# Automated tight Lyapunov analysis for first-order splitting methods

**Pontus Giselsson**

Collaborators

Manu Upadhyaya

Sebastian Banert

Adrien Taylor

# Outline

- **Introduction to first-order methods**
- Examples of first-order methods
- Setting and usage preview
- Algorithm representation
- Lyapunov analysis and main result
- Algorithm examples

# Large-scale optimization

- Many contemporary optimization problems are large-scale
  - Found, e.g., in machine learning applications
  - Billions of decision variables

# Large-scale optimization

- Many contemporary optimization problems are large-scale
  - Found, e.g., in machine learning applications
  - Billions of decision variables
- Algorithms based on linear system solves inapplicable
  - Newton's method
  - Interior point methods
  - Active set methods

# Large-scale optimization

- Many contemporary optimization problems are large-scale
  - Found, e.g., in machine learning applications
  - Billions of decision variables
- Algorithms based on linear system solves inapplicable
  - Newton's method
  - Interior point methods
  - Active set methods
- Need algorithms with lower per iteration cost
  - First-order methods
  - Stochastic first-order methods
  - Coordinate-wise first-order methods

# Large-scale optimization

- Many contemporary optimization problems are large-scale
  - Found, e.g., in machine learning applications
  - Billions of decision variables
- Algorithms based on linear system solves inapplicable
  - Newton's method
  - Interior point methods
  - Active set methods
- Need algorithms with lower per iteration cost
  - **First-order methods**
  - Stochastic first-order methods
  - Coordinate-wise first-order methods

# First-order splitting methods

- We consider first-order methods for finite-sum problems

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{m} f_i(x)$$

and we assume all $f_i$ are convex, but potentially nonsmooth

# First-order splitting methods

- We consider first-order methods for finite-sum problems

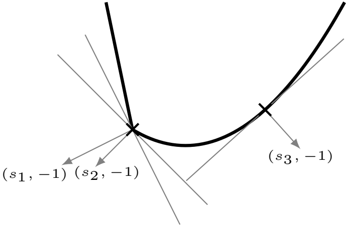
$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{m} f_i(x)$$

  and we assume all $f_i$ are convex, but potentially nonsmooth
- A first-order method evaluates each subgradient $\partial f_i$ either
  - explicitly (via direct evaluation, gradient if $f$ differentiable) or
  - implicitly (via *proximal operator*)

  and linearly combines the results to form iterations

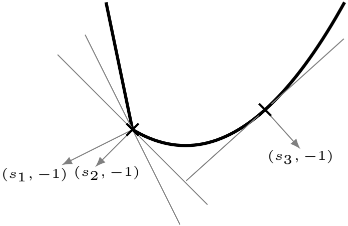
# Subgradients

- A subgradient of $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ at $x \in \mathbb{R}^n$
  - defines the slope $s$ of an affine minorizer to $f$
  - the affine minorizor coincides with $f$ at $x$
  - coincides (if exists) with gradient at differentiable points
  - $(s, -1)$ defines normal to epigraph of $f$



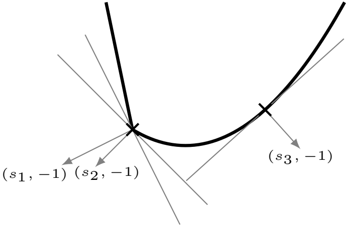
# Subgradients

- A subgradient of $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ at $x \in \mathbb{R}^n$
  - defines the slope $s$ of an affine minorizer to $f$
  - the affine minorizor coincides with $f$ at $x$
  - coincides (if exists) with gradient at differentiable points
  - $(s, -1)$ defines normal to epigraph of $f$
- The set of subgradients at $x$ is called subdifferential at $x$ ($\partial f(x)$)



# Subgradients

- A subgradient of $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ at $x \in \mathbb{R}^n$
  - defines the slope $s$ of an affine minorizer to $f$
  - the affine minorizor coincides with $f$ at $x$
  - coincides (if exists) with gradient at differentiable points
  - $(s, -1)$ defines normal to epigraph of $f$
- The set of subgradients at $x$ is called subdifferential at $x$ ($\partial f(x)$)
- For convex $f$ subgradient exists at least on interior of domain of $f$



# Proximal operator

- The proximal operator is defined as

$$\mathrm{prox}_{\gamma g}(v) = \underset{x}{\mathrm{argmin}} \left( g(x) + \frac{1}{2\gamma}\|x - v\|^2 \right)$$

for some step size $\gamma > 0$

# Proximal operator

- The proximal operator is defined as

$$\operatorname{prox}_{\gamma g}(v) = \underset{x}{\operatorname{argmin}} \left( g(x) + \frac{1}{2\gamma} \|x - v\|^2 \right)$$

  for some step size $\gamma > 0$

- Optimality condition (for proper lower-semicontinuous convex $g$)

$$\gamma^{-1}(v - x) \in \partial g(x)$$

  i.e., $\gamma^{-1}(v - x)$ is subgradient of $g$ at $x$ (implicit step)

# Proximal operator

- The proximal operator is defined as

$$\text{prox}_{\gamma g}(v) = \underset{x}{\text{argmin}} \left( g(x) + \frac{1}{2\gamma} \|x - v\|^2 \right)$$

for some step size $\gamma > 0$

- Optimality condition (for proper lower-semicontinuous convex $g$)

$$\gamma^{-1}(v - x) \in \partial g(x)$$

i.e., $\gamma^{-1}(v - x)$ is subgradient of $g$ at $x$ (implicit step)

- Projection is special case with $g = \iota_C$ where

$$\iota_C(x) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{else} \end{cases}$$

i.e., $\text{prox}_{\gamma \iota_C} = \Pi_C$, where $\Pi_C$ is orthorgonal projection onto $C$

# Problem formulation via subgradients

- The problem of solving

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{m} f_i(x)$$

is, given some mild constraint qualification, equivalent to

$$\text{find } x \in \mathbb{R}^n \text{ such that } 0 \in \sum_{i=1}^{m} \partial f_i(x)$$

## Problem formulation via subgradients

- The problem of solving

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{m} f_i(x)$$

  is, given some mild constraint qualification, equivalent to

$$\text{find } x \in \mathbb{R}^n \text{ such that } 0 \in \sum_{i=1}^{m} \partial f_i(x)$$

- An inclusion problem that is solved by first-order splitting methods

# Outline

- Introduction to first-order methods
- **Examples of first-order methods**
- Setting and usage preview
- Algorithm representation
- Lyapunov analysis and main result
- Algorithm examples

# Gradient method

- Solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x)$$

where $f$ is differentiable

- Iteration given by

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$$

where $\gamma_k > 0$, i.e., take step in negative gradient direction

- Explicit evaluation of (sub)gradient

# Proximal gradient method

- Solves

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\, f_1(x) + f_2(x)$$

  where $f_1$ differentiable and $f_2$ potentially nonsmooth

- Iterates gradient step followed by proximal operator evaluation:

$$x_{k+1} = \text{prox}_{\gamma_k f_2}(x_k - \gamma_k \nabla f_1(x_k))$$

- Explicit and implicit evaluation

# Momentum variations

- Nesterov acceleration variation of proximal gradient method

$$y_k = x_k + \theta_k(x_k - x_{k-1})$$
$$x_{k+1} = \mathrm{prox}_{\gamma_k f_2}(y_k - \gamma_k \nabla f_1(y_k))$$

  where $\theta_k = \frac{k-1}{k+2}$ (for instance)

- Polyak momentum variation of proximal gradient method

$$x_{k+1} = \mathrm{prox}_{\gamma_k f_2}(x_k - \gamma_k \nabla f_1(x_k)) + \theta_k(x_k - x_{k-1})$$

# Douglas–Rachford splitting

- Solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f_1(x) + f_2(x)$$

  where $f_1$ and $f_2$ can be nonsmooth
- Algorithm uses two implicit steps

$$\begin{aligned} x_k &= \text{prox}_{\gamma_k f_1}(z_k) \\ y_k &= \text{prox}_{\gamma_k f_2}(2x_k - z_k) \\ z_{k+1} &= z_k + \lambda_k(y_k - x_k) \end{aligned}$$

- With proper choice of $f_1$ and $f_2$ we get ADMM
- Momentum variations and multi-block extensions exist

# Chambolle–Pock

- Solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f_1(x) + f_2(Lx)$$

where $f_1$ and $f_2$ can be nonsmooth

- Algorithm uses two implicit steps and explicit evaluation of $L$

$$x_{k+1} = \text{prox}_{\tau f_1}(x_k - \tau L^* y_k)$$
$$y_{k+1} = \text{prox}_{\sigma f_2^*}(y_k + \sigma L(2x_k - x_{k-1}))$$

where $f_2^*$ is conjugate function of $f_2$

- Does not entirely fit our framework, but with $L = \text{Id}$ it does

# Other first-order methods

- The Condat–Vu method
- Projective splitting
- The Davis–Yin method
- Minimal lifting methods by Ryu/Malitsky Tam
- Asymmetric forward–backward adjoint splitting
- Forward–backward–forward splitting
- Many more primal–dual methods
- Many momentum variations

# Outline

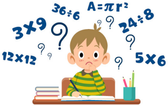
- Introduction to first-order methods
- Examples of first-order methods
- **Setting and usage preview**
- Algorithm representation
- Lyapunov analysis and main result
- Algorithm examples

# Our work

- Methodology for proving first-order algorithm convergence
- Focus on first-order methods for convex optimization that use
  - proximal operator or gradient evaluations
  - scalar multiplications and vector additions with fixed coefficients

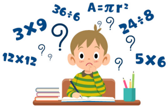
# Proving convergence

- Traditional way:



# Proving convergence

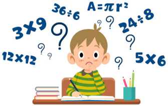
- Traditional way:




- Modern way with computer assisted PEP and IQC:

# Proving convergence

- Traditional way:

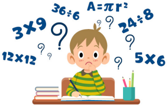



- Modern way with computer assisted PEP and IQC:

- End goal?:

# Proving convergence

- Traditional way:

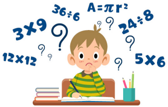



- Modern way with computer assisted PEP and IQC:

- End goal?:

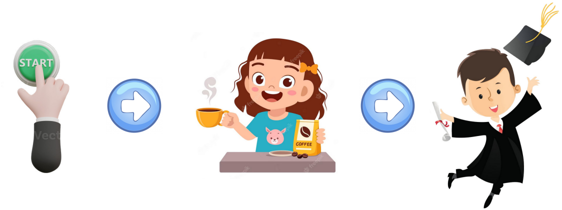

# Proving convergence

- Traditional way:



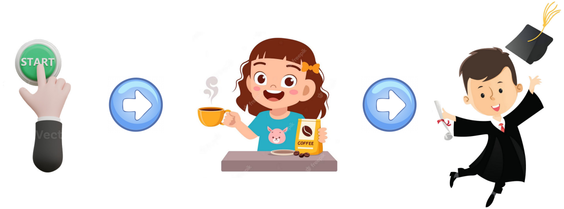

- Modern way with computer assisted PEP and IQC:

- End goal?:



# Towards end goal

- End goal:




- Have contributed to this with automatic Lyapunov analysis

## Example: What we achieved while drinking coffee

- Chambolle–Pock ("with $L = \mathrm{Id}$"): $\underset{x \in \mathcal{H}}{\mathrm{minimize}}(f_1(x) + f_2(x))$

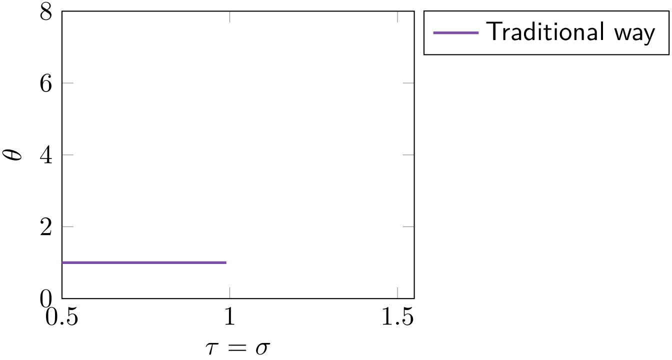
$$x_{k+1} = \mathrm{prox}_{\tau f_1}(x_k - \tau y_k)$$
$$y_{k+1} = \mathrm{prox}_{\sigma f_2^*}(y_k + \tau_2\,(x_{k+1} + \theta(x_{k+1} - x_k)))$$

# Example: What we achieved while drinking coffee

- Chambolle–Pock ("with $L = \mathrm{Id}$"): $\underset{x\in\mathcal{H}}{\text{minimize}}(f_1(x) + f_2(x))$

$$x_{k+1} = \mathrm{prox}_{\tau f_1}(x_k - \tau y_k)$$
$$y_{k+1} = \mathrm{prox}_{\sigma f_2^*}(y_k + \tau_2 (x_{k+1} + \theta(x_{k+1} - x_k)))$$

- Convergent parameter choices (primal-dual gap, $f_1$ and $f_2$ pcc)



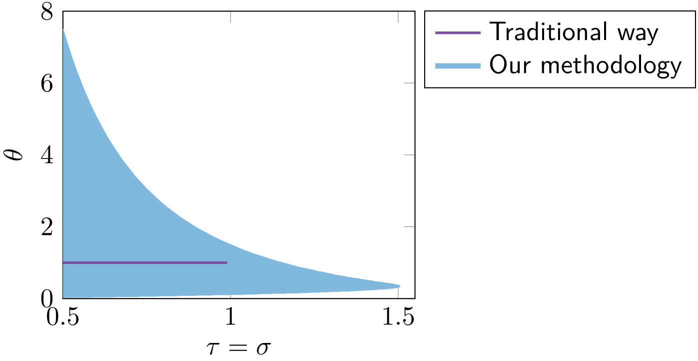

(Caveat: verified on a $0.01 \times 0.01$ grid of region)

# Example: What we achieved while drinking coffee

- Chambolle–Pock ("with $L = \mathrm{Id}$"): $\underset{x \in \mathcal{H}}{\text{minimize}}(f_1(x) + f_2(x))$

$$x_{k+1} = \mathrm{prox}_{\tau f_1}(x_k - \tau y_k)$$
$$y_{k+1} = \mathrm{prox}_{\sigma f_2^*}(y_k + \tau_2\,(x_{k+1} + \theta(x_{k+1} - x_k)))$$

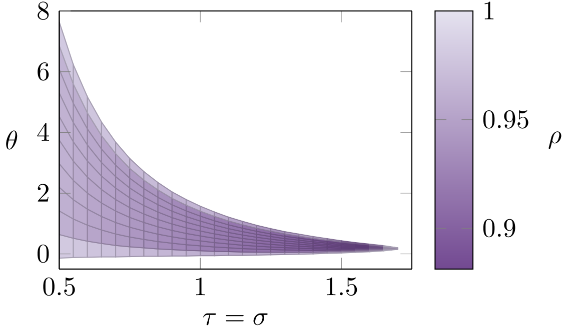
- Convergent parameter choices (primal-dual gap, $f_1$ and $f_2$ pcc)




(Caveat: verified on a $0.01 \times 0.01$ grid of region)

# Chambolle–Pock linear convergence

- Tight contraction rate–both 0.05-strongly convex and 50-smooth:




- Improved rate with larger $\tau = \sigma$

# Chambolle–Pock linear convergence

- Optimal convergence rate for different parameter restrictions[1]

| Parameter restriction | $\tau = \sigma$ | $\theta$ | $\rho$ |
|---|---|---|---|
| All convergent | 1.6 | 0.22 | 0.8812 |
| Cvx+cvx convergent | 1.5 | 0.35 | 0.8891 |
| Traditional | 0.99 | 1 | 0.9266 |
| DR | 1 | 1 | 0.9234 |

- Better rates outside traditional region

[1] for points evaluated on our $0.01 \times 0.01$ grid

# Setting – More formally

- Let $\mathcal{F}_{\sigma_i,\beta_i}$ be class of $\sigma_i$-strongly convex and $\beta_i$-smooth functions
- Convex optimization problems

$$\underset{y\in\mathcal{H}}{\text{minimize}} \sum_{i=1}^{m} f_i(y)$$

  where each $f_i \in \mathcal{F}_{\sigma_i,\beta_i}$ with $0 \leq \sigma_i < \beta_i \leq \infty$
- Associated inclusion problem

$$\text{find } y \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{m} \partial f_i(y)$$

  where $\partial f_i$ are subdifferential operators
- Problem class $\boldsymbol{\mathcal{F}}_{\boldsymbol{\sigma},\boldsymbol{\beta}}$: $f_i \in \mathcal{F}_{\sigma_i,\beta_i}$ and inclusion solvable

# Main result statement

Given a first-order method for an inclusion problem class, we provide

- a necessary and sufficient condition for the existence of a *quadratic Lyapunov inequality* (with a very general ansatz)
- a quadratic Lyapunov inequality if one exists

## The necessary and sufficient condition

- Condition is feasibility of (small) semi-definite program
- Derived with inspiration from
  - performance estimation (PEP) (Drori and Teboulle, Taylor et al.)
  - integral quadratic constraints (IQC) (Lessard et al.)
  - tight automated analysis framework (Taylor/Van Scoy/Lessard)
  - Lyapunov analysis (Taylor/Bach)
- Based on specific algorithm representation for wide applicability

# Outline

- Introduction to first-order methods
- Examples of first-order methods
- Setting and usage preview
- **Algorithm representation**
- Lyapunov analysis and main result
- Algorithm examples

# Algorithm representation

- Algorithm representation on state space form[1]:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= (A \otimes \mathrm{Id})\boldsymbol{x}_k + (B \otimes \mathrm{Id})\boldsymbol{u}_k \\ \boldsymbol{y}_k &= (C \otimes \mathrm{Id})\boldsymbol{x}_k + (D \otimes \mathrm{Id})\boldsymbol{u}_k \\ \boldsymbol{u}_k &\in \boldsymbol{\partial f}(\boldsymbol{y}_k) \\ \boldsymbol{F}_k &= \boldsymbol{f}(\boldsymbol{y}_k), \end{aligned}$$

  where different $(A, B, C, D)$ give rise to different algorithms
- Product space notation for function and subdifferentials

$$\boldsymbol{f}(\boldsymbol{y}) = \Big(f_1\Big(y^{(1)}\Big), \ldots, f_m\Big(y^{(m)}\Big)\Big), \qquad \boldsymbol{\partial f}(\boldsymbol{y}) = \prod_{i=1}^{m} \partial f_i\Big(y^{(i)}\Big)$$

  where

$$\boldsymbol{y} = \Big(y^{(1)}, \ldots, y^{(m)}\Big), \quad \boldsymbol{u} = \Big(u^{(1)}, \ldots, u^{(m)}\Big), \quad \boldsymbol{x} = \Big(x^{(1)}, \ldots, x^{(n)}\Big)$$

  meaning $u_k^{(i)} \in \partial f_i(y_k^{(i)})$ for all $i \in [\![1, m]\!]$
- Linear dynamical system in feedback with subdifferentials

[1] Model used in control literature, Lessard et al. 2016, and similar to model in Morin/Banert/Giselsson

# Algorithms that fit framework

- All first-order methods with
  - iteration-independent parameters
  - exactly one subdifferential evaluation per iteration and function

  fit the framework
- Many of the methods we have seen fit framework

# Chambolle–Pock

- Algorithm (with $L = \mathrm{Id}$):

$$\begin{aligned} x_{k+1} &= \mathrm{prox}_{\tau_1 f_1}(x_k - \tau y_k), \\ y_{k+1} &= \mathrm{prox}_{\tau_2 f_2^*}(y_k + \tau_2 (x_{k+1} + \theta(x_{k+1} - x_k))) \end{aligned}$$

- Algorithm in our state-space representation:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \left( \begin{bmatrix} 1 & -\tau_1 \\ 0 & 0 \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{x}_k + \left( \begin{bmatrix} -\tau_1 & 0 \\ 0 & 1 \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{u}_k, \\ \boldsymbol{y}_k &= \left( \begin{bmatrix} 1 & -\tau_1 \\ 1 & \frac{1}{\tau_2} - \tau_1(1+\theta) \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{x}_k + \left( \begin{bmatrix} -\tau_1 & 0 \\ -\tau_1(1+\theta) & -\frac{1}{\tau_2} \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{u}_k, \\ \boldsymbol{u}_k &\in \boldsymbol{\partial f}(\boldsymbol{y}_k), \end{aligned}$$

- Algorithm parameters appear in $(A, B, C, D)$

## Proximal gradient method with heavy-ball momentum

- Algorithm:

$$x_{k+1} = \operatorname{prox}_{\gamma f_2}(x_k - \gamma \nabla f_1(x_k) + \delta_1(x_k - x_{k-1})) + \delta_2(x_k - x_{k-1})$$

- Algorithm in our state-space representation:

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \left( \begin{bmatrix} 1+\delta_1+\delta_2 & -\delta_1-\delta_2 \\ 1 & 0 \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{x}_k + \left( \begin{bmatrix} -\gamma & -\gamma \\ 0 & 0 \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{u}_k \\
\boldsymbol{y}_k &= \left( \begin{bmatrix} 1 & 0 \\ 1+\delta_1 & -\delta_1 \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{x}_k + \left( \begin{bmatrix} 0 & 0 \\ -\gamma & -\gamma \end{bmatrix}_{\mathrm{Id}} \right) \boldsymbol{u}_k, \\
\boldsymbol{u}_k &\in \boldsymbol{\partial f}(\boldsymbol{y}_k),
\end{aligned}$$

- Algorithm parameters appear in $(A, B, C, D)$
- Same structure as previous algorithm, just new $(A, B, C, D)$

# Algorithm fixed points

- Algorithm fixed points $\boldsymbol{\xi}_\star = (\boldsymbol{x}_\star, \boldsymbol{u}_\star, \boldsymbol{y}_\star, \boldsymbol{F}_\star)$ satisfy

$$
\begin{aligned}
\boldsymbol{x}_\star &= (A \otimes \mathrm{Id})\boldsymbol{x}_\star + (B \otimes \mathrm{Id})\boldsymbol{u}_\star \\
\boldsymbol{y}_\star &= (C \otimes \mathrm{Id})\boldsymbol{x}_\star + (D \otimes \mathrm{Id})\boldsymbol{u}_\star \\
\boldsymbol{u}_\star &\in \boldsymbol{\partial f}(\boldsymbol{y}_\star) \\
\boldsymbol{F}_\star &= \boldsymbol{f}(\boldsymbol{y}_\star)
\end{aligned}
$$

- Algorithm objective: find fixed point $\boldsymbol{\xi}_\star$, extract solution from $\boldsymbol{\xi}_\star$

# Fixed-point encoding property

- We are only interested in algorithms $(A, B, C, D)$ such that

$$\text{finding a fixed point} \qquad \iff \qquad \text{solving inclusion problem}$$

- More specifically:
  - from each solution, it should be possible to construct fixed point
  - from each fixed point, it should be possible to extract solution
- Such algorithms have the *fixed-point encoding property* (FPEP)

# Restrictions on $(A, B, C, D)$

- Let

$$N = \begin{bmatrix} I \\ -\mathbf{1}^\top \end{bmatrix} \in \mathbb{R}^{m \times (m-1)}$$

- Result:

*The algorithm has the fixed-point encoding property*

$$\Longleftrightarrow$$

*The matrices* $(A, B, C, D)$ *satisfy*

$$\operatorname{ran} \begin{bmatrix} BN & 0 \\ DN & -\mathbf{1} \end{bmatrix} \subseteq \operatorname{ran} \begin{bmatrix} I - A \\ -C \end{bmatrix}$$

$$\operatorname{null} \begin{bmatrix} I - A & -B \end{bmatrix} \subseteq \operatorname{null} \begin{bmatrix} N^\top C & N^\top D \\ 0 & \mathbf{1}^\top \end{bmatrix},$$

*(block row/column containing* $N^\top/N$ *removed when* $m = 1$*)*

- $(A, B, C, D)$ of algorithms that “work” satisfy FPEP conditions

# Outline

- Introduction to first-order methods
- Examples of first-order methods
- Setting and usage preview
- Algorithm representation
- **Lyapunov analysis and main result**
- Algorithm examples

# Lyapunov analysis

- We use quadratic $(P, p, T, t, \rho)$-*Lyapunov inequalities*:

  C1. $V(\boldsymbol{\xi}_+, \boldsymbol{\xi}_\star) \leq \rho V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) - R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star)$
  
  C2. $V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq \mathcal{Q}(P, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + p^\top(\boldsymbol{F} - \boldsymbol{F}_\star) \geq 0$
  
  C3. $R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq \mathcal{Q}(T, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + t^\top(\boldsymbol{F} - \boldsymbol{F}_\star) \geq 0$

  where $V, R$ quadratic and $(P, p, T, t, \rho)$ decides convergence in:
  - distance to solution
  - function value suboptimality (if one function) or
  - primal-dual gap (if more than one function)

  depending on $(P, p, T, t)$ linearly $(\rho < 1)$ sublinearly $(\rho = 1)$
- User specifies $(P, p, T, t, \rho)$ to decide on convergence property
- User provides algorithm on $(A, B, C, D)$ form

# Main result

Given:

- a first-order method on state-space representation form
- convergence deciding data $(P, p, T, t)$ and $\rho$

We provide:

- necessary and sufficient condition for existence of $(P, p, T, t, \rho)$-*quadratic Lyapunov inequality* via feasibility of SDP
- a quadratic Lyapunov inequality if one exists

# Outline

- Introduction to first-order methods
- Examples of first-order methods
- Setting and usage preview
- Algorithm representation
- Lyapunov analysis and main result
- **Algorithm examples**

# Using the methodology

We apply our methodology in two different ways:

B1. Find the smallest possible $\rho \in [0, 1[$ via bisection search

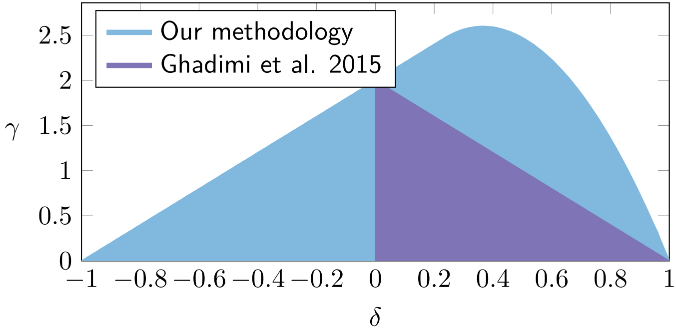
B2. Fix $\rho = 1$ and find range of algorithm parameters for which there exists a $(P, p, T, t, \rho)$-Lyapunov inequality on pre-specified grid

# Gradient method with heavy-ball momentum

- Algorithm

$$x_{k+1} = x_k - \gamma \nabla f_1(x_k) + \delta(x_k - x_{k-1})$$

- Function suboptimality convergence region for $f_1 \in \mathcal{F}_{0,1}$




- Larger parameter region with function suboptimality convergence

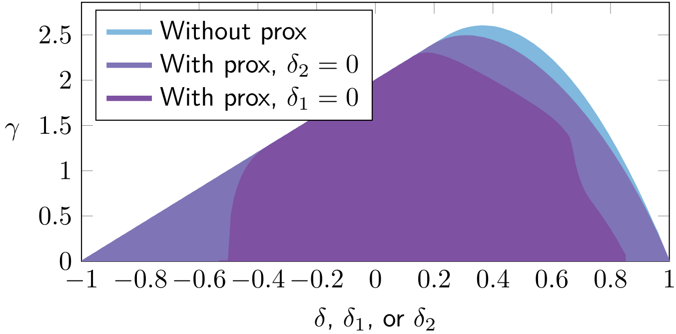
# Proximal gradient method with heavy-ball momentum

- Algorithm

$$x_{k+1} = \text{prox}_{\gamma f_2}(x_k - \gamma \nabla f_1(x_k) + \delta_1(x_k - x_{k-1})) + \delta_2(x_k - x_{k-1})$$

reduces to grad heavy-ball method if $\delta_1 = 0$ or $\delta_2 = 0$

- Duality gap convergence region $f_1 \in \mathcal{F}_{0,1}$ and $f_2 \in \mathcal{F}_{0,\infty}$



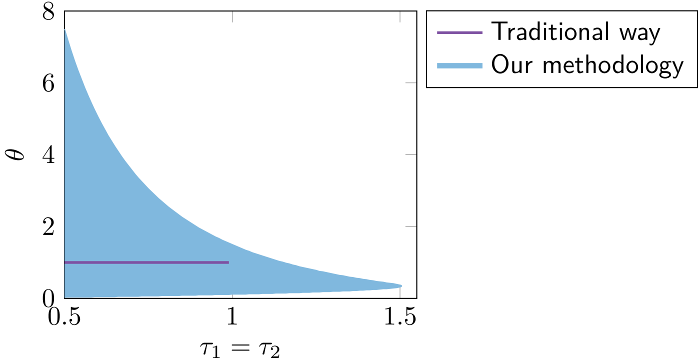

- Convergent parameter region smaller with prox
- Larger region if momentum inside prox

# Chambolle–Pock

- Chambolle–Pock ("with $L = \mathrm{Id}$"): $\underset{x\in\mathcal{H}}{\text{minimize}}(f_1(x) + f_2(x))$

$$x_{k+1} = \mathrm{prox}_{\tau_1 f_1}(x_k - \tau y_k)$$
$$y_{k+1} = \mathrm{prox}_{\tau_2 f_2^*}(y_k + \tau_2 (x_{k+1} + \theta(x_{k+1} - x_k)))$$

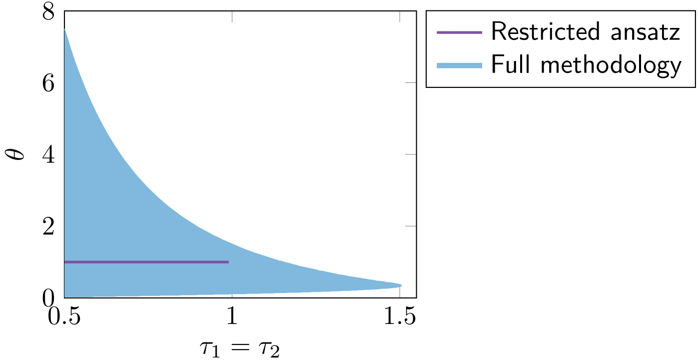
- Convergent parameter choices (primal-dual gap, $f_1$ and $f_2$ pcc)




(Caveat: verified on a $0.01 \times 0.01$ grid of region)

# Chambolle–Pock—Restricted Lyapunov

- Restrict Lyapunov search space to less general (common) ansats
- Convergent parameter choices (primal-dual gap, $f_1$ and $f_2$ pcc)




- Restriction in Lyapunov ansatz gives traditional parameter region

# Summary and future work

**Summary**

- Considered control inspired algorithm framework
- Provided iff conditions for framework to be useful in optimization
- Provided iff conditions for algorithm to admit Lyapunov analysis
- Showed larger convergent parameter ranges for two algorithms

**Future work**

- Handle iteration dependent parameters
- Handle several function evaluations per iteration
- Results are numerical, method for obtaining analytical results
- Not only analysis, but also design of algorithms

# Thank you

arXiv:2302.06713

Related: The Chambolle–Pock method (with general L) converges weakly with $\theta > 1/2$ and $\tau\sigma\|L\|^2 < 4(1 + 2\theta)$

arXiv:2309.03998

# Lyapunov analysis

- Let $\boldsymbol{\xi}_k = (\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{y}_k, \boldsymbol{F}_k)$ and $\boldsymbol{\xi}_\star = (\boldsymbol{x}_\star, \boldsymbol{u}_\star, \boldsymbol{y}_\star, \boldsymbol{F}_\star)$
- Many first-order methods analyzed using *Lyapunov inequalities*

$$V(\boldsymbol{\xi}_{k+1}, \boldsymbol{\xi}_\star) \leq \rho V(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) - R(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star)$$

where $\rho \in [0, 1]$ and

- $V : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is a *Lyapunov function*
- $R : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is a *residual function*

and $\mathcal{S} = \mathcal{H}^n \times \mathcal{H}^m \times \mathcal{H}^m \times \mathbb{R}^m$

# Lyapunov and residual function ansatz

- We consider quadratic ansatzes of the functions $V$ and $R$ given by

$$V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) = \mathcal{Q}(Q, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + q^\top (\boldsymbol{F} - \boldsymbol{F}_\star),$$
$$R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) = \mathcal{Q}(S, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + s^\top (\boldsymbol{F} - \boldsymbol{F}_\star)$$

  where $Q, S \in \mathbb{S}^{n+2m}$, $q, s \in \mathbb{R}^m$ parameterize the functions and

$$\mathcal{Q}(Q, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) = \langle (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star), Q(\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star) \rangle$$

- These quadratic ansatzes are quite general

# Lyapunov analysis conclusions

- Purpose of Lyapunov analysis is to draw convergence conclusion
- Will not know $(Q, q, S, s)$ in advance $\Rightarrow$ lower bound $V$ and $R$
- Let $P, T \in \mathbb{S}^{n+2m}$, $p, t \in \mathbb{R}^m$ and

$$\underline{V}(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) = \mathcal{Q}(P, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + p^\top (\boldsymbol{F} - \boldsymbol{F}_\star)$$
$$\underline{R}(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) = \mathcal{Q}(T, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + t^\top (\boldsymbol{F} - \boldsymbol{F}_\star)$$

- Control conclusion by enforcing nonnegative lower bounds

$$V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq \underline{V}(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq 0$$
$$R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq \underline{R}(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq 0$$

# $(P, p, T, t, \rho)$-quadratic Lyapunov inequality

$(P, p, T, t, \rho)$-*Lyapunov inequality* for algorithm over $\boldsymbol{\mathcal{F}}_{\boldsymbol{\sigma},\boldsymbol{\beta}}$:

C1. $V(\boldsymbol{\xi}_+, \boldsymbol{\xi}_\star) \leq \rho V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) - R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star)$

C2. $V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq \mathcal{Q}(P, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + p^\top(\boldsymbol{F} - \boldsymbol{F}_\star) \geq 0$

C3. $R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \geq \mathcal{Q}(T, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + t^\top(\boldsymbol{F} - \boldsymbol{F}_\star) \geq 0$

## Convergence conclusions

- For $\rho \in [0,1[$:

$$0 \leq \underline{V}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) \leq V(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) \leq \rho^k V(\boldsymbol{\xi}_0, \boldsymbol{\xi}_\star) \to 0$$

  i.e., lower bound converges $\rho$-linearly to 0

- For $\rho = 1$, a telescoping summation gives

$$0 \leq \sum_{k=0}^{\infty} \underline{R}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) \leq \sum_{k=0}^{\infty} R(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) \leq V(\boldsymbol{\xi}_0, \boldsymbol{\xi}_\star)$$

- The choice of $P, T \in \mathbb{S}^{n+2m}$, $p, t \in \mathbb{R}^m$ decides conclusion

# Some choices of $(P, p, T, t)$

- Suppose $\rho \in [0, 1[$ and let $e_i$ be $i$th basis vector and

$$(P, p, T, t) = \left( \begin{bmatrix} C & D & -D \end{bmatrix}^\top e_i e_i^\top \begin{bmatrix} C & D & -D \end{bmatrix}, 0, 0, 0 \right)$$

  then $\underline{V}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) = \left\| y_k^{(i)} - y_\star \right\|^2 \geq 0 \Rightarrow \rho$-linear convergence

- Suppose $\rho = 1$ and $m = 1$ and let

$$(P, p, T, t) = (0, 0, 0, 1)$$

  then $\underline{R}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) = f_1(y_k^{(1)}) - f_1(y_\star) \geq 0$ which gives
  - function suboptimality convergence
  - ergodic $\mathcal{O}(1/k)$ function suboptimality convergence

# $(P, p, T, t)$ for duality gap convergence

- Suppose $\rho = 1$ and $m > 1$ and let

$$(P, p, T, t) = \left(0, 0, \begin{bmatrix} C & D & -D \\ 0 & 0 & I \end{bmatrix}^\top \begin{bmatrix} 0 & -\frac{1}{2}I \\ -\frac{1}{2}I & 0 \end{bmatrix} \begin{bmatrix} C & D & -D \\ 0 & 0 & I \end{bmatrix}, \mathbf{1}\right)$$

  then

$$\begin{aligned} \underline{R}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_\star) &= \sum_{i=1}^{m} \left( f_i\left(y_k^{(i)}\right) - f_i\left(y_\star^{(i)}\right) - \left\langle u_\star^{(i)}, y_k^{(i)} - y_\star^{(i)} \right\rangle \right) \\ &= \mathcal{L}(\boldsymbol{y}, \boldsymbol{u}_\star) - \mathcal{L}(\boldsymbol{y}_\star, \boldsymbol{u}) \geq 0 \end{aligned}$$

  where $\mathcal{L} : \mathcal{H}^m \times \mathcal{H}^m \to \mathbb{R}$ is a *Lagrangian function* giving
  - duality gap convergence
  - ergodic $\mathcal{O}(1/k)$ duality gap convergence
- Generalization to function value suboptimality to $m > 1$

# $(P, p, T, t, \rho)$-**quadratic Lyapunov inequality**

- $(P, p, T, t, \rho)$-*Lyapunov inequality* for algorithm over $\boldsymbol{\mathcal{F}}_{\boldsymbol{\sigma},\boldsymbol{\beta}}$:
  C1. $V(\boldsymbol{\xi}_+, \boldsymbol{\xi}_\star) \le \rho V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) - R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star)$
  C2. $V(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \ge \mathcal{Q}(P, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + p^\top(\boldsymbol{F} - \boldsymbol{F}_\star) \ge 0$
  C3. $R(\boldsymbol{\xi}, \boldsymbol{\xi}_\star) \ge \mathcal{Q}(T, (\boldsymbol{x} - \boldsymbol{x}_\star, \boldsymbol{u}, \boldsymbol{u}_\star)) + t^\top(\boldsymbol{F} - \boldsymbol{F}_\star) \ge 0$
- Conditions should hold for points reachable by algorithm:
  - each $\boldsymbol{\xi} \in \mathcal{S}$ that is *algorithm-consistent* for $\boldsymbol{f}$
  - each *successor* $\boldsymbol{\xi}_+ \in \mathcal{S}$ of $\boldsymbol{\xi}$
  - each fixed point $\boldsymbol{\xi}_\star \in \mathcal{S}$
  - each $\boldsymbol{f} = (f_1, \ldots, f_m) \in \boldsymbol{\mathcal{F}}_{\boldsymbol{\sigma},\boldsymbol{\beta}}$

  which adds complication compared to if $\boldsymbol{\xi}, \boldsymbol{\xi}_+, \boldsymbol{\xi}_\star \in \mathcal{S}^3$

# Traditional way to find Lyapunov inequality

- Use inequalities for function class that algorithm solves
- Combine with algorithm updates
- Manipulate to arrive at Lyapunov inequality

# Main result

Given:

- a first-order method on state-space representation form
- convergence deciding data $(P, p, T, t)$ and $\rho$

We provide:

- a necessary and sufficient condition for the existence of a $(P, p, T, t, \rho)$-*quadratic Lyapunov inequality*
- a quadratic Lyapunov inequality $(Q, q, S, s)$ if one exists

# Necessary and sufficient condition

There exists a Lyapunov inequality satisfying C1-C3

$$\Longleftrightarrow^{(1)}$$

A particular SDP involving $(Q, q, S, s)$ is feasible

(1) Assuming dimension independence and Slater condition

# Necessary and sufficient condition

There exists a Lyapunov inequality satisfying C1-C3

$$\Longleftrightarrow^{(1)}$$

A particular SDP involving $(Q, q, S, s)$ is feasible

$$\text{C1}\begin{cases}\lambda^{\text{C1}}_{(l,i,j)} \geq 0 \text{ for each } l \in [\![1,m]\!] \text{ and distinct } i,j \in \{\varnothing,+,\star\},\\ \Sigma_\varnothing^\top(\rho Q - S)\Sigma_\varnothing - \Sigma_+^\top Q \Sigma_+ + \sum_{l=1}^{m} \sum_{\substack{i,j\in\{\varnothing,+,\star\}\\ i\neq j}} \lambda^{\text{C1}}_{(l,i,j)} \boldsymbol{M}_{(l,i,j)} \succeq 0,\\ \begin{bmatrix}\rho q - s\\ -q\end{bmatrix} + \sum_{l=1}^{m} \sum_{\substack{i,j\in\{\varnothing,+,\star\}\\ i\neq j}} \lambda^{\text{C1}}_{(l,i,j)} \boldsymbol{a}_{(l,i,j)} = 0,\end{cases}$$

$$\text{C2}\begin{cases}\lambda^{\text{C2}}_{(l,i,j)} \geq 0 \text{ for each } l \in [\![1,m]\!] \text{ and distinct } i,j \in \{\varnothing,\star\},\\ \Sigma_\varnothing^\top(Q - P)\Sigma_\varnothing + \sum_{l=1}^{m} \sum_{\substack{i,j\in\{\varnothing,\star\}\\ i\neq j}} \lambda^{\text{C2}}_{(l,i,j)} \boldsymbol{M}_{(l,i,j)} \succeq 0,\\ \begin{bmatrix}q - p\\ 0\end{bmatrix} + \sum_{l=1}^{m} \sum_{\substack{i,j\in\{\varnothing,\star\}\\ i\neq j}} \lambda^{\text{C2}}_{(l,i,j)} \boldsymbol{a}_{(l,i,j)} = 0,\end{cases}$$

$$\text{C3}\begin{cases}\lambda^{\text{C3}}_{(l,i,j)} \geq 0 \text{ for each } l \in [\![1,m]\!] \text{ and distinct } i,j \in \{\varnothing,\star\},\\ \Sigma_\varnothing^\top(S - T)\Sigma_\varnothing + \sum_{l=1}^{m} \sum_{\substack{i,j\in\{\varnothing,\star\}\\ i\neq j}} \lambda^{\text{C3}}_{(l,i,j)} \boldsymbol{M}_{(l,i,j)} \succeq 0,\\ \begin{bmatrix}s - t\\ 0\end{bmatrix} + \sum_{l=1}^{m} \sum_{\substack{i,j\in\{\varnothing,\star\}\\ i\neq j}} \lambda^{\text{C3}}_{(l,i,j)} \boldsymbol{a}_{(l,i,j)} = 0,\end{cases}$$

(1) Assuming dimension independence and Slater condition

# How to arrive at condition?

- C1-C3 equivalent to that optimal value of

$$
\begin{array}{ll}
\text{maximize} & \Phi(\boldsymbol{\xi}, \boldsymbol{\xi}_+, \boldsymbol{\xi}_\star) \\
\text{subject to} & \boldsymbol{x}_+ = (A \otimes \mathrm{Id})\boldsymbol{x} + (B \otimes \mathrm{Id})\boldsymbol{u}, \\
& \boldsymbol{y} = (C \otimes \mathrm{Id})\boldsymbol{x} + (D \otimes \mathrm{Id})\boldsymbol{u}, \\
& \boldsymbol{u} \in \boldsymbol{\partial f}(\boldsymbol{y}), \\
& \boldsymbol{F} = \boldsymbol{f}(\boldsymbol{y}), \\
& \boldsymbol{y}_+ = (C \otimes \mathrm{Id})\boldsymbol{x}_+ + (D \otimes \mathrm{Id})\boldsymbol{u}_+, \\
& \boldsymbol{u}_+ \in \boldsymbol{\partial f}(\boldsymbol{y}_+), \\
& \boldsymbol{F}_+ = \boldsymbol{f}(\boldsymbol{y}_+), \\
& \boldsymbol{x}_\star = (A \otimes \mathrm{Id})\boldsymbol{x}_\star + (B \otimes \mathrm{Id})\boldsymbol{u}_\star, \\
& \boldsymbol{y}_\star = (C \otimes \mathrm{Id})\boldsymbol{x}_\star + (D \otimes \mathrm{Id})\boldsymbol{u}_\star, \\
& \boldsymbol{u}_\star \in \boldsymbol{\partial f}(\boldsymbol{y}_\star), \\
& \boldsymbol{F}_\star = \boldsymbol{f}(\boldsymbol{y}_\star), \\
& \boldsymbol{f} \in \boldsymbol{\mathcal{F}}_{\boldsymbol{\sigma},\boldsymbol{\beta}},
\end{array}
\tag{PEP}
$$

  is non-positive with different quadratic $\Phi$ for C1-C3
- Solved using PEP ideas