# Primal-Dual Proximal Augmented Lagrangian Methods for Differentiable Quadratic Programming: Theory & Implementation


Antoine Bambade[1,2]

[1] *Inria and ENS Paris : Willow and Sierra teams*

[2] *École des Ponts Paris Tech*

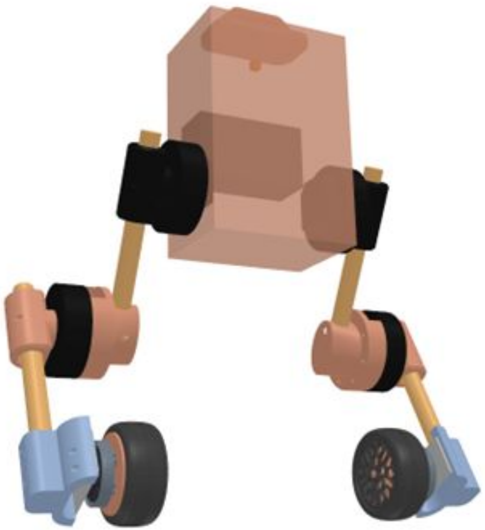

Ph.D defense, 15 January 2024

# Introduction

Antoine Bambade[1,2]
[1]*Inria and ENS Paris : Willow and Sierra teams*
[2]*École des Ponts Paris Tech*

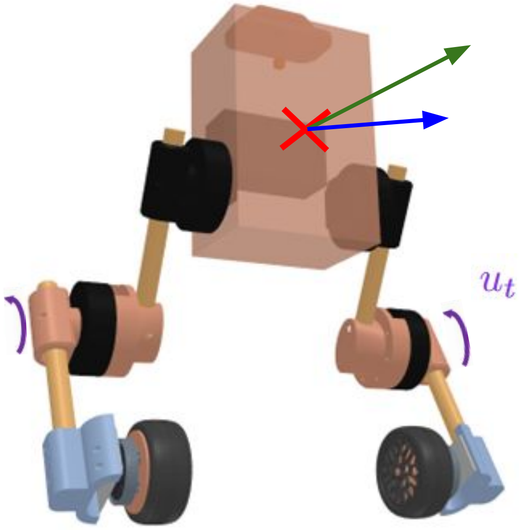
Ph.D defense, 15 January 2024

# Introduction



# Introduction

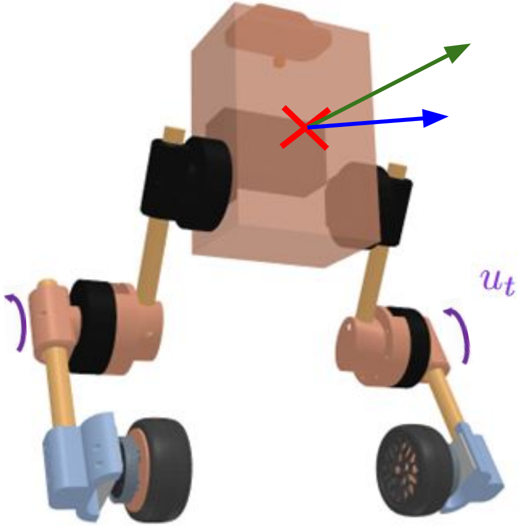
$x_t$ = Center of Mass position, velocity, acceleration



$u_t$ = Control input

# Introduction

$x_t$ = Center of Mass position, velocity, acceleration

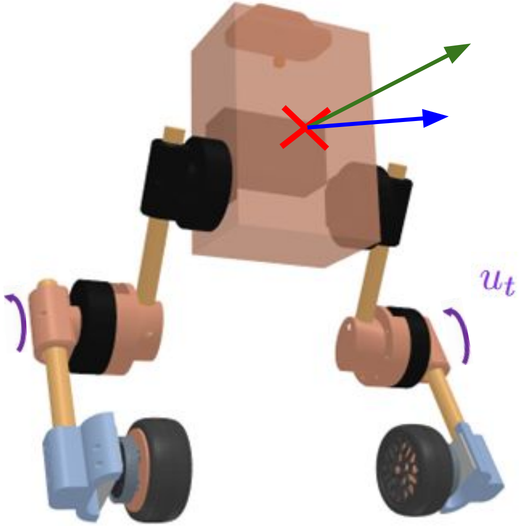


$u_t$ = Control input

$x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t,$ **Dynamic model**

# Introduction

$x_t$ = Center of Mass position, velocity, acceleration



$u_t$ = Control input

**Control trajectory**

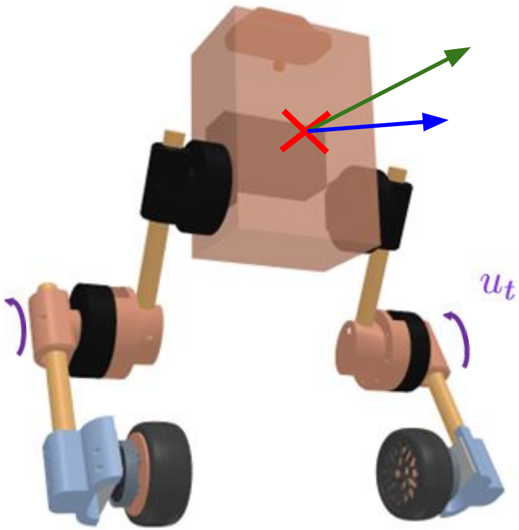
$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \underbrace{\sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2}_{}, $$

$$\text{s.t., } x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

**Dynamic model**

# Introduction

$x_t$ = Center of Mass position, velocity, acceleration



$u_t$ = Control input

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{goal}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{goal}\|_2^2 + w_u \|u_t\|_2^2,$$

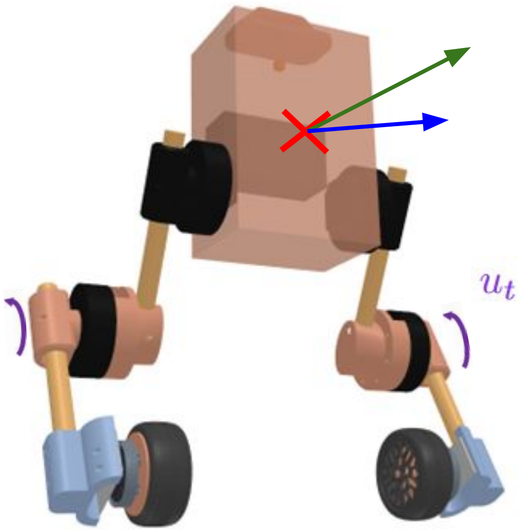
$$\text{s.t.,}\ x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$ **Dynamic model**

$$-x_{max} \le x_t \le x_{max},$$
$$-u_{max} \le u_t \le u_{max}.$$

**Safety constraints**

# Introduction

$x_t = \text{Center of Mass } \text{position}, \text{velocity}, \text{acceleration}$



$u_t = \text{Control input}$

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \underbrace{\sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2}_{}, $$

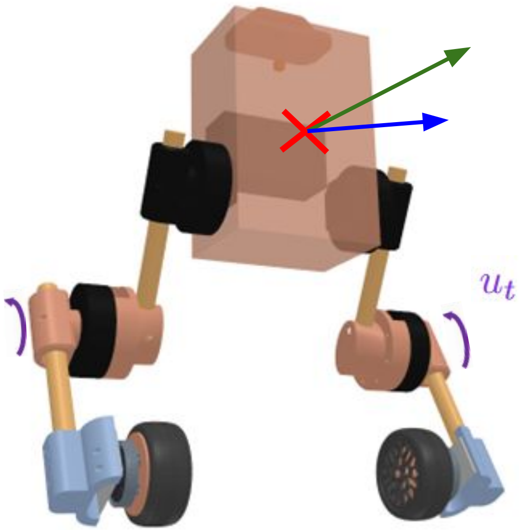
$$\text{s.t.}, \; x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$ Dynamic model

$$-x_{\max} \le x_t \le x_{\max},$$
$$-u_{\max} \le u_t \le u_{\max}.$$
Safety constraints

A convex Quadratic Program

# Introduction

$x_t =$ Center of Mass position, velocity, acceleration



$u_t =$ Control input

**Control trajectory**

$$\overbrace{\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,}$$

$$\text{s.t.,}\ x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$ **Dynamic model**

$$-x_{\max} \leq x_t \leq x_{\max},$$
$$-u_{\max} \leq u_t \leq u_{\max}.$$ **Safety constraints**

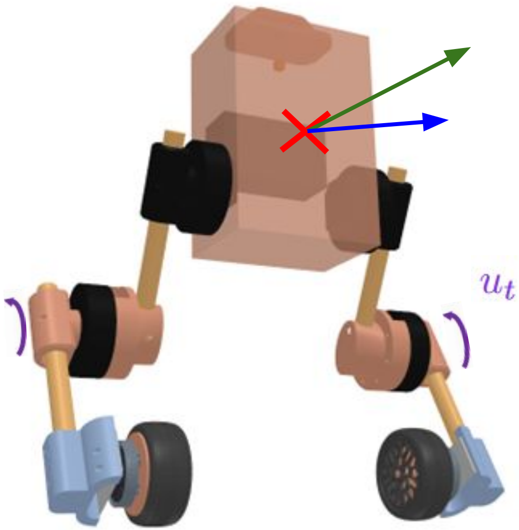
**A convex Quadratic Program**

**Plan**

1. **Solve efficiently these QPs**

# Introduction

$x_t =$ Center of Mass position, velocity, acceleration



$u_t =$ Control input

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t., } x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t, \quad \textbf{Dynamic model}$$

$$\left.\begin{aligned} -x_{\max} \leq x_t \leq x_{\max}, \\ -u_{\max} \leq u_t \leq u_{\max}. \end{aligned}\right\} \textbf{Safety constraints}$$

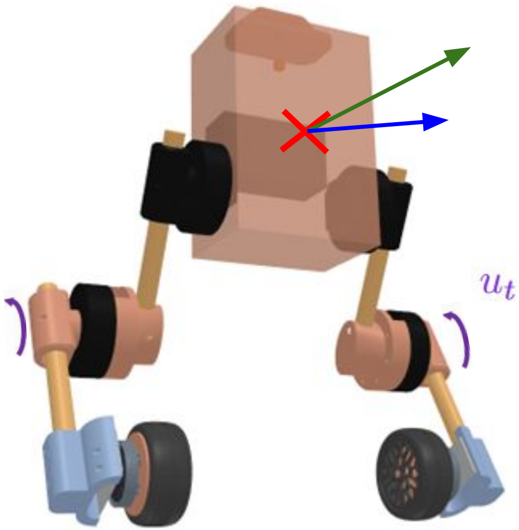
**A convex Quadratic Program**

**Plan**

1. **Solve efficiently these QPs**
2. **Differentiate through solutions**

# Introduction

$x_t =$ Center of Mass position, velocity, acceleration



$u_t =$ Control input

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t.,}\ x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$ **Dynamic model**

$$-x_{\max} \leq x_t \leq x_{\max},$$
$$-u_{\max} \leq u_t \leq u_{\max}.$$
**Safety constraints**

**A convex Quadratic Program**

**Plan**

1. Solve efficiently these QPs : ProxQP
2. Differentiate through solutions : QPLayer

# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

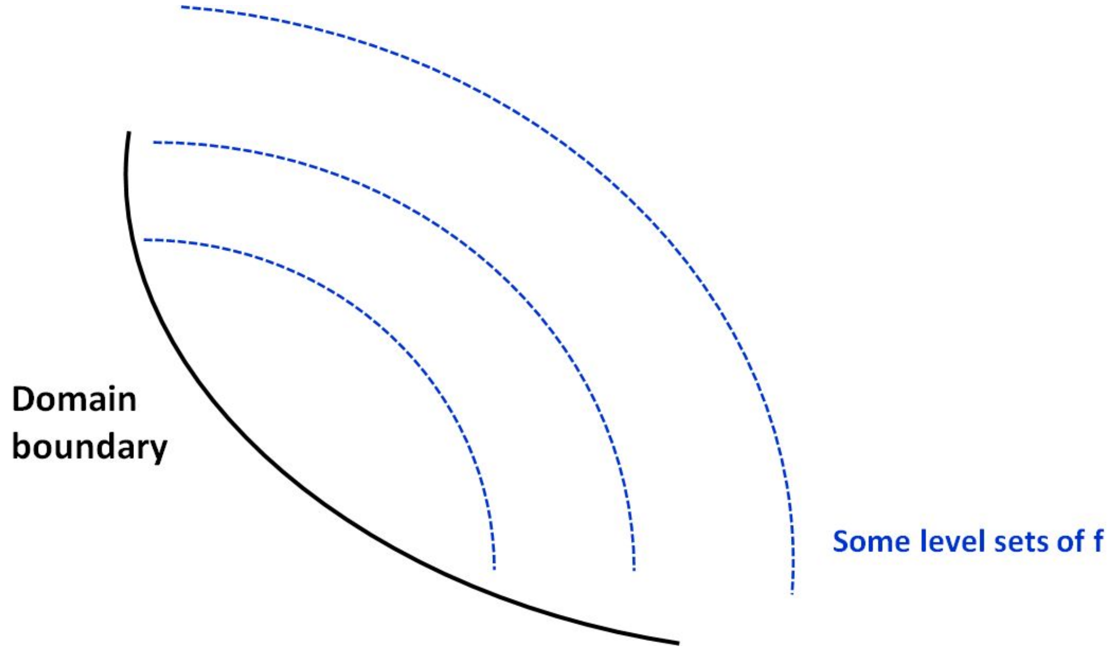
**First ingredient:**

**The proximal point operator**

# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

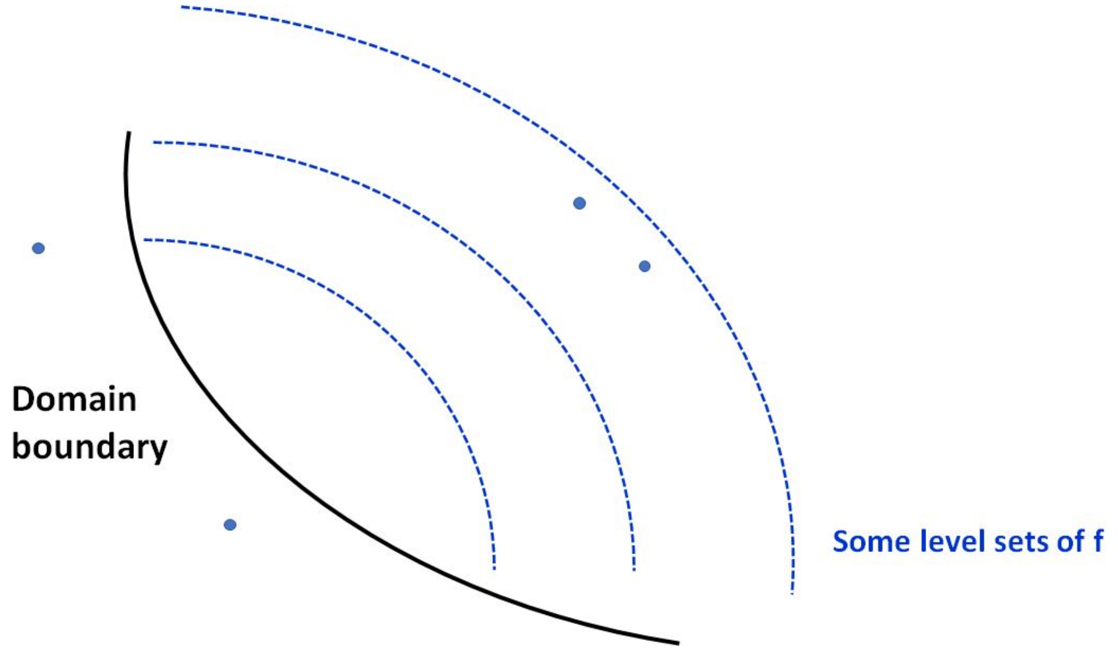
**First ingredient:**
**The proximal point operator**



# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \operatorname*{argmin}_x \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

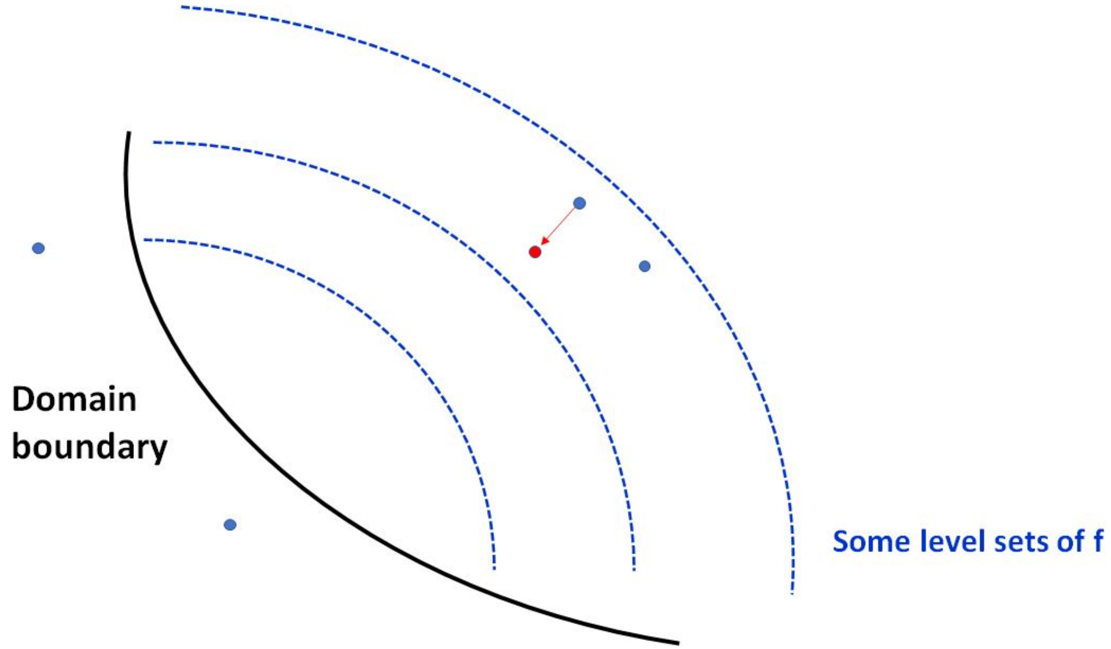
**First ingredient:**
**The proximal point operator**



# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

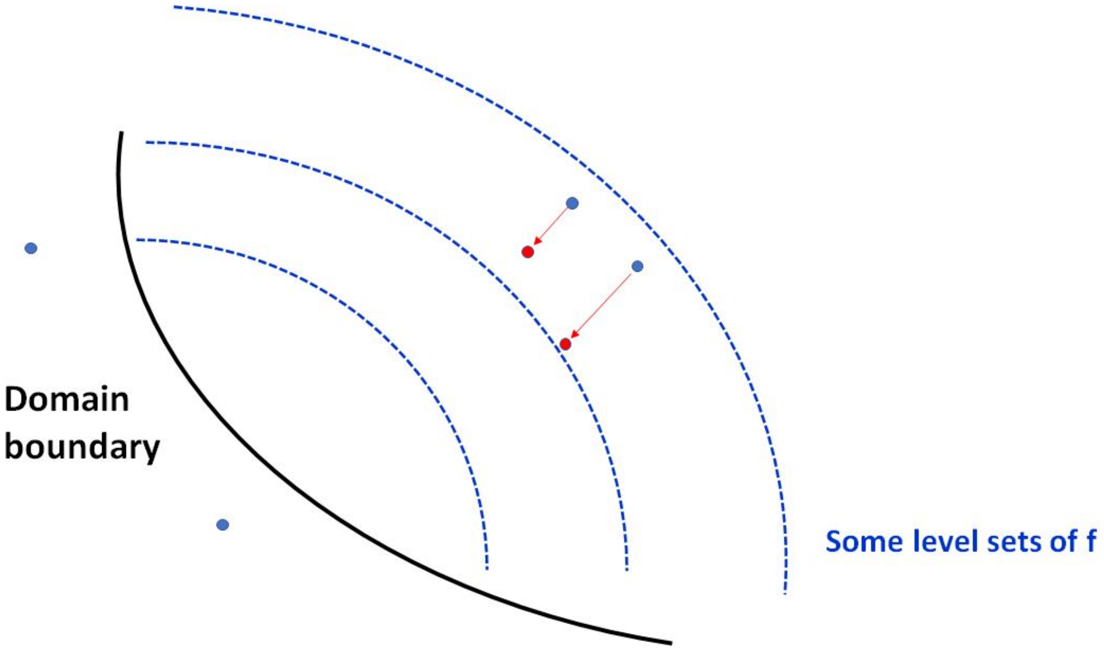
**First ingredient:**
**The proximal point operator**



# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

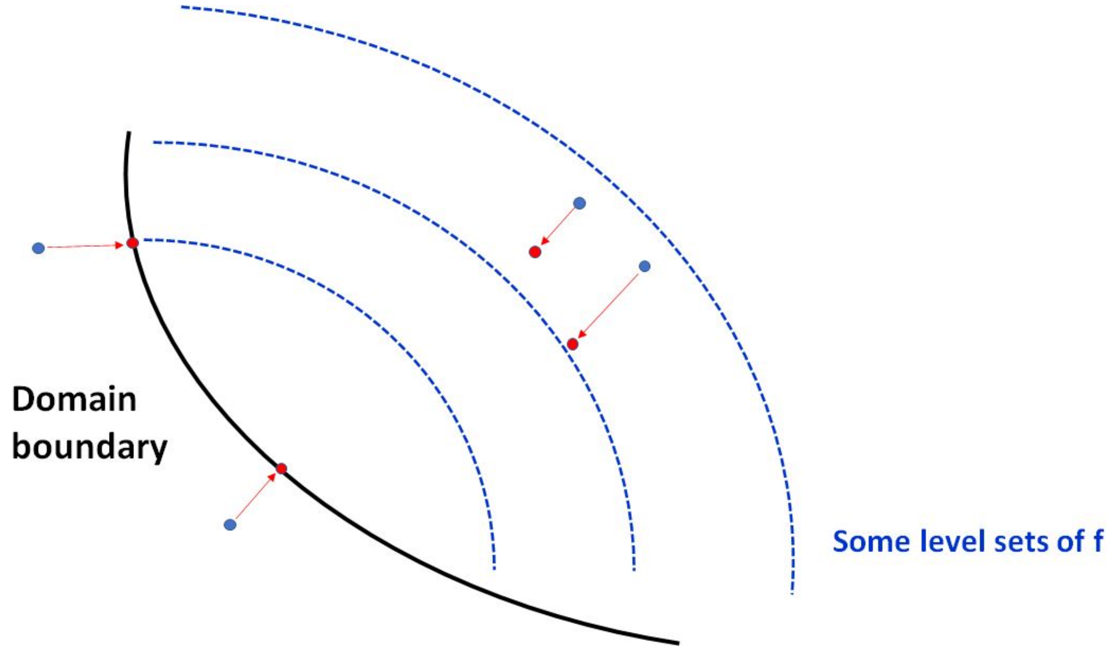
**First ingredient:**
**The proximal point operator**



# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

**First ingredient:**
**The proximal point operator**

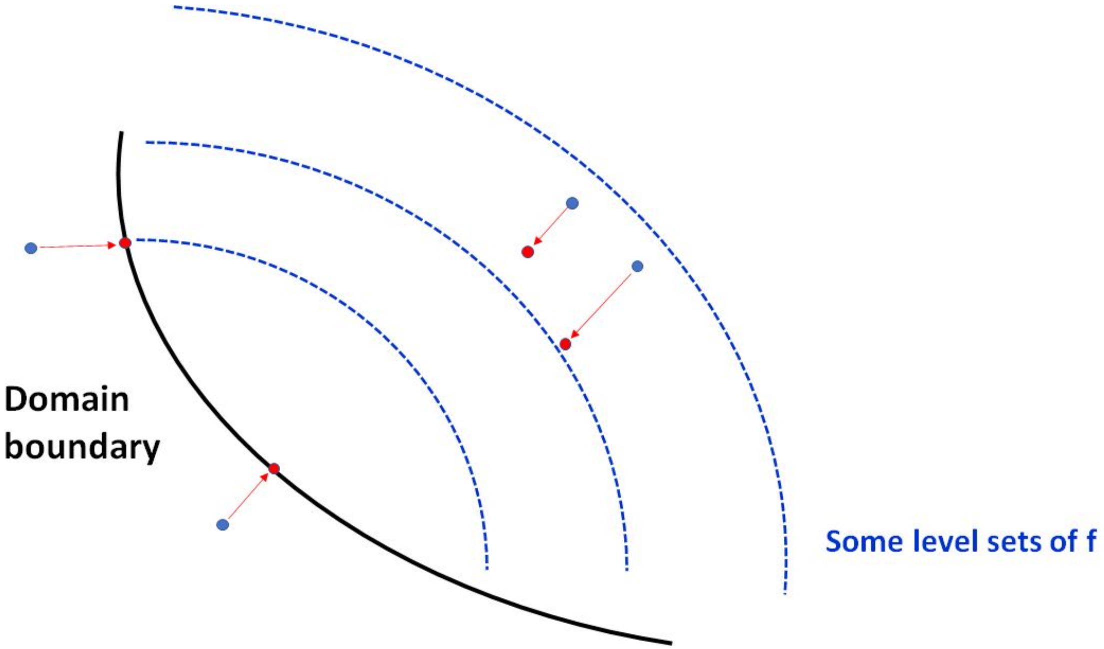


# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

**First ingredient:**
**The proximal point operator**

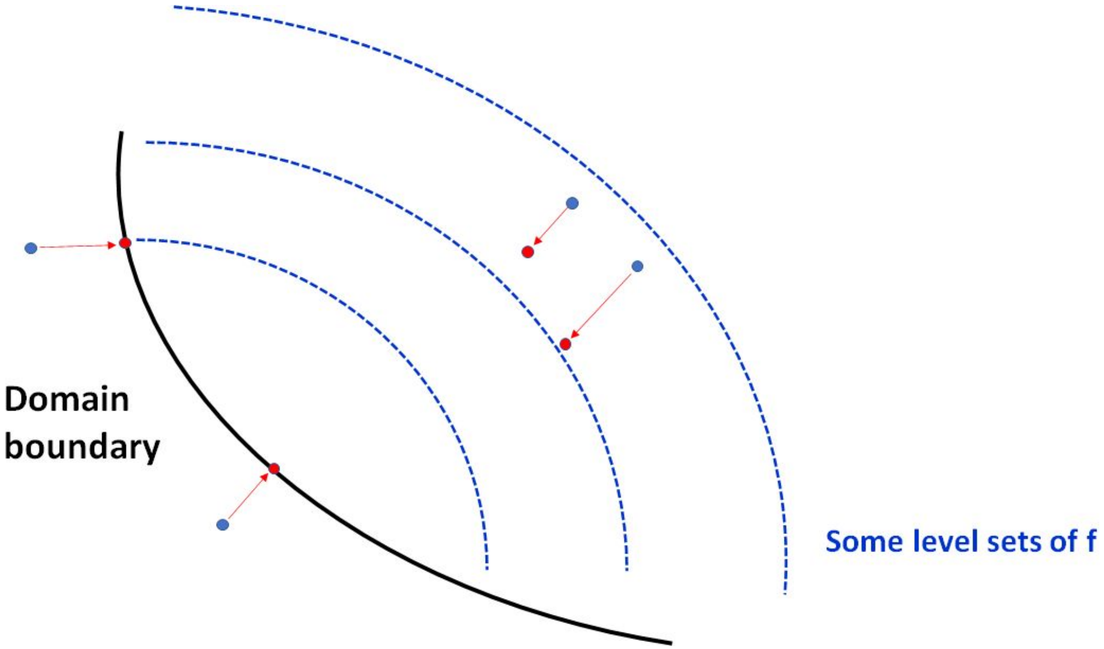
$$x^\star = \mathbf{prox}_{\lambda f}(x^\star),$$



# The proximal point operator

$$\mathbf{prox}_{\lambda f}(v) = \underset{x}{\text{argmin}} \left( f(x) + (1/2\lambda)\|x - v\|_2^2 \right).$$

**First ingredient:**
**The proximal point operator**




$$x^\star = \mathbf{prox}_{\lambda f}(x^\star),$$

$$x^{k+1} = \mathbf{prox}_{\lambda f}(x^k),$$

# The Karush Kuhn Tucker conditions

**Second ingredient:**
**The KKT map**

# The Karush Kuhn Tucker conditions

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

$$\text{s.t.,}\ Ax = b.$$

**Second ingredient:**
**The KKT map**

# The Karush Kuhn Tucker conditions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t.,} \ Ax = b.$$

$$\nabla f(x^\star) + A^\top y^\star = 0,$$
$$b - Ax^\star = 0.$$

**Second ingredient:**
**The KKT map**

# The Karush Kuhn Tucker conditions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t., } Ax = b.$$

$$\nabla f(x^\star) + A^\top y^\star = 0,$$
$$b - Ax^\star = 0.$$

$$G(x, y; H, g, A, b) \stackrel{\text{def}}{=} \begin{bmatrix} Hx + g + A^\top y \\ b - Ax \end{bmatrix}.$$

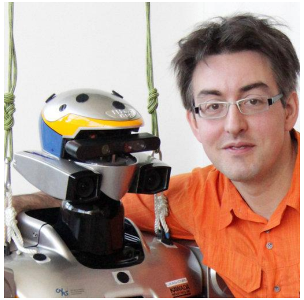
**Second ingredient:**
**The KKT map**

# ProxQP solver

Antoine Bambade[1,2]
[1] *Inria and ENS Paris : Willow and Sierra teams*
[2] *École des Ponts Paris Tech*

Ph.D defense, 15 January 2024

# Associated references

**Conference articles**

- **AB**, S. El-Kazdadi, A. Taylor, J. Carpentier. ProxQP: Yet another Quadratic Programming Solver for Robotics and beyond. In *Robotics: Science and System (RSS)*, 2022;
- W. Jallet, **AB**, N. Mansard, J. Carpentier. Constrained differential dynamic programming: a primal-dual augmented lagrangian approach. In *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022;


**Workshop articles**

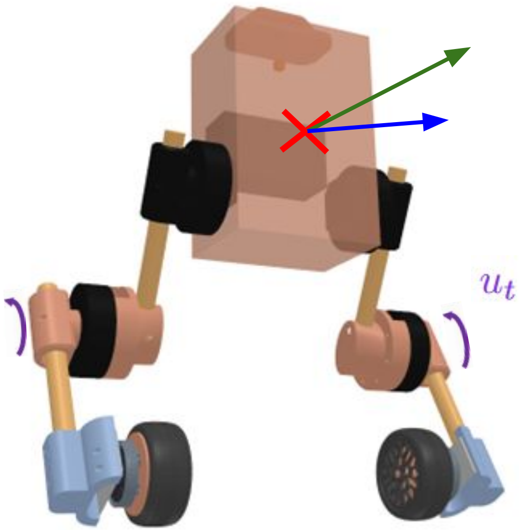
- W. Jallet, **AB**, N. Mansard, J. Carpentier. ProxNLP: a primal-dual augmented Lagrangian solver for nonlinear programming for Robotics and beyond. In *6th Legged Robots Workshop*, 2022;


**Submitted articles**

- **AB**, F. Schramm, S. El-Kazdadi, S. Caron, A. Taylor, J. Carpentier. ProxQP: an Efficient and Versatile Quadratic Programming Solver for Real-Time Robotics Applications and Beyond. Submitted in september 2023 to *IEEE Transactions on Robotics (TRO)*;
- W. Jallet, **AB**, E. Arlaud, S. El-Kazdadi, N. Mansard, J. Carpentier. ProxDDP: Proximal Constrained Trajectory Optimization.Submitted in september 2023 to *IEEE Transactions on Robotics (TRO)*;

# Why using Augmented Lagrangian methods?

$x_t$ = Center of Mass position, velocity, acceleration



$u_t$ = Control input

Control trajectory

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t., } x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

$$-x_{\max} \leq x_t \leq x_{\max},$$

$$-u_{\max} \leq u_t \leq u_{\max}.$$

Safety constraints

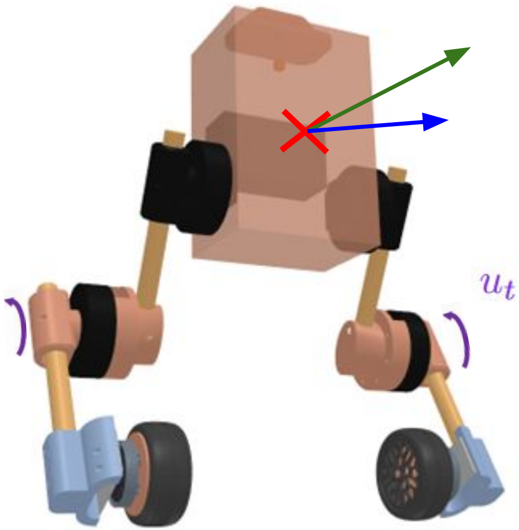
A convex Quadratic Program

Plan

1. Solve efficiently these QPs : ProxQP

# Why using Augmented Lagrangian methods?

**Early stopping**



$u_t = \text{Control input}$

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t., } x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

$$-x_{\max} \leq x_t \leq x_{\max},$$

$$-u_{\max} \leq u_t \leq u_{\max}.$$

**Safety constraints**

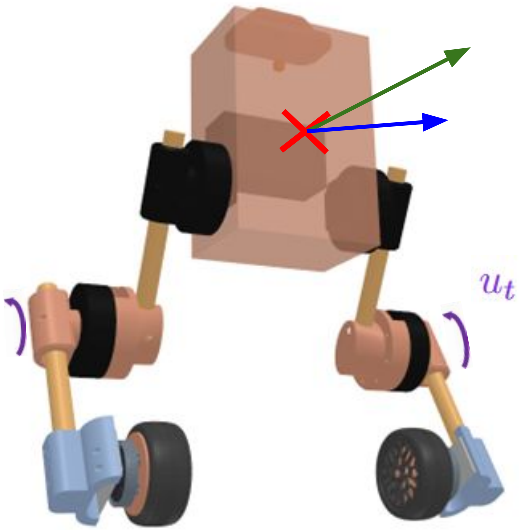
**A convex Quadratic Program**

**Plan**

1. **Solve efficiently these QPs : ProxQP**

# Why using Augmented Lagrangian methods?

**Early stopping** **Warm starting**




**Model Predictive Control (MPC)**

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t.,} \; x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

$$-x_{\max} \leq x_t \leq x_{\max}, \quad -u_{\max} \leq u_t \leq u_{\max}.$$

**Safety constraints**

**A convex Quadratic Program**

**Plan**

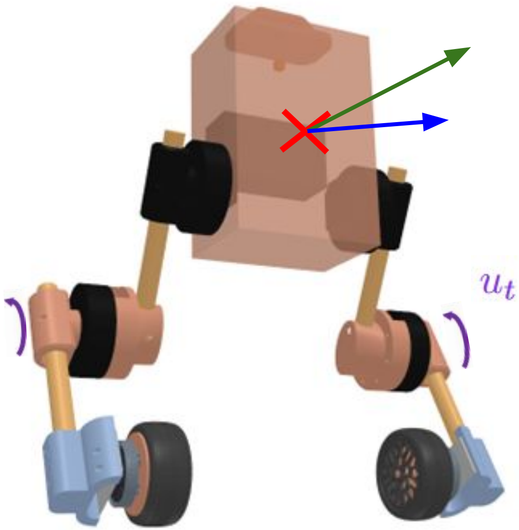
1. **Solve efficiently these QPs : ProxQP**

# Why using Augmented Lagrangian methods?

Early stopping

Warm starting

Sparse and Dense backends



$u_t = \text{Control input}$

Model Predictive Control (MPC)

Control trajectory

$$\min_{x_t,u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t.},\ x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t,$$

$$-x_{\max} \leq x_t \leq x_{\max},$$

$$-u_{\max} \leq u_t \leq u_{\max}.$$

Safety constraints

A convex Quadratic Program

Plan

1. Solve efficiently these QPs : **ProxQP**

# Why using Augmented Lagrangian methods?

| Methods | Early stopping | Warm starting | Sparse and Dense backends |
| --- | --- | --- | --- |
| Active Set | ✗ | ✓ | ✓ |
| Interior Point | ✓ | ✗ | ✓ |
| **Augmented Lagrangian** | ✓ | ✓ | ✓ |

**Model Predictive Control (MPC)**

**Control trajectory**

$$\min_{x_t, u_t} \overbrace{w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,}^{\text{Control trajectory}}$$

$$\text{s.t.,}\ x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

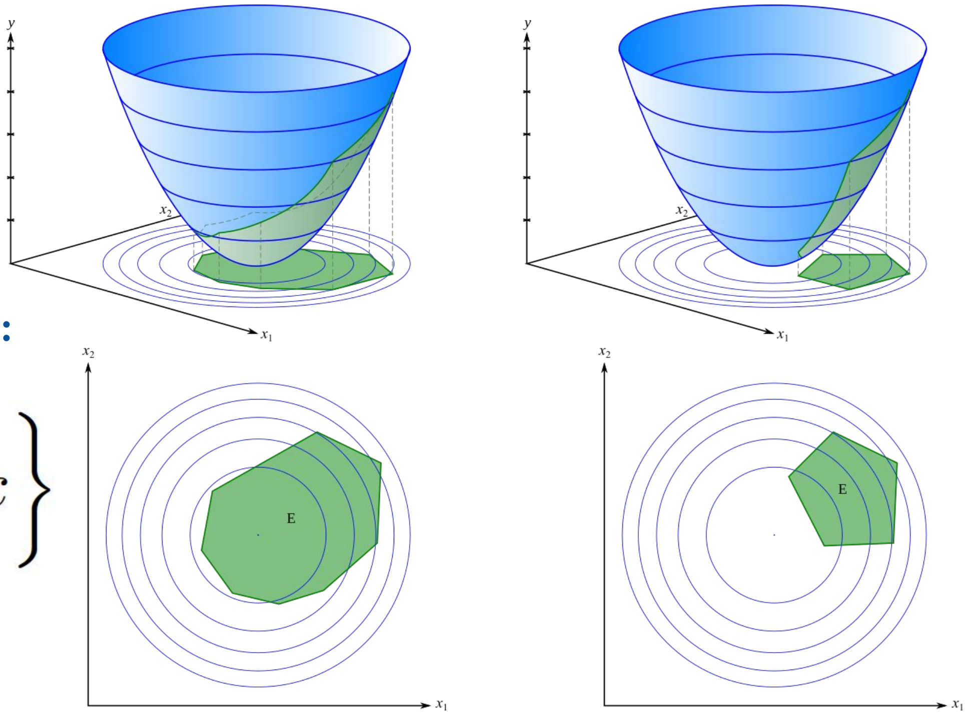
$$\left.\begin{aligned} -x_{\max} \leq x_t \leq x_{\max}, \\ -u_{\max} \leq u_t \leq u_{\max}. \end{aligned}\right\} \text{Safety constraints}$$

**A convex Quadratic Program**

# Why using Augmented Lagrangian methods?

| Methods | Early stopping | Warm starting | Sparse and Dense backends | Solvers |
| --- | --- | --- | --- | --- |
| Active Set | ✗ | ✓ | ✓ | GALAHAD, QUADPROG, DAQP, QPNNLS, QPOASES |
| Interior Point | ✓ | ✗ | ✓ | GUROBI, MOSEK, CVXOPT, ECOS, QPSWIFT, HPIPM, CLARABEL, BPMPD, OOQP |
| Augmented Lagrangian | ✓ | ✓ | ✓ | OSQP, SCS, LANCELOT, QPALM, QPDO |

# Quadratic program definition




**Convex Quadratic Program (QP) standard form:**

$$\min_{x \in \mathbb{R}^n} \left\{ f(x) \stackrel{\text{def}}{=} \frac{1}{2} x^\top H x + g^\top x \right\}$$

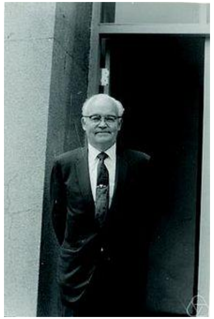
$$\text{s.t., } Cx \leq u,$$

Image source: Wikipedia

# Augmented Lagrangian methods

## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$



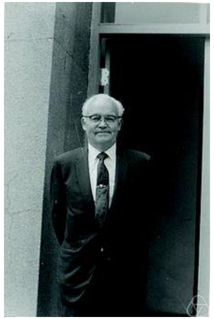
Magnus Hestenes

Michael J.D. Powell

# Augmented Lagrangian methods

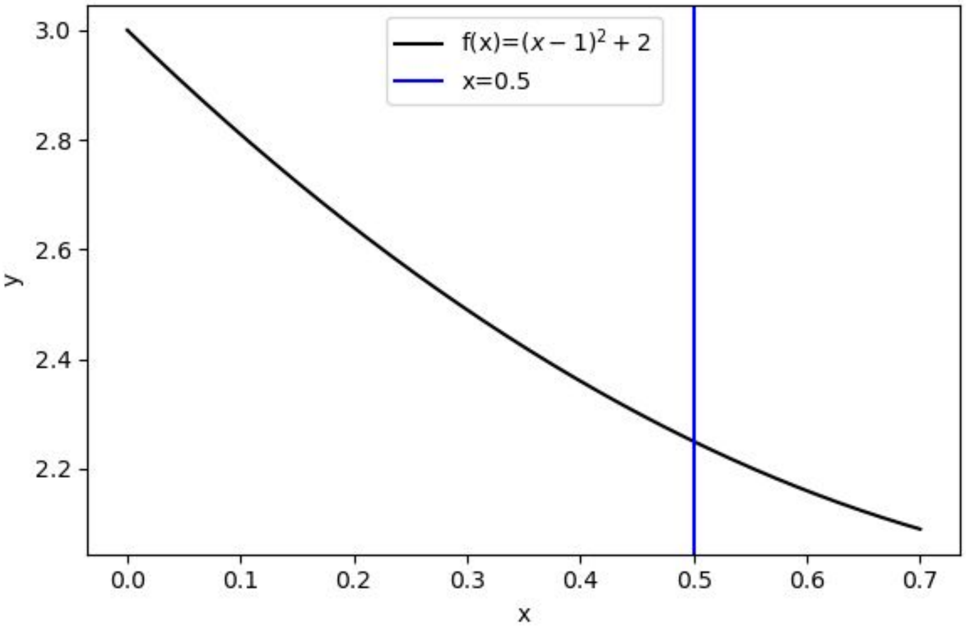
## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$



Magnus Hestenes

Michael J.D. Powell

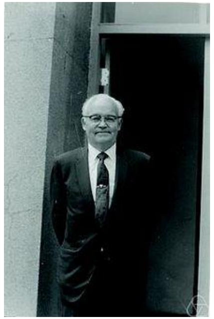



$$\min_{x\in\mathbb{R}} (x-1)^2 + 2,$$
$$\text{s.t.}, \; x \le 0.5,$$

# Augmented Lagrangian methods

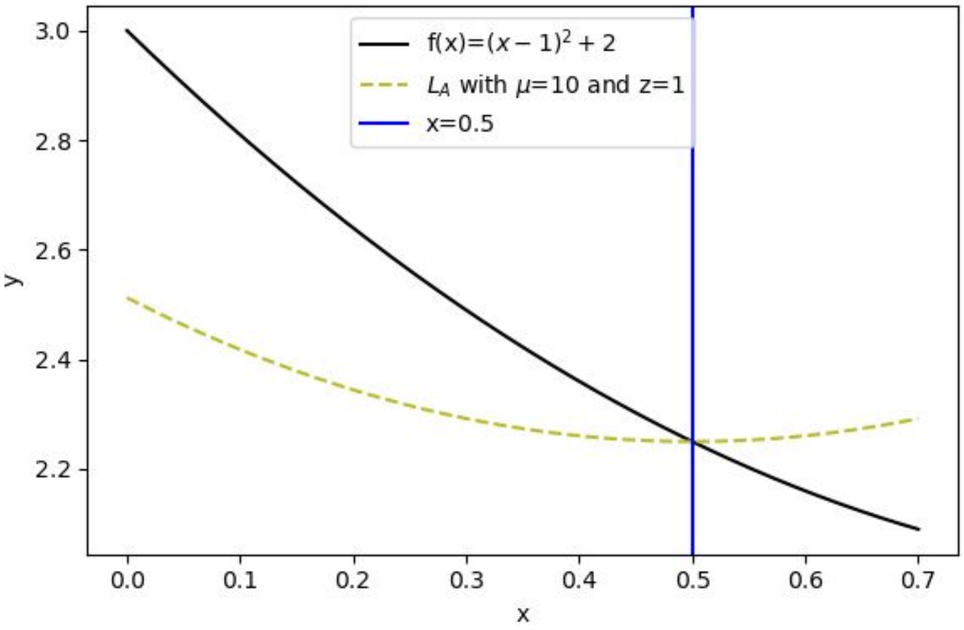
## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$



Magnus Hestenes

Michael J.D. Powell

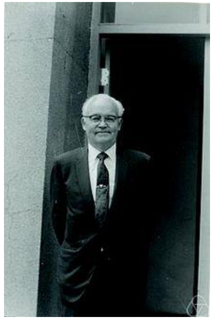



$$\min_{x \in \mathbb{R}} (x-1)^2 + 2,$$
$$\text{s.t., } x \leq 0.5,$$

# Augmented Lagrangian methods

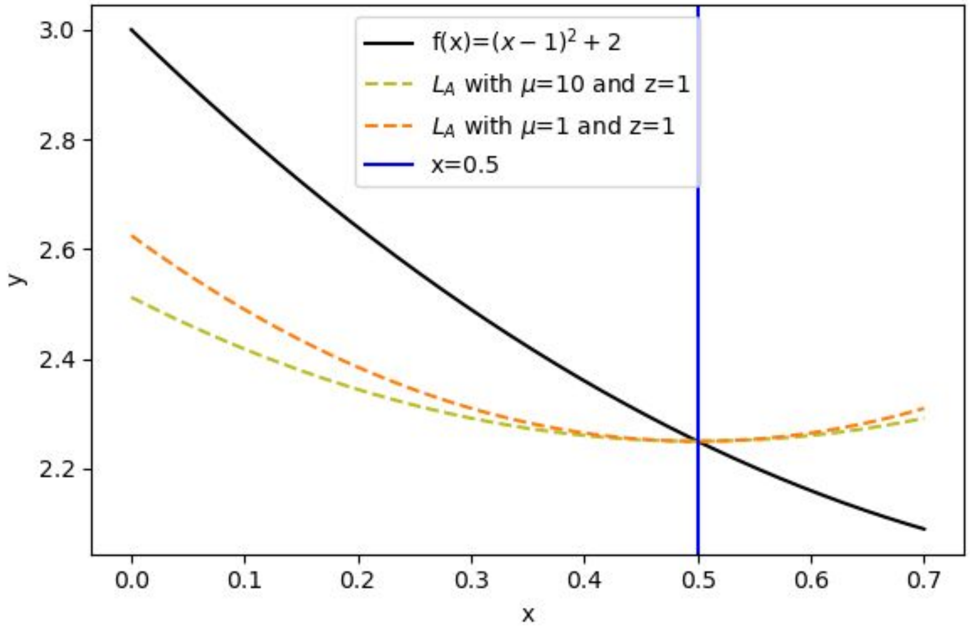
## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$



Magnus Hestenes

Michael J.D. Powell

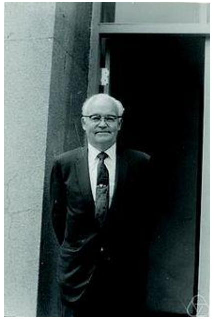



$$\min_{x \in \mathbb{R}} (x-1)^2 + 2,$$
$$\text{s.t.}, \; x \leq 0.5,$$

# Augmented Lagrangian methods

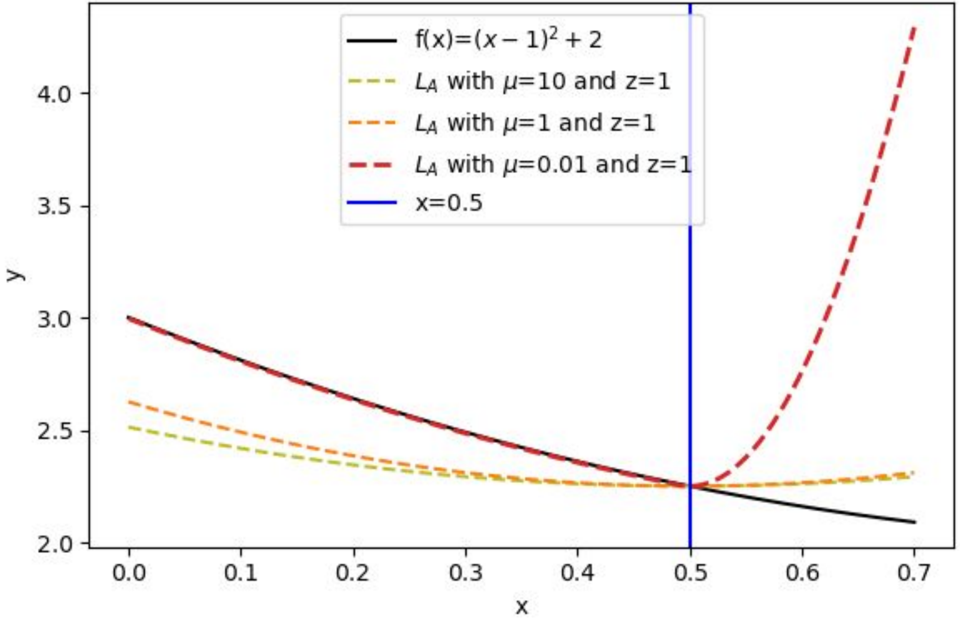
## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$



Magnus Hestenes

Michael J.D. Powell




$$\min_{x \in \mathbb{R}} (x-1)^2 + 2,$$

$$\text{s.t.}, \; x \leq 0.5,$$

# Augmented Lagrangian methods

**Augmented Lagrangian:**

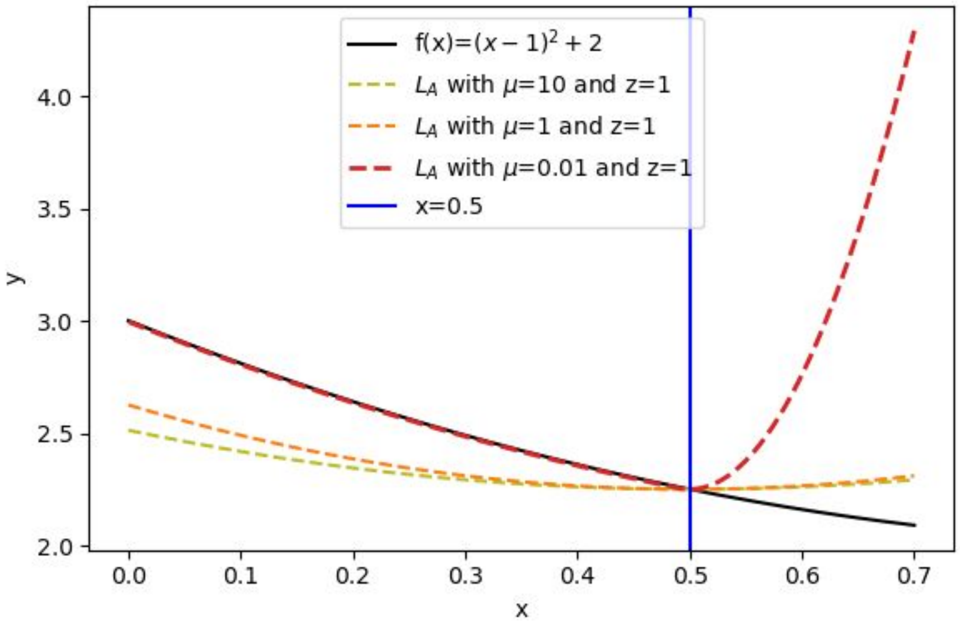
$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$

M. Hestenes (1969) M. J. D. Powell (1969)

**Method of Multipliers:**

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x \in \mathbb{R}^n} \mathcal{L}_A(x, z^k; \mu),$$

$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1} - u)\right]_+,$$




$$\min_{x \in \mathbb{R}} (x-1)^2 + 2,$$
$$\text{s.t., } x \le 0.5,$$

# Augmented Lagrangian methods

## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$

M. Hestenes (1969) M. J. D. Powell (1969)

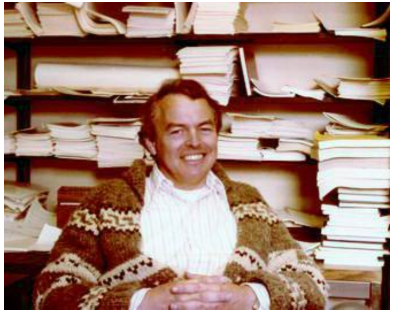
## Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \underset{x \in \mathbb{R}^n}{\arg\min}\ \mathcal{L}_A(x, z^k; \mu),$$

$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1} - u)\right]_+,$$

## Proximal Augmented Lagrangian:

$$\Phi^k_{\rho,\mu}(x) \stackrel{\text{def}}{=} \mathcal{L}_A(x, z^k; \mu) + \underbrace{\frac{\rho}{2}\|x - x^k\|_2^2}_{\text{proximal term}},$$



R. Tyrrell Rockafellar

# Augmented Lagrangian methods

## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$

M. Hestenes (1969) M. J. D. Powell (1969)

## Proximal Augmented Lagrangian:

$$\Phi^k_{\rho,\mu}(x) \stackrel{\text{def}}{=} \mathcal{L}_A(x, z^k; \mu) + \underbrace{\frac{\rho}{2}\|x - x^k\|_2^2}_{\text{proximal term}},$$

R. Tyrrell Rockafellar (1976)

## Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x \in \mathbb{R}^n} \mathcal{L}_A(x, z^k; \mu),$$
$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1} - u)\right]_+,$$

## Proximal Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x \in \mathbb{R}^n} \Phi^k_{\rho,\mu}(x),$$
$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1} - u)\right]_+,$$

# Augmented Lagrangian methods

## Augmented Lagrangian:

$$\mathcal{L}_A(x, z; \mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$

M. Hestenes (1969)

M. J. D. Powell (1969)

## Proximal Augmented Lagrangian:

$$\Phi^k_{\rho,\mu}(x) \stackrel{\text{def}}{=} \mathcal{L}_A(x, z^k; \mu) + \underbrace{\frac{\rho}{2}\|x - x^k\|_2^2}_{\text{proximal term}},$$

R. Tyrrell Rockafellar (1976)

## Primal Dual Proximal Augmented Lagrangian:

$$\mathcal{M}^k_{\rho,\mu^k,\alpha}(x, z),$$

**Contribution:**

Merit function numerically more robust

## Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x\in\mathbb{R}^n} \mathcal{L}_A(x, z^k; \mu),$$

$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1} - u)\right]_+,$$

## Proximal Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x\in\mathbb{R}^n} \Phi^k_{\rho,\mu}(x),$$

$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1} - u)\right]_+,$$

## First step of ProxQP:

$$(x^{k+1}, z^{k+1}) \approx_{\epsilon^k} \arg\min_{x\in\mathbb{R}^n, z\in\mathbb{R}^m} \mathcal{M}^k_{\rho,\mu^k,\alpha}(x, z),$$

# Augmented Lagrangian methods

## Augmented Lagrangian:

$$\mathcal{L}_A(x,z;\mu) \stackrel{\text{def}}{=} f(x) + \underbrace{\frac{1}{2\mu}\left(\|[Cx-u+\mu z]_+\|_2^2 - \|\mu z\|_2^2\right)}_{\text{smoothed shifted penalization}},$$

M. Hestenes (1969)

M. J. D. Powell (1969)

## Proximal Augmented Lagrangian:

$$\Phi^k_{\rho,\mu}(x) \stackrel{\text{def}}{=} \mathcal{L}_A(x,z^k;\mu) + \underbrace{\frac{\rho}{2}\|x-x^k\|_2^2}_{\text{proximal term}},$$

R. Tyrrell Rockafellar (1976)

## Primal Dual Proximal Augmented Lagrangian:

$$\mathcal{M}^k_{\rho,\mu^k,\alpha}(x,z),$$

**Contribution:**
Merit function numerically more robust

## Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x\in\mathbb{R}^n} \mathcal{L}_A(x,z^k;\mu),$$
$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1}-u)\right]_+,$$

## Proximal Method of Multipliers:

$$x^{k+1} \approx_{\epsilon^k} \arg\min_{x\in\mathbb{R}^n} \Phi^k_{\rho,\mu}(x),$$
$$z^{k+1} = \left[z^k + \frac{1}{\mu}(Cx^{k+1}-u)\right]_+,$$

## First step of ProxQP:

$$(x^{k+1},z^{k+1}) \approx_{\epsilon^k} \arg\min_{x\in\mathbb{R}^n, z\in\mathbb{R}^m} \mathcal{M}^k_{\rho,\mu^k,\alpha}(x,z),$$

**Contribution:**
Automatic scheduling of proximal **hyperparameters**

# ProxQP algorithm

**Contribution:**
Automatic scheduling of proximal **hyperparameters**

**while** *Stopping criterion not satisfied* **do**
  $(\hat{x}, \hat{z}) \approx_{\epsilon^k} \arg\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \mathcal{M}^k_{\rho,\mu^k,\alpha}(x, z)$,
  $x^{k+1} = \hat{x}$,
  **if** $\|[Cx^{k+1} - u]_+\|_\infty \leq \epsilon^k_{bcl}$ **then**
    $z^{k+1} = \hat{z}$,
    $\mu^{k+1} = \mu^k$,
    Geometric decrease of $\epsilon^k$, $\epsilon^k_{\text{bcl}}$,
  **else**
    $z^{k+1} = z^k$,
    Geometric decrease of $\mu^k$,
    Slight increase of $\epsilon^k$, $\epsilon^k_{\text{bcl}}$,
  **end**
**end**

**Prox point goes on**

**Restart of prox point**

**Bound Constrained Augmented Lagrangian (BCL)** = globalization strategy

# ProxQP algorithm

**Contribution:**
Automatic scheduling of proximal **hyperparameters**

**while** *Stopping criterion not satisfied* **do**

$(\hat{x}, \hat{z}) \approx_{\epsilon^k} \arg\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \mathcal{M}^k_{\rho, \mu^k, \alpha}(x, z)$,

$x^{k+1} = \hat{x}$,

**if** $\|[Cx^{k+1} - u]_+\|_\infty \leq \epsilon^k_{bcl}$ **then**

$z^{k+1} = \hat{z}$,

$\mu^{k+1} = \mu^k$,

Geometric decrease of $\epsilon^k$, $\epsilon^k_{\text{bcl}}$,

**else**

$z^{k+1} = z^k$,

Geometric decrease of $\mu^k$,

Slight increase of $\epsilon^k$, $\epsilon^k_{\text{bcl}}$,

**end**

**end**

**Prox point goes on** (if branch)

**Restart of prox point** (else branch)

**Bound Constrained Augmented Lagrangian (BCL)** = globalization strategy

**Contribution:**
Proof of global convergence and of **constant step size in finite time**

# ProxQP sub-problem minimization

ProxQP merit function is:

- Strongly convex,
- Piece-wise quadratic,
- Continuously differentiable,
- **Semi-smooth**.

$$\mathcal{M}^k_{\rho,\mu^k,\alpha}(x,z) \stackrel{\text{def}}{=} f(x) + \frac{\rho}{2}\|x - x^k\|_2^2 + \frac{(1-\alpha)\mu^k}{2}\|z\|_2^2 + \frac{1}{2\alpha\mu^k}\|[Cx - u + \mu^k(z^k + (\alpha-1)z)]_+\|_2^2,$$

# ProxQP sub-problem minimization

ProxQP merit function is:

- Strongly convex,
- Piece-wise quadratic,
- Continuously differentiable,
- **Semi-smooth**.

$$\mathcal{M}^k_{\rho,\mu^k,\alpha}(x,z) \stackrel{\text{def}}{=} f(x) + \frac{\rho}{2}\|x - x^k\|_2^2 + \frac{(1-\alpha)\mu^k}{2}\|z\|_2^2 + \frac{1}{2\alpha\mu^k}\|[Cx - u + \mu^k(z^k + (\alpha - 1)z)]_+\|_2^2,$$

Minimization using **semi-smooth Newton** with **exact line-search**.

# ProxQP sub-problem minimization

ProxQP merit function is:

- Strongly convex,
- Piece-wise quadratic,
- Continuously differentiable,
- **Semi-smooth**.

$$\mathcal{M}^k_{\rho,\mu^k,\alpha}(x,z) \stackrel{\text{def}}{=} f(x) + \frac{\rho}{2}\|x - x^k\|_2^2 + \frac{(1-\alpha)\mu^k}{2}\|z\|_2^2 + \frac{1}{2\alpha\mu^k}\|[Cx - u + \mu^k(z^k + (\alpha - 1)z)]_+\|_2^2,$$

Minimization using **semi-smooth Newton** with **exact line-search**.

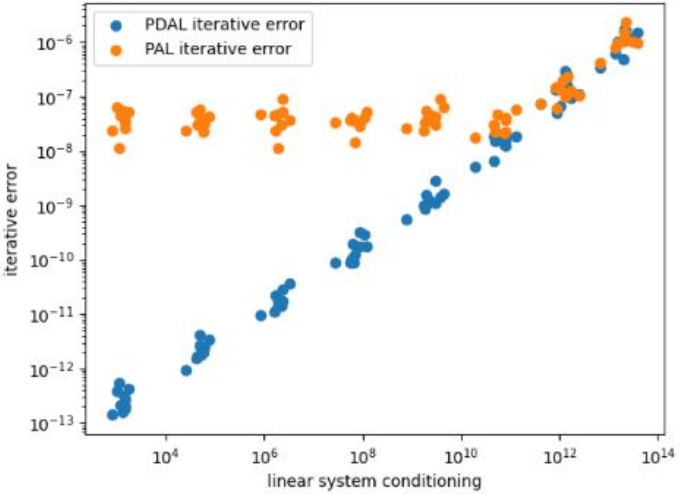



Figure: Linear system error when using $\Phi^k_{\rho,\mu}$ (PAL function) v.s., $\mathcal{M}_{\rho,\mu^k,\alpha}$ (PDAL function) for matrices with increasing ill-conditioning (generated randomly).

# Software contribution

# ProxSuite

THE ADVANCED PROXIMAL OPTIMIZATION TOOLBOX

License BSD 2-Clause | docs online | CI - Linux/OSX/Windows - Cond passing | pypi package 0.6.1 | Anaconda.org 0.6.1

- ✓ **fast**: C++ implementation, with homemade linear Cholesky solver
- ✓ **scalable:** various backends for dense, sparse and matrix-free optimization
- ✓ **easy-to-use:** API closed to OSQP, Python and Julia bindings
- ✓ **open-source:** BSD-license, easily installable

| Conda | Files | Labels | Badges |
|---|---|---|---|

License: BSD-2-Clause

Home: https://github.com/simple-robotics/proxsuite

</> Development: https://github.com/simple-robotics/proxsuite

160860 total downloads

Last upload: 1 month and 25 days ago

Watch 14 | Fork 40 | Starred 303

## Summary

PyPI link

https://pypi.org/project/proxsuite

Total downloads

180,196

Total downloads - 30 days

11,791

Total downloads - 7 days

3,257

# Benchmark setup: solver test set

| Method | Name | Backend | Early stopping | Warm start |
|---|---|---|---|---|
| Active Set | Quadprog | Dense | ✗ | ✓ |
| Active Set | QPOASES | Dense | ✗ | ✓ |
| Interior Point | MOSEK | Sparse | ✓ | ✗ |
| Interior Point | QPSWIFT | Sparse | ✓ | ✗ |
| ADMM | OSQP | Sparse | ✓ | ✓ |
| ADMM | SCS | Sparse | ✓ | ✓ |

# Benchmark setup: problems involved

| Sparsity | Problem type | Reference | Controlled object |
|---|---|---|---|
| Dense | Inverse kinematics | tasts-robots, 2023 | UR3, UR5, Stretch, Dual ARM, KinovaGen2, Sigmaban |
| Dense | SQP | Ferreau et al., 2014 | Chain of masses |
| Dense | MPC | Wieber, 2006a | Humanoid robot |
| Dense | MPC | tasts-robots, 2023 | Upkie robot |
| Sparse | MPC | Wang & Boyd, 2009 | Chain of masses |
| Sparse | MPC | Stellato et al., 2020 | Synthetic |

# Dense QP benchmark

| Task name | ProxQP | QUADPROG | OSQP | QPOASES | SCS | QPSWIFT | MOSEK |
|---|---|---|---|---|---|---|---|
| UR3 IK | **13.2±0.1μs** | 17.0±3.3μs | 15.8±0.2μs | 21.7±0.3μs | ✗ [1] | 52.2±1.2μs | 323.1±74μs [2] |
| UR5 IK | **11.9±0.1μs** | 16.8±0.2μs | 16.5±0.2μs | 21.4±0.2μs | ✗ [1] | 53.8±0.8μs | 310.8±5.3μs [2] |
| DUAL ARMS IK | **17.2±0.1μs** | 23.3±0.2μs | 40.4±0.6μs | 330.4±5.8μs | 81.5±0.6μs | 152.1±1.1μs | 554.4±25.1μs [2] |
| KINOVAGEN2 IK | **15.8±0.1μs** | 18.9±0.2μs | 17.0±0.2μs | 31.4±0.4μs | 46.5±1.3μs | 53.7±0.4μs | 375.4±14.9μs [2] |
| SIGMABAN IK | **14.1±0.2μs** | 25.2±0.4μs | 45.2±0.9μs | 523.6±4.8μs | 68.6±0.5μs | 224.9±2.0μs | 452.5±8.8μs [2] |
| STRETCH IK | **26.9±0.3μs** | 36.7±0.6μs | 53.1±0.9μs | 212.6±0.8μs | 455.3±23.8μs [2] | 152.8±1.5μs | ✗ [3] |
| CHAIN80 SQP | **15.6±0.3ms** | 355.8±0.9ms | 456.5±2.6ms | 182±2.9ms | 837±16.0ms | 2193.9±22.3ms | 1554.6±19.5ms [2] |
| CHAIN80w SQP | **265.7±2.9ms** | 610.1±5.7ms | 2141.9±22.4ms | 467.4±3.3ms [1] | ✗ [4] | ✗ [4] | 3444.2±30.3ms [2] |
| HUMANOID MPC $\epsilon_{ABS} = 10^{-2}$ | **1.6±0.01ms** | 4.5±1.8ms | 1.8±0.01ms | 18.0±5.3ms | 433.0±21.7ms [5] | ✗ [3] | 70.7±0.7ms |
| HUMANOID MPC $\epsilon_{ABS} = 10^{-4}$ | **2.6±0.02ms** | 4.5±1.8ms | 2.7±0.03ms | 18.0±5.3ms | 80.2±2.9ms [5] | ✗ [3] | 68.3±0.3ms |
| HUMANOID MPC $\epsilon_{ABS} = 10^{-6}$ | **4.4±0.05ms** | 4.5±1.8ms | 4.4±0.05ms | 18.0±5.3ms | 64.3±2.4ms [5] | ✗ [3] | 69.7±0.8ms |

[1] The solver throws a factorization error (of non-convexity). [2] The solution does not match the desired accuracy
[3] The solver does not manage to satisfy configuration limits. [4] The solver does not manage to handle upper bound constraints and outputs infeasibility errors.
[5] Low accuracy iterates provokes with SCS warm starts more difficult QPs to solve in a closed-loop strategy.

**Table**: Dense QP benchmarks (average runtime per time-step (IK) and total simulation runtimes).

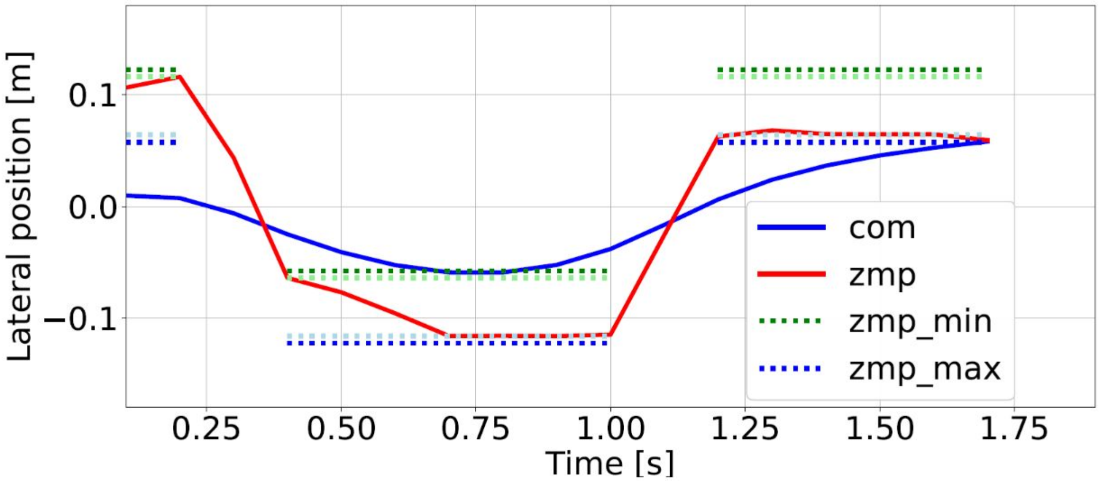
# Sparse QP benchmark

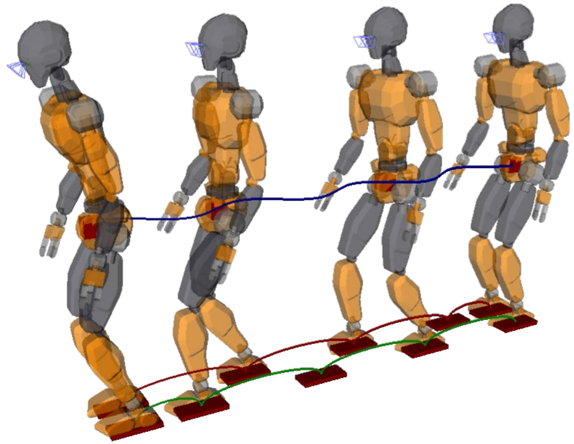
| Problem type | Dimensions | PROXQP | QUADPROG | OSQP | QPOASES | SCS | QPSWIFT | MOSEK |
|---|---|---|---|---|---|---|---|---|
| CONVEX MPC $\epsilon_{abs} = 10^{-3}$ | n=76,m=142 | **5.7±0.2** | ✗[2] | 6.7±0.7 | 37.6±23.9 | 11.9±2.2 | 13.5±0.5 | 82.1±3.5[1] |
| | n=162,m=294 | **16.4±1.7** | ✗[2] | 24.1±6.0 | 393.9±109.9 | 51.0±15.7 | 47.1±4.6 | 211.5±13.3 |
| | n=216,m=392 | **27.6±4.8** | ✗[2] | 48.2±29.4 | 1434.2±480 | 65.6±30.9 | 80.0±3.6 | 311.3±15.4[1] |
| | n=270,m=490 | **43.2±8.3** | ✗[2] | 77.5±65.6 | 2345.5±608 | 104.7±66.0 | 131.0±7.4 | 454.2±18.6 |
| CONVEX MPC $\epsilon_{abs} = 10^{-6}$ | n=76,m=142 | **8.5±1.4** | ✗[2] | 14.1±8.4 | 27.1±15.4 | 25.5±14.6 | 16.5±0.7 | 84.0±3.8[1] |
| | n=162,m=294 | **21.1±2.3** | ✗[2] | 94.0±101.9 | 452.9±98.3 | 101.8±47.9 | 57.8±5.3 | 216.3±14.1[1] |
| | n=216,m=392 | **36.1±4.3** | ✗[2] | 150.1±136.0 | 1626.6±650.1 | 162.0±95.0 | 99.5±4.9 | 314.1±17.4[1] |
| | n=270,m=490 | **57.0±9.0** | ✗[2] | 346.7±362.0 | 2670.3±842.4 | 311.1±122.5 | 164.1±9.9 | 459.6±21.0[1] |
| CHAIN OF MASS $\epsilon_{abs} = 10^{-3}$ | n=462,m=834 | **45.2±0.8** | (28.3±0.6)E3 | 59.1±1.7 | (80.2±1.6)E3 | 161.1±11.1 | 215.3±1.3 | 566.8±14.1 |
| CHAIN OF MASS $\epsilon_{abs} = 10^{-6}$ | n=462,m=834 | **67.8±11.1** | (28.3±0.6)E3 | 104.9±9.6 | (80.2±1.6)E3 | 172.6±9.8 | 248.9±3.1 | 575.5±16.8[1] |

[1] The solution does not satisfy the desired absolute accuracy. [2] QUADPROG cannot solve QPs that are not strictly convex.

**Table**: Sparse convex MPC benchmark (total runtimes in ms for solving 100 simulation steps).

# Robustness to perturbations



**Experiment**: Controlling the lateral center-of-mass trajectory (blue) to maintain the ZMP (red) within the real support polygon (dotted dark green and blue). Light dotted lines are more conservative ZMP bounds.

Image source: S. Caron’s blog

# Robustness to perturbations

| Noise Level | ProxQP | quadprog | OSQP | qpOASES | SCS | qpSWIFT | MOSEK |
|---|---|---|---|---|---|---|---|
| 10.0 | **11.9±9.7%** | 1.0±0.2% | 1.9±0.9% | 1.0±0.2% | 1.0±0.2% | 0±0.0% | 1.0±0.2% |
| 5.0 | **58.38±36.4%** | 1.1±0.3% | 2.1±1.1% | 1.1±0.3% | 1.1±0.4% | 0±0.0% | 1.1±0.3% |
| 1.0 | **100±0.0%** | 1.4±0.8% | 3.5±2.4% | 1.4±0.8% | 1.5±1.1% | 0±0.0% | 1.4±0.9% |
| 0.5 | **100±0.0%** | 1.8±1.2% | 5.5±3.8% | 1.9±1.5% | 2.1±1.6% | 0±0.0% | 1.8±1.2% |
| 0.1 | **100±0.0%** | 3.3±2.6% | 51.6±36.7% | 4.3±3.8% | 4.9±4.3% | 0±0.0% | 3.3±2.6% |
| 0.05 | **100±0.0%** | 3.5±3.2% | 97.6±13.5% | 5.0±6.9% | 7.7±6.5% | 0±0.0% | 3.5±3.2 |
| 0.01 | **100±0.0%** | 4.4±4.4% | **100±0.0%** | 7.7±9.8% | 60.2±37.8% | 0±0.0% | 4.4±4.5% |
| $10^{-3}$ | **100±0.0%** | 5.0±5.2% | **100±0.0%** | 11.4±12.5% | **100±0.0%** | 0±0.0% | 5.0±5.2% |
| $10^{-4}$ | **100±0.0%** | 5.0±5.2% | **100±0.0%** | 15.5±16.8% | **100±0.0%** | 0±0.0% | 5.0±5.2% |
| $10^{-5}$ | **100±0.0%** | 5.0±5.2% | **100±0.0%** | 83.0±36.5% | 99.1±8.9% | 0±0.0% | 5.1±5.3% |
| $10^{-7}$ | **100±0.0%** | 5.0±5.2% | **100±0.0%** | **100±0.0%** | 97±14.8% | 0±0.0% | 44.8±34.2% |
| $10^{-9}$ | **100±0.0%** | 5.0±5.2% | **100±0.0%** | **100±0.0%** | **100±0.0%** | 0±0.0% | **100±0.0%** |
| 0.0 | **100±0.0%** | **100±0.0%** | **100±0.0%** | **100±0.0%** | **100±0.0%** | 0±0.0% | **100±0.0%** |

**Table**: Humanoid locomotion MPC problems with perturbations (percentage of problems solved).

# Test on a real hardware

**Model Predictive Control (MPC)**

**Control trajectory**

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,$$

$$\text{s.t., } x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

**Dynamic model**

$$-x_{\max} \leq x_t \leq x_{\max},$$
$$-u_{\max} \leq u_t \leq u_{\max}.$$

**Safety constraints**

# QPLayer

Antoine Bambade[1,2]

[1] *Inria and ENS Paris : Willow and Sierra teams*

[2] *École des Ponts Paris Tech*

Ph.D defense, 15 January 2024

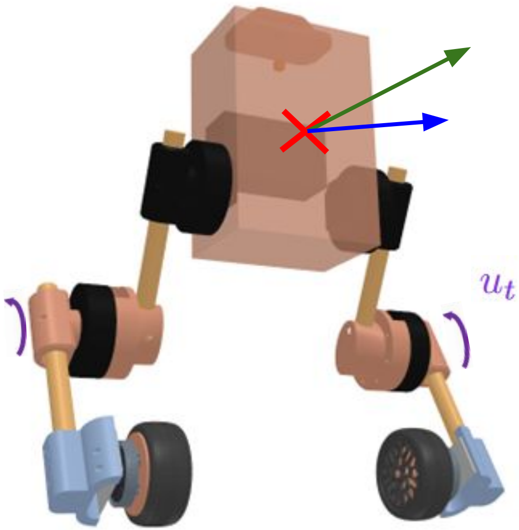
ENS | PSL

École des Ponts ParisTech

Inria INVENTEURS DU MONDE NUMÉRIQUE

# Objective

$x_t =$ Center of Mass position, velocity, acceleration



$u_t =$ Control input

Control trajectory

$$\min_{x_t, u_t} w_T \|x_T - x_{\text{goal}}\|_2^2 + \overbrace{\sum_{t=0}^{T-1} w_x \|x_t - x_{\text{goal}}\|_2^2 + w_u \|u_t\|_2^2,}$$

$$\text{s.t., } x_{t+1} = \mathbf{A} x_t + \mathbf{B} u_t,$$

$$-x_{\max} \leq x_t \leq x_{\max},$$

$$-u_{\max} \leq u_t \leq u_{\max}.$$

Safety constraints

A convex Quadratic Program

**Plan**

2. Differentiate through solutions : QPLayer

# Associated references

**Conference articles**

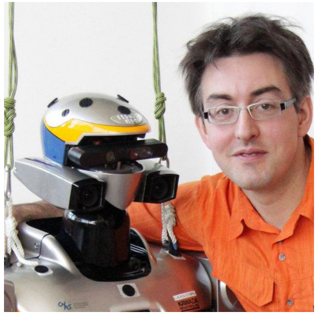
- L. Montaud, Q. Le Lidec, **AB**, V. Petrik, J. Sivic, J. Carpentier. Differentiable collision detection: a randomized smoothing approach. In I*EEE: International Conference on Robotics and Automation (ICRA)*, 2023;

**Submitted articles**

- **AB**, F. Schramm, A. Taylor, J. Carpentier. Leveraging augmented Lagrangian techniques for differentiating over infeasible quadratic programs in machine learning. In *International Conference on Learning Representations (ICLR)*, 2024;
- W. Jallet, **AB**, F. Schramm, Q. Le Lidec, N. Mansard, J. Carpentier. Notes on Importance Sampling of the first order estimator. Communication iterm submitted in september 2023 to *IEEE Transactions on Robotics (TRO)*;






# Standard neural network pipeline

Outputs of current learning pipelines are explicit function of the inputs.

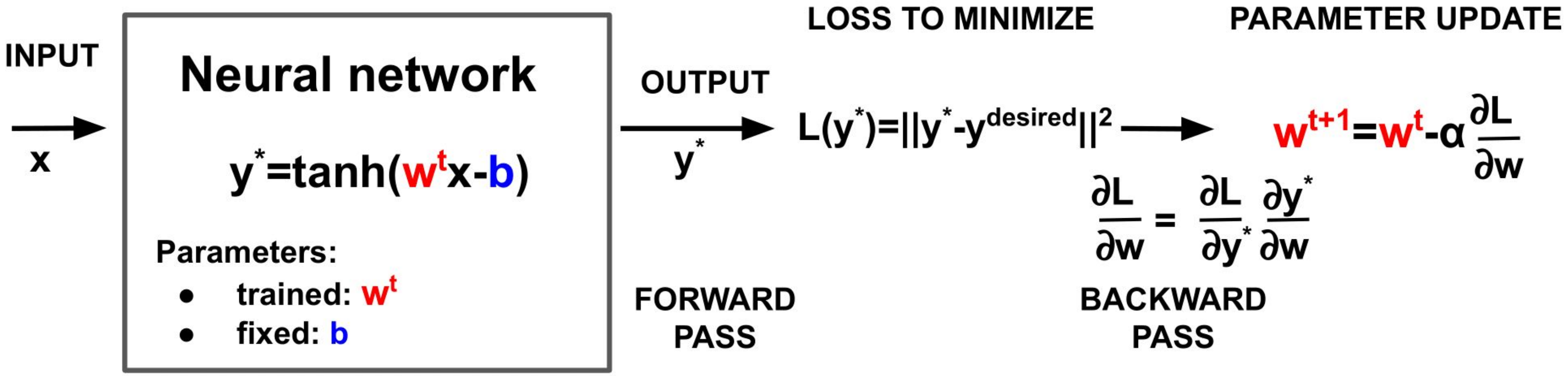



**Figure**: Example of a feedforward neural network.

# Quadratic programming layer pipeline

More recent literature considers differentiable optimization problems as layers.

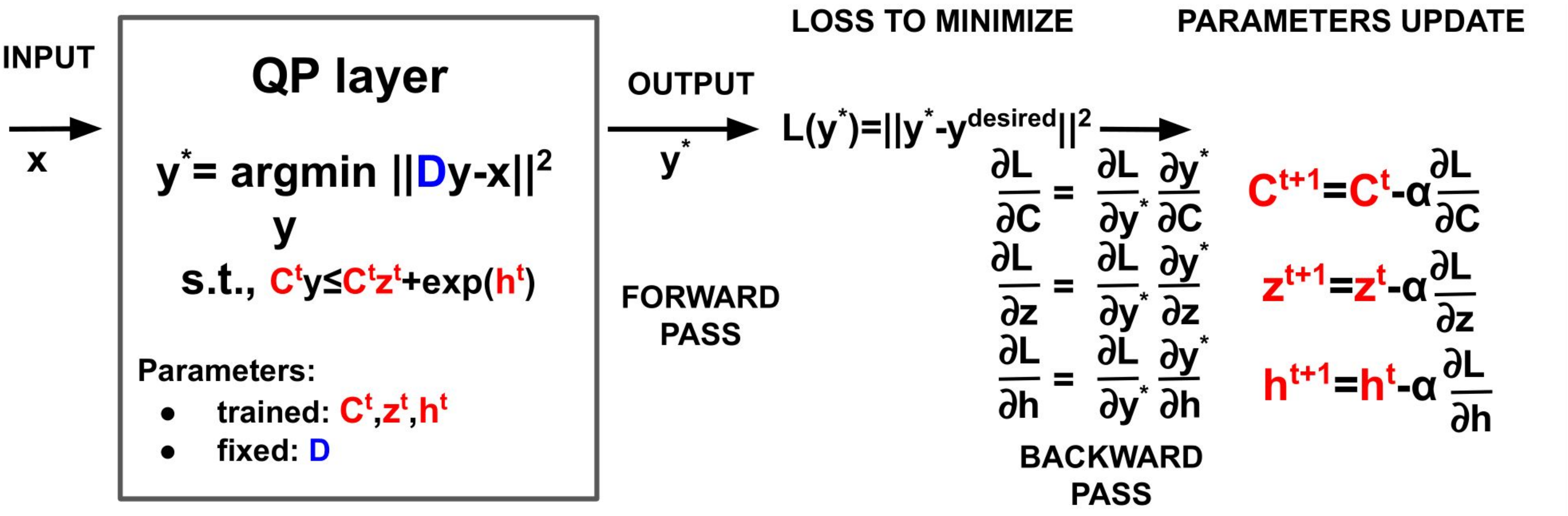



**Figure**: Example of a Quadratic Programming Layer (with D nonsingular)

# QP layers in machine learning

Convex QP layers performs better than a ConvNet for solving Sudokus.

| | | | |
|---|---|---|---|
| | | | 3 |
| 1 | | | |
| | | 4 | |
| 4 | | | 1 |

| | | | |
|---|---|---|---|
| 2 | 4 | 1 | 3 |
| 1 | 3 | 2 | 4 |
| 3 | 1 | 4 | 2 |
| 4 | 2 | 3 | 1 |

**Figure**: Example of Sudoku.

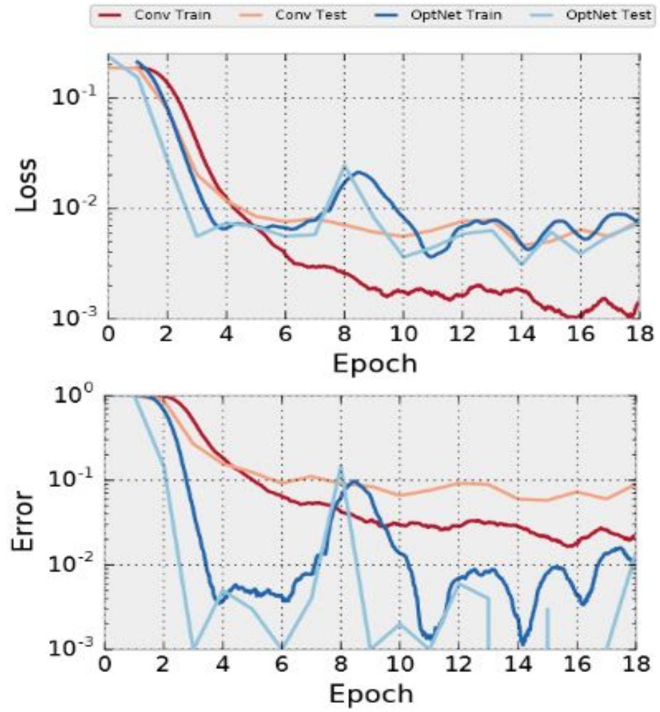



**Figure**: Training and test plots[1].

[1]B. Amos, Z. Kolter (2021)

# QP layers cons: limited trainable architecture

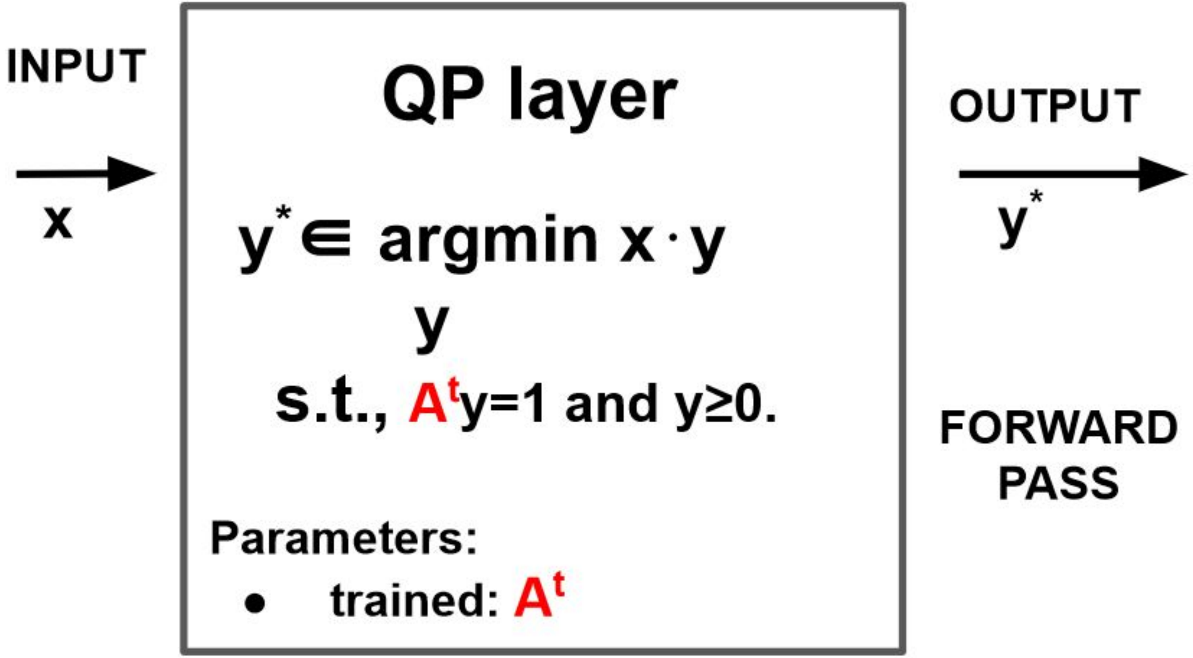
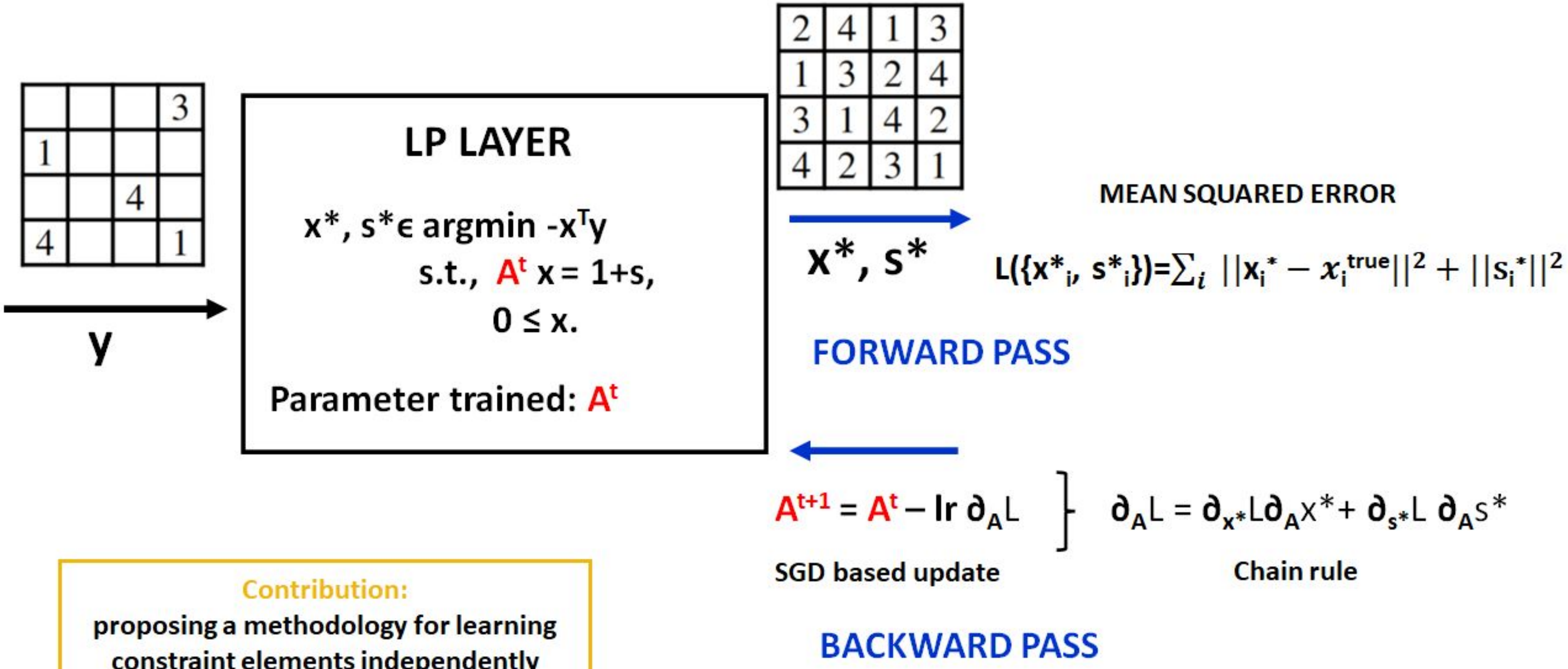


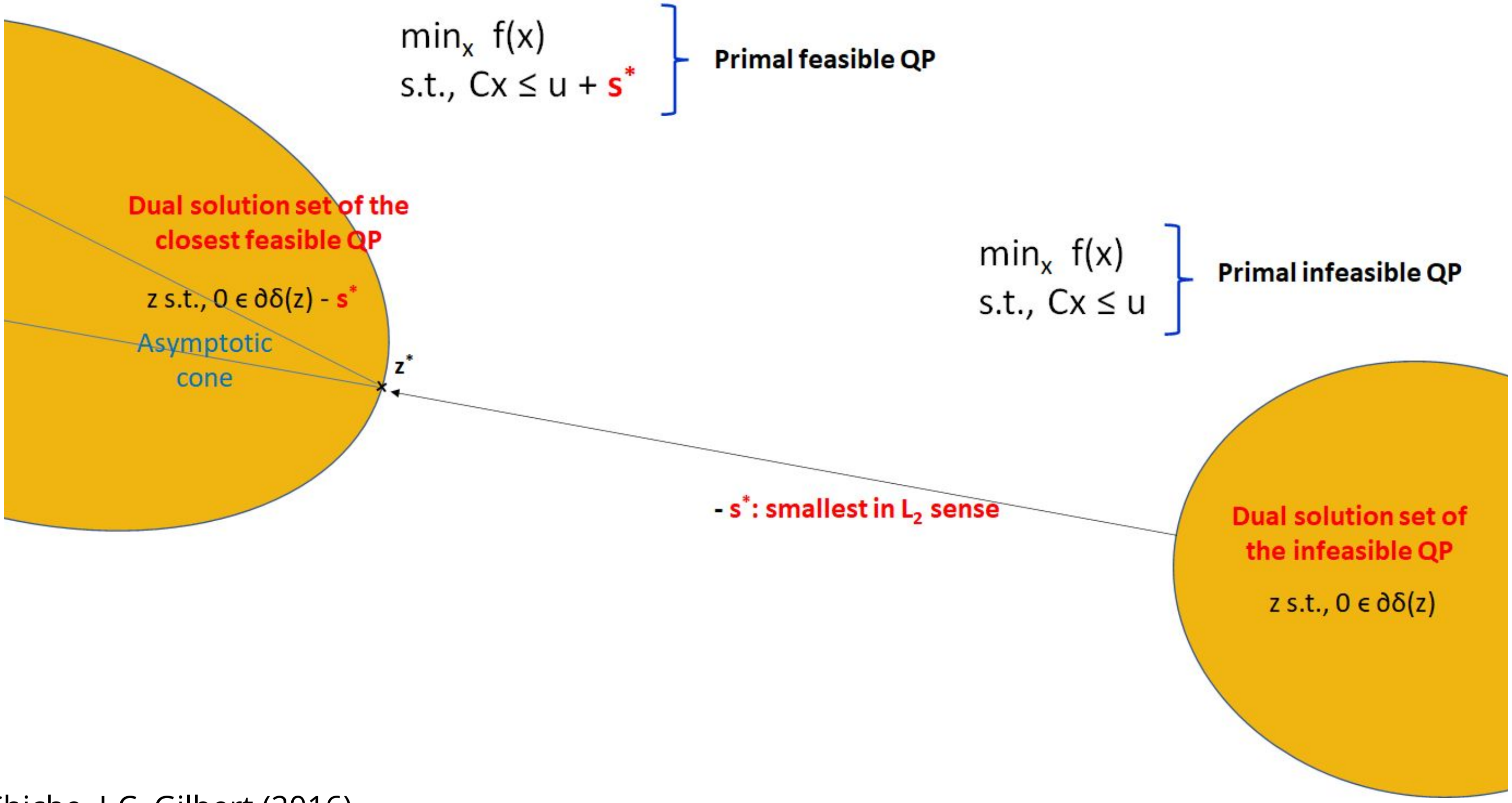

**Figure**: a LP layer. Nothing guarantees during training that the vector of 1 lies in the range space of $A^t$.

# Solution outline: ideal pipeline



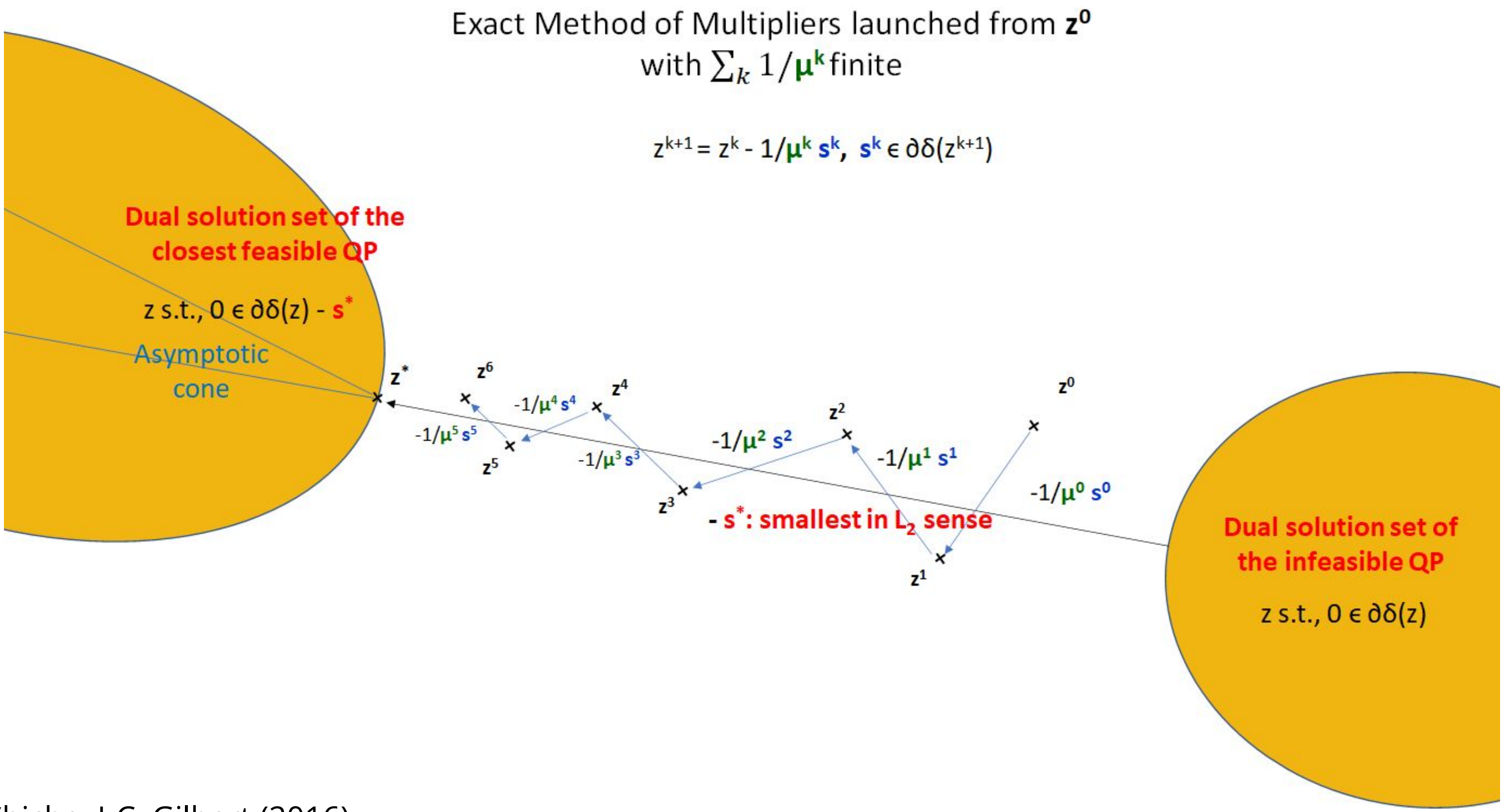

**Contribution:**
proposing a methodology for learning constraint elements independently

# The closest feasible QP problem: definition



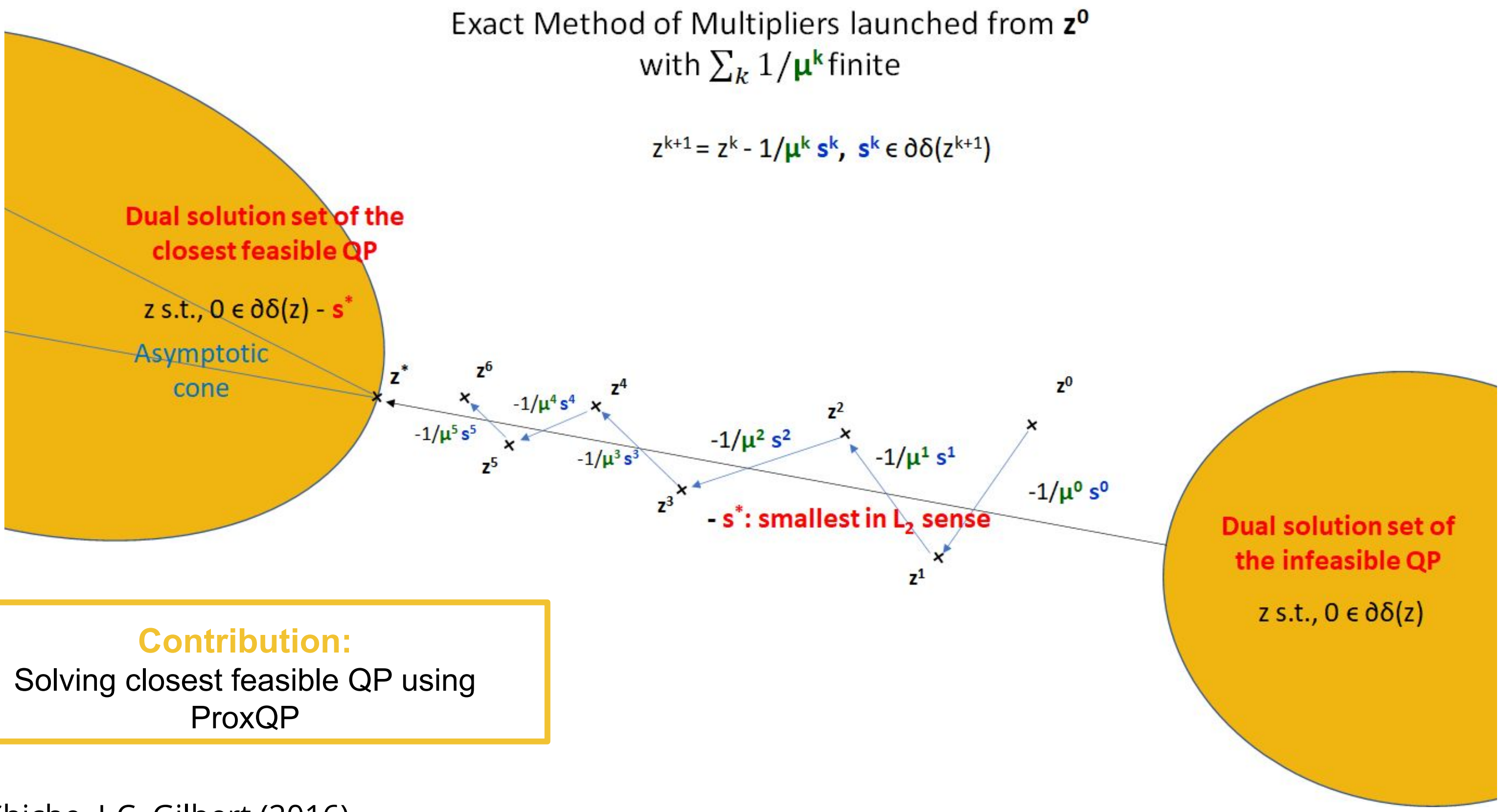

A. Chiche, J-C. Gilbert (2016)

# The closest feasible QP problem: solution method




A. Chiche, J-C. Gilbert (2016)

# The closest feasible QP problem: solution method




**Contribution:**
Solving closest feasible QP using ProxQP

A. Chiche, J-C. Gilbert (2016)

# The backward pass: differentiating closest QP solutions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t., } Ax = b.$$

$$G(x, y; H, g, A, b) \stackrel{\text{def}}{=} \begin{bmatrix} Hx + g + A^\top y \\ b - Ax \end{bmatrix}.$$

# The backward pass: differentiating closest QP solutions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t., } Ax = b.$$

$$G(x, y; H, g, A, b) \stackrel{\text{def}}{=} \begin{bmatrix} Hx + g + A^\top y \\ b - Ax \end{bmatrix}.$$

A classical technique: **the Implicit Function Theorem.**

Let $v^\star = (x^\star, z^\star)^\top$ s.t. $G(x^\star, z^\star;\ \theta) = 0$.
$\theta =$ elements to learn in $\{H, g, A, b\}$.

# The backward pass: differentiating closest QP solutions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t., } Ax = b.$$

$$G(x, y; H, g, A, b) \stackrel{\text{def}}{=} \begin{bmatrix} Hx + g + A^\top y \\ b - Ax \end{bmatrix}.$$

A classical technique: **the Implicit Function Theorem.**

Let $v^\star = (x^\star, z^\star)^\top$ s.t. $G(x^\star, z^\star;\ \theta) = 0$.
$\theta =$ elements to learn in $\{H, g, A, b\}$.

$$\frac{\partial G(x^\star, z^\star;\ \theta)}{\partial v^\star} \begin{bmatrix} \frac{\partial x^\star}{\partial \theta} \\ \frac{\partial z^\star}{\partial \theta} \end{bmatrix} + \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial \theta} = 0.$$

# The backward pass: differentiating closest QP solutions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t.,} \; Ax = b.$$

$$G(x, y; H, g, A, b) \stackrel{\text{def}}{=} \begin{bmatrix} Hx + g + A^\top y \\ b - Ax \end{bmatrix}.$$

A classical technique: **the Implicit Function Theorem.**

Let $v^\star = (x^\star, z^\star)^\top$ s.t. $G(x^\star, z^\star;\ \theta) = 0.$
$\theta =$ elements to learn in $\{H, g, A, b\}.$

$$\left(\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\partial \theta}\right) \in \arg\min_w \left\| \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial v^\star} w + \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial \theta} \right\|_2^2,$$

A. Agrawal et al. (2019); M. Blondel (2021).

# The backward pass: differentiating closest QP solutions

$$\min_{x \in \mathbb{R}} f(x)$$
$$\text{s.t., } Ax = b.$$

$$G(x, y; H, g, A, b) \stackrel{\text{def}}{=} \begin{bmatrix} Hx + g + A^\top y \\ b - Ax \end{bmatrix}.$$

A classical technique: **the Implicit Function Theorem.**

Let $v^\star = (x^\star, z^\star)^\top$ s.t. $G(x^\star, z^\star;\ \theta) = 0.$
$\theta =$ elements to learn in $\{H, g, A, b\}.$

$$\left(\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\partial \theta}\right) \in \arg\min_{w} \left\| \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial v^\star} w + \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial \theta} \right\|_2^2,$$

**Contribution:**
Extend the technique for the closest feasible QP solutions.

A. Agrawal et al. (2019); M. Blondel (2021).

# The backward pass: differentiating closest QP solutions

$$\begin{aligned} s^\star(\theta) = &\arg\min_{s\in\mathbb{R}^{n_i}} \tfrac{1}{2}\|s\|_2^2 \\ &\text{s.t. } x^\star(\theta), z^\star(\theta) \in \arg\min_{x\in\mathbb{R}^n} \max_{z\in\mathbb{R}_+^{n_i}} L(x,z,s;\ \theta), \end{aligned}$$

with $L(x,z,s;\ \theta) \triangleq f(x;\ \theta) + z^\top(C(\theta)x - u(\theta) - s)$.

A classical technique: **the Implicit Function Theorem.**

$$G(x,z,t;\ \theta) \triangleq \begin{bmatrix} \nabla_x f(x;\ \theta) + C(\theta)^\top z \\ C(\theta)x - u(\theta) - t \\ [[t]_- + z]_+ - z \\ C(\theta)^\top [t]_+ \end{bmatrix}$$

**Contribution:**
Extend the technique for the closest feasible QP solutions.

# The backward pass: differentiating closest QP solutions

$$s^\star(\theta) = \arg\min_{s \in \mathbb{R}^{n_i}} \frac{1}{2}\|s\|_2^2$$
$$\text{s.t. } x^\star(\theta), z^\star(\theta) \in \arg\min_{x \in \mathbb{R}^n} \max_{z \in \mathbb{R}_+^{n_i}} L(x, z, s;\ \theta),$$

$$\text{with } L(x, z, s;\ \theta) \triangleq f(x;\ \theta) + z^\top (C(\theta)x - u(\theta) - s).$$

A classical technique: **the Implicit Function Theorem.**

$$G(x, z, t;\ \theta) \triangleq \begin{bmatrix} \nabla_x f(x;\ \theta) + C(\theta)^\top z \\ C(\theta)x - u(\theta) - t \\ [[t]_- + z]_+ - z \\ C(\theta)^\top [t]_+ \end{bmatrix}$$

**Contribution:**
Extend the technique for the closest feasible QP solutions.

The map is **path-differentiable**.

E. Pauwels et al. (2019)

# The backward pass: differentiating closest QP solutions

$$
\begin{aligned}
s^\star(\theta) = &\arg \min_{s \in \mathbb{R}^{n_i}} \tfrac{1}{2}\|s\|_2^2 \\
&\text{s.t. } x^\star(\theta), z^\star(\theta) \in \arg \min_{x \in \mathbb{R}^n} \max_{z \in \mathbb{R}_+^{n_i}} L(x, z, s;\ \theta),
\end{aligned}
$$

$$
\text{with } L(x, z, s;\ \theta) \triangleq f(x;\ \theta) + z^\top (C(\theta)x - u(\theta) - s).
$$

A classical technique: **the Implicit Function Theorem.**

**Contribution:**
Extend the technique for the closest feasible QP solutions.

$$
G(x, z, t;\ \theta) \triangleq \begin{bmatrix} \nabla_x f(x;\ \theta) + C(\theta)^\top z \\ C(\theta)x - u(\theta) - t \\ [[t]_- + z]_+ - z \\ C(\theta)^\top [t]_+ \end{bmatrix}
$$

The map is **path-differentiable**.

$$
\left(\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\partial \theta}\right) \in \arg\min_w \left\| \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial v^\star} w + \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial \theta} \right\|_2^2,
$$

$$
\Pi \frac{\partial t^\star}{\partial \theta} \in \frac{\partial s^\star}{\partial \theta}, \text{ with } \Pi \in \partial([.]_+)(t^\star).
$$

# The backward pass: differentiating closest QP solutions

$$
\begin{aligned}
s^\star(\theta) = & \arg \min_{s \in \mathbb{R}^{n_i}} \tfrac{1}{2}\|s\|_2^2 \\
& \text{s.t. } x^\star(\theta), z^\star(\theta) \in \arg \min_{x \in \mathbb{R}^n} \max_{z \in \mathbb{R}^{n_i}_+} L(x, z, s;\ \theta),
\end{aligned}
$$

$$
\text{with } L(x, z, s;\ \theta) \triangleq f(x;\ \theta) + z^\top (C(\theta)x - u(\theta) - s).
$$

A classical technique: **the Implicit Function Theorem.**

$$
G(x, z, t;\ \theta) \triangleq \begin{bmatrix} \nabla_x f(x;\ \theta) + C(\theta)^\top z \\ C(\theta)x - u(\theta) - t \\ [[t]_- + z]_+ - z \\ C(\theta)^\top [t]_+ \end{bmatrix}
$$

**Contribution:**
Extend the technique for the closest feasible QP solutions.

The map is **path-differentiable**.

$$
\left(\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\partial \theta}\right) \in \arg\min_w \left\| \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial v^\star} w + \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial \theta} \right\|_2^2,
$$

$$
\Pi \frac{\partial t^\star}{\partial \theta} \in \frac{\partial s^\star}{\partial \theta}, \text{ with } \Pi \in \partial([.]_+)(t^\star).
$$

**Contribution:**
Efficient algorithms to solve these problems.

# The backward pass: differentiating closest QP solutions

$$s^\star(\theta) = \arg\min_{s\in\mathbb{R}^{n_i}} \tfrac{1}{2}\|s\|_2^2$$
$$\text{s.t. } x^\star(\theta), z^\star(\theta) \in \arg\min_{x\in\mathbb{R}^n} \max_{z\in\mathbb{R}_+^{n_i}} L(x,z,s;\ \theta),$$

$$\text{with } L(x,z,s;\ \theta) \triangleq f(x;\ \theta) + z^\top(C(\theta)x - u(\theta) - s).$$

**Contribution:**
QPLayer: A full differentiable pipeline in C++ connected with PyTorch.

A classical technique: **the Implicit Function Theorem.**

$$G(x,z,t;\ \theta) \triangleq \begin{bmatrix} \nabla_x f(x;\ \theta) + C(\theta)^\top z \\ C(\theta)x - u(\theta) - t \\ [[t]_- + z]_+ - z \\ C(\theta)^\top [t]_+ \end{bmatrix}$$

**Contribution:**
Extend the technique for the closest feasible QP solutions.

The map is **path-differentiable**.

$$\left(\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\partial \theta}\right) \in \arg\min_w \left\| \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial v^\star} w + \frac{\partial G(x^\star, z^\star;\ \theta)}{\partial \theta} \right\|_2^2,$$

$$\Pi \frac{\partial t^\star}{\partial \theta} \in \frac{\partial s^\star}{\partial \theta}, \text{ with } \Pi \in \partial([.]_+)(t^\star).$$

**Contribution:**
Efficient algorithms to solve these problems.

# Numerical benchmark: back to the Sodoku problem.

Convex QP layers performs better than a ConvNet for solving Sudokus.

| | | | |
|---|---|---|---|
| | | | 3 |
| 1 | | | |
| | | 4 | |
| 4 | | | 1 |

| | | | |
|---|---|---|---|
| 2 | 4 | 1 | 3 |
| 1 | 3 | 2 | 4 |
| 3 | 1 | 4 | 2 |
| 4 | 2 | 3 | 1 |

**Figure**: Example of Sudoku.

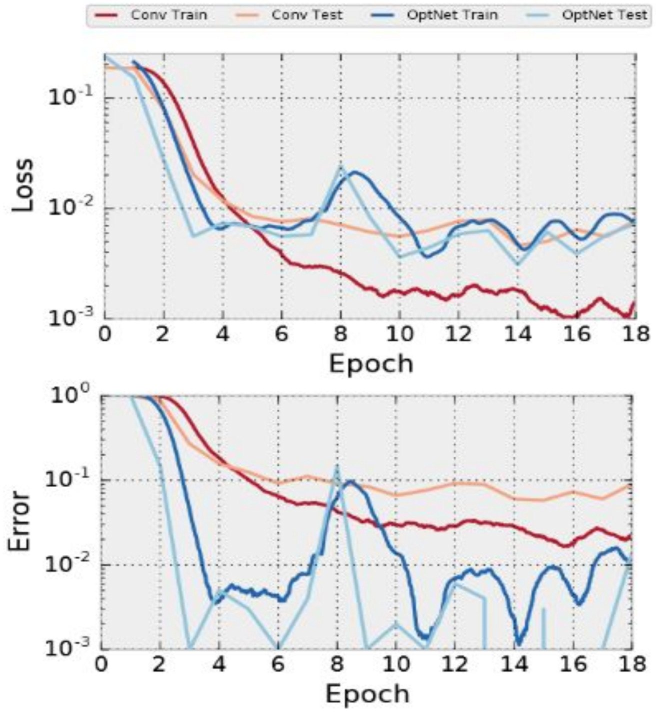
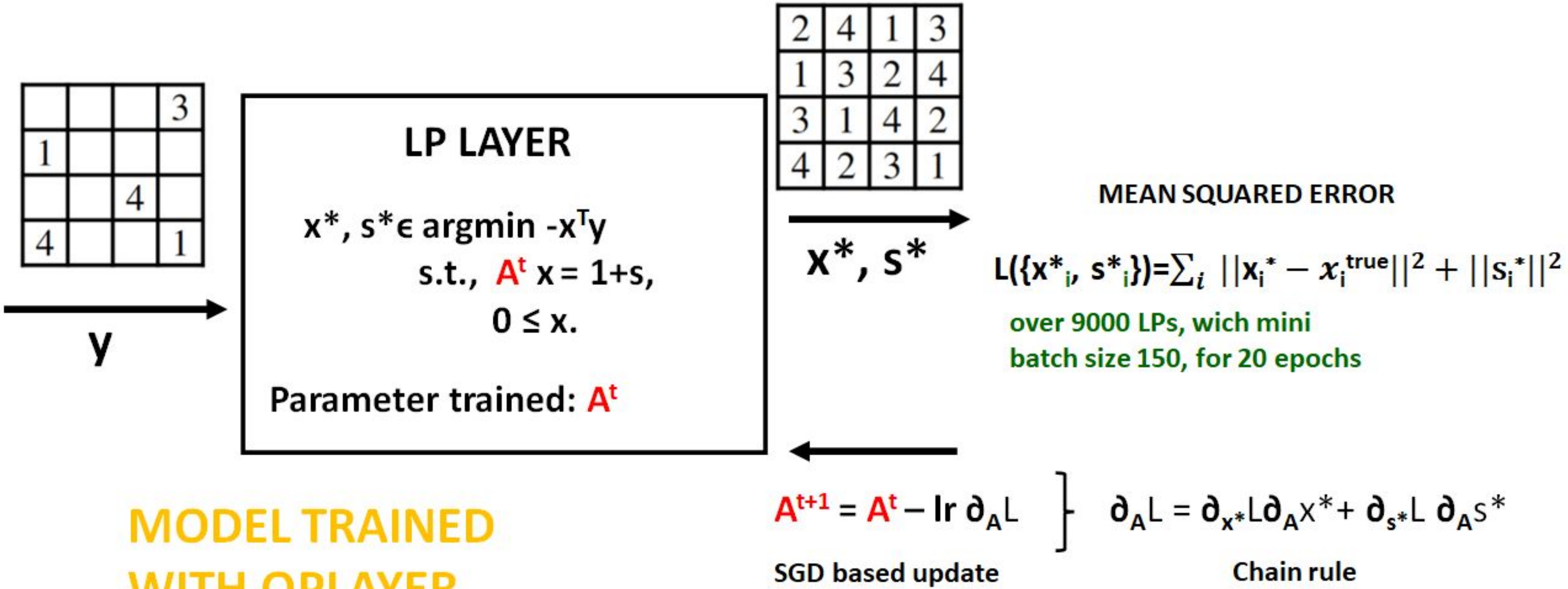



**Figure**: Training and test plots[1].

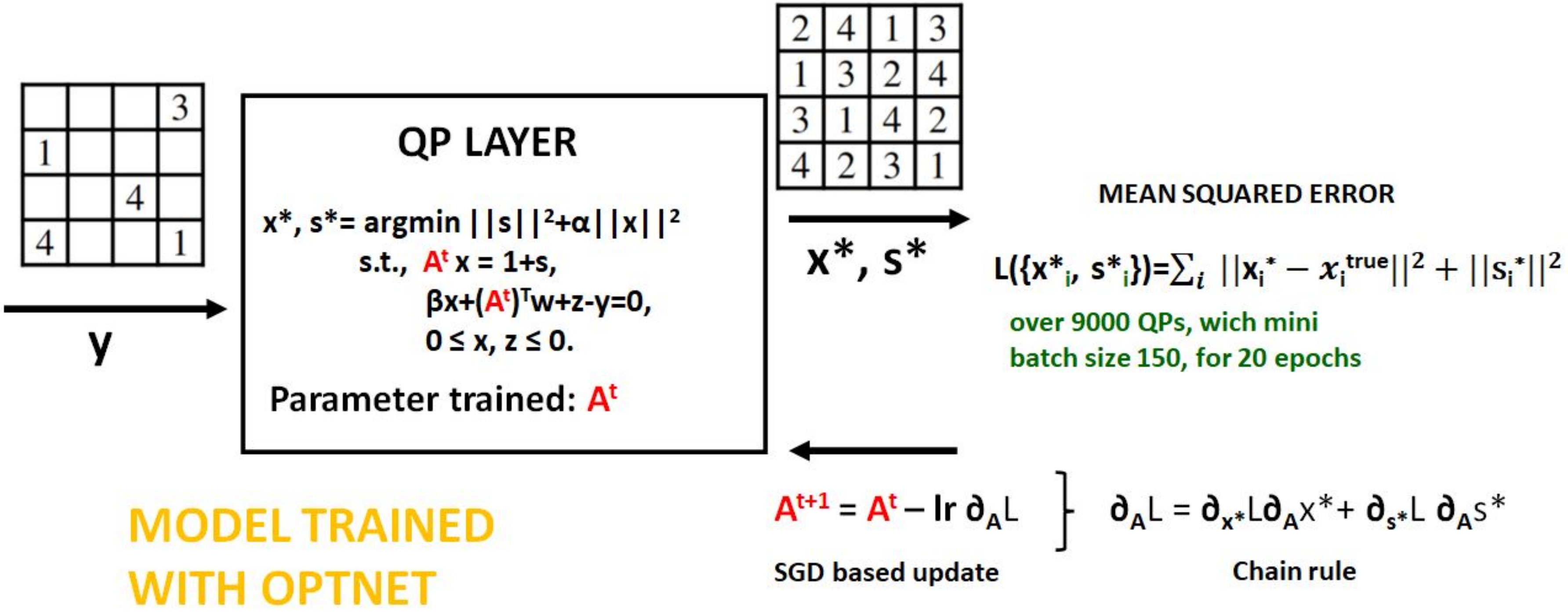
[1] B. Amos, Z. Kolter (2021)

# Architecture QPLayer-learn A

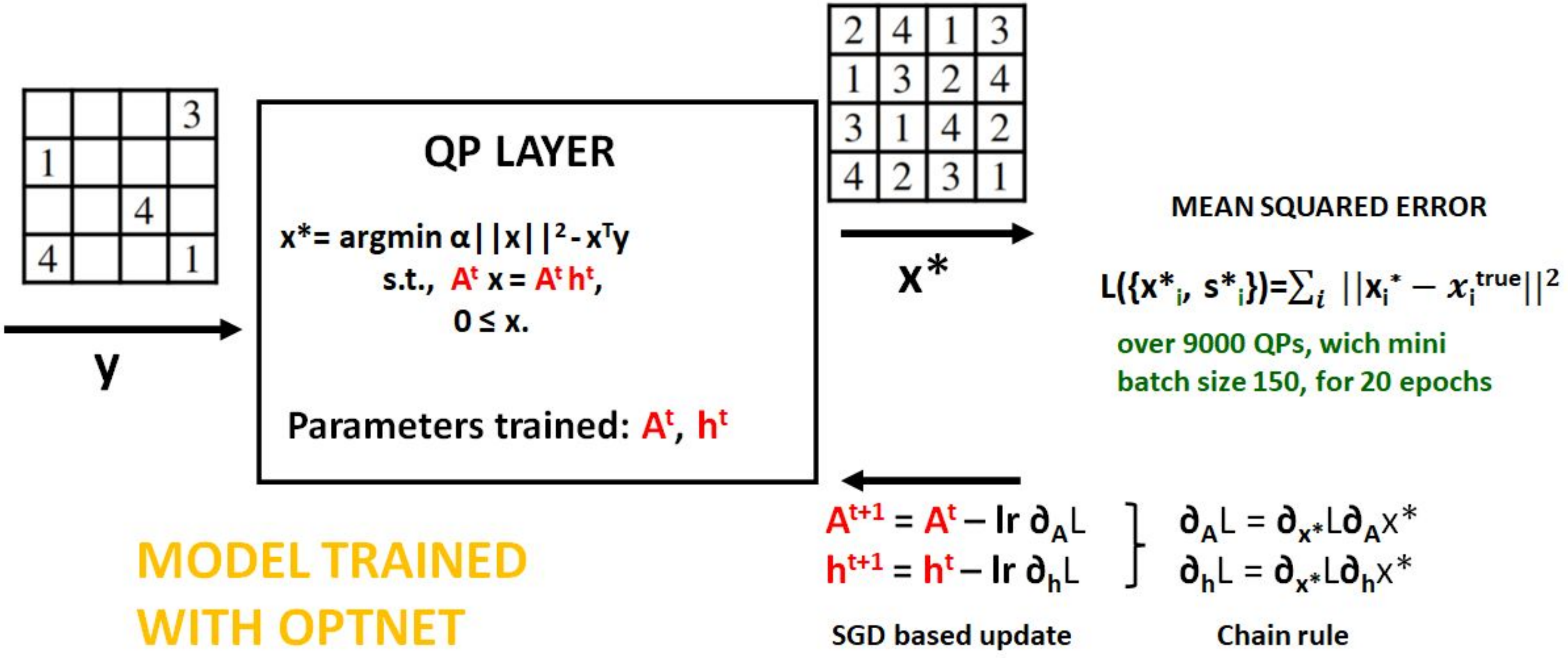



MODEL TRAINED
WITH QPLAYER

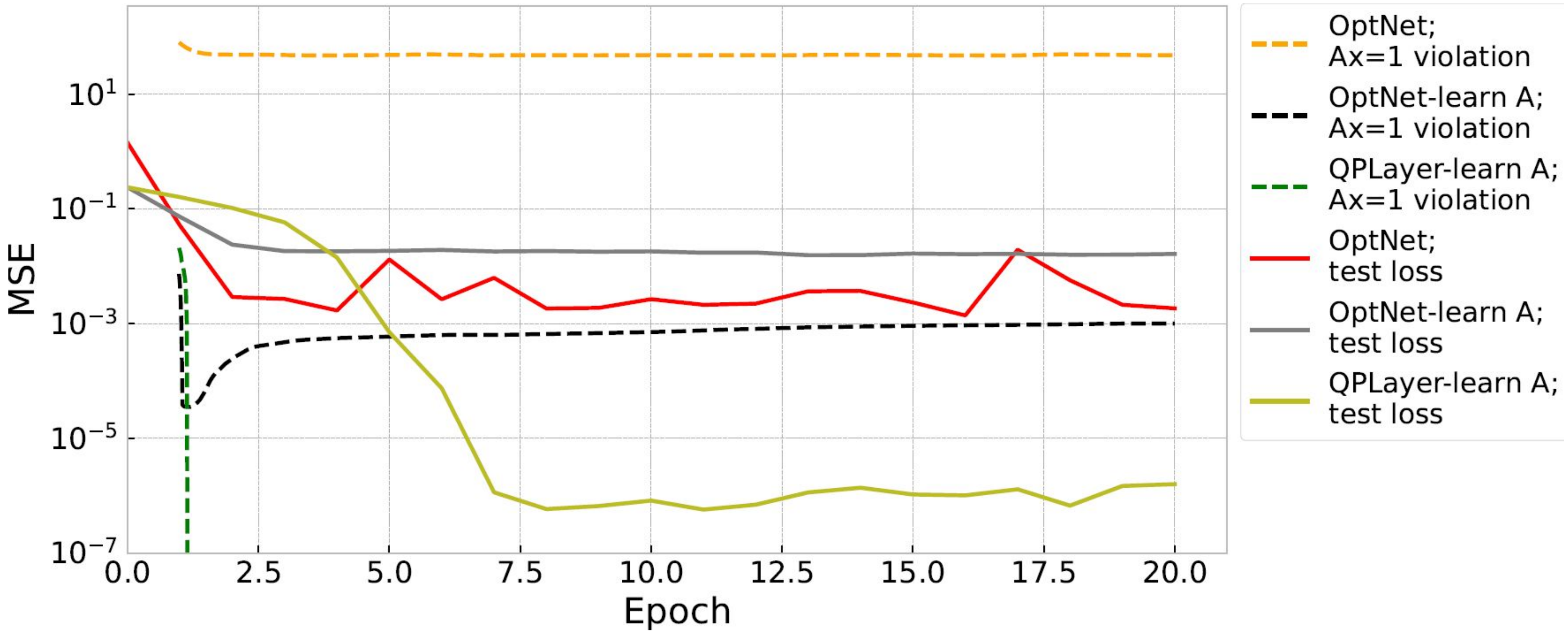
# Architecture OptNet-learn A



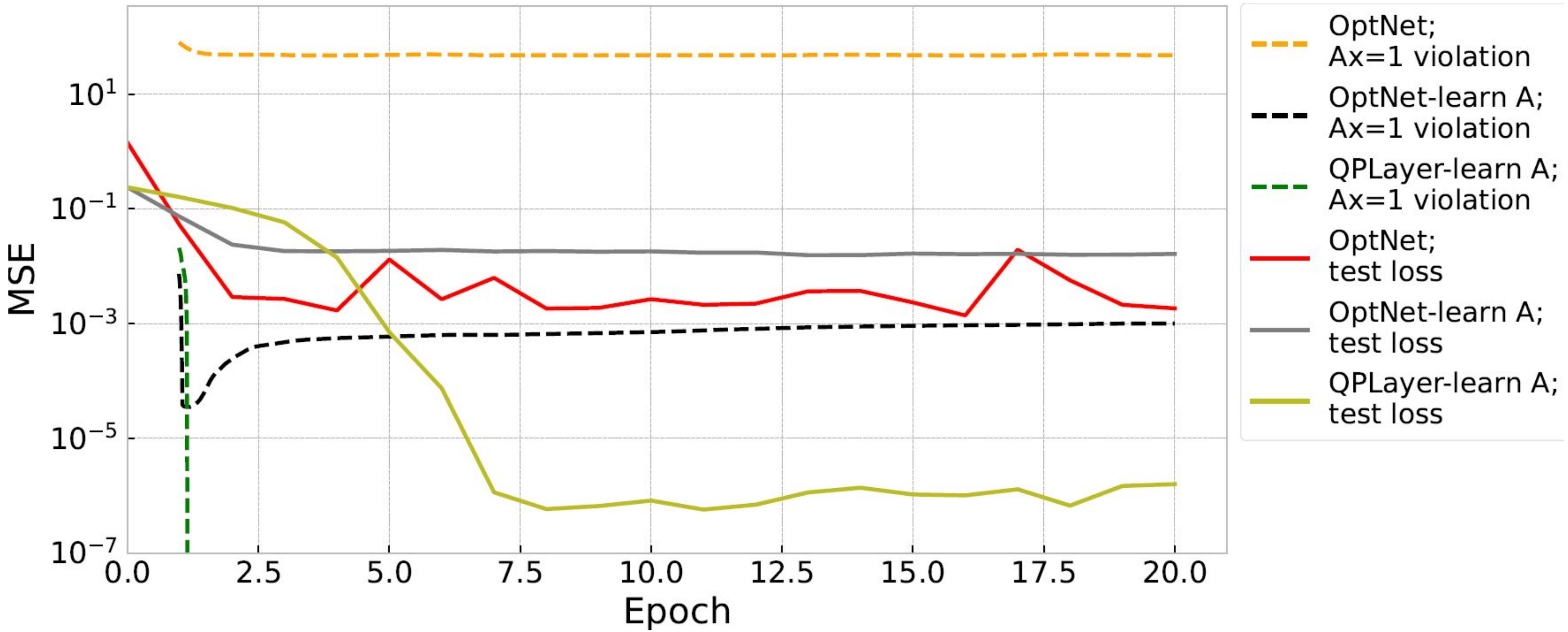
# Architecture OptNet



# Loss comparison

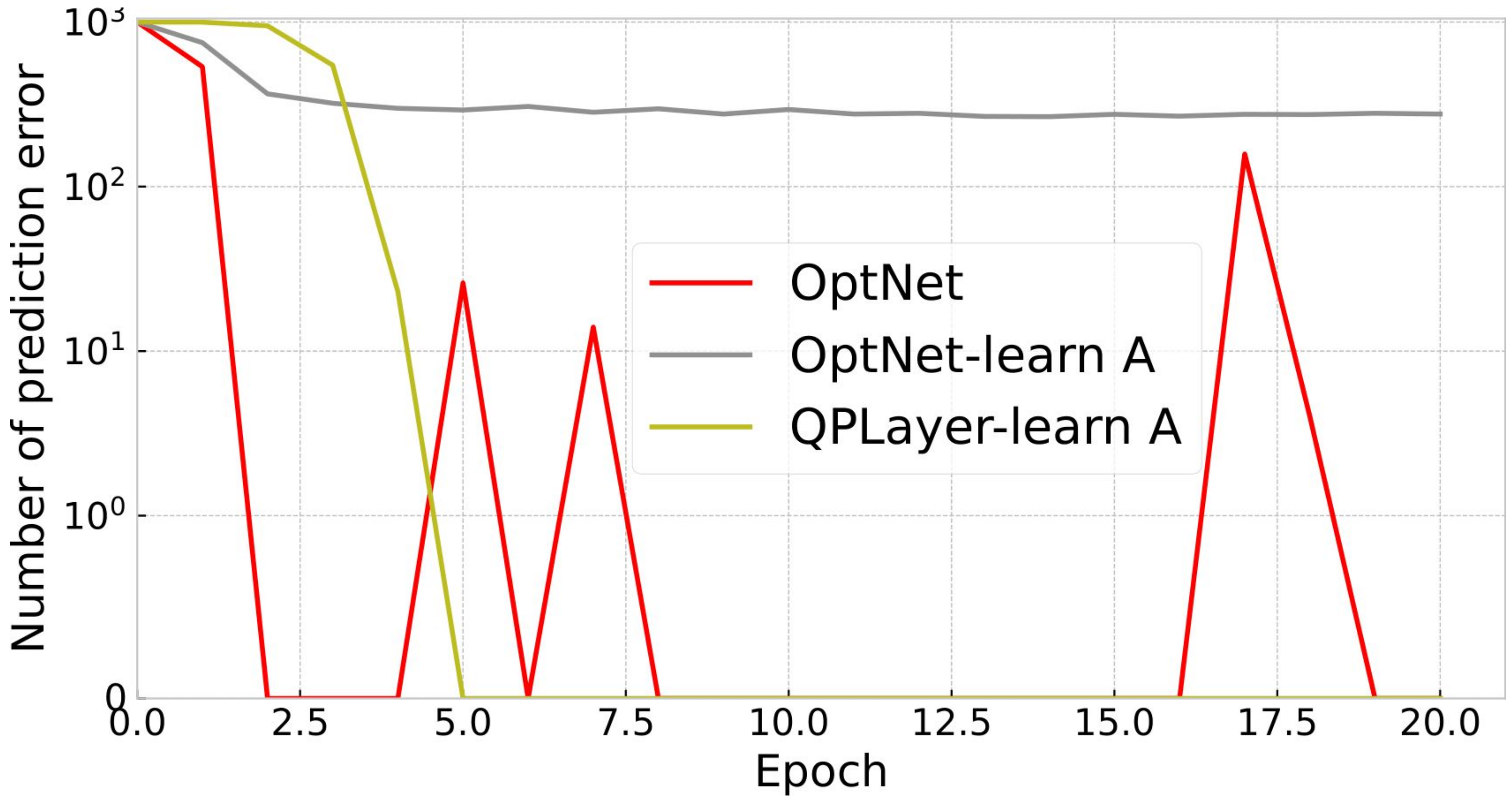


# Loss comparison




**ECJ least square error over all mini-batches = 0**

# Prediction error comparison



# Conclusion

Antoine Bambade[1,2]
[1] *Inria and ENS Paris : Willow and Sierra teams*
[2] *École des Ponts Paris Tech*

Ph.D defense, 15 January 2024

# Conclusion

| | Optimization | Differentiable optimization |
|---|---|---|
| Methodological contributions | ProxQP algorithm | IFT for closest feasible QPs<br>Extended Conservative Jacobian<br>Methodology for learning new QP layers |
| | **Augmented Lagrangian based methods** | |
| Software | ProxQP solver | QPLayer learning pipeline |
| | **ProxSuite library** | |
| Applications | Simulated problems,<br>Real robot | Classic learning tasks (denoising, object recognition, cartpole, Sudoku) |

# Conclusion

| | Optimization | Differentiable optimization |
|---|---|---|
| Methodological contributions | ProxQP algorithm | IFT for closest feasible QPs<br>Extended Conservative Jacobian<br>Methodology for learning new QP layers |
| | **Augmented Lagrangian based methods** | |
| Software | ProxQP solver | QPLayer learning pipeline |
| | **ProxSuite library** | |
| Applications | Simulated problems,<br>Real robot | Classic learning tasks (denoising, object recognition, cartpole, Sudoku) |
| Perspectives | Extension of MM property<br>Readjust control with infeasible perturbations<br>Conic solvers, non convex programming | Applications to control for more structured learning<br>Other conic constraints |

# Questions ?

# Maros Meszaros results

Today, on full test set (sparse backend), ProxQP is second or third.

## Low accuracy

Solvers are compared over the whole test set by shifted geometric mean (shm). Lower is better.

| | Success rate (%) | Runtime (shm) | Primal residual (shm) | Dual residual (shm) | Duality gap (shm) | Cost error (shm) |
|---|---|---|---|---|---|---|
| clarabel | 91.3 | 1.0 | 1.8 | 1086.8 | 1.0 | 1.0 |
| cvxopt | 42.8 | 21.8 | 3.5 | 3.9 | 6604.3 | 7.4 |
| gurobi | 16.7 | 106.5 | 3.6 | 23665.7 | 26882.1 | 45.5 |
| highs | 37.7 | 20.8 | 1.8 | 5.1 | 5469.8 | 8.0 |
| osqp | 21.0 | 19.7 | 2.9 | 3.2 | 4982.6 | 11.6 |
| proxqp | 78.3 | 7.9 | 1.0 | 1.0 | 19.6 | 2.9 |
| scs | 71.0 | 9.3 | 31.3 | 2.2 | 1.6 | 4.2 |

Source: S. Caron's last update qpsolvers benchmark

# Maros Meszaros results

Today, on full test set (sparse backend), ProxQP is second or third.

## High accuracy

Solvers are compared over the whole test set by shifted geometric mean (shm). Lower is better.

| | Success rate (%) | Runtime (shm) | Primal residual (shm) | Dual residual (shm) | Duality gap (shm) | Cost error (shm) |
|---|---|---|---|---|---|---|
| clarabel | 61.6 | 1.0 | 1.0 | 742803.1 | 44.9 | 1.0 |
| cvxopt | 5.8 | 5.8 | 1484115.0 | 126.3 | 1612205372.0 | 5.0 |
| gurobi | 5.1 | 19.2 | 4.3 | 7166137621.8 | 9769259351.6 | 19.8 |
| highs | 0.0 | 3.7 | 5416.6 | 884752.7 | 1987500500.6 | 3.5 |
| osqp | 26.1 | 11.5 | 2.8 | 1.2 | 3235.1 | 11.7 |
| proxqp | 59.4 | 2.4 | 1.4 | 1.0 | 5387.0 | 2.2 |
| scs | 42.8 | 8.0 | 2.5 | 1.0 | 1.0 | 7.6 |

Source: S. Caron's last update qpsolvers benchmark

# Exact penalty function approach by Fletcher

$$\min_x f(x) \quad \text{s.t.} \; Cx \le u, \qquad \text{(P)}$$

# Exact penalty function approach by Fletcher

$$\min_{x} f(x) \quad \text{s.t.} \; Cx \leq u, \qquad \text{(P)}$$

$$\min_{x,s} f(x) + \frac{1}{2\alpha}\|s\|^2 \quad \text{s.t.} \; Cx \leq u + s, \qquad (\text{P}(\alpha))$$

# Exact penalty function approach by Fletcher

$$\min_{x} f(x) \quad \text{s.t.} \quad Cx \leq u, \qquad \text{(P)}$$

$$\min_{x,s} f(x) + \frac{1}{2\alpha}\|s\|^2 \quad \text{s.t.} \quad Cx \leq u + s, \qquad (\text{P}(\alpha))$$

$$\min_{x \in \mathbb{R}^n, y \leq u} \|Cx - y\|_2^2, \qquad s^\star \stackrel{\text{def}}{=} Cx^\star - y^\star.$$

A. Chiche, J-C Gilbert (2016)

# Some (known) converging bias of PPA

$$x^\star = \mathrm{prox}_{\lambda f}(x^\star),$$

$$x^{k+1} = \mathrm{prox}_{\lambda f}(x^k),$$

# Some (known) converging bias of PPA

$$x^\star = \text{prox}_{\lambda f}(x^\star),$$

$$x^{k+1} = \text{prox}_{\lambda f}(x^k),$$

Güler (1991)

$$y^k \stackrel{\text{def}}{=} (x^k - x^{k-1})/\lambda \to v \in \overline{R(\partial f)}$$

# Some properties of the KKT map used

**KKT map for feasible QPs**:

$$G(x,z) = \begin{bmatrix} \nabla f(x) + C^\top z \\ [Cx - u + z]_+ - z \end{bmatrix}.$$

**Saddle subdifferential**:

$$\partial \mathcal{L}(x,z) \stackrel{\text{def}}{=} \begin{bmatrix} \nabla f(x) + C^\top z \\ \partial I_u^\star(z) - Cx \end{bmatrix},$$

$$0 \in \partial \mathcal{L}(x^\star, z^\star) \iff G(x^\star, z^\star) = 0.$$

**-Maximal monotone → can be used for PPA[1],**
**-Polyhedral mapping → outer Lipschitz continuous[2] (key for automatic scheduling).**

[1]E. K. Ruy, S. Boyd (2016); [2]A. L. Dontchev, R. T. Rockafellar (2009)

# Outer Lipschitz continuity and PMM

**Outer Lipschitz continuity**:

$$\exists\, a > 0,\ \tau > 0 \text{ s.t.},\quad \|(u,v)\| \leq \tau$$
$$\Longrightarrow$$
$$\text{dist}_{\partial\mathcal{L}^{-1}(0,0)}(x,z) \leq a\|(u,v)\|, \forall (x,z) \in \partial\mathcal{L}^{-1}(u,v)$$

**Key property of PMM[1]**:

$$\mu\|(x,z) - P_\mu(x,z)\| \leq \tau$$
$$\Longrightarrow$$
$$\text{dist}_{\partial\mathcal{L}^{-1}(0,0)}(P_\mu(x,z)) \leq \frac{a\mu}{\sqrt{1+a^2\mu^2}} \text{dist}_{\partial\mathcal{L}^{-1}(0,0)}(x,z).$$

[1] R. T. Rockafellar (1976)

# Outer Lipschitz continuity v.s. Lojasiewicz inequality

**Outer Lipschitz continuity**:

$$\exists\, a > 0,\ \tau > 0 \text{ s.t., } \ \|(u,v)\| \leq \tau$$
$$\Longrightarrow$$
$$\text{dist}_{\partial\mathcal{L}^{-1}(0,0)}(x,z) \leq a\|(u,v)\|, \forall (x,z) \in \partial\mathcal{L}^{-1}(u,v)$$

**Lojasiewicz inequality**:

$Z \stackrel{\text{def}}{=}$ zero locus of analytic function $f$. For any compact set $K$ in domain of f, there exists $\alpha, C$ s.t.,

$$\text{dist}(x,Z)^{\alpha} \leq C|f(x)|.$$

# More generic growth inequalities[1]

$$\exists \delta > 0 : \forall w \in B(0,\delta), \forall z \in T^{-1}(w), |z - T^{-1}(0)| \leq F(|w|).$$

1. F linear (typical QP case)
   ➔ Linear convergence of PMM (tight bound)
2. F power function with order s at least 1
   ➔ superlinear convergence of order at least s
3. F flat neighbourhood of 0 (and non negative)
   ➔ appropriate stopping criterion (Ar) provides superlinear convergence of order r
4. Growth exceeds any linear bounding
   ➔ sublinear convergence

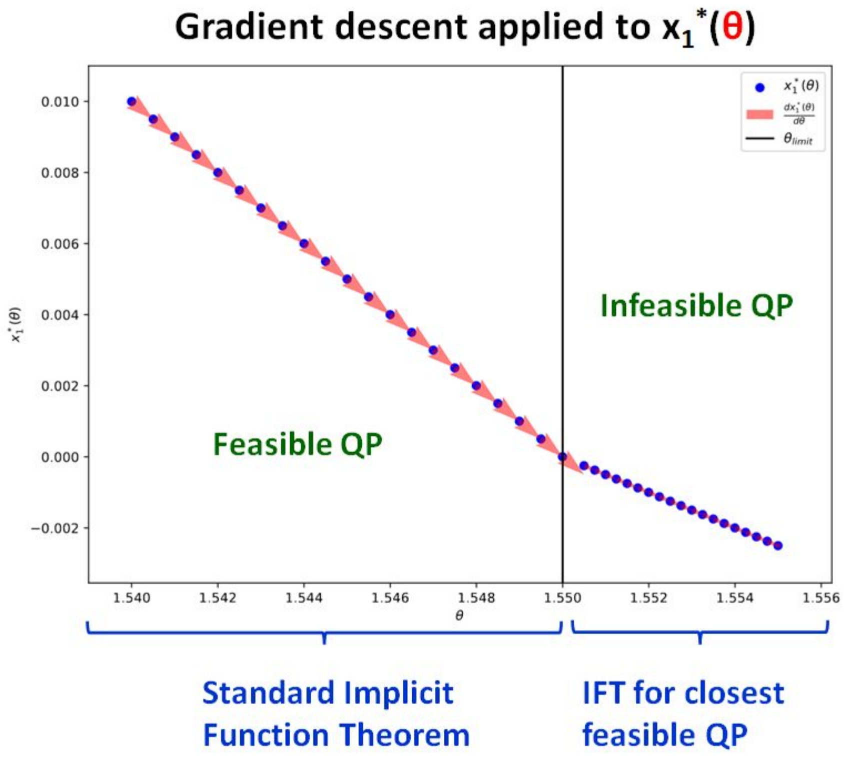
[1]F. J. Luque (1981)

# Why using Augmented Lagrangian methods?

OFFER

| Method | Early stopping | Warm start | Sparse and dense backends | Sub Familly | SOLVERS |
|---|---|---|---|---|---|
| Active Set | ✗ | ✗ | Dense | Primal | GALAHAD |
| | ✗ | ✗ | Dense | Dual | QUADPROG |
| | ✓ | ✓ | Dense | Dual | DAQP |
| | ✓ | ✓ | Dense | Dual | QPNNLS |
| | ✗ | ✓ | Dense | Parametric | QPOASES |
| Interior Point | ✓ | ✗ | Sparse | Primal-Dual | GUROBI |
| | | | Sparse | Primal-Dual | MOSEK |
| | | | Dense | Primal-Dual | CVXOPT |
| | | | Sparse | Primal-Dual | ECOS |
| | | | Sparse | Primal-Dual | QPSWIFT |
| | | | Dense | Primal-Dual | HPIPM |
| | | | Sparse | Primal-Dual | CLARABEL |
| | | | Sparse | Primal-Dual | BPMPD |
| | | | Sparse | Primal-Dual | OOQP |
| Augmented Lagrangian | ✓ | ✓ | Sparse | Primal | OSQP |
| | ✓ | ✓ | Sparse | Self Dual embedding | SCS |
| | ✓ | ✓ | Sparse | MM | LANCELOT |
| | ✓ | ✓ | Sparse | PMM | QPALM |
| | ✗ | ✗ | Sparse | PDPMM | QPDO |

# Solution outline: ideal pipeline




$$x_1^*(\theta), x_2^*(\theta) = \underset{x_1, x_2}{\operatorname{argmin}} \; x_1^2 + x_2^2$$

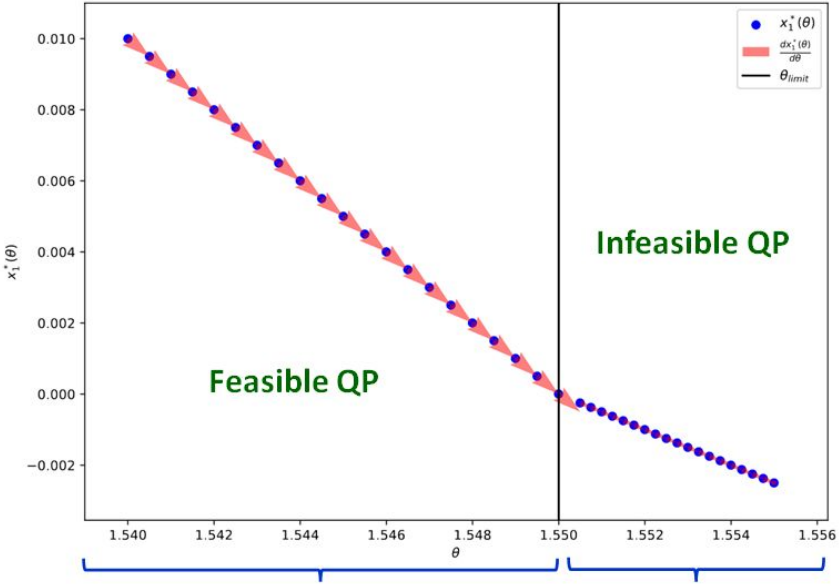
$$\text{s.t., } \theta \leq x_1 + x_2 \leq 1.55$$

$$1.5 \leq 2x_1 + x_2 \leq 1.55$$

Non singular

# Solution outline: ideal pipeline

Gradient descent applied to $x_1^*(\theta)$




Standard Implicit Function Theorem

IFT for closest feasible QP

$$x_1^*(\theta), x_2^*(\theta) = \underset{x_1, x_2}{\operatorname{argmin}} \quad x_1^2 + x_2^2$$

$$\text{s.t., } \theta \leq x_1 + x_2 \leq 1.55$$

$$1.5 \leq 2x_1 + x_2 \leq 1.55$$

Non singular

Feasible QP : $\nabla_\theta x_1^* = -1$

$\nabla_\theta x_2^* = 2$

Infeasible QP : $\nabla_\theta x_1^* = -0.5$

$\nabla_\theta x_2^* = 1.5$

# Solution outline: ideal pipeline

$$x_1^*(\theta), x_2^*(\theta) = \underset{x_1, x_2}{\operatorname{argmin}} \quad x_1 + x_2$$

$$\text{s.t., } \theta \le x_1 + x_2$$
$$1 \le x_1 \le 2$$
$$1 \le x_2 \le 2$$

singular

Infeasible LP : $\theta > 2$

# Solution outline: ideal pipeline

$$x_1^*(\theta), x_2^*(\theta) = \underset{x_1, x_2}{\text{argmin}} \quad x_1 + x_2$$

$$\text{s.t.}, \; \theta \le x_1 + x_2$$
$$1 \le x_1 \le 2$$
$$1 \le x_2 \le 2$$

singular

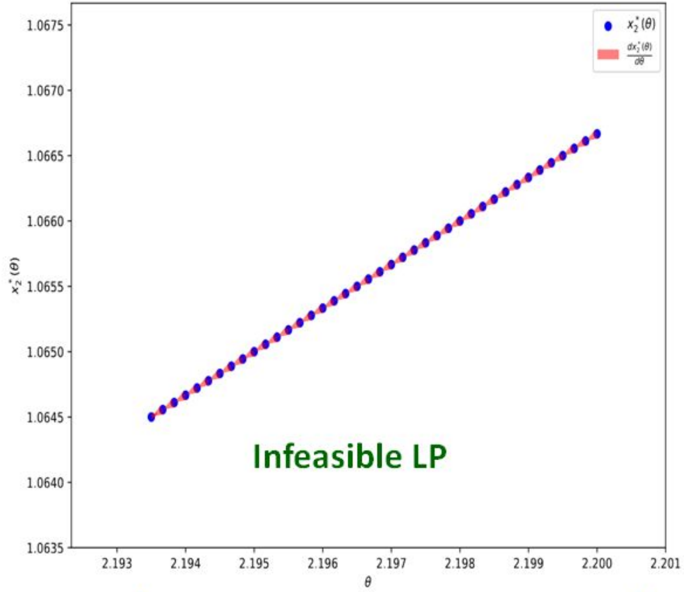
Infeasible LP : $\theta > 2$

Closest feasible LP

$$\left.\begin{aligned} x_1^*(\theta) &= 1 + s^* \\ x_2^*(\theta) &= 1 + s^* \\ \theta - s^* &= x_1^*(\theta) + x_2^*(\theta) \end{aligned}\right] \quad \begin{aligned} x_1^*(\theta) &= (1+\theta)/3 \\ x_2^*(\theta) &= (1+\theta)/3 \end{aligned}$$

# Solution outline: ideal pipeline

Gradient descent applied to $x_2^*(\theta)$




Exact Gradient obtained through least square

$$x_1^*(\theta), x_2^*(\theta) = \underset{x_1, x_2}{\text{argmin}} \quad x_1 + x_2$$

$$\text{s.t.,} \quad \theta \leq x_1 + x_2$$
$$1 \leq x_1 \leq 2$$
$$1 \leq x_2 \leq 2$$

singular

Infeasible LP : $\theta > 2$

Closest feasible LP

$$x_1^*(\theta) = 1 + s^*$$
$$x_2^*(\theta) = 1 + s^*$$
$$\theta - s^* = x_1^*(\theta) + x_2^*(\theta)$$

$$x_1^*(\theta) = (1+\theta)/3$$
$$x_2^*(\theta) = (1+\theta)/3$$