Distributed Banach–Picard Iteration for Locally Contractive Maps
Francisco Andrade, Mário A. T. Figueiredo, Fellow, IEEE, and João Xavier

Abstract—The Banach–Picard iteration is widely used to find fixed points of locally contractive (LC) maps. This article extends the Banach–Picard iteration to distributed settings; specifically, we assume the map of which the fixed point is sought to be the average of individual (not necessarily LC) maps held by a set of agents linked by a communication network. An additional difficulty is that the LC map is not assumed to come from an underlying optimization problem, which prevents exploiting strong global properties, such as convexity or Lipschitzianity. Yet, we propose a distributed algorithm and prove its convergence, in fact showing that it maintains the linear rate of the standard Banach–Picard iteration for the average LC map. As another contribution, our proof imports tools from perturbation theory of linear operators, which, to the best of our knowledge, are scarcely exploited in the theory of distributed computation.

Index Terms—Banach–Picard iteration, consensus, distributed computation, fixed points, perturbation theory (PT) of linear operators.

I. INTRODUCTION

Inferring a desired quantity \( x^* \) from data can often be naturally expressed as a fixed point equation \( H(x^*) = x^* \), the map \( H \) relating the data to the desired quantity. Although in some rare cases, this fixed point equation can be solved in a closed form, more often than not, \( x^* \) has to be numerically approximated using, e.g., the so-called Banach–Picard iteration

\[
x_{k+1} = H(x_k).
\]

For a recent comprehensive review of the fixed-point strategy to inference problems, see [1].

If the entire data, thus the operator \( H \), are available to some agent, that agent can perform the Banach–Picard iteration. In contrast, in the so-called distributed scenario, the data are acquired by spatially dispersed agents who only have access to local data. In such distributed setups, no single agent possesses the full dataset, hence no single agent can compute the map \( H \). Instead, each agent holds a local portion of the data and can communicate only with a subset of the other agents (its neighbors). Nevertheless, the goal remains that of finding a fixed point of \( H \), under the constraints of this distributed configuration: Each agent can only engage in private/local computation and in communication with its neighbors.

A. Problem Statement

We consider a network of \( N \) agents, where the interconnection structure is represented by an undirected and connected graph: The nodes correspond to the agents and an edge between two agents indicates they can communicate (are neighbors). Each agent \( n \in \{1, \ldots, N\} \) holds an operator \( H_n : \mathbb{R}^d \rightarrow \mathbb{R}^d \), and the goal is to compute a fixed point of the average operator

\[
H = \frac{1}{N} \sum_{n=1}^{N} H_n.
\]

Crucially, each agent \( n \) is restricted to performing computations involving \( H_n \) and communicating with its neighbors.

Our only assumption about \( H \) (which may not hold for each \( H_n \)) is that it has an attractor, i.e., a fixed point \( x^* \), such that \( H \) is continuously differentiable in a neighborhood of \( x^* \) and

\[
\rho(J_H(x^*)) < 1
\]

where \( \rho(J_H(x^*)) \) is the spectral radius of the Jacobian of \( H \) at \( x^* \). In other words, \( H \) is locally contractive (LC) in a neighborhood of \( x^* \).

Application of Ostrowski’s Theorem (see the Appendix) to the Banach–Picard iteration (1) ensures that if the sequence \( x_k \) gets sufficiently close to \( x^* \), then it not only converges to \( x^* \), but does so at a linear rate, i.e., it exhibits local linear convergence.

B. Contributions

Our main contribution is a distributed algorithm that, as the corresponding centralized Banach–Picard iteration (1), has local linear convergence at attractors of \( H \).

Although assuming a relatively weak set of conditions—essentially only local linear convergence of the centralized Banach–Picard iteration—and no global structure (e.g., Lipschitzianity or coercivity), we propose a distributed algorithm and prove that it inherits the local linear convergence of its centralized counterpart.

Even though the assumptions are rather weak, they nevertheless suffice to encapsulate relevant algorithms, namely some instances of the expectation maximization (EM) [2], [3] algorithm and the one proposed in [4] for principal component analysis (PCA). A forthcoming article is devoted to the application of the algorithmic framework herein proposed to obtain distributed versions of those algorithms, with local linear convergence guarantees [5].

As an additional contribution, we mention the proof technique, which, as far as we know, depts from the standard proof techniques used in distributed optimization. Specifically, we employ tools from the perturbation theory (PT) of linear operators [6], which, to the best

0018-9286 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
of our knowledge, are scarcely exploited in the context of distributed computation. Arguably, there are proof techniques that resemble a “perturbative argument” on the eigenvalues of a matrix (e.g., [7, Prop. 2.8] and [8, Th. 2]). However, those techniques bypass the subtle issue of the differentiability of the eigenvalues, simply using the formula for the derivative of the determinant. In contrast, the theorem from PT of linear operators that we use simultaneously handles the differentiability issue and simplifies the computation of the derivative.

C. Remarks

Our setup departs from standard ones in two main aspects. First, it encompasses problems that are not naturally expressed as optimization problems. This last notion should be understood with a grain of salt, since a fixed point of $H$ minimizes $\|H(x) - x\|^2$; however, in many cases, there is a more “natural” objective function than this one. For example, if the Jacobian $J_H$ is symmetric in an open, convex set, then there exists a function $f$ such that $H = \nabla f$ [9, Th. 1.3.1], and the Banach–Picard iteration can be seen as a method to find a stationary point of $f(x) - \frac{1}{2}\|x\|^2$.

Second, the notion of attractor herein used is purely local. By assuming (3), we consider only local guarantees. Many optimization problems benefit from global properties, such as Lipschitzianity or strong convexity. Such properties, however, are absent from many relevant algorithms, such as EM, for which only local guarantees can be given.

D. Related Work

In this section, we review relevant related work in distributed computation, highlighting how our contributions differ from that other work.

A setup that closely resembles ours is considered in [10] and [3]; in fact, the problems therein addressed are, respectively, distributed PCA and distributed EM. As shown in the upcoming article [5], our setup encapsulates the problems addressed in [10] and [3]. However, the algorithm in [3] uses a diminishing step size, which, unlike our algorithm, results in a sacrifice of the convergence rate of the centralized EM. The algorithm in [10] is recovered by using our approach to build a distributed version of the algorithm in [4]. Moreover, our work has at least two advantages over that of [10]: We provide a proof of local linear convergence (which [10] does not) and our setup is not restricted to the algorithm in [4].

The works [11]–[13] share a similarity with ours by addressing the distributed computation of fixed points. However, the setups therein considered have much more structure than ours: Lipschitzianity and quasi-nonexpansiveness [12], nonexpansiveness [13], and paracontractiveness [11]. Those are global properties that are absent in algorithms such as EM or the algorithm in [4] for PCA.

A large body of work on distributed optimization has been produced in the last decade; see [14]–[29] for convex optimization, where the last reference considers a stochastic variant, and [30]–[32], as examples of distributed nonconvex optimization. All the algorithms in those works can definitely be seen as distributed algorithms for finding fixed points. However, as their setups stem from optimization, they further assume conditions, such as coercivity, Lipschitzianity, or strong convexity. In our work, none of these properties are assumed, and only a basic local assumption is made.

E. Organization of This Article

This introductory section will conclude with a brief paragraph describing the adopted notation. Section II describes the proposed distributed algorithm. The theorems guaranteeing local linear convergence are presented and proved in Section III. Section IV provides some intuitive insight into the proposed algorithm. Finally, Section V concludes this article, which also points at ongoing and future work. The Appendix contains three theorems that are instrumental in this article.

F. Notation

The set of real $n$ dimensional vectors with positive components is denoted by $\mathbb{R}^n_{>0}$. Matrices and vectors are denoted by upper and lower case letters, respectively. The spectral radius of a matrix $M$ is denoted by $\rho(M)$. Given a map $H$ and a function $f$, $J_H(x)$ and $J^T_f(x)$ denote, respectively, the Jacobian of $H$ and the Hessian of $f$ at $x$. Given a vector $v$, $v_0$ denotes its 0th component; given a matrix $M$, $M_d$ denotes the element on the $d$th line and $d$th column; $M^T$ is its transpose. The $d$-dimensional identity matrix is denoted by $I_d$, and $1_d$ is the $d$-dimensional vector of ones. The Kronecker product is denoted by $\otimes$. The letter $i$ is denotes the imaginary unit ($i^2 = -1$). The fixed point set of a map $H$ is denoted by $\text{Fix}(H)$. Given a matrix $L$, its Moore–Penrose (pseudo)inverse is denoted as $L^+$. The gradient of a function $f$ with respect to vector $w$ is denoted by $\nabla_w f$. If a matrix $M$ is negative (positive) definite, this is denoted by $M < 0$ ($M > 0$). Whenever convenient, we will denote a vector with two stacked blocks, $[v^T, u^T]^T$, simply as $(v, u)$.

II. DISTRIBUTED ALGORITHM

Let $R$ be the map on $\mathbb{R}^{dN}$ defined, for $z = [z_1^T, \ldots, z_N^T]^T$ with $z_j \in \mathbb{R}^d$, by

$$R(z) = \left[ (H_1(z_1) - z_1)^T, \ldots, (H_N(z_N) - z_N)^T \right]^T$$

and let $W = \hat{W} \otimes I_d$, where $\hat{W}$ is the Metropolis weight matrix associated to the agents’ communication graph [33]. The distributed algorithm we propose is presented in Algorithm 1, where $\alpha \in \mathbb{R}_{>0}$.

This algorithm, inspired by EXTRA (see [17] and Section IV in the following), is a particular instance of the parametric family of distributed algorithms given by

$$z^0 \in \mathbb{R}^{dN}$$
$$z^1 = z^0 + \alpha R(z^0) - \eta L z^0$$
$$z^{k+2} = (2I - \eta L)z^{k+1} - (I + 2\beta L - \eta L)z^k + \alpha R(z^{k+1}) - R(z^k)$$

(4)

together with the following assumptions on the $dN \times dN$ matrix $L$: a) $L$ is symmetric and positive semidefinite; b) $\rho(L) < 2$; c) $\ker(L) = \{z \in \mathbb{R}^{dN} : z_1 = \cdots = z_N\}$; and

Algorithm 1: Distributed Banach–Picard iteration.

1: Initialization:
$$z^0 \in \mathbb{R}^{dN}$$
$$z^1 = Wz^0 + \alpha R(z^0)$$

2: Update:
$$z^{k+2} = (I + W)z^{k+1} - \frac{I + W}{2} z^k + \alpha R(z^{k+1} - R(z^k)).$$

1276 IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 68, NO. 2, FEBRUARY 2023
III. CONVERGENCE ANALYSIS

A. Auxiliary Maps

To study the convergence of (4), consequently of Algorithm 1, we introduce two auxiliary maps, whose connection with Algorithm 1 is explained in Section III-B. Let \( \tilde{U} \) be a matrix with columns forming an orthonormal basis of range (L) and let \( F : \mathbb{R}^{dN} \times \mathbb{R}^{dN} \rightarrow \mathbb{R}^{dN} \times \mathbb{R}^{dN} \) and \( \tilde{F} : \mathbb{R}^{dN} \times \mathbb{R}^{d(N-1)} \rightarrow \mathbb{R}^{dN} \times \mathbb{R}^{d(N-1)} \) be the maps defined as, respectively,

\[
F \left( \begin{bmatrix} z \\ w \end{bmatrix} \right) = \left[ \begin{array}{c} z + \alpha R(z) + \beta L \frac{1}{2} w - \eta Lz \\ w - \beta L \frac{1}{2} z \end{array} \right]
\]

and

\[
\tilde{F} \left( \begin{bmatrix} \tilde{z} \\ \tilde{w} \end{bmatrix} \right) = \left[ \begin{array}{c} \tilde{z} + \alpha R(\tilde{z}) + \beta L \frac{1}{2} \tilde{U} \tilde{w} - \eta L\tilde{z} \\ \tilde{w} - \beta L \frac{1}{2} \tilde{L} \tilde{z} \end{array} \right].
\]

Section III-C shows that \( \psi \) preserves condition (3), in the sense that if \( x^* \) is an attractor of \( H \), then \( \psi(x^*) \) is an attractor of \( \tilde{F} \).

B. Connection Between \( F \) and (4)

The presence of matrix \( L^\frac{1}{2} \) prevents the map \( F \) defined in (5), and which defines the iteration (7), from having a distributed implementation (whereas products by \( L \) only require each node to communicate with its neighbors, the same is not true with \( L^\frac{1}{2} \), given that \( L^\frac{1}{2} \) need not be compatible with the graph topology). However, as shown in the following, eliminating the second variable yields (4) and, for the particular choices of \( \eta = 1, \beta = \frac{1}{2}, \) and \( I = W \), Algorithm 1. Consider two consecutive \( z \)-updates, i.e.,

\[
\begin{align*}
z^{k+2} &= z^{k+1} + \alpha R(z^{k+1}) + \beta L^\frac{1}{2} w^{k+1} - \eta Lz^{k+1} \\
&= z^k + \alpha R(z^k) + \beta L^\frac{1}{2} w^k - \eta Lz^k.
\end{align*}
\]

Consider their difference

\[
z^{k+2} = 2z^{k+1} - z^k + \beta L^\frac{1}{2} (w^{k+1} - w^k) - \eta L(z^{k+1} - z^k) + \alpha (R(z^{k+1}) - R(z^k)).
\]

Note that \( w^{k+1} - w^k = -\beta L^\frac{1}{2} z^k \) and observe that its substitution in the \( z^{k+2} \)-update results in the elimination of \( w^{k+1} \) and \( w^k \) and the disappearance of \( L \), i.e.,

\[
z^{k+2} = (2I - \eta L)z^{k+1} - (I + \beta^2 L - \eta L)z^k + \alpha (R(z^{k+1}) - R(z^k)).
\]

C. Convergence Theorems

The following two theorems establish the convergence results. The first is a result about the map \( \tilde{F} \) and the second about the map \( F \).

**Theorem 1:** Let \( \lambda_1, \ldots, \lambda_{dN} \) be the eigenvalues of \( L \) and choose \( \eta > 0 \) and \( \beta > 0 \) such that the two complex roots \( \gamma_1(\lambda_s, \eta, \beta) \) and \( \gamma_2(\lambda_s, \eta, \beta) \) of all the \( dN \) polynomial equations

\[
x^2 + \lambda_s x + \lambda_s \eta^2 + \lambda_s \beta^2 = 0
\]

satisfy the following conditions:

1) \( |1 + \gamma_1(\lambda_s, \eta, \beta)| \leq 1 \)
2) \( |1 + \gamma_2(\lambda_s, \eta, \beta)| \leq |1 + \lambda_s| \), if and only if \( \lambda_s = 0 \) for any \( s = 1, \ldots, dN \) and \( j = 1, 2 \). Then, there exists \( \alpha^* \) such that, for \( 0 < \alpha < \alpha^* \)

\[
\rho(J, \psi(x^*)) < 1.
\]

**Remark 1:** Note that the choices \( \eta = 1, \beta = \frac{1}{2} \) (which lead to Algorithm 1) satisfy the conditions of Theorem 1. In fact, since \( \lambda_s < 2 \)

\[
\gamma_j(\lambda_s, 1, \frac{1}{2}) \in \left\{ -\lambda_s \pm \frac{\sqrt{\lambda_s^2 - 4\lambda_s}}{2} \right\}
\]

for \( j = 1, 2, \) and hence,

\[
|1 + \gamma_j(\lambda_s, \eta, \beta)|^2 \leq (1 - \lambda_s^2) \frac{1 + \lambda_s}{2} + \frac{2 \lambda_s - \lambda_s^2}{4} = 1 - \frac{\lambda_s}{2}.
\]

The fact that \( 0 \leq \lambda_s < 2 \) is enough to yield conditions 1) and 2) of the theorem.

In order to state the next theorem, consider the matrix

\[
\tilde{U} = \frac{1}{\sqrt{N}} \mathbf{1}_N \otimes I_d
\]

the columns of which form an orthonormal basis of \( \text{ker}(L) \).
Theorem 2: Let $\alpha, \eta$, and $\beta$ be such that $\rho(J_F(\psi(x^*)) < 1$ (the existence of such a choice is ensured by Theorem 1). Let $(z^k, w^k)$, for $k = 0, 1, \ldots$, be a trajectory of the iteration (7), initialized at $(z^0, w^0) \in \mathbb{R}^{dN} \times \mathbb{R}^{dN}$. Then, there exists a neighborhood $V$ of $\psi(x^*)$ such that if $(z^k, \hat{U}^T w^k) \in V$ for some $k$, then this trajectory converges to

$$
\left[ \begin{array}{c}
\hat{U}^T w^0 - \frac{1}{2} \beta (L^2) \hat{U} \\end{array} \right]
$$

with at least linear rate.

D. Proof of Theorem 1

We will show that we can trap the eigenvalues of $J_F(\psi(x^*))$ in the open ball in $\mathbb{C}$ of center 0 and radius 1 by adjusting $\alpha, \beta,$ and $\eta$. The Jacobian of $F$ at $\psi(x^*)$ is given by

$$
J_F(\psi(x^*)) = I + \hat{A}(\eta, \beta) + B(\alpha)
$$

where

$$
\hat{A}(\eta, \beta) = \begin{bmatrix}
-\eta L & \beta L^2 \hat{U} \\
-\beta \hat{U}^T L^2 & 0
\end{bmatrix}
$$

and

$$
B(\alpha) = \begin{bmatrix}
\alpha \mathcal{J}_R(1_N \otimes x^*) & 0 \\
0 & 0
\end{bmatrix}
$$

We divide the proof in two lemmas. Lemma 1 shows that a choice of $\eta > 0$ and $\beta > 0$, according to the conditions of the Theorem, implies that the matrix $I + \hat{A}(\eta, \beta)$ has a semisimple eigenvalue equal to 1, and that the remaining eigenvalues have magnitude less than 1. Lemma 2 deals with $\alpha$ and shows, by appealing to a result on the perturbation of semisimple eigenvalues, that this parameter can be tuned to yield the conclusion of Theorem 1.

Lemma 1: Let $\eta > 0$ and $\beta > 0$ be chosen according to the conditions of Theorem 1. Then

i) 1 is a semisimple eigenvalue of $I + \hat{A}(\eta, \beta)$ with multiplicity $d$ and corresponding eigenspace given by $\ker(L) \times \{0\}$;

ii) all the remaining eigenvalues have magnitude less than 1.

Proof: We start by analyzing the eigenvalues of $\hat{A}(\eta, \beta)$; toward that goal, consider the eigenvalues of

$$
A(\eta, \beta) = \begin{bmatrix}
-\eta L & \beta L^2 U \\
-\beta U^T L^2 & 0
\end{bmatrix}
$$

where $U = [\hat{U}, \hat{U}]$ ($\hat{U}$ was defined in (8) and $\hat{U}$ in the first paragraph of Section III-A). Since

$$
A(\eta, \beta) = \begin{bmatrix}
\hat{A}(\eta, \beta) & 0 \\
0 & 0
\end{bmatrix}
$$

the nonzero eigenvalues of $\hat{A}(\eta, \beta)$ coincide with the nonzero eigenvalues of $A(\eta, \beta)$.

Let $V$ be an orthogonal matrix such that $V^T LV = \Lambda$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_{dN})$, and consider the following unitary similarity:

$$
\begin{bmatrix}
V^T & 0 \\
0 & V^T U
\end{bmatrix} \begin{bmatrix}
A(\eta, \beta) & V0U^T \\
0 & U^T \end{bmatrix} = \begin{bmatrix}
-\eta \Lambda & \beta \Lambda^2 \\
-\beta \Lambda^2 & 0
\end{bmatrix}
$$

(9)

The eigenvalues are preserved by unitary similarity and, hence, the eigenvalues of (9) coincide with those of $A(\eta, \beta)$.

Let $\xi$ be an eigenvalue of (9) and $(u, v) \neq (0, 0)$ be an associated eigenvector. There must exist an $s$ such that $u_s \neq 0$ or $v_s \neq 0$, which implies that $\xi$ is an eigenvalue of the $2 \times 2$ matrix

$$
\begin{bmatrix}
-\eta \lambda_s \beta \sqrt{\lambda_s} & \beta \sqrt{\lambda_s} \\
-\beta \sqrt{\lambda_s} & 0
\end{bmatrix}
$$

thus, $\xi$ is a solution of the second-degree equation

$$
x^2 + \eta \lambda_s x + \lambda_s \beta^2 = 0.
$$

(10)

Note that the converse also holds, meaning that any root of (10) gives rise to an eigenvalue of (9). This completely characterizes the eigenvalues of (9).

Now if $\eta > 0$ and $\beta > 0$ are chosen according to the conditions of Theorem 1, then, from the derivation abovementioned

$$
\rho \left( I + \hat{A}(\eta, \beta) \right) \leq 1.
$$

Moreover, if $\lambda$ is an eigenvalue of $I + \hat{A}(\eta, \beta)$ associated to the eigenvector $(u, v)$, such that $|\lambda| = 1$, it follows, from condition 2) of Theorem 1, that $\lambda = 1$ and, therefore, $(u, v) \in \ker(\hat{A}(\eta, \beta))$. Additionally, observe that

a) $\ker(\hat{A}(\eta, \beta)) = \ker(L) \times \{0\}$ (the columns of $\hat{U}$ form a basis for range($L$));

b) the orthogonal of $\ker(\hat{A}(\eta, \beta))$ is equal to range($L$) $\times \mathbb{R}^{d(N-1)}$;

c) both are invariant under $\hat{A}(\eta, \beta)$.

From these observations, we conclude that 1 is a semisimple eigenvalue of $I + \hat{A}(\eta, \beta)$ with multiplicity $d$ and corresponding eigenspace given by $\ker(L) \times \{0\}$, and that all the other eigenvalues have magnitude less than 1.

Lemma 2: Let $\eta > 0$ and $\beta > 0$ be chosen according to the conditions of Theorem 1. Then, there exists $\alpha^*$ such that, for $0 < \alpha < \alpha^*$

$$
\rho \left( J_F(\psi(x^*)) \right) < 1.
$$

Proof: Let $C(\alpha) = I + \hat{A}(\eta, \beta) + B(\alpha)$, which can be seen as a continuous perturbation of the matrix $C(0) = I + \hat{A}(\eta, \beta)$ (recall that $B(0) = 0$). We will show that choosing a sufficiently small $\alpha$ traps the eigenvalues of $C(\alpha)$ inside the ball in $\mathbb{C}$ of center 0 and radius 1.

Theorem 4 (included in the Appendix, for convenience; see also [6]) implies that there are $2dN - d$ continuous and complex-valued functions $\lambda_1(\alpha), \ldots, \lambda_{2dN-d}(\alpha)$, such that $\{\lambda_s(\alpha) : s = 1, \ldots, 2dN - d\}$ is the set of eigenvalues of $C(\alpha)$. By reordering, if necessary, we may assume that, for $s = 1, \ldots, d, \lambda_s(0) = 1$ (recall, from Lemma 1, that 1 is a semisimple eigenvalue of $C(0)$). With this choice of order, it follows that $|\lambda_s(\alpha)| < 1$, for $d + 1 \leq s \leq 2dN - d$.

By continuity, there exists $\alpha_1 > 0$ ensuring that $|\lambda_s(\alpha)| < 1$, for all $0 \leq \alpha < \alpha_1$ and for $d + 1 \leq s \leq 2dN - d$.

Theorem 5 (included in the Appendix, for convenience; see also [34]) implies that $\lambda_1(\alpha), \ldots, \lambda_{dN}(\alpha)$ are, not only continuous, but also differentiable at 0. Moreover, their derivatives at 0 are among the eigenvalues of

$$
[\hat{U}^T \left( \begin{array}{c}
0 \\
0
\end{array} \right) \left( \frac{dC}{d\alpha} \right)_{\alpha=0} \left( \begin{array}{c}
0 \\
0
\end{array} \right) \hat{U}^T \right] = \hat{U}^T J_F(1_N \otimes x^*) \hat{U}.
$$

We observe that $\hat{U}^T J_F(1_N \otimes x^*) \hat{U} = J^H(x^*) - I$.

Condition (3), i.e., $\rho(J^H(x^*) < 1$, has not yet entered the game and it is here that it plays a crucial role. In fact, observe that condition (3) implies that the eigenvalues of $J^H(x^*) - I$ have negative real part.

A geometrical argument now settles the question: Each $\lambda_s(\alpha)$, for $s = 1, \ldots, d$, can be viewed as a continuous curve in $\mathbb{R}^2$ and can, therefore, be written as $\lambda_s(\alpha) = \left( \lambda_s^{(1)}(\alpha), \lambda_s^{(2)}(\alpha) \right)$. So far, we proved that these curves satisfy

$$
\lambda_s(0) = \left( \begin{array}{c}
1 \\
0
\end{array} \right)
$$

and

$$
\frac{d\lambda_s^{(1)}(\alpha)}{d\alpha} \bigg|_{\alpha=0} < 0.
$$

By observing that

$$
\frac{d\|\lambda_s(\alpha)\|^2}{d\alpha} \bigg|_{\alpha=0} = 2 \frac{d\lambda_s^{(1)}(\alpha)}{d\alpha} \bigg|_{\alpha=0} < 0
$$

and

$$
\|\lambda_s(0)\|^2 = 1,
$$
we conclude that there must exist an \( \alpha_2 > 0 \) such that
\[
\| \lambda_s(\alpha) \|^2 < 1
\]
for \( s = 1, \ldots, d \) and for \( 0 < \alpha < \alpha_2 \). Finally, choosing \( \alpha^* = \min\{\alpha_1, \alpha_2\} \), it follows that, for \( 0 < \alpha < \alpha^* \),
\[
\rho(C(\alpha)) < 1.
\]

**E. Proof of Theorem 2**

For any vector \( w \in \mathbb{R}^{dN} \), let \( \tilde{w} := \tilde{U} \tilde{T} w \) and \( \hat{w} := \hat{U} \tilde{T} w \). Observe that \( w = \tilde{w} + \hat{w} \) is the orthogonal decomposition of \( w \) with components in \( \text{ker}(L) \) and \( \text{range}(L) \).

Consider the sequence \( (z^k, w^k), k = 0, 1, \ldots \), produced by the iteration (7), with a given initialization \( (z^0, w^0) \). We denote \( \tilde{w}^k = \hat{w}^0 \), for all \( k \), and \( z^k \) only depends on \( \tilde{w}^k \). From these two facts, it follows that the sequence produced by the iteration
\[
\begin{bmatrix}
  u^{k+1} \\
  v^{k+1}
\end{bmatrix}
= F
\begin{bmatrix}
  u^k \\
  v^k
\end{bmatrix}
\tag{11}
\]
with initialization \( u^0 = z^0 \) and \( v^0 = \tilde{w}^0 \) must satisfy
\[
\begin{align*}
  z^k &= u^k \\
  w^k &= \tilde{w}^0 + v^k,
\end{align*}
\]
for all \( k \). Moreover, for all \( k, v^k = \tilde{w}^k \).

Said differently, the observations abovementioned show that to understand the trajectories followed by \( (z^k, w^k) \), given an arbitrary initialization \( (z^0, w^0) \), we can just study the trajectories initialized in \( \mathbb{R}^{dN} \times \text{range}(L) \). Moreover, the second component of a trajectory initialized in \( \mathbb{R}^{dN} \times \text{range}(L) \) stays in \( \text{range}(L) \).

Furthermore, \( \tilde{U} \) establishes an isomorphism between \( \mathbb{R}^{dN} \times \text{range}(L) \), with inverse \( \hat{U} \tilde{T} \). With this in mind, let \( (z^0, w^0) \in \mathbb{R}^{dN} \times \text{range}(L) \) and observe that
\[
\begin{align*}
  z^{k+1} &= z^k + \alpha R(z^k) + \beta L z^k - \eta L z^k \\
  \tilde{U} \tilde{T} w^{k+1} &= \tilde{U} \tilde{T} w^k - \beta \tilde{U} \tilde{T} z^k.
\end{align*}
\]

Defining \( \tilde{w}^{k+1} = \tilde{U} \tilde{T} w^k \), we obtain
\[
\begin{align*}
  z^{k+1} &= z^k + \alpha R(z^k) + \beta \tilde{U} \tilde{T} \tilde{w}^k - \eta L z^k \\
  \tilde{w}^{k+1} &= \tilde{w}^k - \beta \tilde{U} \tilde{T} z^k.
\end{align*}
\]

We conclude that
\[
\begin{bmatrix}
  z^k \\
  w^k
\end{bmatrix}
= \begin{bmatrix}
  z^k \\
  \tilde{w}^k
\end{bmatrix}
\]
and
\[
\begin{bmatrix}
  z^{k+1} \\
  \tilde{w}^{k+1}
\end{bmatrix}
= \tilde{F}
\begin{bmatrix}
  z^k \\
  w^k
\end{bmatrix}.
\]

These observations, combined with Theorem 1 and Ostrowski’s Theorem (see the Appendix), are enough to conclude the proof of Theorem 2.

**IV. Intuition for (4)**

To provide some intuitive insight into the rationale that leads to the Banach–Picard iteration (7), and, after the elimination of the second variable, to (4), suppose for a moment that the maps \( H_n \) in (2) had the form \( H_n = I - \nabla f_n \) for some functions \( f_n : \mathbb{R}^d \to \mathbb{R} \), and note that, in this case, the Banach–Picard iteration with \( H \) reduces to the familiar gradient descent with unit step size for finding a stationary point of
\[
f = \frac{1}{N} \sum_{i=1}^{N} f_n.
\]

Moreover, condition (3), in this case, reads
\[
\rho(J_H(x^*)) = \rho(I - \nabla^2 f(x^*)) < 1
\]
which implies that \( J_H^2(x^*) > 0 \), showing that, in this case, \( f \) is locally strictly convex, thus \( x^* \) is a local minimum of \( f \).

A common approach to pursue a distributed algorithm to maximize \( f \) is to formulate the problem as (see, e.g., [25], [28], and [27])
\[
\begin{align*}
  \text{minimize} & \quad \sum_{n=1}^{N} f_n(z_n) \\
  \text{subject to} & \quad \beta L^2 z = 0
\end{align*}
\]
where \( w \) is the vector or Lagrange multipliers and \( \eta > 0 \) (see [36]). The \textit{augmented Lagrangian method} consists in alternating between minimizing \( \mathcal{L}(z, w) \) with respect to \( z \) and taking a gradient ascent step with respect to \( w \), while keeping the other variable fixed. If instead of exact minimization, the \( z \)-step is itself a gradient ascent step, we have the Arrow–Hurwitz–Uzawa method (see [28])
\[
\begin{align*}
  z^{k+1} &= z^k - \alpha \nabla z \mathcal{L}(z^k, w^k) \\
  w^{k+1} &= w^k + \alpha \nabla w \mathcal{L}(z^k, w^k) = w^k + \alpha \beta L^2 z^k.
\end{align*}
\]

Setting \( \alpha \eta = \eta \) and \( \alpha \beta = \beta \), (4) recovers (4) for the case where \( H_n = I - \nabla f_n \). What this shows is that (4) applied to this particular case, can be interpreted as the Arrow–Hurwitz–Uzawa method to find a saddle point of the augmented Lagrangian.

As a final observation, note that, if we again let \( H_n = I - \nabla f_n \), then Algorithm 1 is nothing but the algorithm in [17] for the concave case.

**V. Conclusion and Future Work**

In this article, we proposed an algorithm for the distributed computation of attractors of average maps. The conditions of the average map considered were rather minimal, yet sufficiently strong to encapsulate relevant algorithms, such as the EM algorithm or the algorithm in [4] for PCA, as it is demonstrated in detail and experimentally assessed in an upcoming article [5]. Regardless of the minimality of the set of conditions, we were, nevertheless, able to provide local guarantees of linear convergence by employing a proof technique based on PT for linear operators, which, to the best of our knowledge, had not been used in distributed optimization. In an upcoming article [5], we verify the attractor condition for the EM algorithm and the algorithm in [4] for PCA, and therefore, we propose a distributed EM algorithm and a distributed algorithm for PCA.

**APPENDIX**

**Auxiliary Results**

This appendix contains three theorems that are instrumental in the proofs presented in Sections III-D and III-E.

---

1There is a slight abuse of notation here: We are using \( \tilde{w}^k \) and \( \hat{w}^k \) as \( \tilde{U} \tilde{T} w^k \) and \( \hat{U} \tilde{T} w^k \), respectively.
Theorem 3 (Ostrowski’s theorem; for a proof, see [37]): Suppose $G : \mathcal{D} \subseteq \mathbb{R}^d \to \mathbb{R}^d$ has a fixed point $x^*$ in the interior of $D$. Suppose as well that $G$ is $C^1$ in a neighborhood of $x^*$ and that

$$\rho(J_G(x^*)) < 1.$$ 

Then, there exists a norm $\| \cdot \|$ a ball $B = \{ x \in \mathbb{R}^d : \| x - x^* \| < \delta \}$, and a number $0 < \sigma < 1$ such that, for all $x \in B$

$$\| G(x) - x^* \| \leq \sigma \| x - x^* \|.$$ 

Theorem 4 (For a proof, see [6]): Let $G(x)$ be an unordered $N$-tuple of complex numbers depending continuously on a real variable $x$ in a (closed or open) interval $I$. Then, there exists $N$ single-valued continuous functions, $\mu_n(x)$, $n = 1, \ldots , N$, the values of which constitute the $N$-tuple $G(x)$, for each $x \in I$.

Theorem 5 (For a proof, see [34]): Let $A$ and $B$ be $n \times n$ square matrices and consider the parameteric family $C(\alpha) = A + \alpha B$. Let $\lambda$ be a semisimple eigenvalue of $A$ with multiplicity $p$ (geometric and algebraic multiplicities coincide), and suppose that there is an orthogonal $n \times n$ matrix $U = [U, \hat{U}]$, where $U$ and $\hat{U}$ have, respectively, dimensions $n \times p$ and $n \times (n - p)$, such that

$$U^T AU = \begin{bmatrix} I_p & 0 \\ 0 & A \end{bmatrix}$$

and where $\lambda$ is not an eigenvalue of $\hat{A}$. Let $\mu_1(\alpha), \ldots , \mu_p(\alpha)$ be the $p$ eigenvalue curves of $C(\alpha)$ that satisfy $\mu_j(0) = \lambda$, for $j = 1, \ldots , p$ (the existence of these curves is guaranteed by Theorem 4). Then, each $\mu_j$ is differentiable at 0 and the derivatives of $\mu_1(\alpha), \ldots , \mu_p(\alpha)$ at 0 are among the eigenvalues of $U^T \hat{B}U$.

References

[1] P. Combettes and J.-C. Pesquet, “Fixed point strategies in data science,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3878–3905, 2021.

[2] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Hoboken, NJ, USA: Wiley, 2007.

[3] S. S. Pereira, R. López-Valcarce, and A. Pages-Zamora, “Parameter estimation in wireless sensor networks with faulty transducers: A distributed EM approach,” *Signal Process.*, vol. 144, pp. 226–237, 2018.

[4] T. D. Sanger, “Optimal unsupervised learning in a single layer linear feedforward neural network,” *Neural Netw.*, vol. 2, no. 6, pp. 459–473, 1989.

[5] F. Andrade, M. Figueiredo, and J. Xavier, “Distributed Banach–Picard iteration: Application to distributed EM and distributed PCA,” 2021, arXiv:2106.10665.

[6] T. Kato, *Perturbation Theory for Linear Operators*. Berlin, Germany: Springer, 1966.

[7] D. Jakovetić, B. Bajović, and N. Li, “Accelerated distributed Nesterov gradient descent,” *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, Jun. 2020.

[8] F. Mansoori and E. Wei, “A general framework of exact primal-dual first-order algorithms for distributed optimization,” in *Proc. 58th IEEE Conf. Decis. Control*, 2019, pp. 6386–6391.

[9] D. Jakovetić, B. Bajović, J. Xavier, and J. M. F. Moura, “Primal–dual methods for large-scale and distributed convex optimization and data analytics,” *Proc. IEEE*, vol. 108, no. 11, pp. 1923–1938, Nov. 2020.

[10] T. Kato, *Perturbation Theory for Linear Operators*. Berlin, Germany: Springer, 1966.

[11] F. Mansoori and E. Wei, “A general framework of exact primal-dual first-order algorithms for distributed optimization,” in *Proc. 58th IEEE Conf. Decis. Control*, 2019, pp. 6386–6391.

[12] D. Jakovetić, B. Bajović, J. Xavier, and J. M. F. Moura, “Primal–dual methods for large-scale and distributed convex optimization and data analytics,” *Proc. IEEE*, vol. 108, no. 11, pp. 1923–1938, Nov. 2020.

[13] A. Fallah, M. Gurbuzbalaban, A. Ozdaglar, U. Simsekli, and L. Zhu, “Robust distributed accelerated stochastic gradient methods for multi-agent networks,” 2019, arXiv:1910.08701.

[14] P. D. Lorenzo and G. Scutari, “NEXT: In-network nonconvex optimization,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[15] T. Tatarenko and B. Touri, “Non-convex distributed optimization,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3744–3757, Aug. 2017.

[16] S. Vlaski and A. H. Sayed, “Distributed learning in non-convex environments—Part I: Agreement at a linear rate,” *IEEE Trans. Signal Process.*, vol. 69, pp. 1242–1256, 2021.

[17] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[18] P. Lancaster, “On eigenvalues of matrices dependent on a parameter,” *Numerische Mathematik*, vol. 6, no. 1, pp. 377–387, 1964.

[19] A. Mohktari and A. Ribeiro, “DSA: Decentralized double stochastic averaging gradient algorithm,” *J. Mach. Learn. Res.*, vol. 17, pp. 1–35, 2016.

[20] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. New York, NY, USA: Acad. Press, 2014.

[21] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. Philadelphia, PA, USA: SIAM, 2000.