MATHEMATICS OF COMPUTATION
Volume 00, Number 0, Pages 000–000
S 0025-5718(XX)0000-0

A FAST PROXIMAL GRADIENT METHOD AND CONVERGENCE ANALYSIS FOR DYNAMIC MEAN FIELD PLANNING

JIAJIA YU, RONGJIE LAI, WUCHEN LI, AND STANLEY OSHER

Abstract. In this paper, we propose an efficient and flexible algorithm to solve dynamic mean-field planning problems based on an accelerated proximal gradient method. Besides an easy-to-implement gradient descent step in this algorithm, a crucial projection step becomes solving an elliptic equation whose solution can be obtained by conventional methods efficiently. By induction on iterations used in the algorithm, we theoretically show that the proposed discrete solution converges to the underlying continuous solution as the grid becomes finer.Furthermore, we generalize our algorithm to mean-field game problems and accelerate it using multilevel and multigrid strategies. We conduct comprehensive numerical experiments to confirm the convergence analysis of the proposed algorithm, to show its efficiency and mass preservation property by comparing it with state-of-the-art methods, and to illustrates its flexibility for handling various mean-field variational problems.

1. Introduction

Mean field planning (MFP) problems study how a large number of similar rational agents make strategic movements to minimize their cost in a process satisfying given initial and terminal density distributions [2, 20, 27, 28, 29, 40, 44, 45]. On the one hand, MFP can be viewed as a generalization of optimal transport (OT) [11, 12, 43, 48] where no interaction cost is considered in the process. On the other hand, MFP is also a special case of mean field game (MFG) problems where the terminal density is often provided implicitly [19, 21, 22, 29, 30, 32, 33, 35]. MFP, MFG and OT have wide applications in economics [1, 5, 25], engineering [24, 26, 50], quantum chemistry [18, 23], image processing [31, 41] as well as machine learning [8, 47, 49, 51].

More specifically, the dynamic MFP problem has the following optimization formulation:

$$\begin{aligned} \min_{\rho,\boldsymbol{m}} & \int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t + \int_0^1 \mathcal{F}(\rho(t,\cdot))\mathrm{d}t \\ \text{s.t. } & \partial_t\rho + \mathrm{div}_{\boldsymbol{x}}\,\boldsymbol{m} = 0, \boldsymbol{m}\cdot\boldsymbol{n} = 0 \text{ for } \boldsymbol{x}\in\partial\Omega, \rho(0,\cdot)=\rho_0, \rho(1,\cdot)=\rho_1. \end{aligned} \tag{1.1}$$

2020 *Mathematics Subject Classification.* Primary 49M41, 49M25, 65K10.

Key words and phrases. Mean field planning, optimal transport, mean field games, multigrid method, FISTA.

J. Yu and R. Lai's work are supported in part by an NSF Career Award DMS–1752934.

W. Li and S. Osher's work are supported in part by AFOSR MURI FP 9550-18-1-502.

where $\rho(t,\boldsymbol{x})\geq 0$ is the densities of agents, $\boldsymbol{m}:=\rho\boldsymbol{v}$ with $\boldsymbol{v}$ representing the strategy(control) of this agent, and any pair of feasible $(\rho,\boldsymbol{m})$ satisfies mass conservation and zero boundary flux conditions with initial and terminal densities of ρ being ρ_0,ρ_1 provided as: In this variational problem, $L(\rho,\boldsymbol{m})$ denotes the dynamic cost, $\mathcal{F}$ models the interaction cost. L is convex in $\rho,\boldsymbol{m}$, $\mathcal{F}$ is convex in ρ and $\frac{\delta\mathcal{F}}{\delta\rho}$ exists. Specially, with $\mathcal{F}=0$ and a specific choice of L, variational problem (1.1) reduces to the dynamic formulation of optimal transport (OT) proposed in [11, 12]. By relaxing the given terminal density as an implicit condition regularized by a convex functional $\mathcal{G}$ with $\frac{\delta\mathcal{G}}{\delta\rho}$ existing, one can retrieve a class of MFG as the following formulation [16, 30, 35]:

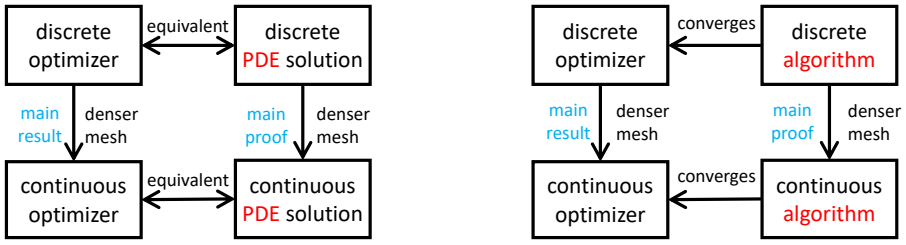
$$\begin{aligned}\min_{\rho,\boldsymbol{m}}\int_0^1\int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t+\int_0^1\mathcal{F}(\rho(t,\cdot))\mathrm{d}t+\mathcal{G}(\rho(1,\cdot))\\ \text{s.t. }\partial_t\rho+\mathrm{div}_{\boldsymbol{x}}\,\boldsymbol{m}=0,\boldsymbol{m}\cdot\boldsymbol{n}=0\text{ for }\boldsymbol{x}\in\partial\Omega,\rho(0,\cdot)=\rho_0.\end{aligned}\tag{1.2}$$

The goal is to find the local minimizers and therefore solving the KKT system of (1.1) (or (1.2)).

Several numerical methods have been established to solve dynamic MFP, MFG and OT problems. One class of methods is based on solving partial differential equations (PDEs) corresponding to the KKT system of the variational problem [2, 3, 4, 17], where conventional numerical methods in nonlinear PDEs can be applied. This class of methods can also be applied to handle general MFP and MFG problems that may not come from variational formulas. However, the nonlinearity of the PDE system limits the solvers to handle broader choices of the dynamic cost L and interaction cost $\mathcal{F}$.

Another class of methods focuses on the variational formulas of dynamic MFP, MFG and OT problems. By naturally combining with recent advances from optimization, existing methods include several first-order optimization algorithms to solve dynamic OT problems such as augmented Lagrangian [14, 15, 42], primal-dual [36] and G-prox [34], etc. These methods work on either the Lagrangian or the dual problem of the original optimization problem, particularly for dynamic OT where $\mathcal{F}\equiv 0$. These algorithms work very well since the involved sub-optimization problems have closed-form solutions.

We would like to propose a method that can efficiently compute the mean-field type of problems with mass preservation property and flexibility on a broad range of objective functions. Note that the mass conservation constraint in MFP is linear. A straightforward calculation shows that projection to the constraint set can be obtained from solving a linear equation, the standard Poisson equation. This motivates us to propose another algorithm to solve MFP problems based on the proximal gradient descent method [46, 9]. This method is composed of a gradient descent step and a projection step. For MFP problems with a smooth objective function, the gradient values can be evaluated in an element-wise manner. It also enjoys the flexibility to handle a broader range of L and $\mathcal{F}$. More importantly, the projection step leads to mass preservation in each iteration. The crucial part in the projection step is a fixed linear solver which can be computed efficiently by conventional fast algorithms. In this work, we use an accelerated version of the proximal gradient descent method, referred to as the fast iterative soft threshold algorithm (FISTA) [10], to solve the MFP problems. After that, we further generalize our algorithm to handle MFG problems. In addition, inspired by [7, 38, 39], we also



(a) Proof in [2] (b) Our proof

FIGURE 1. Sketch of two approaches of convergence proof.

apply multigrid and multilevel methods to speed up the proposed algorithm. Our numerical experiments illustrate the efficiency, mass preservation and flexibility of the proposed algorithm to different MFP problems as well as MFG problems. The vanilla version of our algorithm performs comparable with state-of-the-art methods, while the multigrid and multilevel accelerated versions are more efficient than state-of-the-art methods.

Besides proposing a new algorithm for MFP problems, we analyze errors between the discrete solution and the continuous solution. Since MFP is a functional optimization problem, all numerical methods on a given mesh grid only provide approximated solutions to the continuous problem. It is important to understand how close the discrete numerical solution is to the continuous solution on a given mesh grid. Our analysis is from the algorithm perspective. We first derive an algorithm to optimize the variational problem and discretize each step of our algorithm. Our main effort is to prove that at each iteration, the discrete values are not far from the underlying continuous function values on grid points. Therefore we can show that the discrete algorithm converges to the continuous optimizer on grid points under certain smoothness conditions. Similar types of analysis may not be conveniently conducted in the existing methods including augmented Lagrangian, primal-dual and G-Prox since it could be difficult to have desired perturbation analysis of solving cubic equations involved in these three methods. We remark that [13, 14] show the Γ-convergence for static problems by finite element methods, but they acknowledge that their assumptions for convergence to hold are more involved to check for dynamic problems. In this paper, we are working on dynamic problems. And we also notice that the convergence analysis for dynamic problems has been studied in [2, 3, 6] from the PDE perspective, where the authors argue solution of discrete KKT converges to the continuous solution based on the equivalence of continuous systems and discrete systems. We indicate the major difference between our error analysis based on optimization perspective and error analysis based on PDEs perspective in Figure 1.

Contributions: We summarize our contributions as follows:

(1) We propose to use an accelerate proximal gradient method to solve the MFP problem (1.1).

(2) We analyze the error between the each iteration of discrete optimization and its continuous counter part. We prove that the discrete solution converges to continuous optimizer on grid points as the mesh size converges.
(3) We apply multilevel and multigrid strategies to to accelerate our algorithm. We also generalize our algorithm to solve MFG problems.
(4) We conduct comprehensive numerical experiments to illustrate the efficiency and flexibility of our algorithms.

2. Review

In this section, we briefly review MFP problem and provide several examples which will be computed in the experiment section.

Consider the model on time interval $[0,1]$ and space region $\Omega \in \mathbb{R}^D$. Let ρ be the density of agents through $t \in [0,1]$, $\boldsymbol{m}$ be the flux of the density which models strategies (control) of the agents, and $(\rho, \boldsymbol{m}) \in \mathcal{C}$:

(2.1)
$$\mathcal{C} := \left\{ \begin{array}{c} (\rho, \boldsymbol{m}) : \rho : [0,1] \times \Omega \to \mathbb{R}^+, \|\rho\|_{L^1} < +\infty, \int_\Omega \rho(t, \boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1, \forall t \in [0,1], \\ \boldsymbol{m} : [0,1] \times \Omega \to \mathbb{R}^D \text{ is Lebesgue measurable}, \end{array} \right\}.$$

We are interested in ρ with given initial and terminal density ρ_0, ρ_1 and $(\rho, \boldsymbol{m})$ satisfying zero boundary flux and mass conservation law, which gives the constraint set $\mathcal{C}(\rho_0, \rho_1)$:

$$\mathcal{C}(\rho_0, \rho_1) := \mathcal{C} \cap \left\{ \begin{array}{c} (\rho, \boldsymbol{m}) : \partial_t \rho + \mathrm{div}_{\boldsymbol{x}} \boldsymbol{m} = 0, \\ \boldsymbol{m} \cdot \boldsymbol{n} = 0 \text{ for } \boldsymbol{x} \in \partial\Omega, \rho(0, \cdot) = \rho_0, \rho(1, \cdot) = \rho_1, \end{array} \right\}. \tag{2.2}$$

where equations hold in the sense of distribution.

We denote $L : \mathbb{R}^+ \times \mathbb{R}^D \to \overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ as the dynamic cost function (e.g. (2.5) in this paper) and $\mathcal{F} : \Omega^* \to \overline{\mathbb{R}}$ as a functional modeling interaction cost. The goal of MFP is to minimize the total cost among all feasible $(\rho, \boldsymbol{m}) \in \mathcal{C}(\rho_0, \rho_1)$. Therefore the problem can be formulated as

$$\min_{(\rho, \boldsymbol{m}) \in \mathcal{C}(\rho_0, \rho_1)} \int_0^1 \int_\Omega L(\rho(t, \boldsymbol{x}), \boldsymbol{m}(t, \boldsymbol{x})) \mathrm{d}\boldsymbol{x} \mathrm{d}t + \int_0^1 \mathcal{F}(\rho(t, \cdot)) \mathrm{d}t. \tag{2.3}$$

It is clear to see $\mathcal{C}(\rho_0, \rho_1)$ is convex and compact. In addition, the mass conservation law $\partial_t \rho + \mathrm{div}_{\boldsymbol{x}} \boldsymbol{m} = 0$ and zero flux boundary condition $\boldsymbol{m} \cdot \boldsymbol{n} = 0, \boldsymbol{x} \in \partial\Omega$ imply that $\mathcal{C}(\rho_0, \rho_1) \neq \emptyset$ if and only if $\int_\Omega \rho_0 = \int_\Omega \rho_1$. Once $\mathcal{C}(\rho_0, \rho_1)$ is non-empty, the existence and uniqueness of the optimizer depends on L and $\mathcal{F}$.

There are many different choices of $\mathcal{F}$. In this paper, we consider

$$\mathcal{F}(\rho(t, \cdot)) := \lambda_E \int_\Omega F_E(\rho(t, \boldsymbol{x})) \mathrm{d}\boldsymbol{x} + \lambda_Q \int_\Omega \rho(t, \boldsymbol{x}) Q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{2.4}$$

where $\lambda_E, \lambda_Q \geq 0$ are two parameters, $F_E : \mathbb{R}^+ \to \mathbb{R}$ serves as a function to regularize ρ, and $Q(\boldsymbol{x}) : \Omega \to \overline{\mathbb{R}}$ provides a moving preference for density ρ. Consider an illustrative example by choosing $\Omega_0 \subset \Omega$ and assuming $Q(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \Omega_0 \\ +\infty, & \boldsymbol{x} \notin \Omega_0 \end{cases}$, then the mass has to move within Ω_0 in order to keep the cost finite. In more general choice of Q, $\rho(t, \boldsymbol{x})$ tends to be smaller at the place where $Q(\boldsymbol{x})$ is larger and vice versa.

We then briefly discuss several concrete examples which will be considered in our numerical experiments.

Example 2.1 (Optimal transport [11])**.** In this paper, we consider a typical dynamic cost function L by

$$L(\beta_0,\boldsymbol{\beta}) := \begin{cases} \frac{\|\boldsymbol{\beta}\|^2}{2\beta_0} & \text{if } \beta_0 > 0 \\ 0 & \text{if } \beta_0 = 0, \boldsymbol{\beta} = \mathbf{0} \\ +\infty & \text{if } \beta_0 = 0, \boldsymbol{\beta} \neq \mathbf{0}. \end{cases}, \tag{2.5}$$

If $\lambda_E = \lambda_Q = 0$, the MFP becomes the dynamic formulation of optimal transport problem:

$$\text{(OT)} \quad \min_{\rho,\boldsymbol{m}\in\mathcal{C}(\rho_0,\rho_1)} \int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t. \tag{2.6}$$

Since $\boldsymbol{m} = \rho\boldsymbol{v}$, this definition of L makes sure that $\boldsymbol{m} = \mathbf{0}$ wherever $\rho = 0$. Because $\lambda_E = \lambda_Q = 0$, OT can be viewed as a special case of MFP where masses move freely in Ω through $t \in [0,1]$.

Example 2.2 (Crowd motion [47])**.** Consider $F_E : \mathbb{R}^+ \to \mathbb{R}, \rho \mapsto \begin{cases} \rho\log(\rho), & \rho > 0 \\ 0, & \rho = 0 \end{cases}$, and write $\Omega^+ := \Omega \cap \{\boldsymbol{x} \in \Omega : \rho(t,\boldsymbol{x}) > 0\}$, we have the crowd motion model

$$\min_{\rho,\boldsymbol{m}\in\mathcal{C}(\rho_0,\rho_1)} \left\{ \begin{array}{c} \displaystyle\int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_E \int_0^1 \int_{\Omega^+} \rho(t,\boldsymbol{x})\log(\rho(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t + \lambda_Q \int_0^1 \int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \end{array} \right\}. \tag{2.7}$$

With F_E decreasing on $[0,e^{-1}]$ and increasing on $[e^{-1},+\infty)$, $\rho(t,\boldsymbol{x})$ tends to be close to e^{-1} everywhere. So we expect to have the density $\rho(t,\boldsymbol{x})$ to be not sparse and not very large everywhere.

Example 2.3. If $F_E : \mathbb{R}^+ \to \mathbb{R}, \rho \mapsto \begin{cases} \frac{1}{|p|}\rho^p, & \rho > 0 \\ 0, & \rho = 0 \end{cases}$, where $p = 2$ or -1, then we have the following two models.

$$\min_{\rho,\boldsymbol{m}\in\mathcal{C}(\rho_0,\rho_1)} \left\{ \begin{array}{c} \displaystyle\int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_E \int_0^1 \int_\Omega \frac{\rho^2(t,\boldsymbol{x})}{2}\mathrm{d}\boldsymbol{x}\mathrm{d}t + \lambda_Q \int_0^1 \int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \end{array} \right\} \tag{2.8}$$

$$\min_{\rho,\boldsymbol{m}\in\mathcal{C}(\rho_0,\rho_1)} \left\{ \begin{array}{c} \displaystyle\int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_E \int_0^1 \int_{\Omega^+} \frac{1}{\rho(t,\boldsymbol{x})}\mathrm{d}\boldsymbol{x}\mathrm{d}t + \lambda_Q \int_0^1 \int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \end{array} \right\} \tag{2.9}$$

In (2.8), by Cauchy-Schwarz inequality, we have

$$\left(\int_\Omega \rho(t,\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2 \leq \int_\Omega \rho^2(t,\boldsymbol{x})\mathrm{d}\boldsymbol{x} \int_\Omega 1\mathrm{d}\boldsymbol{x}, \tag{2.10}$$

therefore $\int_\Omega \rho^2(t,\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ has a lower bound and achieves the lower bound when $\rho(t,\cdot)$ is a constant over Ω. Therefore, model (2.8) guides the solution density uniformly distributed over Ω . In (2.9), since the total mass $\int_\Omega \rho(t,\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ is fixed and $\dfrac{1}{\rho}$ is larger when ρ is smaller, the value of regularization term $\lambda_E \displaystyle\int_{\Omega^+} \frac{1}{\rho(t,\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ is smaller if $\rho(t,\boldsymbol{x})$ accumulates at several sites and vanishes at other regions. Therefore model (2.9) pursues a sparse optimizer $\rho(t,\boldsymbol{x})$.

Example 2.4 (A MFG model [16, 30, 35])**.** We provide an example of the MFG model (1.2) to complete this section In the cases, the terminal density ρ_1 is not explicitly provided but it satisfies a given preference. This preference can be imposed by regularizing $\rho(1,\cdot)$ in the same spirit as $\int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ and obtain the following MFG model,

(2.11)
$$\min_{(\rho,\boldsymbol{m})\in\mathcal{C}(\rho_0)} \left\{ \begin{array}{l} \displaystyle\int_0^1\int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_E\int_0^1\int_\Omega \rho(t,\boldsymbol{x})\log(\rho(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t + \lambda_Q\int_0^1\int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_G\int_\Omega \rho(1,\boldsymbol{x})G(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \end{array} \right\}$$

Here $\lambda_G > 0$ is a parameter, $G : \Omega \to \overline{\mathbb{R}}$ gives a preference of the distribution of $\rho(1,\boldsymbol{x})$ and the constraint set $\mathcal{C}(\rho_0)$ is similar to $\mathcal{C}(\rho_0,\rho_1)$:

$$\mathcal{C}(\rho_0) := \mathcal{C} \cap \left\{ \begin{array}{l} (\rho,\boldsymbol{m}) : \partial_t\rho + \mathrm{div}_{\boldsymbol{x}}\,\boldsymbol{m} = 0, \\ \qquad\qquad \boldsymbol{m}\cdot\boldsymbol{n} = 0 \text{ for } \boldsymbol{x}\in\partial\Omega, \rho(0,\cdot) = \rho_0, \end{array} \right\}. \tag{2.12}$$

3. Algorithm

In this section, we first briefly review FISTA algorithm proposed in [10]. Using a first-optimize-then-discretize approach, we describe the FISTA algorithm on variational problem (1.1). After that, we provide the details of our discretization and implementation for the MFP. In the end of this section, we discuss a different approach based on first-discretize-then-optimize strategy which turns out leading to same discrete algorithm.

To solve general nonsmooth convex model

$$\min_{\boldsymbol{z}} u(\boldsymbol{z}) + v(\boldsymbol{z}),$$

where u is a smooth convex function and v is convex but possibly nonsmooth, one can apply proximal gradient method [9, 46].

$$\boldsymbol{z}^{(k+1)} := \mathrm{Prox}_{\eta^{(k)}v}\left(\boldsymbol{z}^{(k)} - \eta^{(k)}\nabla u\left(\boldsymbol{z}^{(k)}\right)\right).$$

Here $\eta^{(k)} > 0$ is the stepsize and the proximal operator is defined as:

$$\mathrm{Prox}_{\eta v}(\boldsymbol{z}) := \operatorname*{argmin}_{\boldsymbol{y}}\left\{ v(\boldsymbol{y}) + \frac{1}{2\eta}\|\boldsymbol{y}-\boldsymbol{z}\|_2^2 \right\}. \tag{3.1}$$

In particular, for an indicator function $\chi_{\mathcal{C}}(\boldsymbol{z}) = \begin{cases} 0, & \boldsymbol{z} \in \mathcal{C} \\ +\infty, & \boldsymbol{z} \notin \mathcal{C} \end{cases}$ of a convex set $\mathcal{C}$, its proximal operator is exactly the projection operator to $\mathcal{C}$, i.e.

$$\mathrm{Prox}_{\eta\chi_{\mathcal{C}}}(\boldsymbol{z}) = \mathrm{Proj}_{\mathcal{C}}(\boldsymbol{z}) = \underset{\boldsymbol{y}\in\mathcal{C}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{z}\|_2^2, \quad \forall \eta > 0.$$

FISTA is essentially an accelerated proximal gradient algorithm [10]. It introduces $\widehat{\boldsymbol{z}}^{(k)}$ as a linear combination of $\boldsymbol{z}^{(k)}$ and $\boldsymbol{z}^{(k-1)}$ in each iteration, and conducts proximal gradient on $\widehat{\boldsymbol{z}}^{(k)}$ to obtain $\boldsymbol{z}^{(k+1)}$. The algorithm is summarized in (3.2), where the stepsizes $\eta^{(k)}$ can either be a constant or be obtained by a backtracking line search.

$$\begin{cases} \boldsymbol{z}^{(k+1)} = \mathrm{Prox}_{\eta^{(k)}v}\left(\widehat{\boldsymbol{z}}^{(k)} - \eta^{(k)}\nabla u(\widehat{\boldsymbol{z}}^{(k)})\right); \\ \tau^{(k+1)} = \dfrac{1}{2}\left(1+\sqrt{1+4\left(\tau^{(k)}\right)^2}\right); \\ \widehat{\boldsymbol{z}}^{(k+1)} = \boldsymbol{z}^{(k+1)} + \dfrac{\tau^{(k)}-1}{\tau^{(k+1)}}\left(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}\right). \end{cases} \tag{3.2}$$

As proved in [10], if $\boldsymbol{z}^* = \operatorname{argmin}_{\boldsymbol{z}} u(\boldsymbol{z}) + v(\boldsymbol{z})$, and $\left\{\boldsymbol{z}^{(k)}\right\}$ is generated by FISTA, then

$$\left[u\left(\boldsymbol{z}^{(k)}\right) + v\left(\boldsymbol{z}^{(k)}\right)\right] - \left[u(\boldsymbol{z}^*) + v(\boldsymbol{z}^*)\right] = \mathcal{O}\left(\frac{1}{(k+1)^2}\right).$$

3.1. **FISTA for MFP.** To apply the above FISTA method to problem (1.1), let us write

$$\min_{\rho,\boldsymbol{m}\in\mathcal{C}(\rho_0,\rho_1)} \mathcal{Y}(\rho,\boldsymbol{m}) := \int_0^1 \int_\Omega Y(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}),\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t, \tag{3.3}$$

where

$$Y(\beta_0,\boldsymbol{\beta},\boldsymbol{x}) = L(\beta_0,\boldsymbol{\beta}) + \lambda_E F_E(\beta_0) + \lambda_Q \beta_0 Q(\boldsymbol{x}). \tag{3.4}$$

For convenience, we write $F_E' = \dfrac{\mathrm{d}}{\mathrm{d}\beta_0}F_E, Y_0 = \dfrac{\partial}{\partial\beta_0}Y$, $\nabla_{\boldsymbol{\beta}}Y = \left(\dfrac{\partial}{\partial\beta_d}Y\right)_{d=1,\cdots,D}$ and $L_0 = \dfrac{\partial}{\partial\beta_0}L, \nabla_{\boldsymbol{\beta}}L = \left(\dfrac{\partial}{\partial\beta_d}L\right)_{d=1,\cdots,D}$. This yields

$$\begin{cases} Y_0(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}),\boldsymbol{x}) = L_0(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x})) + \lambda_E F_E'(\rho(t,\boldsymbol{x})) + \lambda_Q Q(\boldsymbol{x}), \\ \nabla_{\boldsymbol{\beta}}Y(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}),\boldsymbol{x}) = \nabla_{\boldsymbol{\beta}}L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x})), \quad d = 1,\cdots D \end{cases} \tag{3.5}$$

To apply FISTA to this problem, we need to compute the gradients $\delta_\rho\mathcal{Y}(\rho,\boldsymbol{m})$, $\delta_{\boldsymbol{m}}\mathcal{Y}(\rho,\boldsymbol{m})$ and the projection $\mathrm{Proj}_{\mathcal{C}(\rho_0,\rho_1)}(\rho,\boldsymbol{m})$.

Gradient descent. Let the boundary values $\rho(0,\cdot) = \rho_0$, $\rho(1,\cdot) = \rho_1$ and $\boldsymbol{m}(t,\boldsymbol{x})\cdot\boldsymbol{n} = 0$ for $\boldsymbol{x}\in\partial\Omega$ being fixed. By variational calculus, we have

$$\begin{aligned} \delta_\rho\mathcal{Y}(\rho,\boldsymbol{m})(t,\boldsymbol{x}) &= Y_0(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}),\boldsymbol{x}), \\ \delta_{\boldsymbol{m}}\mathcal{Y}(\rho,\boldsymbol{m})(t,\boldsymbol{x}) &= \nabla_{\boldsymbol{\beta}}Y(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}),\boldsymbol{x}). \end{aligned} \tag{3.6}$$

Then with step-size $\eta^{(k)}$, the descent step can be written as

$$\left(\rho^{(k+\frac{1}{2})},\boldsymbol{m}^{(k+\frac{1}{2})}\right) = \left(\widehat{\rho}^{(k)} - \eta^{(k)}\delta_\rho\mathcal{Y}(\widehat{\rho}^{(k)},\widehat{\boldsymbol{m}}^{(k)}), \widehat{\boldsymbol{m}}^{(k)} - \eta^{(k)}\delta_{\boldsymbol{m}}\mathcal{Y}(\widehat{\rho}^{(k)},\widehat{\boldsymbol{m}}^{(k)})\right), \tag{3.7}$$

Projection. The projection step solves the following minimization problem

$$\left(\rho^{(k+1)}, \boldsymbol{m}^{(k+1)}\right) = \underset{\rho,\boldsymbol{m}\in\mathcal{C}(\rho_0,\rho_1)}{\operatorname{argmin}} \frac{1}{2}\left\|\rho-\rho^{(k+\frac{1}{2})}\right\|_{L^2}^2 + \frac{1}{2}\left\|\boldsymbol{m}-\boldsymbol{m}^{(k+\frac{1}{2})}\right\|_{L^2}^2. \tag{3.8}$$

Since the boundary values are fixed and boundary conditions are always satisfied, we only need to introduce dual variable $\phi^{(k+1)}(t,\boldsymbol{x})$, which is C^1 on $[0,1]\times\Omega$, for mass conservation equation $\partial_t\rho + \operatorname{div}_{\boldsymbol{x}}\boldsymbol{m} = 0$. Consider a Lagrangian function

$$\begin{aligned}\mathcal{L}(\rho,\boldsymbol{m},\phi) :&= \frac{1}{2}\left\|\rho-\rho^{(k+\frac{1}{2})}\right\|_{L^2}^2 + \frac{1}{2}\left\|\boldsymbol{m}-\boldsymbol{m}^{(k+\frac{1}{2})}\right\|_{L^2}^2 + \langle\phi, \partial_t\rho + \operatorname{div}_{\boldsymbol{x}}\boldsymbol{m}\rangle\\ &= \frac{1}{2}\left\|\rho-\rho^{(k+\frac{1}{2})}\right\|_{L^2}^2 + \frac{1}{2}\left\|\boldsymbol{m}-\boldsymbol{m}^{(k+\frac{1}{2})}\right\|_{L^2}^2 - \langle\partial_t\phi,\rho\rangle - \langle\nabla_{\boldsymbol{x}}\phi,\boldsymbol{m}\rangle\\ &\quad + \langle\phi(1,\cdot),\rho_1\rangle - \langle\phi(0,\cdot),\rho_0\rangle.\end{aligned} \tag{3.9}$$

$\left(\rho^{(k+1)}, \boldsymbol{m}^{(k+1)}, \phi^{(k+1)}\right)$ is the saddle point of $\mathcal{L}(\rho,\boldsymbol{m},\phi)$ if and only if

$$\begin{cases}\delta_\rho\mathcal{L}\left(\rho^{(k+1)}, \boldsymbol{m}^{(k+1)}, \phi^{(k+1)}\right) = 0,\\ \delta_{\boldsymbol{m}}\mathcal{L}\left(\rho^{(k+1)}, \boldsymbol{m}^{(k+1)}, \phi^{(k+1)}\right) = 0,\\ \delta_\phi\mathcal{L}\left(\rho^{(k+1)}, \boldsymbol{m}^{(k+1)}, \phi^{(k+1)}\right) = 0.\end{cases} \tag{3.10}$$

This yields

$$\begin{cases}\rho^{(k+1)} = \rho^{(k+\frac{1}{2})} + \partial_t\phi^{(k+1)},\\ \boldsymbol{m}^{(k+1)} = \boldsymbol{m}^{(k+\frac{1}{2})} + \nabla_{\boldsymbol{x}}\phi^{(k+1)}.\end{cases} \tag{3.11}$$

and

$$\partial_t\rho^{(k+1)} + \operatorname{div}_{\boldsymbol{x}}\boldsymbol{m}^{(k+1)} = 0. \tag{3.12}$$

Combining (3.11) and (3.12), it is clear that the dual variable $\phi^{(k+1)}$ solves the Poisson equation

$$\begin{cases}-\Delta_{t,\boldsymbol{x}}\phi^{(k+1)}(t,\boldsymbol{x}) = \partial_t\rho^{(k+\frac{1}{2})}(t,\boldsymbol{x}) + \operatorname{div}_{\boldsymbol{x}}\boldsymbol{m}^{(k+\frac{1}{2})}(t,\boldsymbol{x}), & 0\le t\le 1, \boldsymbol{x}\in\Omega\\ \partial_t\phi^{(k+1)}(t,\boldsymbol{x}) = 0, & t = 0,1, \boldsymbol{x}\in\Omega\\ \nabla_{\boldsymbol{x}}\phi^{(k+1)}(t,\boldsymbol{x})\cdot\boldsymbol{n} = 0, & 0\le t\le 1, \boldsymbol{x}\in\partial\Omega,\end{cases} \tag{3.13}$$

Therefore, we can obtain the projection (3.8) in two steps: solving the Poisson equation (3.13) and update $\rho, \boldsymbol{m}$ by (3.11).

The FISTA algorithm for MFP problem (3.3) is summarized in Algorithm 1.

Remark 3.1. To compute the projection, we need to solve a Poisson equation with Neumann boundary conditions (3.13). Since for any $\boldsymbol{x}\in\Omega$, $\rho^{(k+\frac{1}{2})}(0,\boldsymbol{x}) = \rho_0(\boldsymbol{x}), \rho^{(k+\frac{1}{2})}(1,\boldsymbol{x}) = \rho_1(\boldsymbol{x})$ and for any $t\in[0,1], \boldsymbol{x}\in\partial\Omega$, $\boldsymbol{m}^{(k+\frac{1}{2})}(t,\boldsymbol{x})\cdot\boldsymbol{n} = 0$, we

Algorithm 1 FISTA for MFP

Parameters ρ_0, ρ_1

Initialization $\tau^{(1)} = 1$,

$\rho^{(0)}(0,\cdot) = \widehat{\rho}^{(0)}(0,\cdot) = \rho_0,\ \rho^{(0)}(1,\cdot) = \widehat{\rho}^{(0)}(1,\cdot) = \rho_1,$

$\rho^{(0)}(t,\cdot) = \widehat{\rho}^{(0)}(t,\cdot) = 1$ for $0 < t < 1$,

$\boldsymbol{m}^{(0)}(\cdot,\boldsymbol{x}) \cdot \boldsymbol{n} = \widehat{\boldsymbol{m}}^{(0)}(\cdot,\boldsymbol{x}) \cdot \boldsymbol{n} = 0$ for $\boldsymbol{x} \in \partial\Omega$,

$\boldsymbol{m}^{(0)}(\cdot,\boldsymbol{x}) = \widehat{\boldsymbol{m}}^{(0)}(\cdot,\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega\backslash\partial\Omega$.

for $k = 0, 1, 2, \ldots$ **do**

gradient descent

(3.14)
$$\begin{cases} \rho^{(k+\frac{1}{2})}(t,\boldsymbol{x}) = \widehat{\rho}^{(k)}(t,\boldsymbol{x}) - \eta^{(k)} Y_0\left(\widehat{\rho}^{(k)}(t,\boldsymbol{x}), \widehat{\boldsymbol{m}}^{(k)}(t,\boldsymbol{x}), \boldsymbol{x}\right), \\ \qquad\qquad 0 < t < 1, \boldsymbol{x} \in \Omega. \\ \boldsymbol{m}^{(k+\frac{1}{2})}(t,\boldsymbol{x}) = \widehat{\boldsymbol{m}}^{(k)}(t,\boldsymbol{x}) - \eta^{(k)} \nabla_{\boldsymbol{m}} Y\left(\widehat{\rho}^{(k)}(t,\boldsymbol{x}), \widehat{\boldsymbol{m}}^{(k)}(t,\boldsymbol{x}), \boldsymbol{x}\right), \\ \qquad\qquad 0 \le t \le 1, \boldsymbol{x} \in \Omega\backslash\partial\Omega. \end{cases}$$

projection solve $\phi^{(k+1)}$ for

(3.15)
$$\begin{cases} -\Delta_{t,\boldsymbol{x}} \phi^{(k+1)}(t,\boldsymbol{x}) = \partial_t \rho^{(k+\frac{1}{2})}(t,\boldsymbol{x}) + \operatorname{div}_{\boldsymbol{x}} \boldsymbol{m}^{(k+\frac{1}{2})}(t,\boldsymbol{x}), & 0 \le t \le 1, \boldsymbol{x} \in \Omega \\ \partial_t \phi^{(k+1)}(t,\boldsymbol{x}) = 0, & t = 0, 1, \boldsymbol{x} \in \Omega \\ \nabla_{\boldsymbol{x}} \phi^{(k+1)}(t,\boldsymbol{x}) \cdot \boldsymbol{n} = 0, & 0 \le t \le 1, \boldsymbol{x} \in \partial\Omega, \end{cases}$$

and conduct

(3.16)
$$\begin{cases} \rho^{(k+1)} = \rho^{(k+\frac{1}{2})} + \partial_t \phi^{(k+1)}, \\ \boldsymbol{m}^{(k+1)} = \boldsymbol{m}^{(k+\frac{1}{2})} + \nabla_{\boldsymbol{x}} \phi^{(k+1)}. \end{cases}$$

update

$$\tau^{(k+1)} = \frac{1 + \sqrt{1 + 4\left(\tau^{(k)}\right)^2}}{2},$$

$$\omega^{(k)} = \frac{\tau^{(k)} - 1}{\tau^{(k+1)}},$$

(3.17)
$$\left(\widehat{\rho}^{(k+1)}, \widehat{\boldsymbol{m}}^{(k+1)}\right) = \left(1 + \omega^{(k)}\right)\left(\rho^{(k+1)}, \boldsymbol{m}^{(k+1)}\right) - \omega^{(k)}\left(\rho^{(k)}, \boldsymbol{m}^{(k)}\right).$$

end for

have

$$\begin{aligned} &\int_{[0,1]\times\Omega} \partial_t \rho^{(k+\frac{1}{2})}(t,\boldsymbol{x}) + \operatorname{div}_{\boldsymbol{x}} \boldsymbol{m}^{(k+\frac{1}{2})}(t,\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\mathrm{d}t \\ =& \int_\Omega \left(\rho^{(k+\frac{1}{2})}(1,\boldsymbol{x}) - \rho^{(k+\frac{1}{2})}(0,\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} + \int_0^1 \int_{\partial\Omega} \boldsymbol{m}^{(k+\frac{1}{2})}(t,\boldsymbol{x}) \cdot \mathrm{d}\boldsymbol{s}\mathrm{d}t \\ =& \int_\Omega \left(\rho_1^{(k+\frac{1}{2})}(\boldsymbol{x}) - \rho_0^{(k+\frac{1}{2})}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} \\ =& 0, \end{aligned}$$

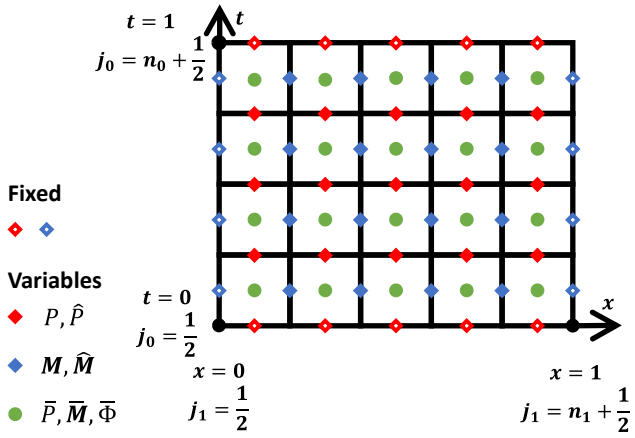


FIGURE 2. Illustration of staggered grids for the case $d = 1$.

This means (3.13) is solvable and the solution is unique up to a constant. In addition, the constant does not count in projection step because in (3.11), we only need $\partial_t \phi^{(k+1)}$ and $\nabla_{\boldsymbol{x}} \phi^{(k+1)}$. Therefore the projection step is well-defined.

3.2. **Discretization and implementation.** For convenience, we here assume $\Omega = [0,1]^D$. Then the boundary conditions of $\boldsymbol{m} = (m_1, \cdots, m_D)$ is provided as:

$$m_d(t, \boldsymbol{x}) = 0, \text{ if } x_d = 0, \text{or } 1, \quad \text{for } d = 1, \cdots, D$$

Consider a uniform grid with n_0 segments on time interval $[0,1]$ and n_d segments on the d-th space dimension. Namely, the mesh size on each dimension is $\Delta_d = \frac{1}{n_d}, d = 0, \cdots, D$, and the staggered grid points are $t_j = (j - \frac{1}{2})\Delta_0, (x_d)_j = (j - \frac{1}{2})\Delta_d$. We use a multi-dimensional index vector $\boldsymbol{j} = (j_0, j_1, \cdots, j_D)$ to indicate a grid point $(t_{j_0}, \boldsymbol{x_j})$, where $\boldsymbol{x_j} := ((x_1)_{j_1}, \cdots, (x_D)_{j_D})$. We further write $u_{\boldsymbol{j}} := u(t_{j_0}, \boldsymbol{x_j})$ the value of function u on the grid point and $U_{\boldsymbol{j}}$ the proposed numerical approximation of $u_{\boldsymbol{j}}$. Our discretization of ρ and $\boldsymbol{m}$ defined on different staggered grids. For convenience, we list the following index sets:

$$\begin{aligned}
J_d &:= \left\{ \frac{3}{2}, \frac{5}{2}, \cdots, n_d - \frac{1}{2} \right\}, \\
\overline{J}_d &:= \{1, 2, \cdots, n_d\}, \\
\boldsymbol{J}_d &:= \overline{J}_0 \times \overline{J}_1 \times \cdots \times \overline{J}_{d-1} \times J_d \times \overline{J}_{d+1} \times \cdots \overline{J}_D, \\
\overline{\boldsymbol{J}} &:= \overline{J}_0 \times \overline{J}_1 \times \cdots \times \overline{J}_D.
\end{aligned}$$

Figure 2 illustrates an 1D example, where $n_0 = 4, n_1 = 5$ and grid points related to $\boldsymbol{J}_0$, $\boldsymbol{J}_1$ and $\overline{\boldsymbol{J}}$ are annotated as red solid diamonds, blue solid diamonds and green solid dots, respectively.

We use $P, \boldsymbol{M}$ and $\overline{\Phi}$ to denote the discretization of $\rho, \boldsymbol{m}$ and ϕ, respectively. They are defined as:

$$\begin{aligned}
P &:= \{P_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\boldsymbol{J}_0} \in \mathbb{V}_0 := \mathbb{R}^{(n_0-1)\times n_1\times\cdots\times n_D},\\
M_d &:= \{(M_d)_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\boldsymbol{J}_d} \in \mathbb{V}_d := \mathbb{R}^{n_0\times n_1\times\cdots\times n_{d-1}\times(n_d-1)\times n_{d+1}\times\cdots\times n_D},\\
\boldsymbol{M} &:= \{M_d\}_{d=1,2,\cdots,D} \in \mathbb{V}_1\times\cdots\mathbb{V}_D,\\
\overline{\Phi} &:= \{\overline{\Phi}_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}} \in \overline{\mathbb{V}} := \mathbb{R}^{n_0\times n_1\times\cdots\times n_D}.
\end{aligned}$$

Moreover, we also define:

$$\begin{aligned}
\overline{P} &:= \{\overline{P}_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}} \in \overline{\mathbb{V}} := \mathbb{R}^{n_0\times n_1\times\cdots\times n_D},\\
\overline{M}_d &:= \{(\overline{M}_d)_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}} \in \overline{\mathbb{V}},\\
\overline{\boldsymbol{M}} &:= \{\overline{M}_d\}_{d=1,2,\cdots,D} \in \overline{\mathbb{V}}^D.
\end{aligned}$$

Based on the above settings, we next discuss details of computing the objective value, implementing gradient descent and conducting the projection step.

Objective value. To compute objective function, we need the value of $\rho(t,\boldsymbol{x})$ and $\boldsymbol{m}(t,\boldsymbol{x})$ on the same point $(t,\boldsymbol{x})$. While P, M_d are defined on different grids, a natural idea is to transform them to the same central grid $\overline{\boldsymbol{J}}$ first. For convenience, let $M_0 \equiv P$. We can define the average operators as:

$$\mathcal{A}_d : \mathbb{V}_d \to \overline{\mathbb{V}}, \quad M_d \mapsto \overline{M}_d = \mathcal{A}_d(M_d), \quad \text{for } d = 0, 1, \cdots, D,$$

$$(\overline{M}_d)_{\boldsymbol{j}} := \begin{cases} \dfrac{1}{2}(M_d)_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}}, & j_d = 1,\\ \dfrac{1}{2}\left((M_d)_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}} + (M_d)_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}\right), & j_d = 2,3,\cdots,n_d-1, \quad \forall \boldsymbol{j}\in\overline{\boldsymbol{J}}\\ \dfrac{1}{2}(M_d)_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}, & j_d = n_d. \end{cases}$$

$$\mathcal{A} : \mathbb{V}_1\times\cdots\mathbb{V}_D \to \overline{\mathbb{V}}^D, \quad \boldsymbol{M} \mapsto \{\mathcal{A}_d(M_d)\}_{d=1,\cdots,D}.$$

where $\boldsymbol{e}_d \in \mathbb{R}^{D+1}$ has 1 in the $(d+1)$-th entry and 0 elsewhere, $d = 0, \cdots, D$. The boundary conditions of $\boldsymbol{M}$ are implicitly included in the average operator. We further define $\overline{P}_{\mathcal{A}} \in \overline{\mathbb{V}}$ to indicate density boundary conditions,

$$(\overline{P}_{\mathcal{A}})_{\boldsymbol{j}} := \begin{cases} \dfrac{1}{2}\rho_0(\boldsymbol{x}_{\boldsymbol{j}}), & j_0 = 1,\\ 0, & j_0 = 2,3,\cdots,n_0-1, \qquad \forall \boldsymbol{j}\in\overline{\boldsymbol{J}}\\ \dfrac{1}{2}\rho_1(\boldsymbol{x}_{\boldsymbol{j}}), & j_0 = n_0. \end{cases}$$

As an example, in Figure 2, $\mathcal{A}_0$ maps the red solid dots to the green dots. $\mathcal{A}_1$ maps the blue solid dots to the green dots, and the red hollow dots contribute to the non-zero entries of $\overline{P}_{\mathcal{A}}$.

Now, we are ready to evaluate the objective function by averaging P and M from their staggered grids to the central grid. Namely, We define $\overline{P}, \overline{M} \in \overline{\mathbb{V}}$ as

$$\overline{P} := \mathcal{A}_0(P) + \overline{P}_{\mathcal{A}}, \quad \overline{\boldsymbol{M}} := \mathcal{A}(\boldsymbol{M}) \tag{3.18}$$

then we approximate the objective value by

$$\left(\prod_{d=0}^{D}\Delta_d\right)\mathcal{Y}(P,\boldsymbol{M}),$$

where

$$\mathcal{Y}(P,\boldsymbol{M}) := \overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}}) := \sum_{\boldsymbol{j}\in\overline{\boldsymbol{J}}} Y\left(\overline{P}_{\boldsymbol{j}},\overline{\boldsymbol{M}}_{\boldsymbol{j}},\boldsymbol{x}_{\boldsymbol{j}}\right). \tag{3.19}$$

Gradient descent. To fulfil gradient descent, we first average $(P,\boldsymbol{M})$ from different grids $\boldsymbol{J}_d$ to grid $\overline{\boldsymbol{J}}$ by (3.18) and compute gradient values pointwisely

$$\begin{aligned}
&\partial_{\overline{P}}\overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}}) := \left\{Y_0\left(\overline{P}_{\boldsymbol{j}},\overline{\boldsymbol{M}}_{\boldsymbol{j}},\boldsymbol{x}_{\boldsymbol{j}}\right)\right\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}},\\
&\partial_{\overline{M}_d}\overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}}) := \left\{Y_d\left(\overline{P}_{\boldsymbol{j}},\overline{\boldsymbol{M}}_{\boldsymbol{j}},\boldsymbol{x}_{\boldsymbol{j}}\right)\right\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}}, \quad d=1,\cdots,D,\\
&\partial_{\overline{\boldsymbol{M}}}\overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}}) := \left\{\partial_{\overline{M}_d}\overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}})\right\}_{d=1,\cdots,D}.
\end{aligned} \tag{3.20}$$

Then we average gradient values back to different grids $\boldsymbol{J}_d$. Defining another sets of average operator as

$$\begin{aligned}
&\mathcal{A}_d^*:\overline{\mathbb{V}}\to\mathbb{V}_d, \overline{M}_d\mapsto M_d,\ (M_d)_{\boldsymbol{j}} := \frac{1}{2}\left((\overline{M}_d)_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}} + (\overline{M}_d)_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}\right),\\
&\mathcal{A}^*:\overline{\mathbb{V}}^D\to\mathbb{V}_1\times\cdots\mathbb{V}_D, \overline{\boldsymbol{M}}\mapsto\{\mathcal{A}_d^*(\overline{M}_d)\}_{d=1,\cdots,D},
\end{aligned}$$

we obtain desired gradient values:

$$\begin{aligned}
&\partial_P\mathcal{Y}(P,\boldsymbol{M}) = \mathcal{A}_0^*\left(\partial_{\overline{P}}\overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}})\right),\\
&\partial_{\boldsymbol{M}}\mathcal{Y}(P,\boldsymbol{M}) = \mathcal{A}^*\left(\partial_{\overline{\boldsymbol{M}}}\overline{\mathcal{Y}}(\overline{P},\overline{\boldsymbol{M}})\right).
\end{aligned} \tag{3.21}$$

Combining (3.20),(3.21), we can implement gradient descent step (3.14) on discrete meshes by:

$$\left(P^{(k+\frac{1}{2})},\boldsymbol{M}^{(k+\frac{1}{2})}\right) = \left(\widehat{P}^{(k)},\widehat{\boldsymbol{M}}^{(k)}\right) - \eta^{(k)}\left(\partial_P\mathcal{Y}(\widehat{P}^{(k)},\widehat{\boldsymbol{M}}^{(k)}),\partial_{\boldsymbol{M}}\mathcal{Y}(\widehat{P}^{(k)},\widehat{\boldsymbol{M}}^{(k)})\right), \tag{3.22}$$

Projection. To compute projection, we use the finite difference method to discretize the corresponding differential operators in the PDE constraint. We first define discrete partial derivative:

$$\mathcal{D}_d:\mathbb{V}_d\to\overline{\mathbb{V}},\quad M_d\mapsto\mathcal{D}_d(M_d),\quad \text{for } d=0,\cdots,D$$

$$(\mathcal{D}_d(M_d))_{\boldsymbol{j}} := \begin{cases}
\dfrac{1}{\Delta_d}(M_d)_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}}, & j_d=1,\\
\dfrac{1}{\Delta_d}\left((M_d)_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}} - (M_d)_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}\right), & j_d=2,3,\cdots,n_d-1,\\
-\dfrac{1}{\Delta_d}(M_d)_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}, & j_d=n_d,
\end{cases}$$

discrete divergence:

$$\mathrm{Div}:\mathbb{V}_0\times\mathbb{V}_1\times\cdots\mathbb{V}_D\to\overline{\mathbb{V}},\quad (P,\boldsymbol{M})\mapsto\mathcal{D}_0(P)+\sum_{d=1}^{D}\mathcal{D}_d(M_d),$$

and the term $\overline{P}_{\mathcal{D}}\in\overline{\mathbb{V}}$ to impose boundary conditions:

$$(\overline{P}_{\mathcal{D}})_{\boldsymbol{j}} := \begin{cases}
-\dfrac{1}{\Delta_0}\rho_0\left(\boldsymbol{x}_{\boldsymbol{j}}\right), & j_0=1,\\
0, & j_0=2,3,\cdots,n_0-1,,\\
\dfrac{1}{\Delta_0}\rho_1\left(\boldsymbol{x}_{\boldsymbol{j}}\right), & j_0=n_0.
\end{cases}$$

Then the RHS of first equation in (3.15) can be approximated by

$$\mathrm{Div}\left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right) + \overline{P}_{\mathcal{D}}.$$

We approximate ∂_d with a central difference and ∂_{dd} with a three-point stencil finite difference. By homogenenous Neumann boundary condition, we have discrete second-order derivative operators

$$\begin{aligned}
&\mathcal{D}_{dd} : \overline{\mathbb{V}} \to \overline{\mathbb{V}}, \quad \overline{\Phi} \mapsto \mathcal{D}_{dd}(\overline{\Phi}),\\
&\left(\mathcal{D}_{dd}(\overline{\Phi})\right)_{\boldsymbol{j}} := \begin{cases} \frac{1}{\Delta_d^2}\left(-\overline{\Phi}_{\boldsymbol{j}} + \overline{\Phi}_{\boldsymbol{j}+\boldsymbol{e}_d}\right), & j_d = 1,\\ \frac{1}{\Delta_d^2}\left(\overline{\Phi}_{\boldsymbol{j}-\boldsymbol{e}_d} - 2\overline{\Phi}_{\boldsymbol{j}} + \overline{\Phi}_{\boldsymbol{j}+\boldsymbol{e}_d}\right), & j_d = 2, 3, \cdots, n_d - 1,\\ \frac{1}{\Delta_d^2}\left(\overline{\Phi}_{\boldsymbol{j}-\boldsymbol{e}_d} - \overline{\Phi}_{\boldsymbol{j}}\right), & j_d = n_d, \end{cases}\\
&\mathrm{Lap} : \overline{\mathbb{V}} \to \overline{\mathbb{V}}, \quad \overline{\Phi} \mapsto \mathcal{D}_{00}(\overline{\Phi}) + \sum_{d=1}^{D} \mathcal{D}_{dd}(\overline{\Phi}).
\end{aligned}$$

The Poisson equation (3.15) on grids is therefore

$$-\mathrm{Lap}\left(\overline{\Phi}^{(k+1)}\right) = \mathrm{Div}\left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right) + \overline{P}_{\mathcal{D}}, \tag{3.23}$$

Defining another set of derivative operators

$$\begin{aligned}
&\mathcal{D}_d^* : \overline{\mathbb{V}} \to \mathbb{V}_d, \overline{\Phi} \mapsto \mathcal{D}_d^*(\overline{\Phi}),\ (\mathcal{D}_d^*(\overline{\Phi}))_{\boldsymbol{j}} := \frac{1}{\Delta_d}\left((\overline{\Phi})_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}} - \overline{\Phi}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}\right)\\
&\mathrm{Grad} : \overline{\mathbb{V}} \to \mathbb{V}_0 \times \mathbb{V}_1 \times \cdots \mathbb{V}_D, \overline{\Phi} \mapsto \left\{\mathcal{D}_d^*(\overline{\Phi})\right\}_{d=0,1,\cdots,D},
\end{aligned}$$

we obtain the second step of projection, the discretization of (3.16):

$$\left(P^{(k+1)}, \boldsymbol{M}^{(k+1)}\right) = \left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right) + \mathrm{Grad}\left(\overline{\Phi}^{(k+1)}\right). \tag{3.24}$$

Combining above ingredients, we have FISTA for MFP on discrete mesh summarized in Algorithm 2.

Remark 3.2. The discrete operators $\mathrm{Div}, \mathrm{Grad}$ and Lap are consistent in the following sense. For space $\overline{\mathbb{V}}$ and $\mathbb{V}_0 \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_D$, if we view the elements $\overline{\Phi}$ and $(P, \boldsymbol{M})$ as long vectors, we can define the inner product as

$$\begin{aligned}
\left\langle \overline{\Phi}^1, \overline{\Phi}^2 \right\rangle &:= \sum_{\boldsymbol{j} \in \overline{\boldsymbol{J}}} \overline{\Phi}^1_{\boldsymbol{j}} \overline{\Phi}^2_{\boldsymbol{j}},\\
\left\langle (P^1, \boldsymbol{M}^1), (P^2, \boldsymbol{M}^2) \right\rangle &:= \sum_{\boldsymbol{j} \in \boldsymbol{J}_0} P^1_{\boldsymbol{j}} P^2_{\boldsymbol{j}} + \sum_{d=1}^{D} \sum_{\boldsymbol{j} \in \boldsymbol{J}_d} (M_d^1)_{\boldsymbol{j}} (M_d^2)_{\boldsymbol{j}}.
\end{aligned} \tag{3.29}$$

and define the induced norm as $\|\cdot\|_F$. Then simple calculation shows that for any $\overline{\Phi} \in \overline{\mathbb{V}}$ and $(P, \boldsymbol{M}) \in \mathbb{V}_0 \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_D$, the following equation holds

$$\begin{aligned}
&\mathrm{Lap}\left(\overline{\Phi}\right) = \mathrm{Div} \circ \mathrm{Grad}\left(\overline{\Phi}\right),\\
&\left\langle -\mathrm{Grad}(\overline{\Phi}), (P, \boldsymbol{M}) \right\rangle = \left\langle \overline{\Phi}, \mathrm{Div}(P, \boldsymbol{M}) \right\rangle,
\end{aligned} \tag{3.30}$$

These match the relations between $\mathrm{div}_{t,\boldsymbol{x}}, \nabla_{t,\boldsymbol{x}}$ and $\Delta_{t,\boldsymbol{x}}$ on continuous spaces.

Algorithm 2 FISTA for MFP on discrete mesh

Parameters ρ_0, ρ_1

Initialization $\tau^{(1)} = 1$, $P^{(0)} = \widehat{P}^{(0)} = \mathbf{1}$, and $M_d^{(0)} = \widehat{M_d}^{(0)} = \mathbf{1}$.

for $k = 0, 1, 2, \ldots$ **do**

gradient descent

$$\begin{cases} P^{(k+\frac{1}{2})} = \widehat{P}^{(k)} - \eta^{(k)} \partial_P \mathcal{Y}(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}), \\ \boldsymbol{M}^{(k+\frac{1}{2})} = \widehat{\boldsymbol{M}}^{(k)} - \eta^{(k)} \partial_{\boldsymbol{M}} \mathcal{Y}(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}), \end{cases} \tag{3.25}$$

projection solve $\overline{\Phi}^{(k+1)}$ for

$$-\operatorname{Lap}\left(\overline{\Phi}^{(k+1)}\right) = \operatorname{Div}\left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right) + \overline{P}_{\mathcal{D}}, \tag{3.26}$$

and conduct

$$\left(P^{(k+1)}, \boldsymbol{M}^{(k+1)}\right) = \left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right) + \operatorname{Grad}\left(\overline{\Phi}^{(k+1)}\right). \tag{3.27}$$

update

$$\tau^{(k+1)} = \frac{1 + \sqrt{1 + 4(\tau^{(k)})^2}}{2},$$

$$\omega^{(k)} = \frac{\tau^{(k)} - 1}{\tau^{(k+1)}},$$

$$\left(\widehat{P}^{(k+1)}, \widehat{\boldsymbol{M}}^{(k+1)}\right) = \left(1 + \omega^{(k)}\right)\left(P^{(k+1)}, \boldsymbol{M}^{(k+1)}\right) - \omega^{(k)}\left(P^{(k)}, \boldsymbol{M}^{(k)}\right). \tag{3.28}$$

end for

Remark 3.3. Directly solving the large linear system (3.26) could be very expensive. Thanks to the special structure of the operator Lap, there exist $\lambda^{\mathbf{1}} = 0$, $\lambda^{\boldsymbol{i}} < 0, \boldsymbol{i} \in \overline{\boldsymbol{J}} \backslash \{\mathbf{1}\}$ and an orthonormal basis $\left\{\Psi^{\boldsymbol{i}}\right\}_{\boldsymbol{i} \in \overline{\boldsymbol{J}}}$ of $(\overline{\mathbb{V}}, \|\cdot\|_F)$ such that

$$\operatorname{Lap}(\overline{\Phi}) := \sum_{\boldsymbol{i} \in \overline{\boldsymbol{J}}} \lambda_{\boldsymbol{i}} \left\langle \overline{\Phi}, \Psi^{\boldsymbol{i}} \right\rangle \Psi^{\boldsymbol{i}}. \tag{3.31}$$

Therefore one way to define the inverse of Lap is

$$\operatorname{Lap}^{-1}(\overline{\Phi}) := \sum_{\boldsymbol{i} \in \overline{\boldsymbol{J}} \backslash \{\mathbf{1}\}} \frac{1}{\lambda_{\boldsymbol{i}}} \left\langle \overline{\Phi}, \Psi^{\boldsymbol{i}} \right\rangle \Psi^{\boldsymbol{i}}. \tag{3.32}$$

This leads to a discrete cosine transform method to solve (3.26).

To derive the discrete Algorithm 2, we optimize the continuous problem (3.3) by Algorithm 1 then discretize the algorithm. This is a first-optimize-then-discretize approach. We can also consider a first-discretize-then-optimize approach. In fact, using our proposed discretization for MFP, the two approaches lead the same algorithm, as illustrated in Figure 3.2. This is mainly because of the consistent relation of discrete operators discussed in Remark 3.2.

Based on previous notations, we discretize the original problem (3.3) to

$$\min_{(P, \boldsymbol{M}) \in \mathcal{C}(\overline{P}_{\mathcal{D}})} \mathcal{Y}(P, \boldsymbol{M}) := \sum_{\boldsymbol{j} \in \overline{\boldsymbol{J}}} Y\left((\mathcal{A}_0(P) + \overline{P}_{\mathcal{A}})_{\boldsymbol{j}}, \mathcal{A}(\boldsymbol{M})_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}}\right), \tag{3.33}$$

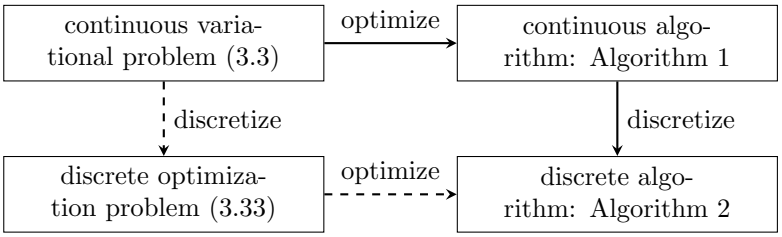


FIGURE 3. Equivalent approaches to obtain the discrete Algorithm 2

where the constraints are linear and the constraint set is convex:

$$\mathcal{C}(\overline{P}_{\mathcal{D}}) := \left\{(P, \boldsymbol{M}) : \mathcal{D}_0(P) + \overline{P}_{\mathcal{D}} + \mathrm{Div}(\boldsymbol{M}) = \mathbf{0}\right\}. \tag{3.34}$$

To optimize the problem with FISTA, we first compute gradient. For any $P, \boldsymbol{M}$, we define the corresponding values on $\overline{\boldsymbol{J}}$ by $\overline{P} := \mathcal{A}_0(P) + \overline{P}_{\mathcal{A}}, \overline{\boldsymbol{M}} := \mathcal{A}(\boldsymbol{M})$ be , then

$$\mathcal{Y}(P, \boldsymbol{M}) = \sum_{\boldsymbol{j} \in \overline{\boldsymbol{J}}} Y\left(\overline{P}_{\boldsymbol{j}}, \overline{\boldsymbol{M}}_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}}\right). \tag{3.35}$$

We will have (3.20) by taking partial derivatives w.r.t $\overline{P}, \overline{\boldsymbol{M}}$, and then (3.21) by chain rule. Therefore the gradient descent step is exactly (3.25). For projection, based on the inner product defined as (3.29) and induced norm, we can formulate the Lagrangian as

$$\mathcal{L}(P, M, \overline{\Phi}) := \frac{1}{2}\left\|(P, M) - \left(P^{(k+\frac{1}{2})}, M^{(k+\frac{1}{2})}\right)\right\|_F^2 + \left\langle \overline{\Phi}^{(k+1)}, \mathrm{Div}(P, \boldsymbol{M}) + \overline{P}_D \right\rangle. \tag{3.36}$$

Because of the consistency of the discrete operators (3.30), we know that (3.26),(3.27) computes the projection to $\mathcal{C}(\overline{P}_{\mathcal{D}})$. Therefore the FISTA algorithm to the discrete MFP problem (3.33) is exactly Algorithm 2.

4. Convergence

One major difference between Algorithm 1 and Algorithm 2 is that the former one is for the continuous setup while the latter one is for a given discretized mesh grid although both algorithms provide convergence sequences according to the FISTA theory. It is natural to ask if the discretized solution can converge to the continuous solution when the mesh grid size Δ_d goes to zero. Specifically, with a given step-size sequence $\left\{\eta^{(k)}\right\}_k$, let the sequences $\left\{\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)}\right)\right\}_k$, $\left\{\left(\rho^{(k)}, \boldsymbol{m}^{(k)}\right), \left(\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}\right)\right\}_k$ be obtained from Algorithm 1, and $\left\{\left(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}\right)\right\}_k$, $\left\{\left(P^{(k)}, \boldsymbol{M}^{(k)}\right), \left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right)\right\}_k$ from Algorithm 2. If $\left(\rho^{(k)}, \boldsymbol{m}^{(k)}\right) \to (\rho^*, \boldsymbol{m}^*)$ and $\left(P^{(k)}, \boldsymbol{M}^{(k)}\right) \to (P^*, \boldsymbol{M}^*)$ as $k \to \infty$, we would like to check whether $(P^*, \boldsymbol{M}^*)$ converge to $(\rho^*, \boldsymbol{m}^*)$ as the mesh grid size converge to zero. In this section, we theoretically analyze and provide a positive answer to this question under certain conditions.

We first introduce some notations. With given step-size sequence $\{\eta^{(k)}\}_k$, let $\left\{\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)}\right)\right\}_k$, $\left\{\left(\rho^{(k)}, \boldsymbol{m}^{(k)}\right), \left(\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}\right)\right\}_k$ be obtained from Algorithm 1. With the same step-size sequence and initialization $P^{(0)} = \widehat{P}^{(0)} = \rho^{(0)}_{\boldsymbol{J}_0}$, $\boldsymbol{M}^{(0)} =$

$\widehat{\boldsymbol{M}}^{(0)} = \boldsymbol{m}_{\boldsymbol{J}}^{(0)}$, let $\left\{\left(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}\right)\right\}_k$, $\left\{\left(P^{(k)}, \boldsymbol{M}^{(k)}\right), \left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right)\right\}_k$ be obtained from Algorithm 2. For any index set $\boldsymbol{J}_d, \overline{\boldsymbol{J}}$, we write the continuous functions ρ and $\boldsymbol{m}$ evaluating on corresponding discrete grids as

$$\rho_{\boldsymbol{J}_0} := \{\rho_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\boldsymbol{J}_0}, \quad (m_d)_{\boldsymbol{J}_d} := \{(m_d)_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\boldsymbol{J}_d}, \quad \boldsymbol{m}_{\boldsymbol{J}} := \{(m_d)_{\boldsymbol{J}_d}\}_{d=1,2,\cdots,D},$$
$$\rho_{\overline{\boldsymbol{J}}} := \{\rho_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}}, \quad (m_d)_{\overline{\boldsymbol{J}}} := \{(m_d)_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\overline{\boldsymbol{J}}}, \quad \boldsymbol{m}_{\overline{\boldsymbol{J}}} := \left\{(m_d)_{\overline{\boldsymbol{J}}}\right\}_{d=1,2,\cdots,D}.$$

Let $m_0 \equiv \rho, M_0 \equiv P$. For any $k = 0, \frac{1}{2}, 1, 1+\frac{1}{2}, \cdots$, we define the error on grid points $\boldsymbol{J}_d$ by

$$\begin{aligned}
E_d^{(k)} &:= M_d^{(k)} - \left(m_d^{(k)}\right)_{\boldsymbol{J}_d}, \quad d = 0, \cdots, D\\
\boldsymbol{E}_{\boldsymbol{m}}^{(k)} &:= \left\{E_d^{(k)}\right\}_{d=1,\cdots,D},\\
\boldsymbol{E}^{(k)} &:= \left\{E_0^{(k)}, \boldsymbol{E}_{\boldsymbol{m}}^{(k)}\right\}.
\end{aligned}$$

Similarly, for $k = 0, 1, 2, \cdots$, we define

$$\begin{aligned}
\widehat{E}_d^{(k)} &:= \widehat{M}_d^{(k)} - \left(\widehat{m_d}^{(k)}\right)_{\boldsymbol{J}_d}, \quad d = 0, \cdots, D\\
\widehat{E}_{\boldsymbol{m}}^{(k)} &:= \left\{\widehat{E}_d^{(k)}\right\}_{d=1,\cdots,D},\\
\widehat{\boldsymbol{E}}^{(k)} &:= \left\{\widehat{E}_0^{(k)}, \widehat{\boldsymbol{E}}_{\boldsymbol{m}}^{(k)}\right\}\\
E_\phi^{(k)} &:= \overline{\Phi}^{(k)} - \phi_{\overline{\boldsymbol{J}}}^{(k)},
\end{aligned}$$

Recall that in Remark 3.2, we introduce induced norm $\|\cdot\|_F$ on space $\mathbb{V}_0 \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_D$ and $\overline{\mathbb{V}}$ as

$$\|\boldsymbol{E}\|_F := \left(\sum_{d=0}^{D}\sum_{\boldsymbol{j}\in\boldsymbol{J}_d}(E_d)_{\boldsymbol{j}}^2\right)^{\frac{1}{2}}, \quad \|E_\phi\|_F := \left(\sum_{\boldsymbol{j}\in\overline{\boldsymbol{J}}}(E_\phi)_{\boldsymbol{j}}^2\right)^{\frac{1}{2}}.$$

We here define 2-norm $\|\cdot\|_2$ as

$$\|\cdot\|_2 := \left(\prod_{d=0}^{D}\Delta_d\right)^{\frac{1}{2}}\|\cdot\|_F.$$

Next we propose several assumptions before stating the main theorem.

Assumption 1. Let ρ_0, ρ_1 be given initial and terminal densities. With above notations, we assume the following conditions hold for any $k = 0, 1, \cdots$,

(1) $\rho_0, \rho_1, \rho^{(k)}, \boldsymbol{m}^{(k)}, \rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)}$, are C^2, and $\phi^{(k)}$ are C^3,
(2) There exist $\underline{\rho} \le \overline{\rho}$, such that $\widehat{\rho}^{(k)}(t,\boldsymbol{x}), \widehat{P}_{\boldsymbol{j}}^{(k)} \in [\underline{\rho}, \overline{\rho}]$,
(3) $\widehat{\boldsymbol{m}}^{(k)}(t,\boldsymbol{x}), \widehat{M}_{\boldsymbol{j}}^{(k)} \in \Omega_{\boldsymbol{m}} \subset \mathbb{R}^D$,
(4) Y_d's are C_Y-Lipschitz continuous on $[\underline{\rho}, \overline{\rho}] \times \Omega_{\boldsymbol{m}} \times [0,1]^D$, i.e. for $d = 0, \cdots, D$ and any $(\beta_0^1, \boldsymbol{\beta}^1, \boldsymbol{x}^1), (\beta_0^2, \boldsymbol{\beta}^2, \boldsymbol{x}^2) \in [\underline{\rho}, \overline{\rho}] \times \Omega_{\boldsymbol{m}} \times [0,1]^D$,

$$\left|Y_d(\beta_0^1, \boldsymbol{\beta}^1, \boldsymbol{x}^1) - Y_d(\beta_0^2, \boldsymbol{\beta}^2, \boldsymbol{x}^2)\right| \le C_Y \left\|(\beta_0^1, \boldsymbol{\beta}^1, \boldsymbol{x}^1) - (\beta_0^2, \boldsymbol{\beta}^2, \boldsymbol{x}^2)\right\|_1. \tag{4.1}$$

Remark 4.1. Assumption 1 is accessible for very general cases. In fact, when $\rho_0, \rho_1, \rho^{(0)}, \boldsymbol{m}^{(0)}$ are C^2 and Y is C^1, one can show that assumption 1 holds by induction on k. And assumptions 2 and 3 are true as long as $\left\{\rho^{(k)}, \boldsymbol{m}^{(k)}\right\}$ and $\left\{P^{(k)}, \boldsymbol{M}^{(k)}\right\}$ converges. With a typical choice $Y(\beta_0, \boldsymbol{\beta}, \boldsymbol{x}) = L(\beta_0, \boldsymbol{\beta})$ where L is defined in (2.5), we retrieve the optimal transport problem. Both (3.3) and (3.33) have unique minimizers $\{\rho^*, m^*\}$ and $\{P^*, M^*\}$ and both algorithms converges. If in addition ρ_0, ρ_1 are C^2 and $\min_{\boldsymbol{x}}\{\rho_0(\boldsymbol{x}), \rho_1(\boldsymbol{x})\} \geq \underline{\rho} > 0$, then Assumption 1 hold with continuous initialization and carefully chosen step-sizes. We also would like to point out that the current analysis is based on the smoothness assumptions. We admit that for ρ_0, ρ_1 that are only Lebesgue measurable, our proof is not applicable. The convergence under weaker assumptions and more general spaces is an interesting topic to be explored in the future.

We now state our main theorem which characterizes the error bound with respect the grid size.

Theorem 4.2. *If Assumption 1 hold for $k = 0, 1, \cdots$, then*

$$\left\|\boldsymbol{E}^{(k)}\right\|_2 \leq C\left(\sum_{d=0}^{D} \Delta_d\right) = \mathcal{O}\left(\sum_{d=0}^{D} \Delta_d\right). \tag{4.2}$$

Here C is a constant depending on dimension D, Lipschitz constant C_Y, stepsizes $\{\eta^{(s)}\}_{s=1,\cdots,k}$ and sequences $\{\widehat{\rho}^{(s)}, \widehat{\boldsymbol{m}}^{(s)}\}_{s=1,\cdots,k}$ but it is independent of $\{\Delta_d\}_{d=0,\cdots,D}$.

Remark 4.3. This theoretical bound is not sharp as the constant C in the worst case is not bounded in term of k. This suggests that we may have to choose extremely fine grid to accommodate the size of C. However, our numerical results will show that a reasonably fine mesh is good enough to achieve satisfactory accuracy.

Note that the above theorem analyzes error bounds at each iteration along optimization paths from the continuous setup and its discretized counterpart. Consequently, we can have the following convergence analysis if both sequences from the continuous and discretized optimization converge (i.e. choice of the step size satisfies convergence conditions used in FISTA [10]).

Corollary 4.4. *Suppose that $\left\{\left(P^{(k)}, \boldsymbol{M}^{(k)}\right)\right\}_k$ and $\left\{\left(\rho^{(k)}, \boldsymbol{m}^{(k)}\right)\right\}_k$ satisfy all conditions in Theorem 4.2. If in addition, there exist $(P^*, \boldsymbol{M}^*)$, $(\rho^*, \boldsymbol{m}^*)$ such that $\rho^* \in C^1, m_d^* \in C^1$ and*

$$\begin{aligned} \lim_{k\to\infty} \left\|\left(P^{(k)}, \boldsymbol{M}^{(k)}\right) - (P^*, \boldsymbol{M}^*)\right\|_2 &= 0, \\ \lim_{k\to\infty} \left\|\left(\rho^{(k)}, \boldsymbol{m}^{(k)}\right) - (\rho^*, \boldsymbol{m}^*)\right\|_{L_2} &= 0, \end{aligned} \tag{4.3}$$

where $\|\cdot\|_{L_2}$ denotes the standard L_2-norm in the function space. Let $\Delta = \max_{d=0,\cdots,D} \Delta_d$, then

$$\lim_{\Delta\to 0} \|\boldsymbol{E}^*\|_2 := \lim_{\Delta\to 0} \left\|(P^*, \boldsymbol{M}^*) - \left(\rho^*_{\boldsymbol{J}_0}, \boldsymbol{m}^*_{\boldsymbol{J}}\right)\right\|_2 = 0. \tag{4.4}$$

Proof. By triangular inequality,

$$\begin{aligned} &\left\|(P^*, \boldsymbol{M}^*) - (\rho^*_{\boldsymbol{J}_0}, \boldsymbol{m}^*_{\boldsymbol{J}})\right\|_2 \\ \leq &\left\|\left(P^{(k)}, \boldsymbol{M}^{(k)}\right) - (P^*, \boldsymbol{M}^*)\right\|_2 + \left\|\boldsymbol{E}^{(k)}\right\|_2 + \left\|\left(\rho^{(k)}_{\boldsymbol{J}_0}, \boldsymbol{m}^{(k)}_{\boldsymbol{J}}\right) - (\rho^*_{\boldsymbol{J}_0}, \boldsymbol{m}^*_{\boldsymbol{J}})\right\|_2 \end{aligned}$$

For any $\epsilon > 0$, there exists k_ϵ such that

$$\begin{aligned} \left\| \left(P^{(k_\epsilon)}, \boldsymbol{M}^{(k_\epsilon)} \right) - (P^*, \boldsymbol{M}^*) \right\|_2 &\leq \frac{\epsilon}{4}, \\ \left\| \left(\rho^{(k_\epsilon)}, \boldsymbol{m}^{(k_\epsilon)} \right) - (\rho^*, \boldsymbol{m}^*) \right\|_{L_2} &\leq \frac{\epsilon}{4} \end{aligned} \tag{4.5}$$

By numerical integration, there exists a constant C_1 depending on $d, \rho^{(k_\epsilon)}, \boldsymbol{m}^{(k_\epsilon)}$, $\rho^*, \boldsymbol{m}^*$ and independent of Δ_d such that

$$\begin{aligned} &\left\| \left(\rho_{\boldsymbol{J}_0}^{(k_\epsilon)}, \boldsymbol{m}_{\boldsymbol{J}}^{(k_\epsilon)} \right) - \left(\rho_{\boldsymbol{J}_0}^*, \boldsymbol{m}_{\boldsymbol{J}}^* \right) \right\|_2^2 \\ &\leq \int_0^1 \int_\Omega \left\| \left(\rho^{(k_\epsilon)}, \boldsymbol{m}^{(k_\epsilon)} \right) - (\rho^*, \boldsymbol{m}^*) \right\|_2^2 \mathrm{d}\boldsymbol{x}\mathrm{d}t + C_1 \sum_{d=0}^{D} \Delta_d, \end{aligned} \tag{4.6}$$

By Theorem 4.2, there exist a constant C_2 independent of Δ_d such that

$$\left\| \boldsymbol{E}^{(k_\epsilon)} \right\|_2 \leq C_2 \sum_{d=0}^{D} \Delta_d.$$

Let $\delta = \dfrac{\epsilon}{(D+1)(|C_1| + |C_2|)}$.Then for any Δ_d satisfying $\max_{d=0,\cdots,D} \Delta_d \leq \delta$, we have

$$\left| C_1 \sum_{d=0}^{D} \Delta_d \right| + \left| C_2 \sum_{d=0}^{D} \Delta_d \right| \leq \frac{\epsilon}{2}, \tag{4.7}$$

Combining (4.5), (4.6) and (4.7), we conclude that for any $\epsilon > 0$, there exist δ with all $\{\Delta_d\}_d$ satisfying $\Delta \leq \delta$ such that $\|\boldsymbol{E}^*\|_2 \leq \epsilon$. □

To prove Theorem 4.2, we need to establish three lemmas to analyse the error introduced in each main steps of the algorithm. After that, the proof of Theorem 4.2 can be obtained by induction.

Lemma 4.5. *If Assumption 1 hold for $k = 0, 1, \cdots$, then*

$$\left\| \boldsymbol{E}^{(k+\frac{1}{2})} \right\|_2 \leq C(D, C_Y, \eta^{(k)}) \left\| \widehat{\boldsymbol{E}}^{(k)} \right\|_2 + C \left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) \left(\sum_{d=0}^{D} \Delta_d \right) + \mathcal{O} \left(\sum_{d=0}^{D} \Delta_d^2 \right). \tag{4.8}$$

Proof. By definition of $E_d^{(k+\frac{1}{2})}$, we substitute discrete variables in (3.25) by the sum of error and continuous variables. This leads to

$$\left(E_d^{(k+\frac{1}{2})} \right)_{\boldsymbol{j}} + \left(m_d^{(k+\frac{1}{2})} \right)_{\boldsymbol{j}} = \left(\widehat{E}_d^{(k)} \right)_{\boldsymbol{j}} + \left(\widehat{m}_d^{(k)} \right)_{\boldsymbol{j}} - \eta^{(k)} \left(\partial_{M_d} \mathcal{Y}(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}) \right)_{\boldsymbol{j}}$$

From (3.14), we have

$$\left(m_d^{(k+\frac{1}{2})} \right)_{\boldsymbol{j}} = \left(\widehat{m}_d^{(k)} \right)_{\boldsymbol{j}} - \eta^{(k)} Y_d \left(\widehat{\rho}_{\boldsymbol{j}}^{(k)}, \widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)}, \boldsymbol{x}_{\boldsymbol{j}} \right).$$

Combining above gives us

$$\left(E_d^{(k+\frac{1}{2})} \right)_{\boldsymbol{j}} = \left(\widehat{E}_d^{(k)} \right)_{\boldsymbol{j}} - \eta^{(k)} \left[\left(\partial_{M_d} \mathcal{Y}(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}) \right)_{\boldsymbol{j}} - Y_d \left(\widehat{\rho}_{\boldsymbol{j}}^{(k)}, \widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right]$$

Therefore we have the norm estimation

(4.9)
$$\left\|\boldsymbol{E}^{(k+\frac{1}{2})}\right\|_F \le \left\|\widehat{\boldsymbol{E}}^{(k)}\right\|_F + \eta^{(k)}\left[\sum_{d=0}^{D}\sum_{\boldsymbol{j}\in\boldsymbol{J}_d}\left|\left(\partial_{M_d}\mathcal{Y}(\widehat{P}^{(k)},\widehat{\boldsymbol{M}}^{(k)})\right)_{\boldsymbol{j}} - Y_d\left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right|^2\right]^{\frac{1}{2}}$$

For any $\boldsymbol{j}\in\boldsymbol{J}_d$, the definition of $\left(\partial_{M_d}\mathcal{Y}(\widehat{P}^{(k)},\widehat{\boldsymbol{M}}^{(k)})\right)_{\boldsymbol{j}}$ in (3.21) yields,

$$\begin{aligned}
&\left|\left(\partial_{M_d}\mathcal{Y}\left(\widehat{P}^{(k)},\widehat{\boldsymbol{M}}^{(k)}\right)\right)_{\boldsymbol{j}} - Y_d\left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right| \\
\le&\frac{1}{2}\left|\left(\partial_{\overline{M}_d}\overline{\mathcal{Y}}\left(\overline{\widehat{P}}^{(k)},\overline{\widehat{\boldsymbol{M}}}^{(k)}\right)\right)_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}} - Y_d\left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right| \\
&+\frac{1}{2}\left|\left(\partial_{\overline{M}_d}\overline{\mathcal{Y}}\left(\overline{\widehat{P}}^{(k)},\overline{\widehat{\boldsymbol{M}}}^{(k)}\right)\right)_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}} - Y_d\left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right| \\
=&\frac{1}{2}\left|Y_d\left(\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}},\overline{\widehat{\boldsymbol{M}}}^{(k)}_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}},\boldsymbol{x}_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}}\right) - Y_d\left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right| \\
&+\frac{1}{2}\left|Y_d\left(\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}},\overline{\widehat{\boldsymbol{M}}}^{(k)}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}},\boldsymbol{x}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}\right) - Y_d\left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right| \\
\le&\frac{C_Y}{2}\left\|\left(\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}},\overline{\widehat{\boldsymbol{M}}}^{(k)}_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}},\boldsymbol{x}_{\boldsymbol{j}+\frac{\boldsymbol{e}_d}{2}}\right) - \left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right\|_1 \\
&+\frac{C_Y}{2}\left\|\left(\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}},\overline{\widehat{\boldsymbol{M}}}^{(k)}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}},\boldsymbol{x}_{\boldsymbol{j}-\frac{\boldsymbol{e}_d}{2}}\right) - \left(\widehat{\rho}_{\boldsymbol{j}}^{(k)},\widehat{\boldsymbol{m}}_{\boldsymbol{j}}^{(k)},\boldsymbol{x}_{\boldsymbol{j}}\right)\right\|_1
\end{aligned} \tag{4.10}$$

Note that $\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}}$, $\left(\overline{\widehat{M}}^{(k)}_d\right)_{\boldsymbol{j}}$ can be written as the sum of errors and continuous values:

$$\begin{aligned}
\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}} &= \left(\mathcal{A}_0\left(\widehat{E}_0^{(k)}+\widehat{\rho}_{\boldsymbol{J}_0}^{(k)}\right)+\overline{P}_{\mathcal{A}}\right)_{\boldsymbol{j}} \\
&= \left(\mathcal{A}_0\left(\widehat{E}_0^{(k)}\right)\right)_{\boldsymbol{j}} + \left(\mathcal{A}_0\left(\widehat{\rho}_{\boldsymbol{J}_0}^{(k)}\right)+\overline{P}_{\mathcal{A}}\right)_{\boldsymbol{j}} \\
&= \left(\mathcal{A}_0\left(\widehat{E}_0^{(k)}\right)\right)_{\boldsymbol{j}} + \widehat{\rho}_{\boldsymbol{j}}^{(k)} + \mathcal{O}(\Delta_0^2) \\
\left(\overline{\widehat{M}}^{(k)}_d\right)_{\boldsymbol{j}} &= \left(\mathcal{A}_d\left(\widehat{E}_d^{(k)}\right)\right)_{\boldsymbol{j}} + \left(\widehat{m}_d^{(k)}\right)_{\boldsymbol{j}} + \mathcal{O}(\Delta_d^2),
\end{aligned}$$

where the last equality in the above two equations are obtained from using Taylor expansion to $\widehat{\rho}^{(k)}$ and $\widehat{m}_d^{(k)}$. We further have:

$$
\begin{aligned}
&\left\| \left(\overline{\widehat{P}}^{(k)}_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}}, \overline{\widehat{\boldsymbol{M}}}^{(k)}_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}}, \boldsymbol{x}_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}} \right) - \left(\widehat{\rho}^{(k)}_{\boldsymbol{j}}, \widehat{\boldsymbol{m}}^{(k)}_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right\|_1 \\
\leq & \left\| \left(\left(\mathcal{A}_0 \left(\widehat{E}_0^{(k)} \right) \right)_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}}, \left(\mathcal{A} \left(\widehat{\boldsymbol{E}}^{(k)}_{\boldsymbol{m}} \right) \right)_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}}, \boldsymbol{0} \right) \right\|_1 \\
& + \left\| \left(\widehat{\rho}^{(k)}_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}}, \widehat{\boldsymbol{m}}^{(k)}_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}}, \boldsymbol{x}_{\boldsymbol{j}\pm\frac{\boldsymbol{e}_d}{2}} \right) - \left(\widehat{\rho}^{(k)}_{\boldsymbol{j}}, \widehat{\boldsymbol{m}}^{(k)}_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right\|_1 + \mathcal{O}\left(\Delta_d^2 \right) \\
\leq & \frac{1}{2} \sum_{d'=0}^{D} \left| \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}} + \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}+\boldsymbol{e}_d} \right| + \frac{1}{2} \sum_{d'=0}^{D} \left| \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}} + \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}-\boldsymbol{e}_d} \right| \\
& + C\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) \Delta_d + \mathcal{O}\left(\Delta_d^2 \right),
\end{aligned} \tag{4.11}
$$

where $C\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) = \max\left\{ \frac{\partial}{\partial x_d} \widehat{m}^{(k)}_{d'}(t, \boldsymbol{x}) : d' = 0, \cdots, D \right\}$. Combining (4.10) and (4.11) provides:

$$
\begin{aligned}
& \left| \left(\partial_{M_d} \mathcal{Y}\left(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)} \right) \right)_{\boldsymbol{j}} - Y_d\left(\widehat{\rho}^{(k)}_{\boldsymbol{j}}, \widehat{\boldsymbol{m}}^{(k)}_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right| \\
\leq & \frac{C_Y}{4} \sum_{d'=0}^{D} \left| \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}} + \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}+\boldsymbol{e}_d} \right| + \frac{C_Y}{4} \sum_{d'=0}^{D} \left| \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}} + \left(\widehat{E}^{(k)}_{d'} \right)_{\boldsymbol{j}-\boldsymbol{e}_d} \right| \\
& + C\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) \Delta_d + \mathcal{O}\left(\Delta_d^2 \right).
\end{aligned}
$$

and applying the triangle inequality yields:

$$
\begin{aligned}
& \left[\sum_{d=0}^{D} \sum_{\boldsymbol{j}\in\boldsymbol{J}_d} \left| \left(\partial_{M_d} \mathcal{Y}(\widehat{P}^{(k)}, \widehat{\boldsymbol{M}}^{(k)}) \right)_{\boldsymbol{j}} - Y_d\left(\widehat{\rho}^{(k)}_{\boldsymbol{j}}, \widehat{\boldsymbol{m}}^{(k)}_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right|^2 \right]^{\frac{1}{2}} \\
\leq & \left[\sum_{d=0}^{D} \sum_{\boldsymbol{j}\in\boldsymbol{J}_d} \sum_{d'=0}^{D} \frac{C_Y^2}{8} \left(\left(\widehat{E}^{(k)}_{d'} \right)^2_{\boldsymbol{j}-\boldsymbol{e}_d} + 2\left(\widehat{E}^{(k)}_{d'} \right)^2_{\boldsymbol{j}} + \left(\widehat{E}^{(k)}_{d'} \right)^2_{\boldsymbol{j}+\boldsymbol{e}_d} \right) \right]^{\frac{1}{2}} \\
& + \left[\sum_{d=0}^{D} \sum_{\boldsymbol{j}\in\boldsymbol{J}_d} C^2\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) \Delta_d^2 \right]^{\frac{1}{2}} + \left(\prod_{d=0}^{D} n_d \right)^{\frac{1}{2}} \mathcal{O}\left(\sum_{d=0}^{D} \Delta_d^2 \right) \\
\leq & C(D, C_Y) \left\| \widehat{\boldsymbol{E}}^{(k)} \right\|_F \\
& + \left(\prod_{d=0}^{D} n_d \right)^{\frac{1}{2}} C\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) \left(\sum_{d=0}^{D} \Delta_d \right) + \left(\prod_{d=0}^{D} n_d \right)^{\frac{1}{2}} \mathcal{O}\left(\sum_{d=0}^{D} \Delta_d^2 \right),
\end{aligned}
$$

Together with estimation (4.9), we have

$$
\begin{aligned}
\left\| \boldsymbol{E}^{(k+\frac{1}{2})} \right\|_2 \leq & \left(1 + C(D, C_Y, \eta^{(k)}) \right) \left\| \widehat{\boldsymbol{E}}^{(k)} \right\|_2 \\
& + C\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)} \right) \left(\sum_{d=0}^{D} \Delta_d \right) + \mathcal{O}\left(\sum_{d=0}^{D} \Delta_d^2 \right).
\end{aligned} \tag{4.12}
$$

Therefore we prove the lemma. □

Next we examine the error introduced in projection step.

Lemma 4.6. *Suppose that $\rho_0, \rho_1, \rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}$ are C^2, and $\phi^{(k+1)}$ is C^3, then*

$$\left\|\boldsymbol{E}^{(k+1)}\right\|_2 \leq 2\left\|\boldsymbol{E}^{(k+\frac{1}{2})}\right\|_2 + C\left(\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \phi^{(k+1)}\right)\left(\sum_{d=0}^{D}\Delta_d\right).$$

Proof. By definition of error terms and (3.26), we have

$$-\operatorname{Lap}\left(E_\phi^{(k+1)} + \phi_{\overline{\boldsymbol{J}}}^{(k+1)}\right) = \operatorname{Div}\left(\boldsymbol{E}^{(k+\frac{1}{2})} + \left(\rho_{\boldsymbol{J}_0}^{(k+\frac{1}{2})}, \boldsymbol{m}_{\boldsymbol{J}}^{(k+\frac{1}{2})}\right)\right) + \overline{P}_{\mathcal{D}}. \tag{4.13}$$

Since $\left(\phi^{(k+1)}, \rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}\right)$ satisfies (3.15), and $\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}$ are C^2, by Taylor expansions, we have

$$-\operatorname{Lap}\left(\phi_{\overline{\boldsymbol{J}}}^{(k+1)}\right) = \operatorname{Div}\left(\rho_{\boldsymbol{J}_0}^{(k+\frac{1}{2})}, \boldsymbol{m}_{\boldsymbol{J}}^{(k+\frac{1}{2})}\right) + \overline{P}_D + \boldsymbol{C}_1\left(\sum_{d=0}^{D}\Delta_d\right) \tag{4.14}$$

Here $\boldsymbol{C}_1 = \boldsymbol{C}_1\left(\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \phi^{(k+1)}\right) \in \overline{\mathbb{V}}$ indicates its entries are constants depending on $\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \phi^{(k+1)}$. Combining (4.13) and (4.14) gives us

$$-\operatorname{Lap}\left(E_\phi^{(k+1)}\right) = \operatorname{Div}\left(\boldsymbol{E}^{(k+\frac{1}{2})}\right) + \boldsymbol{C}_1\left(\sum_{d=0}^{D}\Delta_d\right). \tag{4.15}$$

Similarly, the second step on discrete mesh (3.27) gives

$$\begin{aligned}\boldsymbol{E}^{(k+1)} + \left(\rho_{\boldsymbol{J}_0}^{(k+1)}, \boldsymbol{m}_{\boldsymbol{J}}^{(k+1)}\right) =& \boldsymbol{E}^{(k+\frac{1}{2})} + \operatorname{Grad}\left(E_\phi^{(k+1)}\right) \\ &+ \left(\rho_{\boldsymbol{J}_0}^{(k+\frac{1}{2})}, \boldsymbol{m}_{\boldsymbol{J}}^{(k+\frac{1}{2})}\right) + \operatorname{Grad}\left(\phi_{\overline{\boldsymbol{J}}}^{(k+1)}\right),\end{aligned} \tag{4.16}$$

and on continuous setting (3.16) gives

$$\left(\rho_{\boldsymbol{J}_0}^{(k+1)}, \boldsymbol{m}_{\boldsymbol{J}}^{(k+1)}\right) = \left(\rho_{\boldsymbol{J}_0}^{(k+\frac{1}{2})}, \boldsymbol{m}_{\boldsymbol{J}}^{(k+\frac{1}{2})}\right) + \operatorname{Grad}\left(\phi_{\overline{\boldsymbol{J}}}^{(k+1)}\right) + \boldsymbol{C}_2\left(\sum_{d=0}^{D}\Delta_d\right), \tag{4.17}$$

Thus we have:

$$\boldsymbol{E}^{(k+1)} = \boldsymbol{E}^{(k+\frac{1}{2})} + \operatorname{Grad}\left(E_\phi^{(k+1)}\right) + \boldsymbol{C}_2\left(\sum_{d=0}^{D}\Delta_d\right), \tag{4.18}$$

where $\boldsymbol{C}_2 = \boldsymbol{C}_2\left(\phi^{(k+1)}\right) \in \overline{\mathbb{V}}$.

Combining (4.15) and (4.18), we obtain

$$\begin{aligned}\boldsymbol{E}^{(k+1)} =& \left(\mathrm{Id} - \operatorname{Grad}\circ\operatorname{Lap}^{-1}\circ\operatorname{Div}\right)\boldsymbol{E}^{(k+\frac{1}{2})} \\ &- \operatorname{Grad}\circ\operatorname{Lap}^{-1}\boldsymbol{C}_1\left(\sum_{d=0}^{D}\Delta_d\right) + \boldsymbol{C}_2\left(\sum_{d=0}^{D}\Delta_d\right)\end{aligned} \tag{4.19}$$

Claim: $\|\operatorname{Grad}\circ\operatorname{Lap}^{-1}\circ\operatorname{Div}\|_2 \leq 1$, $\|\operatorname{Grad}\circ\operatorname{Lap}^{-1}\|_2 \leq \dfrac{1}{4}$.

Therefore

$$\begin{aligned}\left\|\boldsymbol{E}^{(k+1)}\right\|_2 &\leq 2\left\|\boldsymbol{E}^{(k+\frac{1}{2})}\right\|_2+\frac{1}{4}\left\|\boldsymbol{C}_1\left(\sum_{d=0}^{D}\Delta_d\right)\right\|_2+\left\|\boldsymbol{C}_2\left(\sum_{d=0}^{D}\Delta_d\right)\right\|_2\\ &\leq 2\left\|\boldsymbol{E}^{(k+\frac{1}{2})}\right\|_2+C\left(\sum_{d=0}^{D}\Delta_d\right),\end{aligned}$$

and C depends on $\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \phi^{(k+1)}$.

Proof of claim: It is easy to check that with

$$\begin{aligned}\lambda^{\boldsymbol{i}} &= -4\sum_{d=0}^{D} n_d^2 \sin^2\left(\frac{(i_d-1)\pi}{2n_d}\right),\\ \Psi_{\boldsymbol{j}}^{\boldsymbol{i}} &= \prod_{d=0}^{D}\sqrt{\frac{1+\delta_{1i_d}}{n_d}}\cos\left(\left(j_d-\frac{1}{2}\right)\frac{(i_d-1)\pi}{n_d}\right),\end{aligned}$$

$\left\{\Psi^{\boldsymbol{i}}\right\}_{\boldsymbol{i}\in\overline{\boldsymbol{J}}}$ forms an orthonormal basis of $(\overline{\mathbb{V}}, \|\cdot\|_F)$, and for any $\overline{\Phi}\in\overline{\mathbb{V}}$,

$$\mathrm{Lap}(\overline{\Phi}) := \sum_{\boldsymbol{i}\in\overline{\boldsymbol{J}}}\lambda_{\boldsymbol{i}}\left\langle\overline{\Phi}, \Psi^{\boldsymbol{i}}\right\rangle\Psi^{\boldsymbol{i}}. \tag{4.20}$$

For $d=0,1,\cdots,D$, and $\boldsymbol{i}\in\overline{\boldsymbol{J}}, i_d\neq 1$, let $\sigma^{d,\boldsymbol{i}}\in\mathbb{R}$ and $\boldsymbol{\Psi}^{d,\boldsymbol{i}}\in\mathbb{V}_0\times\mathbb{V}_1\times\cdots\times\mathbb{V}_d$ be:

$$\begin{aligned}\sigma^{d,\boldsymbol{i}} &= -2n_d\sin\left(\frac{(i_d-1)\pi}{2n_d}\right),\\ \boldsymbol{\Psi}^{d,\boldsymbol{i}} &= \left\{\Psi_{d'}^{d,\boldsymbol{i}}\right\}_{d'=0,1,\cdots,D},\\ &\text{where } \Psi_d^{d,\boldsymbol{i}} = \frac{1}{\sigma^{d,\boldsymbol{i}}}\mathcal{D}_d^*\left(\Psi^{\boldsymbol{i}}\right), \quad \Psi_{d'}^{d,\boldsymbol{i}} = \boldsymbol{0}, d'\neq d,\end{aligned}$$

then $\left\{\boldsymbol{\Psi}^{d,\boldsymbol{i}}\right\}$ forms an orthonormal basis of $(\mathbb{V}_0\times\mathbb{V}_1\times\cdots\times\mathbb{V}_D, \|\cdot\|_F)$.

Since $\|\cdot\|_2 = \left(\prod_{d=0}^{D}\Delta_d\right)^{\frac{1}{2}}\|\cdot\|_F$, we next compute the $\left\|\mathrm{Grad}\circ\mathrm{Lap}^{-1}\circ\mathrm{Div}\right\|_2$, $\left\|\mathrm{Grad}\circ\mathrm{Lap}^{-1}\right\|_2$ with basis of $(\overline{\mathbb{V}}, \|\cdot\|_F)$ and $(\mathbb{V}_0\times\mathbb{V}_1\times\cdots\times\mathbb{V}_D, \|\cdot\|_F)$. For any basis $\Psi^{\boldsymbol{i}}\in\overline{\mathbb{V}}$,

$$\begin{aligned}&\mathrm{Grad}\circ\mathrm{Lap}^{-1}\left(\Psi^{\boldsymbol{1}}\right) = \boldsymbol{0},\\ &\mathrm{Grad}\circ\mathrm{Lap}^{-1}\left(\Psi^{\boldsymbol{i}}\right) = \sum_{\substack{d=0,\\ i_d\neq 1}}^{D}\frac{\sigma^{d,\boldsymbol{i}}}{\lambda^{\boldsymbol{i}}}\Psi^{d,\boldsymbol{i}}, \quad \boldsymbol{i}\neq\boldsymbol{1}\end{aligned}$$

thus when $n_d>1$ for $d=0,\cdots,D$, we have

$$\left\|\mathrm{Grad}\circ\mathrm{Lap}^{-1}\right\|_2 \leq \max_{\boldsymbol{i}\in\overline{\boldsymbol{J}}\setminus\{\boldsymbol{1}\}}\left(\frac{1}{\left(\lambda^{\boldsymbol{i}}\right)^2}\sum_{\substack{d=0,\\ i_d\neq 1}}^{D}\left(\sigma^{d,\boldsymbol{i}}\right)^2\right)^{\frac{1}{2}} = \max_{\boldsymbol{i}\in\overline{\boldsymbol{J}}\setminus\{\boldsymbol{1}\}}\frac{1}{|\lambda^{\boldsymbol{i}}|}\leq\frac{1}{4}.$$

And for any basis $\Psi^{d,\boldsymbol{i}} \in \mathbb{V}_0 \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_D$,

$$\begin{aligned}
&\mathrm{Grad} \circ \mathrm{Lap}^{-1} \circ \mathrm{Div}\left(\Psi^{d,\boldsymbol{i}}\right) \\
=&\mathrm{Grad} \circ \mathrm{Lap}^{-1}\left(\frac{1}{\sigma^{d,\boldsymbol{i}}}\mathcal{D}_d \circ \mathcal{D}_d^*\left(\Psi^{\boldsymbol{i}}\right)\right) = \mathrm{Grad} \circ \mathrm{Lap}^{-1}\left(-\sigma^{d,\boldsymbol{i}}\Psi^{\boldsymbol{i}}\right) \\
=&\mathrm{Grad}\left(-\frac{\sigma^{d,\boldsymbol{i}}}{\lambda^{\boldsymbol{i}}}\Psi^{\boldsymbol{i}}\right) = -\sum_{\substack{d'=0,\\ i_{d'}\neq 1}}^{D} \frac{\sigma^{d,\boldsymbol{i}}\sigma^{d',\boldsymbol{i}}}{\lambda^{\boldsymbol{i}}}\Psi^{d',\boldsymbol{i}}
\end{aligned}$$

therefore

$$\begin{aligned}
\left\|\mathrm{Grad} \circ \mathrm{Lap}^{-1} \circ \mathrm{Div}\right\|_2 \leq& \max_{d=0,1,\cdots,D} \max_{\boldsymbol{i}\in\overline{\boldsymbol{J}}, i_d\neq 1} \left(\left(\frac{\sigma^{d,\boldsymbol{i}}}{\lambda^{\boldsymbol{i}}}\right)^2 \sum_{\substack{d'=0,\\ i_{d'}\neq 1}}^{D}\left(\sigma^{d',\boldsymbol{i}}\right)^2\right)^{\frac{1}{2}} \\
=& \max_{d=0,1,\cdots,D} \max_{\boldsymbol{i}\in\overline{\boldsymbol{J}}, i_d\neq 1} \left(\frac{\left(\sigma^{d,\boldsymbol{i}}\right)^2}{|\lambda^{\boldsymbol{i}}|}\right)^{\frac{1}{2}} \leq 1.
\end{aligned}$$

The claim and thus the lemma are proved. □

The last step is to estimate the error introduced in linear interpolation step (3.17),(3.28).

Lemma 4.7.

$$\left\|\widehat{\boldsymbol{E}}^{(k+1)}\right\|_2 \leq \left|1+\omega^{(k)}\right|\left\|\boldsymbol{E}^{(k+1)}\right\|_2 + \left|\omega^{(k)}\right|\left\|\boldsymbol{E}^{(k)}\right\|_2.$$

Proof. By definition of error terms and (3.28)

$$\begin{aligned}
&\widehat{E}_d^{(k+1)} + \left(\widehat{m}_d^{(k+1)}\right)_{\boldsymbol{J}_d} \\
=&\left(1+\omega^{(k)}\right)\left(E_d^{(k+1)} + \left(m_d^{(k+1)}\right)_{\boldsymbol{J}_d}\right) - \omega^{(k)}\left(E_d^{(k)} + \left(m_d^{(k)}\right)_{\boldsymbol{J}_d}\right),
\end{aligned}$$

and by (3.17)

$$\left(\widehat{m}_d^{(k+1)}\right)_{\boldsymbol{J}_d} = \left(1+\omega^{(k)}\right)\left(m_d^{(k+1)}\right)_{\boldsymbol{J}_d} - \omega^{(k)}\left(m_d^{(k)}\right)_{\boldsymbol{J}_d}.$$

Therefore we have

$$\widehat{\boldsymbol{E}}^{(k+1)} = \left(1+\omega^{(k)}\right)\boldsymbol{E}^{(k+1)} - \omega^{(k)}\boldsymbol{E}^{(k)}.$$

By triangular inequality, the lemma is proved. □

With Lemma 4.5-Lemma 4.7, we can show Theorem 4.2 by induction.
Proof of Theorem 4.2:

Proof. We first restate results from Lemma 4.5-Lemma 4.7:

$$\left\|\boldsymbol{E}^{(k+1)}\right\|_2 \leq 2\left\|\boldsymbol{E}^{(k+\frac{1}{2})}\right\|_2 + C\left(\rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \phi^{(k+1)}\right)\left(\sum_{d=0}^{D}\Delta_d\right),$$

$$\left\|\boldsymbol{E}^{(k+\frac{1}{2})}\right\|_2 \leq C(D, C_Y, \eta^{(k)})\left\|\widehat{\boldsymbol{E}}^{(k)}\right\|_2 + C\left(\widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)}\right)\left(\sum_{d=0}^{D}\Delta_d\right) + \mathcal{O}\left(\sum_{d=0}^{D}\Delta_d^2\right),$$

$$\left\|\widehat{\boldsymbol{E}}^{(k)}\right\|_2 \leq \left|1+\omega^{(k-1)}\right|\left\|\boldsymbol{E}^{(k)}\right\|_2 + \left|\omega^{(k-1)}\right|\left\|\boldsymbol{E}^{(k-1)}\right\|_2.$$

From these, we obtain

(4.21)
$$\left\|\boldsymbol{E}^{(1)}\right\|_2 \leq C\left\|\widehat{\boldsymbol{E}}^{(0)}\right\|_2 + C\sum_{d=0}^{D}\Delta_d + \mathcal{O}\left(\sum_{d=0}^{D}\Delta_d^2\right),$$

(4.22)
$$\left\|\boldsymbol{E}^{(k+1)}\right\|_2 \leq C\left\|\boldsymbol{E}^{(k)}\right\|_2 + C\left\|\boldsymbol{E}^{(k-1)}\right\|_2 + C\sum_{d=0}^{D}\Delta_d + \mathcal{O}\left(\sum_{d=0}^{D}\Delta_d^2\right), \quad k \geq 1$$

where C depends on $D, C_Y, \eta^{(k)}, \rho^{(k+\frac{1}{2})}, \boldsymbol{m}^{(k+\frac{1}{2})}, \widehat{\rho}^{(k)}, \widehat{\boldsymbol{m}}^{(k)}, \phi^{(k+1)}$.

The initialization gives us

$$\left\|\boldsymbol{E}^{(0)}\right\|_2 = 0, \left\|\widehat{\boldsymbol{E}}^{(0)}\right\|_2 = 0,$$

Then based on (4.22), it is straightforward to show (4.2) by applying induction on k directly. □

5. Generalization and acceleration

In this section, we generalize the proposed algorithm to solve potential MFG problems. Moreover, we also discuss how to use multilevel and multigrid strategies to speed up our algorithm.

5.1. Generalization to potential MFG.

To apply FISTA to the MFG problem (1.2), we follow a first-discretize-then-optimize approach. One crucial difference between MFG and MFP is whether $\rho(1,\cdot)$ is provided explicitly. For MFG, we consider a discretization in Figure 4 and modify our previous notations related to ρ.

The index set and discrete variable are now

$$J_0 := \left\{\frac{3}{2}, \frac{5}{2}, \cdots, n_0 + \frac{1}{2}\right\}, \quad P := \{P_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\boldsymbol{J}_0} \in \mathbb{V}_0 := \mathbb{R}^{n_0\times n_1\times\cdots\times n_D},$$

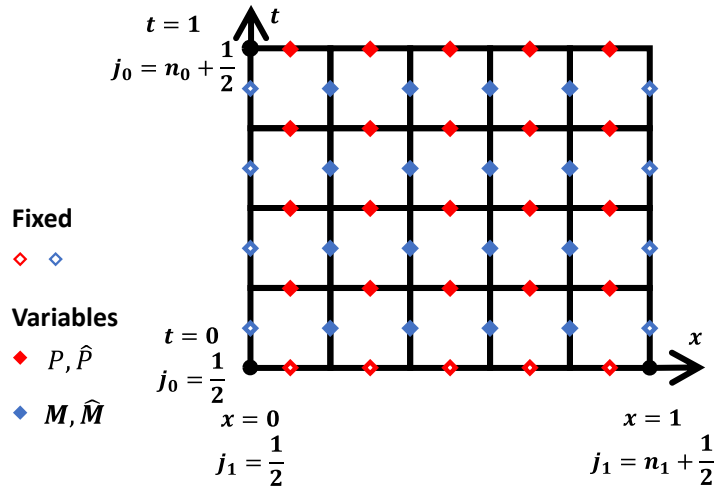


FIGURE 4. Illustration of discretization (MFG).

and the discrete operators are

$$\mathcal{A}_0 : \mathbb{V}_0 \to \overline{\mathbb{V}}, P \mapsto \overline{P},$$

$$\overline{P}_{\boldsymbol{j}} := \begin{cases} \frac{1}{2} P_{\boldsymbol{j}+\frac{\boldsymbol{e}_0}{2}}, & j_0 = 1, \\ \frac{1}{2}\left(P_{\boldsymbol{j}+\frac{\boldsymbol{e}_0}{2}} + P_{\boldsymbol{j}-\frac{\boldsymbol{e}_0}{2}}\right), & j_0 = 2, 3, \cdots, n_0, \end{cases}$$

$$\mathcal{D}_0 : \mathbb{V}_0 \to \overline{\mathbb{V}}, P \mapsto \mathcal{D}_0(P),$$

$$(\mathcal{D}_0(P))_{\boldsymbol{j}} := \begin{cases} \frac{1}{\Delta_0} P_{\boldsymbol{j}+\frac{\boldsymbol{e}_0}{2}}, & j_0 = 1, \\ \frac{1}{\Delta_0}\left(P_{\boldsymbol{j}+\frac{\boldsymbol{e}_0}{2}} - P_{\boldsymbol{j}-\frac{\boldsymbol{e}_0}{2}}\right), & j_0 = 2, 3, \cdots, n_0. \end{cases}$$

Since the boundary condition is only at $t = 0$, we modify $\overline{P}_{\mathcal{A}}, \overline{P}_{\mathcal{D}} \in \overline{\mathbb{V}}$ to

$$(\overline{P}_{\mathcal{A}})_{\boldsymbol{j}} := \begin{cases} \frac{1}{2}\rho_0\left(\boldsymbol{x}_{\boldsymbol{j}}\right), & j_0 = 1, \\ 0, & j_0 = 2, 3, \cdots, n_0, \end{cases}$$

$$(\overline{P}_{\mathcal{D}})_{\boldsymbol{j}} := \begin{cases} -\frac{1}{\Delta_0}\rho_0\left(\boldsymbol{x}_{\boldsymbol{j}}\right), & j_0 = 1, \\ 0, & j_0 = 2, 3, \cdots, n_0. \end{cases}$$

Take model (2.11) in Section 2 as an example, the discrete problem can be formulated as

(5.1)

$$\begin{aligned} \min_{(P,\boldsymbol{M})\in\mathcal{C}_{MFG}(\overline{P}_{\mathcal{D}})} \mathcal{Y}_{MFG}(P,\boldsymbol{M}) :=& \Delta_0 \sum_{d=0}^{D} \sum_{j_d=1}^{n_d} J_{MFG}\left((\mathcal{A}_0(P) + \overline{P}_{\mathcal{A}})_{\boldsymbol{j}}, \mathcal{A}(\boldsymbol{M})_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}}\right) \\ &+ \lambda_G \sum_{\substack{\boldsymbol{j}\in\boldsymbol{J}_0, \\ j_0 = n_0 + \frac{1}{2}}} P_{\boldsymbol{j}} G(\boldsymbol{x}_{\boldsymbol{j}}) \end{aligned}$$

where

$$J_{MFG}(\beta_0, \boldsymbol{\beta}, \boldsymbol{x}) := L(\beta_0, \boldsymbol{\beta}) + \lambda_E \beta_0 \log(\beta_0) + \lambda_Q \beta_0 Q(\boldsymbol{x}), \tag{5.2}$$

$$\mathcal{C}_{MFG}(\overline{P}_{\mathcal{D}}) := \left\{ (P, \boldsymbol{M}) : \mathcal{D}_0(P) + \overline{P}_{\mathcal{D}} + \mathrm{Div}(\boldsymbol{M}) = \mathbf{0} \right\}. \tag{5.3}$$

Since this is an optimization problem with linear constraints, we apply FISTA to it as detailed in Algorithm 3. In the algorithm, $\mathcal{A}_0^\top, \mathcal{A}^\top, \mathcal{D}_0^\top, \mathrm{Div}^\top$ are conjugate operators of $\mathcal{A}_0, \mathcal{A}, \mathcal{D}_0, \mathrm{Div}$ in norm $\|\cdot\|_F$. Similar to what we discussed before, one can have:

$$\begin{cases} \partial_{\overline{P}} \mathcal{Y}_{MFG}(P, \boldsymbol{M}) := \left\{ \Delta_0 (J_{MFG})_{\beta_0} \left((\mathcal{A}_0(P) + \overline{P}_{\mathcal{A}})_{\boldsymbol{j}}, \mathcal{A}(\boldsymbol{M})_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right\}_{\boldsymbol{j} \in \overline{\boldsymbol{J}}}, \\ \partial_{\overline{\boldsymbol{M}}} \mathcal{Y}_{MFG}(P, \boldsymbol{M}) := \left\{ \Delta_0 (J_{MFG})_{\boldsymbol{\beta}} \left((\mathcal{A}_0(P) + \overline{P}_{\mathcal{A}})_{\boldsymbol{j}}, \mathcal{A}(\boldsymbol{M})_{\boldsymbol{j}}, \boldsymbol{x}_{\boldsymbol{j}} \right) \right\}_{\boldsymbol{j} \in \overline{\boldsymbol{J}}}, \end{cases}$$

and

$$\begin{cases} (\partial_P \mathcal{Y}_{MFG}(P, \boldsymbol{M}))_{\boldsymbol{j}} := \left(\mathcal{A}_0^\top \left(\partial_{\overline{P}} \mathcal{Y}_{MFG}(P, \boldsymbol{M}) \right) \right)_{\boldsymbol{j}}, & j_0 \neq n_0 + \frac{1}{2} \\ (\partial_P \mathcal{Y}_{MFG}(P, \boldsymbol{M}))_{\boldsymbol{j}} := \left(\mathcal{A}_0^\top \left(\partial_{\overline{P}} \mathcal{Y}_{MFG}(P, \boldsymbol{M}) \right) \right)_{\boldsymbol{j}} + \lambda_G G(\boldsymbol{x}_{\boldsymbol{j}}), & \\ & j_0 = n_0 + \frac{1}{2} \\ \partial_{\boldsymbol{M}} \mathcal{Y}_{MFG}(P, \boldsymbol{M}) := \mathcal{A}^\top \left(\partial_{\overline{\boldsymbol{M}}} \mathcal{Y}_{MFG}(P, \boldsymbol{M}) \right). & \end{cases}$$

Remark 5.1. In Algorithm 3, we also need to solve a discrete Poisson equation (5.4) and the approach is similar as presented in Remark 3.3.

5.2. **Multilevel and multigrid FISTA.** Inspired by [37, 39], we borrow ideas from multigrid and multilevel methods in numerical PDEs to our variational problem. We first restrict our initialization to coarser meshes and solve the optimization problem to a certain accuracy. Then we successive refine the mesh and solve the problem until we obtain the solution on a desired fine mesh. According to [37, 39], the solution on the coarse mesh approximates that on the finer mesh and gives a better initialization when solving the problem on the finer mesh. Therefore, these methods can reduce computational cost on the finest level and thus accelerate the proposed algorithm. The implementation details are presented in this section.

For notation simplicity, we assume $h = \Delta_0 = \Delta_1 = \cdots = \Delta_D$ in this section. Let ${}^h\Omega$ be a grid with $h = \Delta_d$, ${}^h\boldsymbol{J}_d$ be the certain $\boldsymbol{J}_d$ on the grid. Then index ${}^h\boldsymbol{j} \in {}^h\boldsymbol{J}_d$ stands for the point $h\boldsymbol{j}$. If there is no ambiguity, we can omit the prescript of $\boldsymbol{j}$. For example, we define ${}^h u_{\boldsymbol{j}} = u\left(h(\boldsymbol{j} - \frac{\mathbf{1}}{2})\right)$ for any function u and approximate the value by ${}^h U_{\boldsymbol{j}}$.

Consider L levels of grids ${}^{h_1}\Omega, \ldots, {}^{h_L}\Omega$ where the finest level is ${}^{h_1}\Omega$, and $h_l := 2^{l-1} h_1$. We first define how to prolongate values on a coarser grid into a finer grid. Assume that ${}^h\boldsymbol{j} \in {}^h\boldsymbol{J}_d$ stands for point $h\left(\boldsymbol{j} - \frac{\mathbf{1}}{2}\right)$ on the finer grid ${}^h\Omega$, we define its neighbourhood on the coarser grid ${}^{2h}\Omega$ as

$$\begin{aligned} {}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}} := \Bigg\{ {}^{2h}\boldsymbol{i} \in {}^{2h}\boldsymbol{J}_d : & \left\| 2h \left({}^{2h}\boldsymbol{i} - \frac{\mathbf{1}}{2} \right) - h \left({}^{h}\boldsymbol{j} - \frac{\mathbf{1}}{2} \right) \right\|_2 \\ & = \min_{{}^{2h}\boldsymbol{i}' \in {}^{2h}\boldsymbol{J}_d} \left\| 2h \left({}^{2h}\boldsymbol{i}' - \frac{\mathbf{1}}{2} \right) - h \left({}^{h}\boldsymbol{j} - \frac{\mathbf{1}}{2} \right) \right\|_2 \Bigg\}. \end{aligned} \tag{5.5}$$

Algorithm 3 FISTA for MFG

Parameters ρ_0, ρ_1

Initialization $\tau^{(1)} = 1$, $P^{(0)} = \widehat{P}^{(0)} = \mathbf{1}$, and $M_d^{(0)} = \widehat{M_d}^{(0)} = \mathbf{1}$.

for $k = 0, 1, 2, \ldots$ **do**

gradient descent:

$$\begin{cases} P^{(k+\frac{1}{2})} = P^{(k)} - \eta^{(k)} \partial_P \mathcal{Y}_{MFG}(P^{(k)}, \boldsymbol{M}^{(k)}), \\ \boldsymbol{M}^{(k+\frac{1}{2})} = \boldsymbol{M}^{(k)} - \eta^{(k)} \partial_{\boldsymbol{M}} \mathcal{Y}_{MFG}(P^{(k)}, \boldsymbol{M}^{(k)}) \end{cases}$$

projection: solve $\overline{\Phi}^{(k+1)}$ for

$$\left(\mathcal{D}_0 \mathcal{D}_0^\top + \mathrm{Div}\,\mathrm{Div}^\top\right) \overline{\Phi}^{(k+1)} = \mathcal{D}_0 \left(P^{(k+\frac{1}{2})}\right) + \overline{P}_{\mathcal{D}} + \mathrm{Div}\left(\boldsymbol{M}^{(k+\frac{1}{2})}\right), \tag{5.4}$$

and project $\left(P^{(k+\frac{1}{2})}, \boldsymbol{M}^{(k+\frac{1}{2})}\right)$ to $\mathcal{C}_{MFG}(\overline{P}_{\mathcal{D}})$ by

$$\begin{cases} P^{(k+1)} = P^{(k+\frac{1}{2})} - \mathcal{D}_0^\top \left(\overline{\Phi}^{(k+1)}\right), \\ \boldsymbol{M}^{(k+1)} = \boldsymbol{M}^{(k+\frac{1}{2})} - \mathrm{Div}^\top \left(\overline{\Phi}^{(k+1)}\right). \end{cases}$$

update

$$\tau^{(k+1)} = \frac{1 + \sqrt{1 + 4\left(\tau^{(k)}\right)^2}}{2},$$

$$\omega^{(k)} = \frac{\tau^{(k)} - 1}{\tau^{(k+1)}},$$

$$\left(\widehat{P}^{(k+1)}, \widehat{\boldsymbol{M}}^{(k+1)}\right) = \left(1 + \omega^{(k)}\right) \left(P^{(k+1)}, \boldsymbol{M}^{(k+1)}\right) - \omega^{(k)} \left(P^{(k)}, \boldsymbol{M}^{(k)}\right).$$

end for

Then with boundary values

$$\begin{aligned} {}^{2h}P_{\boldsymbol{i}} &= {}^{2h}(\rho_0)_{\boldsymbol{i}}, \quad i_0 = \frac{1}{2}, \\ {}^{2h}P_{\boldsymbol{i}} &= {}^{2h}(\rho_1)_{\boldsymbol{i}}, \quad i_0 = \frac{1}{2h} + \frac{1}{2}, \\ {}^{2h}(M_d)_{\boldsymbol{i}} &= 0, \quad i_d = \frac{1}{2}, \frac{1}{2h} + \frac{1}{2}, \end{aligned}$$

we define the prolongation $({}^hP, {}^h\boldsymbol{M}) = \mathrm{Pro}({}^{2h}P, {}^{2h}\boldsymbol{M})$ by averaging values in neighbourhoods:

$$\begin{cases} {}^hP_{\boldsymbol{j}} := \dfrac{1}{\left|{}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}\right|} \displaystyle\sum_{{}^{2h}\boldsymbol{i} \in {}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}} {}^{2h}P_{\boldsymbol{i}}, & \forall\, {}^h\boldsymbol{j} \in {}^h\boldsymbol{J}_0, \\ {}^h(M_d)_{\boldsymbol{j}} := \dfrac{1}{\left|{}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}\right|} \displaystyle\sum_{{}^{2h}\boldsymbol{i} \in {}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}} {}^{2h}(M_d)_{\boldsymbol{i}} & \forall\, {}^h\boldsymbol{j} \in {}^h\boldsymbol{J}_d. \end{cases} \tag{5.6}$$

An example of prolongation in 1D is shown in the left panel of Figure 5.

From a finer grid to a coarser grid, the neighbourhood is defined inversely. Suppose ${}^{2h}\boldsymbol{i} \in {}^{2h}\boldsymbol{J}_d$, its neighbourhood is the set of all ${}^h\boldsymbol{j} \in {}^h\boldsymbol{J}_d$ whose neighbourhood

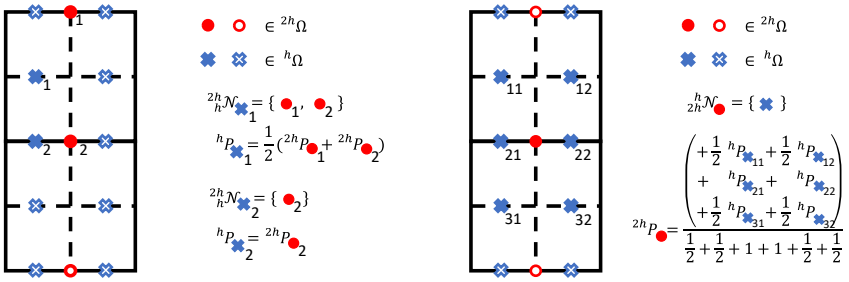


FIGURE 5. Illustration of Prolongation (left) and Restriction (right) for 1D case.

includes ${}^{2h}\boldsymbol{i}$:

$$ {}^{h}_{2h}\mathcal{N}_{\boldsymbol{i}} := \left\{ {}^{h}\boldsymbol{j} \in {}^{h}\boldsymbol{J}_d : {}^{2h}\boldsymbol{i} \in {}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}} \right\}. \tag{5.7} $$

and the restriction from finer level to coarser level $({}^{2h}P, {}^{2h}M) = \mathrm{Res}({}^{h}P, {}^{h}M)$ is defined by a weighted average over neighbourhoods:

$$ \begin{cases} {}^{2h}P_{\boldsymbol{i}} := \displaystyle\sum_{\boldsymbol{j}\in {}^{h}_{2h}\mathcal{N}_{\boldsymbol{i}}} \frac{1}{\left|{}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}\right|} {}^{h}P_{\boldsymbol{j}} \Big/ \sum_{\boldsymbol{j}\in {}^{h}_{2h}\mathcal{N}_{\boldsymbol{i}}} \frac{1}{\left|{}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}\right|}, & \forall\, {}^{2h}\boldsymbol{i} \in {}^{2h}\boldsymbol{J}_0, \\ {}^{2h}(M_d)_{\boldsymbol{i}} := \displaystyle\sum_{\boldsymbol{j}\in {}^{h}_{2h}\mathcal{N}_{\boldsymbol{i}}} \frac{1}{\left|{}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}\right|} {}^{h}(M_d)_{\boldsymbol{j}} \Big/ \sum_{\boldsymbol{j}\in {}^{h}_{2h}\mathcal{N}_{\boldsymbol{i}}} \frac{1}{\left|{}^{2h}_{h}\mathcal{N}_{\boldsymbol{j}}\right|}, & \forall\, {}^{2h}\boldsymbol{i} \in {}^{2h}\boldsymbol{J}_d. \end{cases} \tag{5.8} $$

An example of restriction is shown in the right panel of Figure 5.

We describe our multigrid FISTA in Algorithm 4, in which Algorithm $2_K(\cdot)$ means run Algorithm 2 for K iterations and Algorithm $2(\cdot)$ means run the algorithm till convergence. The first two inputs of Algorithm $2(\cdot)$ are initial and terminal densities ρ_0, ρ_1, and the last two inputs are initialization $P^{(0)} = \widehat{P}^{(0)}$, $\boldsymbol{M}^{(0)} = \widehat{\boldsymbol{M}}^{(0)}$.

Note that to keep the cost of Algorithm 4 low, we need to choose a K not very large. Motivated by [39], we can remove the pre-smoothing steps by setting $K = 0$ and this leads to our Algorithm 5: multilevel FISTA.

6. Numerical experiments

In this section, we conduct comprehensive experiments to show the efficiency and effectiveness of the proposed numerical algorithms. We first numerical verify the convergence of rate of the algorithm related to the mesh size. After that, our computation efficiency tests demonstrate that the proposed Algorithm 2 has comparable efficiency with the state-of-the-art methods. Interestingly, the proposed multilevel method performs around 10 times faster than existing methods. We further illustrate the flexibility of our algorithms on different MFP problems. In all the numerical experiments, we use the dynamic cost function L defined in (2.5). All of our numerical experiments are implemented in Matlab on a PC with an Intel(R) i7-8550U 1.80GHz CPU and 16 GB memory. The codes to reproduce the

Algorithm 4 Multigrid FISTA for MFP

Parameters $L, h_l = 2^{l-1}h, K, {}^{h_l}\rho_0, {}^{h_l}\rho_1 (l = 1, \ldots, L)$
Initialization ${}^{h_1}P^{(0)} = \mathbf{1}$, ${}^{h_1}\boldsymbol{M}^{(0)} = \mathbf{1}$

pre-smoothing

$$\left({}^{h_1}P, {}^{h_1}\boldsymbol{M}\right) = \text{Algorithm 2}_K\left(\rho_0, \rho_1, {}^{h_1}P^{(0)}, {}^{h_1}\boldsymbol{M}^{(0)}\right)$$

for $l = 2, 3, \ldots, L$ **do**

$$\left({}^{h_l}P, {}^{h_l}\boldsymbol{M}\right) = \text{Algorithm 2}_K\left(\rho_0, \rho_1, \text{Res}\left({}^{h_{l-1}}P, {}^{h_{l-1}}\boldsymbol{M}\right)\right)$$

end for

correction and post-smoothing

$$\left({}^{h_L}P, {}^{h_L}\boldsymbol{M}\right) = \text{Algorithm 2}\left(\rho_0, \rho_1, {}^{h_L}P, {}^{h_L}\boldsymbol{M}\right)$$

for $l = L-1, L-2, \ldots, 1$ **do**

$$\left({}^{h_l}P, {}^{h_l}\boldsymbol{M}\right) = \left({}^{h_l}P, {}^{h_l}\boldsymbol{M}\right) + \text{Algorithm 2}\left(\rho_0, \rho_1, \text{Pro}\left({}^{h_{l+1}}P, {}^{h_{l+1}}\boldsymbol{M}\right)\right) - \text{Pro}\left({}^{h_{l+1}}P, {}^{h_{l+1}}\boldsymbol{M}\right)$$

end for

Algorithm 5 Multilevel FISTA for MFP

Parameters $L, h_l = 2^{l-1}h, {}^{h_l}\rho_0, {}^{h_l}\rho_1 (l = 1, \ldots, L)$
Initialization ${}^{h_L}P^{(0)} = \mathbf{1}$, ${}^{h_L}\boldsymbol{M}^{(0)} = \mathbf{1}$
$\left({}^{h_L}P, {}^{h_L}\boldsymbol{M}\right) = \text{Algorithm 2}\left(\rho_0, \rho_1, {}^{h_L}P^{(0)}, {}^{h_L}\boldsymbol{M}^{(0)}\right)$
for $l = L-1, L-2, \ldots, 1$ **do**
$\left({}^{h_l}P, {}^{h_l}\boldsymbol{M}\right) = \text{Algorithm 2}\left(\rho_0, \rho_1, \text{Pro}\left({}^{h_{l+1}}P, {}^{h_{l+1}}\boldsymbol{M}\right)\right)$
end for

numerical results in this paper are available in `https://github.com/Jiajia-Yu/FISTA_MFP_euc`.

6.1. **Convergence rate.** To numerically verify the theoretical convergence analysis discussed in Section 4, we first apply the proposed numerical algorithm to a simple 1D OT example with exact solution as follows.

Let $\Omega = [0,1]$, $\rho_0(x) = x + \frac{1}{2}, \rho_1(x) = 1$. Then we can have the following theoretical solution of the OT between ρ_0 and ρ_1.

$$\rho^*(t,x) = \begin{cases} x + \dfrac{1}{2}, & t = 0, \\ \dfrac{\sqrt{2tx + \left(\frac{t}{2} - 1\right)^2} + t - 1}{t\sqrt{2tx + \left(\frac{t}{2} - 1\right)^2}}, & 0 < t \le 1. \end{cases} \tag{6.1}$$

TABLE 1. Convergence rate of Algorithm 2 applied to 1D OT problem ($k = 50000$).

Δ_0	Δ_1	$\left\|\boldsymbol{E}^{(k,*)}\right\|_2$	order	$\left\|\boldsymbol{E}^{(k,*)}\right\|_\infty$	order	W_2^2 error	order
1/16	1/64	3.19E-04		2.88E-03		4.88E-06	
1/32	1/128	1.08E-04	1.56	1.47E-03	0.97	1.22E-06	2.00
1/64	1/256	3.76E-05	1.53	7.44E-04	0.98	3.05E-07	2.00
1/128	1/512	1.37E-05	1.46	3.62E-04	1.04	7.63E-08	2.00

$$m^*(t,x) = \begin{cases} \dfrac{1}{4}x(x-1)(2x+1), & t = 0, \\ \dfrac{x}{t^2} - \dfrac{3-t}{2t^3}\sqrt{2tx + \left(\dfrac{t}{2}-1\right)^2} & \\ \quad - \dfrac{(t-1)(t^2-4)}{8t^3}\dfrac{1}{\sqrt{2tx + \left(\frac{t}{2}-1\right)^2}} - \dfrac{3t-4}{2t^3}, & 0 < t \le 1. \end{cases} \tag{6.2}$$

We also know $W_2^2(\rho_0, \rho_1) = \dfrac{1}{120}$.

Note that it would be quite difficult to check $\boldsymbol{E}^{(k)}$ as we do not have evolution path, $\rho^{(k)}$ and $\boldsymbol{m}^{(k)}$, in the continuous Algorithm 1. Instead, we compute the following values:

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_2 := \sqrt{\Delta_0\Delta_1}\left[\sum_{\boldsymbol{j}\in\boldsymbol{J}_0}\left|P_{\boldsymbol{j}}^{(k)} - \rho_{\boldsymbol{j}}^*\right|^2 + \sum_{\boldsymbol{j}\in\boldsymbol{J}_1}\left|M_{\boldsymbol{j}}^{(k)} - m_{\boldsymbol{j}}^*\right|^2\right]^{\frac{1}{2}},$$

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_\infty := \max\left\{\max_{\boldsymbol{j}\in\boldsymbol{J}_0}\left|P_{\boldsymbol{j}}^{(k)} - \rho_{\boldsymbol{j}}^*\right|, \max_{\boldsymbol{j}\in\boldsymbol{J}_1}\left|M_{\boldsymbol{j}}^{(k)} - m_{\boldsymbol{j}}^*\right|\right\},$$

$$W_2^2 \text{ error: } \left|\Delta_0\Delta_1\mathcal{Y}\left(P^{(k)}, \boldsymbol{M}^{(k)}\right) - W_2^2(\rho_0,\rho_1)\right|.$$

Here $\left\|\boldsymbol{E}^{(k,*)}\right\|_2$ is related to $\left\|\boldsymbol{E}^{(k)}\right\|_2$ by:

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_2 \le \left\|\boldsymbol{E}^{(k)}\right\|_2 + \left\|\left(\rho_{\boldsymbol{J}_0}^{(k)}, \boldsymbol{m}_{\boldsymbol{J}}^{(k)}\right) - \left(\rho_{\boldsymbol{J}_0}^*, \boldsymbol{m}_{\boldsymbol{J}}^*\right)\right\|_2$$

For given Δ_0, Δ_1, we can choose very large k such that

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_2 \le \left\|\boldsymbol{E}^{(k)}\right\|_2 + \epsilon^{(k)}$$

and $\epsilon^{(k)} \ll \Delta_0 + \Delta_1$. Fixing k, according to our theoretical analysis, we expect to observe at least

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_2 = \mathcal{O}\left(\Delta_0 + \Delta_1\right).$$

and

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_\infty \le \left\|\boldsymbol{E}^{(k,*)}\right\|_F = \left(\Delta_0\Delta_1\right)^{-\frac{1}{2}}\left\|\boldsymbol{E}^{(k,*)}\right\|_2 = \mathcal{O}\left(1\right).$$

Numerical results are shown in Table 1 where we observe

$$\left\|\boldsymbol{E}^{(k,*)}\right\|_2 = \mathcal{O}\left(\Delta_0^{1.5} + \Delta_1^{1.5}\right), \quad \left\|\boldsymbol{E}^{(k,*)}\right\|_\infty = \mathcal{O}\left(\Delta_0 + \Delta_1\right).$$

TABLE 2. Time comparison of OT in 1D for different grid sizes ($n_0 = 64, tol = 10^{-4}$, F=FISTA, A=ALG, G=G-prox) with the best performance highlighted in red.

	Iter			Time (s)			Time(s)/Iter		
n_1	F	A	G	F	A	G	F	A	G
256	611	435	426	1.74	2.99	2.93	2.85E-03	6.86E-03	6.88E-03
512	611	435	429	3.06	5.91	6.92	5.00E-03	1.36E-02	1.61E-02
1024	611	435	430	7.60	12.93	12.85	1.24E-02	2.97E-02	2.99E-02
2048	611	435	431	24.84	36.15	32.72	4.07E-02	8.31E-02	7.59E-02
4096	611	435	431	51.79	69.99	68.09	8.48E-02	1.61E-01	1.58E-01

TABLE 3. Time comparison of OT in 2D for different grid sizes ($n_0 = 64$, F=FISTA, A=ALG, G=G-prox) with the best performance highlighted in red.

	Iter			Time (s)			Time(s)/Iter		
n_1, n_2	F	A	G	F	A	G	F	A	G
128	116	64	66	46.20	46.07	46.49	3.98E-01	7.20E-01	7.04E-01
256	116	64	66	212.31	201.52	190.31	1.83E+00	3.15E+00	2.88E+00
512	116	64	66	810.86	761.65	752.59	6.99E+00	1.19E+01	1.14E+01

This indicates that convergence rate of our numerical experiments perform better than theoretical prediction. This is not surprise as the way of our theoretical analysis may not be sharp.

6.2. **Computation efficiency.** In this part, we would like to demonstrate the efficiency of our algorithms by comparing with state-of-the-art methods for dynamic OT problems. We apply our algorithms to OT problems with ρ_0, ρ_1 being Gaussian distribution densities, and compare the results and computation time with those using ALG(augmented Lagrangian) [11, 12] and G-prox [34]. For all approaches, the stopping criteria are

$$\left\| \left(P^{(k+1)}, \boldsymbol{M}^{(k+1)} \right) - \left(P^{(k)}, \boldsymbol{M}^{(k)} \right) \right\|_2 \leq tol.$$

In Table 2 and Table 3, we report computation time and number of iterations for each algorithms on different grid sizes in 1D and 2D. From the tables, the proposed Algorithm 2 outperforms ALG and G-prox in 1D and achieves similar efficiency in 2D. Interestingly, CPU time per iteration in our algorithm is the least among these three algorithms. This is because, at each iteration, solving a Poisson equation is required for all three algorithms while our method does not need to solve $\prod_{d=0}^{D} n_d$ cubic equations required in ALG and G-prox. Therefore our method needs less time in 1D experiment although it needs more iterations to achieve the given stopping criteria. While, this computation save is marginal comparing with the cost of solving Poisson equation in 2D. Thus, our method spend comparable time instead of less time in this 2D experiment.

Moreover, as shown in Table 4 and Table 5, we further accelerate the proposed algorithm by at most 10 times with the help of multilevel and multigrid strategies.

TABLE 4. Efficiency and accuracy comparisons of OT in 1D ($n_0 = 64, n_1 = 256$) with the best performance highlighted in red.

	Num Iter	Time (s)	Stationarity Residue	Feasibility Residue	Mass Residue
FISTA	611	1.723	3.27E-05	2.28E-13	1.33E-15
ALG	435	2.840	9.43E-05	2.41E-04	1.64E-08
G-prox	426	2.761	1.93E-04	1.88E-04	2.96E-08
MLFISTA	882	0.422	7.97E-05	2.28E-13	1.11E-15
MGFISTA ($K = 5$)	1448	1.195	4.79E-05	2.33E-13	1.77E-15
MGFISTA ($K = 10$)	1517	1.341	3.95E-05	2.28E-13	2.22E-15

TABLE 5. Efficiency and accuracy comparison of OT in 2D ($n_0 = 64, n_1 = n_2 = 256$) with the best performance highlighted in red.

	Num Iter	Time (s)	Stationarity Residue	Feasibility Residue	Mass Residue
FISTA	116	232.560	9.22E-04	6.01E-13	1.42E-14
ALG	64	211.043	8.75E-04	4.99E-03	3.10E-03
G-prox	66	208.696	9.29E-04	6.88E-03	3.10E-03
MLFISTA	162	12.853	3.43E-03	2.95E-13	2.07E-14
MGFISTA ($K = 5$)	315	134.226	1.07E-03	5.99E-13	1.67E-14
MGFISTA ($K = 10$)	315	170.580	9.86E-04	6.00E-13	2.02E-14

We also compute the residue of being a stationary point, residue of feasibility constraint (2.2), and residue of mass conservation to check the accuracy of the solutions. From the residue comparisons listed in the tables, it is clear to see that all of our algorithms provide solutions with far more better mass preservation property than results from ALG and G-prox methods due to the nature of the projection step in our method. Qualitatively, Figure 6 also shows that all 6 algorithms in our experiments provide satisfactory results in accuracy.

Remark 6.1. From Table 4 and Table 5, we observe that with multilevel and multigrid strategies, the algorithms take more number of iterations to converge. This is because we need more iterations on the coarser mesh to obtain a good initialization on the finer mesh. But since each iteration on the coarser mesh is less expensive, schemes with multilevel and multigrid strategy take much less time to complete.

6.3. **MFP with obstacles.** Most numerical examples of MFP in literature consider Ω to be a regular region, i.e. $\Omega = [0, 1] \times [0, 1]$. However, in real application, problems defined in irregular regions might make the implementation very complicated. One potential way of handling irregular domain is to set Q to be an indicator function of obstacles which leads to solutions staying in the irregular domain. In a different example, [42] provides an interesting optimal transport example where the region is a maze with many "walls". Here we consider several illustrative cases where there are one or two pieces of obstacles in our square domain and show

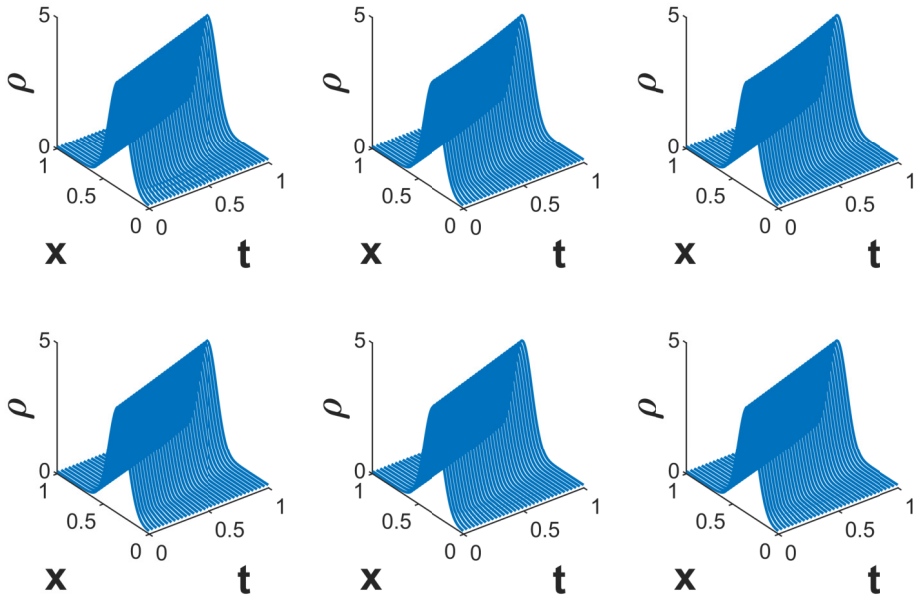


FIGURE 6. Qualitative comparisons of $\rho(t,\cdot)$ in 1D. Row 1 from left to right: FISTA, ALG, G-Prox. Row 2 from left to right: MLFISTA, MGFISTA($K=5$), MGFISTA($K=10$).

that our algorithm can deal with this case without modification of implementation. More detailed study along this direction will be explored in our future work.

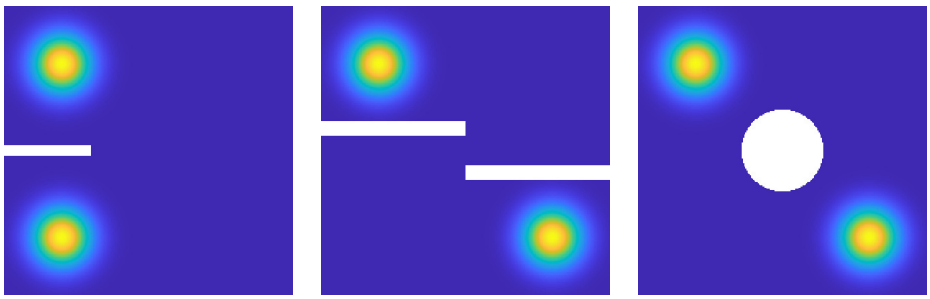
To be precise, letting $\Omega = \left[-\frac{1}{2},\frac{1}{2}\right]\times\left[-\frac{1}{2},\frac{1}{2}\right]$, we consider MFP problem with objective function

$$\int_0^1\int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t+\lambda_Q\int_0^1\int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x},$$

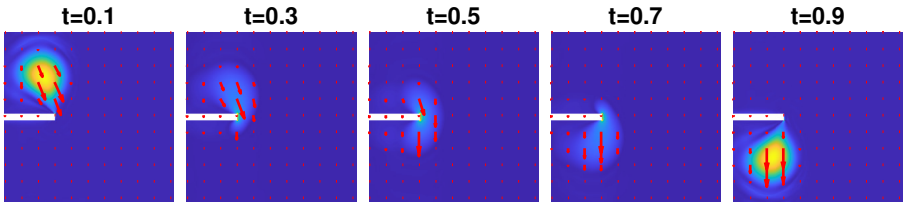
Different choices of ρ_0,ρ_1,Q are shown in the first row of Figure 7 and $Q(\boldsymbol{x})=\begin{cases}1, & \boldsymbol{x}\in\Omega_0\\ 0, & \boldsymbol{x}\notin\Omega_0\end{cases}$ where Ω_0 is the white region. By setting λ_Q to be a very large number (e.g. $\lambda_Q = 8\times10^4$ in our implementation), we expect the set Ω_0 to be viewed as an obstacle and the density evolution to circumvent the region. The snapshots of the evolution shown in Figure 7 demonstrate the success of our algorithm that the mass circumvents the obstacles very well.

6.4. **Flexibility.** As one of the greatest advantages, our method enjoys flexibility to handle different types of objective functions in variational MFP problems. To show the effectiveness of our algorithm, we apply Algorithm 2 to the five models listed in Section 2. We can also observe how different objective functions affect the density evolutions.

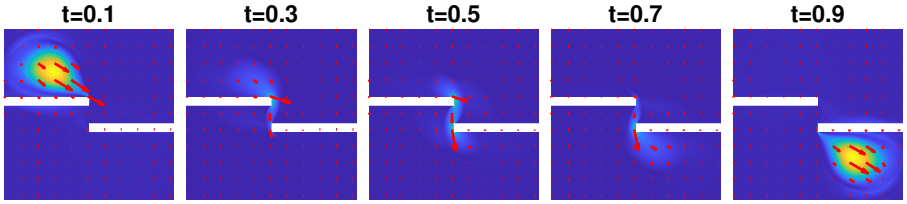
Let $\Omega=[0,1]\times[0,1]$, ρ_0,ρ_1 being two images shown in Figure 8, $G(\boldsymbol{x})=-\rho_1(\boldsymbol{x})$ and $Q(\boldsymbol{x})=\begin{cases}0, & \rho_0(\boldsymbol{x})\neq0 \text{ or } \rho_1(\boldsymbol{x})\neq0,\\ 1, & \text{otherwise},\end{cases}$. We consider MFP problem of the



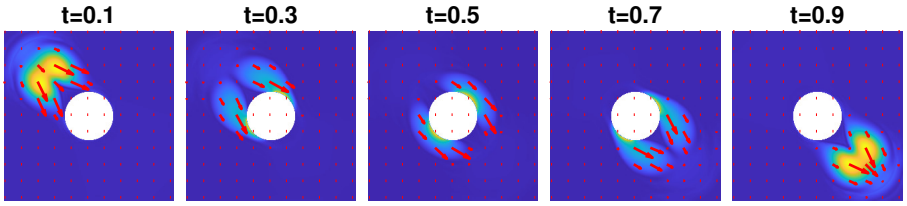
(a) Example 1 (b) Example 2 (c) Example 3



(d) Example 1



(e) Example 2



(f) Example 3

FIGURE 7. (a-c): Initial density ρ_0, terminal density ρ_1 and obstacle region Ω_0 highlighted as white regions. (d-f) Snapshots of ρ at $t = 0.1, 0.3, 0.5, 0.7, 0.9$.

following form

$$\min_{(\rho,\boldsymbol{m})\in\mathcal{C}(\rho_0,\rho_1)} \left\{ \begin{array}{c} \displaystyle\int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}),\boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_E \int_0^1 \int_\Omega F_E(\rho(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t + \lambda_Q \int_0^1 \int_\Omega \rho(t,\boldsymbol{x})Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \end{array} \right\},$$

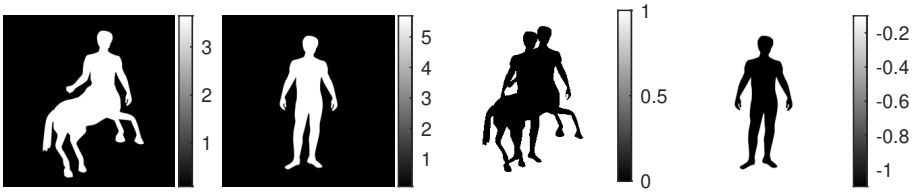


FIGURE 8. From left to right: initial density ρ_0, final ρ_1, interaction penalty $Q(\boldsymbol{x})$ and terminal density regularizer $G(\boldsymbol{x})$

We apply the proposed algorithm to the following four MFP models discussed in example 2.1-2.3:

$$\text{(OT) } \lambda_E = \lambda_Q = 0, \tag{6.3}$$

$$\text{(Model 1) } \lambda_E = 0.01, \lambda_Q = 0.1, F_E : \mathbb{R}^+ \to \mathbb{R}, \rho \mapsto \begin{cases} \rho \log(\rho), & \rho > 0 \\ 0, & \rho = 0 \end{cases} \tag{6.4}$$

$$\text{(Model 2) } \lambda_E = 0.01, \lambda_Q = 0.1, F_E : \mathbb{R}^+ \to \mathbb{R}, \rho \mapsto \frac{\rho^2}{2}, \tag{6.5}$$

$$\text{(Model 3) } \lambda_E = 0.01, \lambda_Q = 0.1, F_E : \mathbb{R}^+ \to \mathbb{R}, \rho \mapsto \begin{cases} \frac{1}{\rho}, & \rho > 0 \\ 0, & \rho = 0 \end{cases} \tag{6.6}$$

and a MFG model shown in example 2.4

$$\min_{(\rho,\boldsymbol{m})\in\mathcal{C}(\rho_0)} \begin{cases} \displaystyle\int_0^1 \int_\Omega L(\rho(t,\boldsymbol{x}), \boldsymbol{m}(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_E \int_0^1 \int_\Omega \rho(t,\boldsymbol{x}) \log(\rho(t,\boldsymbol{x}))\mathrm{d}\boldsymbol{x}\mathrm{d}t + \lambda_Q \int_0^1 \int_\Omega \rho(t,\boldsymbol{x}) Q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \\ \displaystyle +\lambda_G \int_\Omega \rho(1,\boldsymbol{x}) G(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \end{cases} \tag{6.7}$$

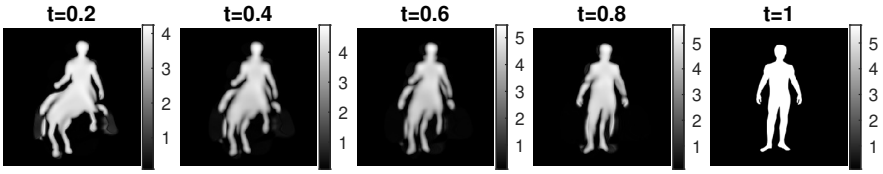
with $\lambda_E = 0.01, \lambda_Q = 0.1, \lambda_G = 1$. It is worth mentioning that to solve model (6.3)-(6.6), we must rescale ρ_1, ρ_1 such that $\int_\Omega \rho_0 = \int_\Omega \rho_1$ but we do not have to rescale $G(\boldsymbol{x})$ for ρ_0 in (6.7).

Figure 9 show the snapshots of the density evolutions. Since (6.5)-(6.7) set the space preference to the evolution, the mass evolutions are within the dark region and the optimal transport model (6.3) has a more free evolution style.

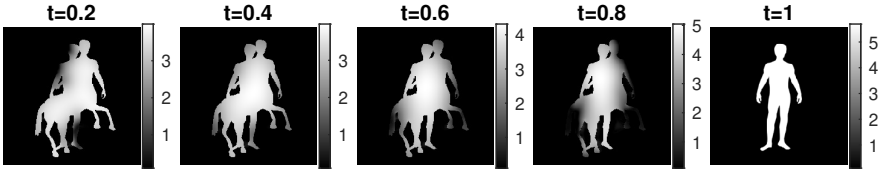
Comparing model (6.4),(6.5) with (6.6), we observe that the mass evolution of model (6.4),(6.5) are dense, while that of (6.6) experiences a congest-flatten process and tends to be sparse. This is compatible with our discussions in Section 2.

7. Conclusion

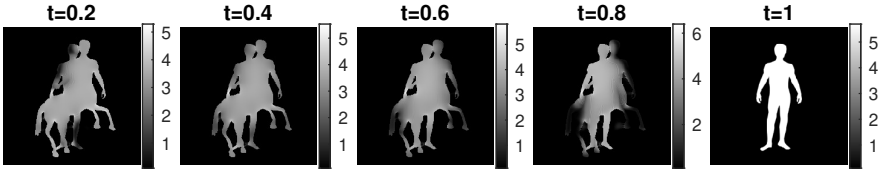
In this paper, we propose an efficient and flexible algorithm to solve potential MFP problems based on an accelerated proximal gradient algorithm. In the optimal transport setting, we can converge faster or nearly as fast as G-prox and approach optimizer with the same accuracy. With multilevel and multigrid strategies, our algorithm can be accelerated up to 10 times without sacrificing accuracy. In broader settings of MFP and MFG, our method is more flexible than primal-dual or dual algorithms as it enjoys the flexibility to handle differentiable objective functions.



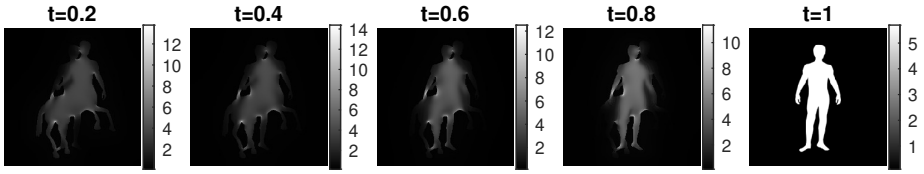
(a) OT: $F_E(a) = F_0(a) := 0$.



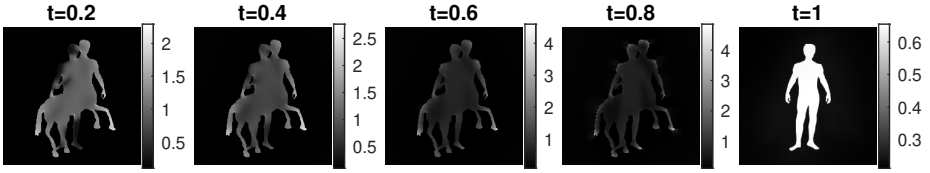
(b) Model 1:$F_E(a) = a \log a, a > 0$.



(c) Model 2: $F_E(a) = \frac{a^2}{2}$.



(d) Model 3: $F_E(a) = \frac{1}{a}, a > 0$.



(e) MFG: $F_E(a) = a \log a, a > 0$.

FIGURE 9. Snapshot of ρ at $t = 0.2, 0.4, 0.6, 0.8, 1$ (from left to right)

Theoretically, we, for the first time based on an optimization point of view, analyze the error introduced by discretizing $\rho, \boldsymbol{m}$, and show that under some mild assumptions, our algorithm converges to the optimizer. In the future, we expect to extend the proposed algorithms for non-potential mean field games, which have vast applications in mathematical finance, communications, and data science.

REFERENCES

1. Yves Achdou, Francisco J Buera, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll, *Partial differential equation models in macroeconomics*, Philosophical Transactions of the

Royal Society A: Mathematical, Physical and Engineering Sciences **372** (2014), no. 2028, 20130397.
2. Yves Achdou, Fabio Camilli, and Italo Capuzzo-Dolcetta, *Mean field games: numerical methods for the planning problem*, SIAM Journal on Control and Optimization **50** (2012), no. 1, 77–109.
3. ———, *Mean field games: convergence of a finite difference method*, SIAM Journal on Numerical Analysis **51** (2013), no. 5, 2585–2612.
4. Yves Achdou and Italo Capuzzo-Dolcetta, *Mean field games: numerical methods*, SIAM Journal on Numerical Analysis **48** (2010), no. 3, 1136–1162.
5. Yves Achdou, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll, *Income and wealth distribution in macroeconomics: A continuous-time approach*, Tech. report, National Bureau of Economic Research, 2017.
6. Yves Achdou and Mathieu Laurière, *Mean field games and applications: Numerical aspects*, arXiv preprint arXiv:2003.04444 (2020).
7. Yves Achdou and Victor Perez, *Iterative strategies for solving linearized discrete mean field games systems*, Networks & Heterogeneous Media **7** (2012), no. 2, 197.
8. Martin Arjovsky, Soumith Chintala, and Léon Bottou, *Wasserstein gan*, arXiv preprint arXiv:1701.07875 (2017).
9. Heinz H Bauschke, Regina S Burachik, Patrick L Combettes, Veit Elser, D Russell Luke, and Henry Wolkowicz, *Fixed-point algorithms for inverse problems in science and engineering*, vol. 49, Springer Science & Business Media, 2011.
10. Amir Beck and Marc Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences **2** (2009), no. 1, 183–202.
11. Jean-David Benamou and Yann Brenier, *A numerical method for the optimal time-continuous mass transport problem and related problems*, Contemporary mathematics **226** (1999), 1–12.
12. ———, *A computational fluid mechanics solution to the monge-kantorovich mass transfer problem*, Numerische Mathematik **84** (2000), no. 3, 375–393.
13. Jean-David Benamou and Guillaume Carlier, *Augmented lagrangian methods for transport optimization, mean-field games and degenerate pdes*, (2014).
14. ———, *Augmented lagrangian methods for transport optimization, mean field games and degenerate elliptic equations*, Journal of Optimization Theory and Applications **167** (2015), no. 1, 1–26.
15. Jean-David Benamou, Guillaume Carlier, and Maxime Laborde, *An augmented lagrangian approach to wasserstein gradient flows and applications*, ESAIM: Proceedings and Surveys **54** (2016), 1–17.
16. Jean-David Benamou, Guillaume Carlier, and Filippo Santambrogio, *Variational mean field games*, Active Particles, Volume 1, Springer, 2017, pp. 141–171.
17. Jean-David Benamou, Brittany D Froese, and Adam M Oberman, *Numerical solution of the optimal transportation problem using the monge–ampère equation*, Journal of Computational Physics **260** (2014), 107–126.
18. Giuseppe Buttazzo, Luigi De Pascale, and Paola Gori-Giorgi, *Optimal-transport formulation of electronic density-functional theory*, Physical Review A **85** (2012), no. 6, 062502.
19. Pierre Cardaliaguet, *Weak solutions for first order mean field games with local coupling*, Analysis and geometry in control theory and its applications, Springer, 2015, pp. 111–158.
20. Pierre Cardaliaguet, Guillaume Carlier, and Bruno Nazaret, *Geodesics for a class of distances in the space of probability measures*, Calculus of Variations and Partial Differential Equations **48** (2013), no. 3, 395–420.
21. Pierre Cardaliaguet and P Jameson Graber, *Mean field games systems of first order*, ESAIM: Control, Optimisation and Calculus of Variations **21** (2015), no. 3, 690–722.
22. Pierre Cardaliaguet, Alpár R Mészáros, and Filippo Santambrogio, *First order mean field games with density constraints: pressure equals price*, SIAM Journal on Control and Optimization **54** (2016), no. 5, 2672–2709.
23. Codina Cotar, Gero Friesecke, and Claudia Klüppelberg, *Density functional theory and optimal transportation with coulomb cost*, Communications on Pure and Applied Mathematics **66** (2013), no. 4, 548–599.
24. Antonio De Paola, Vincenzo Trovato, David Angeli, and Goran Strbac, *A mean field game approach for distributed control of thermostatic loads acting in simultaneous energy-frequency response markets*, IEEE Transactions on Smart Grid **10** (2019), no. 6, 5987–5999.

25. Alfred Galichon, *Optimal transport methods in economics*, Princeton University Press, 2018.
26. Diogo Gomes and João Saúde, *A mean-field game approach to price formation in electricity markets*, arXiv preprint arXiv:1807.07088 (2018).
27. Diogo Gomes and Tommaso Seneci, *Displacement convexity for first-order mean-field games*, arXiv preprint arXiv:1807.07090 (2018).
28. Diogo A Gomes et al., *Mean field games models—a brief survey*, Dynamic Games and Applications **4** (2014), no. 2, 110–154.
29. P Jameson Graber and Alpár R Mészáros, *Sobolev regularity for first order mean field games*, Annales de l'Institut Henri Poincaré C, Analyse non linéaire, vol. 35, Elsevier, 2018, pp. 1557–1576.
30. Olivier Guéant, Jean-Michel Lasry, and Pierre-Louis Lions, *Mean field games and applications*, Paris-Princeton lectures on mathematical finance 2010, Springer, 2011, pp. 205–266.
31. Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent, *Optimal mass transport for registration and warping*, International Journal of computer vision **60** (2004), no. 3, 225–240.
32. Minyi Huang, Peter E Caines, and Roland P Malhamé, *Large-population cost-coupled lqg problems with nonuniform agents: individual-mass behavior and decentralized ε-nash equilibria*, IEEE transactions on automatic control **52** (2007), no. 9, 1560–1571.
33. Minyi Huang, Roland P Malhamé, Peter E Caines, et al., *Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle*, Communications in Information & Systems **6** (2006), no. 3, 221–252.
34. Matt Jacobs, Flavien Léger, Wuchen Li, and Stanley Osher, *Solving large-scale optimization problems with a convergence rate independent of grid size*, SIAM Journal on Numerical Analysis **57** (2019), no. 3, 1100–1123.
35. Jean-Michel Lasry and Pierre-Louis Lions, *Mean field games*, Japanese journal of mathematics **2** (2007), no. 1, 229–260.
36. Wonjun Lee, Rongjie Lai, Wuchen Li, and Stanley Osher, *Generalized unnormalized optimal transport and its fast algorithms*, arXiv preprint arXiv:2001.11530 (2020).
37. Robert Michael Lewis and Stephen G Nash, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM Journal on Scientific Computing **26** (2005), no. 6, 1811–1837.
38. Haoya Li, Yuwei Fan, and Lexing Ying, *A simple multiscale method for mean field games*, arXiv preprint arXiv:2007.04594 (2020).
39. Jialin Liu, Wotao Yin, Wuchen Li, and Yat Tin Chow, *Multilevel optimal transport: a fast approximation of wasserstein-1 distances*, SIAM Journal on Scientific Computing **43** (2021), no. 1, A193–A220.
40. Carlo Orrieri, Alessio Porretta, and Giuseppe Savaré, *A variational approach to the mean field planning problem*, Journal of Functional Analysis **277** (2019), no. 6, 1868–1957.
41. Nicolas Papadakis, *Optimal transport for image processing*, Ph.D. thesis, 2015.
42. Nicolas Papadakis, Gabriel Peyré, and Edouard Oudet, *Optimal transport with proximal splitting*, SIAM Journal on Imaging Sciences **7** (2014), no. 1, 212–238.
43. Gabriel Peyré, Marco Cuturi, et al., *Computational optimal transport*, Foundations and Trends® in Machine Learning **11** (2019), no. 5-6, 355–607.
44. Alessio Porretta, *On the planning problem for a class of mean field games*, Comptes Rendus Mathematique **351** (2013), no. 11-12, 457–462.
45. ———, *On the planning problem for the mean field games system*, Dynamic Games and Applications **4** (2014), no. 2, 231–256.
46. R Tyrrell Rockafellar, *Convex analysis*, vol. 36, Princeton university press, 1970.
47. Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung, *A machine learning framework for solving high-dimensional mean field game and mean field control problems*, Proceedings of the National Academy of Sciences **117** (2020), no. 17, 9183–9193.
48. Cédric Villani, *Optimal transport: old and new*, vol. 338, Springer Science & Business Media, 2008.
49. E Weinan, Jiequn Han, and Qianxiao Li, *A mean-field optimal control formulation of deep learning*, Research in the Mathematical Sciences **6** (2019), no. 1, 10.
50. Chungang Yang, Jiandong Li, Min Sheng, Alagan Anpalagan, and Jia Xiao, *Mean field game-theoretic framework for interference and energy-aware control in 5g ultra-dense networks*, IEEE Wireless Communications **25** (2017), no. 1, 114–121.

51. Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang, *Mean field multi-agent reinforcement learning*, International Conference on Machine Learning, PMLR, 2018, pp. 5571–5580.

Department of Mathematics, Rensselaer Polytechnic Institute, Troy, New York 12180
Email address: yuj12@rpi.edu

Department of Mathematics, Rensselaer Polytechnic Institute, Troy, New York 12180
Email address: lair@rpi.edu

Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208
Email address: wuchen@mailbox.sc.edu

Department of Mathematics, University of California, Los Angeles, Los Angeles, Carlifornia 90095
Email address: sjo@math.ucla.edu