set	Dense	LGPL-2.1	—
qpSWIFT	`qpswift`	Interior point	Sparse	GPL-3.0	✗
quadprog	`quadprog`	Active set	Dense	GPL-2.0	✗
SCS	`scs`	Augmented Lagrangian	Sparse	MIT	✓

https://github.com/qpsolvers/qpsolvers

Nonlinear optimal control libraries

crocoddyl

Contact Robot Optimal Control by Differential Dynamic Library

acados

Sequential implementation of nonlinear solver

S. Wright, J. Nocedal et al., “Numerical optimization,” Springer Science, vol. 35, no. 67-68, p. 7, 1999.

G. Frison and M. Diehl, “Hpipm: a high-performance quadratic programming framework for model predictive control,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 6563–6569, 2020.

H. J. Ferreau, H. G. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit mpc,” International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, vol. 18, no. 8, pp. 816–830, 2008

Wang and S. Boyd, “Fast model predictive control using online optimization,” IEEE Transactions on control systems technology, vol. 18, no. 2, pp. 267–278, 2009.

C. Dohrmann and R. Robinett, “Dynamic programming method for constrained discrete-time optimal control,” Journal of Optimization Theory and Applications, vol. 101, pp. 259–283, 1999.

Contact invariant Optimization

Next week: MPC

Dynamic Constraint Satisfaction

The Sparse Constrained SQP is able to update its solutions online
to satisfy dynamically added constraints.