

On Local Linear Convergence of Projected Gradient Descent for Unit-Modulus Least Squares

Trung Vu, *Member, IEEE*, and Raviv Raich, *Senior Member, IEEE*, and Xiao Fu, *Senior Member, IEEE*

Abstract—The unit-modulus least squares (UMLS) problem has a wide spectrum of applications in signal processing, e.g., phase-only beamforming, phase retrieval, radar code design, and sensor network localization. Scalable first-order methods such as projected gradient descent (PGD) have recently been studied as a simple yet efficient approach to solving the UMLS problem. Existing results on the convergence of PGD for UMLS often focus on global convergence to stationary points. As a non-convex problem, only a sublinear convergence rate has been established. However, these results do not explain the fast convergence of PGD frequently observed in practice. This manuscript presents a novel analysis of convergence of PGD for UMLS, justifying the linear convergence behavior of the algorithm near the solution. By exploiting the local structure of the objective function and the constraint set, we establish an exact expression for the convergence rate and characterize the conditions for linear convergence. Simulations show that our theoretical analysis corroborates numerical examples. Furthermore, variants of PGD with adaptive step sizes are proposed based on the new insight revealed in our convergence analysis. The variants show substantial acceleration in practice.

Index Terms—Unit-modulus least squares, projected gradient descent, linear convergence analysis.

I. INTRODUCTION

UNIT-modulus least squares (UMLS) is formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N} \quad & \frac{1}{2} \|\Phi \mathbf{w} - \mathbf{h}\|^2 \\ \text{s.t.} \quad & |w_i|^2 = 1 \text{ for } i = 1, \dots, N, \end{aligned} \quad (1)$$

where $\Phi \in \mathbb{C}^{M \times N}$ and $\mathbf{h} \in \mathbb{C}^M$. This problem arises in numerous machine learning and signal processing applications including, but not limited to, phase-only beamforming [1], [2], phase recovery [3], radar code design [4], [5], sensor network localization [6], and edge computing [7] (see Section II-A for further details).

It is well-known that UMLS is a non-convex NP-hard problem [8]. One traditional approach to this problem is semi-definite relaxation (SDR). In [9], Luo *et al.* recast (1) as a quadratically constrained quadratic programming (QCQP)

Manuscript received ?; revised ?; accepted ?. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ketan Rajawat.

Trung Vu, Raviv Raich, and Xiao Fu are with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, USA. E-mails: vutru@oregonstate.edu; raich@oregonstate.edu; and xiao.fu@oregonstate.edu

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material includes proofs of Lemmas 2-7 and a demonstration of the region of convergence in a 2-D setting. This material is 442KB in size.

problem and then lifted it to an N^2 -dimensional problem with a rank-1 constraint. By dropping the non-convex rank constraint, the resulting problem is convex and can be solved via interior point methods. The major disadvantage of SDR is the high computational complexity ($O(N^7)$ flops and $O(N^2)$ memory units), which is not suitable for large-scale problems in modern applications. Another approach that has recently been proposed by Tranter *et al.* [1] is projected gradient descent (PGD). Since the projection onto the unit-modulus manifold is simple and low-cost, PGD is shown to be efficient in large-scale settings. Notably, the authors in [1] show that despite the lack of convexity, the algorithm converges globally to a set of stationary points of (1) and the rate of convergence is at least sublinear. Recently, Zhang *et al.* [2] extend the fixed-step-size algorithm to variable-step-size versions that empirically improve the convergence speed of PGD. Nonetheless, the authors prove that all limit points of the iterates converge to the KKT points, no further result on the convergence speed is provided.

Motivated by the aforementioned results, this manuscript provides an in-depth convergence analysis of PGD for UMLS. First, we observe in practice that the algorithm frequently exhibits linear convergence near a local minimum of the problem. This is significantly faster than the sublinear convergence proven in [1]. Second, the bounding technique in [1] is rather conservative since it focuses on global characterization yet ignores the local structure of the problem around the solution. In particular, while UMLS is not a globally convex problem, it can still possess a benign geometry around a local minimum. In such a scenario, one can expect that PGD will converge linearly to the local minimum similar to gradient descent for unconstrained minimization of a smooth and strongly convex function [10]. With this intuition, our goal here is to provide an analytical framework to uncover the fast linear convergence behavior of PGD near a local minimum of the UMLS problem.¹ The UMLS problem is an instance of constrained least squares (CLS) and the local convergence analysis for PGD naturally follows from the unified framework in our previous work [12]. Under a more general setting of the constraint set, we showed the conditions for asymptotic linear convergence, the convergence rate, and the region of convergence. However, the application of the general theory in [12] needs to be

¹Preliminary aspects of this work appeared in an earlier conference version [11], where we study the local convergence of PGD for minimizing a quadratic over the unit sphere. When $N = 1$, the UMLS problem and the spherically constrained least squares problem coincide. For $N > 1$, UMLS introduces a more complex constraint set in the form of the cross-product of multiple spherical constraints.

customized for each specific problem. As elaborated later in Section III-A, this work analyzes a number of aspects that the framework in [12] did not cover and offers a more concise and elegant analysis route, skipping some steps used in [12].

Our contribution in this work is three-fold. First, by exploiting the structure of the problem near local minima, we are able to identify the sufficient conditions for local linear convergence of PGD with a fixed step size and obtain an exact expression of the convergence rate. Second, we establish the region of convergence in which initializing the algorithm is guaranteed to converge to the desired local minimum. The theoretical rate predicts accurately the empirical convergence rate in our numerical simulation. Third, in practical applications where prior knowledge of the solution is not available, we propose two adaptive-step-size variants of PGD that require the same iteration complexity while offering faster linear convergence compared to the optimal fixed step size in theory.

The rest of the paper is organized as follows. Section II presents some motivating examples of the UMLS problem and its real-valued formulation along with the PGD algorithm for solving UMLS. Section III summarizes existing results on the convergence of PGD for UMLS in the literature, highlighting the fundamental similarity between the UMLS problem and the spherically constrained least squares problem. Our convergence analysis is presented in Section IV, including solution properties, algorithm properties, and linear convergence properties. In Section V, we propose two variants of PGD for UMLS that use adaptive step size schemes to effectively obtain fast linear convergence without prior knowledge of the solution. Finally, in Section VI, we perform numerical experiments to verify our theoretical analysis.

Notation: Throughout the paper, we use the notations $\|\cdot\|_F$ and $\|\cdot\|_2$ to denote the Frobenius norm and the spectral norm of a matrix, respectively. Additionally, $\|\cdot\|$ is used on a vector to denote the Euclidean norm. Boldfaced symbols are reserved for vectors and matrices. $(\cdot)^*$, $(\cdot)^\top$, and $(\cdot)^H$ denote the complex conjugate, the transpose, and the Hermitian transpose, respectively. The $t \times t$ identity matrix is denoted by \mathbf{I}_t . The t -dimensional vector of all zeros and the t -dimensional vector of all ones are denoted by $\mathbf{0}_t$ and $\mathbf{1}_t$, respectively. The notations \otimes denotes the Kronecker product between two matrices and $\text{vec}(\cdot)$ denotes the vectorization of a matrix by stacking its columns on top of one another. For a complex number z , $\Re(z)$ and $\Im(z)$ denote the real and imaginary parts of z , respectively. Given an n -dimensional vector \mathbf{x} , x_i denotes its i th element and $\text{diag}(\mathbf{x})$ denotes the $n \times n$ diagonal matrix with the corresponding diagonal entries x_1, \dots, x_n . Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the i th largest eigenvalue and the i th largest singular value of \mathbf{X} are denoted by $\lambda_i(\mathbf{X})$ and $\sigma_i(\mathbf{X})$, respectively. The spectral radius of \mathbf{X} is defined as $\rho(\mathbf{X}) = \max_i |\lambda_i(\mathbf{X})|$ and is less than or equal to the spectral norm, i.e., $\rho(\mathbf{X}) \leq \|\mathbf{X}\|_2$ [13]. If \mathbf{X} is square and invertible, the condition number of \mathbf{X} is defined as $\kappa(\mathbf{X}) = \sigma_1(\mathbf{X})/\sigma_n(\mathbf{X})$. Finally, we use $\mathbf{X} \succ 0$ to indicate the matrix \mathbf{X} is positive definite (PD) and $\mathbf{X} \succeq 0$ to indicate the matrix \mathbf{X} is positive semi-definite (PSD).

II. PROBLEM STATEMENT

In this section, we present three applications that motivate the use of unit-modulus least squares and proceed with introducing fundamental concepts in formulating the UMLS problem as a standard constrained least squares optimization and the PGD algorithm for solving it.

A. Motivation

Phase-only Beamforming. In transmit beamforming, we wish to design a weight vector $\mathbf{w} \in \mathbb{C}^N$ associated with N antennas to enhance the transmission of the signals towards certain directions while suppressing the transmission towards other directions in an effort to avoid interference. For a uniform linear array (ULA) of N antennas with M discretization of the angle space (see Fig. 1(a)), the goal is to design a weight vector \mathbf{w} such that it linearly combines the signals from N antennas to form a desired output signal $y_m = \mathbf{w}^H \mathbf{a}(\theta_m)$, for $m = 1, 2, \dots, M$, where $\mathbf{a}(\theta) = [1, e^{-j\frac{2\pi\Delta}{\lambda} \sin \theta}, \dots, e^{-j(N-1)\frac{2\pi\Delta}{\lambda} \sin \theta}]^\top$ is the steering vector associated with $\theta \in [-\pi, \pi]$. Here, λ is the wavelength and Δ is the array spacing. In addition, modern beamforming applications often involve large-scale settings (e.g., massive multiple-input-multiple-output (MIMO) systems) in which hardware such as power amplifiers can be costly. It has been shown [14], [15] that by constraining \mathbf{w} to have constant modulus (i.e., $|w_i| = 1$ for all $i = 1, \dots, N$), one can reduce hardware complexity while still effectively produce the desired beam patterns. Thus, the phase-only beamforming problem can be formulated as a UMLS problem of (1)

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N} \quad & \frac{1}{2} \|\mathbf{A}\mathbf{w} - \mathbf{y}\|^2 \\ \text{s.t.} \quad & |w_i|^2 = 1 \text{ for } i = 1, \dots, N, \end{aligned} \quad (2)$$

where $\mathbf{A} = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_M)]^\top$ and $\mathbf{y} = [y_1, y_2, \dots, y_M]^\top$. Figure 1(b) illustrates an example of the phase-only beamforming aimed towards one user in the direction centered 0.

Phase Recovery. The phase recovery problem aims to find a signal $\mathbf{x} \in \mathbb{C}^n$ from the magnitude of its linear measurement $\mathbf{b} = |\mathbf{A}\mathbf{x}|$, where $\mathbf{A} \in \mathbb{C}^{m \times n}$ where $|\cdot|$ is applied to a vector elementwise. In [3], Waldspurger *et al.* proposed to explicitly separate the amplitude and phase variables and only optimize the values of the phase variables. By representing $\mathbf{A}\mathbf{x} = \text{diag}(\mathbf{b})\mathbf{u}$, where $\mathbf{u} \in \mathbb{C}^m$ satisfying $|u_i| = 1$, for $i = 1, \dots, m$, the phase recovery problem is formulated as

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{C}^n, \mathbf{u} \in \mathbb{C}^m} \quad & \frac{1}{2} \|\mathbf{A}\mathbf{x} - \text{diag}(\mathbf{b})\mathbf{u}\|^2 \\ \text{s.t.} \quad & |u_i|^2 = 1 \text{ for } i = 1, \dots, m. \end{aligned} \quad (3)$$

Minimizing the least squares w.r.t \mathbf{x} yields $\mathbf{x} = \mathbf{A}^\dagger \text{diag}(\mathbf{b})\mathbf{u}$, where $\mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$ is the pseudo inverse of \mathbf{A} , and substituting back into (3) yields a standard UMLS problem

$$\begin{aligned} \min_{\mathbf{u} \in \mathbb{C}^m} \quad & \frac{1}{2} \|(\mathbf{A}\mathbf{A}^\dagger - \mathbf{I}_m) \text{diag}(\mathbf{b})\mathbf{u}\|^2 \\ \text{s.t.} \quad & |u_i|^2 = 1 \text{ for } i = 1, \dots, m. \end{aligned} \quad (4)$$

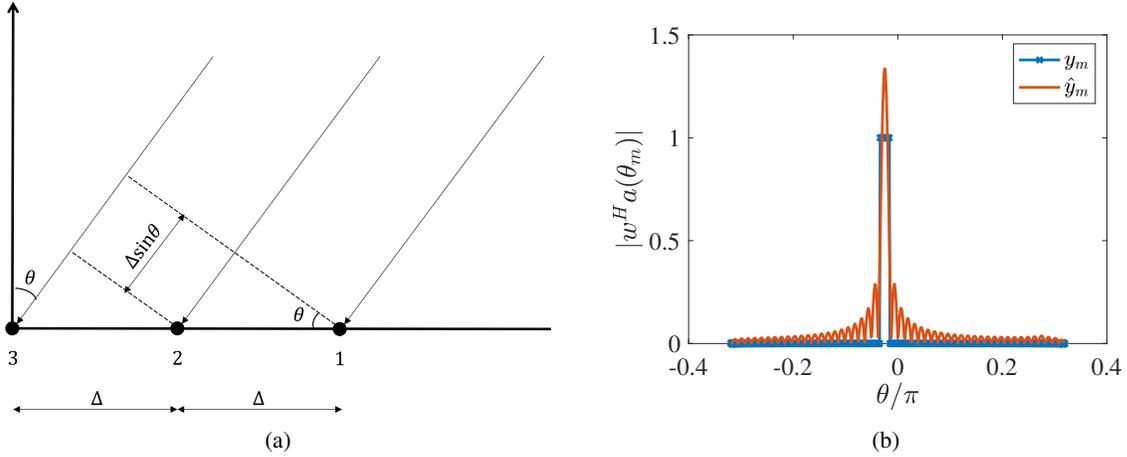


Fig. 1: (a) Geometry of a uniform linear array. (b) Phase-only beamforming: forming a beampattern aimed towards the direction $[-\pi/60, \pi/60]$ by solving (2). The blue line represents the desired \mathbf{y} with $y_m = 1$ if $-\pi/60 \leq \theta_m \leq \pi/60$ and $y_m = 0$ otherwise. The red line represents the phase-only beamforming solution $\hat{\mathbf{y}}$, where $\hat{y}_m = |\mathbf{w}^H \mathbf{a}(\theta_m)|$, for $m = 1, \dots, M$.

Note that this problem maps to (1) with $\Phi = (\mathbf{A}\mathbf{A}^\dagger - \mathbf{I}_m) \text{diag}(\mathbf{b})$ and $\mathbf{h} = \mathbf{0}$.

Radar Code Design. We consider a monostatic radar system that transmits a linearly encoded burst of pulses. The vector model for the backscattered signal is given by [16]

$$\mathbf{v} = \alpha(\mathbf{c} \odot \mathbf{p}) + \mathbf{w},$$

where α accounts for the channels propagation effects, \mathbf{w} denotes the noise component, \mathbf{c} is the code vector, and \mathbf{p} is the temporal steering vector. Here, \odot is the Hadamard (element-wise) product. Assuming that \mathbf{w} is a zero-mean complex-valued circular Gaussian vector with known positive definite covariance matrix $\Sigma = \mathbb{E}[\mathbf{w}\mathbf{w}^H]$. The signal-to-noise ratio (SNR) is given by

$$\text{SNR} = |\alpha|^2 \mathbf{c}^H (\Sigma^{-1} \odot (\mathbf{p}\mathbf{p}^H)^*) \mathbf{c}.$$

Our goal is to design the codes that optimize the SNR of the radar system. Moreover, in many active sensing and communication systems, these codes are often desired to be unimodular as they have an optimal peak-to-average-power ratio (PAR) [5]. Therefore, unimodular code design can be formulated as a unit-modulus quadratic programming

$$\begin{aligned} \min_{\mathbf{c} \in \mathbb{C}^N} \quad & |\alpha|^2 \mathbf{c}^H (\Sigma^{-1} \odot (\mathbf{p}\mathbf{p}^H)^*) \mathbf{c} \\ \text{s.t.} \quad & |c_i|^2 = 1 \text{ for } i = 1, \dots, N. \end{aligned} \quad (5)$$

Let $\mathbf{D}_p = \text{diag}(\mathbf{p})$ and $\Phi = |\alpha| \Sigma^{-1/2} \mathbf{D}_p$, then (5) can be mapped to the UMLS problem (1) with $\mathbf{h} = 0$.

B. Real-valued Formulation of the UMLS Problem

For the convenience of analysis, we consider the following real-valued reparametrization of (1):

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^{2N}} \quad & \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \\ \text{s.t.} \quad & x_{2i-1}^2 + x_{2i}^2 = 1 \text{ for } i = 1, \dots, N, \end{aligned} \quad (6)$$

where $\mathbf{A} \in \mathbb{R}^{2M \times 2N}$ is partitioned into 2×2 blocks of form

$$\tilde{\mathbf{A}}_{ij} = \begin{bmatrix} \Re(\Phi_{ij}) & -\Im(\Phi_{ij}) \\ \Im(\Phi_{ij}) & \Re(\Phi_{ij}) \end{bmatrix}, \quad (7)$$

for $i = 1, \dots, M$ and $j = 1, \dots, N$. In addition, $\mathbf{x} = [\Re(w_1), \Im(w_1), \dots, \Re(w_N), \Im(w_N)]^\top$ and $\mathbf{b} = [\Re(h_1), \Im(h_1), \dots, \Re(h_M), \Im(h_M)]^\top$ are real-valued vectors. Next, we introduce the concepts of the 2-selection operator that selects the i th coordinate pair of a $2N$ -dimensional vector. Since the unit-modulus constraint involves every pair of coordinates of \mathbf{x} , this operator allows us to simplify the representation of our result throughout the rest of the paper:

Definition 1. For each $i = 1, \dots, N$, the i th 2-selection operator is defined by $\mathbf{S}_i: \mathbb{R}^{2N} \rightarrow \mathbb{R}^2$ such that

$$\mathbf{S}_i(\mathbf{x}) = \begin{bmatrix} x_{2i-1} \\ x_{2i} \end{bmatrix},$$

where $\mathbf{x} = [x_1, x_2, \dots, x_{2N}]^\top$.

It is noteworthy that the 2-selection operator is linear. Using this operator, we can represent any vector $\mathbf{x} \in \mathbb{R}^{2N}$ as

$$\mathbf{x} = \sum_{i=1}^N \mathbf{e}_i \otimes \mathbf{S}_i(\mathbf{x}), \quad (8)$$

where \mathbf{e}_i is the i th vector in the natural basis of \mathbb{R}^N . Now we define the constraint set of the UMLS problem (10) based on the 2-selection operator.

Definition 2. The unit-modulus set is defined by

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{2N} : \|\mathbf{S}_i(\mathbf{x})\|^2 = 1, \forall i = 1, \dots, N\}. \quad (9)$$

Using Definition 2, one can rewrite the optimization problem (6) as follows

$$\min_{\mathbf{x} \in \mathcal{C}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2. \quad (10)$$

For convenience, we denote the objective $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$.

Algorithm 1: Projected Gradient Descent (PGD)

Input: $\mathbf{x}^{(0)} \in \mathbb{R}^{2N}$ **Output:** $\{\mathbf{x}^{(k)}\}_{k=0}$

1: **for** $k = 0, 1, \dots$ **do**
 2: $\mathbf{x}^{(k+1)} = \mathcal{P}_C(\mathbf{x}^{(k)} - \eta \mathbf{A}^\top(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}))$
 3: \triangleright where \mathcal{P}_C is defined in (13)

C. Projected Gradient Descent for UMLS

To define the projection onto the unit-modulus set \mathcal{C} , let us introduce the distance function from a point $\mathbf{x} \in \mathbb{R}^{2N}$ to \mathcal{C} as

$$d(\mathbf{x}, \mathcal{C}) = \inf_{\mathbf{y} \in \mathcal{C}} \{\|\mathbf{y} - \mathbf{x}\|\}. \quad (11)$$

The set of all projections of \mathbf{x} onto \mathcal{C} is then given by

$$\Pi_{\mathcal{C}}(\mathbf{x}) = \{\mathbf{y} \in \mathcal{C} \mid \|\mathbf{y} - \mathbf{x}\| = d(\mathbf{x}, \mathcal{C})\}. \quad (12)$$

It is well-known [17] that if \mathcal{C} is closed, then for any $\mathbf{x} \in \mathbb{R}^n$, $\Pi_{\mathcal{C}}(\mathbf{x})$ is non-empty. Additionally, since the unit-modulus set \mathcal{C} is non-convex, $\Pi_{\mathcal{C}}(\mathbf{x})$ can have more than one element. An orthogonal projection onto \mathcal{C} is defined as $\mathcal{P}_C : \mathbb{R}^{2N} \rightarrow \mathcal{C}$ such that $\mathcal{P}_C(\mathbf{x})$ is chosen as an element of $\Pi_{\mathcal{C}}(\mathbf{x})$ based on a prescribed scheme (e.g., based on lexicographic order). In particular, we define the orthogonal projection $\mathcal{P}_C(\mathbf{x})$ as projecting each coordinate pair of $\mathbf{x} \in \mathbb{R}^{2N}$ onto the unit 1-sphere

$$\mathcal{S}_i(\mathcal{P}_C(\mathbf{x})) = \begin{cases} \frac{\mathcal{S}_i(\mathbf{x})}{\|\mathcal{S}_i(\mathbf{x})\|} & \text{if } \mathcal{S}_i(\mathbf{x}) \neq \mathbf{0}_2, \\ [1, 0]^\top \triangleq \mathbf{s} & \text{if } \mathcal{S}_i(\mathbf{x}) = \mathbf{0}_2, \end{cases} \quad (13)$$

for $i = 1, \dots, N$, where $\mathcal{S}_i(\cdot)$ is given in Definition 1. It is noted that when $\mathcal{S}_i(\mathbf{x}) = \mathbf{0}_2$, the set of projections of $\mathbf{0}_2$ onto the unit 1-sphere is non-singleton, i.e., the entire unit 1-sphere. In such case, we choose a certain element \mathbf{s} in this set (e.g., $[1, 0]^\top$) as the value of $\mathcal{S}_i(\mathcal{P}_C(\mathbf{x}))$. We emphasize that this choice of projection does not affect our subsequent analysis of local convergence.

Starting from some initial point $\mathbf{x}^{(0)}$, the PGD algorithm for solving (10) performs the following iterative update (see Algorithm 1):

$$\mathbf{x}^{(k+1)} = \mathcal{P}_C(\mathbf{x}^{(k)} - \eta \mathbf{A}^\top(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})), \quad (14)$$

where $\eta > 0$ is a fixed step size. In the literature, PGD is also known as the gradient projection (GP) algorithm (e.g., [1]).

III. PRELIMINARIES

A. PGD for Constrained Least Squares

As mentioned in Section I, this work is a non-trivial application of the unified convergence analysis framework in [12]. In particular, we analyze a number of aspects that the general framework did not cover and offer a more concise and elegant analysis route, skipping some steps used in [12]. We establish the connection between the convergence of PGD and fundamental properties of the problem, such as the Riemannian Hessian \mathbf{H} and the Lagrange multiplier γ (see Table I). Here, the scale-invariance property of the projection onto the unit-modulus set removes the need of analyzing the

regularity of the projection at $\mathbf{z}_\eta^* = \mathbf{x}^* - \eta \mathbf{A}^\top(\mathbf{A}\mathbf{x}^* - \mathbf{b})$ as in [12], resulting in a more elegant analysis (see our proof of Theorem 1 in Section IV-D). If we were simply following the steps in the general framework, one needs to extend the first-order expansion of the projection onto the unit-modulus set to the case $\mathbf{x} \notin \mathcal{C}$ and continue the more cumbersome derivation thereby. While in [12] we demonstrate the simple spherically-constrained LS problem, this is not straightforward for UMLS, considering that there are multiple spherical constraints applying simultaneously on all pairs of coordinates of the optimization variable. That said, the main focus in this work is not on the refined analysis details but more importantly on the outcome of the analysis for the specific UMLS problem. The closed-form expression of the rate as well as the connection between the convergence of PGD and fundamental properties of the problem can be used as a benchmark for analytical comparison with other algorithms and help in better understanding their performance difference.

In [18], Luenberger et. al. considers the general case of equality-constrained minimization:

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s. t. } \mathbf{h}(\mathbf{x}) = \mathbf{0},$$

and study the asymptotic convergence of the geodesic descent algorithm. In [18] - Eqn. (31), the authors provide the asymptotic rate of linear convergence (on the function side) as

$$\rho^* = \left(\frac{A - a}{A + a} \right)^2,$$

where A and a are the largest and smallest eigenvalues of the Hessian of the Lagrangian at the local solution \mathbf{x}^* , restricted to the tangent space to the constraint set at \mathbf{x}^* . This result has some similarities with our local convergence result in the context of UMLS. First, both works consider an optimization problem with equality constraints, where each $h_i(\mathbf{x})$ can be viewed as $\|\mathcal{S}_i(\mathbf{x})\|^2 - 1$, for $i = 1, \dots, N$. Second, both the algorithm considered in [18] and our PGD algorithm utilize the first-order gradient information and the projection operator onto the constraint set. Third, both analyses offer linear convergence guarantees with the rates depending on the local curvature of the problem at the solution. However, we would like to highlight the **differences** between the two results as follows. First, the algorithm considered in [18] is not the same as the projected gradient descent algorithm used in our work. In fact, Luenberger considers the geodesics descent algorithm, which uses the projection of the gradient of the objective function onto the tangent plane at $\mathbf{x}^{(k)}$ as a descent direction. Then, the algorithm performs an exact line search to move along the corresponding geodesic of the constraint surface. Compared to the geodesics descent algorithm, our PGD algorithm is simpler as we first take a fixed step toward the gradient of the objective function and then project it back onto the constraint set. Second, since the geodesics descent algorithm uses an exact line search scheme for the step size, the rate of convergence is provided as the optimal rate that only depends on the local curvature. On the other hand, our rate of convergence depends on the step size η of PGD. Optimizing our rate over η yields similar merit (see Eqn. (26)). Third, the proof technique in [18] relies on Kantorovich's inequality

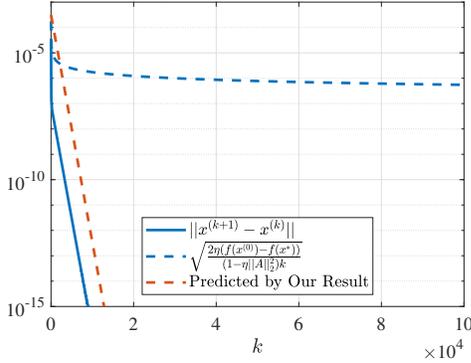


Fig. 2: Plot of $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$ (blue solid) generated by PGD for UMLS with a fixed step size $\eta = 0.9/\|\mathbf{A}\|_2^2$. The blue dashed line represents the sublinear bound given by (15). The red dashed line is based on our linear upper bound proposed in this work. Further details of the data generated for this figure are given later in our simulation in Section VI.

and the optimal property of exact line search (see Eqn. (54) in [18]-Chapter 12). In this work, we follow a different path with a fixed step size choice that linearizes the projection operator in the neighborhood of the solution.

B. Existing Convergence Results on PGD for UMLS

The sublinear convergence of PGD to a set of stationary points of UMLS was studied in [1]. First, Tranter *et al.* showed that any limiting point \mathbf{x}^* of the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ generated by Algorithm 1 is also a stationary point of (10). Second, they proved that for PGD with a fixed step size $0 < \eta < 1/\|\mathbf{A}\|_2^2$, the convergence of $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ to a set of stationary points of (10) is sublinear. In particular, the authors provided a sublinear bound on the distance between two consecutive iterates as follows²

$$\min_{0 \leq l \leq k-1} \|\mathbf{x}^{(l+1)} - \mathbf{x}^{(l)}\| \leq \sqrt{\frac{2\eta(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*))}{(1 - \eta\|\mathbf{A}\|_2^2)k}}. \quad (15)$$

However, it is noted that the sublinear bound given by (15) is based on the worst-case analysis. In practice, we observe the algorithm enjoys fast linear convergence to a local minimum \mathbf{x}^* of (10). Figure 2 illustrates the striking difference between the sublinear bound on $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$ given by the RHS of (15) (blue dashed line) and the corresponding linearly converging empirical value obtained by running the PGD algorithm (blue solid line). The additional bound on $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$ (red dashed line) is derived from the bound on $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|$ given by (25) in the next section and the application of triangle inequality: $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| + \|\mathbf{x}^{(k)} - \mathbf{x}^*\|$. We observe that the red dashed line and the blue solid line are parallel to each other, while the blue dashed line deviates quickly from the other two lines as k increases. In the next

²We note that in [1], the authors actually derived the convergence bound on a surrogate function $Q(\cdot)$ that quantifies the stationarity condition of (10). From Eqn. (23b) in [1], we have the value of $Q(\cdot)$ at iteration k equals to $\frac{1}{\eta^2} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|^2$. In the literature, such convergence metric is related to the generalized gradient norm, (e.g., [19]-Section 2.3.2).

section, we study this unexplained convergence phenomenon of PGD for UMLS. We will provide exact formulations of the linear convergence rate and the region of convergence. The selection of the fixed step size $0 < \eta < 1/\|\mathbf{A}\|_2^2$ in [1] is conservative as it may exclude the optimal choice of η . We will demonstrate in our simulation that larger step sizes enable faster convergence of PGD for UMLS.

C. Least Squares with Unit-Norm Constraint

A closely-related problem to UMLS is the unit-norm least squares (UNLS)

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^N} \quad & \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|^2 = 1, \end{aligned} \quad (16)$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{b} \in \mathbb{R}^N$. While UMLS requires each of the N coordinate pairs of the solution lies on the unit 1-sphere, UNLS requires the solution itself lies on the $N - 1$ -sphere. Unlike the case of unit-modulus constraint, minimizing a quadratic form over the unit sphere is not NP-hard and is solvable as an eigenvalue problem [20], [21]. The convergence of PGD for UNLS has recently been studied in [11], [22]. Table I summarizes the existing convergence result on UNLS and the new convergence result on UMLS we derive in this paper, highlighting the connection between the two works.

IV. CONVERGENCE ANALYSIS

This section presents the convergence analysis of PGD for UMLS. We begin with the properties of the solution of the problem and the PGD algorithm. Next, we present the main result on the convergence of PGD for UMLS. Finally, we provide detailed proof at the end of the section.

A. Solution Properties

The Lagrange function corresponding to (10) is given by

$$L(\mathbf{x}, \gamma) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 - \frac{1}{2} \sum_{i=1}^N \gamma_i (x_{2i-1}^2 + x_{2i}^2 - 1),$$

where $\gamma \in \mathbb{R}^N$ is the Lagrange multiplier. The derivatives of L with respect to \mathbf{x} can be computed as

$$\begin{cases} \nabla_{\mathbf{x}} L(\mathbf{x}, \gamma) = \mathbf{A}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) - (\text{diag}(\gamma) \otimes \mathbf{I}_2) \mathbf{x}, \\ \nabla_{\mathbf{x}}^2 L(\mathbf{x}, \gamma) = \mathbf{A}^\top \mathbf{A} - \text{diag}(\gamma) \otimes \mathbf{I}_2. \end{cases} \quad (17)$$

It can be shown that any feasible point $\mathbf{x} \in \mathcal{C}$ is also a regular point of the constraint set. Specifically, we first represent the constraints as $\mathbf{h} : \mathbb{R}^{2N} \rightarrow \mathbb{R}^N$ such that $\mathbf{h}(\mathbf{x}) = \mathbf{0}_N$, where $h_i(\mathbf{x}) = \|\mathbf{S}_i(\mathbf{x})\|^2 - 1$ for $i = 1, \dots, N$. Then, the Jacobian of all the constraints at \mathbf{x} , defined as $J_{ij} = \partial h_i(\mathbf{x}) / \partial x_j$, is given by

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_1^\top \otimes \mathbf{S}_1^\top(\mathbf{x}) \\ \dots \\ \mathbf{e}_N^\top \otimes \mathbf{S}_N^\top(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{N \times 2N}.$$

³This is a more intuitive but not the most general constraint on the step size. The original version of this condition on the step size is given in Theorem 1.

	Unit-norm constraint [11]	Unit-modulus constraint (this work)
Problem formulation	$\min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \ \mathbf{A}\mathbf{x} - \mathbf{b}\ ^2$ s.t. $\ \mathbf{x}\ = 1$	$\min_{\mathbf{x} \in \mathbb{R}^{2N}} \frac{1}{2} \ \mathbf{A}\mathbf{x} - \mathbf{b}\ ^2$ s.t. $\ \mathbf{S}_i(\mathbf{x})\ = 1, \forall i = 1, \dots, N$
First-order necessary condition	$\exists \gamma \in \mathbb{R} : \mathbf{A}^\top(\mathbf{A}^\top \mathbf{x}^* - \mathbf{b}) = \gamma \mathbf{x}^*$	$\exists \gamma \in \mathbb{R}^N : \mathbf{A}^\top(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = (\text{diag}(\gamma) \otimes \mathbf{I}_2) \mathbf{x}^*$
Reduced Riemannian Hessian	$\mathbf{H} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} \mathbf{Z} - \gamma \mathbf{I}_N$	$\mathbf{H} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} \mathbf{Z} - \text{diag}(\gamma)$
Second-order <i>necessary</i> condition	$\mathbf{H} \succeq \mathbf{0}_N$	$\mathbf{H} \succeq \mathbf{0}_N$
Second-order <i>sufficient</i> condition	$\mathbf{H} \succ \mathbf{0}_N$	$\mathbf{H} \succ \mathbf{0}_N$
Fixed-point condition on step size	$1 - \eta\gamma > 0$	$\mathbf{I}_N - \eta \text{diag}(\gamma) \succ \mathbf{0}_N$
Convergence condition on the step size	$\eta(\lambda_1(\mathbf{H}) + 2\gamma) < 2$	$\eta(\lambda_1(\mathbf{H}) + 2 \max_i \gamma_i) < 2$ (³)
Linear convergence rate	$\rho(\mathbf{I}_N - \eta(1 - \eta\gamma)^{-1} \mathbf{H})$	$\rho(\mathbf{I}_N - \eta(\mathbf{I}_N - \eta \text{diag}(\gamma))^{-1} \mathbf{H})$

TABLE I: Comparison between the existing convergence analysis of PGD for least squares with unit-norm constraint [11] and the novel convergence analysis of PGD for unit-modulus constraint proposed in this paper. In each case, \mathbf{x}^* is a stationary point and \mathbf{Z} is a basis matrix for the null space of the Jacobian of all constraints at \mathbf{x}^* .

Since $\mathbf{J}(\mathbf{x})$ is full row rank for any $\mathbf{x} \in \mathcal{C}$, \mathbf{x} is a regular point of the constraint set (see Chapter 11 in [18]). The following lemma establishes the first-order necessary conditions for local optima of UMLS problems.

Lemma 1. *The first-order necessary conditions for $\mathbf{x}^* \in \mathbb{R}^{2N}$ to be a local minimum of (10) are $\mathbf{x}^* \in \mathcal{C}$ and there exists a Lagrange multiplier $\gamma \triangleq \gamma(\mathbf{x}^*) \in \mathbb{R}^N$ such that*

$$\mathbf{A}^\top(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = (\text{diag}(\gamma) \otimes \mathbf{I}_2) \mathbf{x}^*. \quad (18)$$

Any point satisfying the foregoing first-order necessary conditions is called a **stationary point** of (10).

By setting $\nabla_{\mathbf{x}} L(\mathbf{x}, \gamma)$ in (17) to $\mathbf{0}$, the proof of Lemma 1 follows the same derivation in [18]-Chapter 11.3. Next, we examine the second-order conditions for local optima of problem (10) via the basis of the tangent space to \mathcal{C} at \mathbf{x}^* . The following lemma provides further insight into these conditions.

Lemma 2. *Let \mathbf{x}^* be a stationary point of problem (10) with the corresponding Lagrange multiplier γ . A basis of the tangent space to \mathcal{C} at \mathbf{x}^* is given by the semi-orthogonal matrix $\mathbf{Z} \in \mathbb{R}^{2N \times N}$ such that*

$$\mathbf{Z} = \sum_{i=1}^N \mathbf{e}_i \mathbf{e}_i^\top \otimes \mathbf{v}_i, \quad (19)$$

where $\mathbf{v}_i = [-x_{2i}^*, x_{2i-1}^*]^\top$. Denote the **reduced Riemannian Hessian** associated with \mathbf{x}^* by

$$\mathbf{H} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} \mathbf{Z} - \text{diag}(\gamma). \quad (20)$$

The second-order necessary condition for \mathbf{x}^* to be a local minimum of (10) is $\mathbf{H} \succeq \mathbf{0}_N$. The second-order sufficient condition for \mathbf{x}^* to be a **strict** local minimum of (10) is $\mathbf{H} \succ \mathbf{0}_N$.

The proof of Lemma 2 is given in Supplementary Material - Section A.

Remark 1. *The concept of Riemannian Hessian has been well-studied in differential geometry (e.g., [23]). From (20), one can see that the first term takes into account the curvature of the objective function restricted to the unit-modulus manifold \mathcal{C} . On the other hand, the second term characterizes the curvature of the manifold \mathcal{C} . While this is an elementary*

result in differential geometry, we include the proof detail in Supplementary Material - Section B for self-containedness.

B. Algorithm Properties

The PGD algorithm can be viewed as a fixed-point iteration and hence, can be analyzed via the existing tools from fixed-point theory. We first define the convergent point of the PGD update (14) as follows.

Definition 3. *The point $\mathbf{x} \in \mathcal{C}$ is a **fixed point** of Algorithm 1 with step size $\eta > 0$ if it satisfies*

$$\mathbf{x} = \mathcal{P}_{\mathcal{C}}(\mathbf{x} - \eta \mathbf{A}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})). \quad (21)$$

If the constraint set \mathcal{C} is convex, any fixed point of Algorithm 1 is also an optimal solution of the constrained least squares problem [24]. Since the unit-modulus constraint set is non-convex, we show that any fixed point of Algorithm 1 is a stationary point of (10) as follows.

Lemma 3. *The vector \mathbf{x}^* is a fixed point of Algorithm 1 with step size $\eta > 0$ if and only if \mathbf{x}^* is a stationary point of the non-convex problem (10) and the corresponding Lagrange multiplier γ satisfies*

$$\begin{cases} \gamma_i < 1/\eta & \text{if } \mathbf{S}_i(\mathbf{x}^*) \neq \mathbf{s} \\ \gamma_i \leq 1/\eta & \text{if } \mathbf{S}_i(\mathbf{x}^*) = \mathbf{s} \end{cases} \quad \forall i = 1, \dots, N, \quad (22)$$

where \mathbf{s} is defined in (13).

The proof of this lemma is given in Supplementary Material - Section C. Lemma 3 suggests that when η is sufficiently small, all stationary points of (10) can be fixed points of Algorithm 1. As the step size η increases, fewer stationary points satisfying (22) can be fixed points of the algorithm. Next, we study the first-order Taylor expansion of the projection $\mathcal{P}_{\mathcal{C}}$ about a point in \mathcal{C} .

Proposition 1. *For any $\mathbf{x} \in \mathcal{C}$ and $\delta \in \mathbb{R}^{2N}$, we have*

$$\mathcal{P}_{\mathcal{C}}(\mathbf{x} + \delta) = \mathbf{x} + \mathbf{Z}\mathbf{Z}^\top \delta + \mathbf{q}(\delta), \quad (23)$$

where $\mathbf{Z} = \sum_{i=1}^N \mathbf{e}_i \mathbf{e}_i^\top \otimes \mathbf{v}_i$, for $\mathbf{v}_i = [-x_{2i}, x_{2i-1}]^\top$, and $\mathbf{q} : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$ satisfies $\|\mathbf{q}(\delta)\| \leq 2\|\delta\|^2$.

The proof of this proposition is given in Appendix A. It is noteworthy from Proposition 1 that the projection $\mathcal{P}_{\mathcal{C}}$ is

differentiable at any $\mathbf{x} \in \mathcal{C}$. Second, the derivative of $\mathcal{P}_{\mathcal{C}}$, given by $\mathbf{Z}\mathbf{Z}^\top$, coincides with the orthogonal projection onto the tangent space to \mathcal{C} at \mathbf{x} [25]. Third, the expansion (23) is universal, regardless of the magnitude of δ .

C. Main Result

We are now in position to state our main result on the linear convergence of PGD for UMLS.

Theorem 1. *Consider a stationary point $\mathbf{x}^* \in \mathcal{C}$ of the UMLS problem (10) with the corresponding Lagrange multiplier $\gamma \triangleq \gamma(\mathbf{x}^*) \in \mathbb{R}^N$ defined in (18) and the reduced Riemannian Hessian $\mathbf{H} \triangleq \mathbf{H}(\mathbf{x}^*) \in \mathbb{R}^{N \times N}$ defined in (20). Let $\{\mathbf{x}^{(k)}\}_{k=0}^\infty \subset \mathbb{R}^{2N}$ be the sequence generated by Algorithm 1 with a fixed step size $\eta > 0$. Assume that*

- (C1) $\mathbf{H} \succ \mathbf{0}_N$ (sufficient condition for \mathbf{x}^* being a strict local minimum),
- (C2) $\eta\gamma_i \neq 1$ for all $i = 1, \dots, N$, and
- (C3) $\rho(\mathbf{M}_\eta) < 1$ where

$$\mathbf{M}_\eta = \mathbf{I}_N - \eta \left(\mathbf{I}_N - \eta \text{diag}(\gamma) \right)^{-1} \mathbf{H}. \quad (24)$$

Then, there exists a finite constant $c_0(\mathbf{x}^*, \eta)$ (with a closed-form expression given in Lemma 9) such that for any $\mathbf{x}^{(0)} \in \mathcal{C}$ satisfying $\|\mathbf{x}^{(0)} - \mathbf{x}^*\| < c_0(\mathbf{x}^*, \eta)$, the sequence $\{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|\}_{k=0}^\infty$ converges to 0. Furthermore, if $\|\mathbf{x}^{(0)} - \mathbf{x}^*\| < \rho(\mathbf{M}_\eta)c_0(\mathbf{x}^*, \eta)$, it holds for any integer $k \geq 0$ that

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|}{\|\mathbf{x}^{(0)} - \mathbf{x}^*\|} < \left(1 - \frac{\|\mathbf{x}^{(0)} - \mathbf{x}^*\|}{\rho(\mathbf{M}_\eta)c_0(\mathbf{x}^*, \eta)} \right)^{-1} \rho^k(\mathbf{M}_\eta). \quad (25)$$

In (25), Algorithm 1 with fixed step size η is said to converge **linearly** to \mathbf{x}^* with a rate of $\rho(\mathbf{M}_\eta)$. The ball of radius $c_0(\mathbf{x}^*, \eta)$ centered at \mathbf{x}^* is called the **region of convergence**.

Theorem 1 suggests that PGD in Algorithm 1 initialized near a strict local minimum as indicated by (C1) with a proper step size η following the requirements in (C2) and (C3) converges linearly to the local minimum. The theorem establishes three key results for the linear convergence of Algorithm 1: the region of convergence, the rate of convergence, and the bound on the error through iterations. Notably, while the previous result in [1] proves the sublinear convergence to a set of stationary points of (10), our result in Theorem 1 shows the linear convergence to a strict local minimum. It is worthwhile mentioning that the linear convergence of $\{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|\}_{k=0}^\infty$ given by (25) matches with the definition of R-linear convergence in [26]-Appendix A.⁴

Note that Theorem 1 does not explicitly suggest an upper bound on η that ensures convergence and it may appear that PGD with arbitrarily large step size η still converges. However, to ensure convergence, the implicit condition on η in (C3) must hold. To provide an intuition for the step size requirement in

⁴Compared to Q-linear convergence, R-linear convergence concerns the overall rate of decrease in the error, rather than the decrease over each individual step of the algorithm. A more elaborate bound on the convergence of non-linear difference equations of the form (72) is developed in [27], in terms of the number of iterations to reach certain accuracy. In this work, we use a simpler result in Lemmas 13 and 14 (given in Supplementary Material - Section H) to demonstrate the linear convergence.

this condition, let us consider a more restrictive condition that suffices (C3):

Lemma 4. *Let $\eta > 0$ be a step size such that*

$$(C3') \quad \eta(\lambda_1(\mathbf{H}) + 2\bar{\gamma}) < 2, \text{ where } \bar{\gamma} = \max_i \gamma_i.$$

Then, Condition (C3) in Theorem 1 holds, i.e., $\rho(\mathbf{M}_\eta) < 1$.

The proof of Lemma 4 is given in Supplementary Material - Section D. When $\lambda_1(\mathbf{H}) + 2\bar{\gamma} \leq 0$, any step size $\eta > 0$ satisfies (C3') and hence, satisfies (C3). When $\lambda_1(\mathbf{H}) + 2\bar{\gamma} > 0$, (C3') suggests an upper bound on η that is sufficient but not necessary for (C3), i.e., $\eta < 2/(\lambda_1(\mathbf{H}) + 2\bar{\gamma})$. As can be seen from Table I, Condition (C3') is similar to the convergence condition in the case of unit-norm constraint.

In Theorem 1, Condition (C3) suggest a non-linear relationship between the convergence rate $\rho(\mathbf{M}_\eta)$ and the step size η . In principle, one can find the optimal step size for local linear convergence by solving the 1-D optimization

$$\begin{aligned} \eta^* &= \underset{\eta > 0}{\operatorname{argmin}} \rho(\mathbf{M}_\eta(\mathbf{x}^*)) \\ &= \underset{\eta > 0}{\operatorname{argmin}} \rho \left(\left(\mathbf{I}_N - \eta(\mathbf{I}_N - \eta \text{diag}(\gamma(\mathbf{x}^*)))^{-1} \mathbf{H}(\mathbf{x}^*) \right) \right). \end{aligned} \quad (26)$$

In the last equation, we spell out the dependence on \mathbf{x}^* to emphasize that the prior knowledge of the local minimum is critical for determining the optimal step size. In Section V, we propose two variants of PGD with adaptive step size schemes that do not require prior knowledge of \mathbf{M}_η to select the optimal step size. The proposed algorithms enjoy the fast convergence of PGD with a fixed optimal step size while remaining the same computational complexity per iteration.

D. Proof of Theorem 1

This subsection presents a proof of Theorem 1, arranging the key ideas into lemmas and deferring their proofs to the appendix. Let us begin with the claim that the strict local minimum \mathbf{x}^* in Theorem 1 is also a fixed point of PGD with the appropriate choice of the step size η .

Lemma 5. *Consider the same setting as Theorem 1. Assume that Conditions (C1)-(C3) in Theorem 1 hold. Then, \mathbf{x}^* is a fixed point of Algorithm 1 with the given step size η and its corresponding Lagrange multiplier γ satisfies $\gamma_i < 1/\eta$, for all $i = 1, \dots, N$.*

The proof of Lemma 5 is given in Supplementary Material - Section E. Next, we establish a recursion on the error vector, based on the modulus scale-invariance property and the first-order approximation of the projection in Proposition 1.

Lemma 6. *Consider the same setting as Theorem 1. Assume that Conditions (C1)-(C3) in Theorem 1 hold. Let $\mathbf{D}_\eta = (\mathbf{I}_N - \eta \text{diag}(\gamma))^{-1}$ and $\delta^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ be the error vector at the k th iteration of Algorithm 1. Then, for any integer $k \geq 0$, we have*

$$\begin{aligned} \delta^{(k+1)} &= \mathbf{Z}\mathbf{Z}^\top (\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \delta^{(k)} \\ &\quad + \mathbf{q} \left((\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \delta^{(k)} \right), \end{aligned} \quad (27)$$

where \mathbf{Z} at \mathbf{x}^* and \mathbf{q} are defined in Proposition 1.

The proof of Lemma 6 is given in Supplementary Material - Section F. Equation (27) can be viewed as an approximately linear dynamic on the error $\delta^{(k)}$. As the error becomes sufficiently small, the residual term $\mathbf{q}((\mathbf{D}_\eta \otimes \mathbf{I}_{2N})(\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A})\delta^{(k)})$ is negligible while the linear term $\mathbf{Z}\mathbf{Z}^\top(\mathbf{D}_\eta \otimes \mathbf{I}_{2N})(\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A})\delta^{(k)}$ dominates. It has been well-studied in the literature [11], [27]–[30] that the linear convergence rate of (27) is the spectral radius of the linear operator $\mathbf{Z}\mathbf{Z}^\top(\mathbf{D}_\eta \otimes \mathbf{I}_{2N})(\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A})$. However, following the argument about the structural constraint on the error vector in [30], we emphasize the fact $\delta^{(k)} = \mathcal{P}_C(\mathbf{x}^* + \delta^{(k)}) - \mathcal{P}_C(\mathbf{x}^*)$ is the difference between two points on the unit-modulus manifold and show that the error vector is dominated by the component on the tangent space to \mathcal{C} at \mathbf{x}^* .

Lemma 7. *Consider the same setting as Theorem 1. At the k th iteration of Algorithm 1, we have*

$$\delta^{(k)} = \mathbf{Z}\mathbf{Z}^\top \delta^{(k)} + \mathbf{q}(\delta^{(k)}), \quad (28)$$

where \mathbf{Z} at \mathbf{x}^* and \mathbf{q} are defined in Proposition 1.

The proof of Lemma 7 is given in Supplementary Material - Section G. Next, combining Lemmas 6 and 7, we obtain a recursion on the error vector that implicitly enforces it to lie on the tangent space to \mathcal{C} at \mathbf{x}^* as follows.

Lemma 8. *Consider the same setting as Theorem 1. Assume that Conditions (C1)–(C3) in Theorem 1 hold. Then by Lemmas 6 and 7, the error vector at the k th iteration of Algorithm 1 satisfies*

$$\delta^{(k+1)} = \mathbf{Z}\mathbf{M}_\eta \mathbf{Z}^\top \delta^{(k)} + \hat{\mathbf{q}}(\delta^{(k)}), \quad (29)$$

where \mathbf{Z} at \mathbf{x}^* is defined in Proposition 1, $\hat{\mathbf{q}}: \mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$ satisfies $\|\hat{\mathbf{q}}(\delta)\| \leq 2c_\eta(c_\eta + 1)\|\delta\|^2$, and $c_\eta = \|((\mathbf{I}_N - \eta \text{diag}(\boldsymbol{\gamma}))^{-1} \otimes \mathbf{I}_{2N})(\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A})\|_2$.

The proof of Lemma 8 is given in Appendix B. Finally, we show the convergence of $\{\delta^{(k)}\}_{k=0}^\infty$ by recognizing that (i) the spectral radius of $\mathbf{Z}\mathbf{M}_\eta \mathbf{Z}^\top$ is the same as that of \mathbf{M}_η and (ii) the recursion (29) is an approximately linear difference equation that is convergent for $\delta^{(0)}$ sufficiently close to $\mathbf{0}_{2N}$.

Lemma 9. *Consider the same setting as Theorem 1. Assume that Conditions (C1)–(C3) in Theorem 1 hold. Let us define $\bar{\gamma} = \max_i \gamma_i$, $\underline{\gamma} = \min_i \gamma_i$ and*

$$c_0(\mathbf{x}^*, \eta) = \frac{1 - \rho(\mathbf{M}_\eta) \ 1 - \eta\bar{\gamma}}{2c_\eta(c_\eta + 1) \ 1 - \eta\underline{\gamma}}, \quad (30)$$

where c_η is defined in Lemma 8. If $\|\delta^{(0)}\| < c_0(\mathbf{x}^*, \eta)$, then the sequence $\{\delta^{(k)}\}_{k=0}^\infty$ converges to $\mathbf{0}_{2N}$. Furthermore, let $c_1(\mathbf{x}^*, \eta) = \rho(\mathbf{M}_\eta)c_0(\mathbf{x}^*, \eta)$. Then, for any $\|\delta^{(0)}\| < c_1(\mathbf{x}^*, \eta)$ and integer $k \geq 0$, we have

$$\|\delta^{(k)}\| \leq \left(1 - \frac{\|\delta^{(0)}\|}{c_1(\mathbf{x}^*, \eta)}\right)^{-1} \left(\frac{1 - \eta\underline{\gamma}}{1 - \eta\bar{\gamma}}\right)^{1/2} \|\delta^{(0)}\| \rho^k(\mathbf{M}_\eta). \quad (31)$$

The proof of Lemma 9 is given in Appendix C. With this lemma, we complete our proof of Theorem 1.

Algorithm 2: Backtracking PGD (Bt-PGD)

Input: $\mathbf{x}^{(0)} \in \mathbb{R}^{2N}$, $\alpha \in (0, 1]$, $\beta \in (0, 1)$
Output: $\{\mathbf{x}^{(k)}\}_{k=0}$

- 1: $\eta_0 = 1$
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: $\mathbf{g}_k = \mathbf{A}^\top(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})$
- 4: $\eta_k = \eta_k / \beta$
- 5: **repeat**
- 6: $\eta_k = \beta \eta_k$
- 7: $\tilde{\mathbf{g}}_{\eta_k} = (\mathbf{x}^{(k)} - \mathcal{P}_C(\mathbf{x}^{(k)} - \eta_k \mathbf{g}_k)) / \eta_k$
- 8: **until** $\tilde{\mathbf{g}}_{\eta_k}^\top \mathbf{A}^\top \mathbf{A} \tilde{\mathbf{g}}_{\eta_k} \leq \frac{1}{\eta_k} \|\tilde{\mathbf{g}}_{\eta_k}\|^2$
- 9: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \eta_k \tilde{\mathbf{g}}_{\eta_k}$
- 10: $\eta_{k+1} = \eta_k / \alpha$

V. IMPLEMENTATION ASPECTS

This subsection describes two practical variants of PGD with adaptive step size that can be used when no prior knowledge of the solution is available: PGD with backtracking line search (Algorithm 2) and Nesterov’s accelerated PGD with adaptive restart (Algorithm 3).

A. Backtracking PGD (Bt-PGD)

In backtracking PGD, the step size is chosen to approximately minimize the objective function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ along the ray $\{\mathbf{x} - \eta \tilde{\mathbf{g}}_\eta \mid \eta > 0\}$, where

$$\tilde{\mathbf{g}}_\eta = \frac{1}{\eta} \left(\mathbf{x} - \mathcal{P}_C(\mathbf{x} - \eta \mathbf{A}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})) \right)$$

is the generalized gradient. To guarantee certain decrease in the objective function, we use the following backtracking condition [19]

$$f(\mathbf{x} - \eta \tilde{\mathbf{g}}_\eta) \leq f(\mathbf{x}) - \eta \tilde{\mathbf{g}}_\eta^\top \nabla f(\mathbf{x}) + \frac{\eta}{2} \|\tilde{\mathbf{g}}_\eta\|^2. \quad (32)$$

Since $f(\cdot)$ is a quadratic, it can be expanded as

$$f(\mathbf{x} - \eta \tilde{\mathbf{g}}_\eta) = f(\mathbf{x}) - \eta \tilde{\mathbf{g}}_\eta^\top \nabla f(\mathbf{x}) + \eta^2 \tilde{\mathbf{g}}_\eta^\top \nabla^2 f \tilde{\mathbf{g}}_\eta. \quad (33)$$

Substituting (33) back into the LHS of (32) and using the fact that $\nabla^2 f = \mathbf{A}^\top \mathbf{A}$, we obtain the simplified backtracking condition $\tilde{\mathbf{g}}_{\eta_k}^\top \mathbf{A}^\top \mathbf{A} \tilde{\mathbf{g}}_{\eta_k} \leq \frac{1}{\eta_k} \|\tilde{\mathbf{g}}_{\eta_k}\|^2$ as in Algorithm 2-Line 8. It is worthwhile to note that a factor of $1/\alpha$ is applied to increase the step size at the end of each iteration to encourage the algorithm to explore larger step sizes with faster convergence. We emphasize that this strategy is different from the well-known backtracking line search method in the literature (e.g., [31]), in which the step size η is reset to 1 before the backtracking line search is performed. As a result, the constant α in Algorithm 2 should not be interpreted as the fraction of the decrease in the objective function as in [31]-Algorithm 9.2.

B. Adaptive Restart Nesterov’s Accelerated PGD (ARNAPGD)

Next, we present an acceleration technique for PGD, named adaptive restart Nesterov’s accelerated projected gradient descent (ARNAPGD). In unconstrained optimization, it has

been well-known that Nesterov's accelerated gradient (NAG) [10] can dramatically improve the linear convergence rate of gradient descent (GD) for minimizing a μ -strongly convex, L -smooth function. As pointed out in [32]-Proposition 12, GD with a fixed step size $\alpha = 1/L$ has convergence rate $\rho \leq \sqrt{(L - \mu)/(L + \mu)}$, while NAG with fixed parameters $\alpha = 1/L$ and $\beta = (\sqrt{L} - \sqrt{\mu})/(\sqrt{L} + \sqrt{\mu})$ has convergence rate $\rho \leq \sqrt{1 - \sqrt{\mu/L}}$. Since NAG requires a specific choice of parameters that depends on L and μ , O'Donoghue and Candes [33] proposed a more practical variant called the Nesterov's accelerated gradient with adaptive restart (ARNAG) that recovers the same rate of convergence with no prior knowledge of function parameters. In order to adapt this idea in the context of gradient descent for unconstrained optimization to the context of projected gradient descent for constrained optimization, we modify the original ARNAG with a gradient scheme as follows. First, we utilize the generalized gradient $\tilde{\mathbf{g}}_\eta$ and the backtracking condition in (28) for PGD to determine the gradient step size (see Lines 7-10 in Algorithm 3). Second, we use the restart condition as $\tilde{\mathbf{g}}_{\eta_k}^\top(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) > 0$ (see Lines 16-17 in Algorithm 3). It is noted that in the original unconstrained optimization context, the gradient scheme restarts whenever the momentum term and the negative gradient are making an obtuse angle. In our constrained optimization context, we restart when the momentum seems to be taking us in a bad direction, as measured by the negative of the *generalized gradient* at that point. The advantage of this acceleration is it has the same computational complexity per iteration as PGD and Bt-PGD⁵ while achieving significantly faster convergence rate. Further details on ARNAPGD are provided in Algorithm 3. In the next section, we compare the performance of PGD with a fixed optimal step size, Bt-PGD, and ARNAPGD for UMLS.

VI. NUMERICAL EVALUATION

This section demonstrates the correctness of our theoretical result on the linear convergence of PGD for UMLS in Theorem 1. We show through numerical simulation that our predicted rate of convergence matches the decrease in the distance to the solution through iterations. Moreover, we illustrate the effectiveness of the two variants of PGD with adaptive step sizes proposed in Section V. For additional details, in Supplementary Material - Section I, we present a simple 2-D example of the region of convergence to demonstrate our theoretical bound in (30).

A. PGD with a Fixed Step Size

Data generation. In the following, we create a UMLS setting in which $\mathbf{x}^* \in \mathcal{C}$ satisfies

$$\begin{cases} \mathbf{A}^\top(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = (\text{diag}(\boldsymbol{\gamma}) \otimes \mathbf{I}_2)\mathbf{x}^* \\ \mathbf{H} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} \mathbf{Z} - \text{diag}(\boldsymbol{\gamma}) \succ \mathbf{0}_N \end{cases}$$

as follows. First, we generate two matrices \mathfrak{R} and \mathfrak{S} of size $M \times N$, where $M = 50$ and $N = 40$, with i.i.d normally

⁵The number of matrix-vector products in ARNAPGD is exactly the same as that in Bt-PGD.

Algorithm 3: Adaptive restart Nesterov's accelerated PGD (ARNAPGD) with gradient scheme

Input: $\mathbf{x}^{(0)} \in \mathbb{R}^{2N}$, $\alpha \in (0, 1]$, $\beta \in (0, 1)$
Output: $\{\mathbf{x}^{(k)}\}_{k=0}$

- 1: $\eta_0 = 1$
- 2: $\theta_0 = 1$
- 3: $\mathbf{y}^{(0)} = \mathbf{x}^{(0)}$
- 4: **for** $k = 0, 1, 2, \dots$ **do**
- 5: $\mathbf{g}_k = \mathbf{A}^\top(\mathbf{A}\mathbf{y}^{(k)} - \mathbf{b})$
- 6: $\eta_k = \eta_k / \beta$
- 7: **repeat**
- 8: $\eta_k = \beta \eta_k$
- 9: $\tilde{\mathbf{g}}_{\eta_k} = (\mathbf{y}^{(k)} - \mathcal{P}_{\mathcal{C}}(\mathbf{x}^{(k)} - \eta_k \mathbf{g}_k)) / \eta_k$
- 10: **until** $\tilde{\mathbf{g}}_{\eta_k}^\top \mathbf{A}^\top \mathbf{A} \tilde{\mathbf{g}}_{\eta_k} \leq \frac{1}{\eta_k} \|\tilde{\mathbf{g}}_{\eta_k}\|^2$
- 11: $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k)} - \eta_k \tilde{\mathbf{g}}_{\eta_k}$
- 12: $\theta_{k+1} = \frac{2\theta_k}{\theta_k + \sqrt{\theta_k^2 + 4}}$
- 13: $\beta_{k+1} = \theta_k(1 - \theta_k) / (\theta_k^2 + \theta_{k+1})$
- 14: $\mathbf{y}^{(k+1)} = \mathbf{x}^{(k+1)} + \beta_{k+1}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$
- 15: $\eta_{k+1} = \eta_k / \alpha$
- 16: **if** $\tilde{\mathbf{g}}_{\eta_k}^\top(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) > 0$ **then**
- 17: $\theta_{k+1} = 1$

distributed ($\mathcal{N}(0, 1)$) entries. The matrix \mathbf{A} is computed from \mathfrak{R} and \mathfrak{S} using (7). Second, we generate a random vector $\mathbf{v} \in \mathbb{R}^N$ with i.i.d normally distributed entries following $\mathcal{N}(0, 0.1^2)$ and a random vector $\mathbf{t} \in \{-1, 1\}^N$ with uniformly distributed entries. Then, we obtain \mathbf{x}^* and $\boldsymbol{\gamma}$ by setting

$$\begin{cases} \gamma_i = t_i \|\mathbf{S}_i(\mathbf{A}^\top \mathbf{v})\| \\ \mathbf{S}_i(\mathbf{x}^*) = \mathbf{S}_i(\mathbf{A}^\top \mathbf{v}) / \gamma_i \end{cases} \quad \text{for } i = 1, \dots, N.$$

Next, the matrices \mathbf{Z} and \mathbf{H} are obtained by (19) and (20), respectively. If \mathbf{H} is not PD, we re-run the foregoing generation process multiple times until $\mathbf{H} \succ \mathbf{0}_N$. This guarantees Condition (C1) in Theorem 1 is satisfied. Finally, we compute $\mathbf{b} = \mathbf{A}\mathbf{x}^* - \mathbf{v}$ and initialize $\mathbf{x}^{(0)}$ near \mathbf{x}^* by adding a random noise with i.i.d normally distributed entries following $\mathcal{N}(0, 0.001^2)$ to \mathbf{x}^* .⁶

Results. Figure 3(a) demonstrates the convergence rate $\rho(\mathbf{M}_\eta)$ (blue solid line) as a function of the step size η . Recall that $\mathbf{M}_\eta = \mathbf{I}_N - \eta(\mathbf{I}_N - \eta \text{diag}(\boldsymbol{\gamma}))^{-1} \mathbf{H}$ and hence, $\rho(\mathbf{M}_\eta)$ is a non-linear function of η . It can be seen from the plot that $\rho(\mathbf{M}_\eta)$ approaches 1 (slow convergence) when η approaches either 0 or $\eta_{\max} = 2.44$. The optimal step size that yields the fastest convergence for PGD with a fixed step size is $\eta^* = \text{argmin}_{\eta > 0} \rho(\mathbf{M}_\eta) = 2.4328$. Figure 3(b) shows the convergence of PGD with various fixed step sizes. We observe that for $\eta > \eta_{\max}$ (the overlapping red and yellow solid lines), the algorithm diverges from the designed strict local minimum \mathbf{x}^* . For step sizes less than η_{\max} , our theoretical rate (dashed lines) matches well with the empirical rate (solid lines). Moreover, PGD with the optimal step size η^* converges roughly twice as fast as one with the step size $\eta = 1/\|\mathbf{A}\|_2^2$

⁶For the purpose of demonstration, we use small additive noise to ensure the algorithms converges locally to \mathbf{x}^* .

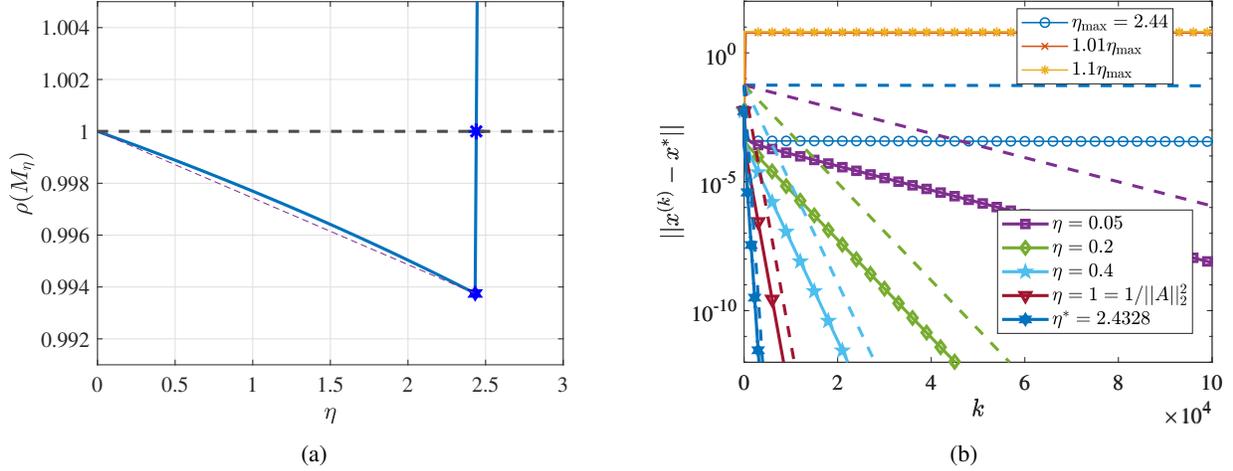


Fig. 3: Convergence of PGD with a fixed step size for UMLS. (a) Plot of the convergence rate $\rho(\mathbf{M}_\eta)$ as a function of the step size η . The black dashed line is the line $\eta = 1$, emphasizing that the local convergence is guaranteed when $\rho(\mathbf{M}_\eta) < 1$. The blue star represents the maximum step size η_{\max} such that $\rho(\mathbf{M}_{\eta_{\max}}) = 1$, while the blue hexagram represents the optimal step size is $\eta^* = \operatorname{argmin}_{\eta > 0} \rho(\mathbf{M}_\eta)$. The purple dashed line connecting the blue hexagram and the point $(0, 1)$ is included to demonstrate the non-linearity of $\rho(\mathbf{M}_\eta)$ for $\eta \in (0, \eta^*)$. (b) Plot of the distance between the current update and the local minimum as a function of the number of iterations for various fixed step sizes. Dashed lines represent the corresponding upper bounds $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|$ with exponential decay, i.e., $\rho^k(\mathbf{M}_\eta)$ up to a constant.

proposed in [1], suggesting that the latter choice, while being commonly used in the literature, is conservative.

B. Adaptive Schemes for Step Size

To illustrate the role of α in exploring larger step sizes with faster convergence while balancing the cost of backtracking steps, we plot the error through iterations $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|$ against the number of matrix-vector products, which dominates the computational complexity per iteration, in Fig. 4. The data used in this simulation is the same as in the previous section. While the smaller values of α seem to yield faster convergence (see Fig. 4(a)), they indeed require more backtracking steps at each iteration (see Fig. 4(b)). As a result, the overall computation is higher for smaller values of α . It can be seen from Fig. 4(c) that the best choice of α is $\alpha = \beta = 0.8$. In addition, we observe that the total cost of Bt-PGD is comparable to that of PGD with the optimal fixed step size. However, Bt-PGD does not use any prior knowledge about the solution \mathbf{x}^* . Fig. 4(d) shows the fluctuation in the step size η around the optimal value $\eta^* = 2.4328$. It is interesting to note that even though $\eta > \eta_{\max}$ at some iterations, the algorithm can converge to the designed local minimum \mathbf{x}^* .

Figure 5 compares the performance of four algorithms in solving the foregoing UMLS setting: PGD with a fixed step $\eta = 1/\|\mathbf{A}\|_2^2$ (used in [1]), PGD with a fixed optimal step η^* (given by (26)), Bt-PGD with the optimal choice $\alpha = \beta = 0.8$ (Algorithm 2), and ARNAPGD (Algorithm 3). The data used in this simulation is the same as in the previous section. We observe that all three algorithms proposed in this work outperform PGD with the commonly used step size $\eta = 1/\|\mathbf{A}\|_2^2$ (blue solid line). It is also highlighted that ARNAPGD (purple solid line) obtains significantly faster convergence compared

to the other algorithms while remaining similar computational complexity per iteration. The overlap between the red solid line and the yellow solid line in Fig. 5 indicates that bt-PGD recovers the optimal rate of convergence. Finally, we note that both Algorithm 2 and Algorithm 3 do not come with convergence guarantees in our setting since \mathcal{C} is non-convex. Nonetheless, on the practical side, they do not require prior knowledge of the solution and their effectiveness is depicted clearly through our numerical results.

C. Phase-only Beamforming

This section demonstrates the performance of our proposed algorithms in the phase-only beamforming example in Section II-A. Due to potential differences in magnitude between \mathbf{y} and $\mathbf{A}\mathbf{w}$, we consider a scaled version of (2)

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N, s \in \mathbb{C}} \quad & \frac{1}{2} \|\mathbf{A}\mathbf{w} - s\mathbf{y}\|^2 \\ \text{s.t.} \quad & |w_i|^2 = 1 \text{ for } i = 1, \dots, N. \end{aligned} \quad (34)$$

By differentiating the objective function w.r.t s and setting the resulting gradient to zero, we can obtain the optimal s in terms of \mathbf{w} as $s^* = \mathbf{y}^H \mathbf{A}\mathbf{w} / \|\mathbf{y}\|^2$. Substituting this s^* and simplifying yields the following alternative equivalent optimization

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N} \quad & \frac{1}{2} \mathbf{w}^H \mathbf{A}^H \left(\mathbf{I}_M - \frac{\mathbf{y}\mathbf{y}^H}{\|\mathbf{y}\|^2} \right) \mathbf{A}\mathbf{w} \\ \text{s.t.} \quad & |w_i|^2 = 1 \text{ for } i = 1, \dots, N. \end{aligned} \quad (35)$$

It is noted that our formulation is similar to the auto-scaling formulation in [1]. However, their approach considers an automatic normalization of $\mathbf{A}\mathbf{w}$ and employs an alternating minimization w.r.t. \mathbf{w} and s . On the other hand, our approach

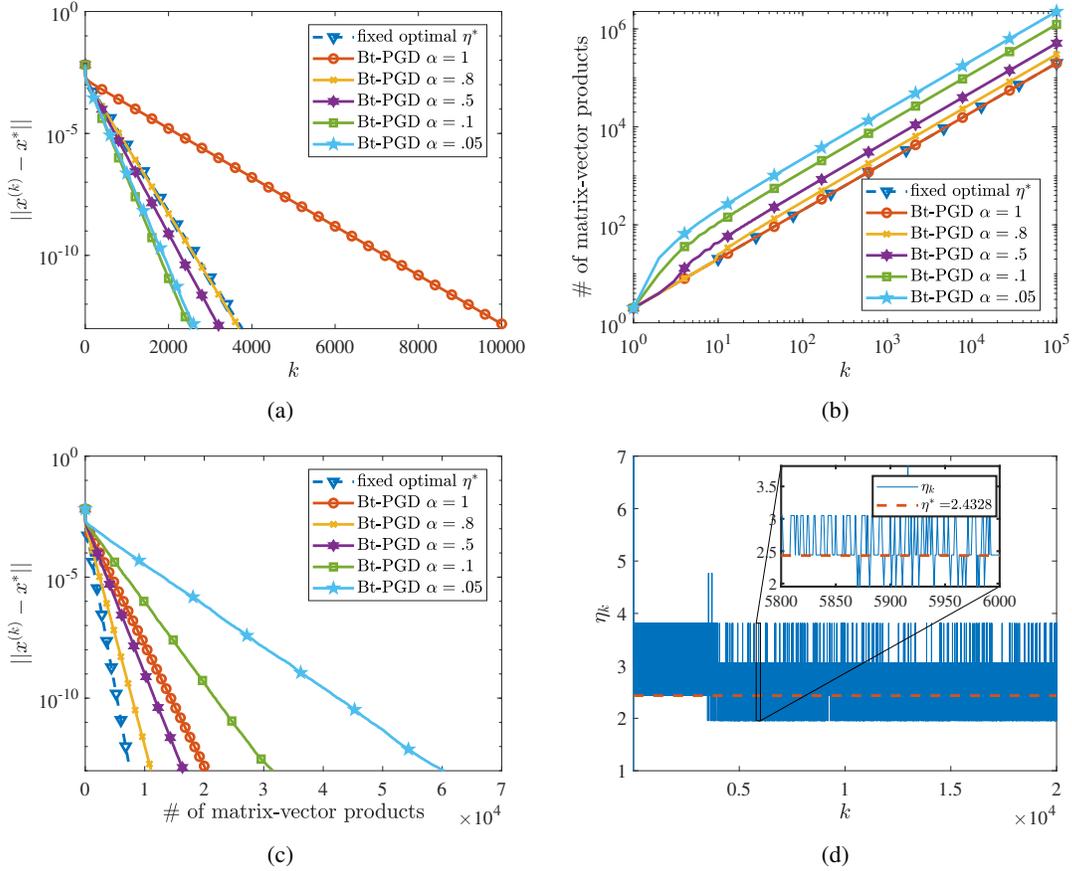


Fig. 4: Convergence of Bt-PGD with various values of α and a fixing value of $\beta = 0.8$. (a) Plot of the distance from the current update of Bt-PGD to the local minimum as a function of the number of iterations. A dashed blue line is included as an illustration of the convergence of PGD with the fixed optimal step size η^* . (b) Plot of the number of matrix-vector products used by Bt-PGD as a function of the number of iterations. (c) Plot of the distance from the current update of Bt-PGD to the local minimum as a function of the number of matrix-vector products. (d) Plot of the backtracking step size η as a function of the number of iterations for Bt-PGD with $\alpha = \beta = 0.8$. A zoom-in plot is included on top of the original plot for enhanced visualization. After a few thousand iterations, we observe that the adaptive step size η_k fluctuates around the optimal step size $\eta^* = 2.4328$ (red dashed line).

considers an optimal scaling of \mathbf{y} and in the following, we will reformulate it as a standard UMLS problem. Given any solution \mathbf{w}^* of problem (35), $e^{j\phi}\mathbf{w}^*$ is also a solution, for any angle ϕ . Thus, for the uniqueness of the solution, we assume that $w_1 = \mathbf{e}_1^H \mathbf{w} = 1$. Let $\bar{\mathbf{E}}_1 = [\mathbf{e}_2, \dots, \mathbf{e}_N]$ be the $N \times (N-1)$ matrix such that $\tilde{\mathbf{w}} \triangleq \bar{\mathbf{E}}_1^H \mathbf{w} = [w_2, \dots, w_N]^T$. Then, (35) can be reformulated as

$$\begin{aligned} \min_{\tilde{\mathbf{w}} \in \mathbb{C}^N} \quad & \frac{1}{2} \tilde{\mathbf{w}}^H \bar{\mathbf{E}}_1^H \mathbf{A}^H \left(\mathbf{I}_M - \frac{\mathbf{y}\mathbf{y}^H}{\|\mathbf{y}\|^2} \right) \mathbf{A} \bar{\mathbf{E}}_1 \tilde{\mathbf{w}} \\ & + \Re \left(\mathbf{e}_1^H \mathbf{A}^H \left(\mathbf{I}_M - \frac{\mathbf{y}\mathbf{y}^H}{\|\mathbf{y}\|^2} \right) \mathbf{A} \bar{\mathbf{E}}_1 \tilde{\mathbf{w}} \right) \\ \text{s.t.} \quad & |\tilde{w}_i|^2 = 1 \text{ for } i = 1, \dots, N. \end{aligned} \quad (36)$$

Let $\mathbf{I}_M - \mathbf{y}\mathbf{y}^H/\|\mathbf{y}\|^2 = \mathbf{Q}\mathbf{Q}^H$, for $\mathbf{Q} \in \mathbb{C}^{M \times (M-1)}$ and $\mathbf{Q}^H \mathbf{Q} = \mathbf{I}_{M-1}$, (36) is equivalent to the following constrained least squares

$$\min_{\tilde{\mathbf{w}} \in \mathbb{C}^N} \frac{1}{2} \|\tilde{\mathbf{A}}\tilde{\mathbf{w}} - \tilde{\mathbf{y}}\|^2 \text{ s.t. } |\tilde{w}_i|^2 = 1 \text{ for } i = 1, \dots, N, \quad (37)$$

where $\tilde{\mathbf{A}} = \mathbf{Q}^H \mathbf{A} \bar{\mathbf{E}}_1 \in \mathbb{C}^{(M-1) \times (N-1)}$ and $\tilde{\mathbf{y}} = -\mathbf{Q}^H \mathbf{A} \mathbf{e}_1 \in \mathbb{C}^{M-1}$. Note that both the non-scaled version (2) and the scaled version (37) of the phase-only beamforming problem fall under the UMLS setting.

We study the convergence of Algorithms 1, 2, and 3 with a ULA scenario with $N = 50$ antennas and the angle space $[-\pi/2, \pi/2]$ discretized into $M = 800$ regions. The target direction is the range $[-\pi/60, \pi/60]$, with the desired output vector \mathbf{y} satisfying $y_m = 1$ if $-\pi/60 \leq \theta_m \leq \pi/60$ and $y_m = 0$ otherwise, for $m = 1, \dots, M$. As a sanity check, we also include the variable step size gradient projection (VSGP) in the comparison. We note that this is an adaptive step size variant of PGD proposed in [2] for solving constant-modulus least squares (CMLS). While VSGP was shown to converge to a KKT point of the CMLS problem, no further analysis of the convergence speed is presented in [2]. In our implementation, we use two different values of the shrinking parameter μ in VSGP, which are 0.3 (used in [2]) and 0.8 (our tuned value).

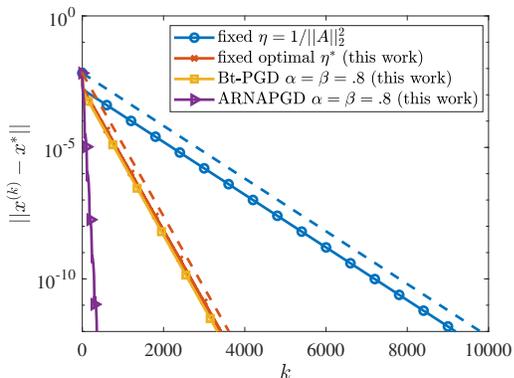


Fig. 5: Plot of the distance from the current update to the local minimum \mathbf{x}^* as a function of the number of iterations, for four algorithms: PGD with a fixed step $\eta = 1/\|\mathbf{A}\|_2^2$ [1] (blue solid line), PGD with a fixed optimal step η^* given by (26) (red solid line), Bt-PGD with $\alpha = \beta = 0.8$ (yellow solid line), and ARNAPGD (red solid line). Dashed lines represent the corresponding upper bounds on $\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|$ with exponential decay, i.e., $\rho^k(\mathbf{M}_\eta)$ up to a constant. All algorithms have the same computational complexity per iteration.

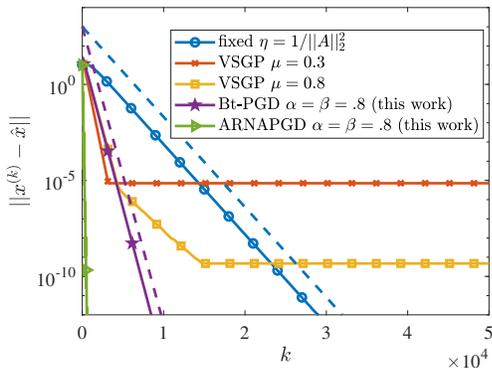


Fig. 6: Plot of the distance from the current update to the local minimum $\hat{\mathbf{x}}^7$, for five algorithms: PGD with a fixed step $\eta = 1/\|\mathbf{A}\|_2^2$ [1] (blue solid line), PGD with a variable step size - VSGP [2] with $\mu = 0.3$ (red solid line) and $\mu = 0.8$ (yellow solid line), Bt-PGD with $\alpha = \beta = 0.8$ (purple solid line), and ARNAPGD (green solid line). Dashed lines represent the corresponding upper bounds on $\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|$ with exponential decay, i.e., $\rho^k(\mathbf{M}_\eta)$ up to a constant. All algorithms share the same computational complexity per iteration.

In this simulation, since no ground truth solutions are available, we use the convergent point of ARNAPGD $\hat{\mathbf{x}}$ as an approximation of \mathbf{x}^* and measure the distance from the current update to $\hat{\mathbf{x}}$ as the error through iterations. As can be seen from Fig. 6, all algorithms exhibit linear convergence, with the fixed-step-size PGD being the slowest and ARNAPGD being the fastest. The VSGP and Bt-PGD algorithms have a similar rate of linear convergence, both approaching the

⁷Since the true optimum \mathbf{x}^* is not unknown, we use $\hat{\mathbf{x}}$ as an approximation of \mathbf{x}^* with the KKT error 10^{-12} .

optimal rate of convergence for fixed step size PGD. However, VSGP seems to stop early and converge to a different solution than other algorithms. This can be explained by the effect of the shrinking parameter μ for the step size in VSGP, making it smaller through iterations. Comparing VSGP with $\mu = 0.3$ (red solid line) and $\mu = 0.8$ (yellow solid line), we observe that the error in the former converges to around 10^{-5} while the error in the latter converges to around 10^{-10} . In contrast, our bt-PGD algorithm increases the backtracking step size by a factor of $1/\alpha$ at each iteration. Finally, it is highlighted that our upper bound in (25) predicts well the rate of convergence for both fixed-step-size PGD ($\eta = 1/\|\mathbf{A}\|_2^2$) and Bt-PGD (nearly-optimal step size).

VII. CONCLUSION AND FUTURE WORK

We introduced a novel analysis of linear convergence of projected gradient descent for the unit-modulus least squares problem. Our analysis reveals that near strict local minima, the convergence is linear as opposed to sublinear as suggested in [1]. Moreover, we identified the sufficient conditions for linear convergence and provided an exact expression of the linear convergence rate. The theoretical rate predicts accurately the asymptotic convergence of PGD for UMLS in our numerical simulation. On the practical side, we propose two variants of PGD with adaptive step sizes that obtain fast convergence without prior knowledge about the solution.

For future work, we plan to improve our bound on the region of convergence. This requires further investigation into the bounding techniques used in the proof of Theorem 1. Another potential direction is to develop the analysis for linear convergence of Bt-PGD and ARNAPGD. While convergence guarantees for backtracking line search and Nesterov's accelerated gradient have been proposed in the optimization literature [10], [31], they often involve the spectral radius that depends linearly on the step size η . The UMLS problem, on the other hand, involves the spectral radius $\rho(\mathbf{M}_\eta)$ that depends non-linearly on η . This makes it challenging for determining closed-form expressions of the optimal step size in both plain PGD and accelerated PGD.

APPENDIX A

PROOF OF PROPOSITION 1

The proof of this lemma is based on the following result for the projection onto the unit sphere [12]:

Lemma 10. (Rephrased from Lemma 5 in [12]) Let \mathbf{x} be a point on the unit sphere \mathcal{S}^{n-1} . Then, for any $\delta \in \mathbb{R}^n$, the projection onto \mathcal{S}^{n-1} satisfies

$$\mathcal{P}_{\mathcal{S}^{n-1}}(\mathbf{x} + \delta) = \mathbf{x} + (\mathbf{I} - \mathbf{x}\mathbf{x}^\top)\delta + \mathbf{q}_{\mathcal{S}^{n-1}}(\delta), \quad (38)$$

where $\|\mathbf{q}_{\mathcal{S}^{n-1}}(\delta)\| \leq 2\|\delta\|^2$.

Applying Lemma 10 to the 1-D unit circle \mathcal{S}^1 (corresponding to the case $n = 2$), we have, for each $i = 1, \dots, N$,

$$\begin{aligned} \mathcal{S}_i(\mathcal{P}_{\mathcal{C}}(\mathbf{x} + \delta)) &= \mathcal{P}_{\mathcal{S}^1}(\mathcal{S}_i(\mathbf{x} + \delta)) = \mathcal{P}_{\mathcal{S}^1}(\mathcal{S}_i(\mathbf{x}) + \mathcal{S}_i(\delta)) \\ &= \mathcal{S}_i(\mathbf{x}) + (\mathbf{I}_2 - \mathcal{S}_i(\mathbf{x})\mathcal{S}_i(\mathbf{x})^\top)\mathcal{S}_i(\delta) + \mathbf{q}_{\mathcal{S}^1}(\mathcal{S}_i(\delta)) \\ &= \mathcal{S}_i(\mathbf{x}) + \mathbf{v}_i\mathbf{v}_i^\top\mathcal{S}_i(\delta) + \mathbf{q}_{\mathcal{S}^1}(\mathcal{S}_i(\delta)), \end{aligned}$$

where $\mathbf{v}_i = [-x_{2i}, x_{2i-1}]^\top$. Using the property of the 2-selection operator in (8), we further have

$$\begin{aligned} \mathcal{P}_C(\mathbf{x} + \boldsymbol{\delta}) &= \sum_{i=1}^N \mathbf{e}_i \otimes \mathbf{S}_i(\mathcal{P}_C(\mathbf{x} + \boldsymbol{\delta})) \\ &= \sum_{i=1}^N \mathbf{e}_i \otimes \left(\mathbf{S}_i(\mathbf{x}) + \mathbf{v}_i \mathbf{v}_i^\top \mathbf{S}_i(\boldsymbol{\delta}) + \mathbf{q}_{S^1}(\mathbf{S}_i(\boldsymbol{\delta})) \right) \\ &= \sum_{i=1}^N \mathbf{e}_i \otimes \mathbf{S}_i(\mathbf{x}) + \sum_{i=1}^N (\mathbf{e}_i \otimes \mathbf{v}_i \mathbf{v}_i^\top) \mathbf{S}_i(\boldsymbol{\delta}) + \mathbf{q}(\boldsymbol{\delta}) \\ &= \mathbf{x} + \mathbf{Z}\mathbf{Z}^\top \boldsymbol{\delta} + \mathbf{q}(\boldsymbol{\delta}), \end{aligned} \quad (39)$$

where $\mathbf{q}(\boldsymbol{\delta})$ satisfies $\mathbf{S}_i(\mathbf{q}(\boldsymbol{\delta})) = \mathbf{q}_{S^1}(\mathbf{S}_i(\boldsymbol{\delta}))$ and

$$\begin{aligned} \|\mathbf{q}(\boldsymbol{\delta})\|^2 &= \sum_{i=1}^N \|\mathbf{S}_i(\mathbf{q}(\boldsymbol{\delta}))\|^2 = \sum_{i=1}^N \|\mathbf{q}_{S^1}(\mathbf{S}_i(\boldsymbol{\delta}))\|^2 \\ &\leq \sum_{i=1}^N (2\|\mathbf{S}_i(\boldsymbol{\delta})\|^2)^2 \leq \left(\sum_{i=1}^N 2\|\mathbf{S}_i(\boldsymbol{\delta})\|^2 \right)^2 \\ &= 4 \left(\sum_{i=1}^N (\delta_{2i-1}^2 + \delta_{2i}^2) \right)^2 = 4 \left(\sum_{j=1}^{2N} \delta_j^2 \right)^2 = 4\|\boldsymbol{\delta}\|^4. \end{aligned}$$

This completes our proof of the lemma.

APPENDIX B PROOF OF LEMMA 8

Substituting (28) back into the first term on the RHS of (27), we have

$$\begin{aligned} \boldsymbol{\delta}^{(k+1)} &= \mathbf{Z}\mathbf{Z}^\top (\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \mathbf{Z}\mathbf{Z}^\top \boldsymbol{\delta}^{(k)} \\ &\quad + \mathbf{Z}\mathbf{Z}^\top (\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \mathbf{q}(\boldsymbol{\delta}^{(k)}) \\ &\quad + \mathbf{q}((\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \boldsymbol{\delta}^{(k)}). \end{aligned} \quad (40)$$

From Lemma 11 in Supplementary Material - Section H and the fact that $\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}_N$, we can represent (40) as

$$\begin{aligned} \boldsymbol{\delta}^{(k+1)} &= \mathbf{Z}\mathbf{D}_\eta \mathbf{Z}^\top (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \mathbf{Z}\mathbf{Z}^\top \boldsymbol{\delta}^{(k)} + \hat{\mathbf{q}}(\boldsymbol{\delta}^{(k)}) \\ &= \mathbf{Z}\mathbf{D}_\eta (\mathbf{I}_N - \eta \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} \mathbf{Z}) \mathbf{Z}^\top \boldsymbol{\delta}^{(k)} + \hat{\mathbf{q}}(\boldsymbol{\delta}^{(k)}), \end{aligned} \quad (41)$$

where $\hat{\mathbf{q}}(\boldsymbol{\delta}) = \mathbf{Z}\mathbf{Z}^\top (\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \mathbf{q}(\boldsymbol{\delta}) + \mathbf{q}((\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \boldsymbol{\delta})$. Recall that $\mathbf{H} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} \mathbf{Z} - \text{diag}(\boldsymbol{\gamma})$. Thus, (41) is equivalent to

$$\begin{aligned} \boldsymbol{\delta}^{(k+1)} &= \mathbf{Z}\mathbf{D}_\eta (\mathbf{I}_N - \eta \text{diag}(\boldsymbol{\gamma}) - \mathbf{H}) \mathbf{Z}^\top \boldsymbol{\delta}^{(k)} + \hat{\mathbf{q}}(\boldsymbol{\delta}^{(k)}) \\ &= \mathbf{Z}(\mathbf{I}_N - \eta \mathbf{D}_\eta \mathbf{H}) \mathbf{Z}^\top \boldsymbol{\delta}^{(k)} + \hat{\mathbf{q}}(\boldsymbol{\delta}^{(k)}). \end{aligned} \quad (42)$$

Substituting $\mathbf{M}_\eta = \mathbf{I}_N - \eta \mathbf{D}_\eta \mathbf{H}$ into (42) yields (29).

To bound the norm of $\hat{\mathbf{q}}(\boldsymbol{\delta}^{(k)})$, we use the triangle inequality and the product norm inequality as follows

$$\begin{aligned} \|\hat{\mathbf{q}}(\boldsymbol{\delta})\| &\leq \|\mathbf{Z}\mathbf{Z}^\top (\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \mathbf{q}(\boldsymbol{\delta})\| \\ &\quad + \|\mathbf{q}((\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \boldsymbol{\delta})\| \\ &\leq \|\mathbf{Z}\mathbf{Z}^\top\|_2 \|(\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A})\|_2 \|\mathbf{q}(\boldsymbol{\delta})\| \\ &\quad + \|\mathbf{q}((\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \boldsymbol{\delta})\|. \end{aligned}$$

Since $\|\mathbf{q}(\boldsymbol{\delta})\| \leq 2\|\boldsymbol{\delta}\|^2$ (see Proposition 1) and $c_\eta = \|(\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A})\|_2$, we further obtain

$$\begin{aligned} \|\hat{\mathbf{q}}(\boldsymbol{\delta})\| &\leq \|\mathbf{Z}\mathbf{Z}^\top\|_2 2c_\eta 2\|\boldsymbol{\delta}\|^2 + 2\|(\mathbf{D}_\eta \otimes \mathbf{I}_2) (\mathbf{I}_{2N} - \eta \mathbf{A}^\top \mathbf{A}) \boldsymbol{\delta}\|^2 \\ &\leq 2c_\eta \|\mathbf{Z}\mathbf{Z}^\top\|_2 \|\boldsymbol{\delta}\|^2 + 2c_\eta^2 \|\boldsymbol{\delta}\|^2 \\ &\leq 2c_\eta \|\boldsymbol{\delta}\|^2 + 2c_\eta^2 \|\boldsymbol{\delta}\|^2, \end{aligned}$$

where the last inequality stems from $\|\mathbf{Z}\mathbf{Z}^\top\|_2 \leq 1$ since $\mathbf{Z}\mathbf{Z}^\top$ is an orthogonal projection matrix. This completes our proof of the lemma.

APPENDIX C PROOF OF LEMMA 9

The proof in this section relies on Lemma 8 in the manuscript and Lemmas 12 and 13 in Supplementary Material - Section H. Let $\tilde{\boldsymbol{\delta}}^{(k)} = (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \boldsymbol{\delta}^{(k)}$. Left-multiplying both sides of (29) with $(\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2)$, we have

$$\begin{aligned} \tilde{\boldsymbol{\delta}}^{(k+1)} &= (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \mathbf{Z} \mathbf{M}_\eta \mathbf{Z}^\top \boldsymbol{\delta}^{(k)} + (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \hat{\mathbf{q}}(\boldsymbol{\delta}^{(k)}) \\ &= (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \mathbf{Z} \mathbf{M}_\eta \mathbf{Z}^\top (\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)} \\ &\quad + (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \hat{\mathbf{q}}((\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)}). \end{aligned} \quad (43)$$

Using Lemma 11 in Supplementary Material - Section H and substituting $\mathbf{M}_\eta = \mathbf{I}_N - \eta \mathbf{D}_\eta^{-1} \mathbf{H}$ into the RHS of (43) yield

$$\begin{aligned} \tilde{\boldsymbol{\delta}}^{(k+1)} &= \mathbf{Z} \mathbf{D}_\eta^{-1/2} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1} \mathbf{H}) \mathbf{D}_\eta^{1/2} \mathbf{Z}^\top \tilde{\boldsymbol{\delta}}^{(k)} + \tilde{\mathbf{q}}(\tilde{\boldsymbol{\delta}}^{(k)}) \\ &= \mathbf{Z} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top \tilde{\boldsymbol{\delta}}^{(k)} + \tilde{\mathbf{q}}(\tilde{\boldsymbol{\delta}}^{(k)}), \end{aligned} \quad (44)$$

where $\tilde{\mathbf{q}}(\tilde{\boldsymbol{\delta}}^{(k)}) = (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \hat{\mathbf{q}}((\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)})$ satisfies

$$\begin{aligned} \|\tilde{\mathbf{q}}(\tilde{\boldsymbol{\delta}}^{(k)})\| &\leq \|\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2\|_2 \|\hat{\mathbf{q}}((\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)})\| \\ &= \|\mathbf{D}_\eta^{-1/2}\|_2 \|\hat{\mathbf{q}}((\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)})\| \\ &\leq \|\mathbf{D}_\eta^{-1/2}\|_2 2c_\eta (c_\eta + 1) \|(\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)}\|^2 \\ &\leq 2c_\eta (c_\eta + 1) \|\mathbf{D}_\eta^{-1/2}\|_2 \|\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2\|_2^2 \|\tilde{\boldsymbol{\delta}}^{(k)}\|^2 \\ &\leq 2c_\eta (c_\eta + 1) \|\mathbf{D}_\eta^{-1/2}\|_2 \|\mathbf{D}_\eta^{1/2}\|_2^2 \|\tilde{\boldsymbol{\delta}}^{(k)}\|^2 \\ &= 2c_\eta (c_\eta + 1) (1 - \eta\gamma)^{1/2} (1 - \eta\bar{\gamma})^{-1} \|\tilde{\boldsymbol{\delta}}^{(k)}\|^2, \end{aligned}$$

where the last equality stems from $\|\mathbf{D}_\eta^{-1/2}\|_2 = (1 - \eta\gamma)^{1/2}$ and $\|\mathbf{D}_\eta^{1/2}\|_2 = (1 - \eta\bar{\gamma})^{-1/2}$. Let $q = 2c_\eta (c_\eta + 1) (1 - \eta\gamma)^{1/2} (1 - \eta\bar{\gamma})^{-1}$. Taking the norm of both sides of (44) and then using the triangle inequality on the RHS, we obtain

$$\begin{aligned} \|\tilde{\boldsymbol{\delta}}^{(k+1)}\| &= \|\mathbf{Z} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top \tilde{\boldsymbol{\delta}}^{(k)} + \tilde{\mathbf{q}}(\tilde{\boldsymbol{\delta}}^{(k)})\| \\ &\leq \|\mathbf{Z} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top \tilde{\boldsymbol{\delta}}^{(k)}\| + \|\tilde{\mathbf{q}}(\tilde{\boldsymbol{\delta}}^{(k)})\|. \end{aligned}$$

Since $\mathbf{Z} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top$ is symmetric, its spectral norm equals to its spectral radius. The last inequality can be rewritten as

$$\begin{aligned} \|\tilde{\boldsymbol{\delta}}^{(k+1)}\| &\leq \rho(\mathbf{Z} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top) \|\tilde{\boldsymbol{\delta}}^{(k)}\| \\ &\quad + q \|\tilde{\boldsymbol{\delta}}^{(k)}\|^2. \end{aligned} \quad (45)$$

Moreover, it can be seen from (44) that $\mathbf{Z} (\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top = (\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \mathbf{Z} \mathbf{M}_\eta \mathbf{Z}^\top (\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2)^{-1}$, which in turns implies the two matrices $\mathbf{Z} (\mathbf{I}_N -$

$\eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2} \mathbf{Z}^\top$ and $\mathbf{Z} \mathbf{M}_\eta \mathbf{Z}^\top$ are similar and have the same spectral radius. In particular, we have

$$\begin{aligned} \rho(\mathbf{Z}(\mathbf{I}_N - \eta \mathbf{D}_\eta^{-1/2} \mathbf{H} \mathbf{D}_\eta^{-1/2}) \mathbf{Z}^\top) &= \rho(\mathbf{Z} \mathbf{M}_\eta \mathbf{Z}^\top) \\ &= \rho(\mathbf{M}_\eta), \end{aligned} \quad (46)$$

where the second equality stems from Lemma 12 in Supplementary Material - Section H. Substituting (46) into the RHS of (45), we obtain

$$\|\tilde{\boldsymbol{\delta}}^{(k+1)}\| \leq \rho(\mathbf{M}_\eta) \|\tilde{\boldsymbol{\delta}}^{(k)}\| + q \|\tilde{\boldsymbol{\delta}}^{(k)}\|^2.$$

Applying Lemma 14 with $b_k = \|\tilde{\boldsymbol{\delta}}^{(k)}\|$, $\rho = \rho(\mathbf{M}_\eta)$, and $c = (1 - \eta\gamma)^{1/2} c_1(\mathbf{x}^*, \eta)$, it holds that if $\|\tilde{\boldsymbol{\delta}}^{(0)}\| < c$, then

$$\|\tilde{\boldsymbol{\delta}}^{(k)}\| \leq \left(1 - \frac{\|\tilde{\boldsymbol{\delta}}^{(0)}\|}{c}\right)^{-1} \|\tilde{\boldsymbol{\delta}}^{(0)}\| \rho^k(\mathbf{M}_\eta). \quad (47)$$

Recall that $\boldsymbol{\delta}^{(k)} = (\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)}$. On the one hand, the LHS of (47) can be lower-bounded as $\|\tilde{\boldsymbol{\delta}}^{(k)}\| \geq (1 - \eta\gamma)^{1/2} \|\boldsymbol{\delta}^{(k)}\|$, due to the fact that

$$\begin{aligned} \|\boldsymbol{\delta}^{(k)}\| &= \|(\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2) \tilde{\boldsymbol{\delta}}^{(k)}\| \leq \|\mathbf{D}_\eta^{1/2} \otimes \mathbf{I}_2\|_2 \|\tilde{\boldsymbol{\delta}}^{(k)}\| \\ &= \|\mathbf{D}_\eta^{1/2}\|_2 \|\tilde{\boldsymbol{\delta}}^{(k)}\| = (1 - \eta\gamma)^{-1/2} \|\tilde{\boldsymbol{\delta}}^{(k)}\|. \end{aligned}$$

On the other hand, the RHS of (47) can be upper-bounded as follows. Since

$$\begin{aligned} \|\tilde{\boldsymbol{\delta}}^{(0)}\| &= \|(\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2) \boldsymbol{\delta}^{(0)}\| \leq \|\mathbf{D}_\eta^{-1/2} \otimes \mathbf{I}_2\|_2 \|\boldsymbol{\delta}^{(0)}\| \\ &= \|\mathbf{D}_\eta^{-1/2}\|_2 \|\boldsymbol{\delta}^{(0)}\| = (1 - \eta\gamma)^{1/2} \|\boldsymbol{\delta}^{(0)}\|, \end{aligned} \quad (48)$$

we have

$$\begin{aligned} &\left(1 - \frac{\|\tilde{\boldsymbol{\delta}}^{(0)}\|}{c}\right)^{-1} \|\tilde{\boldsymbol{\delta}}^{(0)}\| \rho^k(\mathbf{M}_\eta) \\ &\leq \left(1 - \frac{(1 - \eta\gamma)^{1/2} \|\boldsymbol{\delta}^{(0)}\|}{c}\right)^{-1} (1 - \eta\gamma)^{1/2} \|\boldsymbol{\delta}^{(0)}\| \rho^k(\mathbf{M}_\eta) \\ &= \left(1 - \frac{\|\boldsymbol{\delta}^{(0)}\|}{c_1(\mathbf{x}^*, \eta)}\right)^{-1} (1 - \eta\gamma)^{1/2} \|\boldsymbol{\delta}^{(0)}\| \rho^k(\mathbf{M}_\eta). \end{aligned} \quad (49)$$

From the lower bound $(1 - \eta\gamma)^{1/2} \|\boldsymbol{\delta}^{(k)}\|$ and the upper bound in (49), we obtain (31). Finally, the region of convergence $\|\boldsymbol{\delta}^{(0)}\| < c_1(\mathbf{x}^*, \eta)$ is sufficient to guarantee that $\|\tilde{\boldsymbol{\delta}}^{(0)}\| < c = (1 - \eta\gamma)^{1/2} c_1(\mathbf{x}^*, \eta)$ due to (48). This completes our proof of the lemma.

REFERENCES

- [1] J. Tranter, N. D. Sidiropoulos, X. Fu, and A. Swami, "Fast unit-modulus least squares with applications in beamforming," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2875–2887, 2017.
- [2] M. Zhang, J. Li, S. Zhu, and X. Chen, "Fast and simple gradient projection algorithms for phase-only beamforming," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10620–10632, 2021.
- [3] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, Max-Cut and complex semidefinite programming," *Math. Program.*, vol. 149, no. 1, pp. 47–81, 2015.
- [4] P. Thompson, "Adaptation by direct phase-shift adjustment in narrow-band adaptive antenna systems," *IEEE Trans. Antennas Propag.*, vol. 24, no. 5, pp. 756–760, 1976.
- [5] M. Soltanalian and P. Stoica, "Designing unimodular codes via quadratic optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1221–1234, 2014.
- [6] X. Fu, F. K. Chan, W.-K. Ma, and H. So, "A complex-valued semidefinite relaxation approach for two-dimensional source localization using distance measurements and imperfect receiver positions," in *IEEE Int. Conf. Signal Process.*, vol. 2. IEEE, 2012, pp. 1491–1494.

- [7] S. Huang, S. Wang, R. Wang, M. Wen, and K. Huang, "Reconfigurable intelligent surface assisted mobile edge computing with heterogeneous learning tasks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 2, pp. 369–382, 2021.
- [8] S. Zhang and Y. Huang, "Complex quadratic optimization and semidefinite programming," *SIAM J. Optim.*, vol. 16, no. 3, pp. 871–890, 2006.
- [9] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010.
- [10] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer New York, NY, 2003, vol. 87.
- [11] T. Vu, R. Raich, and X. Fu, "On convergence of projected gradient descent for minimizing a large-scale quadratic over the unit sphere," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.* IEEE, 2019, pp. 1–6.
- [12] T. Vu and R. Raich, "On asymptotic linear convergence of projected gradient descent for constrained least squares," *IEEE Trans. Signal Process.*, vol. 4, pp. 4061–4076, 2022.
- [13] C. D. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2000, vol. 71.
- [14] T. H. Ismail and Z. M. Hamici, "Array pattern synthesis using digital phase control by quantized particle swarm optimization," *IEEE Trans. Antennas Propag.*, vol. 58, no. 6, pp. 2142–2145, 2010.
- [15] C.-J. Lu, W.-X. Sheng, Y.-B. Han, and X.-F. Ma, "A novel adaptive phase-only beamforming algorithm based on semidefinite relaxation," in *IEEE Int. Symp. Phased Array Syst. Technol.* IEEE, 2013, pp. 617–621.
- [16] A. De Maio, S. De Nicola, Y. Huang, S. Zhang, and A. Farina, "Code design to optimize radar detection performance under accuracy and similarity constraints," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5618–5629, 2008.
- [17] F. Vasiliev and A. Y. Ivanitskiy, *In-depth analysis of linear programming*. Springer Dordrecht, 2013.
- [18] D. G. Luenberger, Y. Ye *et al.*, *Linear and nonlinear programming*. Springer New York, NY, 1984, vol. 2.
- [19] A. Beck, *First-order methods in optimization*. SIAM, 2017.
- [20] D. C. Sorensen, "Minimization of a large-scale quadratic function subject to a spherical constraint," *SIAM J. Optim.*, vol. 7, no. 1, pp. 141–161, 1997.
- [21] W. W. Hager, "Minimizing a quadratic over a sphere," *SIAM J. Optim.*, vol. 12, no. 1, pp. 188–208, 2001.
- [22] A. Beck and Y. Vaisbourd, "Globally solving the trust region subproblem using simple first-order methods," *SIAM J. Optim.*, vol. 28, no. 3, pp. 1951–1967, 2018.
- [23] J. M. Lee, *Introduction to Riemannian manifolds*. Springer Cham, 2018.
- [24] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, pp. 334–334, 1997.
- [25] A. S. Lewis and J. Malick, "Alternating projections on manifolds," *Math. Oper. Res.*, vol. 33, no. 1, pp. 216–234, 2008.
- [26] N. Jorge and J. W. Stephen, *Numerical optimization*. Spinger, 2006.
- [27] T. Vu and R. Raich, "A closed-form bound on the asymptotic linear convergence of iterative methods via fixed point analysis," *Optim. Lett.*, vol. 17, no. 3, pp. 643–656, 2022.
- [28] R. Bellman, *Stability theory of differential equations*. Courier Corporation New York, 2008.
- [29] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964.
- [30] T. Vu and R. Raich, "Exact linear convergence rate analysis for low-rank symmetric matrix completion via gradient descent," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* IEEE, 2021, pp. 3240–3244.
- [31] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [32] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM J. Optim.*, vol. 26, no. 1, pp. 57–95, 2016.
- [33] B. O'donoghue and E. Candes, "Adaptive restart for accelerated gradient schemes," *Found. Comput. Math.*, vol. 15, no. 3, pp. 715–732, 2015.
- [34] B. Wang and F. Zhang, "Some inequalities for the eigenvalues of the product of positive semidefinite hermitian matrices," *Linear Algebra Appl.*, vol. 160, pp. 113–118, 1992.
- [35] T. Vu and R. Raich, "Adaptive step size momentum method for deconvolution," in *Proc. IEEE Stat. Signal Process. Workshop*, 2018, pp. 438–442.



Trung Vu received the B.S. degree in Computer Science from Hanoi University of Science and Technology, Hanoi, Vietnam, in 2014. He received the Ph.D. degree in Computer Science at the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon, USA, in 2022. He is currently a postdoctoral research associate at the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, Maryland, USA. His research interests include optimization methods, independent component analysis, and matrix factorization with applications in machine learning and signal processing. Dr. Vu received a Best Student Paper Award at IEEE MLSP 2019.

dependent component analysis, and matrix factorization with applications in machine learning and signal processing. Dr. Vu received a Best Student Paper Award at IEEE MLSP 2019.



Raviv Raich (S'98–M'04–SM'17) received the B.Sc. and M.Sc. degrees from Tel-Aviv University, Tel-Aviv, Israel, in 1994 and 1998, respectively, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 2004, all in electrical engineering. Between 1999 and 2000, he was a Researcher with the Communications Team, Industrial Research, Ltd., Wellington, New Zealand. From 2004 to 2007, he was a Postdoctoral Fellow with the University of Michigan, Ann Arbor. He has been with the School of Electrical Engineering

and Computer Science, Oregon State University, Corvallis, as an Assistant Professor (2007–2013), Associate Professor (2013–2023), and is currently a Professor (2023–present). His co-authored papers at the IEEE MLSP workshop 2015, ICASSP 2018, and IEEE MLSP 2019 workshop received best paper awards. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2011 to 2014. He served as a member (2011–2016) and then chair (2017–2018) of the Machine Learning for Signal Processing (MLSP) Technical Committee (TC) of the IEEE Signal Processing Society. Since 2019, he is a member of the Signal Processing Theory and Methods (SPTM), TC of the IEEE Signal Processing Society. His research interests include: probabilistic modeling and optimization in signal processing and machine learning.



Xiao Fu (Senior Member, IEEE) received the Ph.D. degree in Electronic Engineering from The Chinese University of Hong Kong (CUHK), Shatin, N.T., Hong Kong, in 2014. He was a Postdoctoral Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, from 2014 to 2017. Since 2017, he has been with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA, where he is currently an Associate Professor. His research interests

include the broad area of signal processing and machine learning.

Dr. Fu received a Best Student Paper Award at ICASSP 2014. His coauthored papers received Best Student Paper Awards from IEEE CAMSAP 2015 and IEEE MLSP 2019, respectively. He received the 2022 IEEE Signal Processing Society (SPS) Best Paper Award and the 2022 IEEE SPS Donald G. Fink Overview Paper Award. He was a recipient of the Outstanding Postdoctoral Scholar Award at University of Minnesota in 2016, the Engelbrecht Early Career Faculty Award from the College of Engineering at Oregon State University in 2023, and the National Science Foundation (NSF) CAREER Award in 2022. He serves as a member of the IEEE SPS Sensor Array and Multichannel Technical Committee (SAM-TC) and the Signal Processing Theory and Methods Technical Committee (SPTM-TC). Since 2023, he has been the Chair of the IEEE SPS Oregon Chapter. He is currently an Editor of SIGNAL PROCESSING and an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING.