
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Jung, Alexander

A Fixed-Point of View on Gradient Methods for Big Data

Published in:
Frontiers in Applied Mathematics and Statistics

DOI:
[10.3389/fams.2017.00018](https://doi.org/10.3389/fams.2017.00018)

Published: 01/01/2017

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Jung, A. (2017). A Fixed-Point of View on Gradient Methods for Big Data. *Frontiers in Applied Mathematics and Statistics*, 3, 1-11. <https://doi.org/10.3389/fams.2017.00018>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.



A Fixed-Point of View on Gradient Methods for Big Data

Alexander Jung*

Department of Computer Science, Aalto University, Espoo, Finland

Interpreting gradient methods as fixed-point iterations, we provide a detailed analysis of those methods for minimizing convex objective functions. Due to their conceptual and algorithmic simplicity, gradient methods are widely used in machine learning for massive data sets (big data). In particular, stochastic gradient methods are considered the de-facto standard for training deep neural networks. Studying gradient methods within the realm of fixed-point theory provides us with powerful tools to analyze their convergence properties. In particular, gradient methods using inexact or noisy gradients, such as stochastic gradient descent, can be studied conveniently using well-known results on inexact fixed-point iterations. Moreover, as we demonstrate in this paper, the fixed-point approach allows an elegant derivation of accelerations for basic gradient methods. In particular, we will show how gradient descent can be accelerated by a fixed-point preserving transformation of an operator associated with the objective function.

OPEN ACCESS

Edited by:

Juergen Prestin,
University of Luebeck, Germany

Reviewed by:

Yekini Shehu,
University of Nigeria, Nsukka, Nigeria
Ming Tian,
Civil Aviation University of China,
China

*Correspondence:

Alexander Jung
alexander.jung@aalto.fi

Specialty section:

This article was submitted to
Mathematics of Computation and
Data Science,
a section of the journal
Frontiers in Applied Mathematics and
Statistics

Received: 26 July 2017

Accepted: 21 August 2017

Published: 05 September 2017

Citation:

Jung A (2017) A Fixed-Point of View
on Gradient Methods for Big Data.
Front. Appl. Math. Stat. 3:18.
doi: 10.3389/fams.2017.00018

Keywords: convex optimization, fixed point theory, big data, machine learning, contraction mapping, gradient descent, heavy balls

1. INTRODUCTION

One of the main recent trends within machine learning and data analytics using massive data sets is to leverage the inferential strength of the vast amounts of data by using relatively simple, but fast, optimization methods as algorithmic primitives [1]. Many of these optimization methods are modifications of the basic gradient descent (GD) method. Indeed, computationally more heavy approaches, such as interior point methods, are often infeasible for a given limited computational budget [2].

Moreover, the rise of deep learning has brought a significant boost for the interest in gradient methods. Indeed, a major insight within the theory of deep learning is that for typical high-dimensional models, e.g., those represented by deep neural networks, most of the local minima of the cost function (e.g., the empirical loss or training error) are reasonably close (in terms of objective value) to the global optimum [3]. These local minima can be found efficiently by gradient methods such as stochastic gradient descent (SGD), which is considered the de-facto standard algorithmic primitive for training deep neural networks [3].

This paper elaborates on the interpretation of some basic gradient methods such as GD and its variants as fixed-point iterations. These fixed-point iterations are obtained for operators associated with the convex objective function. Emphasizing the connection to fixed-point theory unleashes some powerful tools, e.g., on the acceleration of fixed-point iterations [4] or inexact fixed-point iterations [5, 6], for the analysis and construction of convex optimization methods.

In particular, we detail how the convergence of the basic GD iterations can be understood from the contraction properties of a specific operator which is associated naturally with a differentiable objective function. Moreover, we work out in some detail how the basic GD method can be

accelerated by modifying the operator underlying GD in a way that preserves its fixed-points but decreases the contraction factor which implies faster convergence by the contraction mapping theorem.

1.1. Outline

We discuss the basic problem of minimizing convex functions in Section (2). We then derive GD, which is a particular first order method, as a fixed-point iteration in Section (3). In Section (4), we introduce one of the most widely used computational models for convex optimization methods, i.e., the model of first order methods. In order to assess the efficiency of GD, which is a particular instance of a first order method, we present in Section (5) a lower bound on the number of iterations required by any first order method to reach a given sub-optimality. Using the insight provided from the fixed-point interpretation we show how to obtain an accelerated variant of GD in Section (6), which turns out to be optimal in terms of convergence rate.

1.2. Notation

The set of natural numbers is denoted $\mathbb{N} := \{1, 2, \dots\}$. Given a vector $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{C}^n$, we denote its l th entry by x_l . The (hermitian) transpose and trace of a square matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$ are denoted $(\mathbf{A}^H) \mathbf{A}^T$ and $\text{tr}\{\mathbf{A}\}$, respectively. The Euclidian norm of a vector \mathbf{x} is denoted $\|\mathbf{x}\| := \sqrt{\mathbf{x}^H \mathbf{x}}$. The spectral norm of a matrix \mathbf{M} is denoted $\|\mathbf{M}\| := \max_{\|\mathbf{x}\|=1} \|\mathbf{M}\mathbf{x}\|$. The spectral decomposition of a positive semidefinite (psd) matrix $\mathbf{Q} \in \mathbb{C}^{n \times n}$ is $\mathbf{Q} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$ with matrix $\mathbf{U} = (\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(n)})$ whose columns are the orthonormal eigenvectors $\mathbf{u}^{(i)} \in \mathbb{C}^n$ of \mathbf{Q} and the diagonal matrix $\mathbf{\Lambda}$ containing the eigenvalues $\lambda_1(\mathbf{Q}) \geq \dots \geq \lambda_n(\mathbf{Q}) \geq 0$. For a square matrix \mathbf{M} , we denote its spectral radius as $\rho(\mathbf{M}) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } \mathbf{M}\}$.

2. CONVEX FUNCTIONS

A function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if

$$f((1 - \alpha)\mathbf{x} + \alpha\mathbf{y}) \leq (1 - \alpha)f(\mathbf{x}) + \alpha f(\mathbf{y})$$

holds for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\alpha \in [0, 1]$ [2]. For a differentiable function $f(\cdot)$ with gradient $\nabla f(\mathbf{x})$, a necessary and sufficient condition for convexity is [7, p. 70]

$$f(\mathbf{y}) \geq f(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^T \nabla f(\mathbf{x}),$$

which has to hold for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Our main object of interest in this paper is the optimization problem

$$\mathbf{x}_0 \in \arg \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}). \tag{1}$$

Given a convex function $f(\mathbf{x})$, we aim at finding a point \mathbf{x}_0 with lowest function value $f(\mathbf{x}_0)$, i.e., $f(\mathbf{x}_0) = \min_{\mathbf{x}} f(\mathbf{x})$.

In order to motivate our interest in optimization problems like Equation (1), consider a machine learning problem based on training data $\mathcal{X} := \{\mathbf{z}^{(i)}\}_{i=1}^N$ consisting of N data points $\mathbf{z}^{(i)} = (\mathbf{d}^{(i)}, y^{(i)})$ with feature vector $\mathbf{d}^{(i)} \in \mathbb{R}^n$ (which might

represent the RGB pixel values of a webcam snapshot) and output or label $y^{(i)} \in \mathbb{R}$ (which might represent the local temperature during the snapshot). We wish to predict the label $y^{(i)}$ by a linear combination of the features, i.e.,

$$y^{(i)} \approx \mathbf{x}^T \mathbf{d}^{(i)}. \tag{2}$$

The choice for the weight vector $\mathbf{x} \in \mathbb{R}^n$ is typically based on balancing the empirical risk incurred by the predictor (2), i.e.,

$$(1/N) \sum_{i=1}^N (y^{(i)} - \mathbf{x}^T \mathbf{d}^{(i)})^2,$$

with some regularization term, e.g., measured by the squared norm $\|\mathbf{x}\|^2$. Thus, the learning problem amounts to solving the optimization problem

$$\mathbf{x}_0 = \arg \min_{\mathbf{x} \in \mathbb{R}^n} (1/N) \sum_{i=1}^N (y^{(i)} - \mathbf{x}^T \mathbf{d}^{(i)})^2 + \lambda \|\mathbf{x}\|^2. \tag{3}$$

The learning problem (3) is precisely of the form (1) with the convex objective function

$$f(\mathbf{x}) = (1/N) \sum_{i=1}^N (y^{(i)} - \mathbf{x}^T \mathbf{d}^{(i)})^2 + \lambda \|\mathbf{x}\|^2. \tag{4}$$

By choosing a large value for the regularization parameter λ , we de-emphasize the relevance of the training error and thus avoid overfitting. However, choosing λ too large induces a bias if the true underlying weight vector has a large norm [3, 8]. A principled approach to find a suitable value of λ is cross validation [3, 8].

2.1. Differentiable Convex Functions

Any differentiable function $f(\cdot)$ is accompanied by its gradient operator

$$\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n, \mathbf{x} \mapsto \nabla f(\mathbf{x}). \tag{5}$$

While the gradient operator ∇f is defined for any (even non-convex) differentiable function, the gradient operator of a convex function satisfies a strong structural property, i.e., it is a monotone operator [9].

2.2. Smooth and Strongly Convex Functions

If all second order partial derivatives of the function $f(\cdot)$ exist and are continuous, then $f(\cdot)$ is convex if and only if [7, p.71]

$$\nabla^2 f(\mathbf{x}) \succeq \mathbf{0} \text{ for every } \mathbf{x} \in \mathbb{R}^n.$$

We will focus on a particular class of twice differentiable convex functions, i.e., those with Hessian $\nabla^2 f(\mathbf{x})$ satisfying

$$L \leq \lambda_l(\nabla^2 f(\mathbf{x})) \leq U \text{ for every } \mathbf{x} \in \mathbb{R}^n, \tag{6}$$

with some known constants $U \geq L > 0$.

The set of convex functions $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying (6) will be denoted $\mathcal{S}_n^{L,U}$. As it turns out, the difficulty of finding the minimum of some function $f(\cdot) \in \mathcal{S}_n^{L,U}$ using gradient methods is essentially governed by the

$$\text{condition number } \kappa = U/L \text{ of the function class } \mathcal{S}_n^{L,U}. \quad (7)$$

Thus, regarding the difficulty of optimizing the functions $f(\cdot) \in \mathcal{S}_n^{L,U}$, the absolute values of the bounds L and U in (6) are not crucial, only their ratio $\kappa = U/L$ is.

One particular sub-class of functions $f(\cdot) \in \mathcal{S}_n^{L,U}$, which is of paramount importance for the analysis of gradient methods, are quadratic functions of the form

$$f(\mathbf{x}) = (1/2)\mathbf{x}^T \mathbf{Q}\mathbf{x} + \mathbf{q}^T \mathbf{x} + c, \quad (8)$$

with some vector $\mathbf{q} \in \mathbb{R}^n$ and a psd matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ having eigenvalues $\lambda(\mathbf{Q}) \in [L, U]$. As can be verified easily, the gradient and Hessian of a quadratic function of the form (8) are obtained as $\nabla f(\mathbf{x}) = \mathbf{Q}\mathbf{x} + \mathbf{q}$ and $\nabla^2 f(\mathbf{x}) = \mathbf{Q}$, respectively.

It turns out that most of the results (see below) on gradient methods for minimizing quadratic functions of the form (8), with some matrix \mathbf{Q} having eigenvalues $\lambda(\mathbf{Q}) \in [L, U]$, apply (with minor modifications) also when expanding their scope from quadratic functions to the larger set $\mathcal{S}_n^{L,U}$. This should not come as a surprise, since any function $f(\cdot) \in \mathcal{S}_n^{L,U}$ can be approximated locally around a point \mathbf{x}_0 by a quadratic function which is obtained by a truncated Taylor series [10]. In particular, we have Rudin [10, Theorem 5.15]

$$f(\mathbf{x}) = f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \nabla f(\mathbf{x}_0) + (1/2)(\mathbf{x} - \mathbf{x}_0)^T \nabla^2 f(\mathbf{u})(\mathbf{x} - \mathbf{x}_0), \quad (9)$$

where $\mathbf{u} = \eta\mathbf{x} + (1-\eta)\mathbf{x}_0$ with some $\eta \in [0, 1]$.

The crucial difference between the quadratic function (8) and a general function $f(\cdot) \in \mathcal{S}_n^{L,U}$ is that the matrix $\nabla^2 f(\mathbf{z})$ appearing in the quadratic form in (9) typically varies with the point \mathbf{x} . In particular, we can rewrite (9) as

$$f(\mathbf{x}) = f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \nabla f(\mathbf{x}_0) + (1/2)(\mathbf{x} - \mathbf{x}_0)^T \mathbf{Q}(\mathbf{x} - \mathbf{x}_0) + (1/2)(\mathbf{x} - \mathbf{x}_0)^T (\nabla^2 f(\mathbf{z}) - \mathbf{Q})(\mathbf{x} - \mathbf{x}_0), \quad (10)$$

with $\mathbf{Q} = \nabla^2 f(\mathbf{x}_0)$. The last summand in (10) quantifies the approximation error

$$\varepsilon(\mathbf{x}) = f(\mathbf{x}) - \tilde{f}(\mathbf{x}) = (1/2)(\mathbf{x} - \mathbf{x}_0)^T (\nabla^2 f(\mathbf{z}) - \mathbf{Q})(\mathbf{x} - \mathbf{x}_0) \quad (11)$$

obtained when approximating a function $f(\cdot) \in \mathcal{S}_n^{L,U}$ with the quadratic $\tilde{f}(\mathbf{x})$ obtained from (8) with the choices

$$\begin{aligned} \mathbf{Q} &= \nabla^2 f(\mathbf{x}_0), \\ \mathbf{q} &= \nabla f(\mathbf{x}_0) - \mathbf{Q}\mathbf{x}_0 \text{ and} \\ c &= f(\mathbf{x}_0) + (1/2)\mathbf{x}_0^T \mathbf{Q}\mathbf{x}_0 - \mathbf{x}_0^T \nabla f(\mathbf{x}_0). \end{aligned}$$

According to (6), which implies $\|\nabla^2 f(\mathbf{x}_0)\|, \|\nabla^2 f(\mathbf{z})\| \leq U$, we can bound the approximation error (11) as

$$\varepsilon(\mathbf{x}) \leq U\|\mathbf{x} - \mathbf{x}_0\|^2.$$

Thus, we can ensure a arbitrarily small approximation error ε by considering $f(\cdot)$ only over a neighborhood $\mathcal{B}(\mathbf{x}_0, r) := \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_0\| \leq r\}$ with sufficiently small radius $r > 0$.

Let us now verify that learning a (regularized) linear regression model (cf. 3) amounts to minimizing a convex quadratic function of the form (8). Indeed, using some elementary linear algebraic manipulations, we can rewrite the objective function in (4) as a quadratic of the form (8) using the particular choices $\mathbf{Q} = \mathbf{Q}_{\text{LR}}$ and $\mathbf{q} = \mathbf{q}_{\text{LR}}$ with

$$\mathbf{Q}_{\text{LR}} = \lambda \mathbf{I} + \frac{1}{N} \sum_{i=1}^N \mathbf{d}^{(i)} (\mathbf{d}^{(i)})^T, \text{ and } \mathbf{q}_{\text{LR}} = \frac{2}{N} \sum_{i=1}^N y^{(i)} \mathbf{d}^{(i)}. \quad (12)$$

The eigenvalues of the matrix \mathbf{Q}_{LR} obey [11]

$$\lambda \leq \lambda_l(\mathbf{Q}_{\text{LR}}) \leq \lambda + \lambda_1(\mathbf{D}^T \mathbf{D})$$

with the data matrix $\mathbf{D} := (\mathbf{d}^{(1)}, \dots, \mathbf{d}^{(N)}) \in \mathbb{R}^{n \times N}$. Hence, learning a regularized linear regression model via (3) amounts to minimizing a convex quadratic function $f(\cdot) \in \mathcal{S}_n^{L,U}$ with $L = \lambda$ and $U = \lambda + \lambda_1(\mathbf{D}^T \mathbf{D})$, where λ denotes the regularization parameter used in (3).

3. GRADIENT DESCENT

Let us now show how one of the most basic methods for solving the problem (1), i.e., the GD method, can be obtained naturally as fixed-point iterations involving the gradient operator ∇f (cf. 5).

Our point of departure is the necessary and sufficient condition [7]

$$\nabla f(\mathbf{x}_0) = \mathbf{0}, \quad (13)$$

for a vector $\mathbf{x}_0 \in \mathbb{R}^n$ to be optimal for the problem (1) with a convex differentiable objective function $f(\cdot) \in \mathcal{S}_n^{L,U}$.

Lemma 1. *We have $\nabla f(\mathbf{x}) = \mathbf{0}$ if and only if the vector $\mathbf{x} \in \mathbb{R}^n$ is a fixed point of the operator*

$$\mathcal{T}^{(\alpha)} : \mathbb{R}^n \rightarrow \mathbb{R}^n : \mathbf{x} \mapsto \mathbf{x} - \alpha \nabla f(\mathbf{x}), \quad (14)$$

for an arbitrary but fixed non-zero $\alpha \in \mathbb{R} \setminus \{0\}$. Thus,

$$\nabla f(\mathbf{x}) = \mathbf{0} \text{ if and only if } \mathcal{T}^{(\alpha)} \mathbf{x} = \mathbf{x}.$$

Proof. Consider a vector \mathbf{x} such that $\nabla f(\mathbf{x}) = \mathbf{0}$. Then,

$$\mathcal{T}^{(\alpha)} \mathbf{x} \stackrel{(14)}{=} \mathbf{x} - \alpha \nabla f(\mathbf{x}) = \mathbf{x}.$$

Conversely, let \mathbf{x} be a fixed point of $\mathcal{T}^{(\alpha)}$, i.e.,

$$\mathcal{T}^{(\alpha)} \mathbf{x} = \mathbf{x}. \quad (15)$$

Then,

$$\begin{aligned} \nabla f(\mathbf{x}) &\stackrel{\alpha \neq 0}{=} (1/\alpha)(\mathbf{x} - (\mathbf{x} - \alpha \nabla f(\mathbf{x}))) \\ &\stackrel{(14)}{=} (1/\alpha)(\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{x}) \\ &\stackrel{(15)}{=} \mathbf{0}. \end{aligned}$$

□

According to Lemma (1), the solution \mathbf{x}_0 of the optimization problem (1) is obtained as the fixed point of the operator $\mathcal{T}^{(\alpha)}$ (cf. 14) with some non-zero α . As we will see shortly, the freedom in choosing different values for α can be exploited in order to compute the fixed points of $\mathcal{T}^{(\alpha)}$ more efficiently.

A straightforward approach to finding the fixed-points of an operator $\mathcal{T}^{(\alpha)}$ is via the fixed-point iteration

$$\mathbf{x}^{(k+1)} = \mathcal{T}^{(\alpha)}\mathbf{x}^{(k)}. \tag{16}$$

By tailoring a fundamental result of analysis (cf. [10], Theorem 9.23], we can characterize the convergence of the sequence $\mathbf{x}^{(k)}$ obtained from (16).

Lemma 2. Assume that for some $q \in [0, 1)$, we have

$$\|\mathcal{T}^{(\alpha)}\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{y}\| \leq q\|\mathbf{x} - \mathbf{y}\|, \tag{17}$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Then, the operator $\mathcal{T}^{(\alpha)}$ has a unique fixed point \mathbf{x}_0 and the iterates $\mathbf{x}^{(k)}$ (cf. 16) satisfy

$$\|\mathbf{x}^{(k)} - \mathbf{x}_0\| \leq \|\mathbf{x}^{(0)} - \mathbf{x}_0\|q^k. \tag{18}$$

Proof. Let us first verify that the operator $\mathcal{T}^{(\alpha)}$ cannot have two different fixed points. Indeed, assume there would be two different fixed points \mathbf{x}, \mathbf{y} such that

$$\mathbf{x} = \mathcal{T}^{(\alpha)}\mathbf{x}, \text{ and } \mathbf{y} = \mathcal{T}^{(\alpha)}\mathbf{y}. \tag{19}$$

This would imply, in turn,

$$\begin{aligned} q\|\mathbf{x} - \mathbf{y}\| &\stackrel{(17)}{\geq} \|\mathcal{T}^{(\alpha)}\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{y}\| \\ &\stackrel{(19)}{=} \|\mathbf{x} - \mathbf{y}\|. \end{aligned}$$

However, since $q < 1$, this inequality can only be satisfied if $\|\mathbf{x} - \mathbf{y}\| = 0$, i.e., we must have $\mathbf{x} = \mathbf{y}$. Thus, we have shown that no two different fixed points can exist. The existence of one unique fixed point \mathbf{x}_0 follows from Rudin [10, Theorem 9.23].

The estimate (18) can be obtained by induction and noting

$$\begin{aligned} \|\mathbf{x}^{(k+1)} - \mathbf{x}_0\| &\stackrel{(16)}{=} \|\mathcal{T}^{(\alpha)}\mathbf{x}^{(k)} - \mathbf{x}_0\| \\ &\stackrel{(a)}{=} \|\mathcal{T}^{(\alpha)}\mathbf{x}^{(k)} - \mathcal{T}^{(\alpha)}\mathbf{x}_0\| \\ &\stackrel{(17)}{\leq} q\|\mathbf{x}^{(k)} - \mathbf{x}_0\|. \end{aligned}$$

Here, step (a) is valid since \mathbf{x}_0 is a fixed point of $\mathcal{T}^{(\alpha)}$, i.e., $\mathbf{x}_0 = \mathcal{T}^{(\alpha)}\mathbf{x}_0$. □

In order to apply Lemma (2) to (16), we have to ensure that the operator $\mathcal{T}^{(\alpha)}$ is a contraction, i.e., it satisfies (17) with some contraction coefficient $q \in [0, 1)$. For the operator $\mathcal{T}^{(\alpha)}$ (cf. 14) associated with the function $f(\cdot) \in \mathcal{S}_n^{L,U}$ this can be verified by standard results from vector analysis.

Lemma 3. Consider the operator $\mathcal{T}^{(\alpha)} : \mathbf{x} \mapsto \mathbf{x} - \alpha \nabla f(\mathbf{x})$ with some convex function $f(\cdot) \in \mathcal{S}_n^{L,U}$. Then,

$$\|\mathcal{T}^{(\alpha)}\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{y}\| \leq q(\alpha)\|\mathbf{x} - \mathbf{y}\|$$

with contraction factor

$$q(\alpha) = \max\{|1 - U\alpha|, |1 - L\alpha|\}. \tag{20}$$

Proof. First,

$$\begin{aligned} \mathcal{T}^{(\alpha)}\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{y} &\stackrel{(14)}{=} (\mathbf{x} - \mathbf{y}) - \alpha(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})) \\ &\stackrel{(a)}{=} (\mathbf{x} - \mathbf{y}) - \alpha \nabla^2 f(\mathbf{z})(\mathbf{x} - \mathbf{y}) \\ &= (\mathbf{I} - \alpha \nabla^2 f(\mathbf{z}))(\mathbf{x} - \mathbf{y}) \end{aligned} \tag{21}$$

using $\mathbf{z} = \eta\mathbf{x} + (1 - \eta)\mathbf{y}$ with some $\eta \in [0, 1]$. Here, we used in step (a) the mean value theorem of vector calculus [10, Theorem 5.10].

Combining (21) with the submultiplicativity of Euclidean and spectral norm [11, p. 55] yields

$$\|\mathcal{T}^{(\alpha)}\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{y}\| \leq \|\mathbf{x} - \mathbf{y}\| \|\mathbf{I} - \alpha \nabla^2 f(\mathbf{z})\|. \tag{22}$$

The matrix $\mathbf{M}^{(\alpha)} := \mathbf{I} - \alpha \nabla^2 f(\mathbf{z})$ is symmetric ($\mathbf{M}^{(\alpha)} = (\mathbf{M}^{(\alpha)})^T$) with real-valued eigenvalues [11]

$$\lambda_l(\mathbf{M}^{(\alpha)}) \in [1 - U\alpha, 1 - L\alpha]. \tag{23}$$

Since also

$$\begin{aligned} \|\mathbf{M}^{(\alpha)}\| &= \max\{|\lambda_l|\} \\ &\stackrel{(23)}{\leq} \max\{|1 - U\alpha|, |1 - L\alpha|\}, \end{aligned} \tag{24}$$

we obtain from (22)

$$\|\mathcal{T}^{(\alpha)}\mathbf{x} - \mathcal{T}^{(\alpha)}\mathbf{y}\| \stackrel{(24)}{\leq} \|\mathbf{x} - \mathbf{y}\| \max\{|1 - U\alpha|, |1 - L\alpha|\}.$$

□

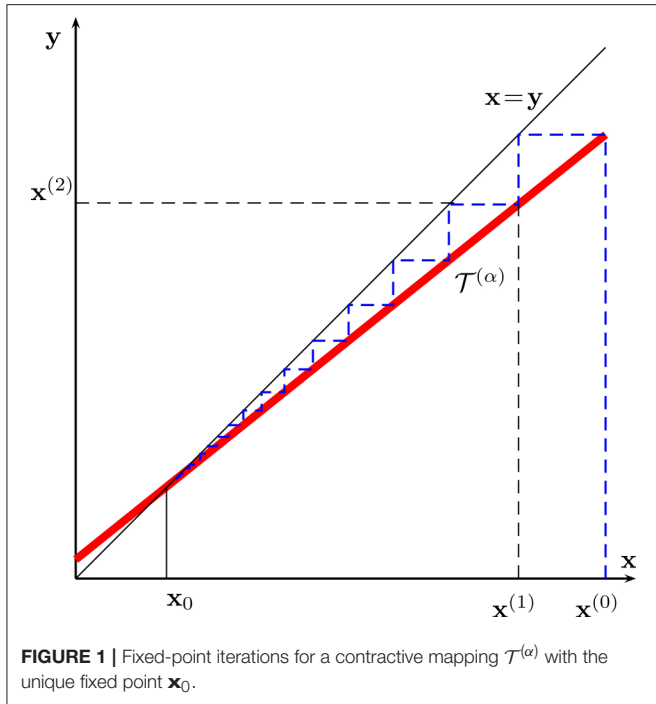
It will be handy to write out the straightforward combination of Lemma (2) and Lemma (3).

Lemma 4. Consider a convex function $f(\cdot) \in \mathcal{S}_n^{L,U}$ with the unique minimizer \mathbf{x}_0 , i.e., $f(\mathbf{x}_0) = \min_{\mathbf{x}} f(\mathbf{x})$. We then construct the operator $\mathcal{T}^{(\alpha)} : \mathbf{x} \mapsto \mathbf{x} - \alpha \nabla f(\mathbf{x})$ with a step size α such that

$$q(\alpha) \stackrel{(20)}{=} \max\{|1 - U\alpha|, |1 - L\alpha|\} < 1.$$

Then, starting from an arbitrary initial guess $\mathbf{x}^{(0)}$, the iterates $\mathbf{x}^{(k)}$ (cf. 16) satisfy

$$\|\mathbf{x}^{(k)} - \mathbf{x}_0\| \leq \|\mathbf{x}^{(0)} - \mathbf{x}_0\| [q(\alpha)]^k. \tag{25}$$



According to Lemma (4), and also illustrated in **Figure 1**, starting from an arbitrary initial guess $\mathbf{x}^{(0)}$, the sequence $\mathbf{x}^{(k)}$ generated by the fixed-point iteration (16) is guaranteed to converge to the unique solution \mathbf{x}_0 of (1), i.e., $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}_0$. What is more, this convergence is quite fast, since the error $\|\mathbf{x}^{(k)} - \mathbf{x}_0\|$ decays at least exponentially according to (25). Loosely speaking, this exponential decrease implies that the number of additional iterations required to have on more correct digit in $\mathbf{x}^{(k)}$ is constant.

Let us now work out the iterations (16) more explicitly by inserting the expression (14) for the operator $\mathcal{T}(\alpha)$. We then obtain the following equivalent representation of (16):

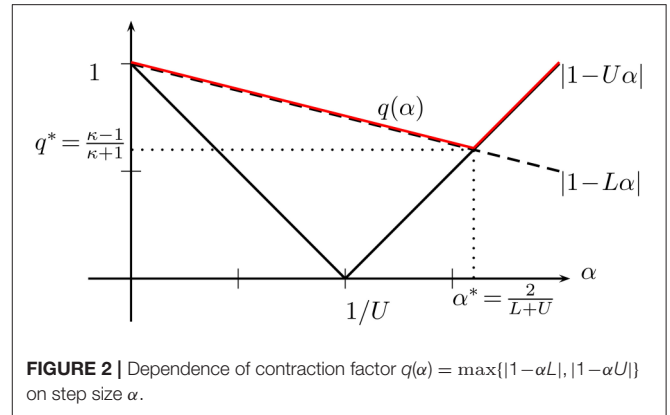
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}). \tag{26}$$

This iteration is nothing but plain vanilla GD using a fixed step size α [3].

Since the GD iteration (26) is precisely the fixed-point iteration (16), we can use Lemma (4) to characterize the convergence (rate) of GD. In particular, convergence of GD is ensured by choosing the step size of GD (26) such that $q(\alpha) = \max\{|1 - U\alpha|, |1 - L\alpha|\} < 1$. Moreover, in order to make the convergence as fast as possible we need to choose the step size $\alpha = \alpha^*$ which makes the contraction factor $q(\alpha)$ (cf. 20) as small as possible.

In **Figure 2**, we illustrate how the quantities $|1 - \alpha L|$ and $|1 - \alpha U|$ evolve as the step size α (cf. 26) is varied. From **Figure 2** we can easily read off the optimal choice

$$\alpha^* = \frac{2}{L + U} \tag{27}$$



yielding the smallest possible contraction factor

$$q^* = \min_{\alpha \in [0,1]} q(\alpha) = \frac{U - L}{U + L} \stackrel{20}{=} \frac{\kappa - 1}{\kappa + 1}.$$

We have arrived at the following characterization of GD for minimizing convex functions $f(\cdot) \in \mathcal{S}_n^{L,U}$.

Theorem 5. Consider the optimization problem (1) with objective function $f(\cdot) \in \mathcal{S}_n^{L,U}$, where the parameters L and U are fixed and known. Starting from an arbitrarily chosen initial guess $\mathbf{x}^{(0)}$, we construct a sequence by GD (26) using the optimal step size (27). Then,

$$\|\mathbf{x}^{(k)} - \mathbf{x}_0\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\mathbf{x}^{(0)} - \mathbf{x}_0\|. \tag{28}$$

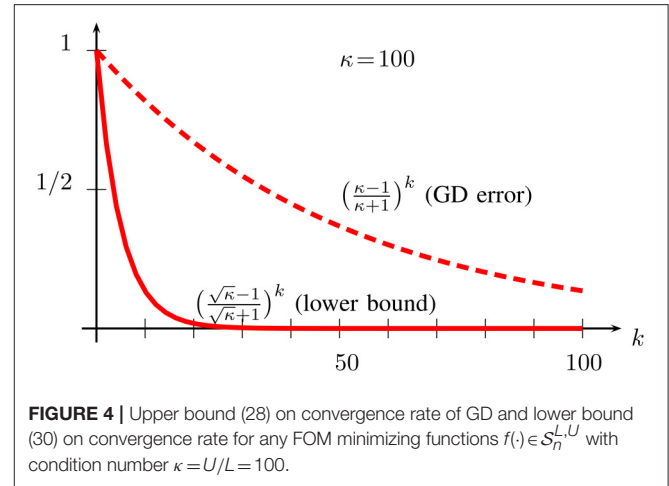
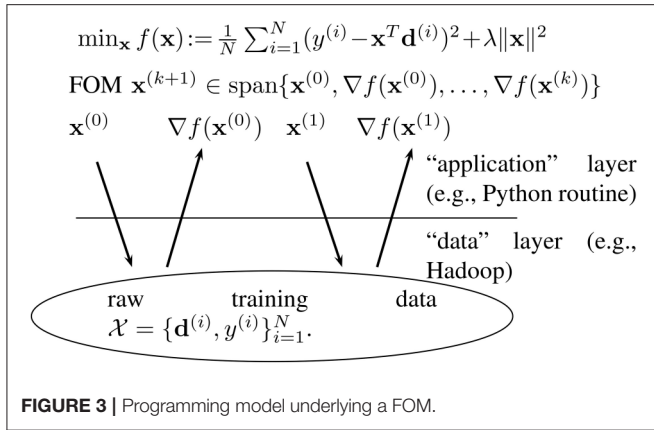
In what follows, we will use the shorthand $\mathcal{T} := \mathcal{T}(\alpha^*)$ for the gradient operator $\mathcal{T}(\alpha)$ (cf. 14) obtained for the optimal step size $\alpha = \alpha^*$ (cf. 27).

4. FIRST ORDER METHODS

Without a computational model taking into account a finite amount of resources, the study of the computational complexity inherent to (1) becomes meaningless. Consider having unlimited computational resources at our disposal. Then, we could build an “optimization device” which maps each function $f(\cdot) \in \mathcal{S}_n^{L,U}$ to its unique minimum \mathbf{x}_0 . Obviously, this approach is infeasible since we cannot perfectly represent such a mapping, let alone its domain $\mathcal{S}_n^{L,U}$, using a physical hardware which allows us only to handle finite sets instead of continuous spaces like $\mathcal{S}_n^{L,U}$.

Let us further illustrate the usefulness of using a computational model in the context of machine learning from massive data sets (big data). In particular, as we have seen in the previous section, the regularized linear regression model (3) amounts to minimizing a convex quadratic function (8) with the particular choices (12). Even for this most simple machine learning model, it is typically infeasible to have access to a complete description of the objective function (8).

Indeed, in order to fully specify the quadratic function in (8), we need to fully specify the matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and the vector



$\mathbf{q} \in \mathbb{R}^n$. For the (regularized) linear regression model (3) this would require to compute \mathbf{Q}_{LR} (cf. 12) from the training data $\mathcal{X} = \{\mathbf{z}^{(i)}\}_{i=1}^N$. Computing the matrix \mathbf{Q}_{LR} in a naive way, i.e., without exploiting any additional structure, amounts to a number of arithmetic operations on the order of $N \cdot n^2$. This might be prohibitive in a typical big data application with N and n being on the order of billions and using distributed storage of the training data \mathcal{X} [12].

There has emerged a widely accepted computational model for convex optimization which abstracts away the details of the computational (hard- and software) infrastructure. Within this computational model, an optimization method for solving (1) is not provided with a complete description of the objective function, but rather it can access the objective function only via an “oracle” [2, 13].

We might think of an oracle model as an application programming interface (API), which specifies the format of queries which can be issued by a convex optimization method executed on an application layer (cf. Figure 3). There are different types of oracle models but one of the most popular type (in particular for big data applications) is a first order oracle [13]. Given a query point $\mathbf{x} \in \mathbb{R}^n$, a first order oracle returns the gradient $\nabla f(\mathbf{x})$ of the objective function at this particular point.

A first order method (FOM) aims at solving (1) by sequentially querying a first order oracle, at the current iterate $\mathbf{x}^{(k)}$, to obtain the gradient $\nabla f(\mathbf{x}^{(k)})$ (cf. Figure 3). Using the current and past information obtained from the oracle, a FOM then constructs the new iterate $\mathbf{x}^{(k+1)}$ such that eventually $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}_0$. For the sake of simplicity and without essential loss in generality, we will only consider FOMs whose iterates $\mathbf{x}^{(k)}$ satisfy [13]

$$\mathbf{x}^{(k)} \in \text{span}\{\mathbf{x}^{(0)}, \nabla f(\mathbf{x}^{(0)}), \dots, \nabla f(\mathbf{x}^{(k-1)})\}. \quad (29)$$

5. LOWER BOUNDS ON NUMBER OF ITERATIONS

According to Section (3), solving (1) can be accomplished by the simple GD iterations (26). The particular choice α^* (27) for the step size α in (26) ensures the convergence rate $\left(\frac{\kappa-1}{\kappa+1}\right)^k$ with the condition number $\kappa = U/L$ of the function class $S_n^{L,U}$. While

this convergence is quite fast, i.e., the error decays exponentially with iteration number k , we would, of course, like to know how efficient this method is in general.

As detailed in Section (4), in order to study the computational complexity and efficiency of convex optimization methods, we have to define a computational model such as those underlying FOMs (cf. Figure 3). The next result provides a fundamental lower bound on the convergence rate of any FOM (cf. 29) for solving (1).

Theorem 6. Consider a particular FOM, which for a given convex function $f(\cdot) \in S_n^{L,U}$ generates iterates $\mathbf{x}^{(k)}$ satisfying (29). For fixed L, U there is a sequence of functions $f_n(\cdot) \in S_n^{L,U}$ (indexed by dimension n) such that

$$\|\mathbf{x}^{(k)} - \mathbf{x}_0\| \geq \|\mathbf{x}^{(0)} - \mathbf{x}_0\| \frac{1-1/\sqrt{\kappa}}{1+1/\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k - |\delta(n)| \quad (30)$$

with a sequence $\delta(n)$ such that $\lim_{n \rightarrow \infty} |\delta(n)| = 0$.

Proof. see Section (8.1). □

There is a considerable gap between the upper bound (28) on the error achieved by GD after k iterations and the lower bound (30) which applies to any FOM which is run for the same number iterations. In order to illustrate this gap, we have plotted in Figure 4 the upper and lower bound for the (quite moderate) condition number $\kappa = 100$.

Thus, there might exist a FOM which converges faster than the GD method (28) and comes more close to the lower bound (30). Indeed, in the next section, we will detail how to obtain an accelerated FOM by applying a fixed point preserving transformation to the operator \mathcal{T} (cf. 16), which is underlying the GD method (26). This accelerated gradient method is known as the heavy balls (HB) method [14] and effectively achieves the lower bound (30), i.e., the HB method is already optimal among all FOMs for solving (1) with an objective function $f(\cdot) \in S_n^{L,U}$.

6. ACCELERATING GRADIENT DESCENT

Let us now show how to modify the basic GD method (26) in order to obtain an accelerated FOM, whose convergence rate essentially matches the lower bound (30) for the function class $S_n^{L,U}$ with condition number $\kappa = U/L > 1$ (cf. 7) and is therefore optimal among all FOMs.

Our derivation of this accelerated gradient method, which is inspired by the techniques used in Ghadimi et al. [15], starts from an equivalent formulation of GD as the fixed-point iteration

$$\bar{\mathbf{x}}^{(k)} = \bar{\mathcal{T}}\bar{\mathbf{x}}^{(k-1)} \tag{31}$$

with the operator

$$\bar{\mathcal{T}}: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n} : \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{u} - \alpha \nabla \mathbf{u} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathcal{T}\mathbf{u} \\ \mathbf{u} \end{pmatrix}. \tag{32}$$

As can be verified easily, the fixed-point iteration (31) starting from an arbitrary initial guess $\bar{\mathbf{x}}^{(0)} = \begin{pmatrix} \mathbf{z}^{(0)} \\ \mathbf{y}^{(0)} \end{pmatrix}$ is related to the GD iterate $\mathbf{x}^{(k)}$ (cf. 26), using initial guess $\mathbf{z}^{(0)}$, as

$$\bar{\mathbf{x}}^{(k)} = \begin{pmatrix} \mathbf{x}^{(k)} \\ \mathbf{x}^{(k-1)} \end{pmatrix} \tag{33}$$

for all iterations $k \geq 1$.

By the equivalence (33), Theorem (5) implies that for any initial guess $\bar{\mathbf{x}}^{(0)}$ the iterations (31) converge to the fixed point

$$\bar{\mathbf{x}}_0 = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_0 \end{pmatrix} \in \mathbb{R}^{2n} \tag{34}$$

with \mathbf{x}_0 being the unique minimizer of $f(\cdot) \in S_n^{L,U}$. Moreover, the convergence rate of the fixed-point iterations (31) is precisely the same as those of the GD method, i.e., governed by the decay of $(\frac{\kappa-1}{\kappa+1})^k$, which is obtained for the optimal step size $\alpha = \alpha^*$ (cf. 27).

We will now modify the operator $\bar{\mathcal{T}}$ in (32) to obtain a new operator $\mathcal{M}: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ which has the same fixed points (34) but improved contraction behavior, i.e., the fixed point iteration

$$\tilde{\mathbf{x}}^{(k)} = \mathcal{M}\tilde{\mathbf{x}}^{(k-1)}, \tag{35}$$

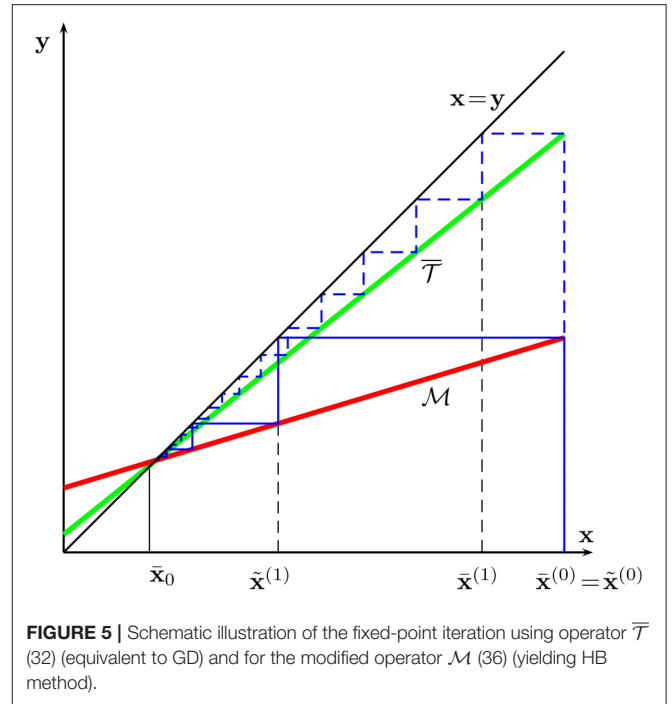
will converge faster than those obtained from $\bar{\mathcal{T}}$ in (31) (cf. Figure 5). In particular, this improved operator \mathcal{M} is defined as

$$\mathcal{M}: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n} : \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{u} - \tilde{\alpha} \nabla \mathbf{u} + \tilde{\beta}(\mathbf{u} - \mathbf{v}) \\ \mathbf{u} \end{pmatrix}, \tag{36}$$

with

$$\tilde{\alpha} = \frac{4}{(\sqrt{U} + \sqrt{L})^2}, \text{ and } \tilde{\beta} = \left[\frac{\sqrt{U} - \sqrt{L}}{\sqrt{U} + \sqrt{L}} \right]^2. \tag{37}$$

As can be verified easily, the fixed point $(\mathbf{x}_0^T, \mathbf{x}_0^T)^T$ of $\bar{\mathcal{T}}$ is also a fixed point of \mathcal{M} .



Before we analyze the convergence rate of the fixed-point iteration (35), let us work out explicitly the FOM which is represented by the fixed-point iteration (35). To this end, we partition the k th iterate, for $k \geq 1$, as

$$\tilde{\mathbf{x}}^{(k)} = \begin{pmatrix} \mathbf{x}_{\text{HB}}^{(k)} \\ \mathbf{x}_{\text{HB}}^{(k-1)} \end{pmatrix}. \tag{38}$$

Inserting (38) into (35), we have for $k \geq 1$

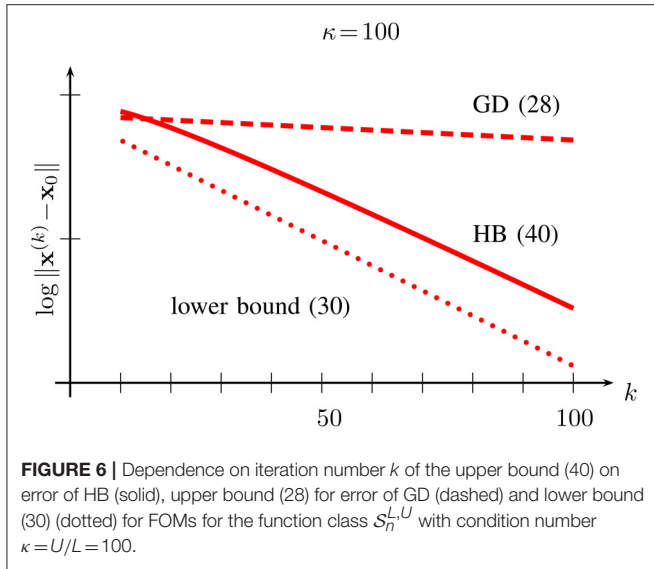
$$\mathbf{x}_{\text{HB}}^{(k)} = \mathbf{x}_{\text{HB}}^{(k-1)} - \tilde{\alpha} \nabla f(\mathbf{x}_{\text{HB}}^{(k-1)}) + \tilde{\beta}(\mathbf{x}_{\text{HB}}^{(k-1)} - \mathbf{x}_{\text{HB}}^{(k-2)}) \tag{39}$$

with the convention $\mathbf{x}_{\text{HB}}^{(-1)} = \mathbf{0}$. The iteration (39) defines the HB method [14] for solving the optimization problem (1). As can be verified easily, like the GD method, the HB method is a FOM. However, contrary to the GD iteration (26), the HB iteration (39) also involves the penultimate iterate $\mathbf{x}_{\text{HB}}^{(k-2)}$ for determining the new iterate $\mathbf{x}_{\text{HB}}^{(k)}$.

We will now characterize the converge rate of the HB method (39) via its fixed-point equivalent (35). To this end, we restrict ourselves to the subclass of $S_n^{L,U}$ given by quadratic functions of the form (8).

Theorem 7. Consider the optimization problem (1) with objective function $f(\cdot) \in S_n^{L,U}$ which is a quadratic (26). Starting from an arbitrarily chosen initial guess $\mathbf{x}_{\text{HB}}^{(-1)}$ and $\mathbf{x}_{\text{HB}}^{(0)}$, we construct a sequence $\mathbf{x}_{\text{HB}}^{(k)}$ via iterating (27). Then,

$$\|\mathbf{x}_{\text{HB}}^{(k)} - \mathbf{x}_0\| \leq C(\kappa)k \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k (\|\mathbf{x}_{\text{HB}}^{(0)} - \mathbf{x}_0\| + \|\mathbf{x}_{\text{HB}}^{(-1)} - \mathbf{x}_0\|). \tag{40}$$



with

$$C(\kappa) = 4(2 + 2\tilde{\beta} + \tilde{\alpha}) \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1}.$$

Proof. see Section (8.2). □

The upper bound (40) differs from the lower bound (30) by the factor k . However, the discrepancy is rather decent as this linear factor in (40) grows much slower than the exponential $(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^k$ in (40) decays. In **Figure 6**, we depict the upper bound (40) on the error of the HB iterations (39) along with the upper bound (28) on the error of the GD iterations (26) and the lower bound (30) on the error of any FOM after k iterations.

We highlight that, strictly speaking, the bound (40) only applies to a subclass of smooth strongly convex functions $f(\cdot) \in \mathcal{S}_n^{L,U}$, i.e., it applies only to quadratic functions of the form (8). However, as discussed in Section (2), given a particular point \mathbf{x} , we can approximate an arbitrary function $f(\cdot) \in \mathcal{S}_n^{L,U}$ with a quadratic function $\tilde{f}(\mathbf{x})$ of the form (8). The approximation error $\varepsilon(\mathbf{x})$ (cf. (11)) will be small for all points \mathbf{x} sufficiently close to \mathbf{x}_0 . Making this reasoning more precise and using well-known results on fixed-point iterations with inexact updates [6], one can verify that the bound (40) essentially applies to any function $f(\cdot) \in \mathcal{S}_n^{L,U}$.

7. CONCLUSIONS

We have presented a fixed-point theory of some basic gradient methods for minimizing convex functions. The approach via fixed-point theory allows for a rather elegant analysis of the convergence properties of these gradient methods. In particular, their convergence rate is obtained as the contraction factor for an operator associated with the objective function.

The fixed-point approach is also appealing since it leads rather naturally to the acceleration of gradient methods via fixed-point

preserving transformations of the underlying operator. We plan to further develop the fixed-point theory of gradient methods in order to accommodate stochastic variants of GD such as SGD. Furthermore, we can bring the popular class of proximal methods into the picture by replacing the gradient operator underlying GD with the proximal operator.

However, by contrast to FOMs (such as the GD method), proximal methods use a different oracle model (cf. **Figure 3**). In particular, proximal methods require an oracle which can evaluate the proximal mapping efficiently which is typically more expensive than gradient evaluations. Nonetheless, the popularity of proximal methods is due to the fact that for objective functions arising in many important machine learning applications, the proximal mapping can be evaluated efficiently.

8. PROOFS OF MAIN RESULTS

In this section we present the (somewhat lengthy) proofs for the main results stated in Sections (5) and (6).

8.1. Proof of Theorem (6)

Without loss of generality we consider FOM which use the initial guess $\mathbf{x}^{(0)} = \mathbf{0}$. Let us now construct a function $f_n(\cdot) \in \mathcal{S}_n^{L,U}$ which is particularly difficult to optimize by a FOM (cf. 29) such as the GD method (26). In particular, this function is the quadratic

$$\hat{f}(\mathbf{x}) = (1/2)\mathbf{x}^T \mathbf{P} \mathbf{x} + \tilde{\mathbf{q}}^T \mathbf{x} \tag{41}$$

with vector

$$\tilde{\mathbf{q}} = \frac{L(\kappa - 1)}{4} (1, 0, \dots, 0)^T \in \mathbb{R}^n \tag{42}$$

and matrix

$$\mathbf{P} = (L/4)(\kappa - 1)\tilde{\mathbf{Q}} + L\mathbf{I} \in \mathbb{R}^{n \times n}. \tag{43}$$

The matrix $\tilde{\mathbf{Q}}$ is defined row-wise by successive circular shifts of its first row

$$\tilde{\mathbf{q}} = (2, -1, 0, \dots, 0, -1)^T \in \mathbb{R}^n. \tag{44}$$

Note that the matrix \mathbf{P} in (43) is a circulant matrix [16] with orthonormal eigenvectors $\{\mathbf{u}^{(l)}\}_{l=1}^n$ given element-wise as

$$u_i^{(l)} = (1/\sqrt{n}) \exp(j2\pi(i-1)(l-1)/n). \tag{45}$$

The eigenvalues $\lambda_l(\mathbf{P})$ of the circulant matrix \mathbf{P} are obtained as the discrete Fourier transform (DFT) coefficients of its first row [16]

$$\mathbf{p} = \frac{L(\kappa - 1)}{4} \tilde{\mathbf{q}} + L\mathbf{e}_1^T \stackrel{(44)}{=} \frac{L(\kappa - 1)}{4} (2, -1, 0, \dots, 0, -1) + L(1, 0, \dots, 0)^T, \tag{46}$$

i.e.,

$$\lambda_l(\mathbf{P}) = \sum_{i=1}^n p_i \exp(-j2\pi(i-1)(l-1)/n) \tag{47}$$

$$\stackrel{(46)}{=} (L/2)(\kappa-1)(1-\cos(-2\pi(i-1)/n)) + L.$$

Thus, $\lambda_l(\mathbf{P}) \in [L, U]$ and, in turn, $f_n(\cdot) \in \mathcal{S}_n^{L,U}$ (cf. 6).

Consider the sequence $\mathbf{x}^{(k)}$ generated by some FOM, i.e., which satisfies (29), for the particular objective function $f_n(\mathbf{x})$ (cf. 41) using initial guess $\mathbf{x}_0 = \mathbf{0}$. It can be verified easily that the k th iterate $\mathbf{x}^{(k)}$ has only zero entries starting from index $k + 1$, i.e.,

$$x_l^{(k)} = 0 \text{ for all } l \in \{k + 1, \dots, n\}.$$

This implies

$$\|\mathbf{x}^{(k)} - \mathbf{x}_0\| \geq |x_{0,k+1}|. \tag{48}$$

The main part of the proof is then to show that the minimizer \mathbf{x}_0 for the particular function $f_n(\cdot)$ cannot decay too fast, i.e., we will derive a lower bound on $|x_{0,k+1}|$.

Let us denote the DFT coefficients of the finite length discrete time signal represented by the vector $\tilde{\mathbf{q}}$ as

$$c_l = \sum_{i=1}^n \tilde{q}_i \exp(-j2\pi(i-1)l/n)$$

$$\stackrel{(42)}{=} (L/4)(\kappa-1). \tag{49}$$

Using the optimality condition (13), the minimizer for (41) is

$$\mathbf{x}_0 = -\mathbf{P}^{-1}\tilde{\mathbf{q}}. \tag{50}$$

Inserting the spectral decomposition $\mathbf{P} = \sum_{l=1}^n \lambda_l \mathbf{u}^{(l)} (\mathbf{u}^{(l)})^H$ [16, Theorem 3.1] of the psd matrix \mathbf{P} into (50),

$$x_{0,k} = -(\mathbf{P}^{-1}\tilde{\mathbf{q}})_k$$

$$\stackrel{(45)}{=} -(1/n) \sum_{i=1}^n (c_i/\lambda_i) \exp(j2\pi(i-1)(k-1)/n)$$

$$\stackrel{(47),(49)}{=} -\frac{1}{n} \sum_{i=1}^n \frac{\exp(j2\pi(i-1)(k-1)/n)}{2(1-\cos(-2\pi(i-1)/n)) + 4/(\kappa-1)}. \tag{51}$$

We will also need a lower bound on the norm $\|\mathbf{x}_0\|$ of the minimizer of $f_n(\cdot)$. This bound can be obtained from (50) and $\lambda_l(\mathbf{P}) \in [L, U]$, i.e., $\lambda_l(\mathbf{P}^{-1}) \in [1/U, 1/L]$,

$$\|\mathbf{x}_0\| \leq (1/L)\|\tilde{\mathbf{q}}\| \stackrel{(42)}{=} \frac{\kappa-1}{4}. \tag{52}$$

The last expression in (51) is a Riemann sum for the integral $\int_{\theta=0}^1 \frac{\exp(-j2\pi\theta)}{2(1-\exp(-j2\pi\theta)) + 4/(\kappa-1)} d\theta$. Indeed, by basic calculus [10, Theorem 6.8]

$$x_{0,k} = - \int_{\theta=0}^1 \frac{\exp(j2\pi(k-1)\theta)}{2(1-\cos(2\pi\theta)) + 4/(\kappa-1)} d\theta + \delta(n) \tag{53}$$

where the error $\delta(n)$ becomes arbitrarily small for sufficiently large n , i.e., $\lim_{n \rightarrow \infty} |\delta(n)| = 0$.

According to Lemma (9),

$$\int_{\theta=0}^1 \frac{\exp(j2\pi(k-1)\theta)}{2(1-\cos(2\pi\theta)) + 4/(\kappa-1)} d\theta = \frac{\kappa-1}{4\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k,$$

which, by inserting into (53), yields

$$x_{0,k} = -\frac{\kappa-1}{4\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k + \delta(n). \tag{54}$$

Putting together the pieces,

$$\|\mathbf{x}^{(k)} - \mathbf{x}_0\| \stackrel{(48)}{\geq} |x_{0,k+1}|$$

$$\stackrel{(54)}{\geq} \frac{\kappa-1}{4\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right) \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k - |\delta(n)|$$

$$\stackrel{(52)}{\geq} \|\mathbf{x}_0\| \frac{1-1/\sqrt{\kappa}}{1+\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k - |\delta(n)|$$

$$\stackrel{\mathbf{x}^{(0)}=\mathbf{0}}{=} \|\mathbf{x}^{(0)} - \mathbf{x}_0\| \frac{1-1/\sqrt{\kappa}}{1+\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k - |\delta(n)|.$$

8.2. Proof of Theorem (7)

By evaluating the operator \mathcal{M} (cf. 36) for a quadratic function $f(\cdot)$ of the form (8), we can verify

$$\mathcal{M}\mathbf{x} - \mathcal{M}\mathbf{y} = \mathbf{R}(\mathbf{x} - \mathbf{y}) \tag{55}$$

with the matrix

$$\mathbf{R} = \begin{pmatrix} (1+\tilde{\beta})\mathbf{I} - \tilde{\alpha}\mathbf{Q} & -\tilde{\beta}\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{pmatrix}. \tag{56}$$

This matrix $\mathbf{R} \in \mathbb{R}^{2n \times 2n}$ is a 2×2 block matrix whose individual blocks can be diagonalized simultaneously via the orthonormal eigenvectors $\mathbf{U} = (\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(n)})$ of the psd matrix \mathbf{Q} . Inserting the spectral decomposition $\mathbf{Q} = \mathbf{U} \text{diag}\{\lambda_i\}_{i=1}^n \mathbf{U}^H$ into (56),

$$\mathbf{R} = \mathbf{U} \mathbf{P} \mathbf{B} \mathbf{P}^H \mathbf{U}^H, \tag{57}$$

with some (orthonormal) permutation matrix \mathbf{P} and a block diagonal matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}^{(1)} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{B}^{(n)} \end{pmatrix}, \mathbf{B}^{(i)} = \begin{pmatrix} 1+\tilde{\beta}-\tilde{\alpha}\lambda_i & -\tilde{\beta} \\ 1 & 0 \end{pmatrix}. \tag{58}$$

Combining (57) with (55) and inserting into (35) yields

$$\tilde{\mathbf{x}}^{(k)} - \tilde{\mathbf{x}}_0 = \mathbf{U} \mathbf{P} \mathbf{B}^k \mathbf{P}^H \mathbf{U}^H (\tilde{\mathbf{x}}^{(0)} - \tilde{\mathbf{x}}_0). \tag{59}$$

In order to control the convergence rate of the iterations (35), i.e., the decay of the error $\|\tilde{\mathbf{x}}^{(k)} - \tilde{\mathbf{x}}_0\|$, we will now derive an

upper bound on the spectral norm of the block diagonal matrix \mathbf{B}^k (cf. 58).

Due to the block diagonal structure (58), we can control the norm of \mathbf{B}^k via controlling the norm of the powers of its diagonal blocks $(\mathbf{B}^{(i)})^k$ since

$$\|\mathbf{B}^k\| = \max_i \|(\mathbf{B}^{(i)})^k\|. \tag{60}$$

A pen and paper exercise reveals

$$\rho(\mathbf{B}^{(i)}) = \tilde{\beta}^{1/2} \stackrel{(37)}{=} \frac{\sqrt{U}-\sqrt{L}}{\sqrt{U}+\sqrt{L}} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}. \tag{61}$$

Combining (61) with Lemma (8) yields

$$(\mathbf{B}^{(i)})^k = \begin{pmatrix} \lambda_1^k & d \\ 0 & \lambda_2^k \end{pmatrix}, \tag{62}$$

with $|\lambda_1|, |\lambda_2| \leq \tilde{\beta}^{1/2}$ and $d \leq k(2+2\tilde{\beta}+\tilde{\alpha})\tilde{\beta}^{(k-1)/2}$. Using the shorthand $\tilde{c} = (2+2\tilde{\beta}+\tilde{\alpha})$, we can estimate the spectral norm of \mathbf{B}^k as

$$\begin{aligned} \|\mathbf{B}^k\| &\stackrel{(60)}{=} \max_i \|(\mathbf{B}^{(i)})^k\| \\ &\stackrel{(62)}{\leq} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \left(1+k\tilde{c}\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right). \end{aligned} \tag{63}$$

Combining (63) with (59),

$$\begin{aligned} \|\tilde{\mathbf{x}}^{(k)} - \tilde{\mathbf{x}}_0\| &\leq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \left(1+k\tilde{c}\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right) \|\tilde{\mathbf{x}}^{(0)} - \tilde{\mathbf{x}}_0\| \\ &\stackrel{\tilde{c} \geq 1}{\leq} 2k\tilde{c}\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|\tilde{\mathbf{x}}^{(0)} - \tilde{\mathbf{x}}_0\|. \end{aligned} \tag{64}$$

Using (38), the error bound (64) can be translated into an error bound on the HB iterates $\mathbf{x}_{\text{HB}}^{(k)}$, i.e.,

$$\begin{aligned} \|\mathbf{x}_{\text{HB}}^{(k)} - \mathbf{x}_0\| &\leq \\ 4k\tilde{c}\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k &(\|\mathbf{x}_{\text{HB}}^{(0)} - \mathbf{x}_0\| + \|\mathbf{x}_{\text{HB}}^{(-1)} - \mathbf{x}_0\|). \end{aligned}$$

9. TECHNICALITIES

We collect some elementary results from linear algebra and analysis, which are required to prove our main results.

Lemma 8. Consider a matrix $\mathbf{M} = \begin{pmatrix} a & b \\ 1 & 0 \end{pmatrix} \in \mathbb{R}^{2 \times 2}$ with spectral radius $\rho(\mathbf{M})$. Then, there is an orthonormal matrix $\mathbf{U} \in \mathbb{C}^{2 \times 2}$ such that

$$\mathbf{M}^k = \mathbf{U} \begin{pmatrix} \lambda_1^k & d \\ 0 & \lambda_2^k \end{pmatrix} \mathbf{U}^H \text{ for } k \in \mathbb{N}, \tag{65}$$

where $|\lambda_1|, |\lambda_2| \leq \rho(\mathbf{M})$ and $|d| \leq k(|a| + |b| + 1)\rho^{k-1}(\mathbf{M})$.

Proof. Consider an eigenvalue λ_1 of the matrix \mathbf{M} with normalized eigenvector $\mathbf{u} = (u_1, u_2)^H$, i.e., $\mathbf{M}\mathbf{u} = \lambda_1\mathbf{u}$ with $\|\mathbf{u}\| = 1$. According to Golub and Van Loan [11, Lemma 7.1.2], we can find a normalized vector $\mathbf{v} = (v_1, v_2)^H$, orthogonal to \mathbf{u} , such that

$$\mathbf{M} = (\mathbf{u}, \mathbf{v}) \begin{pmatrix} \lambda_1 & d \\ 0 & \lambda_2 \end{pmatrix} (\mathbf{u}, \mathbf{v})^H, \tag{66}$$

or equivalently

$$\begin{pmatrix} \lambda_1 & d \\ 0 & \lambda_2 \end{pmatrix} = (\mathbf{u}, \mathbf{v})^H \mathbf{M} (\mathbf{u}, \mathbf{v}), \tag{67}$$

with some eigenvalue λ_2 of \mathbf{M} . As can be read off (67), $d = u_1(u_2a + v_2b) + v_1u_2$ which implies (65) since $|u_1|, |u_2|, |v_1|, |v_2| \leq 1$. Based on (66), we can verify (65) by induction. \square

Lemma 9. For any $\kappa > 1$ and $k \in \mathbb{N}$,

$$\int_{\theta=0}^1 \frac{\exp(j2\pi(k-1)\theta)}{2(1-\cos(2\pi\theta))+4/(\kappa-1)} d\theta = \frac{\kappa-1}{4\sqrt{\kappa}} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k. \tag{68}$$

Proof. Let us introduce the shorthand $z = \exp(j2\pi\theta)$ and further develop the LHS of (68) as

$$\begin{aligned} &\int_{\theta=0}^1 \frac{z^{k-1}}{2(1-(z^{-1}+z)/2)+4/(\kappa-1)} d\theta \\ &= \int_{\theta=0}^1 \frac{z^k}{2(z-(1+z^2)/2)+4z/(\kappa-1)} d\theta. \end{aligned} \tag{69}$$

The denominator of the integrand in (69) can be factored as

$$2(z-(1+z^2)/2)+4z/(\kappa-1) = -(z-z_1)(z-z_2) \tag{70}$$

with

$$z_1 = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}, \text{ and } z_2 = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}. \tag{71}$$

Inserting (70) into (69),

$$\begin{aligned} &\int_{\theta=0}^1 \frac{z^k}{2(z-(1+z^2)/2)+4/(\kappa-1)} d\theta \\ &= - \int_{\theta=0}^1 \frac{z^k}{(z-z_1)(z-z_2)} d\theta \\ &= \int_{\theta=0}^1 -\frac{z^k(z_1-z_2)^{-1}}{z-z_1} + \frac{z^k(z_1-z_2)^{-1}}{z-z_2} d\theta. \end{aligned} \tag{72}$$

Since $|z_2| < 1$, we can develop the second term in (72) by using the identity [17, Section 2.7]

$$\int_{\theta=0}^1 \frac{\exp(j2\pi k\theta)}{\exp(j2\pi\theta) - \alpha} d\theta = \alpha^{k-1} \text{ for } k \in \mathbb{N}, \alpha \in \mathbb{R}, |\alpha| < 1. \tag{73}$$

Since $|z_1| > 1$, we can develop the first term in (72) by using the identity [17, Section 2.7]

$$\int_{\theta=0}^1 \frac{\exp(j2\pi k\theta)}{\exp(j2\pi\theta) - \alpha} d\theta = 0 \text{ for } k \in \mathbb{N}, \alpha \in \mathbb{R}, |\alpha| > 1. \quad (74)$$

Applying (73) and (74) to (72),

$$\int_{\theta=0}^1 \frac{z^k}{2(z - (1+z^2)/2) + 4/(\kappa - 1)} d\theta = \frac{z_2^{k-1}}{z_1 - z_2}. \quad (75)$$

Inserting (75) into (69), we arrive at

$$\int_{\theta=0}^1 \frac{\exp(j2\pi(k-1)\theta)}{2(1 - \cos(2\pi\theta)) + 4/(\kappa - 1)} d\theta = \frac{z_2^{k-1}}{z_1 - z_2}. \quad (76)$$

The proof is finished by combining (76) with the identity

$$\frac{1}{z_1 - z_2} \stackrel{(71)}{=} \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} - \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} = \frac{4\sqrt{\kappa}}{\kappa - 1}.$$

□

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and approved it for publication.

ACKNOWLEDGMENTS

This paper is a wrap-up of the lecture material created for the course Convex Optimization for Big Data over Networks, taught at Aalto University in spring 2017. The student feedback on the lectures has been a great help to develop the presentation of the contents. In particular, the detailed feedback of students Stefan Mojsilovic and Matthias Grezet on early versions of the paper is appreciated sincerely.

REFERENCES

1. Bottou L, Bousquet O. The tradeoffs of large scale learning. In: *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, BC (2008). p. 161–8.
2. Cevher V, Becker S, Schmidt M. Convex optimization for big data: scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Process Mag.* (2014) **31**:32–43. doi: 10.1109/MSP.2014.2329397
3. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge, MA: MIT Press (2016).
4. Walker HF, Ni P. Anderson acceleration for fixed-point iterations. *SIAM J Numer Anal.* (2011) **49**:1715–35. doi: 10.1137/10078356X
5. Birken P. Termination criteria for inexact fixed-point schemes. *Num Lin Alg App.* (2015) **22**:702–16. doi: 10.1002/nla.1982
6. Alfeld P. Fixed point iteration with inexact function values. *Math Comput.* (1982) **38**:87–98. doi: 10.1090/S0025-5718-1982-0637288-5
7. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge: Cambridge University Press (2004). doi: 10.1017/CBO9780511804441
8. Bishop CM. *Pattern Recognition and Machine Learning*. New York, NY: Springer (2006).
9. Bauschke HH, Combettes PL. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY: Springer (2010).
10. Rudin W. *Principles of Mathematical Analysis, 3rd Edn*. New York, NY: McGraw-Hill (1976).
11. Golub GH, Van Loan CF. *Matrix Computations, 3rd Edn*. Baltimore, MD: Johns Hopkins University Press (1996).
12. Giannakis GB, Slavakis K, Mateos G. Signal processing for big data. In: *Tutorial at EUSIPCO*. Lisbon (2014).
13. Nesterov Y. *Introductory Lectures on Convex Optimization. vol. 87 of Applied Optimization*. Boston, MA: Kluwer Academic Publishers (2004).
14. Polyak BT. Some methods of speeding up the convergence of iteration methods. *USSR Comput Math Math Phys.* (1964) **4**:1–17.
15. Ghadimi E, Shames I, Johansson M. Multi-step gradient methods for networked optimization. *IEEE Trans Signal Process.* (2013) **61**: 5417–29.
16. Gray RM. *Toeplitz and Circulant Matrices: A review. vol. 2 of Foundations and Trends in Communications and Information Theory*. Boston, MA (2006).
17. Oppenheim AV, Schaffer RW, Buck JR. *Discrete-Time Signal Processing, 2nd Edn*. Englewood Cliffs, NJ: Prentice Hall (1998).

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Jung. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.