

分类号 \_\_\_\_\_

密级 \_\_\_\_\_

U D C \_\_\_\_\_

编号 \_\_\_\_\_

# 北 京 大 学

## 博士后研究工作报告

Efficient primal-dual algorithms  
for optimal transport-like problems

罗浩

工作完成日期 2020 年 9 月—2022 年 7 月

报告提交日期 2022 年 7 月

北 京 大 学 (北京)

2022 年 7 月

# 北京大学博士后研究报告原创性声明和使用授权说明

## 原创性声明

本人郑重声明：所呈交的博士后研究报告，是本人在合作导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

报告作者签名： 日期： 年 月 日

## 博士后研究报告使用授权说明

(必须装订在提交学校图书馆的印刷本中)

本人完全了解北京大学关于收集、保存、使用博士后研究报告的规定，即：

- 按照学校要求提交博士后研究报告的印刷本和电子版本；
- 学校有权保存博士后研究报告的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存博士后研究报告；
- 因特殊原因需要延迟发布博士后研究报告电子版，授权学校一年/两年/三年以后，在校园网上全文发布。

(涉密报告在解密后遵循此规定。)

报告作者签名： 合作导师签名：  
日期： 年 月 日

# 求解最优运输类问题的 高效原始-对偶算法

Efficient primal-dual algorithms  
for optimal transport-like problems

博士后姓名 罗浩

流动站(一级学科)名称 北京大学 数学

专业(二级学科)名称 计算数学

研究工作起始时间 2020 年 09 月 10 日

研究工作期满时间 2022 年 07 月 06 日

北京大学 数学科学学院

2022 年 07 月

## 摘 要

本报告主要关注一类最优输运问题的高效求解算法。首先，我们将引入一些原始-对偶流模型并为其设计相应的、具有指数衰减性质的李亚普洛夫函数。接着，我们利用连续模型的时间离散格式来得到原始对偶算法，并证明(超)线性或次线性收敛率。此外，借助于问题本身的特殊结构，我们可以得到内问题的精确，或结合代数多重网格法与半光滑牛顿迭代进行快速求解。最后，我们会提供一些数值试验以验证所提出的算法的高效性。

关键词：最优传输；蒙日-康托洛维奇问题；线性规划；原始-对偶动力系统；李雅普诺夫函数；时间离散；半光滑牛顿；代数多重网格

# Abstract

In this report, we focus on efficient algorithms for solving a large class of optimal transport-like problems. We will first introduce some primal-dual flow models and equip them with proper Lyapunov functions that possess exponential decay property. Then, based on time discretizations of the continuous dynamics, we obtain new primal-dual methods and prove the non-ergodic ((super-)linear or sublinear)convergence rates. Besides, by exploring the special structure, the inner problems of the proposed methods either admit closed solution or can be solved by the semi-smooth Newton iteration with algebraic multigrid method. Moreover, numerical experiments are provided to validate the efficiency of our algorithms.

**Keywords:** Optimal transport; Monge–Kantorovich problem; Linear programming; Primal-dual dynamics; Lyapunov function; Time discretization; Semi-smooth Newton; Algebraic multigrid

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Optimal Transport . . . . .	1
1.1.1	Monge's problem . . . . .	1
1.1.2	Kantorovich relaxation . . . . .	2
1.1.3	Transport-like Problems . . . . .	3
1.1.4	Generalized transportation problem . . . . .	5
1.2	Existing Methods . . . . .	6
1.2.1	Entropy regularization . . . . .	6
1.2.2	Classical LP solvers . . . . .	7
1.3	Notations . . . . .	8
<b>2</b>	<b>Accelerated Primal-Dual Flow Dynamics</b>	<b>10</b>
2.1	A Novel Primal-Dual Model . . . . .	10
2.1.1	The classical saddle-point dynamics . . . . .	11
2.1.2	A discrete perspective . . . . .	13
2.1.3	The primal-dual flow . . . . .	15
2.2	Two Extended Flow Models . . . . .	19
2.2.1	Accelerated primal-dual flow . . . . .	20
2.2.2	The two-block case . . . . .	22
<b>3</b>	<b>Time Discretizations</b>	<b>25</b>
3.1	An Implicit Scheme . . . . .	25
3.2	An Implicit-Explicit Scheme . . . . .	29
3.2.1	A single-step analysis . . . . .	30
3.2.2	Non-ergodic convergence rate . . . . .	33

<b>4 Two Alternating Direction Methods of Multipliers</b>	<b>35</b>
4.1 Semi-proximal ADMM . . . . .	35
4.1.1 Matrix inverse formulae . . . . .	37
4.2 Convergence of the Semi-proximal ADMM . . . . .	38
4.2.1 Ergodic sublinear rate . . . . .	38
4.2.2 Global linear convergence . . . . .	39
4.3 An Accelerated Proximal ADMM . . . . .	40
4.3.1 Main algorithm . . . . .	40
4.3.2 Proof of the nonergodic rate . . . . .	42
4.4 Numerical Experiments . . . . .	45
<b>5 An Inexact SsN-AMG-based Primal-Dual Algorithm</b>	<b>50</b>
5.1 Inexact SsN-based Primal-Dual Method . . . . .	50
5.1.1 An inexact primal-dual algorithm . . . . .	51
5.1.2 Rate of convergence . . . . .	53
5.1.3 An SsN method for the subproblem (5.5) . . . . .	57
5.2 An Equivalent Graph Laplacian System . . . . .	58
5.2.1 The reduced problem . . . . .	58
5.2.2 An equivalent graph Laplacian . . . . .	59
5.2.3 A hybrid framework . . . . .	60
5.3 Classical AMG . . . . .	61
5.3.1 Multilevel $W$ -cycle . . . . .	62
5.3.2 Coarsening and interpolation . . . . .	64
5.4 Numerical Tests . . . . .	66
5.4.1 Performance of AMG . . . . .	66
5.4.2 IPD-SsN-AMG method . . . . .	69
<b>Bibliography</b>	<b>78</b>
<b>Publications</b>	<b>91</b>

# Chapter 1

## Introduction

### 1.1. Optimal Transport

The optimal mass transport proposed by Monge, can be dated back to as early as the 1780s. Later, Kantorovich [61] introduced a convex relaxation of Monge's original formulation and applied it to economics. Since then, this topic attracted more attention, and it also played an increasing role in imaging processing [57, 84, 93], machine learning [4, 34, 60] and statistics [87, 97]. Here, we give a very brief introduction to two basic formulations of optimal transport, and we refer to [3, 95, 101, 102] for comprehensive theoretical investigations.

#### 1.1.1. Monge's problem

Given a source probability measure  $\mu$  on  $X$  and a target measure  $\nu$  on  $Y$ , where  $X, Y \subset \mathbb{R}^n$  are the underlying spaces. Monge's problem aims to find a transport map  $T : X \rightarrow Y$  (cf. Fig. 1.1), that pushes  $\mu$  forward to  $\nu$  and minimizes the total cost

$$\int_X c(x, T(x)) \, d\mu(x),$$

where  $c : X \times Y \rightarrow \mathbb{R}_+$  is called the cost function. Here, we say  $T$  pushes  $\mu$  forward to  $\nu$  if  $T\#\mu = \nu$ , namely  $\nu(B) = \mu(T^{-1}(B))$  for all measurable subset  $B \subset Y$ .

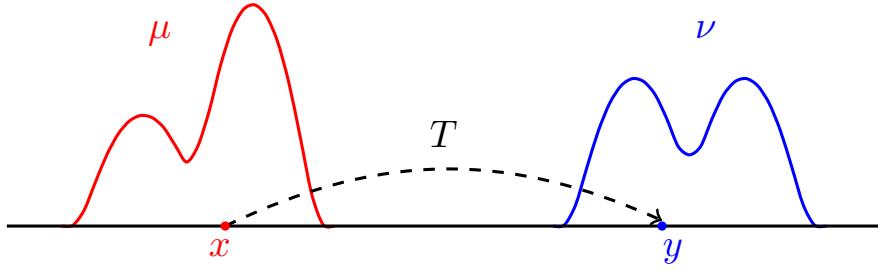


Figure 1.1: The illustration of optimal mass transport.

### 1.1.2. Kantorovich relaxation

Instead of the transport map  $T$ , which might not even exist, Kantorovich suggests considering a set of probability measures over the product space  $X \times Y$ , with the marginal distributions on  $X$  and  $Y$  being  $\mu$  and  $\nu$ , respectively. This is called the set of transport plans between  $\mu$  and  $\nu$ :

$$\Pi(\mu, \nu) := \{\pi \in \mathcal{P}(X \times Y) : \sigma_X \# \pi = \mu, \sigma_Y \# \pi = \nu\},$$

where  $\mathcal{P}(X \times Y)$  denotes the probability space over  $X \times Y$ , and  $\sigma_X : X \times Y \rightarrow X$  and  $\sigma_Y : X \times Y \rightarrow Y$  are projections:

$$\sigma_X(x, y) = x, \quad \sigma_Y(x, y) = y \quad \forall (x, y) \in X \times Y.$$

This leads to the Monge–Kantorovich problem: find  $\pi \in \Pi(\mu, \nu)$  that minimizes the total cost

$$\int \int_{X \times Y} c(x, y) \, d\pi(x, y). \quad (1.1)$$

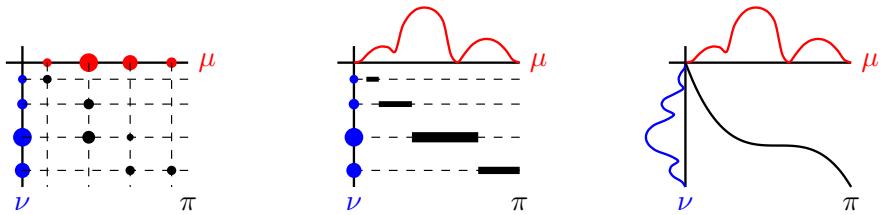


Figure 1.2: Examples of transport plan for discrete case (left), semi-discrete case (medium) and continuous case (right).

### 1.1.3. Transport-like Problems

In addition to the two standard problems in the previous section, we list several typical transport-like programmings arising from either mathematical extensions or practical applications. Those problems share the common feature of marginal constraint and we will treat them in a unified way, by introducing a generalized formulation.

#### Discrete optimal transport

In the setting of discrete optimal mass transportation [17, 61], problem (1.1) becomes a standard linear programming (LP):

$$\min_{X \in \mathcal{B}(\mu, \nu)} \langle C, X \rangle := \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij}, \quad (1.2)$$

where  $C \in \mathbb{R}_+^{m \times n}$  is the cost matrix,  $\mu \in \mathbb{R}_+^n$  and  $\nu \in \mathbb{R}_+^m$  satisfy the mass conservation:  $\mathbf{1}_n^\top \mu = \mathbf{1}_m^\top \nu$ , and

$$\mathcal{B}(\mu, \nu) := \{X \in \mathbb{R}_+^{m \times n} : X^\top \mathbf{1}_m = \mu, X \mathbf{1}_n = \nu\}$$

denotes the *transportation polytope*, with  $\mathbf{1}_n(\mathbf{1}_m) \in \mathbb{R}^n(\mathbb{R}^m)$  being the vector of all ones. According to [19, Chapter 8],  $\mathcal{B}(\mu, \nu)$  is nonempty, convex and bounded. It follows immediately from [11, Corollary 2.3] that (1.2) admits at least one solution.

When  $m = n$  and  $\mu = \nu = \mathbf{1}_n$ , the transportation polytope coincides with the Birkhoff polytope  $\mathcal{B}_n := \{X \in \mathbb{R}_+^{n \times n} : X^\top \mathbf{1}_n = \mathbf{1}_n, X \mathbf{1}_n = \mathbf{1}_n\}$ , which consists of all  $n$ -by- $n$  doubly stochastic matrices. The celebrated Birkhoff–Von-Neumann theorem (cf. [85, Theorem 17]) states that  $\mathcal{B}_n$  is the convex hull of the set of all permutation matrices. Therefore, it has frequently been used to relax some combinatorial or nonconvex problems [45]. In particular, the optimal transport (1.2) is exactly the convex relaxation of the linear assignment problem [20].

### Birkhoff projection

Given any  $\Phi \in \mathbb{R}_+^{n \times n}$ , the Birkhoff projection in terms of the Frobenius norm  $\|\cdot\|_F$  considered in [62, 73] reads as

$$\min_{X \in \mathcal{B}_n} \frac{1}{2} \|X - \Phi\|_F^2, \quad (1.3)$$

which actually seeks the nearest doubly matrix of  $\Phi$  and usually arises from the relaxations of some nonconvex programings [45, 58, 75]. In numerical simulation for circuit networks [6], we have to fix some components:

$$X_{ij} = \Phi_{ij}, \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, \quad (1.4)$$

where  $\mathcal{I}, \mathcal{J} \subset \{1, 2, \dots, n\}$  are two given index sets, and this leads to the problem of finding the best approximation in the Birkhoff polytope  $\mathcal{B}_n$  with prescribed entry constraint [5, 48].

### Capacity constrained optimal transport

The optimal transportation with capacity constraints explored recently by Korman and McCann [63], imposes additional restriction  $X \leq \Gamma$  to the classical optimal transport (1.2), i.e.,

$$\min_{X \in \mathcal{B}_\Gamma(\mu, \nu)} \langle C, X \rangle, \quad (1.5)$$

where  $\mathcal{B}_\Gamma(\mu, \nu) := \{X \in \mathcal{B}(\mu, \nu) : X \leq \Gamma\}$  and  $\Gamma \in \mathbb{R}_+^{m \times n}$  describes the capacity limitation. In other words,  $\Gamma_{ij}$  denotes the maximal mass that can be transferred from  $i$  to  $j$ . Assume  $\mathcal{B}_\Gamma(\mu, \nu)$  is nonempty, then it is a convex subset of the transportation polytope  $\mathcal{B}(\mu, \nu)$  and invoking [11, Corollary 2.3] again implies the existence of optimal solution. Capacity constrained optimal transport possesses interesting saturation phenomenon (bang-bang characterization) and symmetry property; see [64]. For theoretical study of the dual problem of (1.5), we refer to [65, 66].

### Partial optimal transport

In standard optimal transport (1.2), the marginal distributions  $\mu$  and  $\nu$  are required to have the same mass. Mathematically, this is quite restrictive and practically, the *unbalanced* case  $\mathbf{1}_n^\top \mu \neq \mathbf{1}_m^\top \nu$  stems from the

positive-unlabeled learning [23] and the representation of dynamic meshes for controlling the volume of objects with free boundaries [69].

This leads to a problem called partial optimal transport [9, 30]. More precisely, we aim to transport only a given fraction of mass  $a \in (0, a_{\max}]$  where  $a_{\max} := \min\{\mathbf{1}_n^\top \mu, \mathbf{1}_m^\top \nu\}$ , and minimize the total cost

$$\min_{X \in \mathbb{R}_+^{m \times n}} \langle C, X \rangle \quad \text{s.t. } X^\top \mathbf{1}_m \leq \mu, \quad X \mathbf{1}_n \leq \nu, \quad \mathbf{1}_m^\top X \mathbf{1}_n = a. \quad (1.6)$$

Clearly, when  $a = \mu^\top \mathbf{1}_n = \nu^\top \mathbf{1}_m$ , this amounts to the classical optimal transport (1.2). Well-posedness of (1.6) (in the continuous setting) was established in [21] and extended by Figalli [44].

#### 1.1.4. Generalized transportation problem

Clearly, the transport plan  $X$  belongs to a box region  $\mathcal{K} := \{X \in \mathbb{R}^{m \times n} : \Theta \leq X \leq \Gamma\}$ , where  $\Theta, \Gamma \in \mathbb{R}^{m \times n}$  with  $0 \leq \Theta_{ij} < \infty$  and  $\Theta_{ij} \leq \Gamma_{ij} \leq \infty$ . Introduce two slack variables  $y \in \mathbb{R}^n$  and  $z \in \mathbb{R}^m$ , together with their feasible regions  $\mathcal{Y} = \mathbb{R}_+^n$  (or  $\{0\}$ ) and  $\mathcal{Z} = \mathbb{R}_+^m$  (or  $\{0\}$ ). This allows us to include the unbalanced case  $\mathbf{1}_n^\top \mu \neq \mathbf{1}_m^\top \nu$ . We also impose the total mass constraint  $\pi(X) = a$ , where  $a \in (0, \min\{\mathbf{1}_n^\top \mu, \mathbf{1}_m^\top \nu\}]$  and  $\pi : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^l$  is a linear operator.

Then the generalized transportation problem reads as follows

$$\min_{(X,y,z) \in \Omega} H(X) \quad \text{s.t. } X^\top \mathbf{1}_m + y = \mu, \quad X \mathbf{1}_n + z = \nu, \quad \pi(X) = a, \quad (1.7)$$

where  $\Omega := \mathcal{K} \times \mathcal{Y} \times \mathcal{Z}$  and  $H(X) := \sigma/2 \|X - \Phi\|_F^2 + \langle C, X \rangle$  with  $\sigma \geq 0$  and  $\Phi \in \mathbb{R}_+^{m \times n}$ . This generic formulation contains all problems mentioned previously in Section 1.1.3, and also include other transport-like problems such as capacity constrained optimal transport [63] and the machine loading problem [43].

As usual, denote by  $\text{vec}(\times)$  the vector expanded by the matrix  $\times$  by column. Let  $f(x) := \sigma/2 \|x - \phi\|^2 + c^\top x$  with  $c = \text{vec}(C)$  and  $\phi = \text{vec}(\Phi)$ , and introduce  $\mathcal{X} := \{x \in \mathbb{R}^{mn} : \theta_i \leq x_i \leq \gamma_i, 1 \leq i \leq mn\}$  with  $\theta = \text{vec}(\Theta)$  and  $\gamma = \text{vec}(\Gamma)$ . Suppose the linear operator  $\pi$  admits a matrix representation  $\Pi \in \mathbb{R}^{l \times mn}$  such that  $\pi(X) = \Pi \text{vec}(X)$  for all  $X \in \mathbb{R}^{m \times n}$ . Then, we

rearrange (1.7) as a standard affine constrained optimization problem:

$$\min_{(x,y,z) \in \Sigma} f(x) \quad \text{s.t. } Gx + I_Y y + I_Z z = b, \quad (1.8)$$

where  $\Sigma := \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$  and

$$G := \begin{pmatrix} T \\ \Pi \end{pmatrix}, \quad T := \begin{pmatrix} I_n \otimes \mathbf{1}_m^\top \\ \mathbf{1}_n^\top \otimes I_m \end{pmatrix}, \quad I_Y := \begin{pmatrix} I_n \\ O \\ O \end{pmatrix}, \quad I_Z := \begin{pmatrix} O \\ I_m \\ O \end{pmatrix}, \quad b := \begin{pmatrix} \mu \\ \nu \\ a \end{pmatrix}. \quad (1.9)$$

## 1.2. Existing Methods

### 1.2.1. Entropy regularization

Entropy regularization, i.e., the logarithmic barrier function, can be used to relax the nonnegativity restriction and provide an approximate optimization problem that possesses some nice properties including strong convexity of the primal form (which corresponds to the smoothness of the dual problem) and closed projection onto the marginal constraint.

The well-known Sinkhorn algorithm [35, 96] and its greedy adaptation, called Greenhorn [2], are fixed-point iterations. Sinkhorn's algorithm was proved to converge linearly (cf. [85, Theorem 35]) but the rate is exponentially degenerate with respect to the regularization parameter, and the theoretical complexity bound  $\mathcal{O}(n^2/\epsilon^2)$  can be found in [38, 77]. Here, we mention that the iterative Bregman projection with KL divergence [9] is equivalent to Sinkhorn's algorithm. By virtue of the smoothness of the dual problem, accelerated mirror descent method has been proposed in [38, 77] and the improved complexity is  $\mathcal{O}(n^{2.5}\sqrt{\ln n}/\epsilon)$ . For more methods using entropy regularization, we refer to [29, 46, 47, 50, 51, 103], and one can also consult [8, 85, 89] on quite complete surveys about existing numerical methods.

It is worth noticing that, the solution to the entropy regularized problem exists uniquely, and it converges (exponentially) to an optimal transport plan with maximal entropy among all the optimal plans, as the regularization parameter vanishes; see [33] and [89, Proposition 4.1]. However, practically, one cannot choose arbitrarily small parameters due to the round-off issue and

instability effect. That being said, the log-domain technique [30] helps enhance the numerical stability, and both vectorization and parallelization can be applied to Sinkhorn’s algorithm. This makes it very popular in real applications, especially for computing Wasserstein distances between histograms.

The complexity of matrix-vector multiplication in each iteration of the Sinkhorn algorithm is  $\mathcal{O}(n^2)$  for general cases. If the transport cost function enjoys a separable structure with  $d$  blocks, then efficient implementation achieves the reduced cost  $\mathcal{O}(n^{1+1/d})$ ; see [89, Section 4.3]. More recently, for translation invariance cost functions (for instance, the Euclidean distance) that imply the cyclic property, Liao et al. [74] proposed an optimal  $\mathcal{O}(n)$  algorithm. Hence, to provide an *approximate* optimal transport plan with moderate regularization parameter, entropy-based methods are efficient, especially for some cost functions with nice properties.

### 1.2.2. Classical LP solvers

On the other hand, augmented Lagrangian method (ALM) and alternating direction method of multipliers (ADMM) can be applied to transport-like problems as well. By the celebrate stability result of linear inequality system [83, 92], we have global linear convergence for ADMM [39]. For general convex objectives, the provable nonergodic rate of many accelerated variants of (linearized) ALM and ADMM is  $\mathcal{O}(1/k)$ ; see [71, 80, 81, 99, 111]. The method [111, Algorithm 1] possesses a faster sublinear rate  $\mathcal{O}(1/k^2)$  but involves a subproblem that is a large scale quadratic programming (of dimension  $n^2$ ) in terms of the primal variable.

There are also some second-order LP solvers such as the interior-point method [68, 88] and semismooth Newton-based algorithms [5, 14, 72, 78]. These methods have to solve a symmetric positive definite (SPD) system per iteration, and prevailing linear solvers are (sparse) Cholesky decomposition and preconditioned conjugate gradient (PCG). However, we mention that the corresponding SPD system might be nearly singular and ill-conditioned as the problem size increases, and thus the number of iterations grows dramatically. Therefore, efficient and robust linear solver plays an important role. We also

refer the readers to [1, 10, 89] for some combinatorial methods.

### 1.3. Notations

At the end of this part, let us make some conventions. The angle bracket  $\langle \cdot, \cdot \rangle$  stands for the usual Euclidean inner product of two vectors. For any SPD matrix  $A \in \mathbb{R}^{n \times n}$ ,  $\langle \cdot, \cdot \rangle_A := \langle A \cdot, \cdot \rangle$  means the  $A$ -inner product and  $\|\cdot\|_A = \sqrt{\langle \cdot, \cdot \rangle_A}$  is the induced  $A$ -norm. When no confusion arises, the  $A$ -norm of a matrix  $B \in \mathbb{R}^{n \times n}$  is defined by

$$\|B\|_A := \sup_{x \in \mathbb{R}^n \setminus \{0\}} \frac{\|Bx\|_A}{\|x\|_A}.$$

Let  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  be an extended real-valued function. We say  $f$  is proper, closed and convex, if it satisfies the following:

- *proper*: its domain  $\mathbf{dom} f := \{x \in \mathbb{R}^n : f(x) < \infty\}$  is nonempty;
- *closed* (or *lower semi-continuous*): the level set  $\{x \in \mathbb{R}^n : f(x) \leq t\}$  is closed for any  $t \in \mathbb{R}$ ;
- *convex*: for any  $\theta \in [0, 1]$ ,

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad \forall x, y \in \mathbf{dom} f.$$

The *subdifferential* of  $f$  at any  $x \in \mathbf{dom} f$  is defined by

$$\partial f(x) := \{\xi \in \mathbb{R}^n : f(y) \geq f(x) + \langle \xi, y - x \rangle \quad \forall y \in \mathbf{dom} f\}.$$

Given  $\eta > 0$ , define the *proximal mapping*  $\mathbf{prox}_{\eta f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  of  $f$  by that

$$\mathbf{prox}_{\eta f}(x) := \operatorname{argmin}_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{1}{2\eta} \|y - x\|^2 \right\} \quad \forall x \in \mathbb{R}^n.$$

It is called  $\mu$ -convex with parameter  $\mu \geq 0$ , if

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) - \frac{\mu}{2}\theta(1 - \theta)\|x - y\|^2,$$

for all  $x, y \in \mathbf{dom} f$  and  $\theta \in [0, 1]$ . This is also equivalent to

$$f(y) \geq f(x) + \langle \xi, y - x \rangle + \frac{\mu}{2} \|y - x\|^2 \quad \forall \xi \in \partial f(x), \quad (1.10)$$

for all  $x, y \in \mathbf{dom} f$ . We say  $f$  is  $L$ -smooth if  $f \in C^1(\mathbb{R}^n)$  and  $\nabla f$  is  $L$ -Lipschitz continuous with  $0 < L < \infty$ :

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathbb{R}^n.$$

The set of all  $L$ -smooth functions is denoted by  $C_L^1(\mathbb{R}^n)$ , and we say  $f \in \mathcal{S}_{\mu,L}^{1,1}(\mathbb{R}^n)$  if  $f$  is  $L$ -smooth and  $\mu$ -convex, with  $0 \leq \mu \leq L < \infty$ . In addition, for any  $f \in \mathcal{S}_{\mu,L}^{1,1}(\mathbb{R}^n)$ , we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2 \quad \forall x, y \in \mathbb{R}^n. \quad (1.11)$$

Let  $O \subset \mathbb{R}^n$  be a nonempty closed convex subset. Denote by  $\delta_O : \mathbb{R}^n \rightarrow \{0, +\infty\}$  the indicator function of  $O$ , namely

$$\delta_O(x) = \begin{cases} 0 & \text{if } x \in O, \\ +\infty & \text{else.} \end{cases}$$

The subdifferential of  $\delta_O$  at any  $x \in O$  is also written by  $\mathcal{N}_O(x)$ . Besides, for simplicity, we set  $\mathbf{proj}_O(x) = \mathbf{prox}_{\eta\delta_O}(x)$  for all  $\eta > 0$  and  $x \in \mathbb{R}^n$ .

# Chapter 2

## Accelerated Primal-Dual Flow Dynamics

In this chapter, we focus on the dynamical system approach for linearly constrained convex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t. } Ax = b, \quad (2.1)$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$  and  $f$  is a (smooth) convex function.

In Section 2.1, we present a novel primal-dual flow model, which is a modification of the standard saddle-point dynamics and admits a suitable Lyapunov function. Then in Section 2.2, we consider two extensions: one is a hybridization of the primal-dual flow and the Nesterov accelerated gradient flow [82]; the other is designed for the two-block case (2.28).

### 2.1. A Novel Primal-Dual Model

The constrained minimization (2.1) can be reformulated as a minimax problem

$$\min_{x \in \mathbb{R}^n} \max_{\lambda \in \mathbb{R}^m} \{ \mathcal{L}(x, \lambda) := f(x) + \langle \lambda, Ax - b \rangle \}, \quad (2.2)$$

where  $\mathcal{L}(\cdot, \cdot)$  is the Lagrangian function. The pair  $(x^*, \lambda^*)$  is called a saddle point, i.e., the solution to (2.2), if it satisfies

$$\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda) \quad \forall (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m.$$

Clearly,  $x^*$  is a solution to (2.1), and  $\lambda^*$  is the corresponding Lagrange multiplier.

Introduce

$$X = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad \text{and} \quad \mathcal{M}(X) = \begin{pmatrix} \nabla_x \mathcal{L}(x, \lambda) \\ -\nabla_\lambda \mathcal{L}(x, \lambda) \end{pmatrix} = \begin{pmatrix} \nabla f(x) + A^\top \lambda \\ b - Ax \end{pmatrix}.$$

Note that  $\mathcal{M}$  is a maximally monotone operator:

$$\langle \mathcal{M}(X) - \mathcal{M}(Y), X - Y \rangle \geq 0 \quad \forall X, Y \in \mathbb{R}^n \times \mathbb{R}^m, \quad (2.3)$$

and  $\mathcal{M}(X^*) = 0$  for any saddle point  $X^* = (x^*, \lambda^*)$ .

### 2.1.1. The classical saddle-point dynamics

The classical saddle-point dynamics consists of two coupled flow maps:

$$x(t) : \mathbb{R}_+ \rightarrow \mathbb{R}^n \quad \text{and} \quad \lambda(t) : \mathbb{R}_+ \rightarrow \mathbb{R}^m,$$

which are governed by

$$\begin{cases} x' + \nabla_x \mathcal{L}(x, \lambda) = 0, \\ \lambda' - \nabla_\lambda \mathcal{L}(x, \lambda) = 0. \end{cases} \quad (2.4a)$$

$$(2.4b)$$

Introduce  $X(t) = (x(t), \lambda(t))$  and rewrite (2.4) as follows

$$X' + \mathcal{M}(X) = 0.$$

If  $f \in C_L^1(\mathbb{R}^n)$ , then according to [37, Section 4.2], there exists a unique solution pair  $X = (x, \lambda)$  with

$$x \in C^1(\mathbb{R}_+; \mathbb{R}^n), \quad \text{and} \quad \lambda \in C^1(\mathbb{R}_+; \mathbb{R}^m).$$

Since  $\mathcal{M}$  is maximally monotone and  $\mathcal{M}(X^*) = 0$ , we have

$$\frac{1}{2} \frac{d}{dt} \|X - X^*\|^2 = \langle X', X - X^* \rangle = -\langle \mathcal{M}(X), X - X^* \rangle \stackrel{(2.3)}{\leq} 0. \quad (2.5)$$

This implies the distance between the trajectory  $X(t)$  and the saddle point  $X^*$  is decreasing (or nonincreasing). In addition, asymptotic convergence

results can be found in [28, 42]. However, in general cases, we know little about the convergence rate of the continuous trajectory.

To study the convergence behavior of the saddle-point dynamics (2.4), let us consider a simple but illustrative example

$$\mathcal{L}_\epsilon(x, \lambda) = \frac{\epsilon}{2}(x-1)^2 + \lambda(x-2), \quad x, \lambda \in \mathbb{R}, \quad (2.6)$$

with  $\epsilon > 0$ . The unique saddle point is  $X_\epsilon^* = (2, -\epsilon)$ . Solution trajectories with  $\epsilon = 1$ ,  $10^{-1}$  and  $10^{-3}$  are displayed in Figs. 2.1, 2.2 and 2.3, respectively. It can be observed that, as  $\epsilon$  decreases, the problem becomes degenerate, and the convergence is dramatically slow.

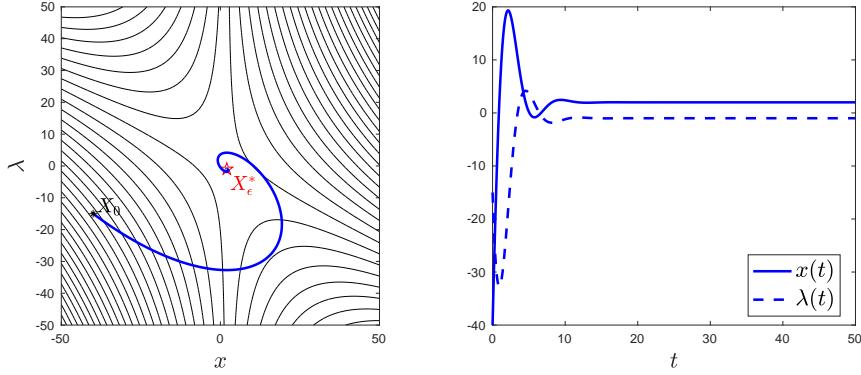


Figure 2.1: Solution trajectories of the saddle-point dynamics (2.4) with  $\epsilon = 1$ .

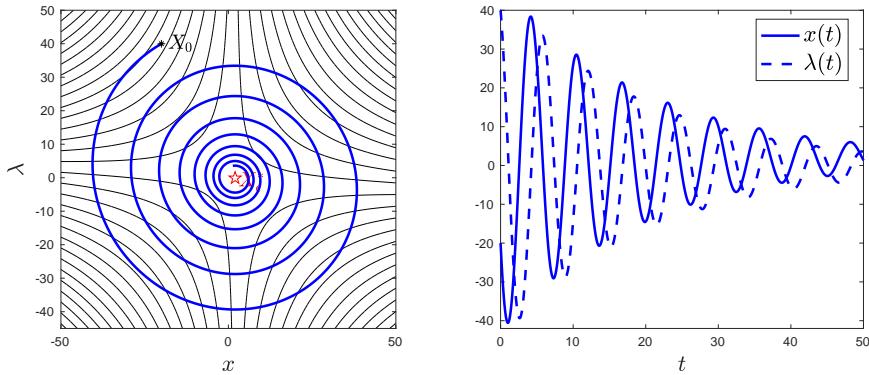


Figure 2.2: Solution trajectories of the saddle-point dynamics (2.4) with  $\epsilon = 10^{-1}$ .

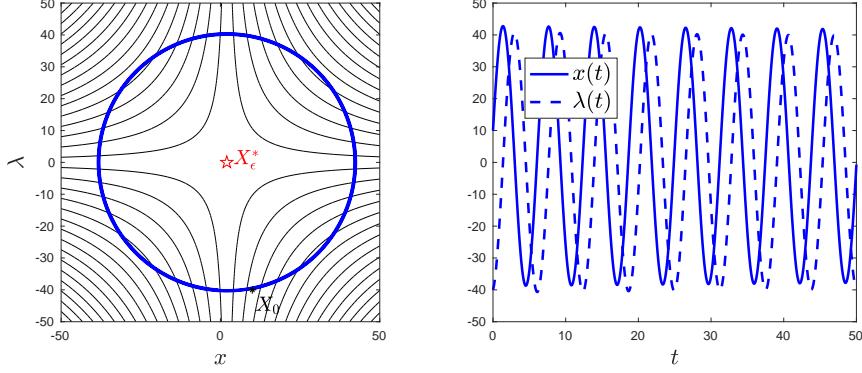


Figure 2.3: *Solution trajectories of the saddle-point dynamics (2.4) with  $\epsilon = 10^{-3}$ .*

### 2.1.2. A discrete perspective

Consider a semi-implicit time discretization of (2.4):

$$\begin{cases} \frac{x_{k+1} - x_k}{\sigma} + \nabla_x \mathcal{L}(x_{k+1}, \lambda_k) = 0, \end{cases} \quad (2.7a)$$

$$\begin{cases} \frac{\lambda_{k+1} - \lambda_k}{\tau} - \nabla_\lambda \mathcal{L}(x_{k+1}, \lambda_{k+1}) = 0, \end{cases} \quad (2.7b)$$

where  $\sigma, \tau > 0$  denote the step sizes. A reformulation leads to the Arrow–Hurwicz algorithm [112]

$$\begin{cases} x_{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \mathcal{L}(x, \lambda_k) + \frac{1}{2\sigma} \|x - x_k\|^2 \right\}, \\ \lambda_{k+1} = \underset{\lambda \in \mathbb{R}^m}{\operatorname{argmax}} \left\{ \mathcal{L}(x_{k+1}, \lambda) - \frac{1}{2\tau} \|\lambda - \lambda_k\|^2 \right\}. \end{cases} \quad (2.8)$$

Following [55] and [32, Chapter 8], we use the PPA-like interpretation to demonstrate the lack of symmetry of the semi-implicit scheme (2.7). To do this, we set

$$Q = \begin{pmatrix} I/\sigma & -A^\top \\ O & I/\tau \end{pmatrix},$$

and notice the variational inequality characterization:

$$\langle Q(X_{k+1} - X_k) + \mathcal{M}(X_{k+1}), Z - X_{k+1} \rangle \geq 0 \quad \forall Z \in \mathbb{R}^n \times \mathbb{R}^m.$$

Taking  $Z = X^*$  and utilizing the fact:  $\mathcal{M}(X^*) = 0$ , we find that

$$\begin{aligned}
 & \frac{1}{2} \|X_{k+1} - X^*\|_Q^2 - \frac{1}{2} \|X_k - X^*\|_Q^2 \\
 &= \langle Q(X_{k+1} - X_k), X_{k+1} - X^* \rangle - \frac{1}{2} \|X_{k+1} - X_k\|_Q^2 \\
 &\quad + \frac{1}{2} \langle (Q^\top - Q)(X_{k+1} - X^*), X_k - X^* \rangle \\
 &\leq \underbrace{-\langle \mathcal{M}(X_{k+1}), X_{k+1} - X^* \rangle}_{\leq 0 \text{ by (2.3)}} - \frac{1}{2} \|X_{k+1} - X_k\|_Q^2 \\
 &\quad + \frac{1}{2} \langle (Q^\top - Q)(X_{k+1} - X^*), X_k - X^* \rangle \\
 &\leq -\frac{1}{2} \|X_{k+1} - X_k\|_Q^2 + \frac{1}{2} \langle (Q^\top - Q)(X_{k+1} - X^*), X_k - X^* \rangle.
 \end{aligned} \tag{2.9}$$

As  $Q$  is not symmetric, it seems impossible to get rid of the cross term and obtain the contraction estimate that corresponds to the discrete analogue to (2.5). What's even worse, the Arrow–Hurwicz algorithm (2.8), i.e., the semi-implicit scheme (2.7), is not necessarily convergent [56].

On the other hand, let us focus on the primal-dual hybrid gradient (PDHG) method [22]

$$\begin{cases} x_{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \mathcal{L}(x, \lambda_k) + \frac{1}{2\sigma} \|x - x_k\|^2 \right\}, \\ \lambda_{k+1} = \underset{\lambda \in \mathbb{R}^m}{\operatorname{argmax}} \left\{ \mathcal{L}(2x_{k+1} - x_k, \lambda) - \frac{1}{2\tau} \|\lambda - \lambda_k\|^2 \right\}, \end{cases} \tag{2.10}$$

which is also equivalent to

$$\begin{cases} \frac{x_{k+1} - x_k}{\sigma} + \nabla_x \mathcal{L}(x_{k+1}, \lambda_k) = 0, \end{cases} \tag{2.11a}$$

$$\begin{cases} \frac{\lambda_{k+1} - \lambda_k}{\tau} - \nabla_\lambda \mathcal{L}(x_{k+1} + x_{k+1} - x_k, \lambda_{k+1}) = 0. \end{cases} \tag{2.11b}$$

Compared with the previous scheme (2.7), there is an additional extrapolation term  $x_{k+1} - x_k$  in (2.11b). Moreover, applying the above PPA-like framework to (2.10), one observes that the estimate (2.9) is improved to

$$\frac{1}{2} \|X_{k+1} - X^*\|_{\hat{Q}}^2 - \frac{1}{2} \|X_k - X^*\|_{\hat{Q}}^2 \leq -\frac{1}{2} \|X_{k+1} - X_k\|_{\hat{Q}}^2,$$

where

$$\hat{Q} = \begin{pmatrix} I/\sigma & -A^\top \\ -A & I/\tau \end{pmatrix}$$

is a symmetrization of  $Q$ . In other words, the modification in (2.11b) brings hidden symmetrization and improves the original algorithm.

### 2.1.3. The primal-dual flow

Assume  $f$  is  $\mu$ -convex with  $\mu \geq 0$  and introduce two time rescaling factors  $\gamma$  and  $\theta$  that satisfy

$$\gamma' = \mu - \gamma, \quad \theta' = -\theta, \quad (2.12)$$

with positive initial conditions  $\gamma(0) = \gamma_0 > 0$  and  $\theta(0) = \theta_0 > 0$ . It is not hard to obtain the exact expressions:

$$\gamma(t) = \mu + (\gamma_0 - \mu)e^{-t}, \quad \theta(t) = \theta_0 e^{-t}. \quad (2.13)$$

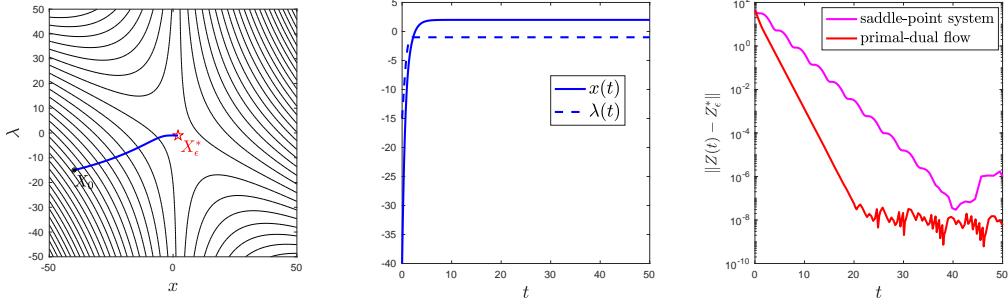


Figure 2.4: *Solution trajectories (left and medium) of the primal dynamics (2.14) with  $\epsilon = 1$ . The right one shows the convergence behaviors of the trajectories to the saddle-point dynamics (2.4) and the primal dynamics (2.14).*

Motivated by the modified discretization (2.11) that corresponds to the PDHG method (2.10), we present a novel primal-dual flow model

$$\begin{cases} \gamma x' + \nabla_x \mathcal{L}(x, \lambda) = 0, \\ \theta \lambda' - \nabla_\lambda \mathcal{L}(x + x', \lambda) = 0, \end{cases} \quad (2.14a)$$

$$(2.14b)$$

with arbitrary initial conditions

$$\lambda(0) = \lambda_0 \in \mathbb{R}^m, \quad x(0) = x_0 \in \mathbb{R}^n.$$

The modification term  $x'$  in (2.14b) is related to the extrapolation term  $x_{k+1} - x_k$  in (2.11b). Hopefully, our primal-dual flow (2.14) is more stable and robust than the saddle-point dynamics (2.4). Numerical performances on the singular problem (2.6) are provided in Figs. 2.4, 2.5 and 2.6.

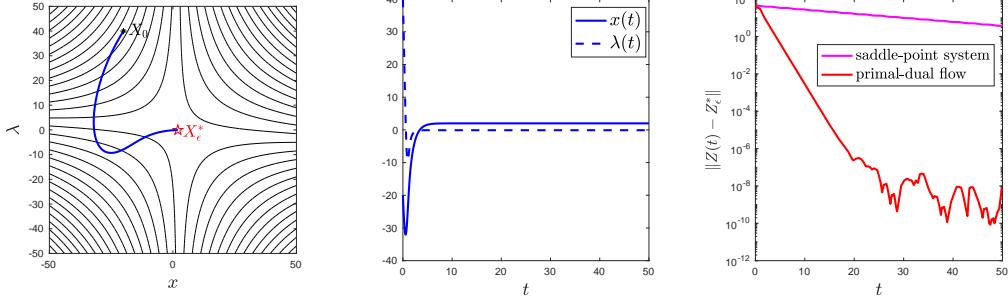


Figure 2.5: Solution trajectories (left and medium) of the primal dynamics (2.14) with  $\epsilon = 10^{-1}$ . The right one shows the convergence behaviors of the trajectories to the saddle-point dynamics (2.4) and the primal dynamics (2.14).

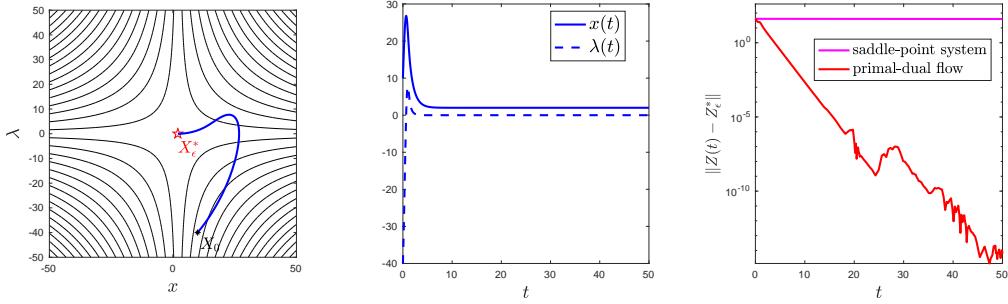


Figure 2.6: Solution trajectories (left and medium) of the primal dynamics (2.14) with  $\epsilon = 10^{-3}$ . The right one shows the convergence behaviors of the trajectories to the saddle-point dynamics (2.4) and the primal dynamics (2.14).

Define a Lyapunov function

$$\mathcal{E}(t) := \mathcal{L}(x(t), \lambda^*) - \mathcal{L}(x^*, \lambda(t)) + \frac{\gamma(t)}{2} \|x(t) - x^*\|^2 + \frac{\theta(t)}{2} \|\lambda(t) - \lambda^*\|^2. \quad (2.15)$$

We shall prove that  $\mathcal{E}(t)$  decays exponentially.

**Theorem 2.1.1.** *If  $f \in \mathcal{S}_{\mu,L}^{1,1}(\mathbb{R}^n)$ , then the primal-dual flow (2.14) admits a unique solution pair  $(x, \lambda)$  with  $x \in C^1(\mathbb{R}_+; \mathbb{R}^n)$  and  $\lambda \in C^1(\mathbb{R}_+; \mathbb{R}^m)$ . Moreover, we have*

$$\frac{d}{dt}\mathcal{E}(t) \leq -\mathcal{E}(t) - \gamma(t) \|x'(t)\|^2. \quad (2.16)$$

which gives the exponential decay

$$e^t\mathcal{E}(t) + \int_0^t \gamma(s)e^s \|x'(s)\|^2 ds \leq \mathcal{E}(0), \quad 0 \leq t < \infty. \quad (2.17)$$

**Proof.** Let us first prove the well-posedness. Define  $\mathcal{F} : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \times \mathbb{R}^m$  by that

$$\mathcal{F}(t, X) := \begin{pmatrix} -\frac{1}{\gamma(t)} \nabla_x \mathcal{L}(x, \lambda) \\ \frac{1}{\theta(t)} \nabla_\lambda \mathcal{L}(x, \lambda) \end{pmatrix} \quad \forall X = \begin{pmatrix} x \\ \lambda \end{pmatrix}.$$

Then the primal-dual flow (2.14) can be rewritten as

$$X'(t) = \mathcal{F}(t, X(t)), \quad (2.18)$$

and a direct calculation yields that, for all  $X, Y \in \mathbb{R}^n \times \mathbb{R}^m$  and  $0 \leq s \leq t$ ,

$$\|\mathcal{F}(t, X) - \mathcal{F}(s, Y)\| \leq C_0(L + \|A\|) (|t-s| \|X - X^*\| + \|X - Y\|) e^t,$$

where  $X^* = (x^*, \lambda^*)$  and the bounded positive constant  $C_0$  depends only on  $\gamma_0, \theta_0$  and  $\mu$ . This means  $\mathcal{F}$  is locally Lipschitz continuous and according to [54, Proposition 6.2.1] and [16, Corollary A.2], the ordinary differential equation (2.18) exists a unique classical solution  $X = (x, \lambda)$ , with  $x \in C^1(\mathbb{R}_+; \mathbb{R}^n)$  and  $\lambda \in C^1(\mathbb{R}_+; \mathbb{R}^m)$ .

We then prove (2.16), which yields (2.17) immediately. By (2.12), (2.14a) and (2.14b), it holds that

$$\begin{aligned} \frac{d}{dt}\mathcal{E}(t) &= \langle \nabla_\gamma \mathcal{E}, \gamma' \rangle + \langle \nabla_\theta \mathcal{E}, \theta' \rangle + \langle \nabla_x \mathcal{E}, x' \rangle + \langle \nabla_\lambda \mathcal{E}, \lambda' \rangle \\ &= \underbrace{\frac{\mu - \gamma}{2} \|x - x^*\|^2 - \frac{\theta}{2} \|\lambda - \lambda^*\|^2}_{\mathbb{I}_1} - \underbrace{\frac{1}{\gamma} \langle \nabla_x \mathcal{L}(x, \lambda), \nabla_x \mathcal{L}(x, \lambda^*) \rangle}_{\mathbb{I}_2} \\ &\quad - \underbrace{\langle \nabla_x \mathcal{L}(x, \lambda), x - x^* \rangle + \langle \nabla_\lambda \mathcal{L}(x + x', \lambda), \lambda - \lambda^* \rangle}_{\mathbb{I}_3} \\ &:= \mathbb{I}_1 + \mathbb{I}_2 + \mathbb{I}_3. \end{aligned}$$

Observing  $\nabla_x \mathcal{L}(x, \lambda^*) = \nabla_x \mathcal{L}(x, \lambda) - A^\top(\lambda - \lambda^*)$ , we have

$$\mathbb{I}_2 = -\frac{1}{\gamma} \left\langle \nabla_x \mathcal{L}(x, \lambda), \nabla_x \mathcal{L}(x, \lambda) - A^\top(\lambda - \lambda^*) \right\rangle = -\gamma \|x'\|^2 - \langle Ax', \lambda - \lambda^* \rangle.$$

Reformulate  $\mathbb{I}_3$  as follows

$$\begin{aligned} \mathbb{I}_3 &= \langle \nabla_\lambda \mathcal{L}(x + x', \lambda), \lambda - \lambda^* \rangle - \langle Ax - Ax^*, \lambda - \lambda^* \rangle - \langle \nabla_x \mathcal{L}(x, \lambda^*), x - x^* \rangle \\ &= \langle Ax', \lambda - \lambda^* \rangle - \langle \nabla_x \mathcal{L}(x, \lambda^*), x - x^* \rangle. \end{aligned}$$

Since  $f$  is  $\mu$ -convex, we claim that  $\mathcal{L}(\cdot, \lambda^*)$  is also  $\mu$ -convex. It follows from (1.11) that

$$\mathbb{I}_3 - \langle Ax', \lambda - \lambda^* \rangle + \frac{\mu}{2} \|x - x^*\|^2 \leq \mathcal{L}(x^*, \lambda^*) - \mathcal{L}(x, \lambda^*) = \mathcal{L}(x^*, \lambda) - \mathcal{L}(x, \lambda^*),$$

where we used the fact that  $\mathcal{L}(x^*, \cdot)$  is a constant. Consequently, collecting the above estimates proves (2.16) and completes the proof.  $\square$

Furthermore, we have a corollary which gives: (i) the boundness of  $\lambda(t)$  and  $x(t)$ ; (ii) the integrability of  $\|x'(t)\|$ ; (iii) the exponential decay rates of the Lagrangian  $\mathcal{L}(x(t), \lambda^*) - \mathcal{L}(x^*, \lambda(t))$ , the primal objective residual  $|f(x(t)) - f(x^*)|$  and the feasibility violation  $\|Ax(t) - b\|$ .

**Corollary 2.1.1.** *If  $f \in \mathcal{S}_{\mu, L}^{1,1}(\mathbb{R}^n)$ , then for the unique solution pair  $(x, \lambda) : \mathbb{R}_+ \rightarrow \mathbb{R}^n \times \mathbb{R}^m$  of the primal-dual flow (2.14), we have the following estimates.*

- $\sqrt{\gamma_0 + \gamma_{\min}} e^t \|x'(t)\| \in L^2(0, \infty)$ .
- $0 \leq \mathcal{L}(x(t), \lambda^*) - \mathcal{L}(x^*, \lambda(t)) \leq e^{-t} \mathcal{E}(0)$ .
- $\lambda(t)$  is bounded:  $\theta_0 \|\lambda(t) - \lambda^*\|^2 \leq 2\mathcal{E}(0)$ .
- $x(t)$  is bounded:  $\gamma_0 \|x(t) - x^*\|^2 \leq 2\mathcal{E}(0)$  and  $\gamma_{\min} \|x(t) - x^*\|^2 \leq 2e^{-t} \mathcal{E}(0)$  with  $\gamma_{\min} := \min\{\gamma_0, \mu\}$ .
- $\|Ax(t) - b\| \leq e^{-t} \mathcal{R}_0$  and  $|f(x(t)) - f(x^*)| \leq e^{-t} (\mathcal{E}(0) + \mathcal{R}_0 \|\lambda^*\|)$ , where

$$\mathcal{R}_0 := \sqrt{2\theta_0 \mathcal{E}(0)} + \theta_0 \|\lambda_0 - \lambda^*\| + \|Ax_0 - b\|.$$

**Proof.** By (2.13), we have

$$\gamma(t) \geq \max \{ \gamma_{\min}, \gamma_0 e^{-t} \} \quad \forall t \geq 0. \quad (2.19)$$

The first to the fourth follow directly from (2.15), (2.17) and (2.19).

Let us prove the last one. Define  $\xi(t) := \lambda(t) - \theta^{-1}(t)(Ax(t) - b)$ . In view of (2.12) and (2.14b), it follows that

$$\frac{d\xi}{dt} = \lambda'(t) - \theta^{-1}(t)(Ax'(t) + Ax(t) - b) = 0,$$

which says  $\xi(t) = \xi(0)$  for all  $t \geq 0$  and also implies

$$\|Ax(t) - b\| = \theta(t) \|\lambda(t) - \xi(0)\| \leq \theta(t) (\|\lambda(t) - \lambda^*\| + \|\xi(0) - \lambda^*\|).$$

Hence, from the fact  $\theta(t) = \theta_0 e^{-t}$  and the boundness of  $\|\lambda(t) - \lambda^*\|$ , we have

$$\|Ax(t) - b\| \leq e^{-t} \left( \sqrt{2\theta_0 \mathcal{E}(0)} + \theta_0 \|\xi(0) - \lambda^*\| \right) \leq e^{-t} \mathcal{R}_0. \quad (2.20)$$

Besides, it follows from (2.17) that

$$0 \leq \mathcal{L}(x(t), \lambda^*) - \mathcal{L}(x^*, \lambda(t)) = f(x(t)) - f(x^*) + \langle \lambda^*, Ax(t) - b \rangle \leq e^{-t} \mathcal{E}(0),$$

which together with the previous estimate (2.20) gives

$$|f(x(t)) - f(x^*)| \leq \|\lambda^*\| \|Ax(t) - b\| + e^{-t} \mathcal{E}(0) \leq e^{-t} (\mathcal{E}(0) + \mathcal{R}_0 \|\lambda^*\|).$$

This establishes the proof.  $\square$

## 2.2. Two Extended Flow Models

We now consider two extensions of the primal-dual flow (2.14). The first applies the Nesterov accelerated gradient (NAG) flow [82] to the primal variable but leaves the multiplier invariant, and the second is designed for the two-block case (2.28).

### 2.2.1. Accelerated primal-dual flow

A natural extension of (2.14) is

$$\begin{cases} \theta\lambda' - \nabla_\lambda \mathcal{L}(x + x', \lambda) = 0, \\ \gamma x'' + (\mu + \gamma)x' + \nabla_x \mathcal{L}(x, \lambda) = 0, \end{cases} \quad (2.21a)$$

$$\begin{cases} \theta\lambda' - \nabla_\lambda \mathcal{L}(v, \lambda) = 0, \\ \gamma v' + \mu(v - x) + \nabla_x \mathcal{L}(x, \lambda) = 0, \end{cases} \quad (2.21b)$$

where  $\gamma$  and  $\theta$  are still governed by (2.12). Introduce an auxiliary variable  $v = x + x'$  and rewrite (2.21) as

$$\begin{cases} x' + x - v = 0, \\ \theta\lambda' - \nabla_\lambda \mathcal{L}(v, \lambda) = 0, \end{cases} \quad (2.22a)$$

$$\begin{cases} \theta\lambda' - \nabla_\lambda \mathcal{L}(v, \lambda) = 0, \\ \gamma v' + \mu(v - x) + \nabla_x \mathcal{L}(x, \lambda) = 0, \end{cases} \quad (2.22b)$$

$$\begin{cases} \theta\lambda' - \nabla_\lambda \mathcal{L}(v, \lambda) = 0, \\ \gamma v' + \mu(v - x) + \nabla_x \mathcal{L}(x, \lambda) = 0, \end{cases} \quad (2.22c)$$

with arbitrary initial conditions

$$\lambda(0) = \lambda_0 \in \mathbb{R}^m, \quad x(0) = x_0 \in \mathbb{R}^n, \quad \text{and} \quad v(0) = v_0 \in \mathbb{R}^n.$$

Both (2.21) and (2.22) are called *accelerated primal-dual* (APD) flow. Similarly with before, we claim that the the APD flow system (2.22) admits a unique classical solution pair  $(\lambda, x, v)$  with

$$\lambda \in C^2(\mathbb{R}_+; \mathbb{R}^m), \quad x \in C^2(\mathbb{R}_+; \mathbb{R}^n), \quad \text{and} \quad v \in C^1(\mathbb{R}_+; \mathbb{R}^n).$$

Inspired by (2.15), let us equip the APD flow (2.22) with a suitable Lyapunov function

$$\mathcal{E}(t) := \mathcal{L}(x(t), \lambda^*) - \mathcal{L}(x^*, \lambda(t)) + \frac{\gamma(t)}{2} \|v(t) - x^*\|^2 + \frac{\theta(t)}{2} \|\lambda(t) - \lambda^*\|^2. \quad (2.23)$$

**Theorem 2.2.1.** *Assume  $f \in \mathcal{S}_{\mu, L}^{1,1}(\mathbb{R}^n)$ . Let  $(\lambda, x, v)$  be the unique solution to (2.22), then for  $\mathcal{E}(t)$  defined by (2.23), it holds that*

$$\frac{d}{dt} \mathcal{E}(t) \leq -\mathcal{E}(t) - \frac{\mu}{2} \|x'(t)\|^2, \quad (2.24)$$

which implies

$$\mathcal{E}(t) + \frac{\mu}{2} \int_0^t e^{s-t} \|x'(s)\|^2 ds \leq e^{-t} \mathcal{E}(0), \quad 0 \leq t < \infty. \quad (2.25)$$

Moreover, we have

$$\begin{cases} \|Ax(t) - b\| \leq e^{-t} \mathcal{R}_0, \\ |f(x(t)) - f(x^*)| \leq e^{-t} (\mathcal{E}(0) + \mathcal{R}_0 \|\lambda^*\|), \end{cases}$$

where  $\mathcal{R}_0 := \sqrt{2\theta_0 \mathcal{E}(0)} + \theta_0 \|\lambda_0 - \lambda^*\| + \|Ax_0 - b\|$ .

**Proof.** A direct computation gives

$$\begin{aligned} \frac{d}{dt} \mathcal{E}(t) &= \langle x', \nabla_x \mathcal{L}(x, \lambda^*) \rangle + \frac{\gamma'}{2} \|v - x^*\|^2 + \langle \gamma v', v - x^* \rangle \\ &\quad + \frac{\theta'}{2} \|\lambda - \lambda^*\|^2 + \langle \theta \lambda', \lambda - \lambda^* \rangle. \end{aligned}$$

In view of (2.12) and (2.22), we replace all temporal derivatives with their right hand sides and obtain  $\mathcal{E}'(t) = \mathbb{I}_1 + \mathbb{I}_2$ , where

$$\begin{cases} \mathbb{I}_1 := -\frac{\theta}{2} \|\lambda - \lambda^*\|^2 + \frac{\mu - \gamma}{2} \|v - x^*\|^2 + \mu \langle x - v, v - x^* \rangle, \\ \mathbb{I}_2 := \langle \nabla_x \mathcal{L}(x, \lambda^*), v - x \rangle - \langle \nabla_x \mathcal{L}(x, \lambda), v - x^* \rangle + \langle \nabla_\lambda \mathcal{L}(v, \lambda), \lambda - \lambda^* \rangle. \end{cases}$$

Recall the identity

$$\mu \langle x - v, v - x^* \rangle = \frac{\mu}{2} (\|x - x^*\|^2 - \|v - x^*\|^2 - \|v - x\|^2), \quad (2.26)$$

which is trivial but very useful. We rewrite  $\mathbb{I}_1$  as follows

$$\mathbb{I}_1 = \frac{\mu}{2} \|x - x^*\|^2 - \frac{\gamma}{2} \|v - x^*\|^2 - \frac{\theta}{2} \|\lambda - \lambda^*\|^2 - \frac{\mu}{2} \|v - x\|^2. \quad (2.27)$$

Inserting the splitting

$$\langle \nabla_x \mathcal{L}(x, \lambda^*), v - x \rangle = \langle \nabla_x \mathcal{L}(x, \lambda^*), x^* - x \rangle + \langle \nabla_x \mathcal{L}(x, \lambda^*), v - x^* \rangle$$

into  $\mathbb{I}_2$  and using  $\nabla_x \mathcal{L}(x, \lambda^*) - \nabla_x \mathcal{L}(x, \lambda) = A^\top(\lambda^* - \lambda)$ , we find

$$\begin{aligned} \mathbb{I}_2 &= \langle \nabla_x \mathcal{L}(x, \lambda^*), x^* - x \rangle + \langle A^\top(\lambda^* - \lambda), v - x^* \rangle + \langle \nabla_\lambda \mathcal{L}(v, \lambda), \lambda - \lambda^* \rangle \\ &= \langle \nabla_x \mathcal{L}(x, \lambda^*), x^* - x \rangle. \end{aligned}$$

Since  $f \in \mathcal{S}_{\mu, L}^{1,1}$ ,  $\mathcal{L}(\cdot, \lambda^*)$  is  $\mu$ -convex and thus by (1.11),

$$\mathbb{I}_2 + \frac{\mu}{2} \|x - x^*\|^2 \leq \mathcal{L}(x^*, \lambda^*) - \mathcal{L}(x, \lambda^*) = \mathcal{L}(x^*, \lambda) - \mathcal{L}(x, \lambda^*).$$

Now, in view of  $x' = v - x$ , collecting this and (2.27) implies (2.24).

From (2.24) follows (2.25), and analogously to Corollary 2.1.1, it is not hard to establish the exponential decay estimates of the feasibility  $\|Ax(t) - b\|$  and the primal objective  $|f(x(t)) - f(x^*)|$ . This completes the proof.  $\square$

### 2.2.2. The two-block case

Let us consider the two-block case

$$\min_{x \in \mathbb{R}^n, y \in \mathbb{R}^r} F(x, y) := f(x) + g(y) \quad \text{s.t. } Ax + By = b, \quad (2.28)$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times r}$ ,  $b \in \mathbb{R}^m$ ,  $f \in \mathcal{S}_{\mu_f, L_f}^{1,1}(\mathbb{R}^n)$  and  $g \in \mathcal{S}_{\mu_g, L_g}^{1,1}(\mathbb{R}^r)$  are two given functions.

It is clear that the previous APD flow (2.21) can be applied directly to the problem (2.28). To utilize the separable structure, we adopt different time rescaling factors for  $x$  and  $y$  respectively, and propose another continuous model:

$$\begin{cases} 0 = \gamma x'' + (\gamma + \mu_f)x' + \nabla_x \mathcal{L}(x, y, \lambda), \end{cases} \quad (2.29a)$$

$$\begin{cases} 0 = \theta \lambda' - \nabla_\lambda \mathcal{L}(x + x', y + y', \lambda), \end{cases} \quad (2.29b)$$

$$\begin{cases} 0 = \beta y'' + (\beta + \mu_g)y' + \nabla_y \mathcal{L}(x, y, \lambda), \end{cases} \quad (2.29c)$$

where  $\gamma, \beta$  and  $\theta$  are scaling factors and governed by

$$\gamma' = \mu_f - \gamma, \quad \beta' = \mu_g - \beta, \quad \theta' = -\theta, \quad (2.30)$$

with positive initial conditions:  $\theta(0) = \theta_0 > 0$ ,  $\gamma(0) = \gamma_0 > 0$  and  $\beta(0) = \beta_0 > 0$ . It is not hard to obtain the exact solution of (2.30):

$$\gamma(t) = \gamma_0 e^{-t} + \mu_f(1 - e^{-t}), \quad \beta(t) = \beta_0 e^{-t} + \mu_g(1 - e^{-t}), \quad \theta(t) = e^{-t}.$$

Similarly with (2.22), an alternative presentation of (2.29) reads as follows

$$\begin{cases} x' = v - x, \end{cases} \quad (2.31a)$$

$$\begin{cases} \gamma v' = \mu_f(x - v) - \nabla_x \mathcal{L}(x, y, \lambda), \end{cases} \quad (2.31b)$$

$$\begin{cases} \theta \lambda' = \nabla_\lambda \mathcal{L}(v, u, \lambda), \end{cases} \quad (2.31c)$$

$$\begin{cases} \beta u' = \mu_g(y - u) - \nabla_y \mathcal{L}(x, y, \lambda), \end{cases} \quad (2.31d)$$

$$\begin{cases} y' = u - y. \end{cases} \quad (2.31e)$$

This seems a little bit complicated but for algorithm designing and convergence analysis, it is more convenient for us to start from (2.31) and treat  $(\gamma, \beta, \theta)$  as unknowns that solve (2.30).

Let us modify the Lyapunov function (2.23) by that

$$\mathcal{E}(t) := \mathcal{L}(x, y, \lambda^*) - \mathcal{L}(x^*, y^*, \lambda) + \frac{\theta}{2} \|\lambda - \lambda^*\|^2 + \frac{\beta}{2} \|u - y^*\|^2 + \frac{\gamma}{2} \|v - x^*\|^2. \quad (2.32)$$

Proceeding as before, it is not hard to conclude the existence and uniqueness of a classical  $C^1$  solution to (2.31). Besides, we note that  $x$ ,  $y$  and  $\lambda$  are  $C^2$ .

**Theorem 2.2.2.** *If  $f \in \mathcal{S}_{\mu_f, L_f}^{1,1}(\mathbb{R}^n)$  and  $g \in \mathcal{S}_{\mu_g, L_g}^{1,1}(\mathbb{R}^r)$ , then for  $\mathcal{E}(t)$  defined by (2.32), it holds that*

$$\frac{d}{dt} \mathcal{E}(t) \leq -\mathcal{E}(t) - \frac{\mu_f}{2} \|x'(t)\|^2 - \frac{\mu_g}{2} \|y'(t)\|^2, \quad (2.33)$$

which yields the exponential decay

$$2e^t \mathcal{E}(t) + \int_0^t e^s \left( \mu_f \|x'(s)\|^2 + \mu_g \|y'(s)\|^2 \right) ds \leq 2\mathcal{E}(0), \quad (2.34)$$

for all  $0 \leq t < \infty$ .

**Proof.** As (2.34) can be obtained easily from (2.33), it is sufficient to establish the latter. Set  $\Theta := (\gamma, \beta, \theta)$  and  $Z := (x, y, v, u, \lambda)$ . Let us start from the identity

$$\frac{d}{dt} \mathcal{E}(t) = \langle \nabla_Z \mathcal{E}, Z' \rangle + \langle \nabla_\Theta \mathcal{E}, \Theta' \rangle.$$

By (2.30), it is trivial that

$$\langle \nabla_\Theta \mathcal{E}, \Theta' \rangle = -\frac{\theta}{2} \|\lambda - \lambda^*\|^2 + \frac{\mu_f - \gamma}{2} \|v - x^*\|^2 + \frac{\mu_g - \beta}{2} \|u - y^*\|^2,$$

and according to (2.22), a direct computation gives

$$\begin{aligned} & \langle \nabla_Z \mathcal{E}, Z' \rangle \\ &= \langle \lambda - \lambda^*, \nabla_\lambda \mathcal{L}(v, u, \lambda) \rangle + \langle v - x, \nabla_x \mathcal{L}(x, y, \lambda^*) \rangle + \langle u - y, \nabla_y \mathcal{L}(x, y, \lambda^*) \rangle \\ & \quad + \langle v - x^*, \mu_f(x - v) - \nabla_x \mathcal{L}(x, y, \lambda) \rangle + \langle u - y^*, \mu_g(y - u) - \nabla_y \mathcal{L}(x, y, \lambda) \rangle. \end{aligned}$$

Shifting  $\lambda$  to  $\lambda^*$  gives

$$\begin{aligned} & -\langle v - x^*, \nabla_x \mathcal{L}(x, y, \lambda) \rangle - \langle u - y^*, \nabla_y \mathcal{L}(x, y, \lambda) \rangle \\ &= -\langle v - x^*, \nabla_x \mathcal{L}(x, y, \lambda^*) \rangle - \langle u - y^*, \nabla_y \mathcal{L}(x, y, \lambda^*) \rangle - \langle \lambda - \lambda^*, Av + Bu - b \rangle, \end{aligned}$$

where we have used the optimality condition  $Ax^* + By^* = b$ . It follows that

$$\begin{aligned} \langle \nabla_Z \mathcal{E}, Z' \rangle &= \langle x^* - x, \nabla_x \mathcal{L}(x, y, \lambda^*) \rangle + \langle y^* - y, \nabla_y \mathcal{L}(x, y, \lambda^*) \rangle \\ &\quad + \mu_f \langle x - v, v - x^* \rangle + \mu_g \langle y - u, u - y^* \rangle. \end{aligned} \quad (2.35)$$

Since  $f$  is  $\mu_f$ -convex and  $g$  is  $\mu_g$ -convex, using (1.11), we obtain

$$\begin{aligned} &\langle x^* - x, \nabla_x \mathcal{L}(x, y, \lambda^*) \rangle + \langle y^* - y, \nabla_y \mathcal{L}(x, y, \lambda^*) \rangle \\ &\leq \mathcal{L}(x^*, y^*, \lambda) - \mathcal{L}(x, y, \lambda^*) - \frac{\mu_f}{2} \|x - x^*\|^2 - \frac{\mu_g}{2} \|y - y^*\|^2. \end{aligned}$$

Similarly with (2.26), we rearrange the other two cross terms in (2.35) and put everything together to get

$$\frac{d}{dt} \mathcal{E}(t) \leq -\mathcal{E}(t) - \frac{\mu_f}{2} \|x - v\|^2 - \frac{\mu_g}{2} \|y - u\|^2.$$

Observing  $x - v = x'$  and  $y - u = y'$ , we obtain (2.33) and complete the proof of this theorem.  $\square$

# Chapter 3

## Time Discretizations

The continuous models proposed in the previous chapter paves the way for designing optimization methods for solving the linearly constrained problem (2.1) (and the two-block case (2.28)).

In this chapter, an implicit scheme and an implicit-explicit scheme shall be proposed respectively in Sections 3.1 and 3.2. These lead to new efficient primal-dual methods that will be applied to solving optimal transport-like problems. The corresponding nonergodic convergence rates will be established via tailored discrete Lyapunov functions.

### 3.1. An Implicit Scheme

In this section, let us consider an implicit Euler method for the APD flow (2.21): given the initial guesses  $x_0, v_0 \in \mathbb{R}^n$  and  $\lambda_0 \in \mathbb{R}^m$ , do the iteration

$$\theta_k \frac{\lambda_{k+1} - \lambda_k}{\alpha_k} = \nabla_\lambda \mathcal{L}(v_{k+1}, \lambda_{k+1}), \quad (3.1a)$$

$$\frac{x_{k+1} - x_k}{\alpha_k} = v_{k+1} - x_{k+1}, \quad (3.1b)$$

$$\gamma_k \frac{v_{k+1} - v_k}{\alpha_k} \in \mu(x_{k+1} - v_{k+1}) - \partial_x \mathcal{L}(x_{k+1}, \lambda_{k+1}), \quad (3.1c)$$

where  $\partial_x \mathcal{L}(x, \lambda) := \partial f(x) + A^\top \lambda$  and  $\alpha_k > 0$  denotes the step size. Note that in discrete level, we are allowed to consider nonsmooth  $f$  and (3.1c) becomes a difference inclusion. The parameter system (2.12) is also discretized

implicitly:

$$\frac{\gamma_{k+1} - \gamma_k}{\alpha_k} = \mu - \gamma_{k+1}, \quad \frac{\theta_{k+1} - \theta_k}{\alpha_k} = -\theta_{k+1}. \quad (3.2)$$

Before the convergence analysis, let us have a look at the solvability. By (3.1b), we express  $v_{k+1}$  in terms of  $x_{k+1}$  and  $x_k$  and plug it into (3.1a) and (3.1c) to obtain

$$\left\{ \begin{array}{l} x_{k+1} \in \tilde{x}_k - \eta_k (\partial f(x_{k+1}) + A^\top \lambda_{k+1}), \\ \lambda_{k+1} = \lambda_k - \frac{1}{\theta_k} (Ax_k - b) + \frac{1}{\theta_{k+1}} (Ax_{k+1} - b), \end{array} \right. \quad (3.3a)$$

$$\left\{ \begin{array}{l} x_{k+1} \in \tilde{x}_k - \eta_k (\partial f(x_{k+1}) + A^\top \lambda_{k+1}), \\ \lambda_{k+1} = \lambda_k - \frac{1}{\theta_k} (Ax_k - b) + \frac{1}{\theta_{k+1}} (Ax_{k+1} - b), \end{array} \right. \quad (3.3b)$$

where  $\eta_k = \alpha_k^2 / \tau_k$  and

$$\tau_k := \gamma_k + \mu \alpha_k + \gamma_k \alpha_k, \quad \tilde{x}_k := \tau_k^{-1} ((\gamma_k + \mu \alpha_k) x_k + \gamma_k \alpha_k v_k). \quad (3.4)$$

Eliminating  $\lambda_{k+1}$  from (3.3a) to get

$$x_{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ f(x) + \frac{1}{2\theta_{k+1}} \|Ax - b\|^2 + \frac{1}{2\eta_k} \|x - \hat{x}_k\|^2 \right\}, \quad (3.5)$$

where  $\hat{x}_k := \tilde{x}_k - \eta_k A^\top (\lambda_k - \theta_k^{-1} (Ax_k - b))$ . Note that the quadratic penalty term  $\|Ax - b\|^2$  comes from  $\lambda_{k+1}$ , which is coupled with  $x_{k+1}$ . If we drop that term, then (3.5) is very close to the proximal ALM. On the other hand, by (3.3a), we have

$$x_{k+1} = \mathbf{prox}_{\eta_k f} (\tilde{x}_k - \eta_k A^\top \lambda_{k+1}),$$

and putting this into (3.3b) gives a nonlinear equation

$$\theta_{k+1} \lambda_{k+1} - A \mathbf{prox} (\tilde{x}_k - \eta_k A^\top \lambda_{k+1}) = \tilde{\lambda}_k, \quad (3.6)$$

where  $\tilde{\lambda}_k := \theta_{k+1}^{-1} (\lambda_k - \theta_k^{-1} (Ax_k - b)) - b$ .

Below let us present convergence analysis of the implicit scheme (3.1) via the discrete Lyapunov function

$$\mathcal{E}_k := \mathcal{L}(x_k, \lambda^*) - \mathcal{L}(x^*, \lambda_k) + \frac{\gamma_k}{2} \|v_k - x^*\|^2 + \frac{\theta_k}{2} \|\lambda_k - \lambda^*\|^2 \quad \forall k \in \mathbb{N}, \quad (3.7)$$

which agrees with the discrete version of (2.23).

**Theorem 3.1.1.** *Assume  $f$  is  $\mu$ -convex with  $\mu \geq 0$ . Then for the fully implicit scheme (3.1) with any step size  $\alpha_k > 0$ , we have the contraction*

$$\mathcal{E}_{k+1} - \mathcal{E}_k \leq -\alpha_k \mathcal{E}_{k+1}, \quad \text{for all } k \in \mathbb{N}. \quad (3.8)$$

Moreover, it holds that

$$\left\{ \begin{array}{l} \|Ax_k - b\| \leq \frac{\theta_k}{\theta_0} \mathcal{R}_0, \end{array} \right. \quad (3.9a)$$

$$\left\{ \begin{array}{l} 0 \leq \mathcal{L}(x_k, \lambda^*) - \mathcal{L}(x^*, \lambda_k) \leq \frac{\theta_k}{\theta_0} \mathcal{E}_0, \end{array} \right. \quad (3.9b)$$

$$\left\{ \begin{array}{l} |f(x_k) - f(x^*)| \leq \frac{\theta_k}{\theta_0} (\mathcal{E}_0 + \mathcal{R}_0 \|\lambda^*\|), \end{array} \right. \quad (3.9c)$$

where  $\mathcal{R}_0 = \sqrt{2\theta_0 \mathcal{E}_0} + \theta_0 \|\lambda_0 - \lambda^*\| + \|Ax_0 - b\|$ .

**Proof.** Mimicking the proof of the continuous setting (cf. Theorem 2.2.1), we replace the derivative with the difference  $\mathcal{E}_{k+1} - \mathcal{E}_k = \mathbb{I}_1 + \mathbb{I}_2 + \mathbb{I}_3$ , where

$$\left\{ \begin{array}{l} \mathbb{I}_1 := \mathcal{L}(x_{k+1}, \lambda^*) - \mathcal{L}(x_k, \lambda^*), \\ \mathbb{I}_2 := \frac{\theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2 - \frac{\theta_k}{2} \|\lambda_k - \lambda^*\|^2, \\ \mathbb{I}_3 := \frac{\gamma_{k+1}}{2} \|v_{k+1} - x^*\|^2 - \frac{\gamma_k}{2} \|v_k - x^*\|^2. \end{array} \right.$$

Let us set the first term  $\mathbb{I}_1$  aside and consider the estimates for  $\mathbb{I}_2$  and  $\mathbb{I}_3$ . For a start, by the equation of the sequence  $\{\theta_k\}$  in (3.2), an evident calculation yields that

$$\begin{aligned} \mathbb{I}_2 &= \frac{\theta_{k+1} - \theta_k}{2} \|\lambda_{k+1} - \lambda^*\|^2 + \frac{\theta_k}{2} (\|\lambda_{k+1} - \lambda^*\|^2 - \|\lambda_k - \lambda^*\|^2) \\ &= -\frac{\alpha_k \theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2 + \frac{\theta_k}{2} (\|\lambda_{k+1} - \lambda^*\|^2 - \|\lambda_k - \lambda^*\|^2) \\ &= -\frac{\alpha_k \theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2 - \frac{\theta_k}{2} \|\lambda_{k+1} - \lambda_k\|^2 + \theta_k \langle \lambda_{k+1} - \lambda_k, \lambda_{k+1} - \lambda^* \rangle. \end{aligned}$$

According to (3.1a), we rewrite the last cross term to obtain

$$\mathbb{I}_2 = -\frac{\alpha_k \theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2 - \frac{\theta_k}{2} \|\lambda_{k+1} - \lambda_k\|^2 + \alpha_k \langle Av_{k+1} - b, \lambda_{k+1} - \lambda^* \rangle. \quad (3.10)$$

Similarly, by (3.2), the term  $\mathbb{I}_3$  admits the decomposition

$$\mathbb{I}_3 = \frac{\alpha_k (\mu - \gamma_{k+1})}{2} \|v_{k+1} - x^*\|^2 - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2 + \gamma_k \langle v_{k+1} - v_k, v_{k+1} - x^* \rangle. \quad (3.11)$$

In view of (3.1c), it is not hard to find

$$\gamma_k (v_{k+1} - v_k) = \mu \alpha_k (x_{k+1} - v_{k+1}) - \alpha_k (\xi_{k+1} + A^\top \lambda_{k+1}),$$

where  $\xi_{k+1} \in \partial f(x_{k+1})$ . Hence,  $\mathbb{I}_3$  can be further expanded by that

$$\begin{aligned}\mathbb{I}_3 &= \mu\alpha_k \langle x_{k+1} - v_{k+1}, v_{k+1} - x^* \rangle - \alpha_k \langle \xi_{k+1} + A^\top \lambda^*, v_{k+1} - x^* \rangle \\ &\quad + \frac{\alpha_k(\mu - \gamma_{k+1})}{2} \|v_{k+1} - x^*\|^2 - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2 \\ &\quad - \alpha_k \langle Av_{k+1} - b, \lambda_{k+1} - \lambda^* \rangle,\end{aligned}\tag{3.12}$$

where the last term in the above equality offsets the last term in (3.10). The first cross term in (3.12) is rewritten as

$$\langle x_{k+1} - v_{k+1}, v_{k+1} - x^* \rangle = \frac{1}{2} \left( \|x_{k+1} - x^*\|^2 - \|x_{k+1} - v_{k+1}\|^2 - \|v_{k+1} - x^*\|^2 \right).$$

Observing (3.1b), we split the second cross term in (3.12) and get

$$\begin{aligned}& -\alpha_k \langle \xi_{k+1} + A^\top \lambda^*, v_{k+1} - x^* \rangle \\ &= -\langle \xi_{k+1} + A^\top \lambda^*, x_{k+1} - x_k \rangle - \alpha_k \langle \xi_{k+1} + A^\top \lambda^*, x_{k+1} - x^* \rangle,\end{aligned}$$

By the fact that  $\mathcal{L}(\cdot, \lambda^*)$  is  $\mu$ -convex and  $\xi_{k+1} + A^\top \lambda^* \in \partial_x \mathcal{L}(x_{k+1}, \lambda^*)$ , we obtain from (1.10) that

$$\begin{aligned}& \frac{\mu\alpha_k}{2} \|x_{k+1} - x^*\|^2 - \alpha_k \langle \xi_{k+1} + A^\top \lambda^*, v_{k+1} - x^* \rangle \\ &\leq \mathcal{L}(x_k, \lambda^*) - \mathcal{L}(x_{k+1}, \lambda^*) + \alpha_k (\mathcal{L}(x^*, \lambda^*) - \mathcal{L}(x_{k+1}, \lambda^*)).\end{aligned}\tag{3.13}$$

Note that the first term in (3.13) nullifies  $\mathbb{I}_1$  exactly. We find, after rearranging terms and dropping the surplus negative square term  $-\|x_{k+1} - v_{k+1}\|^2$ , that

$$\mathcal{E}_{k+1} - \mathcal{E}_k \leq -\alpha_k \mathcal{E}_{k+1} - \frac{\theta_k}{2} \|\lambda_{k+1} - \lambda_k\|^2 - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2,\tag{3.14}$$

which implies (3.8) immediately.

From (3.8), we have

$$\mathcal{E}_{k+1} \leq \frac{\mathcal{E}_k}{1 + \alpha_k} = \frac{\theta_{k+1}}{\theta_k} \mathcal{E}_k \implies \mathcal{E}_k \leq \frac{\theta_k}{\theta_0} \mathcal{E}_0,\tag{3.15}$$

which promises (3.9b) directly. Hence it is enough to establish (3.9a) and (3.9c). By (3.1a), we find

$$\lambda_{k+1} = \lambda_k - \frac{1}{\theta_k} (Ax_k - b) + \frac{1}{\theta_{k+1}} (Ax_{k+1} - b).\tag{3.16}$$

Whence, it follows that

$$\lambda_k - \frac{1}{\theta_k}(Ax_k - b) = \lambda_0 - \frac{1}{\theta_0}(Ax_0 - b), \quad k \in \mathbb{N}, \quad (3.17)$$

which implies the estimate

$$\|Ax_k - b\| = \theta_k \|\lambda_k - \lambda_0 + \theta_0^{-1}(Ax_0 - b)\| \leq \theta_k \|\lambda_k - \lambda_0\| + \frac{\theta_k}{\theta_0} \|Ax_0 - b\|.$$

Thanks to the estimate (3.15), we have  $\theta_0 \|\lambda_k - \lambda^*\|^2 \leq 2\mathcal{E}_0$  and moreover,

$$\|Ax_k - b\| \leq \theta_k \|\lambda_k - \lambda^*\| + \theta_k \|\lambda_0 - \lambda^*\| + \frac{\theta_k}{\theta_0} \|Ax_0 - b\| \leq \frac{\theta_k}{\theta_0} \mathcal{R}_0,$$

which proves (3.9a). In addition, it is clear that

$$\begin{aligned} 0 &\leq \mathcal{L}(x_k, \lambda^*) - \mathcal{L}(x^*, \lambda_k) = f(x_k) - f(x^*) + \langle \lambda^*, Ax_k - b \rangle \\ &\leq \mathcal{L}(x_k, \lambda^*) - \mathcal{L}(x^*, \lambda_k), \end{aligned}$$

and thus there holds

$$\begin{aligned} |f(x_k) - f(x^*)| &\leq \|\lambda^*\| \|Ax_k - b\| + \mathcal{L}(x_k, \lambda^*) - \mathcal{L}(x^*, \lambda_k) \\ &\leq \frac{\theta_k}{\theta_0} (\mathcal{E}_0 + \|\lambda^*\| \mathcal{R}_0). \end{aligned}$$

This establishes (3.9c) and finishes the proof.  $\square$

## 3.2. An Implicit-Explicit Scheme

We now propose an implicit-explicit scheme for the two-block variant primal-dual flow model (2.31): given the initial guesses  $x_0, v_0 \in \mathbb{R}^n$ ,  $y_0, u_0 \in \mathbb{R}^r$  and  $\lambda_0 \in \mathbb{R}^m$ , do the iteration

$$\begin{cases} \frac{x_{k+1} - x_k}{\alpha_k} = v_{k+1} - x_{k+1}, \end{cases} \quad (3.18a)$$

$$\begin{cases} \gamma_k \frac{v_{k+1} - v_k}{\alpha_k} \in \mu_f(x_{k+1} - v_{k+1}) - \partial_x \mathcal{L}(x_{k+1}, y_{k+1}, \tilde{\lambda}_{k+1}), \end{cases} \quad (3.18b)$$

$$\begin{cases} \theta_k \frac{\lambda_{k+1} - \lambda_k}{\alpha_k} = \nabla_\lambda \mathcal{L}(v_{k+1}, u_{k+1}, \lambda_{k+1}), \end{cases} \quad (3.18c)$$

$$\begin{cases} \beta_k \frac{u_{k+1} - u_k}{\alpha_k} \in \mu_g(y_{k+1} - u_{k+1}) - \partial_y \mathcal{L}(x_{k+1}, y_{k+1}, \tilde{\lambda}_{k+1}), \end{cases} \quad (3.18d)$$

$$\begin{cases} \frac{y_{k+1} - y_k}{\alpha_k} = u_{k+1} - y_{k+1}, \end{cases} \quad (3.18e)$$

where  $\alpha_k > 0$  is the time step size and

$$\tilde{\lambda}_{k+1} = \lambda_k + \alpha_k / \theta_k (Av_{k+1} + Bu_k - b). \quad (3.19)$$

Analogously to (3.1), both  $f$  and  $g$  can be nonsmooth with  $\partial_x \mathcal{L}(x, y, \lambda) := \partial f(x) + A^\top \lambda$  and  $\partial_y \mathcal{L}(x, y, \lambda) := \partial g(y) + A^\top \lambda$ , respectively. The parameter system (2.30) is discretized implicitly by that

$$\frac{\gamma_{k+1} - \gamma_k}{\alpha_k} = \mu_f - \gamma_{k+1}, \quad \frac{\beta_{k+1} - \beta_k}{\alpha_k} = \mu_g - \beta_{k+1}, \quad \frac{\theta_{k+1} - \theta_k}{\alpha_k} = -\theta_{k+1}. \quad (3.20)$$

Rearrange (3.18) in the usual primal-dual formulation:

$$v_{k+1} = x_{k+1} + (x_{k+1} - x_k) / \alpha_k, \quad (3.21a)$$

$$x_{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \mathcal{L}(x, y_{k+1}, \tilde{\lambda}_{k+1}) + \frac{\eta_{f,k}}{2\alpha_k^2} \|x - \tilde{x}_k\|^2 \right\}, \quad (3.21b)$$

$$\lambda_{k+1} = \lambda_k + \alpha_k / \theta_k (Av_{k+1} + Bu_{k+1} - b), \quad (3.21c)$$

$$y_{k+1} = \underset{y \in \mathbb{R}^r}{\operatorname{argmin}} \left\{ \mathcal{L}(x_{k+1}, y, \tilde{\lambda}_{k+1}) + \frac{\eta_{g,k}}{2\alpha_k^2} \|y - \tilde{y}_k\|^2 \right\}, \quad (3.21d)$$

$$u_{k+1} = y_{k+1} + (y_{k+1} - y_k) / \alpha_k, \quad (3.21e)$$

where  $\eta_{f,k} := (\alpha_k + 1)\gamma_k + \mu_f \alpha_k$ ,  $\eta_{g,k} := (\alpha_k + 1)\beta_k + \mu_g \alpha_k$  and

$$\tilde{x}_k := \frac{\alpha_k \gamma_k v_k + (\gamma_k + \mu_f \alpha_k) x_k}{\eta_{f,k}}, \quad \tilde{y}_k := \frac{\alpha_k \beta_k u_k + (\beta_k + \mu_g \alpha_k) y_k}{\eta_{g,k}}. \quad (3.22)$$

By (3.20), (3.21a) and (3.21e), we can also rewrite (3.22) as

$$\tilde{x}_k = x_k + \frac{\gamma_k \alpha_k}{\eta_{f,k} \alpha_{k-1}} (x_k - x_{k-1}), \quad \tilde{y}_k = y_k + \frac{\beta_k \alpha_k}{\eta_{g,k} \alpha_{k-1}} (y_k - y_{k-1}), \quad (3.23)$$

with  $\alpha_{-1} = 1$ ,  $x_{-1} = 2x_0 - v_0$  and  $y_{-1} = 2y_0 - u_0$ .

### 3.2.1. A single-step analysis

We shall give a one-iteration analysis for the implicit-explicit discretization (3.18). Then the nonergodic mixed-type convergence rates of our first family of methods can be obtained. To do this, introduce a Lyapunov function

$$\begin{aligned} \mathcal{E}_k := & \mathcal{L}(x_k, y_k, \lambda^*) - \mathcal{L}(x^*, y^*, \lambda_k) + \frac{\gamma_k}{2} \|v_k - x^*\|^2 \\ & + \frac{\beta_k}{2} \|u_k - y^*\|^2 + \frac{\theta_k}{2} \|\lambda_k - \lambda^*\|^2, \end{aligned} \quad (3.24)$$

which corresponds to the discrete analogue to (2.32).

**Lemma 3.2.1.** *Let  $k \in \mathbb{N}$  be fixed and assume  $f$  is  $\mu_f$ -convex and  $g$  is  $\mu_g$ -convex with  $\mu_f, \mu_g \geq 0$ . For the implicit-explicit scheme (3.18) with (3.20) and the step size  $\alpha_k > 0$ , we have*

$$\begin{aligned} \mathcal{E}_{k+1} - \mathcal{E}_k &\leq -\alpha_k \mathcal{E}_{k+1} + \frac{\theta_k}{2} \left\| \lambda_{k+1} - \tilde{\lambda}_{k+1} \right\|^2 \\ &\quad - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2 - \frac{\beta_k}{2} \|u_{k+1} - u_k\|^2. \end{aligned} \quad (3.25)$$

**Proof.** Let us calculate the difference  $\mathcal{E}_{k+1} - \mathcal{E}_k = \mathbb{I}_1 + \mathbb{I}_2 + \mathbb{I}_3 + \mathbb{I}_4$ , where

$$\begin{aligned} \mathbb{I}_1 &:= \mathcal{L}(x_{k+1}, y_{k+1}, \lambda^*) - \mathcal{L}(x_k, y_k, \lambda^*), \\ \mathbb{I}_2 &:= \frac{\theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2 - \frac{\theta_k}{2} \|\lambda_k - \lambda^*\|^2, \\ \mathbb{I}_3 &:= \frac{\gamma_{k+1}}{2} \|v_{k+1} - x^*\|^2 - \frac{\gamma_k}{2} \|v_k - x^*\|^2, \\ \mathbb{I}_4 &:= \frac{\beta_{k+1}}{2} \|u_{k+1} - y^*\|^2 - \frac{\beta_k}{2} \|u_k - y^*\|^2. \end{aligned}$$

In what follows, we aim to estimate the above four terms one by one.

By (3.18b) and (3.18d), we find that

$$p_{k+1} := \mu_f(x_{k+1} - v_{k+1}) - \gamma_k \frac{v_{k+1} - v_k}{\alpha_k} \in \partial_x \mathcal{L}(x_{k+1}, y_{k+1}, \tilde{\lambda}_{k+1}), \quad (3.26)$$

$$q_{k+1} := \mu_g(y_{k+1} - u_{k+1}) - \beta_k \frac{u_{k+1} - u_k}{\alpha_k} \in \partial_y \mathcal{L}(x_{k+1}, y_{k+1}, \tilde{\lambda}_{k+1}), \quad (3.27)$$

and it follows that

$$\begin{aligned} \mathbb{I}_1 &= \mathcal{L}(x_{k+1}, y_{k+1}, \tilde{\lambda}_{k+1}) - \mathcal{L}(x_k, y_k, \tilde{\lambda}_{k+1}) \\ &\quad + \left\langle \lambda^* - \tilde{\lambda}_{k+1}, A(x_{k+1} - x_k) + B(y_{k+1} - y_k) \right\rangle \\ &\leq \langle p_{k+1}, x_{k+1} - x_k \rangle + \langle q_{k+1}, y_{k+1} - y_k \rangle \\ &\quad + \left\langle \lambda^* - \tilde{\lambda}_{k+1}, A(x_{k+1} - x_k) + B(y_{k+1} - y_k) \right\rangle. \end{aligned} \quad (3.28)$$

By the equation of the sequence  $\{\theta_k\}$  in (3.20), there holds

$$\begin{aligned} \mathbb{I}_2 &= \frac{\theta_{k+1} - \theta_k}{2} \|\lambda_{k+1} - \lambda^*\|^2 + \frac{\theta_k}{2} (\|\lambda_{k+1} - \lambda^*\|^2 - \|\lambda_k - \lambda^*\|^2) \\ &= -\frac{\alpha_k \theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2 + \theta_k \langle \lambda_{k+1} - \lambda_k, \lambda_{k+1} - \lambda^* \rangle - \frac{\theta_k}{2} \|\lambda_{k+1} - \lambda_k\|^2. \end{aligned}$$

To match the term  $\tilde{\lambda}_{k+1}$  in (3.28), we use (3.18c) to rewrite the above cross

term and obtain

$$\begin{aligned}
 & \theta_k \langle \lambda_{k+1} - \lambda_k, \lambda_{k+1} - \lambda^* \rangle \\
 &= \theta_k \left\langle \lambda_{k+1} - \lambda_k, \lambda_{k+1} - \tilde{\lambda}_{k+1} + \tilde{\lambda}_{k+1} - \lambda^* \right\rangle - \frac{\theta_k}{2} \|\lambda_{k+1} - \lambda_k\|^2 \\
 &= \alpha_k \left\langle Av_{k+1} + Bu_{k+1} - b, \tilde{\lambda}_{k+1} - \lambda^* \right\rangle + \frac{\theta_k}{2} \|\lambda_{k+1} - \tilde{\lambda}_{k+1}\|^2 - \frac{\theta_k}{2} \|\lambda_k - \tilde{\lambda}_{k+1}\|^2.
 \end{aligned}$$

This implies the estimate

$$\begin{aligned}
 \mathbb{I}_2 &\leq \alpha_k \left\langle Av_{k+1} + Bu_{k+1} - b, \tilde{\lambda}_{k+1} - \lambda^* \right\rangle \\
 &\quad + \frac{\theta_k}{2} \|\lambda_{k+1} - \tilde{\lambda}_{k+1}\|^2 - \frac{\alpha_k \theta_{k+1}}{2} \|\lambda_{k+1} - \lambda^*\|^2.
 \end{aligned} \tag{3.29}$$

Similarly, by the equation of  $\{\gamma_k\}$  in (3.20), we have

$$\begin{aligned}
 \mathbb{I}_3 &= \frac{\gamma_{k+1} - \gamma_k}{2} \|v_{k+1} - x^*\|^2 + \frac{\gamma_k}{2} \left( \|v_{k+1} - x^*\|^2 - \|v_k - x^*\|^2 \right) \\
 &= \frac{\alpha_k(\mu_f - \gamma_{k+1})}{2} \|v_{k+1} - x^*\|^2 - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2 + \gamma_k \langle v_{k+1} - v_k, v_{k+1} - x^* \rangle.
 \end{aligned} \tag{3.30}$$

In view of (3.26), we rewrite the last term by that

$$\gamma_k \langle v_{k+1} - v_k, v_{k+1} - x^* \rangle = \mu_f \alpha_k \langle x_{k+1} - v_{k+1}, v_{k+1} - x^* \rangle - \alpha_k \langle p_{k+1}, v_{k+1} - x^* \rangle.$$

Using (3.21a) and summarizing the above decompositions yield that

$$\begin{aligned}
 \mathbb{I}_3 &= -\frac{\alpha_k \gamma_{k+1}}{2} \|v_{k+1} - x^*\|^2 - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2 - \frac{\mu_f \alpha_k}{2} \|x_{k+1} - v_{k+1}\|^2 \\
 &\quad + \frac{\mu_f \alpha_k}{2} \|x_{k+1} - x^*\|^2 - \alpha_k \langle p_{k+1}, x_{k+1} - x^* \rangle - \langle p_{k+1}, x_{k+1} - x_k \rangle.
 \end{aligned} \tag{3.31}$$

Analogously, by (3.20), (3.27) and (3.21e), we have

$$\begin{aligned}
 \mathbb{I}_4 &= -\frac{\alpha_k \beta_{k+1}}{2} \|u_{k+1} - y^*\|^2 - \frac{\beta_k}{2} \|u_{k+1} - u_k\|^2 - \frac{\mu_g \alpha_k}{2} \|y_{k+1} - u_{k+1}\|^2 \\
 &\quad + \frac{\mu_g \alpha_k}{2} \|y_{k+1} - y^*\|^2 - \alpha_k \langle q_{k+1}, y_{k+1} - y^* \rangle - \langle q_{k+1}, y_{k+1} - y_k \rangle.
 \end{aligned} \tag{3.32}$$

Now, collecting (3.29), (3.31) and (3.32), we arrive at the upper bound

$$\begin{aligned}
 \mathbb{I}_2 + \mathbb{I}_3 + \mathbb{I}_4 &\leq -\alpha_k \mathcal{E}_{k+1} + \frac{\theta_k}{2} \|\lambda_{k+1} - \tilde{\lambda}_{k+1}\|^2 \\
 &\quad - \frac{\gamma_k}{2} \|v_{k+1} - v_k\|^2 - \frac{\beta_k}{2} \|u_{k+1} - u_k\|^2 \\
 &\quad - \langle p_{k+1}, x_{k+1} - x_k \rangle - \langle q_{k+1}, y_{k+1} - y_k \rangle \\
 &\quad - \left\langle \lambda^* - \tilde{\lambda}_{k+1}, A(x_{k+1} - x_k) + B(y_{k+1} - y_k) \right\rangle,
 \end{aligned}$$

where we used the  $\mu_f$ -convexity of  $f$  and the  $\mu_g$ -convexity of  $g$  (cf.(1.10)).

Plugging the estimate (3.28) of  $\mathbb{I}_1$  into the above estimate establishes (3.25) and finishes the proof of this lemma.  $\square$

### 3.2.2. Non-ergodic convergence rate

Based on the one-iteration estimate (3.25), we shall establish the non-ergodic convergence rate of the scheme (3.18) with properly chosen step size sequence  $\{\alpha_k\}$ .

In view of (3.19) and (3.21c), we have

$$\lambda_{k+1} - \tilde{\lambda}_{k+1} = \alpha_k / \theta_k B (u_{k+1} - u_k). \quad (3.33)$$

Hence, if  $\alpha_k^2 \|B\|^2 \leq \beta_k \theta_k$ , then

$$\frac{\theta_k}{2} \left\| \lambda_{k+1} - \tilde{\lambda}_{k+1} \right\|^2 = \frac{\alpha_k^2}{2\theta_k} \|B(u_{k+1} - u_k)\|^2 \leq \frac{\beta_k}{2} \|u_{k+1} - u_k\|^2, \quad (3.34)$$

and plugging this into (3.25) gives the contraction estimate

$$\mathcal{E}_{k+1} - \mathcal{E}_k \leq -\alpha_k \mathcal{E}_{k+1} \quad \forall k \in \mathbb{N}. \quad (3.35)$$

**Theorem 3.2.1.** *Assume  $f$  is  $\mu_f$ -convex and  $g$  is  $\mu_g$ -convex with  $\mu_f, \mu_g \geq 0$ . For the implicit-explicit scheme (3.18) with (3.20) and the step size  $\alpha_k = \sqrt{\beta_k \theta_k} / \|B\|$ , we have*

$$\begin{cases} \|Ax_k + By_k - b\| \leq \frac{\theta_k}{\theta_0} \mathcal{R}_0, \\ \mathcal{L}(x_k, y_k, \lambda^*) - \mathcal{L}(x^*, y^*, \lambda_k) \leq \frac{\theta_k}{\theta_0} \mathcal{E}_0, \\ |F(x_k, y_k) - F^*| \leq \frac{\theta_k}{\theta_0} (\mathcal{E}_0 + \|\lambda^*\| \mathcal{R}_0). \end{cases} \quad (3.36)$$

Above,  $\mathcal{R}_0 := \sqrt{2\mathcal{E}_0} + \|\lambda_0 - \lambda^*\| + \|Ax_0 + By_0 - b\|$  and

$$\frac{\theta_k}{\theta_0} \leq \min \left\{ \frac{\|B\|}{\|B\| + \sqrt{\beta_0} k}, \frac{Q^2}{(Q + \sqrt{\theta_0 \beta_{\min}} k)^2} \right\}, \quad (3.37)$$

where  $\beta_{\min} = \min\{\mu_g, \beta_0\}$ ,  $\beta_{\max} = \max\{\mu_g, \beta_0\}$  and

$$Q = \sqrt{\|B\|} + \sqrt{\|B\| + \sqrt{\theta_0 \beta_{\max}}}.$$

**Proof.** Based on the above discussions, the step size choice  $\alpha_k = \sqrt{\beta_k \theta_k} / \|B\|$  promises the contraction (3.35). Similarly with (3.15), we have  $\mathcal{E}_k \leq \theta_k / \theta_0 \mathcal{E}_0$ , which implies

$$\mathcal{L}(x_k, y_k, \lambda^*) - \mathcal{L}(x^*, y^*, \lambda_k) \leq \theta_k / \theta_0 \mathcal{E}_0.$$

Following the proof of (3.9), we can establish

$$\|Ax_k + By_k - b\| \leq \theta_k \mathcal{R}_0, \text{ and } |F(x_k, y_k) - F^*| \leq \theta_k / \theta_0 (\mathcal{E}_0 + \|\lambda^*\| \mathcal{R}_0).$$

This proves (3.36).

It remains to verify the decay estimate (3.37). By (3.20), it follows that

$$\theta_{k+1} - \theta_k = -\alpha_k \theta_{k+1} = -\sqrt{\beta_k \theta_k} \theta_{k+1} / \|B\|, \quad (3.38)$$

and we also have

$$\beta_{k+1} = \frac{\beta_k + \mu_g \alpha_k}{1 + \alpha_k} \geq \frac{\beta_k}{1 + \alpha_k} = \frac{\theta_{k+1}}{\theta_k} \beta_k \implies \beta_k \geq \theta_k \beta_0 / \theta_0.$$

Hence, invoking (3.38) gives

$$\theta_{k+1} - \theta_k \leq -\sqrt{\beta_0 / \theta_0} / \|B\| \theta_k \theta_{k+1}.$$

which leads to

$$\frac{1}{\theta_{k+1}} - \frac{1}{\theta_k} \geq \frac{\sqrt{\beta_0 / \theta_0}}{\|B\|} \implies \frac{\theta_k}{\theta_0} \leq \frac{\|B\|}{\|B\| + \sqrt{\beta_0} k}. \quad (3.39)$$

On the other hand, it is not hard to find that  $\beta_k \geq \beta_{\min}$  and this together with (3.38) gives

$$\theta_{k+1} - \theta_k \leq -\sqrt{\beta_{\min}} / \|B\| \sqrt{\theta_k} \theta_{k+1}. \quad (3.40)$$

Observing that  $\theta_k \leq \theta_0$  and  $\beta_k \leq \beta_{\max}$  for all  $k \in \mathbb{N}$ , we obtain

$$\alpha_k = \sqrt{\beta_k \theta_k} / \|B\| \leq \sqrt{\theta_0 \beta_{\max}} / \|B\|,$$

and this implies

$$\frac{\frac{1}{\sqrt{\theta_{k+1}}} - \frac{1}{\sqrt{\theta_k}}}{\frac{\sqrt{\theta_k}}{\theta_{k+1}} - \frac{1}{\sqrt{\theta_k}}} = \frac{\sqrt{\theta_{k+1}}}{\sqrt{\theta_{k+1}} + \sqrt{\theta_k}} = \frac{1}{1 + \sqrt{1 + \alpha_k}} \geq \frac{\sqrt{\|B\|}}{Q}.$$

Combining this with (3.40) yields that

$$\frac{1}{\sqrt{\theta_{k+1}}} - \frac{1}{\sqrt{\theta_k}} \geq \frac{\sqrt{\beta_{\min}}}{Q} \implies \frac{\theta_k}{\theta_0} \leq \frac{Q^2}{(Q + \sqrt{\theta_0 \beta_{\min}} k)^2}.$$

Collecting this and (3.39) proves (3.37) and concludes the proof.  $\square$

# Chapter 4

## Two Alternating Direction Methods of Multipliers

In this and the next chapter, based on the time discretizations in the previous chapter, we propose efficient optimization solvers for the generalized transportation problem (1.8).

More precisely, a semi-proximal ADMM is firstly considered in Section 4.1, and the global linear rate  $\mathcal{O}(\rho^k)$  will be established in Section 4.2. Then in Section 4.3 we present a minor variant of the implicit-explicit scheme (3.18), which leads to an accelerated ADMM and possesses the nonergodic sublinear rate  $\mathcal{O}(1/k)$ . Moreover, some numerical experiments are provided in Section 4.4.

### 4.1. Semi-proximal ADMM

Let  $M = m + n$  and  $J = mn + M$ . Introduce  $\mathbf{x} = (x, y, z)$  and

$$H = (G, I_Y, I_Z) = \begin{pmatrix} T & I \\ \Pi & O \end{pmatrix}.$$

Rewrite (1.8) as follows

$$\min_{\mathbf{x} \in \mathbb{R}^J, \mathbf{y} \in \mathbb{R}^J} f(\mathbf{x}) + \delta_\Sigma(\mathbf{y}) \quad \text{s.t. } \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{y} = \mathbf{b}, \quad (4.1)$$

where  $f(\mathbf{x}) = \sigma/2 \|\mathbf{x} - \phi\|^2 + \mathbf{c}^\top \mathbf{x}$  and

$$\mathcal{A} = \begin{pmatrix} H \\ I \end{pmatrix}, \quad \mathcal{B} = \begin{pmatrix} O \\ -I \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0}_J \end{pmatrix}.$$

Introduce the Lagrangian

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \delta_\Sigma(\mathbf{y}) + \langle \boldsymbol{\lambda}, \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{y} - \mathbf{b} \rangle.$$

Given any  $\delta > 0$ , define the augmented Lagrangian function for (4.1) by that

$$\mathcal{L}_\delta(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \delta_\Sigma(\mathbf{y}) + \langle \boldsymbol{\lambda}, \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{y} - \mathbf{b} \rangle + \frac{\delta}{2} \|\mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{y} - \mathbf{b}\|^2. \quad (4.2)$$

Also we set  $\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \mathcal{L}_0(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$  for  $\delta = 0$ . Assume (4.2) admits a saddle point  $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*)$  such that

$$\mathcal{L}_\delta(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}) \leq \mathcal{L}_\delta(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \leq \mathcal{L}_\delta(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}^*) \quad \forall (\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) \in \Sigma,$$

Throughout, we use  $\Omega^*$  to denote the saddle point point set of (4.2). In other words, for any  $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \in \Omega^*$ , we have

$$\begin{cases} 0 = \mathcal{A}^\top \boldsymbol{\lambda}^* + \nabla f(\mathbf{x}^*), \\ 0 = \mathbf{b} - \mathcal{A}\mathbf{x}^* - \mathcal{B}\mathbf{y}^*, \\ 0 \in \mathcal{B}^\top \boldsymbol{\lambda}^* + \mathcal{N}_\Sigma(\mathbf{y}^*). \end{cases} \quad (4.3)$$

Recall the semi-proximal ADMM [49]

$$\begin{cases} \mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathbb{R}^J}{\operatorname{argmin}} \left\{ \mathcal{L}_\delta(\mathbf{x}, \mathbf{y}_k, \boldsymbol{\lambda}_k) + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_k\|_P^2 \right\}, \end{cases} \quad (4.4a)$$

$$\begin{cases} \mathbf{y}_{k+1} = \underset{\mathbf{y} \in \mathbb{R}^J}{\operatorname{argmin}} \mathcal{L}_\delta(\mathbf{x}_{k+1}, \mathbf{y}, \boldsymbol{\lambda}_k), \end{cases} \quad (4.4b)$$

$$\begin{cases} \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \delta (\mathcal{A}\mathbf{x}_{k+1} + \mathcal{B}\mathbf{y}_{k+1} - \mathbf{b}), \end{cases} \quad (4.4c)$$

where  $P$  is some symmetric positive semidefinite matrix. Let

$$\widetilde{D} := \operatorname{diag}(I_{mn}, O_{M \times M}), \quad \widetilde{\mathbf{c}} := \begin{pmatrix} \sigma\phi - \mathbf{c} \\ \mathbf{0}_M \end{pmatrix}.$$

Then reformulate (4.4) more clearly as follows

$$\begin{cases} \mathbf{x}_{k+1} = (\sigma\widetilde{D} + P + \delta\mathcal{A}^\top\mathcal{A})^{-1}(P\mathbf{x}_k + \widetilde{\mathbf{c}} - \mathcal{A}^\top\boldsymbol{\lambda}_k - \delta\mathcal{A}^\top(\mathcal{B}\mathbf{y}_k - \mathbf{b})), \\ \mathbf{y}_{k+1} = \operatorname{proj}_\Sigma(\mathbf{x}_{k+1} - \mathcal{B}^\top\boldsymbol{\lambda}_k/\delta), \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \delta(\mathcal{A}\mathbf{x}_{k+1} + \mathcal{B}\mathbf{y}_{k+1} - \mathbf{b}). \end{cases} \quad (4.5)$$

Below, we summarize (4.4) with  $P = \delta I$  in Algorithm 1. Since  $\Sigma$  is a simple box region,  $\mathbf{proj}_\Sigma$  is easy to calculate. The matrix inverse operation admits closed form and shall be given in the next section.

---

**Algorithm 1** Semi-proximal ADMM for solving (4.1)

**Input:**  $\delta > 0$ ,  $\mathbf{x}_0 \in \mathbb{R}^J$ ,  $\mathbf{y}_0 \in \Sigma$  and  $\boldsymbol{\lambda}_0 \in \mathbb{R}^{J+M+l}$ .

- 1: **for**  $k = 0, 1, \dots$  **do**
- 2:   Set  $\mathbf{s}_k = \delta \mathbf{x}_k + \tilde{\mathbf{c}} - \mathcal{A}^\top \boldsymbol{\lambda}_k - \delta \mathcal{A}^\top (\mathcal{B} \mathbf{y}_k - \mathbf{b})$ .
- 3:   Update  $\mathbf{x}_{k+1} = (\sigma \tilde{D} + P + \delta \mathcal{A}^\top \mathcal{A})^{-1} \mathbf{s}_k$  by (4.6), (4.7) and (4.8).
- 4:   Update  $\mathbf{y}_{k+1} = \mathbf{proj}_\Sigma(\mathbf{x}_{k+1} - \mathcal{B}^\top \boldsymbol{\lambda}_k / \delta)$ .
- 5:   Update  $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \delta (\mathcal{A} \mathbf{x}_{k+1} + \mathcal{B} \mathbf{y}_{k+1} - \mathbf{b})$ .
- 6: **end for**

---

**Remark 4.1.1.** Note that the involved matrix-vector multiplication operations are related to  $T$  and  $T^\top$ . It is clear that

$$Tx = \begin{pmatrix} (I_n \otimes \mathbf{1}_m^\top) \text{vec}(X) \\ (\mathbf{1}_n^\top \otimes I_m) \text{vec}(X) \end{pmatrix} = \begin{pmatrix} \text{vec}(X^\top \mathbf{1}_m) \\ \text{vec}(X \mathbf{1}_n) \end{pmatrix},$$

where  $x = \text{vec}(X) \in \mathbb{R}^{mn}$ . This takes almost nothing since it only requires the vectors of row sum and column sum with respect to  $X$ . Besides, for any given  $z = (z_1^\top, z_2^\top)^\top \in \mathbb{R}^M$  with  $z_1 \in \mathbb{R}^m$  and  $z_2 \in \mathbb{R}^n$ , we have

$$T^\top z = (I_n \otimes \mathbf{1}_m) z_1 + (\mathbf{1}_n^\top \otimes I_m) z_2 = \text{vec}(\mathbf{1}_n z_1^\top) + \text{vec}(z_2 \mathbf{1}_m^\top).$$

This just involves vector replication and the operation cost is  $\mathcal{O}(mn)$ .

#### 4.1.1. Matrix inverse formulae

Notice that

$$\mathcal{A}^\top \mathcal{A} = I + H^\top H.$$

If we simply take  $P = \delta I$ , then we can find explicit inverse formula of

$$\mathbf{A} := \sigma \tilde{D} + P + \delta \mathcal{A}^\top \mathcal{A} = \sigma \tilde{D} + 2\delta I + H^\top H$$

by using Sherman–Morrison–Woodbury formula

$$(D + H^\top H)^{-1} = D^{-1} - D^{-1} H^\top (I + H D^{-1} H^\top)^{-1} H D^{-1}, \quad (4.6)$$

where  $D := \sigma\tilde{D} + 2\delta I = \text{diag}((\sigma + 2\delta)I_{mn}, 2\delta I_M)$  is diagonal. Let  $\sigma_1 = 1/(\sigma + 2\delta)$  and  $\sigma_2 := 1/(2\delta)$ . We then focus on

$$\mathbf{H} := I + HD^{-1}H^\top = \begin{pmatrix} (\sigma_2 + 1)I + \sigma_1 TT^\top & \sigma_1 T\Pi^\top \\ \sigma_1 \Pi T^\top & I + \sigma_1 \Pi\Pi^\top \end{pmatrix} := \begin{pmatrix} U & L^\top \\ L & W \end{pmatrix}.$$

The inverse is given by

$$\mathbf{H}^{-1} = \begin{pmatrix} U^{-1} + U^{-1}L^\top S^{-1}LU^{-1} & -U^{-1}L^\top S^{-1} \\ -S^{-1}LU^{-1} & S^{-1} \end{pmatrix}, \quad (4.7)$$

where  $S = W - LU^{-1}L^\top \in \mathbb{R}^{l \times l}$  denotes the Schur complement.

If  $l$  is small, which is indeed true for all transport-like problems given in Section 1.1.3, then  $S$  is easy to invert and what we shall pay attention to is  $U^{-1}$ . Observe that

$$U = \begin{pmatrix} (\sigma_2 + 1 + m\sigma_1)I & \sigma_1 \mathbf{1}_{n \times m} \\ \sigma_1 \mathbf{1}_{m \times n} & (\sigma_2 + 1 + n\sigma_1)I \end{pmatrix} := \begin{pmatrix} V & R^\top \\ R & Q \end{pmatrix}.$$

The inverse is given by

$$U^{-1} = \begin{pmatrix} V^{-1} + V^{-1}R^\top K^{-1}RV^{-1} & -V^{-1}R^\top K^{-1} \\ -K^{-1}RV^{-1} & K^{-1} \end{pmatrix}, \quad (4.8)$$

where  $K = Q - RV^{-1}R^\top = (\sigma_2 + 1 + n\sigma_1)I - \frac{n\sigma_1^2}{\sigma_2 + 1 + n\sigma_1} \mathbf{1}_{m \times m}$ . Using the Sherman–Morrison formula again, we obtain

$$K^{-1} = \frac{1}{\sigma_2 + 1 + n\sigma_1} I + \frac{n\sigma_1^2}{\sigma_2 + 1 + n\sigma_1} \cdot \frac{\mathbf{1}_{m \times m}}{(\sigma_2 + 1)^2 + \sigma_1(\sigma_2 + 1)(m + n)}.$$

Taking this into (4.8) gives the exact expression of  $U^{-1}$ .

According to the discussions in Remark 4.1.1, we claim that the total operation cost of  $\mathbf{A}^{-1}\mathbf{s}$  is  $\mathcal{O}(mn)$ .

## 4.2. Convergence of the Semi-proximal ADMM

### 4.2.1. Ergodic sublinear rate

Based on [49, Theorem 3.2] or [41, Theorem B.1], it is not hard to obtain the convergence and the ergodic sublinear rate  $\mathcal{O}(1/\sqrt{k})$  for the semi-proximal ADMM (4.4); see [49, Theorem 3.3] for instance.

Below, let us briefly summarize these basic results. Introduce a block diagonal matrix  $R := \text{diag}(\delta I_J, \delta I_J, 1/\delta I_{J+M+l})$ . Let  $\mathbf{X}^* = (\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \in \Omega^*$  and denote by  $\{\mathbf{X}_k\} = \{(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k)\}$  the sequence generated by Algorithm 1. The contraction property follows directly from [49, Theorem 3.2]:

$$\begin{aligned} & \|\mathbf{X}_{k+1} - \mathbf{X}^*\|_R^2 - \|\mathbf{X}_k - \mathbf{X}^*\|_R^2 \\ & \leq -\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_R^2 - 2(\mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}_{k+1}) - \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \boldsymbol{\lambda}^*)). \end{aligned} \quad (4.9)$$

Based on (4.9), we can establish the global convergence:  $\lim_{k \rightarrow \infty} \mathbf{X}_k = \mathbf{X}_\infty \in \Omega^*$ ; see [49, Theorem 3.3]. Additionally, for all  $k \geq 1$ , we have the sublinear rate

$$0 \leq \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}_k) - \mathcal{L}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{y}}_k, \boldsymbol{\lambda}^*) \leq \frac{C_1}{k} \|\mathbf{X}_0 - \mathbf{X}^*\|_R^2, \quad (4.10)$$

$$\|\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b}\|^2 \leq \frac{C_2}{k} \|\mathbf{X}_0 - \mathbf{X}^*\|_R^2, \quad (4.11)$$

where  $\tilde{\mathbf{x}}_k := \frac{1}{k} \sum_{i=1}^k \mathbf{x}_i$  and  $\tilde{\mathbf{y}}_k := \frac{1}{k} \sum_{i=1}^k \mathbf{y}_i$ .

#### 4.2.2. Global linear convergence

Following [53, Corollary 3.1] and invoking the celebrated stability result of linear inequality system (cf. [92, Corollary 2.2]), which implies the error bound condition, we are allowed to prove the global linear convergence rate. This means the operation complexity of Algorithm 1 is  $\mathcal{O}(mn |\ln \epsilon| / |\ln \rho|)$  with some constant  $\rho \in (0, 1)$ .

We mention that  $\Sigma$  is a convex polyhedral and for simplicity, we consider the half-space  $\Sigma = \mathbb{R}_+^J$ . Then the necessary optimality condition (4.3) becomes a linear equality-inequality system

$$\mathcal{Q}\mathbf{X}^* = q, \quad \mathcal{S}\mathbf{X}^* \geq 0, \quad (4.12)$$

where  $\mathcal{Q}$  and  $\mathcal{S}$  are constant matrices.

Given any perturbation  $e$ , let  $\mathbf{X}$  solve

$$\mathcal{Q}\mathbf{X} = q + e, \quad \mathcal{S}\mathbf{X} \geq 0, \quad (4.13)$$

then by [92, Corollary 2.2], there exist a constant  $\beta > 0$ , such that

$$\text{dist}^2(\mathbf{X}, \Omega^*) \leq \beta \|e\|^2, \quad (4.14)$$

where  $\text{dist}^2(\mathbf{X}, \Omega^*) := \inf\{\|\mathbf{X} - \mathbf{X}^*\|^2 : \mathbf{X}^* \in \Omega^*\}$ .

**Lemma 4.2.1.** *Let  $\{\mathbf{X}_k\}$  be generated by Algorithm 1, then for all  $k \in \mathbb{N}$ ,*

$$\text{dist}_R^2(\mathbf{X}_{k+1}, \Omega^*) \leq C \|\mathbf{X}_{k+1} - \mathbf{X}_k\|_R^2, \quad (4.15)$$

*with some positive constant  $C > 0$ .*

**Theorem 4.2.1.** *Let  $\{\mathbf{X}_k\}$  be generated by Algorithm 1, then we have the global linear rate*

$$\text{dist}_R^2(\mathbf{X}_k, \Omega^*) \leq \rho^k \text{dist}_R^2(\mathbf{X}_0, \Omega^*), \quad k \in \mathbb{N}, \quad (4.16)$$

*where  $\rho \in (0, 1)$  and  $\text{dist}_R^2(\mathbf{X}, \Omega^*) := \inf_{\mathbf{X}^* \in \Omega^*} \|\mathbf{X} - \mathbf{X}^*\|_R^2$ .*

**Proof.** From the contraction property (4.9) we have

$$\text{dist}_R^2(\mathbf{X}_{k+1}, \Omega^*) - \text{dist}_R^2(\mathbf{X}_k, \Omega^*) \leq -\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_R^2.$$

Thanks to the error bound condition (4.14), it holds that

$$\text{dist}_R^2(\mathbf{X}_{k+1}, \Omega^*) - \text{dist}_R^2(\mathbf{X}_k, \Omega^*) \leq -C_1 \text{dist}_R^2(\mathbf{X}_{k+1}, \Omega^*),$$

with some  $C_1 > 0$ . Hence, we obtain the linear contraction

$$\text{dist}_R^2(\mathbf{X}_{k+1}, \Omega^*) \leq \frac{1}{1+C_1} \text{dist}_R^2(\mathbf{X}_k, \Omega^*).$$

This implies (4.16) and completes the proof.  $\square$

## 4.3. An Accelerated Proximal ADMM

### 4.3.1. Main algorithm

Based on the primal-dual method (3.21) that comes from the implicit-explicit discretization (3.18) for the continuous model (2.31), we propose a minor variant for solving the two-block minimization problem (4.1). More precisely, given  $\mathbf{x}_{-1} = \mathbf{x}_0 \in \mathbb{R}^J$ ,  $\mathbf{y}_{-1} = \mathbf{y}_0 \in \Sigma$  and  $\boldsymbol{\lambda}_0 \in \mathbb{R}^{J+M+l}$ , consider

$$\mathbf{v}_{k+1} = \mathbf{x}_{k+1} + (\mathbf{x}_{k+1} - \mathbf{x}_k)/\alpha_k, \quad (4.17a)$$

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathbb{R}^J}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{y}_{k+1}, \tilde{\boldsymbol{\lambda}}_{k+1}) + \frac{1}{2\alpha_{k+1}} \|\mathbf{x} - \tilde{\mathbf{x}}_k\|^2 \right\}, \quad (4.17b)$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mathcal{A}\mathbf{v}_{k+1} + \mathcal{B}\mathbf{u}_{k+1} - \mathbf{b}, \quad (4.17c)$$

$$\mathbf{y}_{k+1} = \underset{\mathbf{y} \in \mathbb{R}^J}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}, \boldsymbol{\lambda}_{k+1}) + \frac{1}{2\alpha_{k+1}} \|\mathbf{y} - \tilde{\mathbf{y}}_k\|^2 \right\}, \quad (4.17d)$$

$$\mathbf{u}_{k+1} = \mathbf{y}_{k+1} + (\mathbf{y}_{k+1} - \mathbf{y}_k)/\alpha_k, \quad (4.17e)$$

where  $\tilde{\boldsymbol{\lambda}}_{k+1} = \boldsymbol{\lambda}_k + \mathcal{A}\mathbf{v}_{k+1} + \mathcal{B}\mathbf{u}_k - \mathbf{b}$  and

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k + \frac{\alpha_{k+1}}{\alpha_{k-1}}(\mathbf{x}_k - \mathbf{x}_{k-1}), \quad \tilde{\mathbf{y}}_k = \mathbf{y}_k + \frac{\alpha_{k+1}}{\alpha_{k-1}}(\mathbf{y}_k - \mathbf{y}_{k-1}).$$

Moreover, the sequence  $\{\alpha_k\}_{k=-1}^{\infty}$  is defined by that

$$\alpha_{-1}^{-1} = 0, \quad \alpha_k = \frac{1}{k+1}, \quad k \in \mathbb{N}. \quad (4.18)$$

---

**Algorithm 2** Accelerated proximal ADMM for solving (4.1)

---

**Input:**  $\mathbf{x}_0 \in \mathbb{R}^J$ ,  $\mathbf{y}_0 \in \Sigma$  and  $\boldsymbol{\lambda}_0 \in \mathbb{R}^{J+M+l}$ .

- 1: Set  $\mathbf{x}_{-1} = \mathbf{x}_0$ ,  $\mathbf{y}_{-1} = \mathbf{u}_0 = \mathbf{y}_0$ .
- 2: **for**  $k = 0, 1, \dots$  **do**
- 3:   Set  $(\tilde{\mathbf{x}}_k, \tilde{\mathbf{y}}_k) = \frac{2k+2}{k+2}(\mathbf{x}_k, \mathbf{y}_k) - \frac{k}{k+2}(\mathbf{x}_{k-1}, \mathbf{y}_{k-1})$ .
- 4:   Set  $\bar{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k - (k+1)(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b})$ .
- 5:   Set  $\hat{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k - (k+1)(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{u}_k - \mathbf{b})$ .
- 6:   Set  $\mathbf{z}_k = \tilde{\mathbf{c}} - \mathcal{A}^\top \hat{\boldsymbol{\lambda}}_k + (k+2)(\tilde{\mathbf{x}}_k - \mathcal{A}^\top(\mathcal{B}\mathbf{u}_k - \mathbf{b}))$ .
- 7:   Update  $\mathbf{x}_{k+1} = (\sigma\tilde{I} + (k+2)(I + \mathcal{A}\mathcal{A}^\top))^{-1}(\mathbf{z}_k)$  by (4.6)–(4.8).
- 8:   Update  $\mathbf{y}_{k+1} = \text{proj}_\Sigma \left[ \frac{1}{2} \left( \tilde{\mathbf{y}}_k + \mathcal{B}^\top \bar{\boldsymbol{\lambda}}_k / (k+2) - \mathcal{B}^\top(\mathcal{A}\mathbf{x}_{k+1} - \mathbf{b}) \right) \right]$ .
- 9:   Update  $(\mathbf{v}_{k+1}, \mathbf{u}_{k+1}) = (k+2)(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - (k+1)(\mathbf{x}_k, \mathbf{y}_k)$ .
- 10:   Update  $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mathcal{A}\mathbf{v}_{k+1} + \mathcal{B}\mathbf{u}_{k+1} - \mathbf{b}$ .
- 11: **end for**

---

Notice that we use only the Lagrangian function  $\mathcal{L}$  in (4.17a) but the augmented term still exists implicitly since  $\tilde{\boldsymbol{\lambda}}_{k+1}$  depends on  $\mathbf{v}_{k+1}$  and as well as on  $\mathbf{x}_{k+1}$ . To see this, we rewrite (4.17a) as follows

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathbb{R}^J}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{y}_k, \hat{\boldsymbol{\lambda}}_k) + \frac{1}{2\alpha_{k+1}} \|\mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{u}_k - \mathbf{b}\|^2 + \frac{1}{2\alpha_{k+1}} \|\mathbf{x} - \tilde{\mathbf{x}}_k\|^2 \right\}, \quad (4.19)$$

where  $\hat{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k - \alpha_k^{-1}(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{u}_k - \mathbf{b})$ . This gives the solution

$$\mathbf{x}_{k+1} = (\sigma\tilde{I} + \alpha_{k+1}^{-1}(I + \mathcal{A}\mathcal{A}^\top))^{-1} \left( \tilde{\mathbf{c}} - \mathcal{A}^\top \hat{\boldsymbol{\lambda}}_k + \alpha_{k+1}^{-1}(\tilde{\mathbf{x}}_k - \mathcal{A}^\top(\mathcal{B}\mathbf{u}_k - \mathbf{b})) \right),$$

which has closed expression by (4.6), (4.7) and (4.8). Besides, by (4.17b) and (4.17c), we see that

$$\mathbf{y}_{k+1} = \underset{\mathbf{y} \in \mathbb{R}^J}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}, \bar{\boldsymbol{\lambda}}_k) + \frac{1}{2\alpha_{k+1}} \|\mathcal{A}\mathbf{x}_{k+1} + \mathcal{B}\mathbf{y} - \mathbf{b}\|^2 + \frac{1}{2\alpha_{k+1}} \|\mathbf{y} - \tilde{\mathbf{y}}_k\|^2 \right\}, \quad (4.20)$$

where  $\bar{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k - \alpha_k^{-1}(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b})$ . This implies

$$\mathbf{y}_{k+1} = \mathbf{proj}_{\Sigma} \left( \frac{1}{2} (\tilde{\mathbf{y}}_k + \alpha_{k+1} \mathcal{B}^\top \bar{\boldsymbol{\lambda}}_k - \mathcal{B}^\top (\mathcal{A}\mathbf{x}_{k+1} - \mathbf{b})) \right).$$

---

**Algorithm 3** Simplified accelerated proximal ADMM for solving (4.1)

**Input:**  $\mathbf{x}_0, \mathbf{y}_0 \in \mathbb{R}^J$ .

- 1: Set  $\mathbf{x}_{-1} = \mathbf{x}_0, \mathbf{y}_{-1} = \mathbf{u}_0 = \mathbf{y}_0$ .
- 2: **for**  $k = 0, 1, \dots$  **do**
- 3:   Set  $(\tilde{\mathbf{x}}_k, \tilde{\mathbf{y}}_k) = \frac{2k+2}{k+2}(\mathbf{x}_k, \mathbf{y}_k) - \frac{k}{k+2}(\mathbf{x}_{k-1}, \mathbf{y}_{k-1})$ .
- 4:   Set  $\mathbf{z}_k = \tilde{\mathbf{c}} + (k+1)(\mathbf{y}_k - \mathbf{u}_k) + (k+2)(\tilde{\mathbf{x}}_k - \mathcal{A}^\top(\mathcal{B}\mathbf{u}_k - \mathbf{b}))$ .
- 5:   Update  $\mathbf{x}_{k+1} = (\sigma\tilde{I} + (k+2)(I + \mathcal{A}\mathcal{A}^\top))^{-1}(\mathbf{z}_k)$  by (4.6)–(4.8).
- 6:   Update  $\mathbf{y}_{k+1} = \mathbf{proj}_{\Sigma} \left[ \frac{1}{2} (\tilde{\mathbf{y}}_k - \mathcal{B}^\top(\mathcal{A}\mathbf{x}_{k+1} - \mathbf{b})) \right]$ .
- 7:   Update  $(\mathbf{v}_{k+1}, \mathbf{u}_{k+1}) = (k+2)(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - (k+1)(\mathbf{x}_k, \mathbf{y}_k)$ .
- 8: **end for**

---

The iteration procedure (4.17) has been summarized in Algorithm 2. By (4.18), it holds that

$$\alpha_{k+1} - \alpha_k = -\alpha_k \alpha_{k+1} \quad \text{for all } k \in \mathbb{N}. \quad (4.21)$$

Observing this and (4.17c), we have

$$\boldsymbol{\lambda}_{k+1} - \alpha_{k+1}^{-1}(\mathcal{A}\mathbf{x}_{k+1} + \mathcal{B}\mathbf{y}_{k+1} - \mathbf{b}) = \boldsymbol{\lambda}_k - \alpha_k^{-1}(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b}). \quad (4.22)$$

Therefore, if  $\boldsymbol{\lambda}_0 = \mathcal{A}\mathbf{x}_0 + \mathcal{B}\mathbf{y}_0 - \mathbf{b}$ , then  $\boldsymbol{\lambda}_k = \alpha_k^{-1}(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b})$  and the sequence  $\{\boldsymbol{\lambda}_k\}$  can be dropped. For the sake of implementation, we give a simplified version (see Algorithm 3) of Algorithm 2 and then present the nonergodic convergence rate analysis.

### 4.3.2. Proof of the nonergodic rate

Correspondingly, the Lyapunov function (3.24) is modified as follows

$$\begin{aligned} \mathcal{E}_k := & \mathcal{L}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}_k) + \alpha_k \|\mathbf{u}_k - \mathbf{y}^*\|^2 \\ & + \frac{\alpha_k}{2} (\|\mathbf{v}_k - \mathbf{x}^*\|^2 + \|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*\|^2), \end{aligned} \quad (4.23)$$

where  $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \in \Omega^*$  is arbitrary.

**Lemma 4.3.1.** *Let  $\{(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k)\}$  be generated by Algorithm 2, then we have the contraction*

$$\mathcal{E}_{k+1} - \mathcal{E}_k \leq -\alpha_k \mathcal{E}_{k+1} \quad \text{for all } k \in \mathbb{N}, \quad (4.24)$$

which implies  $\mathcal{E}_k \leq \mathcal{E}_0/(k+1)$  for  $k \geq 0$ .

**Proof.** From (4.21) and (4.24) it is clear that

$$\mathcal{E}_{k+1} \leq \frac{\mathcal{E}_k}{1 + \alpha_k} = \frac{\alpha_{k+1}}{\alpha_k} \mathcal{E}_k \implies \mathcal{E}_k \leq \frac{\alpha_k}{\alpha_0} \mathcal{E}_0 = \frac{\mathcal{E}_0}{k+1}.$$

Let us focus on (4.24) and calculate the difference  $\mathcal{E}_{k+1} - \mathcal{E}_k = \mathbb{I}_1 + \mathbb{I}_3 + \mathbb{I}_4 + \mathbb{I}_2$ , where

$$\begin{cases} \mathbb{I}_1 := \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \boldsymbol{\lambda}^*) - \mathcal{L}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}^*), \\ \mathbb{I}_2 := \frac{\alpha_{k+1}}{2} \|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}^*\|^2 - \frac{\alpha_k}{2} \|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*\|^2, \\ \mathbb{I}_3 := \frac{\alpha_{k+1}}{2} \|\mathbf{v}_{k+1} - \mathbf{x}^*\|^2 - \frac{\alpha_k}{2} \|\mathbf{v}_k - \mathbf{x}^*\|^2, \\ \mathbb{I}_4 := \alpha_{k+1} \|\mathbf{u}_{k+1} - \mathbf{y}^*\|^2 - \alpha_k \|\mathbf{u}_k - \mathbf{y}^*\|^2. \end{cases}$$

In what follows, we shall estimate the above four terms one by one.

Since  $\{\mathbf{y}_k\} \subset \Sigma$ , it is clear that

$$\mathbb{I}_1 = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) + \langle \boldsymbol{\lambda}^*, \mathcal{A}(\mathbf{x}_{k+1} - \mathbf{x}_k) + \mathcal{B}(\mathbf{y}_{k+1} - \mathbf{y}_k) \rangle.$$

By (4.17b) and (4.17d), we have

$$\begin{cases} \frac{\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k}{\alpha_{k+1}} = -\nabla f(\mathbf{x}_{k+1}) - \mathcal{A}^\top \tilde{\boldsymbol{\lambda}}_{k+1}, \\ \frac{\mathbf{y}_{k+1} - \tilde{\mathbf{y}}_k}{\alpha_{k+1}} \in -\mathcal{N}_\Sigma(\mathbf{y}_{k+1}) - \mathcal{B}^\top \boldsymbol{\lambda}_{k+1}, \end{cases}$$

which implies that

$$\begin{cases} f(\mathbf{x}_{k+1}) - f(\mathbf{x}) + \langle \tilde{\boldsymbol{\lambda}}_{k+1}, \mathcal{A}(\mathbf{x}_{k+1} - \mathbf{x}) \rangle \leq \left\langle \frac{\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k}{\alpha_{k+1}}, \mathbf{x} - \mathbf{x}_{k+1} \right\rangle, \\ \langle \boldsymbol{\lambda}_{k+1}, \mathcal{B}(\mathbf{y}_{k+1} - \mathbf{y}) \rangle \leq \left\langle \frac{\mathbf{y}_{k+1} - \tilde{\mathbf{y}}_k}{\alpha_{k+1}}, \mathbf{y} - \mathbf{y}_{k+1} \right\rangle, \end{cases} \quad (4.25)$$

for all  $\mathbf{x} \in \mathbb{R}^J$  and  $\mathbf{y} \in \Sigma$ . Taking  $\mathbf{x} = \mathbf{x}_k$  and  $\mathbf{y} = \mathbf{y}_k$  and observing that

$$\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k = \alpha_{k+1}(\mathbf{v}_{k+1} - \mathbf{v}_k) \quad \text{and} \quad \mathbf{y}_{k+1} - \tilde{\mathbf{y}}_k = \alpha_{k+1}(\mathbf{u}_{k+1} - \mathbf{u}_k), \quad (4.26)$$

we obtain the estimate for  $\mathbb{I}_1$ :

$$\begin{aligned}\mathbb{I}_1 &\leq \langle \mathbf{v}_{k+1} - \mathbf{v}_k, \mathbf{x}_k - \mathbf{x}_{k+1} \rangle + 2 \langle \mathbf{u}_{k+1} - \mathbf{u}_k, \mathbf{y}_k - \mathbf{y}_{k+1} \rangle \\ &\quad - \left\langle \tilde{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^*, \mathcal{A}(\mathbf{x}_{k+1} - \mathbf{x}_k) + \mathcal{B}(\mathbf{y}_{k+1} - \mathbf{y}_k) \right\rangle,\end{aligned}$$

where we used the relation  $\boldsymbol{\lambda}_{k+1} - \tilde{\boldsymbol{\lambda}}_{k+1} = \mathcal{B}(\mathbf{u}_{k+1} - \mathbf{u}_k)$  and the fact  $\mathcal{B}^\top \mathcal{B} = I$ .

Analogously to (3.29), by (4.21), it holds that

$$\begin{aligned}\mathbb{I}_2 &= \alpha_k \left\langle \mathcal{A}\mathbf{v}_{k+1} + \mathcal{B}\mathbf{u}_{k+1} - \mathbf{b}, \tilde{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^* \right\rangle - \frac{\alpha_k \alpha_{k+1}}{2} \|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}^*\|^2 \\ &\quad + \frac{\alpha_k}{2} \left( \|\boldsymbol{\lambda}_{k+1} - \tilde{\boldsymbol{\lambda}}_{k+1}\|^2 - \|\boldsymbol{\lambda}_k - \tilde{\boldsymbol{\lambda}}_{k+1}\|^2 \right).\end{aligned}$$

Similarly with (3.30), we have

$$\begin{aligned}\mathbb{I}_3 &= -\frac{\alpha_k}{2} \|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 - \frac{\alpha_k \alpha_{k+1}}{2} \|\mathbf{v}_{k+1} - \mathbf{x}^*\|^2 \\ &\quad + \alpha_k \langle \mathbf{v}_{k+1} - \mathbf{v}_k, \mathbf{v}_{k+1} - \mathbf{x}^* \rangle.\end{aligned}$$

Recalling that  $\mathbf{v}_{k+1} = \mathbf{x}_{k+1} + \alpha_k^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$  (cf.(4.17a)), we have

$$\begin{aligned}\mathbb{I}_3 &= -\frac{\alpha_k}{2} \|\mathbf{v}_{k+1} - \mathbf{v}_k\|^2 - \frac{\alpha_k \alpha_{k+1}}{2} \|\mathbf{v}_{k+1} - \mathbf{x}^*\|^2 \\ &\quad + \langle \mathbf{v}_{k+1} - \mathbf{v}_k, \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \alpha_k \langle \mathbf{v}_{k+1} - \mathbf{v}_k, \mathbf{x}_{k+1} - \mathbf{x}^* \rangle.\end{aligned}$$

The term  $\mathbb{I}_4$  can be derived similarly

$$\begin{aligned}\mathbb{I}_4 &= -\alpha_k \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2 - \alpha_k \alpha_{k+1} \|\mathbf{u}_{k+1} - \mathbf{y}^*\|^2 \\ &\quad + 2 \langle \mathbf{u}_{k+1} - \mathbf{u}_k, \mathbf{y}_{k+1} - \mathbf{y}_k \rangle + 2\alpha_k \langle \mathbf{u}_{k+1} - \mathbf{u}_k, \mathbf{y}_{k+1} - \mathbf{y}^* \rangle.\end{aligned}$$

By (4.25) and (4.26), letting  $\mathbf{x} = \mathbf{x}^*$  and  $\mathbf{y} = \mathbf{y}^*$  gives

$$\begin{aligned}&\langle \mathbf{v}_{k+1} - \mathbf{v}_k, \mathbf{x}_{k+1} - \mathbf{x}^* \rangle + 2 \langle \mathbf{u}_{k+1} - \mathbf{u}_k, \mathbf{y}_{k+1} - \mathbf{y}^* \rangle \\ &\leq \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \tilde{\boldsymbol{\lambda}}_{k+1}) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \tilde{\boldsymbol{\lambda}}_{k+1}) \\ &= \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \boldsymbol{\lambda}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}_{k+1}) - \left\langle \tilde{\boldsymbol{\lambda}}_{k+1} - \boldsymbol{\lambda}^*, \mathcal{A}\mathbf{x}_{k+1} + \mathcal{B}\mathbf{y}_{k+1} - \mathbf{b} \right\rangle.\end{aligned}$$

Combining all together yields that

$$\mathcal{E}_{k+1} - \mathcal{E}_k \leq -\alpha_k \mathcal{E}_{k+1} - \frac{\alpha_k}{2} \left( \|\boldsymbol{\lambda}_{k+1} - \tilde{\boldsymbol{\lambda}}_{k+1}\|^2 + \|\boldsymbol{\lambda}_k - \tilde{\boldsymbol{\lambda}}_{k+1}\|^2 \right).$$

This gives (4.24) and concludes the proof of this lemma.  $\square$

**Theorem 4.3.1.** *Let  $\{(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k)\}$  be generated by Algorithm 2, then for all  $k \in \mathbb{N}$ , we have  $\{\mathbf{y}_k\} \subset \Sigma$  and*

$$\begin{cases} \|\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b}\| \leq \frac{\mathcal{R}_0}{k+1}, \\ |f(\mathbf{x}_k) - f(\mathbf{x}^*)| \leq \frac{1}{k+1} (\mathcal{E}_0 + \|\boldsymbol{\lambda}^*\| \mathcal{R}_0), \end{cases} \quad (4.27a)$$

$$\text{where } \mathcal{R}_0 = \|\mathcal{A}\mathbf{x}_0 + \mathcal{B}\mathbf{y}_0 - \mathbf{b}\| + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_0\| + \sqrt{2\mathcal{E}_0}. \quad (4.27b)$$

**Proof.** In view of (4.22), it holds that

$$\boldsymbol{\lambda}_k - \alpha_k^{-1}(\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b}) = \boldsymbol{\lambda}_0 - (\mathcal{A}\mathbf{x}_0 + \mathcal{B}\mathbf{y}_0 - \mathbf{b}).$$

Thus, following the proof of (3.36), it is not hard to establish (4.27).  $\square$

**Remark 4.3.1.** *Similarly with Algorithm 1, the total cost of Algorithm 2 (and Algorithm 3) per iteration is  $\mathcal{O}(mn)$ . Therefore, according to Theorem 4.3.1, the final operation complexity is  $\mathcal{O}(mn/\epsilon)$ .*

## 4.4. Numerical Experiments

In this section, we compare Algorithm 1 (Prox-ADMM) and Algorithm 3 (Acc-ADMM) with four baseline algorithms: PADMM [99, Algorithm 3.1], ALADMM-NE [71, Algorithm 1], Sinkhorn's algorithm [2] and APDAGD [38, Algorithm 3]. We consider the discrete optimal transport (1.2) and conduct three experiments with different cost matrices: the random cost (4.28), the  $\{0, 1\}$ -valued cost (4.29) and the  $l^p$ -cost (4.30).

Note that our Acc-ADMM is parameter free. For PADMM, we choose  $\rho_0 = 2$  and  $\gamma_k = 1$ ; for ALADMM-NE we set  $\tau = 0.9$  and let it share the same penalty parameter  $\sigma$  with Prox-ADMM; for Sinkhorn's algorithm and APDAGD, we chose different regularization parameter  $\delta$ . To illustrate the convergence performances, we introduce two relative residuals:

$$\mathbf{res}_{\text{obj}}(k) := \frac{|f(\mathbf{x}_k) - f^*|}{1 + |f(\mathbf{x}_0)|} \quad \text{and} \quad \mathbf{res}_{\text{fea}}(k) := \frac{\|\mathcal{A}\mathbf{x}_k + \mathcal{B}\mathbf{y}_k - \mathbf{b}\|}{\|\mathcal{A}\mathbf{x}_0 + \mathcal{B}\mathbf{y}_0 - \mathbf{b}\|},$$

where the minimal objective value  $f^*$  is computed exactly for the  $\{0, 1\}$ -valued cost matrix and approximately via running sufficient enough iterations of Prox-ADMM for other two cases.

**Example 1: Random cost.** In the first example, we consider the random cost

$$C = (C_{ij})_{n \times n}, \quad C_{ij} \sim U([0, 1]), \quad (4.28)$$

where  $U([0, 1])$  denotes the uniform distribution on  $[0, 1]$ . Numerical outputs are reported in Fig. 4.1, which indicates that Prox-ADMM outperforms other methods but we haven't observed the linear convergence. As expected, the errors of Acc-ADMM and PADMM share the same theoretical rate  $\mathcal{O}(1/k)$  while Acc-ADMM has smaller residual of feasibility violation than PADMM. The residuals of Sinkhorn's algorithm and APDAGD decrease quickly but stay flat only with a fixed accuracy  $\mathcal{O}(\delta)$ . We are not allowed to reduce  $\delta$  as small as we can because of the round-off error and numerical instability. In addition, small regularization parameter  $\delta$  also enlarges the iteration numbers for achieving the tolerance  $\mathcal{O}(\delta)$ .

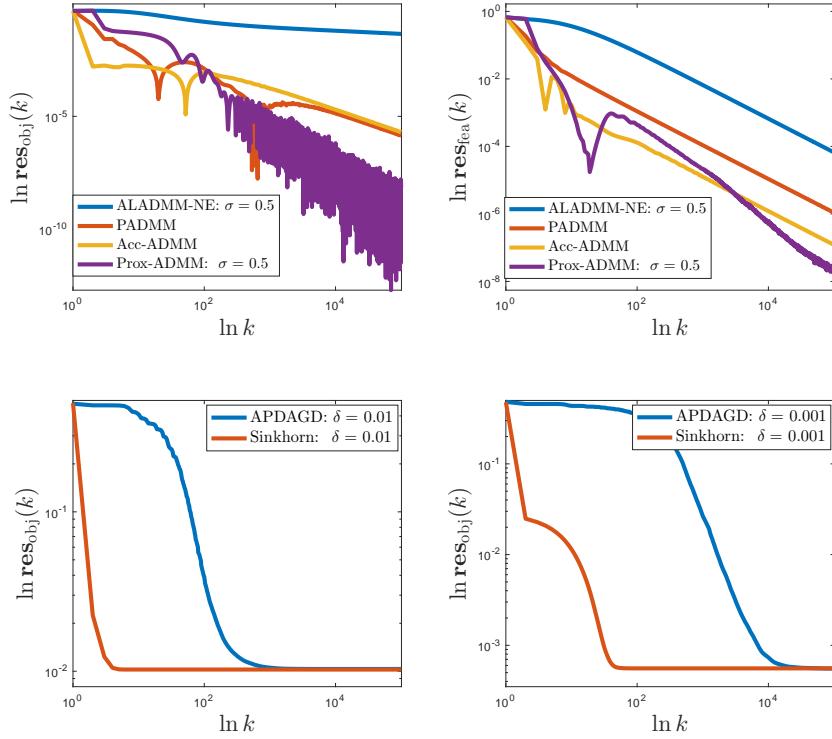


Figure 4.1: Performances of different algorithms with  $n = 500$  and the random cost (4.28).

**Example 2: The  $\{0, 1\}$ -valued cost.** We then consider a special cost

$$C = \mathbf{1}_{n \times n} - I, \quad C_{ij} = \delta_{i \neq j}. \quad (4.29)$$

According to [89, Section 2.6] or [101, Section 7.1], this corresponds to the Wasserstein distance of order 0 and the optimal transportation cost between  $\mu$  and  $\nu$  is the half of the  $l^1$ -norm of their difference, which means that  $f^* = \frac{1}{2} \|\mu - \nu\|_1$ .

From Fig. 4.2 we observe again that Prox-ADMM performs better than others and local linear convergence arises. The convergence behaviors of Acc-ADMM and PADMM are still very similar. As the same in the first example, for a given regularization parameter  $\delta$ , Sinkhorn's algorithm and APDAGD only provide solutions with fixed accuracy  $\mathcal{O}(\delta)$ , and in this example, those two methods behave very closely to each other, apart from the fact that Sinkhorn's algorithm converges more quickly in few initial steps.

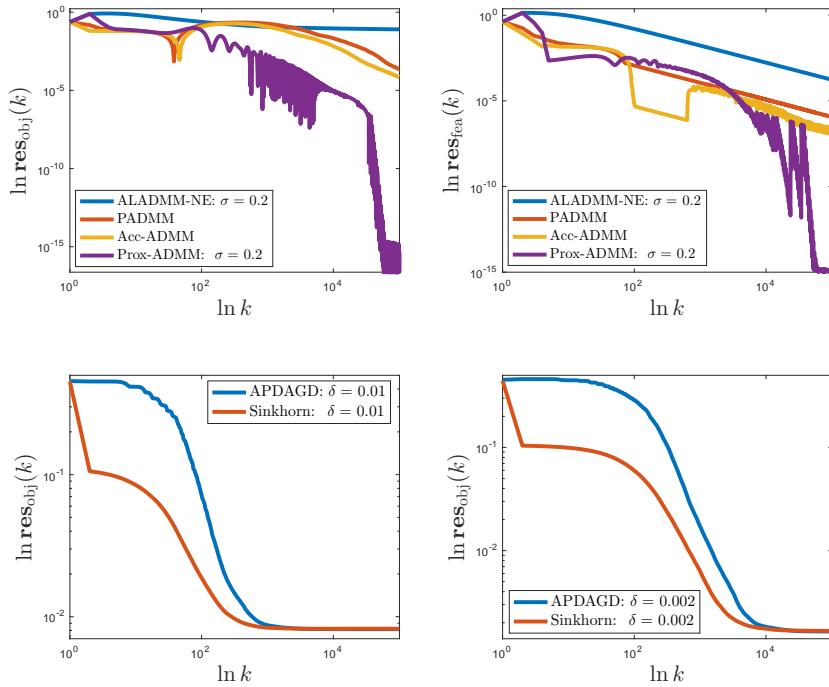


Figure 4.2: Performances of different algorithms with  $n = 400$  and the  $\{0, 1\}$ -valued cost (4.29).

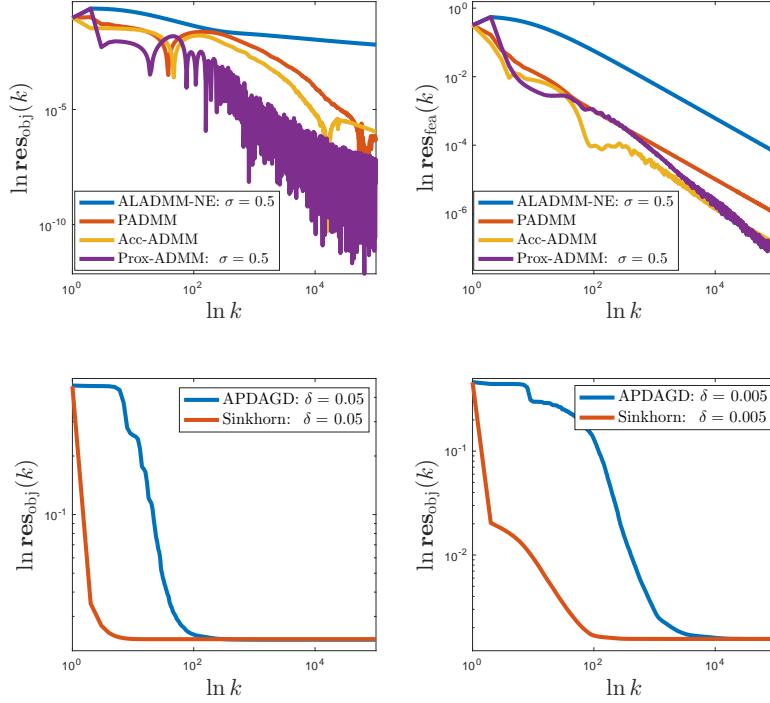


Figure 4.3:  $n = 300$ ,  $C_{ij} = \|x_i - y_j\|$ , where  $x_i$  and  $y_j$  are sampled from  $U([0, 1]^{10})$ .

**Example 3: The  $l^p$ -cost.** Let  $\mu, \nu \in \mathbb{R}_+^n$  be associated with two distributions  $\mu = \sum_{i=1}^n \mu_i \delta_{x_i}$  and  $\nu = \sum_{j=1}^n \nu_j \delta_{y_j}$ , where  $x_i$  and  $y_j$  are sampled from the uniform distribution  $U([0, 1]^m)$ . In this example, we consider the  $l^p$ -cost

$$C = (C_{ij})_{n \times n}, \quad C_{ij} = \|x_i - y_j\|^p, \quad (4.30)$$

where  $1 \leq p < \infty$ . We are interested in two cases:  $p = 1$  and  $p = 2$ , and plot the corresponding numerical results in Figs. 4.3 and 4.4.

We see that in this case, Prox-ADMM, Acc-ADMM and PADMM possess similar decay rates for the objective residual and the feasibility violation. For ALADMM-NE, the feasibility violation converges much faster than its objective residual. This agrees with the numerical results in the previous two examples. Besides, analogously to what showed in Fig. 4.1, APDAGD is not competitive with Sinkhorn's algorithm, and the total iteration number for achieving the accuracy  $\mathcal{O}(\delta)$  grows as the regularization parameter  $\delta$  decreases.

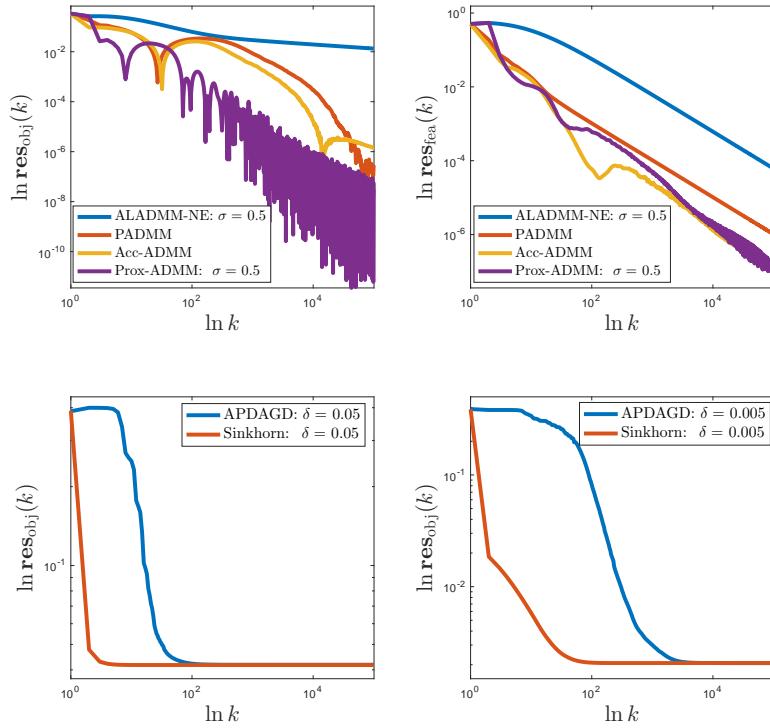


Figure 4.4:  $n = 500$ ,  $C_{ij} = \|x_i - y_j\|^2$ , where  $x_i$  and  $y_j$  are sampled from  $U([0, 1]^5)$ .

# Chapter 5

## An Inexact SsN-AMG-based Primal-Dual Algorithm

In this chapter, we apply the implicit Euler scheme (3.1) to the generalized transportation problem (1.8). For practical computations, in Section 5.1, we propose an inexact version and adopt the semi-smooth Newton method for the subproblem (3.6). Then, in Section 5.2, the linear equation for updating the Newton iteration is transformed equivalently into a graph Laplacian system, for which an efficient algebraic multigrid method shall be presented in Section 5.3. Finally, several numerical tests are reported in Section 5.4.

### 5.1. Inexact SsN-based Primal-Dual Method

For  $\lambda \in \mathbb{R}^{M+l}$  and  $\mathbf{x} = (x, y, z) \in \mathbb{R}^{mn} \times \mathbb{R}^n \times \mathbb{R}^m$ , define the Lagrangian function for (1.8):

$$\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \delta_\Sigma(\mathbf{x}) + \langle \lambda, H\mathbf{x} - b \rangle, \quad (5.1)$$

where  $H = (G, I_Y, I_Z)$ . Notice that  $\mathcal{L}(\cdot, \lambda)$  is convex and we set  $\partial_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \lambda) := \nabla f(\mathbf{x}) + \mathcal{N}_\Sigma(\mathbf{x}) + H^\top \lambda$ , where  $\nabla f(\mathbf{x}) = \sigma(x, \mathbf{0}_M) - \tilde{\mathbf{c}}$  with  $\tilde{\mathbf{c}} = (\sigma\phi - c, \mathbf{0}_M)$ .

### 5.1.1. An inexact primal-dual algorithm

Applying (3.1) to problem (1.8) gives

$$\left\{ \theta_k \frac{\lambda_{k+1} - \lambda_k}{\alpha_k} = \nabla_\lambda \mathcal{L}(\mathbf{v}_{k+1}, \lambda_{k+1}), \quad (5.2a) \right.$$

$$\left. \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\alpha_k} = \mathbf{v}_{k+1} - \mathbf{x}_{k+1}, \quad (5.2b) \right.$$

$$\left. \theta_k \frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{\alpha_k} \in -\partial_x \mathcal{L}(\mathbf{x}_{k+1}, \lambda_{k+1}), \quad (5.2c) \right.$$

where  $\alpha_k > 0$  denotes the step size and the parameter sequence  $\{\theta_k\}$  is updated by

$$\theta_{k+1} - \theta_k = -\alpha_k \theta_{k+1}, \quad \theta_0 = 1. \quad (5.3)$$

By (5.2b), we replace  $\mathbf{v}_{k+1}$  by  $\mathbf{x}_{k+1}$  and then put it into (5.2a) and (5.2c) to obtain

$$\left\{ \theta_{k+1} \lambda_{k+1} = H \mathbf{x}_{k+1} + \tilde{\lambda}_k, \quad (5.4a) \right.$$

$$\left. D_k \mathbf{x}_{k+1} \in \tilde{\mathbf{x}}_k - H^\top \lambda_{k+1} - \mathcal{N}_\Sigma(\mathbf{x}_{k+1}), \quad (5.4b) \right.$$

where  $D_k = \text{diag}(\eta_k I_{mn}, \tau_k I_n, \tau_k I_m)$  and

$$\left\{ \begin{array}{l} \tilde{\mathbf{x}}_k = \tilde{\mathbf{c}} + \theta_k (\mathbf{x}_k + \alpha_k \mathbf{v}_k) / \alpha_k^2, \\ \eta_k = \sigma + \tau_k, \tau_k = \theta_k (1 + \alpha_k) / \alpha_k^2, \\ \tilde{\lambda}_k = \theta_{k+1} [\lambda_k - \theta_k^{-1} (H \mathbf{x}_k - b)] - b. \end{array} \right.$$

From (5.4b) we have  $\mathbf{x}_{k+1} = \mathbf{proj}_\Sigma(D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda_{k+1}))$ . Plugging this into (5.4a) gives a nonlinear equation

$$F_k(\lambda_{k+1}) = 0, \quad (5.5)$$

where the mapping  $F_k : \mathbb{R}^{M+l} \rightarrow \mathbb{R}^{M+l}$  is defined by

$$F_k(\lambda) := \theta_{k+1} \lambda - H \mathbf{proj}_\Sigma(D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda)) - \tilde{\lambda}_k \quad \forall \lambda \in \mathbb{R}^{M+l}. \quad (5.6)$$

It is well-known that  $\mathbf{proj}_\Sigma$  is monotone and 1-Lipschitz continuous (cf. [7, Proposition 12.27]). Letter in Section 5.1.3, we will see that (5.5) is nothing but the Euler equation for minimizing a smooth and strongly convex objective

(see (5.20)). Therefore it admits a unique solution which is denoted by  $\lambda_{k+1}^\#$  (in stead of  $\lambda_{k+1}$ ), and we obtain

$$\mathbf{x}_{k+1}^\# = \mathbf{proj}_\Sigma \left( D_k^{-1} (\tilde{\mathbf{x}}_k - H^\top \lambda_{k+1}^\#) \right), \quad \mathbf{v}_{k+1}^\# = \mathbf{x}_{k+1}^\# + \frac{\mathbf{x}_{k+1}^\# - \mathbf{x}_k}{\alpha_k}. \quad (5.7)$$

This means  $(\mathbf{x}_{k+1}^\#, \mathbf{v}_{k+1}^\#, \lambda_{k+1}^\#)$  is the exact solution to the implicit scheme (5.2a) at the  $k$ -th step.

In practical computations, however, the inner problem (5.5) is often solved approximately. Below, we summarize our inexact primal-dual method in Algorithm 4. Then in Sections 5.1.2 and 5.1.3, we will present the convergence analysis and apply the semi-smooth Newton iteration (Algorithm 5) to solve the nonlinear equation (5.5).

According to Theorem 5.1.1, the convergence rate is related to the step size  $\alpha_k$  and we provide detailed discussions in Remarks 5.1.1 and 5.1.2. In addition, the stop criterion in step 6 of Algorithm 4 is convenient for the upcoming convergence rate proof but not practical as it requires the true solution  $\lambda_{k+1}^\#$ . In numerical experiments, we focus on the quantity  $\|F_k(\lambda_{k+1})\|$ , which provides a computable posterior indicator.

---

**Algorithm 4** Inexact Primal Dual Method for (1.8)
 

---

**Input:**  $\theta_0 = 1$ ,  $\mathbf{v}_0, \mathbf{x}_0 \in \mathbb{R}^{mn} \times \mathbb{R}^n \times \mathbb{R}^m$  and  $\lambda_0 \in \mathbb{R}^{M+l}$ .

- 1: **for**  $k = 0, 1, \dots$  **do**
- 2:     Choose the step size  $\alpha_k > 0$  and the tolerance  $\epsilon_k > 0$ .
- 3:     Set  $\tau_k = \theta_k(1 + \alpha_k)/\alpha_k^2$  and  $\eta_k = \sigma + \tau_k$ .
- 4:     Set  $D_k = \text{diag}(\eta_k I_{mn}, \tau_k I_n, \tau_k I_m)$  and  $\tilde{\mathbf{x}}_k = \tilde{\mathbf{c}} + \theta_k(\mathbf{x}_k + \alpha_k \mathbf{v}_k)/\alpha_k^2$ .
- 5:     Update  $\theta_{k+1} = \theta_k/(1 + \alpha_k)$  and set  $\tilde{\lambda}_k = \theta_{k+1} [\lambda_k - \theta_k^{-1}(H\mathbf{x}_k - b)] - b$ .
- 6:     Apply Algorithm 5 to (5.5) to obtain  $\lambda_{k+1}$  such that  $\|\lambda_{k+1} - \lambda_{k+1}^\#\| \leq \epsilon_k$ .
- 7:     Update  $\mathbf{x}_{k+1} = \mathbf{proj}_\Sigma (D_k^{-1} (\tilde{\mathbf{x}}_k - H^\top \lambda_{k+1}))$ .
- 8:     Update  $\mathbf{v}_{k+1} = \mathbf{x}_{k+1} + (\mathbf{x}_{k+1} - \mathbf{x}_k)/\alpha_k$ .
- 9: **end for**

---

### 5.1.2. Rate of convergence

Let  $\{(\theta_k, \mathbf{x}_k, \mathbf{v}_k, \lambda_k)\}$  be generated by Algorithm 4 with  $\{\alpha_k\}$  and  $\{\epsilon_k\}$ . It is clear that  $\{\mathbf{x}_k\} \subset \Sigma$ . Following (3.7), define a discrete Lyapunov function

$$\mathcal{E}(\theta_k, \mathbf{x}_k, \mathbf{v}_k, \lambda_k) := \mathcal{L}(\mathbf{x}_k, \lambda^*) - \mathcal{L}(\mathbf{x}^*, \lambda_k) + \frac{\theta_k}{2} \left( \|\mathbf{v}_k - \mathbf{x}^*\|^2 + \|\lambda_k - \lambda^*\|^2 \right), \quad (5.8)$$

and for simplicity, we write  $\mathcal{E}_k = \mathcal{E}(\theta_k, \mathbf{x}_k, \mathbf{v}_k, \lambda_k)$ .

According to Theorem 3.1.1, we have

$$\mathcal{E}_{k+1}^\# - \mathcal{E}_k \leq -\alpha_k \mathcal{E}_{k+1}^\#, \quad (5.9)$$

where  $\mathcal{E}_{k+1}^\# := \mathcal{E}(\theta_{k+1}, \mathbf{x}_{k+1}^\#, \mathbf{v}_{k+1}^\#, \lambda_{k+1}^\#)$  and  $(\mathbf{x}_{k+1}^\#, \mathbf{v}_{k+1}^\#, \lambda_{k+1}^\#)$  is the exact solution to the implicit Euler discretization (5.2a) at the  $k$ -th iteration. This also implies that

$$\mathcal{E}_{k+1} - \frac{\mathcal{E}_k}{1 + \alpha_k} = \mathcal{E}_{k+1} - \mathcal{E}_{k+1}^\# + \mathcal{E}_{k+1}^\# - \frac{\mathcal{E}_k}{1 + \alpha_k} \leq \mathcal{E}_{k+1} - \mathcal{E}_{k+1}^\#, \quad (5.10)$$

which leads to the following one-iteration estimate.

**Lemma 5.1.1.** *Let  $(\theta_{k+1}, \mathbf{x}_{k+1}, \mathbf{v}_k, \lambda_{k+1})$  be the output of the  $k$ -th iteration of Algorithm 4 with  $(\theta_k, \mathbf{x}_k, \lambda_k)$  and  $(\alpha_k, \epsilon_k)$ . Then we have*

$$\begin{aligned} \mathcal{E}_{k+1} &\leq \frac{\mathcal{E}_k}{1 + \alpha_k} + \epsilon_k \theta_k^{-1} \alpha_k^2 \|H\| (\sigma \|\mathbf{x}^*\| + \|H^\top \lambda^* - \tilde{\mathbf{c}}\|) + \epsilon_k \theta_k \|\lambda_{k+1} - \lambda^*\| \\ &\quad + \epsilon_k \theta_k^{-1} (\sigma \alpha_k^2 \|H\| \|\mathbf{x}_{k+1} - \mathbf{x}^*\| + \alpha_k \theta_k \|H\| \|\mathbf{v}_{k+1} - \mathbf{x}^*\|). \end{aligned} \quad (5.11)$$

**Proof.** Thanks to (5.10), it is sufficient to focus on the difference

$$\begin{aligned} &\mathcal{E}_{k+1} - \mathcal{E}_{k+1}^\# \\ &= \mathcal{L}(\mathbf{x}_{k+1}, \lambda^*) - \mathcal{L}(\mathbf{x}_{k+1}^\#, \lambda^*) + \frac{\theta_{k+1}}{2} (\|\mathbf{v}_{k+1} - \mathbf{x}^*\|^2 - \|\mathbf{v}_{k+1}^\# - \mathbf{x}^*\|^2) \\ &\quad + \frac{\theta_{k+1}}{2} (\|\lambda_{k+1} - \lambda^*\|^2 - \|\lambda_{k+1}^\# - \lambda^*\|^2). \end{aligned} \quad (5.12)$$

Since  $\mathbf{proj}_\Sigma$  is 1-Lipschitz continuous and

$$\mathbf{x}_{k+1} = \mathbf{proj}_\Sigma (D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda_{k+1})), \quad \mathbf{x}_{k+1}^\# = \mathbf{proj}_\Sigma (D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda_{k+1}^\#)),$$

it follows from the fact  $\|D_k^{-1}\| \leq 1/\tau_k$  that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\#\| \leq \frac{1}{\tau_k} \|H^\top(\lambda_{k+1} - \lambda_{k+1}^\#)\| \leq \frac{\epsilon_k \theta_k^{-1} \alpha_k^2}{1 + \alpha_k} \|H\|. \quad (5.13)$$

By (5.7), we have  $\mathbf{v}_{k+1}^\# = \mathbf{x}_{k+1}^\# + (\mathbf{x}_{k+1}^\# - \mathbf{x}_k)/\alpha_k$ , which together with the update for  $\mathbf{v}_{k+1}$  in Algorithm 4 yields the identity  $\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^\# = (1 + 1/\alpha_k)(\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\#)$ . Hence it holds that

$$\|\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^\#\| \leq \frac{\alpha_k + 1}{\alpha_k} \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\#\| \leq \epsilon_k \theta_k^{-1} \alpha_k \|H\|,$$

which gives

$$\begin{aligned} & \|\mathbf{v}_{k+1} - \mathbf{x}^*\|^2 - \|\mathbf{v}_{k+1}^\# - \mathbf{x}^*\|^2 \\ &= 2\langle \mathbf{v}_{k+1} - \mathbf{x}^*, \mathbf{v}_{k+1} - \mathbf{v}_{k+1}^\# \rangle - \|\mathbf{v}_{k+1} - \mathbf{v}_{k+1}^\#\|^2 \leq 2\epsilon_k \theta_k^{-1} \alpha_k \|H\| \|\mathbf{v}_{k+1} - \mathbf{x}^*\|. \end{aligned}$$

Similarly, we have

$$\|\lambda_{k+1} - \lambda^*\|^2 - \|\lambda_{k+1}^\# - \lambda^*\|^2 \leq 2\epsilon_k \|\lambda_{k+1} - \lambda^*\|.$$

Plugging the above two estimates into (5.12) implies

$$\begin{aligned} \mathcal{E}_{k+1} - \mathcal{E}_{k+1}^\# &\leq \mathcal{L}(\mathbf{x}_{k+1}, \lambda^*) - \mathcal{L}(\mathbf{x}_{k+1}^\#, \lambda^*) \\ &\quad + \epsilon_k \alpha_k \|H\| \|\mathbf{v}_{k+1} - \mathbf{x}^*\| + \epsilon_k \theta_k \|\lambda_{k+1} - \lambda^*\|, \end{aligned} \quad (5.14)$$

where we used the relation  $\theta_{k+1} \leq \theta_k$ .

To the end, let us estimate the first difference term in (5.14) as follows. It is clear that

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_{k+1}^\#) = \sigma \langle \mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\#, \mathbf{x}_{k+1} \rangle - \langle \tilde{\mathbf{c}}, \mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\# \rangle - \frac{\sigma}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\#\|^2.$$

Invoking (5.13) and the fact  $\mathbf{x}_{k+1}^\#, \mathbf{x}_{k+1} \in \Sigma$ , we find

$$\begin{aligned} \mathcal{L}(\mathbf{x}_{k+1}, \lambda^*) - \mathcal{L}(\mathbf{x}_{k+1}^\#, \lambda^*) &= f(\mathbf{x}_{k+1}) - f(\mathbf{x}_{k+1}^\#) + \langle H^\top \lambda^*, \mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\# \rangle \\ &\leq (\sigma \|\mathbf{x}_{k+1}\| + \|H^\top \lambda^* - \tilde{\mathbf{c}}\|) \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^\#\| \\ &\leq \epsilon_k \theta_k^{-1} \alpha_k^2 \|H\| (\sigma \|\mathbf{x}_{k+1}\| + \|H^\top \lambda^* - \tilde{\mathbf{c}}\|). \end{aligned}$$

Combining this with (5.14) and the triangle inequality  $\|\mathbf{x}_{k+1}\| \leq \|\mathbf{x}^*\| + \|\mathbf{x}_{k+1} - \mathbf{x}^*\|$ , we obtain (5.11) and complete the proof.  $\square$

To derive the concrete convergence rate of Algorithm 4, let us introduce

$$\widehat{\varepsilon}_k := \sum_{i=0}^{k-1} \epsilon_i \alpha_i^2 \theta_i^{-2} \quad \text{and} \quad \widetilde{\varepsilon}_k := \sum_{i=0}^{k-1} \epsilon_i \theta_i^{-3/2} (\alpha_i^2 + \alpha_i \sqrt{\theta_i} + \theta_i^{3/2}), \quad k \geq 1,$$

and for  $k = 0$ , set  $\widehat{\varepsilon}_0 = \widetilde{\varepsilon}_0 = 0$ .

**Theorem 5.1.1.** *Let  $\{\mathbf{x}_k\} = \{(x_k, y_k, z_k)\} \subset \Sigma$  and  $\{\lambda_k\}$  be generated by Algorithm 4 with arbitrary step size sequence  $\{\alpha_k\}$  and tolerance sequence  $\{\epsilon_k\}$ . Then for all  $k \in \mathbb{N}$ , there holds that*

$$\begin{aligned} & \mathcal{L}(\mathbf{x}_k, \lambda^*) - \mathcal{L}(\mathbf{x}^*, \lambda_k) + \|H\mathbf{x}_k - b\| + |f(\mathbf{x}_k) - f(\mathbf{x}^*)| \\ & \leq \left( C_1(\sqrt{\widehat{\varepsilon}_k}) + C_2(\widetilde{\varepsilon}_k) \right) \times \prod_{i=0}^{k-1} \frac{1}{1 + \alpha_i}, \end{aligned} \quad (5.15)$$

where both  $C_1(\cdot)$  and  $C_2(\cdot)$  are quadratic functions.

**Proof.** Based on (5.11) and the proof of [79, Lemma 3.3], we are ready to establish

$$\mathcal{L}(\mathbf{x}_k, \lambda^*) - \mathcal{L}(\mathbf{x}^*, \lambda_k) \leq \mathcal{E}_k \leq \theta_k \left( \sqrt{\mathcal{E}_0 + Z\widehat{\varepsilon}_k} + \sqrt{2}Q\widetilde{\varepsilon}_k \right)^2, \quad (5.16)$$

where  $Z := \|H\| (\sigma \|\mathbf{x}^*\| + \|H^\top \lambda^* - \tilde{c}\|)$  and  $Q := 1 + (1 + \sqrt{\sigma}) \|H\|$ . Since  $\mathbf{x}_k \in \Sigma$ , we have

$$0 \leq f(\mathbf{x}_k) - f(\mathbf{x}^*) + \langle \lambda^*, H\mathbf{x}_k - b \rangle = \mathcal{L}(\mathbf{x}_k, \lambda^*) - \mathcal{L}(\mathbf{x}^*, \lambda_k) \leq \mathcal{E}_k \leq \theta_k R_k^2,$$

where  $R_k := \sqrt{\mathcal{E}_0 + Z\widehat{\varepsilon}_k} + \sqrt{2}Q\widetilde{\varepsilon}_k$ , and it follows immediately that

$$|f(\mathbf{x}_k) - f(\mathbf{x}^*)| \leq \theta_k R_k^2 + \|\lambda^*\| \|H\mathbf{x}_k - b\|, \quad (5.17)$$

Below, we aim to prove

$$\|H\mathbf{x}_k - b\| \leq \theta_k \left( \|H\mathbf{x}_0 - b\| + \|\lambda_0 + \lambda^*\| + \sqrt{2}R_k + \|H\|^2 \widehat{\varepsilon}_k + \widetilde{\varepsilon}_k \right), \quad (5.18)$$

which together with (5.16) and (5.17) proves (5.15). Note that  $(\mathbf{x}_{k+1}^\#, \mathbf{v}_{k+1}^\#, \lambda_{k+1}^\#)$  is the exact solution to the implicit Euler discretization (5.2a) at the  $k$ -th iteration and by (5.2a) we have

$$\lambda_{k+1}^\# - \lambda_k = \alpha_k / \theta_k (H\mathbf{v}_{k+1}^\# - b) = \theta_{k+1}^{-1} (H\mathbf{x}_{k+1}^\# - b) - \theta_k^{-1} (H\mathbf{x}_k - b).$$

Therefore, a rearrangement gives

$$\lambda_{k+1} - \lambda_k = E_k + \theta_{k+1}^{-1} (H\mathbf{x}_{k+1} - b) - \theta_k^{-1} (H\mathbf{x}_k - b),$$

where  $E_k := \lambda_{k+1} - \lambda_{k+1}^\# + \theta_{k+1}^{-1} H(\mathbf{x}_{k+1}^\# - \mathbf{x}_{k+1})$ . This also leads to

$$\lambda_k - \lambda_0 = \theta_k^{-1} (H\mathbf{x}_k - b) - (H\mathbf{x}_0 - b) + \sum_{i=0}^{k-1} E_i,$$

and we get

$$\|H\mathbf{x}_k - b\| \leq \theta_k \left( \|H\mathbf{x}_0 - b\| + \|\lambda_k - \lambda_0\| + \sum_{i=0}^{k-1} \|E_i\| \right).$$

Invoking (5.13) implies

$$\|E_k\| \leq \|\lambda_{k+1} - \lambda_{k+1}^\#\| + \theta_{k+1}^{-1} \|H\| \|\mathbf{x}_{k+1}^\# - \mathbf{x}_{k+1}\| \leq \epsilon_k (1 + \alpha_k^2 \theta_k^{-2} \|H\|^2),$$

and using the estimate (5.16) promises that  $\|\lambda_k - \lambda^*\| \leq \sqrt{2}R_k$ . Consequently, we obtain (5.18) and finish the proof of this theorem.  $\square$

According to (5.15), the final rate is obtained as long as the step size  $\alpha_k$  and the error  $\epsilon_k$  are specified. Two examples are given in order.

**Remark 5.1.1.** Consider non-vanishing step size  $\alpha_k \geq \hat{\alpha} > 0$ . If  $\epsilon_k = \mathcal{O}(\theta_k^{3/2}/(k+1)^p)$  with  $p > 1$ , then  $\tilde{\varepsilon}_k < \infty$  and  $\sqrt{\theta_k} \hat{\varepsilon}_k < \infty$ . By (5.15) and the fact that both  $C_1(\cdot)$  and  $C_2(\cdot)$  are quadratic functions, we obtain the final rate

$$\theta_k \left( C_1(\sqrt{\hat{\varepsilon}_k}) + C_2(\tilde{\varepsilon}_k) \right) \leq C_3(\alpha_k^{\max}) \sqrt{\theta_k},$$

where  $\alpha_k^{\max} = \max_{0 \leq i \leq k-1} \{\alpha_i\}$  and  $C_3(\cdot)$  is a quartic function. Therefore, we have at least linear rate since  $\theta_k \leq (1 + \hat{\alpha})^{-k}$ , and superlinear convergence follows provided that  $\alpha_k \rightarrow \infty$ .

**Remark 5.1.2.** We then consider vanishing step size  $\alpha_k \rightarrow 0$ . In particular, assume  $\alpha_k^2 = (k+1)^p \theta_k^3 \theta_{k+1}^{-2}$  with  $p > 0$ , then an elementary calculation yields that  $\theta_k = \mathcal{O}(1/(k+1)^{2+p})$  and  $\alpha_k = \mathcal{O}(1/(k+1))$ . Hence, if  $\epsilon_k = \mathcal{O}(1/(k+1)^q)$  with  $q > 3 + 2p$ , then  $\hat{\varepsilon}_k + \tilde{\varepsilon}_k < \infty$  and we have the sublinear rate

$$\theta_k \left( C_1(\sqrt{\hat{\varepsilon}_k}) + C_2(\tilde{\varepsilon}_k) \right) = \mathcal{O}(1/(k+1)^{2+p}), \text{ with any } p > 0.$$

### 5.1.3. An SsN method for the subproblem (5.5)

For  $\delta_\Sigma(\mathbf{x}) = \delta_{\mathcal{X}}(x) + \delta_{\mathcal{Y}}(y) + \delta_{\mathcal{Z}}(z)$ , define its Moreau–Yosida approximation

$$[\delta_\Sigma](\mathbf{x}) := \min_{\mathbf{v} \in \Sigma} \frac{1}{2} \|\mathbf{v} - \mathbf{x}\|_{D_k}^2 \quad \forall \mathbf{x} \in \mathbb{R}^{mn} \times \mathbb{R}^n \times \mathbb{R}^m, \quad (5.19)$$

and introduce  $\mathcal{F}_k : \mathbb{R}^{M+l} \rightarrow \mathbb{R}$  by that

$$\mathcal{F}_k(\lambda) := \frac{\theta_{k+1}}{2} \|\lambda\|^2 - \langle \tilde{\lambda}_k, \lambda \rangle + \frac{1}{2} \|\tilde{\mathbf{x}}_k - H^\top \lambda\|_{D_k^{-1}}^2 - [\delta_\Sigma](D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda)). \quad (5.20)$$

Note that  $\mathcal{F}_k$  is strongly convex and continuous differentiable with  $\nabla \mathcal{F}_k = F_k$ , where  $F_k$  has been defined in (5.6). Indeed, according to [7, Proposition 12.29],  $[\delta_\Sigma](\cdot)$  is continuous differentiable and  $\nabla [\delta_\Sigma](\mathbf{x}) = D_k(\mathbf{x} - \mathbf{proj}_\Sigma(\mathbf{x}))$ . Moreover, by Moreau’s decomposition [7, Theorem 14.3 (ii)]

$$\mathbf{x} = \mathbf{proj}_\Sigma(\mathbf{x}) + D_k^{-1} \mathbf{prox}_{D_k \delta_\Sigma^*}(D_k \mathbf{x}),$$

we also find that

$$\begin{aligned} \mathcal{F}_k(\lambda) &= \frac{\theta_{k+1}}{2} \|\lambda\|^2 - \langle \tilde{\lambda}_k, \lambda \rangle + \delta_\Sigma^* \left( \mathbf{prox}_{D_k \delta_\Sigma^*}(\tilde{\mathbf{x}}_k - H^\top \lambda) \right) \\ &\quad + \frac{1}{2} \|\mathbf{proj}_\Sigma(D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda))\|_{D_k}^2, \end{aligned} \quad (5.21)$$

where  $\delta_\Sigma^*(\mathbf{x})$  is the conjugate function of  $\delta_\Sigma(\mathbf{x})$  and  $D_k \delta_\Sigma^*$  is understood as  $\eta_k \delta_{\mathcal{X}}^* + \tau_k (\delta_{\mathcal{Y}}^* + \delta_{\mathcal{Z}}^*)$ .

As  $\Sigma = \{\mathbf{x} \in \mathbb{R}^J : \sigma_{1,i} \leq x_i \leq \sigma_{2,i}\}$  is a box region,  $\mathbf{proj}_\Sigma$  is piecewise affine and strongly semismooth (cf. [40, Propositions 4.1.4 and 7.4.7]), and so is  $F_k$  (see [40, Proposition 7.4.4]). Denote by  $\partial \mathbf{proj}_\Sigma(\mathbf{x})$  the Clarke subdifferential [31, Definition 2.6.1] of the proximal mapping  $\mathbf{proj}_\Sigma$  at  $\mathbf{x}$ . Thanks to [76, Table 3], we have

$$\partial \mathbf{proj}_\Sigma(\mathbf{x}) := \left\{ \text{diag}(\chi) : \chi_i \in \begin{cases} \{1\} & \text{if } \sigma_{1,i} < x_i < \sigma_{2,i} \\ [0, 1] & \text{if } x_i \in \{\sigma_{1,i}, \sigma_{2,i}\} \text{ and } \sigma_{1,i} \neq \sigma_{2,i} \\ \{0\} & \text{if } x_i \leq \sigma_{1,i} \text{ or } x_i \geq \sigma_{2,i} \end{cases} \right\}. \quad (5.22)$$

For every  $\lambda \in \mathbb{R}^{M+l}$ , let  $U_k(\lambda) \in \partial \mathbf{proj}_\Sigma(D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda))$  and define an SPD matrix

$$\mathcal{J}_k(\lambda) := \theta_{k+1} I + H D_k^{-1} U_k(\lambda) H^\top. \quad (5.23)$$

Then the semi-smooth Newton (SsN) iteration for (5.5) reads as follows

$$\begin{cases} \mathcal{J}_k(\lambda^j)\xi = -F_k(\lambda^j), \\ \lambda^{j+1} = \lambda^j + \xi. \end{cases} \quad (5.24a)$$

$$(5.24b)$$

Since  $F_k$  is strongly semismooth, there has local quadratic convergence [90, 91]. Below, we summarize the SsN method in Algorithm 5, where a line search procedure [36] is supplemented for global convergence.

---

**Algorithm 5** SsN method for (5.5)
 

---

**Input:**  $\tau \in (0, 1/2)$ ,  $\delta \in (0, 1)$  and  $\lambda \in \mathbb{R}^{M+l}$ .

- 1: **for**  $j = 0, 1, \dots$  **do**
- 2:   Set  $\lambda_{\text{old}} = \lambda$  and  $\mathbf{z}_k = D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda)$ .
- 3:   Compute  $U_k(\lambda) \in \partial \mathbf{proj}_\Sigma(\mathbf{z}_k)$  by (5.22).
- 4:   Solve the linear SPD system  $\mathcal{J}_k(\lambda)\xi = -F_k(\lambda)$ .
- 5:   Find the smallest  $\ell \in \mathbb{N}$  such that  $\mathcal{F}_k(\lambda_{\text{old}} + \delta^\ell \xi) \leq \mathcal{F}_k(\lambda_{\text{old}}) + \tau \delta^\ell \langle F_k(\lambda_{\text{old}}), \xi \rangle$ .
- 6:   Update  $\lambda = \lambda_{\text{old}} + \delta^\ell \xi$ .
- 7: **end for**

---

To update the SsN iteration, we have to solve a linear SPD system in (5.24a). In Section 5.2, we shall explore its hidden graph structure and obtain an equivalent graph Laplacian system, for which an efficient and robust algebraic multigrid method will be proposed in Section 5.3.

## 5.2. An Equivalent Graph Laplacian System

### 5.2.1. The reduced problem

Recall that  $H = (G, I_Y, I_Z)$ , where  $G$ ,  $I_Y$  and  $I_Z$  are defined in (1.9).

Let us rewrite (5.24a) in a generic form

$$\mathcal{H}\xi = (\epsilon I + \mathcal{H}_0)\xi = z, \quad (5.25)$$

where

$$\mathcal{H}_0 = \begin{pmatrix} \text{diag}(t) + T \text{diag}(s) T^\top & T \text{diag}(s) \Pi^\top \\ \Pi \text{diag}(s) T^\top & \Pi \text{diag}(s) \Pi^\top \end{pmatrix}, \quad (5.26)$$

with  $s \in \mathbb{R}_+^{mn}$  and  $t \in \mathbb{R}_+^M$ . Let  $S = \text{diag}(s)$  and  $K = \text{diag}(t)$  and write  $\xi = (\xi_1, \xi_2)$  and  $z = (z_1, z_2)$ , then (5.25) is equivalent to

$$\begin{cases} (\epsilon I + K + TST^\top)\xi_1 + TS\Pi^\top\xi_2 = z_1, \\ \Pi ST^\top\xi_1 + (\epsilon I + \Pi S\Pi^\top)\xi_2 = z_2. \end{cases}$$

Additionally, this gives

$$\begin{cases} \xi_1 = \left(\mathcal{T} - \Psi\tilde{\Pi}^{-1}\Psi^\top\right)^{-1}\left(z_1 - \Psi\tilde{\Pi}^{-1}z_2\right), \\ \xi_2 = \tilde{\Pi}^{-1}\left(z_2 - \Psi^\top\xi_1\right), \end{cases}$$

where  $\mathcal{T} = \epsilon I + K + TST^\top \in \mathbb{R}^{M \times M}$ ,  $\tilde{\Pi} = \epsilon I + \Pi S\Pi^\top \in \mathbb{R}^{l \times l}$  and  $\Psi = TS\Pi^\top \in \mathbb{R}^{M \times l}$ .

Assume  $l$  is small, then  $\tilde{\Pi}$  is easy to invert. This is true for all transport-like problems listed in Section 1.1.3. Indeed, for partial optimal transport (1.6),  $\tilde{\Pi}$  is a constant ( $l = 1$ ) and for other problems,  $\tilde{\Pi}$  is just a vacuum ( $l = 0$ ). Moreover, thanks to Sherman–Woodbury formula, we have

$$\left(\mathcal{T} - \Psi\tilde{\Pi}^{-1}\Psi^\top\right)^{-1} = \mathcal{T}^{-1} + \mathcal{T}^{-1}\Psi\left(\tilde{\Pi} - \Psi^\top\mathcal{T}^{-1}\Psi\right)^{-1}\Psi^\top\mathcal{T}^{-1}.$$

Since  $\tilde{\Pi} - \Psi^\top\mathcal{T}^{-1}\Psi \in \mathbb{R}^{l \times l}$  is invertible with small size, what we shall pay attention to is the inverse of  $\mathcal{T}$ , which corresponds to the reduced linear system

$$\mathcal{T}\xi = (\epsilon I + K + TST^\top)\xi = z. \quad (5.27)$$

### 5.2.2. An equivalent graph Laplacian

Let  $Y \in \mathbb{R}^{m \times n}$  be such that  $\text{vec}(Y) = s$ , then a direct computation yields that

$$\mathcal{T}_0 := TST^\top = \begin{pmatrix} \text{diag}(Y^\top \mathbf{1}_m) & Y^\top \\ Y & \text{diag}(Y\mathbf{1}_n) \end{pmatrix}. \quad (5.28)$$

Besides, set  $\mathcal{Q} = \text{diag}(I_n, -I_m)$  and define  $\mathcal{A}_0 := \mathcal{Q}\mathcal{T}_0\mathcal{Q} \in \mathbb{R}^{M \times M}$ , then  $\mathcal{T}_0$  is spectrally equivalent to  $\mathcal{A}_0$  and a direct calculation gives

$$\mathcal{A}_0 = \begin{pmatrix} \text{diag}(Y^\top \mathbf{1}_m) & -Y^\top \\ -Y & \text{diag}(Y\mathbf{1}_n) \end{pmatrix}.$$

Note that  $\mathcal{A}_0$  is the Laplacian matrix of the bipartite graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ , where  $w = \text{vec}(Y)$ ,  $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$  with  $\mathcal{V}_1 = \{1, 2, \dots, n\}$  and  $\mathcal{V}_2 = m + \mathcal{V}_1$ , and  $\mathcal{E} = \{e = \{i, j\} : w_{(i-1)m+j-n} > 0, i \in \mathcal{V}_1, j \in \mathcal{V}_2\}$ . Consequently the reduced linear system (5.27) now is equivalent to

$$\mathcal{A}u = (\epsilon I + K + \mathcal{A}_0)u = \mathcal{Q}z, \quad (5.29)$$

where  $K = \text{diag}(t)$  is diagonal with nonnegative components  $t \in \mathbb{R}_+^M$ . Clearly, if  $u$  solves (5.29), then the solution to (5.27) is given by  $\xi = \mathcal{Q}u$ .

**Remark 5.2.1.** Recall that  $\text{vec}(Y) = s \in \mathbb{R}_+^{mn}$ . In view of (5.23), we claim that the sparsity pattern of  $s$  (and thus  $Y$ ) is related to that of  $\{x_k\}_{k \in \mathbb{N}}$ .

Indeed, since  $\mathbf{x}_{k+1} = \mathbf{proj}_\Sigma(D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda_{k+1}))$ , we have

$$x_{k+1} = \mathbf{proj}_{\mathcal{X}}(\eta_k^{-1}(\tilde{x}_k - G^\top \lambda_{k+1})),$$

where  $\tilde{x}_k$  is the component of  $\tilde{\mathbf{x}}_k$  in  $\mathcal{X}$ . Since  $s \in \partial \mathbf{proj}_{\mathcal{X}}(\eta_k^{-1}(\tilde{x}_k - G^\top \lambda_{k+1}))$ , by (5.4a) and (5.22), we conclude that  $s$  is very close to the sparsity pattern of  $x_{k+1}$ . Moreover, as  $x_k$  converges to an optimal transport plan, saying  $X^*$ , the sparsity pattern of  $Y$  agrees with that of  $X^*$ .

### 5.2.3. A hybrid framework

We now discuss how to solve the linear SPD system (5.29). If the bipartite graph  $\mathcal{G}$  of  $\mathcal{A}_0$  has  $\kappa$  connected components, then there is a permutation matrix  $\mathcal{P}$  such that

$$\mathcal{P}^\top \mathcal{A}_0 \mathcal{P} = \text{diag}(A_0^1, A_0^2, \dots, A_0^\kappa), \quad (5.30)$$

where each  $A_0^i (1 \leq i \leq \kappa)$  corresponds to the Laplacian matrix of some connected bipartite graph. Since  $\epsilon I + K$  is diagonal, we are allowed to solve  $\kappa$  independent linear systems, each of which takes the form

$$Au = (\epsilon I + \Lambda + A_0)u = d, \quad (5.31)$$

where  $\Lambda$  is diagonal with nonnegative components and  $A_0 \in \mathbb{R}^{N \times N}$  is a connected graph Laplacian, with explicit null space:  $\text{span}\{\mathbf{1}_N\}$ . Note that if

the diagonal part of  $A_0$  has zero component, then it can be further reduced. Thus, without loss of generality, in what follows, assume all diagonal elements of  $A_0$  are positive, which means  $A_0$  has no zero row or column since  $A_0\mathbf{1}_N = 0$ .

Recall that the size of the linear system (5.29) is  $M$ -by- $M$ . Therefore, if  $N \leq M^{1/3}$ , then the solution to (5.31) can be obtained via direct method within  $\mathcal{O}(N^3) \leq \mathcal{O}(M)$  complexity. Otherwise, we shall consider iterative methods. This leads to a hybrid approach, as summarized in Algorithm 6.

---

**Algorithm 6** A Hybrid Solver for (5.27):  $\mathcal{T}\xi = z$ 


---

- 1: Set  $\mathcal{Q} = \text{diag}(I_n, -I_m)$ ,  $\mathcal{A}_0 = \mathcal{Q}\mathcal{T}_0\mathcal{Q}$  and  $\mathcal{A} = \epsilon I + K + \mathcal{A}_0$ .
- 2: Check the connected components of  $\mathcal{A}_0$  and find a permutation matrix  $\mathcal{P}$  such that

$$\mathcal{P}^\top \mathcal{A} \mathcal{P} = \text{diag}(A_1, A_2, \dots, A_\kappa) \quad \text{and} \quad d = \mathcal{P}^\top \mathcal{Q} z = (d_1, d_2, \dots, d_\kappa),$$

where  $A_i \in \mathbb{R}^{n_i \times n_i}$  and  $d_i \in \mathbb{R}^{n_i}$ , for all  $1 \leq i \leq \kappa$ .

- 3: For small component  $n_i \leq M^{1/3}$ , invoke direct method (or PCG) to solve  $A_i u_i = d_i$ .
- 4: For large component  $n_i > M^{1/3}$ , apply iterative solver to  $A_i u_i = d_i$ .
- 5: Recover the solution  $\xi = \mathcal{Q}u$  with  $(u_1, u_2, \dots, u_\kappa)$ .

---

If  $\Lambda \neq O$ , then  $A$  is SPD for all  $\epsilon \geq 0$ . When  $\Lambda$  vanishes,  $A$  becomes nearly singular if  $\epsilon$  is close to zero. This tricky issue increases the number of iterations of standard solvers like Jacobi iteration, Gauss-Seidel iteration, and PCG; see our numerical evidence in Table 5.1, and we refer to [67] for detailed discussions on this. Moreover, standard iterative methods are not robust concerning the problem size as well, which motivates us to consider the algebraic multigrid (AMG) algorithm.

### 5.3. Classical AMG

Multigrid methods are efficient iterative solvers or preconditioners for large sparse linear SPD systems arising from numerical discretizations of partial differential equations (PDEs) [13, 26, 27, 52, 70, 100, 104, 105, 106].

Those linear systems are always ill-conditioned as the mesh size decreases (or equivalently the problem size increases), and standard stationary iterative solvers converge dramatically slowly. However, multigrid methods possess *mesh-independent* convergence rate and can achieve the *optimal* complexity.

The basic multigrid ingredients are error smoothing and coarse grid correction. In the setting of PDE discretizations, the coarse grid is based on geometric mesh and a multilevel hierarchy can also be constructed easily. On the other hand, the multigrid idea has been applied to the case where no geometric mesh is available. In particular, for the graph Laplacian system (5.31), multilevel hierarchy can still be obtained from the adjacency graph to  $A$ . Then different coarsening techniques and interpolations lead to various algebraic multigrid algorithms, such as classical AMG and aggregation-based AMG [12, 18, 110].

### 5.3.1. Multilevel $W$ -cycle

Let us first present an abstract multilevel  $W$ -cycle framework for solving (5.31). There are two steps: the setup phase and the iteration phase.

In the setup phase, we work with a family of coarse spaces:  $\{V_\ell = \mathbb{R}^{N_\ell}\}_{\ell=1}^J$ , where  $N_J < \dots < N_\ell < \dots < N_1 = N$ , and build some basic ingredients that include

- Smoothers:  $R_\ell : \mathbb{R}^{N_\ell} \rightarrow \mathbb{R}^{N_\ell}$  for all  $1 \leq \ell \leq J$ ;
- Prolongation matrices:  $P_\ell : \mathbb{R}^{N_{\ell+1}} \rightarrow \mathbb{R}^{N_\ell}$  for  $1 \leq \ell \leq J-1$ ;
- Coarse level operators:  $A_1 = A$  and  $A_{\ell+1} = P_\ell^\top A_\ell P_\ell$  for all  $1 \leq \ell \leq J-1$ .

The coarsest level size  $N_J$  is very small and in practice,  $N_J = \mathcal{O}(N^{1/3})$  is acceptable. The prolongation operators  $\{P_\ell\}_{\ell=1}^{J-1}$  shall be injective, i.e., each  $P_\ell \in \mathbb{R}^{N_\ell \times N_{\ell+1}}$  has full column rank. Moreover, since  $A$  might be nearly singular, we require that  $P_\ell \mathbf{1}_{N_{\ell+1}} = \mathbf{1}_{N_\ell}$ , then  $A_\ell \mathbf{1}_{N_\ell}$  is close to zero for all  $1 \leq \ell \leq J$ .

In each level, the smoother  $R_\ell \in \mathbb{R}^{N_\ell \times N_\ell}$  is an approximation to  $A_\ell^{-1}$  and possesses smoothing property. For  $\ell = J$ , we can choose  $R_\ell = A_\ell^{-1}$  or

invoke the PCG iteration. For  $1 \leq \ell < J$ , let  $A_\ell = D_\ell + L_\ell + L_\ell^\top$  where  $D_\ell$  is the diagonal part and  $L_\ell$  is the strictly lower triangular part. Then we can consider

- Gauss–Seidel:  $R_\ell = (D_\ell + L_\ell)^{-1}$  for  $\ell = 1$ ;
- Weighted Jacobi:  $R_\ell = \omega D_\ell^{-1}$  with  $\omega \in (0, 1)$  for  $1 \leq \ell \leq J$ .

For  $\ell = 1$ , thanks to the bipartite graph structure of  $A_1 = A$ , the Gauss–Seidel smoother admits explicit expression. Given a smoother  $R_\ell$ , to handle the possibly nearly singular property of  $A_\ell$ , we follow [67, 86] and adopt a special one

$$\widehat{R}_\ell = \frac{\xi_\ell \xi_\ell^\top}{\xi_\ell^\top A_\ell \xi_\ell} + R_\ell \left( I - A_\ell \frac{\xi_\ell \xi_\ell^\top}{\xi_\ell^\top A_\ell \xi_\ell} \right), \quad (5.32)$$

where  $\xi_\ell = \mathbf{1}_{N_\ell}$  is the approximation kernel of  $A_\ell$ .

Then in the iteration phase, we run the process

$$u_{k+1} = u_k + \text{AMG-W}(d - Au_k, 0, 1), \quad k = 0, 1, \dots, \quad (5.33)$$

where  $g = \text{AMG-W}(\zeta, e, \ell)$  is defined by Algorithm 7 in a recursive way.

---

**Algorithm 7** Algebraic Multigrid  $W$ -cycle:  $g = \text{AMG-W}(\zeta, e, \ell)$

---

**Input:**  $\zeta, e \in \mathbb{R}^{N_\ell}$ ,  $1 \leq \ell \leq J$  and  $\theta \in \mathbb{N}_{\geq 1}$ .

- 1: **if**  $\ell = J$  **then**
- 2:    $g = e + \widehat{R}_\ell(\zeta - A_\ell e)$ .
- 3: **else**
- 4:   **for**  $i = 1, 2, \dots, \theta$  **do** {Presmoothing}
- 5:      $e = e + \widehat{R}_\ell(\zeta - A_\ell e)$ .
- 6:   **end for**
- 7:   Restriction:  $\zeta_{\ell+1} = P_\ell^\top(\zeta - A_\ell e)$ .
- 8:   Coarse correction:  $e_{\ell+1} = \text{AMG-W}(\zeta_{\ell+1}, 0, \ell + 1)$ .
- 9:   Coarse correction:  $e_{\ell+1} = \text{AMG-W}(\zeta_{\ell+1}, e_{\ell+1}, \ell + 1)$ .
- 10:   Prolongation:  $e = e + P_\ell e_{\ell+1}$ .
- 11:   **for**  $i = 1, 2, \dots, \theta$  **do** {Postsmoothing}
- 12:      $e = e + \widehat{R}_\ell^\top(\zeta - A_\ell e)$
- 13:   **end for**
- 14: **end if**

---

**Remark 5.3.1.** We mention that the number of smoothing iterations  $\theta \in \mathbb{N}_{\geq 1}$  in Algorithm 7 is fixed and in most cases a small choice, saying  $\theta = 5$ , works well. In addition, the coarsening procedure, which will be introduced in the next section, leads to the reduction  $N_{\ell+1} \approx N_\ell/2$ , and thus the total number of levels is at most  $J = \mathcal{O}(\ln N)$ . Consequently, if

- (i) the convergence rate  $\rho^k$  of the AMG  $W$ -cycle (5.33) is robust, which means  $\rho \in (0, 1)$  is independent of the singular parameter  $\epsilon$  in  $A$  and the problem size  $N$ , and
- (ii) the matrix-vector operations in each iteration of (5.33) is  $\mathcal{O}(\text{nnz}(A))$ , where  $\text{nnz}(A)$  denotes the number of nonzero elements of  $A$ ,

then to achieve a given tolerance  $\varepsilon$ , the total computational work of the AMG  $W$ -cycle (5.33) is optimal  $\mathcal{O}(\text{nnz}(A)|\ln \varepsilon|)$

### 5.3.2. Coarsening and interpolation

In this part, we shall construct the prolongation operators  $\{P_\ell\}_{\ell=1}^{J-1}$ . In the terminology of AMG, it can be done by *coarsening* and *interpolation* [98, 110]. Here, “interpolation” means the operator  $P_\ell : \mathbb{R}^{N_{\ell+1}} \rightarrow \mathbb{R}^{N_\ell}$  provides a good approximation from the coarse level  $\mathbb{R}^{N_{\ell+1}}$  to the fine level  $\mathbb{R}^{N_\ell}$ . According to the hierarchy structure, it is sufficient to consider the case  $\ell = 1$ , which provides a template for coarse levels  $1 < \ell \leq J$ .

#### Maximal independent set

In classical AMG, the coarsening is based on the so-called  $\mathcal{C} \setminus \mathcal{F}$ -splitting. Recall that  $A_1 = A$  and  $N_1 = N$ . Let  $\mathcal{V} = \{1, 2, \dots, N\}$  and define the *strength* function  $s_A : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$  with respect to  $A$  by that

$$s_A(i, j) := \frac{A_{ij}}{\max\{\min_{k \in \mathcal{N}(i)} A_{ik}, \min_{k \in \mathcal{N}(j)} A_{jk}\}} \quad \forall (i, j) \in \mathcal{V} \times \mathcal{V}, \quad (5.34)$$

where  $\mathcal{N}(i) := \{j \in \mathcal{V} \setminus \{i\} : A_{ij} \neq 0\}$ . Given a threshold  $\delta \in (0, 1)$ , we say  $i \in \mathcal{V}$  and  $j \in \mathcal{V}$  are strongly connected if  $s_A(i, j) > \delta$ . We aim to find a *maximal independent set*  $\mathcal{C} = \{j_1, j_2, \dots, j_{N_2}\} \subset \mathcal{V}$ , such that any  $i \in \mathcal{C}$  and

$j \in \mathcal{C}$  are not strongly connected, i.e.,  $s_A(i, j) \leq \delta$ . Then  $\mathcal{C}$  stands for the collection of coarse nodes and its complement  $\mathcal{F} = \mathcal{V} \setminus \mathcal{C} = \{i_1, i_2, \dots, i_{N_f}\}$  denotes the set of fine nodes, where  $N_f = N - N_2$ . Notice that for any  $i \in \mathcal{F}$ ,  $\mathcal{C} \cap \mathcal{N}(i)$  is nonempty.

A basic splitting algorithm (cf. [110, Algorithm 5]) has been described briefly in Algorithm 8. We refer to [98, Appendix A.7] for an variant, where a measure of importance has been introduced to obtain a reasonable distribution of coarse nodes.

---

**Algorithm 8**  $\mathcal{C} \setminus \mathcal{F}$ -splitting
 

---

- 1: Set the threshold  $\delta \in (0, 1)$ .
- 2: Initialize  $\mathcal{C} = \emptyset$  and  $\mathcal{F} = \emptyset$ .
- 3: Mark all nodes in  $\mathcal{V}$  as unvisited:  $\mathcal{U}(i) = \text{true}$  for all  $i \in \mathcal{V}$ .
- 4: **for**  $i = 1, 2, \dots, N$  **do**
- 5:   **if**  $\mathcal{U}(i) = \text{true}$  **then**  $\{i \text{ has not been visited}\}$
- 6:      $\mathcal{N}_s(i) = \{j \in \mathcal{V} : s_A(i, j) > \delta\}$ .
- 7:      $\mathcal{C} = \mathcal{C} \cup \{i\}$  and  $\mathcal{F} = \mathcal{F} \cup \mathcal{N}_s(i)$ .
- 8:      $\mathcal{U}(i) = \text{false}$  and  $\mathcal{U}(k) = \text{false}$  for all  $k \in \mathcal{N}_s(i)$ .
- 9:   **end if**
- 10: **end for**

---

### Interpolation operator

Once the  $\mathcal{C} \setminus \mathcal{F}$ -splitting has been done, we can find a permutation matrix  $\Xi$  such that

$$\Xi^\top A \Xi = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{FC}^\top & A_{CC} \end{pmatrix}. \quad (5.35)$$

Then, we can choose (see [110, Section 12.3])

- *Ideal interpolation:*  $P = \Xi \begin{pmatrix} W \\ I \end{pmatrix}$  with  $W = -A_{FF}^{-1} A_{FC}$ .
- *Standard interpolation:*  $P = \Xi \begin{pmatrix} (I - D_{FF}^{-1} A_{FF})W \\ I \end{pmatrix}$  with  $D_{FF}$  being the diagonal part of  $A_{FF}$ .

In addition, to satisfy  $P\mathbf{1}_{N_2} = \mathbf{1}_{N_1}$ , we need additional scaling transform  $P_1 = \text{diag}(P\mathbf{1}_{N_2}) \setminus P$ , which leads to the desired prolongation operator from level  $\ell = 2$  to level  $\ell = 1$ .

Observing the particular structure of the system (5.31), where  $A_0$  is the Laplacian of some connected bipartite graph, we find

$$A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{FC}^\top & A_{CC} \end{pmatrix},$$

with  $A_{FF}$  and  $A_{CC}$  being diagonal. This yields an approximate  $\mathcal{C} \setminus \mathcal{F}$ -splitting

$$\mathcal{F} = \{1, 2, \dots, n_f\} \quad \text{and} \quad \mathcal{C} = n_f + \{1, 2, \dots, n_c\},$$

where  $n_f + n_c = N$ . Note that  $\mathcal{C}$  might not be a maximal independent set but provides an approximate ideal interpolation.

However, for  $\ell > 1$ , it is not realistic to expect the bipartite structure of  $A_\ell$ , and to avoid inverting  $A_{FF}$ , we shall consider standard interpolation instead.

## 5.4. Numerical Tests

This section is devoted to providing essential numerical experiments for validating the efficiency of our algorithm. In Section 5.4.1, we aim to verify the robust performance of the AMG  $W$ -cycle iteration (5.33) for solving a nearly singular graph Laplacian system. Then in Section 5.4.2, we apply the overall semismooth Newton-AMG-based inexact primal-dual method (see Algorithm 9) to several transport-like problems listed in Section 1.1.3 and conduct extensive comparisons with existing baseline algorithms. All numerical tests are implemented in MATLAB (version R2021a) on a MacBook Air Laptop.

### 5.4.1. Performance of AMG

Consider the linear algebraic system

$$Ax = (\epsilon I + A_h)x = b. \quad (5.36)$$

Here  $A_h$  is the stiffness matrix of the conforming bilinear finite element method [15] for the pure Neumann problem

$$\begin{cases} -\Delta u = g & \text{in } \Omega := (0, 1)^2, \\ \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases}$$

where  $\mathbf{n}$  is the unit outward normal vector of  $\partial\Omega$  and  $g : \Omega \rightarrow \mathbb{R}$  is a square integrable function with vanishing average.

Let  $\Omega_h = \cup_i K_i$  be a subdivision of  $\Omega$ , where each  $K_i$  is a square with edge length  $h = 2^{-k}$ ,  $k \in \mathbb{N}$ . The stiffness matrix  $A_h \in \mathbb{R}^{N_h \times N_h}$  corresponds to  $\Omega_h$  is sparse and symmetric positive semidefinite with  $N_h = (1 + 1/h)^2$ . Moreover,  $A_h$  is the Laplacian matrix of some connected graph  $\mathcal{G}_h$ , which can be obtained from  $\Omega_h$  by adding the two diagonal lines of each element  $K_i$ ; see Fig. 5.1.

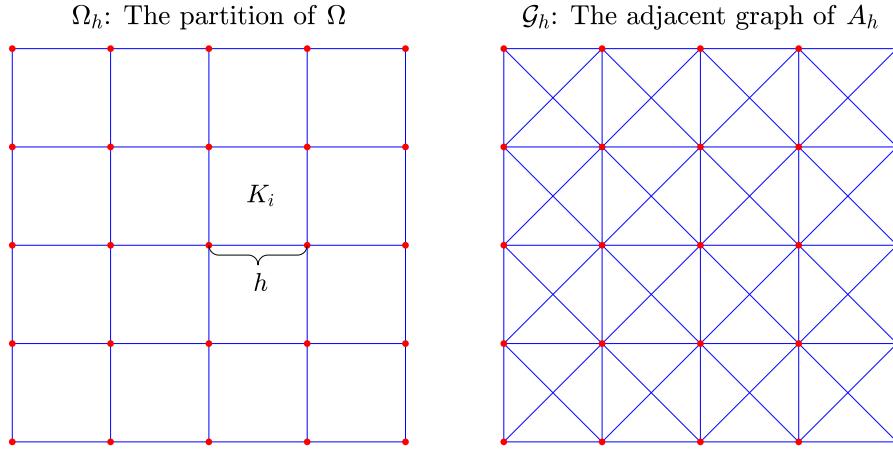


Figure 5.1: *Illustrations of  $\Omega_h$  and  $\mathcal{G}_h$ .*

We apply AMG  $W$ -cycle iteration (5.33) and PCG (cf. [94, Algorithm 9.1]) to (5.36) with different  $\epsilon$  and mesh size  $h$ . For PCG, we choose the diagonal (Jacobi) preconditioner. For AMG, we adopt weighted Jacobi smoother  $R_\ell = 1/2D_\ell$  and the number of smoothing iteration is  $\theta = 5$ . Besides, to obtain a maximal independent set via Algorithm 8 and avoid using the for loop, we adopt a subroutine from the MATLAB software package: *iFEM* [24].

$1/h$	$\epsilon = 10^{-4}$		$\epsilon = 10^{-6}$		$\epsilon = 10^{-8}$		$\epsilon = 10^{-10}$		$\epsilon = 0$	
	itamg	itpcg	itamg	itpcg	itamg	itpcg	itamg	itpcg	itamg	itpcg
$2^4$	9	62	10	66	9	69	9	52	10	52
$2^6$	9	230	9	250	9	266	9	201	10	201
$2^8$	9	789	9	906	9	969	10	740	9	740
$2^{10}$	9	1427	10	3158	9	3531	10	2680	9	2680

Table 5.1: Number of iterations of AMG and PCG.

In Table 5.1, we report the number of iterations of AMG (cf.`itamg`) and PCG (cf.`itpcg`), under the stop criterion

$$\frac{\|Ax_k - b\|}{\|Ax_0 - b\|} \leq \text{Tol} = 10^{-11}.$$

As we can see, AMG is very robust with respect to both the singular parameter  $\epsilon$  and the problem size  $N_h$ . While the number of PCG iterations grows in terms of  $N_h$ . If  $\epsilon$  is decreasing and larger than `Tol`, then due to the nearly singular issue, `itpcg` also increases. When  $\epsilon$  is close to (or is smaller than) `Tol`, the term  $\epsilon I$  in  $A$  is negligible and (5.36) can be viewed almost as a singular system. In this situation, `itpcg` tends to the case  $\epsilon = 0$ . To further show this dependence on  $\epsilon$  more clearly, in Fig. 5.2, we plot the number of iterations for two cases:  $h = 2^{-7}$  and  $h = 2^{-9}$ .

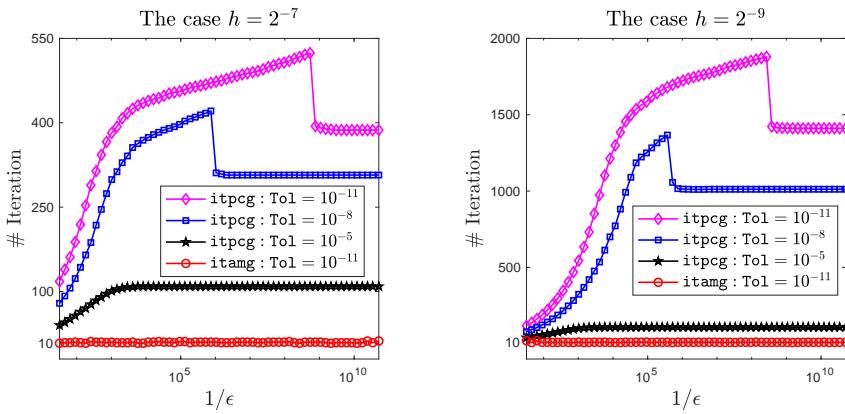


Figure 5.2: Growth behaviors of `itamg` and `itpcg` with respect to the parameter  $\epsilon$  and the tolerance `Tol`.

Except for the number of iterations, we also record two crucial ingredients of the multilevel hierarchy: (i) the number of levels  $J$  and (ii) the *operator complexity* (`opcom` for short), i.e.,

$$\text{opcom} := \frac{\sum_{\ell=1}^J \text{nnz}(A_\ell)}{\text{nnz}(A)}.$$

The quantity `opcom` is often used to measure the computational complexity of the AMG algorithm. From Table 5.2, we might observe the growth magnitude  $\mathcal{O}(|\ln h|)$ , as mentioned in Remark 5.3.1. This is almost negligible and thus both  $J$  and `opcom` are robust with respect to  $h$  and  $\epsilon$ .

1/h	$\epsilon = 10^{-4}$		$\epsilon = 10^{-6}$		$\epsilon = 10^{-8}$		$\epsilon = 10^{-10}$		$\epsilon = 0$	
	$J$	<code>opcom</code>	$J$	<code>opcom</code>	$J$	<code>opcom</code>	$J$	<code>opcom</code>	$J$	<code>opcom</code>
$2^4$	4	1.47	4	1.49	4	1.41	4	1.50	4	1.40
$2^6$	5	1.64	5	1.62	5	1.65	5	1.62	5	1.65
$2^8$	6	1.66	6	1.68	6	1.67	6	1.67	6	1.66
$2^{10}$	7	1.68	7	1.69	7	1.68	7	1.69	7	1.69

Table 5.2: Number of levels and the operator complexity of AMG.

#### 5.4.2. IPD-SsN-AMG method

Combining Algorithms 4, 5, 6 and 7, we obtain the overall Semismooth Newton-AMG-based Inexact Primal-Dual (IPD-SsN-AMG for short) method for solving the generalized transport problem (1.8); see Algorithm 9.

**Algorithm 9** IPD-SsN-AMG method
 

---

**Input:** KKT tolerance:  $\text{KKT\_Tol}$ .

SsN iteration tolerance:  $\text{SsN\_Tol}$ .

Maximal SsN iteration number:  $j_{\max} \in \mathbb{N}$ .

Line search parameters:  $\tau \in (0, 1/2)$ ,  $\delta \in (0, 1)$ .

Initial guesses:  $\beta_0 > 0$ ,  $\lambda_0 \in \mathbb{R}^{m+n+l}$  and  $\mathbf{x}_0 = \mathbf{v}_0 \in \mathbb{R}^{mn+n+m}$ .

- 1: **for**  $k = 0, 1, \dots$  **do**
- 2:   Choose the step size  $\alpha_k > 0$ .
- 3:   Set  $\tau_k = \beta_k(1 + \alpha_k)/\alpha_k^2$  and  $\eta_k = \sigma + \tau_k$ .
- 4:   Set  $D_k = \text{diag}(\eta_k I_{mn}, \tau_k I_n, \tau_k I_m)$  and  $\tilde{\mathbf{x}}_k = \tilde{\mathbf{c}} + \beta_k(\mathbf{x}_k + \alpha_k \mathbf{v}_k)/\alpha_k^2$ .
- 5:   Update  $\beta_{k+1} = \beta_k/(1 + \alpha_k)$  and set  $\tilde{\lambda}_k = \beta_{k+1} [\lambda_k - \beta_{k+1}^{-1}(H\mathbf{x}_k - b)] - b$ .
- 6:   Set  $\lambda_{\text{new}} = \lambda_k$ .
- 7:   **for**  $j = 0, 1, \dots$  **do** {SsN iteration}
- 8:     Set  $\lambda_{\text{old}} = \lambda_{\text{new}}$  and  $\mathbf{z}_k = D_k^{-1}(\tilde{\mathbf{x}}_k - H^\top \lambda_{\text{new}})$ .
- 9:     Compute the diagonal matrix  $U_k \in \partial \text{proj}_\Sigma(\mathbf{z}_k)$  from (5.22).
- 10:    Transform the linear equation

$$\mathcal{J}_k \zeta = (\beta_{k+1} I + H D_k^{-1} U_k H^\top) \zeta = -F_k(\lambda_{\text{old}})$$

into the reduced graph Laplacian system (5.29).

- 11:   Apply Algorithms 6 and 7 to (5.29) and recover the solution  $\zeta$ .
- 12:   Update  $\lambda_{\text{new}} = \lambda_{\text{old}} + \delta^\ell \zeta$  with the smallest nonnegative integer  $\ell \in \mathbb{N}$  that satisfies  $\mathcal{F}_k(\lambda_{\text{old}} + \delta^\ell \zeta) \leq \mathcal{F}_k(\lambda_{\text{old}}) + \tau \delta^\ell \langle F_k(\lambda_{\text{old}}), \zeta \rangle$ .
- 13:   **if**  $\|F_k(\lambda_{\text{new}})\| \leq \text{SsN\_Tol}$  or  $j \geq j_{\max}$  **then**
- 14:     **break**
- 15:   **end if**
- 16:   **end for**
- 17:   Update  $\lambda_{k+1} = \lambda_{\text{new}}$  and  $\mathbf{x}_{k+1} = \text{proj}_\Sigma(\mathbf{z}_k)$ .
- 18:   Update  $\mathbf{v}_{k+1} = \mathbf{x}_{k+1} + (\mathbf{x}_{k+1} - \mathbf{x}_k)/\alpha_k$ .
- 19:   **if**  $\text{Res}(k+1) \leq \text{KKT\_Tol}$  **then** {Check the KKT residual}
- 20:     **break**
- 21:   **end if**
- 22: **end for**

---

Detailed parameter choices and operations of Algorithm 9 are explained in order.

In step 11, the settings of the AMG  $W$ -cycle are the same as that in Section 5.4.1. Note that Algorithm 6 requires the connected components of the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with respect to the Laplacian matrix  $\mathcal{A}_0$  in (5.29). This can be done by using graph searching algorithms [59] such as breadth first search (with the complexity  $\mathcal{O}(|\mathcal{V}||\mathcal{E}|)$ ) and depth first search (with the complexity  $\mathcal{O}(|\mathcal{E}|) = \mathcal{O}(\text{nnz}(\mathcal{A}_0))$ ). Thanks to the bipartite structure, we adopt the MATLAB built-in function `dmpperm` that provides the Dulmage–Mendelsohn decomposition of  $\mathcal{A}_0$  and also returns the connected components.

For SsN iteration, the line search parameters are  $\tau = 0.2$  and  $\delta = 0.9$ , and in step 13, it shall be terminated when either  $j$  is larger than the maximal iteration number  $j_{\max} = 15$  or  $\|F_k(\lambda_{\text{new}})\|$  is smaller than the tolerance  $\text{SsN\_Tol} = \max\{\beta_k(k+1)^{-2}, 10^{-11}\}$ .

Moreover, in step 19 we impose the stop criterion

$$\text{Res}(k) := \max \left\{ \frac{\text{KKT}(x_k)}{\text{KKT}(x_0)}, \frac{\text{KKT}(y_k)}{\text{KKT}(y_0)}, \frac{\text{KKT}(z_k)}{\text{KKT}(z_0)}, \frac{\text{KKT}(\lambda_k)}{\text{KKT}(\lambda_0)} \right\} \leq \text{KKT\_Tol}, \quad (5.37)$$

where  $\text{KKT\_Tol}$  denotes the tolerance and the KKT residuals are defined by

$$\begin{cases} \text{KKT}(x_k) := \|x_k - \mathbf{proj}_{\mathcal{X}}(\sigma\phi + (1-\sigma)x_k - c - G^\top \lambda_k)\|, \\ \text{KKT}(y_k) := \|y_k - \mathbf{proj}_{\mathcal{Y}}(y_k - I_Y^\top \lambda_k)\|, \\ \text{KKT}(z_k) := \|z_k - \mathbf{proj}_{\mathcal{Z}}(z_k - I_Z^\top \lambda_k)\|, \\ \text{KKT}(\lambda_k) := \|Gx_k + I_Y y_k + I_Z z_k - b\|. \end{cases}$$

In the sequel, we investigate the performance of our IPD-SsN-AMG method on specific problems including optimal transport, Birkhoff projection and partial optimal transport. Also, comparisons with the semismooth Newton-based augmented Lagrangian methods proposed in [72, 73] and the accelerated ADMM Algorithm 3 will be presented, under the same stopping condition (5.37) with  $\text{KKT\_Tol} = 10^{-6}$ .

The methods in [72, 73] adopts PCG as the linear system solver, and the (super-)linear convergence analysis is based on classical proximal point framework together with proper error bound assumption. For convenience, we abbreviate these two methods simply as ALM-SsN-PCG.

We note that, as summarized in the introduction part, some other optimization solvers, such as entropy regularization methods and interior-point methods, can also be applied to transport-like problems considered here. However, entropy-based methods provide approximate solutions only with a fixed tolerance, which is almost the same magnitude as the regularization parameter. Interior-point methods utilize the barrier function and require linear system solver as well. Hence, it would be interesting to studying the efficiency comparison between PCG and AMG, and we leave this as our future topic.

## Optimal transport

Let us focus on the optimal mass transport (1.2) with  $m = n \in \mathbb{N}$ . The mass distributions  $\mu, \nu \in \mathbb{R}_+^n$  are generated randomly, and we consider two kinds of cost matrices:

- *Random cost*:

$$C = (C_{ij})_{n \times n} \quad \text{with} \quad C_{ij} \sim \mathcal{U}([0, 1]), \quad (5.38)$$

where  $\mathcal{U}([0, 1])$  denotes the uniform distribution on  $[0, 1]$ ;

- *Quadratic distance cost*:

$$C = (C_{ij})_{n \times n} \quad \text{with} \quad C_{ij} = \|x_i - x_j\|^2, \quad (5.39)$$

where  $\{x_i\}_{i=1}^n$  are the grid points in the uniform subdivision of  $\Omega = (0, 1)^2$  with mesh size  $1/h = \sqrt{n} - 1$ ; see Fig. 5.1.

As discussed in Remark 5.1.1, our IPD-SsN-AMG converges at least linearly as long as the step size is bounded below  $\alpha_k \geq \alpha_0 > 0$ . Practically, we are not allowed to increase  $\alpha_k$  as large as we can. Hence, we choose  $\alpha_k \geq 1$  for small  $k (\leq 10)$  and set  $\alpha_k \in (0, 1)$  for large  $k$ . For ALM-SsN-PCG, there are two crucial parameters  $\sigma_k$  and  $\tau_k$ ; see equation (18) in [72, Algorithm 1]. Theoretically, letting  $\sigma_k$  increase to  $\infty$  and  $\tau_k$  decrease to  $\tau_\infty > 0$  implies superlinear convergence. However, for the sake of practical computation, we set the moderate choice:  $\sigma_k = \mathcal{O}(k^2)$  and  $\tau_k = \mathcal{O}(1/k)$ . Additionally,

we provide a warming-up initial guess for both two algorithms by running Acc-ADMM 100 times.

$m = n$	IPD-SsN-AMG			ALM-SsN-PCG			Acc-ADMM	
	<b>itIPD</b>	<b>itSsN</b>	<b>itamg</b>	<b>itALM</b>	<b>itssn</b>	<b>itpcg</b>	<b>it</b>	<b>residual</b>
			max aver			max aver		
1000	19	170	13 7	46	286	731 190	5000	6.42e-02
2000	29	233	14 7	54	416	1299 235	5000	1.05e-01
3000	29	279	15 7	57	463	2059 284	5000	3.18e-01
4000	39	311	13 6	61	531	2100 264	5000	4.95e-01

Table 5.3: Numerical results for optimal transport with random cost (5.38).

Numerical results with random cost (5.38) and quadratic distance cost (5.39) are listed in Table 5.3 and Table 5.4, respectively. We record (i) the number of iterations (**itIPD** and **itALM**), (ii) the total number of SsN iterations (**itSsN** and **itssn**), and (iii) the maximum (max) and average (aver) iteration number of AMG (**itamg**) and PCG (**itpcg**). Besides, Acc-ADMM is stopped at  $k = 5000$  and we report the corresponding relative KKT residuals.

$m = n$	IPD-SsN-AMG			ALM-SsN-PCG			Acc-ADMM	
	<b>itIPD</b>	<b>itSsN</b>	<b>itamg</b>	<b>itALM</b>	<b>itssn</b>	<b>itpcg</b>	<b>it</b>	<b>residual</b>
			max aver			max aver		
900	29	215	11 7	20	183	628 163	5000	1.03e-01
1600	29	225	11 7	21	226	979 191	5000	1.69e-01
2500	43	328	12 7	25	277	1556 278	5000	2.51e-01
3600	40	352	13 7	35	360	1798 640	5000	3.48e-01

Table 5.4: Numerical results for optimal transport with quadratic distance cost (5.39).

We find that **itIPD** (**itSsN**) is better than **itALM** (**itssn**) for random cost but slightly inferior for quadratic distance cost. Particularly, **itamg** is

more robust than `itpcg`, and we also plot the growth behaviors in Figs. 5.3 and 5.4. As we can see, `itamg` stays around 10 while `itpcg` increases dramatically as  $k$  does.

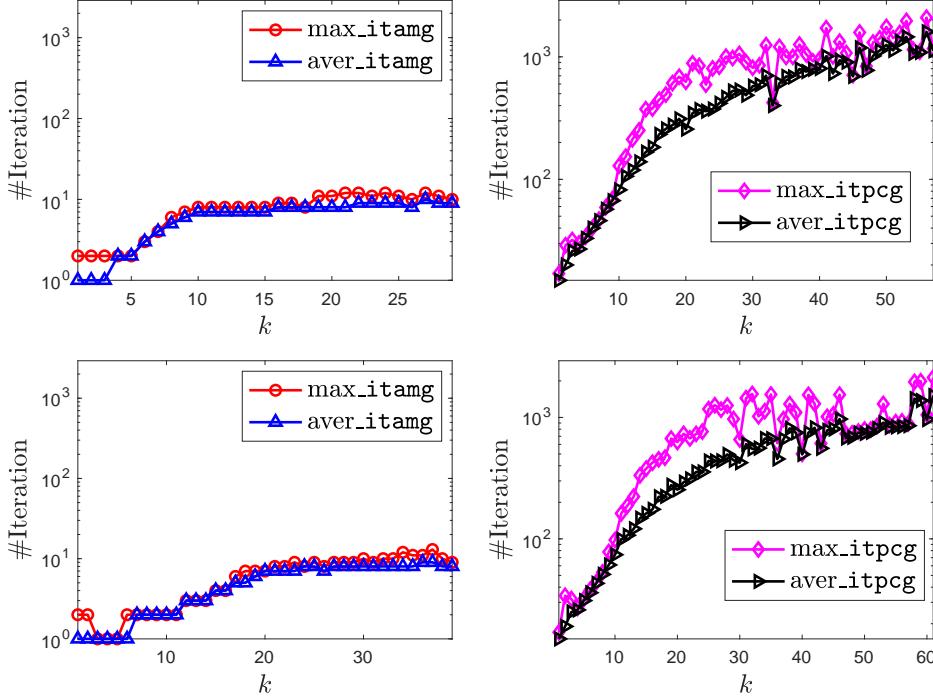


Figure 5.3: *Growth behaviors of `itamg` and `itpcg` for optimal transport with random cost:  $m = n = 3000$  for the top row and  $m = n = 4000$  for the bottom row.*

### Birkhoff projection

We then move to the Birkhoff projection (1.3) with possible entry constraint (1.4). For this problem, we choose fixed large step size  $\alpha_k = 10$  for our IPD-SsN-AMG. Numerical outputs with random data are presented in Tables 5.5 and 5.6. Notice that both two algorithms work well, and `itamg` is still superior than `itpcg` (which is also quite robust). This might be due to the strongly convex property of the problem itself.

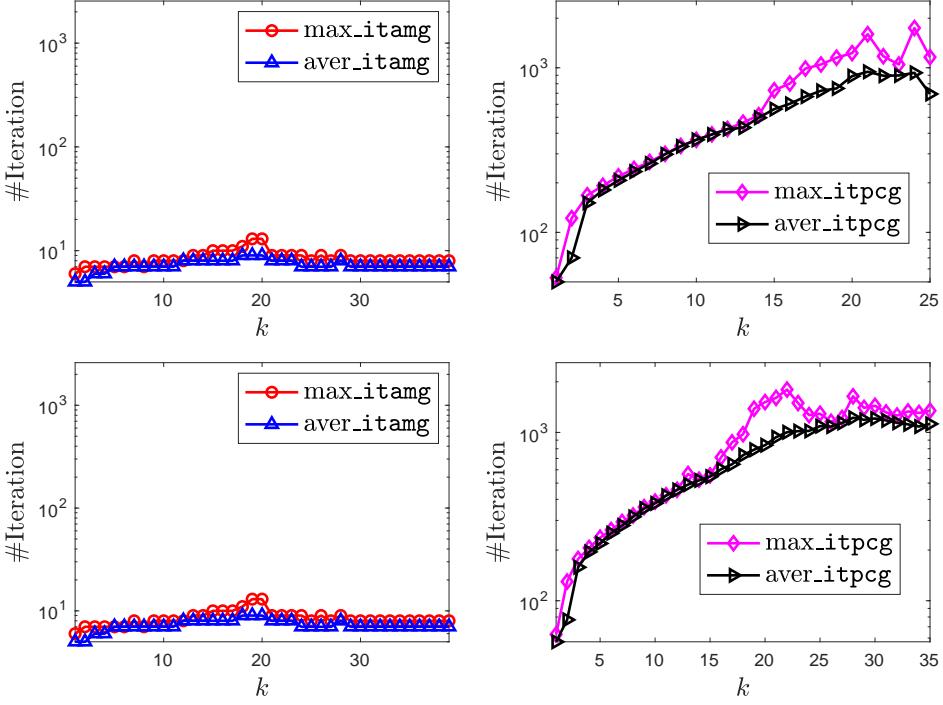


Figure 5.4: Growth behaviors of `itamg` and `itpcg` for optimal transport with quadratic distance cost:  $m = n = 2500$  for the top row and  $m = n = 3600$  for the bottom row.

$m = n$	IPD-SsN-AMG			ALM-SsN-PCG			Acc-ADMM	
	itIPD	itSsN	itamg	itALM	itssn	itpcg	it	residual
			max			max		
2000	6	18	1 1	8	25	15 10	5000	3.33e-03
3000	6	17	1 1	7	24	14 10	5000	3.32e-03
4000	6	19	1 1	7	24	20 10	5000	3.23e-03
5000	6	19	1 1	7	24	26 10	5000	3.20e-03

Table 5.5: Numerical outputs for Birkhoff projection without entry constraint.

$m = n$	IPD-SsN-AMG			ALM-SsN-PCG			Acc-ADMM	
	<b>itIPD</b>	<b>itSsN</b>	<b>itamg</b>	<b>itALM</b>	<b>itssn</b>	<b>itpcg</b>	<b>it</b>	residual
			max aver			max aver		
1000	6	13	1 1	7	21	15 11	5000	3.81e-03
2000	6	20	1 1	8	24	15 10	5000	3.42e-03
3000	6	18	1 1	6	30	14 11	5000	3.30e-03
4000	6	17	1 1	5	42	20 12	5000	3.22e-03

Table 5.6: Numerical outputs for Birkhoff projection with entry constraint.

### Partial optimal transport

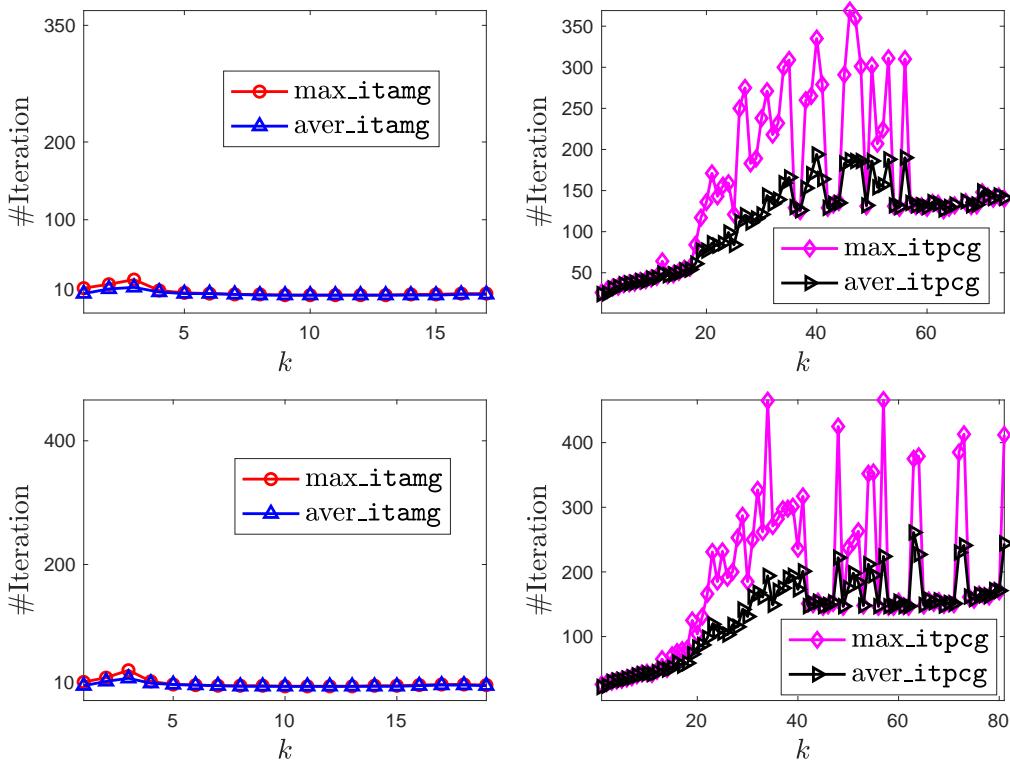
Finally, let us consider the problem of partial optimal transport (1.6) with random cost (5.38) and quadratic distance cost (5.39). Again, the marginal distributions  $\mu$  and  $\nu$  and the fraction of mass  $a$  are generated randomly.

From Tables 5.7 and 5.8, we observe that: (i) similar with the results of optimal transport (see Tables 5.3 and 5.4), **itIPD** is much less than **itALM** for random cost but slightly more than that for quadratic distance cost; (ii) **itSsN** is better than **itssn** for both two cases; (iii) **itamg** stays robust and outperforms **itpcg**.

Growth behaviors of **itamg** and **itpcg** are displayed in Figs. 5.5 and 5.6. One finds that **itamg** is temperately increasing within few initial steps while **itpcg** is not robust with respect to both the iteration process (i.e., the number  $k$ ) and the problem size.

$m = n$	IPD-SsN-AMG			ALM-SsN-PCG			Acc-ADMM	
	itIPD	itSsN	itamg	itALM	itssn	itpcg	it	residual
			max aver			max aver		
1000	20	152	35 14	66	274	452 154	5000	1.55e-01
2000	34	205	25 8	72	352	644 145	5000	4.12e-01
3000	34	225	23 6	74	411	426 87	5000	8.71e-01
4000	33	238	29 6	81	462	555 100	5000	3.22e+01

Table 5.7: Numerical results for partial optimal transport with random cost (5.38).


 Figure 5.5: Growth behaviors of itamg and itpcg for partial optimal transport with random cost:  $m = n = 3000$  for top row and  $m = n = 4000$  for bottom row.

$m = n$	IPD-SsN-AMG				ALM-SsN-PCG				Acc-ADMM	
	itIPD	itSsN	itamg		itALM	itssn	itpcg		it	residual
			max	aver			max	aver		
900	31	154	19	5	18	139	95	60	5000	5.95e-01
1600	32	155	28	5	22	176	98	63	5000	1.77e-01
2500	32	196	36	6	25	223	128	64	5000	6.20e-01
3600	31	204	46	7	29	244	138	66	5000	5.60e-01

Table 5.8: Numerical results for partial optimal transport with quadratic distance cost (5.39).

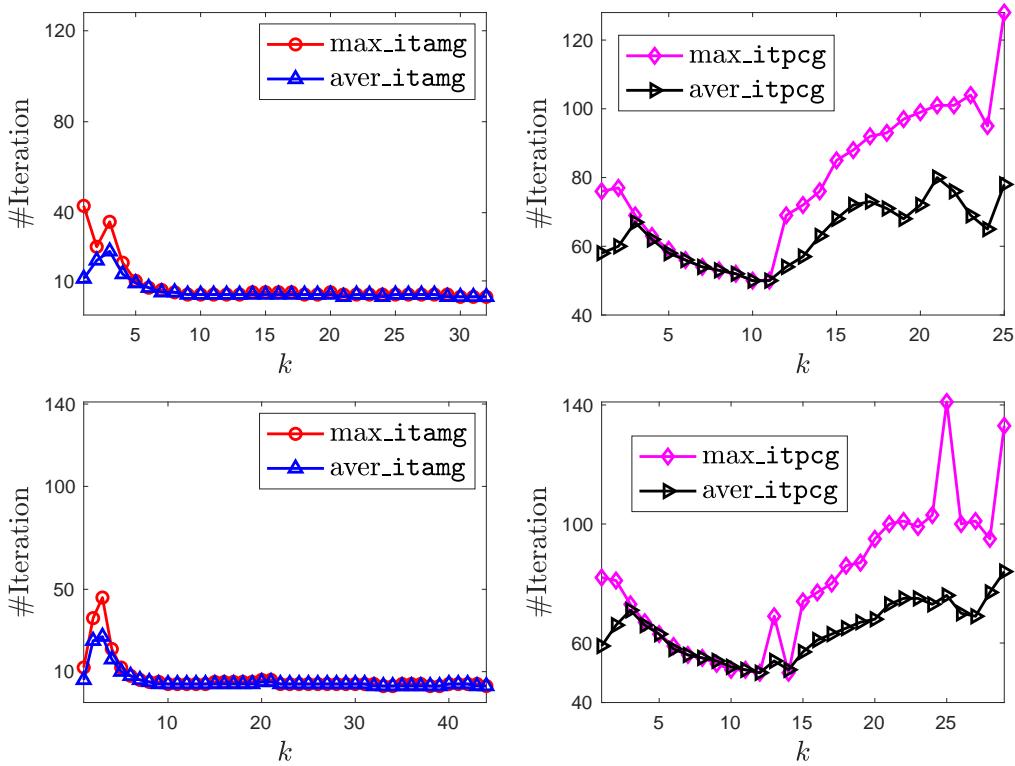


Figure 5.6: Growth behaviors of itamg and itpcg for partial optimal transport with quadratic distance cost:  $m = n = 2500$  for top row and  $m = n = 3600$  for bottom row.

# Bibliography

- [1] B. K. Ahuja. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [2] J. Altschuler, J. Niles-Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.
- [3] L. Ambrosio. Lecture Notes on Optimal Transport Problems. In *Mathematical Aspects of Evolving Interfaces*, volume 1812, pages 1–52. Springer, Berlin, 2003.
- [4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34 th International Conference on Machine Learning*, volume 70, Sydney, Australia, 2017. PMLR.
- [5] Z. Bai, D. Chu, and R. C. E. Tan. Computing the nearest doubly stochastic matrix with a prescribed entry. *SIAM J. Sci. Comput.*, 29(2):635–655, 2007.
- [6] Z. Bai and R. Freund. A partial Padé-via-Lanczos method for reduced-order modeling. *Linear Algebra Appl.*, 332:139–164, 2001.
- [7] H. Bauschke, and P. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer Science+Business Media, New York, 2011.
- [8] J.-D. Benamou. Optimal transportation, modelling and numerical simulation. *Acta Numer.*, 30:249–325, 2021.

- [9] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM J. Sci. Comput.*, 37(2):A1111–A1138, 2015.
- [10] D. P. Bertsekas. Auction algorithms for network flow problems: a tutorial introduction. *Comput. Optim. Appl.*, 1(1):7–66, 1992.
- [11] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [12] R. Blaheta. Algebraic Multilevel Methods with Aggregations: An Overview. In *Large-Scale Scientific Computing*, volume 3743, pages 3–14. Springer Berlin Heidelberg, Berlin, 2006.
- [13] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1–22, 1990.
- [14] C. Brauer, C. Clason, D. Lorenz, and B. Wirth. A Sinkhorn-Newton method for entropic optimal transport. *arXiv:1710.06635*, 2018.
- [15] S. C. Brenner, and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Number 15 in Texts in Applied Mathematics. Springer, New York, NY, 3rd edition, 2008.
- [16] H. Brézis. *Operateurs Maximaux Monotones: Et Semi-Groupes De Contractions Dans Les Espaces De Hilbert*. North-Holland Publishing Co., North-Holland Mathematics Studies, No. 5. Notas de Matemática (50), 1973.
- [17] H. Brezis. Remarks on the Monge–Kantorovich problem in the discrete setting. *Comptes Rendus Mathematique*, 356(2):207–213, 2018.
- [18] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, USA, 2nd edition, 2000.
- [19] R. Brualdi. *Combinatorial Matrix Classes*. Cambridge University Press, New York, 2006.

## Bibliography

---

- [20] R. E. Burkard, M. Dell’Amico, and S. Martello. *Assignment Problems*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2009.
- [21] L. Caffarelli and R. McCann. Free boundaries in optimal transport and Monge–Ampère obstacle problems. *Ann. Math.*, 171(2):673–730, 2010.
- [22] A. Chambolle, and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, 2011.
- [23] L. Chapel and M. Z. Alaya. Partial optimal transport with applications on positive-unlabeled learning. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.
- [24] L. Chen. *iFEM: an integrated finite element methods package in MATLAB*. Technical report, 2009.
- [25] L. Chen. Deriving the X-Z identity from auxiliary space method. In Y. Huang, R. Kornhuber, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering XIX*, volume 78, pages 309–316. Springer, Berlin, 2011.
- [26] L. Chen, X. Hu, and S. M. Wise. Convergence analysis of the fast subspace descent methods for convex optimization problems. *Math. Comput.*, 89(325):2249–2282, 2020.
- [27] L. Chen, R. H. Nochetto, and J. Xu. Optimal multilevel methods for graded bisection grids. *Numer. Math.*, 120:1–34, 2012.
- [28] A. Cherukuri, E. Mallada, and J. Cortés. Asymptotic convergence of constrained primal-dual dynamics. *Syst. Control Lett.*, 87:10–15, 2016.
- [29] A. Chernov, P. Dvurechensky, and A. Gasnikov. Fast primal-dual gradient method for strongly convex minimization problems with linear

constraints. In Y. Kochetov, M. Khachay, V. Beresnev, E. Nurminski, and P. Pardalos, editors, *9th Discrete Optimization and Operations Research*, volume 9869 of *Lecture Notes in Computer Science*, pages 391–403, Vladivostok, Russia, 2016. Springer, Cham.

- [30] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.*, 87(314):2563–2609, 2018.
- [31] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Number 5 in Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1987.
- [32] C. Clason, and T. Valkonen. *Nonsmooth Analysis and Optimization*. <https://arxiv.org/abs/2001.00216>, 2020.
- [33] R. Cominetti and J. S. Martín. Asymptotic analysis of the exponential penalty trajectory in linear programming. *Math. Program.*, 67(1–3):169–187, 1994.
- [34] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.
- [35] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*, pages 2292–2300, 2013.
- [36] J. E. Dennis, and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Number 16 in Classics in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1996.
- [37] B. Djafari-Rouhani, and H. Khatibzadeh. *Nonlinear Evolution and Difference Equations of Monotone Type in Hilbert Spaces*. CRC Press, Boca Raton, 1st edition, 2019.

- [38] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: complexity by accelerated gradient descent is better than by Sinkhorn’s algorithm. *arXiv:1802.04367*, 2018.
- [39] J. Eckstein and D. Bertsekas. An alternating direction method for linear programming. Technical report LIDS-P-1967, Cambridge, 1990.
- [40] F. Facchinei, and J. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems, vol 2*. Springer, New York, 2003.
- [41] M. Fazel, T. Pong, D. Sun, and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM J. Matrix Anal. Appl.*, 34(3):946–977, 2013.
- [42] D. Feijer, and F. Paganini. Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica*, 46:1974–1981, 2010.
- [43] A. R. Ferguson, and G. B. Dantzig. The allocation of aircraft to routes—An example of linear programming under uncertain demand. *Management Science*, 3(1), 1956.
- [44] A. Figalli. The optimal partial transport problem. *Arch. Ration. Mech. Anal.*, 195(2):533–560, 2010.
- [45] F. Fogel, R. Jenatton, F. Bach, and A. d’Aspremont. Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems 26*, pages 1016–1024, 2013.
- [46] A. V. Gasnikov, E. B. Gasnikova, Y. E. Nesterov, and A. V. Chernov. Efficient numerical methods for entropy-linear programming problems. *Comput. Math. Math. Phys.*, 56(4):514–524, 2016.
- [47] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In *30th Conference on Neural Information Processing Systems (NIPS 2016)*, pages 3440–344, Barcelona, Spain, 2016.

- [48] W. Glunt, T. L. Hayden, and R. Reams. The nearest “doubly stochastic” matrix to a real matrix with the same first moment. *Numer. Linear Algebr. Appl.*, 5(6):475–482, 1998.
- [49] Y. Gu, B. Jiang, and D. Han. A semi-proximal-based strictly contractive Peaceman–Rachford splitting method. *arXiv:1506.02221*, 2015.
- [50] S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov. Accelerated alternating minimization, accelerated Sinkhorn’s algorithm and accelerated iterative Bregman projections. *arXiv:1906.03622*, 2021.
- [51] W. Guo, N. Ho, and M. I. Jordan. Accelerated primal-dual coordinate descent for computational optimal transport. *arXiv:1905.09952*, 2020.
- [52] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, Berlin, 2011.
- [53] D. Han, D. Sun, and L. Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Math. Oper. Res.*, 43(2):622–637, 2018.
- [54] A. Haraux. *Systèmes dynamiques dissipatifs et applications*. Recherches en Mathématiques Appliquées [Research in Applied Mathematics], vol 17. Masson, Paris, 1991.
- [55] B. He, and X. Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM J. Imaging Sci.*, 5(1):119–149, 2012.
- [56] B. He, F. Ma, and X. Yuan. An algorithmic framework of generalized primal - dual hybrid gradient methods for saddle point problems. *J Math Imaging Vis*, 58:279–293, 2017.
- [57] R. Hug, E. Maitre, and N. Papadakis. Multi-physics optimal transportation and image interpolation. *ESAIM: Math. Model. Numer. Anal.*, 49(6):1671–1692, 2015.

- [58] B. Jiang, Y.-F. Liu, and Z. Wen.  $l_p$ -norm regularization algorithms for optimization over permutation matrices. *SIAM J. Optim.*, 26(4):2284–2313, 2016.
- [59] D. Jungnickel. *Graphs, Networks and Algorithms*. Number 5 in Algorithms and Computation in Mathematics. Springer, Berlin, 2nd edition, 2005.
- [60] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems 31*, 2018.
- [61] L. Kantorovich. On the translocation of masses. *Dokl. Akad. Nauk. USSR (N.S.)*, 37:199–201, 1942.
- [62] R. N. Khoury. Closest matrices in the space of generalized doubly stochastic matrices. *J. Math. Anal. Appl.*, 222(2):562–568, 1998.
- [63] J. Korman, and R. J. McCann. Optimal transportation with capacity constraints. *Trans. Am. Math. Soc.*, 367(3):1501–1521, 2014.
- [64] J. Korman, and R. J. McCann. Insights into capacity-constrained optimal transport. *Proc. Natl. Acad. Sci. U. S. A.*, 110(25):10064–10067, 2013.
- [65] J. Korman, R. J. McCann, and Seis, Christian. Dual potentials for capacity constrained optimal transport. *Calc. Var. Partial Differ. Equ.*, 154:573–584, 2015.
- [66] J. Korman, R. J. McCann, and Seis, Christian. A penalization approach to linear programming duality with application to capacity constrained transport. *arXiv:1309.3022*, 2013.
- [67] Y.-J. Lee, J. Wu, J. Xu, and L. Zikatanov. Robust subspace correction methods for nearly singular systems. *Math. Models Meth. Appl. Sci.*, 17(11):1937–1963, 2007.

- [68] Y. T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in  $\tilde{O}(\sqrt{\text{rank}})$  iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.
- [69] B. Lévy. Partial optimal transport for a constant-volume Lagrangian mesh with free boundaries. *J. Comput. Phys.*, 451:1–26, 2022.
- [70] B. Li, and X. Xie. BPX preconditioner for nonstandard finite element methods for diffusion problems. *SIAM J. Numer. Anal.*, 54(2):1147–1168, 2016.
- [71] H. Li, and Z. Lin. Accelerated alternating direction method of multipliers: An optimal  $O(1/k)$  nonergodic analysis. *J. Sci. Comput.*, 79(2):671–699, 2019.
- [72] X. Li, D. Sun, and K.-C. Toh. An asymptotically superlinearly convergent semismooth Newton augmented Lagrangian method for Linear Programming. *SIAM J. Optim.*, 30(3):2410–2440, 2020.
- [73] X. Li, D. Sun, and K.-C. Toh. On the efficient computation of a generalized Jacobian of the projector over the Birkhoff polytope. *Math. Program.*, 179(1-2):419–446, 2020.
- [74] Q. Liao, J. Chen, Z. Wang, B. Bai, S. Jin, and H. Wu. Fast Sinkhorn I: An  $O(N)$  algorithm for the Wasserstein-1 metric. *arXiv:2202.10042*, 2022.
- [75] C. H. Lim, and S. J. Wright. Beyond the Birkhoff polytope: convex relaxations for vector permutation problems. In *Advances in Neural Information Processing Systems*, pages 2168–2176, 2014.
- [76] M. Lin, D. Sun, and K.-C. Toh. An augmented Lagrangian method with constraint generations for shape-constrained convex regression problems. *Math. Program.*, 14:223–270, 2022.

- [77] T. Lin, N. Ho, and M. I. Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *International Conference on Machine Learning*, pages 3982–3991. PMLR, 2019.
- [78] Y. Liu, Z. Wen, and W. Yin. A multiscale semi-smooth Newton method for optimal transport. *J. Sci. Comput.*, 91(2):1–39, 2022.
- [79] H. Luo. Accelerated differential inclusion for convex optimization. *Optimization*, <https://doi.org/10.1080/02331934.2021.2002327>, 2021.
- [80] H. Luo. Accelerated primal-dual methods for linearly constrained convex optimization problems. *arXiv:2109.12604*, 2021.
- [81] H. Luo. A unified differential equation solver approach for separable convex optimization: splitting, acceleration and nonergodic rate. *arXiv:2109.13467*, 2021.
- [82] H. Luo, and L. Chen. From differential equation solvers to accelerated first-order methods for convex optimization. *Math. Program.*, <https://doi.org/10.1007/s10107-021-01713-3>, 2021.
- [83] O. L. Mangasarian and T.-H. Shiau. Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems. *SIAM J. Control Optim.*, 25(3):583–595, 1987.
- [84] J. Maas, M. Rumpf, C. Schönlieb, and S. Simon. A generalized model for optimal transport of images including dissipation and density modulation. *ESAIM: Math. Model. Numer. Anal.*, 49(6):1745–1769, 2015.
- [85] Q. Mérigot, and B. Thibert. Optimal Transport: Discretization and Algorithms. In *Handbook of Numerical Analysis*, volume 22, pages 133–212. Elsevier, 2021.
- [86] A. Padiy, O. Axelsson, and B. Polman. Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems. *SIAM J. Matrix Anal. Appl.*, 22(3):793–818, 2001.

- [87] V. M. Panaretos and Y. Zemel. Amplitude and phase variation of point processes. *Ann. Stat.*, 44(2):771–812, 2016.
- [88] O. Pele and M. Werman. Fast and robust earth mover’s distances. In *In 2009 IEEE 12th International Conference on Computer Vision*, pages 460–467, 2009.
- [89] G. Peyré, and M. Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11(5-6):1–257, 2019.
- [90] L. Qi. Convergence analysis of some algorithms for solving nonsmooth equations. *Math. Oper. Res.*, 18(1):227–244, 1993.
- [91] L. Qi, and J. Sun. A nonsmooth version of Newton’s method. *Math. Program.*, 58(1-3):353–367, 1993.
- [92] S. Robinson. Bounds for error in the solution set of a perturbed linear program. *Linear Algebra Appl.*, 6(C):69–81, 1973.
- [93] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [94] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, USA, 2nd edition, 2003.
- [95] F. Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Springer, Cham, 2015.
- [96] R. Sinkhorn. Diagonale quivalence to matrices with prescribed row and columnsums. *The American Mathematical Monthly*, 74(4):402–405, 1967.
- [97] G. Székely and M. Rizzo. Testing for equal distributions in high dimension. In *Inter-Stat (London)*, pages 1–16, 2004.
- [98] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.

- [99] Q. Tran-Dinh, and Y. Zhu. Augmented Lagrangian-based decomposition methods with non-ergodic optimal rates. *arXiv:1806.05280*, 2018.
- [100] Y. Wu, L. Chen, X. Xie, and J. Xu. Convergence analysis of V-Cycle multigrid methods for anisotropic elliptic equations. *IMA J. Numer. Anal.*, 32(4):1329–1347, 2012.
- [101] C. Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2003.
- [102] C. Villani. *Optimal Transport: Old and New*. Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.
- [103] Y. Xie, X. Wang, R. Wang, and H. Zha. A fast proximal point method for computing exact Wasserstein distance. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, Tel Aviv, Isreal, 2019.
- [104] J. Xu. *Theory of Multilevel Methods*. PhD Thesis, Cornell University, Ithaca, New York, 1989.
- [105] J. Xu. A new class of iterative methods for nonself-adjoint or indefinite problems. *SIAM J. Numer. Anal.*, 29(2):303–319, 1992.
- [106] J. Xu. Two-grid discretization techniques for linear and nonlinear PDEs. *SIAM J. Numer. Anal.*, 33(5):1759–1777, 1996.
- [107] J. Xu. *The Method of Subspace Corrections*. Lecture Notes, 2013.
- [108] J. Xu. *Multilevel Iterative Methods*. Lecture Notes. Penn State University, 2017.
- [109] J. Xu, and L. Zikatanov. The method of alternating projections and the method of subspace corrections in Hilbert space. *J. Am. Math. Soc.*, 15(3):573–597, 2002.

## Bibliography

---

- [110] J. Xu, and L. Zikatanov. Algebraic multigrid methods. *Acta Numer.*, 26:591–721, 2017.
- [111] Y. Xu. Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM J. Optim.*, 27(3):1459–1484, 2017.
- [112] M. Zhu, and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. Technical report CAM Report 08-34, UCLA, Los Angeles, CA, USA, 2008.
- [113] L. T. Zikatanov. Two-sided bounds on the convergence rate of two-level methods. *Numer. Linear Algebr. Appl.*, 15(5):439–454, 2008.

# Publications

September. 2020–July. 2022, as a post-doctor at Peking University

- (1) Jun Hu, **Hao Luo**, and Zihang Zhang. An efficient semismooth Newton-AMG-based inexact primal-dual algorithm for generalized transport problems. Preprint, 2022.
- (2) Jun Hu, **Hao Luo**, and Zihang Zhang. Efficient proximal alternating direction methods of multipliers for the Monge–Kantorovich mass transfer problem. Preprint, 2022.
- (3) **Hao Luo**. A primal-dual flow for affine constrained convex optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 28: <https://doi.org/10.1051/cocv/2022032>, 2022.
- (4) Binjie Li, **Hao Luo**, and Xiaoping Xie. Error estimation of a discontinuous Galerkin method for time fractional subdiffusion problems with nonsmooth data. *Fractional Calculus and Applied Analysis*, 25(2):747–782, 2022.
- (5) **Hao Luo**, and Xiaoping Xie. Optimal error estimation of a time-spectral method for fractional diffusion problems with low regularity data. *Journal of Scientific Computing*, 91(1): <https://doi.org/10.1007/s10915-022-01791-1>, 2022.
- (6) **Hao Luo**, and Long Chen. From differential equation solvers to accelerated first-order methods for convex optimization. *Mathematical Programming*, <https://doi.org/10.1007/s10107-021-01713-3>, 2021.

- (7) **Hao Luo.** Accelerated differential inclusion for convex optimization. *Optimization*, <https://doi.org/10.1080/02331934.2021.2002327>, 2021.
- (8) **Hao Luo.** Accelerated primal-dual methods for linearly constrained convex optimization problems. *arXiv:2109.12604*, 2021.
- (9) **Hao Luo.** A unified differential equation solver approach for separable convex optimization: splitting, acceleration and nonergodic rate. *arXiv:2109.13467*, 2021.