SIAM J. MATRIX ANAL. APPL.
Vol. 43, No. 1, pp. 479–511

SEMI-INFINITE LINEAR REGRESSION AND ITS APPLICATIONS*

PAZ FINK SHUSTIN† AND HAIM AVRON†

Abstract. Finite linear least squares is one of the core problems of numerical linear algebra, with countless applications across science and engineering. Consequently, there is a rich and ongoing literature on algorithms for solving linear least squares problems. In this paper, we explore a variant in which the system's matrix has one infinite dimension (i.e., it is a quasimatrix). We call such problems semi-infinite linear regression problems. As we show, the semi-infinite case arises in several applications, such as supervised learning and function approximation, and allows for novel interpretations of existing algorithms. We explore semi-infinite linear regression rigorously and algorithmically. To that end, we give a formal framework for working with quasimatrices, and generalize several algorithms designed for the finite problem to the infinite case. Finally, we suggest the use of various sampling methods for obtaining an approximate solution.

Key words. semi-infnite linear regression, least squares, quasimatrix, chebfun, sampling

AMS subject classifications. 68W20, 68W25, 65F05, 65F10

DOI. 10.1137/21M1411950

1. Introduction.

Consider the classical linear least squares problem: given an $m \times n$ matrix $\mathbf{A}$, and a vector $\mathbf{b}$, we seek to compute

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \,. \tag{1.1}$$

The problem of solving (1.1) is one of the most fundamental problems of numerical linear algebra, and it has countless applications throughout scientific computing and data science. As such, there is a rich literature on algorithms for solving (1.1). In particular, there are algorithms that: compute an approximate solution [16], compute a near exact solution [38, 5, 28], are designed for the overdetermined case [5], designed for the underdetermined case [28], consider also the presence of a regularizer [2, 36], and replace the two-norm with some other norm [10]. The previous list is far from exhaustive. Finding efficient algorithms for solving (1.1) is an active research field.

In this paper, we explore a variant of (1.1) in which $\mathbf{A}$ is no longer a matrix, but a *quasimatrix*, that is a matrix in which one of the two dimensions is infinite (while the other dimension is finite). We call such problems *semi-infinite linear regression.* As we show, the semi-infinite case arises in several applications, such as supervised learning and function approximation, and allows for novel interpretations of existing algorithms. In contrast to the rich literature on the finite (i.e., matrix) variant of (1.1), the semi-infinite case has hardly been treated in the literature (the only exception we are aware of is [44]).

The goal of this paper is to explore semi-infinite linear regression rigorously and algorithmically. To that end, we first define the notion of quasimatrix formally, and give the needed framework for working with quasimatrices, both mathematically and algorithmically. The use of the term "quasimatrix" as a matrix which has columns

*Received by the editors April 13, 2021; accepted for publication (in revised form) by Y. Nakatsukasa October 29, 2021; published electronically March 15, 2022.
https://doi.org/10.1137/21M1411950
Funding: The work of the authors was supported by the BSF grant 2017698.
†Department of Applied Mathematics, Tel-Aviv University, Tel-Aviv, Israel (pazfink@mail.tau.ac.il, haimav@tauex.tau.ac.il).

or rows that are functions first appears in the literature in [41, 44], but has so far been informal. Once we have the mathematical foundations, we define semi-infinite regression formally, and discuss applications.

We then proceed to proposing algorithms for solving semi-infinite linear regression problems. First, we discuss direct methods, which factorize a quasimatrix $\mathbf{A}$ into a product of quasimatrices. Even though most of the algorithms we present are straightforward generalizations of classical methods for finite linear least squares problems, we also show how in some cases the use of quasimatrix operations can be sidestepped.

Next, we discuss iterative methods. It is possible to devise a wide array of iterative methods for solving semi-infinite linear regression by generalizing iterative methods for the finite case. However, for conciseness we show a representative algorithm from each of the two approaches: Krylov subspace methods and stochastic optimization. For Krylov methods, we show how LSMR [18] can be generalized to solve semi-infinite linear regression. For stochastic optimization, we adapt a method based on stochastic variance reduce gradient (SVRG) [23, 48]. Unlike LSMR, the adaptation of SVRG is less straightforward, and requires generalizing it to work with integrable sums.

Motivated by recent literature on randomized numerical linear algebra (RandNLA), we discuss algorithms that are based on *sampling* the semi-infinite linear regression problem. We discuss both randomized sampling and deterministic sampling. For randomized sampling, we discuss a sampling technique which we term as *natural sampling*. Conceptually, natural sampling is analogous to uniformly sampling rows or columns from $\mathbf{A}$ when dealing with finite linear least squares problems, though for semi-infinite linear regression problems, uniform sampling is not always well defined. It is well known from the RandNLA literature that it is better to sample based on the so-called *leverage scores*. For quasimatrices, the analogous operation is sampling using the *leverage function*, a generalization of leverage scores. Similar results have been shown before for restricted cases [3, 12, 4]. We also propose a deterministic sampling method based on Gauss–Legendre quadrature. Interestingly, this method does not have any finite analogue.

Most of the methods we present are based on existing algorithms for the finite linear least squares case to some degree (the only exception is the quadrature sampling). The main contribution of the paper is in the rigorous and systematic treatment of the subject. We hope that our systematic treatment of semi-infinite linear regression will spur additional interest and research on this problem.

2. Quasimatrices.

The term quasimatrix appears in the literature as a name for matrices in which one of their dimensions is infinite [41]. The term was later adapted by the chebfun library **[8]**,[1] and a variety of papers related to that package, and other literature, use it, e.g., [44, 31, 33, 25, 22]. In previous literature, quasimatrices were defined and treated informally as matrices which have columns or rows that are functions. For our purposes, a more formal treatment is needed, and we provide it here. Our approach is in many ways similar to the one taken by [15] to define fundamental notions such as rank and basis as ones derived from linear maps in which the domain is finite dimensional vectors.

Notations and basic terminology. For an integer n, we denote $[n] = \{1, \ldots, n\}$. Scalars are denoted by lowercase Greek letters or by $x, y, \ldots$. Given two Banach spaces, $\mathcal{X}$ and $\mathcal{Y}$, we denote by $\mathcal{B}(\mathcal{X}, \mathcal{Y})$ the Banach space of bounded linear operators

[1] See http://www.chebfun.org/docs/guide/guide06.html

from $\mathcal{X}$ to $\mathcal{Y}$. Given a Banach space $\mathcal{X}$, $\mathcal{X}^*$ denotes the *topological* dual space of $\mathcal{X}$, i.e., the space of continuous linear functionals from $\mathcal{X}$ to $\mathbb{R}$ or $\mathbb{C}$. So, $\mathcal{X}^* = \mathcal{B}(\mathcal{X}, \mathbb{R})$ or $\mathcal{X}^* = \mathcal{B}(\mathcal{X}, \mathbb{C})$ (depending on the context). Vectors are denoted by $\mathbf{x}, \mathbf{y}, \ldots$ and considered as column vectors (unless otherwise stated), and matrices are denoted by $\mathbf{A}, \mathbf{B}, \ldots$ or uppercase Greek letters. Quasimatrices are denoted by $\mathbf{x}, \mathbf{y}, \ldots$ if they are lying in a Hilbert space and otherwise by $\mathbf{A}, \mathbf{B}, \ldots$ (defined later in this paper). For a vector $\mathbf{x}$ or a matrix $\mathbf{A}$, the notation $\mathbf{x}^*$ or $\mathbf{A}^*$ denotes the Hermitian transpose. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$. We use $\mathbf{e}_1, \mathbf{e}_2, \ldots$ to denote the unit vectors and assume that their dimensions are clear from the context. All vectors are considered as column vectors, which can be of finite or infinite dimension.

We use $L_2(\Omega, d\mu)$ to denote the space of complex-valued square integrable functions over Ω with respect to the measure μ, i.e., the inner product in $L_2(\Omega, d\mu)$ is

$$(\mathbf{f}, \mathbf{g})_{L_2(\Omega, d\mu)} \coloneqq \int_\Omega \overline{\mathbf{f}(\boldsymbol{\eta})} \mathbf{g}(\boldsymbol{\eta}) d\mu(\boldsymbol{\eta}) \,.$$

2.1. Quasimatrix algebra. A matrix is a mapping from two indexes to a scalar. Alternatively, a matrix can be viewed as mapping from a finite index set to finite dimensional vectors, where the index set is either the column index or the row index. For a quasimatrix we drop the condition that the mapped vectors are finite dimensional, and instead require them to be from a Hilbert space.

DEFINITION 1. *Let n be a positive integer and let $\mathcal{H}$ be a Hilbert space over $\mathbb{R}$ or $\mathbb{C}$. A* tall quasimatrix *is a mapping from $[n]$ to $\mathcal{H}$. A* wide quasimatrix *is a mapping from $[n]$ to $\mathcal{H}^*$.*

We generally omit the adjectives tall and wide when the text refers to both types, or when the relevant type is clear from the context. We say the size of a tall quasimatrix is $m \times n$ if m is the dimension of the Hilbert space $\mathcal{H}$. We generally write $\infty \times n$ if $\mathcal{H}$ has infinite dimension. A similar notion of size applies for wide quasimatrices. To avoid clutter, henceforth we assume that $\mathcal{H}$ is defined over $\mathbb{C}$, and leave for the reader to deduce how some of the description is somewhat simplified for the real case.

For a tall quasimatrix $\mathbf{A}$, we refer to the values at the various indexes as the *columns* of the quasimatrix. We use the following notation

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_n \end{bmatrix}$$

to denote the tall quasimatrix $\mathbf{A}$ which maps $j \in [n]$ to $\mathbf{a}_j$, where $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathcal{H}$.

Let $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathcal{H}$, and $\mathbf{b}_1^*, \ldots, \mathbf{b}_m^* \in \mathcal{H}^*$ their adjoints. We denote

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1^* \\ \vdots \\ \mathbf{b}_m^* \end{bmatrix}$$

for the wide quasimatrix $\mathbf{B}$ which maps $j \in [m]$ to $\mathbf{b}_j^*$. If a wide quasimatrix $\mathbf{B}$ maps j to $\mathbf{x} \in \mathcal{H}^*$, the Riesz representation theorem implies that there exists a $\mathbf{b} \in \mathcal{H}$ such that $\mathbf{x} = \mathbf{b}^*$, so every wide quasimatrix can be written in this way. For a wide quasimatrix $\mathbf{B}$, we refer to the adjoints of the values at the indexes as the *rows* of the quasimatrix. Note that both the columns of a tall quasimatrix, and the rows of a wide quasimatrix, are vectors in $\mathcal{H}$.

In the rest of section 2, $\mathbf{A}$ is a tall quasimatrix with columns $\mathbf{a}_1, \ldots, \mathbf{a}_n$, and $\mathbf{B}$ is a wide quasimatrix with rows $\mathbf{b}_1^*, \ldots, \mathbf{b}_m^*$.

The conjugate transpose of a tall quasimatrix $\mathbf{A}$ is the wide quasimatrix $\mathbf{B}$ whose coordinates are the adjoints of the corresponding coordinates of $\mathbf{A}$. The conjugate transpose of a wide quasimatrix $\mathbf{B}$ is the tall quasimatrix $\mathbf{A}$ whose coordinates are the adjoints of the corresponding coordinates of $\mathbf{B}$, which corresponds to removing the adjoints. These definitions are consistent with the notations above.

Given a tall quasimatrix $\mathbf{A}$ and a vector $\mathbf{x} \in \mathbb{C}^n$, we define the product of $\mathbf{A}$ and $\mathbf{x}$ as $\mathbf{A}\mathbf{x} = \sum_{j=1}^{n} x_j \mathbf{a}_j$. This definition naturally extends to the definition of a product $\mathbf{A}\mathbf{X}$, where $\mathbf{X} \in \mathbb{C}^{n\times k}$ is the tall quasimatrix whose columns are $\mathbf{A}\mathbf{x}_1, \dots \mathbf{A}\mathbf{x}_k$ (where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are the columns of $\mathbf{X}$). Given a wide quasimatrix $\mathbf{B}$ and an $\mathbf{x} \in \mathcal{H}$ we define

$$\mathbf{B}\mathbf{x} = \begin{bmatrix} \mathbf{b}_1^* \mathbf{x} \\ \vdots \\ \mathbf{b}_m^* \mathbf{x} \end{bmatrix}.$$

This definition naturally extends to the definition of a product of $\mathbf{B}$ and $\mathbf{A}$ as the $m \times n$ matrix whose columns are $\mathbf{B}\mathbf{a}_1, \dots \mathbf{B}\mathbf{a}_n$.

We now define the left product. Given a tall quasimatrix $\mathbf{A}$ and a vector $\mathbf{x} \in \mathcal{H}$, we define $\mathbf{x}^*\mathbf{A} = (\mathbf{A}^*\mathbf{x})^*$. This definition is consistent with viewing $\mathbf{x}^*$ as a $1 \times \infty$ quasimatrix, and the previous definition of $\mathbf{B}\mathbf{A}$. Similarly, $\mathbf{x}^*\mathbf{B} = (\mathbf{B}^*\mathbf{x})^*$ for $\mathbf{x} \in \mathbb{R}^m$. These definitions naturally extend to the left product of a matrix and a quasimatrix. The product algebra we have defined over matrices and quasimatrices is associative (but, of course, not commutative).

It is well known that an $m \times n$ complex matrix is, in fact, a coordinate representation of a linear transformation from $\mathbb{C}^n$ to $\mathbb{C}^m$ under the standard basis, and that choosing a different basis leads to a different matrix (coordinate) representation. Similarly to finite matrices, quasimatrices define bounded linear transformations between finite dimensional Euclidean spaces and $\mathcal{H}$. Concretely, for a tall quasimatrix $\mathbf{A}$, we can define the transformation $\mathcal{A} : \mathbf{x} \in \mathbb{C}^n \mapsto \mathbf{A}\mathbf{x} \in \mathcal{H}$. Conversely, given a bounded linear transformation $\mathcal{A} : \mathbb{C}^n \to \mathcal{H}$, for the tall quasimatrix $\mathbf{A} = [\mathcal{A}\mathbf{e}_1 \cdots \mathcal{A}\mathbf{e}_n]$, the corresponding linear transformation is $\mathcal{A}$. Thus, we can abuse notation and use $\mathbf{A}$ to denote both the quasimatrix and the linear transformation it defines. Likewise, every wide quasimatrix $\mathbf{B}$ defines a bounded linear transformation $\mathbf{B} : \mathbf{x} \in \mathcal{H} \mapsto \mathbf{B}\mathbf{x} \in \mathbb{R}^m$, likewise abusing notation (this is well justified by the Riesz representation theorem). Taking the conjugate transpose of a tall or wide quasimatrix produces a quasimatrix which represents the adjoint of the transformation defined by the original quasimatrix, thus our notation is consistent with that operation as well.

Similarly to the finite dimensional matrix case, the extended product algebra over matrices and quasimatrices is consistent with composition in the transformation spaces. That is, given two matrices or quasimatrices $\mathbf{X}$ and $\mathbf{Y}$, with sizes or underlying Hilbert space such that the product $\mathbf{X}\mathbf{Y}$ is defined, the transformation defined by $\mathbf{X}\mathbf{Y}$ (which is a matrix or a quasimatrix) is exactly the same as the transformation obtained by $\mathbf{X}$ composed with $\mathbf{Y}$. However, we remark that if $\mathbf{X}$ is a tall quasimatrix and $\mathbf{Y}$ is a wide quasimatrix, then we can certainly define the transformation from $\mathcal{H}$ to $\mathcal{H}$ by composing $\mathbf{X}$ with $\mathbf{Y}$, but that transformation is not represented by a quasimatrix.

We have defined columns of a tall quasimatrix and rows of a wide quasimatrix. Defining the rows of a tall quasimatrix or the columns of a wide quasimatrix is less straightforward, and in some senses impossible. Intuitively, if $\mathcal{H}$ is a space of functions over some index set $\Omega \subseteq \mathbb{R}^n$, then row $\boldsymbol{\eta}$ of a tall quasimatrix is simply the evaluation of the columns at a location $\boldsymbol{\eta} \in \Omega$ (and likewise for wide quasimatrices). However,

requiring $\mathcal{H}$ to be a space of functions is somewhat restrictive. In particular, note that L_2 spaces are, strictly speaking, spaces of equivalence classes of functions, and pointwise evaluation is not really well defined.

However, note that if $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), then we can define the notion of rows (or columns) of a tall (wide) quasimatrix in a way that is consistent with the use of identity vectors in finite matrices. If $\mathcal{H}$ is an RKHS over $\mathbb{R}^d$ then for every $\boldsymbol{\eta} \in \mathbb{R}^d$ the pointwise evaluation $\mathbf{f} \in \mathcal{H} \mapsto \mathbf{f}(\boldsymbol{\eta})$ is a bounded linear transformation. Thus, there is a unique $\mathbf{e}_{\boldsymbol{\eta}} \in \mathcal{H}$ such that for every $\mathbf{f} \in \mathcal{H}$ we have $\mathbf{e}_{\boldsymbol{\eta}}^* \mathbf{f} = \mathbf{f}(\boldsymbol{\eta})$. Thus, we define row $\boldsymbol{\eta}$ of an $\infty \times n$ quasimatrix $\mathbf{A}$ over an RKHS $\mathcal{H}$ as $\mathbf{e}_{\boldsymbol{\eta}}^* \mathbf{A} \in \mathbb{R}^{1 \times n}$. For an $m \times \infty$ quasimatrix $\mathbf{B}$, column $\boldsymbol{\eta}$ is defined as $\mathbf{B}\mathbf{e}_{\boldsymbol{\eta}}$.

Many notions related to matrices can be easily generalized to quasimatrices. For example, the rank is the dimension of space spanned by the columns (rows) of a tall (wide) quasimatrix. Obviously, the rank cannot be larger then the size of the finite dimension, and similar properties of matrix rank can be shown for quasimatrix rank.

2.2. Numerical computing with quasimatrices. In subsequent sections, we describe algorithms that "operate" on quasimatrices and functions. Such algorithms assume a model of computation in which functions are primitive types, and certain operations between functions are allowed (e.g., taking the integral of a function). Of course, such computations are not supported in hardware by general purpose computing machines. However, the software package ***chebfun***[2] does provide this abstraction in software [8]. Thus, we refer to this model of computation as the *chebfun model*.

In numerical computing, it is customary to regard floating-point operations (FLOPs) as the costly operations, and thus runtime analysis focuses on counting FLOPs. In the chebfun model, arguably the costly operations are operations on functions. Thus, when analyzing algorithms in the chebfun model we count function operations (FUNOPs).

Specifically, we assume the following operations are supported, each costing one FUNOP: multiplying a function by a scalar, adding or subtracting two functions, evaluating a function at a point, and taking the inner product of two functions,

Of course, wherever possible we attempt to describe algorithms that operate under the standard model of computation (no FUNOPs). Such algorithms usually require additional assumptions on the quasimatrices involved.

2.3. Coordinate representation of quasimatrices over L_2 spaces. As explained in the previous sections, the rows of a tall quasimatrix or the columns of a quasimatrix cannot be defined for quasimatrices over L_2 spaces. For most algorithms we describe that use the chebfun model this is not an issue. However, when we discuss algorithms that perform sampling and operate in the standard model, we need access to rows/columns so they can be sampled. In such cases we need to assume that the algorithm, when applied to quasimatrices over L_2, has additional information in the form of a *coordinate representation* of the quasimatrix it operates on.

DEFINITION 2. *Suppose* $\mathbf{A}$ *is a quasimatrix over* $L_2(\Omega, d\mu)$ *whose finite dimension is* n. *A coordinate representation of* $\mathbf{A}$ *is a function* $\mathbf{z} : \Omega \to \mathbb{C}^n$ *such that*

$$\int_\Omega \|\mathbf{z}(\boldsymbol{\eta})\|_2^2 d\mu(\boldsymbol{\eta}) < \infty$$

and

[2] http://www.chebfun.org/

1. *if* $\mathbf{A}$ *is tall, for every* $\mathbf{x} \in \mathbb{C}^n$,

$$\mathbf{A}\mathbf{x} = \sum_{i=1}^{n} x_i \overline{\mathbf{z}(\cdot)_i},$$

where the above equality should be interpreted in the $L_2(\Omega, d\mu)$ *sense and* $\mathbf{z}(\boldsymbol{\eta})_i$ *is coordinate* i *of* $\mathbf{z}(\boldsymbol{\eta})$ *(for* $\boldsymbol{\eta} \in \Omega$*);*

2. *if* $\mathbf{A}$ *is wide, for every* $\mathbf{x} \in L_2(\Omega, d\mu)$ *and* $j \in [n]$,

$$\mathbf{e}_j^* \mathbf{A}\mathbf{x} = (\overline{\mathbf{z}(\cdot)_j}, \mathbf{x})_{L_2(\Omega, d\mu)}.$$

The definition implies that if $\mathbf{z}$ is a coordinate representation of $\mathbf{A}$ then it is also a coordinate representation of $\mathbf{A}^*$. Essentially, for a tall quasimatrix with a coordinate representation $\mathbf{z}$, column i is $\overline{\mathbf{z}(\cdot)_i}$, and for a wide quasimatrix with a coordinate representation $\mathbf{z}$, row i is $\overline{\mathbf{z}(\cdot)_i}^*$. If the quasimatrix is defined over $L_2(\Omega, d\mu)$, we say that Ω is the *index set* of the infinite dimension. We now say that for an index $\boldsymbol{\eta} \in \Omega$, row $\boldsymbol{\eta}$ of a tall quasimatrix with coordinate representation $\mathbf{z}$ is $\mathbf{z}(\boldsymbol{\eta})^*$, and column $\boldsymbol{\eta}$ of a wide quasimatrix with coordinate representation $\mathbf{z}$ is $\mathbf{z}(\boldsymbol{\eta})$.

Note that the definition also implies the following. For $\mathbf{A}$ we have

$$\mathbf{A}^*\mathbf{A} = \begin{bmatrix} (\overline{\mathbf{z}_\mathbf{A}(\cdot)_1}, \mathbf{z}_\mathbf{A}(\cdot)_1)_{L_2(\Omega,d\mu)} & \cdots & (\overline{\mathbf{z}_\mathbf{A}(\cdot)_1}, \mathbf{z}_\mathbf{A}(\cdot)_n)_{L_2(\Omega,d\mu)} \\ \vdots & \cdots & \vdots \\ (\overline{\mathbf{z}_\mathbf{A}(\cdot)_n}, \mathbf{z}_\mathbf{A}(\cdot)_1)_{L_2(\Omega,d\mu)} & \cdots & (\overline{\mathbf{z}_\mathbf{A}(\cdot)_n}, \mathbf{z}_\mathbf{A}(\cdot)_n)_{L_2(\Omega,d\mu)} \end{bmatrix} = \int_\Omega \mathbf{z}_\mathbf{A}(\boldsymbol{\eta}) \mathbf{z}_\mathbf{A}(\boldsymbol{\eta})^* d\mu(\boldsymbol{\eta})$$

and similarly for the product $\mathbf{B}\mathbf{B}^*$.

2.4. Quasimatrix factorizations. Matrix factorizations such as QR and SVD are used to define direct methods for solving linear regression problems (and more generally, in matrix analysis at large). Thus, it is nosurprising that they can be used to solve semi-infinite linear regression problems as well, as was already noted in [44]. Various quasimatrix factorizations are already mentioned in [45, 8, 44], and are further developed in [43]. They can be formulated in our formal quasimatrix framework (previous aforementioned works used quasimatrices in an informal manner). In Table 1 we detail a few key quasimatrix factorizations of a tall quasimatrix $\mathbf{A}$. Factorizations for a wide quasimatrix $\mathbf{B}$ can be obtained by taking the conjugate transpose of a factorization of $\mathbf{B}^*$. We also detail in Table 1 the FUNOPs cost of forming the various quasimatrix factorizations.

Using the SVD factorization, we define the condition number of a quasimatrix to be $\kappa(\mathbf{A}) \coloneqq \sigma_1/\sigma_n$, where $\Sigma = \mathbf{diag}\,(\sigma_1, \ldots, \sigma_n)$ in the SVD factorization.

3. Semi-infinite linear regression: Problem statement and examples. In this paper, we are mainly concerned with the solution of regularized linear least squares regression problems with quasimatrices. We specifically consider ridge regularization (also called Tikhonov regularization). We call such problems *semi-infinite linear regression (SILR)*. Both the overdetermined case and the underdetermined case are considered. In the overdetermined case, we are given an $\infty \times n$ quasimatrix $\mathbf{A}$ over $\mathcal{H}$, a target $\mathbf{b} \in \mathcal{H}$, and a regularization parameter $\lambda \geq 0$. The goal of SILR is to find $\mathbf{x} \in \mathbb{C}^n$ such that

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\mathcal{H}^2 + \lambda \|\mathbf{x}\|_2^2 = \mathit{minimum}. \tag{3.1}$$

In the underdetermined case, we are given an $n \times \infty$ quasimatrix $\mathbf{A}$ over $\mathcal{H}$, a target $\mathbf{b} \in \mathbb{C}^n$, and a regularization parameter $\lambda \geq 0$. Our goal is to find an $\mathbf{x} \in \mathcal{H}$ such that

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_\mathcal{H}^2 = \mathit{minimum}. \tag{3.2}$$

TABLE 1
Factorizations of a tall Quasimatrix $\mathbf{A}$ *with* n *columns over* $\mathcal{H}$.

	Factorization form	Reference	FUNOPs in the chebfun model
Reduce QR using Gram–Schmidt	$\mathbf{A} = \mathbf{QR}$ $\mathbf{Q} \in \mathbb{R}^{\infty\times n}$, $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}_n$ $\mathbf{R} \in \mathbb{R}^{n\times n}$ upper diagonal $R_{ij} = \begin{cases} (\mathbf{a}_i, \mathbf{a}_j)_{\mathcal{H}} & j \geq i \\ 0 & j < i \end{cases}$	-	$n(n+1)$
Reduce QR using Householder triangulation	$\mathbf{A} = \mathbf{QR}$ $\mathbf{Q} = \mathbf{H}_1\mathbf{H}_2\cdots\mathbf{H}_n\mathbf{ES}$ $\mathbf{H}_1, \ldots, \mathbf{H}_n \in \mathcal{B}(\mathcal{H}, \mathcal{H})$ Householder reflectors $\mathbf{E} = [\ \mathbf{e}_1^{\mathcal{H}} \quad \mathbf{e}_2^{\mathcal{H}} \quad \ldots \quad \mathbf{e}_n^{\mathcal{H}}\ ] \in \mathbb{R}^{\infty\times n}$ $\mathbf{e}_1^{\mathcal{H}}, \mathbf{e}_2^{\mathcal{H}}, \ldots$ predetermined sequence of orthonormal vectors in $\mathcal{H}$. $\mathbf{S} \in \mathbb{R}^{n\times n}$ diagonal sign matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ upper diagonal $R_{ij} = (\mathbf{e}_i^{\mathcal{H}}, \mathbf{a}_j)_{\mathcal{H}}$	[44]	$3n(3n-1)/2+6n$
SVD	$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$ $\mathbf{U} \in \mathbb{R}^{\infty\times n}$, $\mathbf{U}^*\mathbf{U} = \mathbf{I}_n$ $\mathbf{V} \in \mathbb{R}^{n\times n}$, $\mathbf{V}^*\mathbf{V} = \mathbf{I}_n$ $\Sigma \in \mathbb{R}^{n\times n}$ nonnegative diagonal matrix	[8, 7, 26, 24, 25, 44]	QR cost + $(2n-1)n$

For simplicity, in both cases we either assume that $\mathbf{A}$ has full rank or that $\lambda > 0$. This makes the solution unique, and we always denote it by $\mathbf{x}^\star$.

We now give examples in which SILR is involved. We focus on cases where SILR is solved approximately by sampling the quasimatrix in order to turn the problem into regular finite linear regression problem.

3.1. Least squares approximation of a function. Suppose we are given a function $\mathbf{f} \in \mathcal{H} = L_2([-1,1], d\lambda)$ (or any other Hilbert space), and a finite dimensional subspace $\mathcal{V}$ of $\mathcal{H}$ (e.g., the space of polynomials up to a certain degree). We want to find the optimal approximation (in the $\mathcal{H}$ sense) of $\mathbf{f}$ in $\mathcal{V}$, which we denote by $f_{\mathcal{V}}$. Denote by n the dimension of $\mathcal{V}$, and let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be a basis for $\mathcal{V}$. Define the $\infty \times n$ quasimatrix $\mathbf{A} = [\ \mathbf{v}_1 \quad \cdots \quad \mathbf{v}_n\ ]$. Then, $f_{\mathcal{V}} = \mathbf{A}\mathbf{x}^\star$, where

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{f}\|_{\mathcal{H}}^2 . \tag{3.3}$$

A closely related problem is the problem of reconstructing an unknown function f on a domain $\mathcal{X}$ from samples at randomly chosen points [11]. In this problem setting we are given $y_i = f(\mathbf{x}_i) + \epsilon_i$ at m given data points $\mathbf{x}_1, \ldots, \mathbf{x}_m$ sampled independent and identically distributed (i.i.d) from some distribution ρ on $\mathcal{X}$ (we do not assume we have an explicit formula for ρ, or that we can produce additional samples; we only assume such a distribution exists). The scalars $\epsilon_1, \ldots, \epsilon_n$ are noise terms, which might be zero in the noiseless case. We can connect this problem to (3.3) in the following way, originally discussed in [11]. We set up a finite dimensional subspace $\mathcal{V}$ and try to approximate $f_{\mathcal{V}}$ via sampling. Specifically, let $\mathbf{A}_s \in \mathbb{R}^{m\times n}$ be a "rows sample" of the quasimatrix $\mathbf{A}$, i.e., defined by $(\mathbf{A}_s)_{ij} = \mathbf{v}_j(\mathbf{x}_i)$, and let

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}_s\mathbf{x} - \mathbf{y}\|_2^2 . \tag{3.4}$$

The approximation is $\tilde{f_{\mathcal{V}}} = \mathbf{A}\tilde{\mathbf{x}}$. In [11] the authors provide a criterion on s that describes the needed number of samples to ensure that the least squares method is

stable and that its accuracy is comparable to the best approximation error of $\mathbf{f}$ by elements from $\mathcal{V}$. Note that (3.4) is a sampled version of (3.3). We discuss solving SILR problems using sampling in section 6.

3.2. Kernel ridge regression. Kernel ridge regression (KRR) is an important method for supervised learning. Recall the problem of supervised learning: given training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is an input domain and $\mathcal{Y} \subseteq \mathbb{R}$ is an output domain, we wish to infer some functional dependency between the outputs and the inputs [14]. In *KRR,* one starts with a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The kernel is associated with an RKHS $\mathcal{H}_k$ which is the completion of the function space

$$\left\{ \sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i, \cdot) \,|\, \mathbf{x}_i \in \mathcal{X}, \alpha_i \in \mathbb{R}, m \in \mathbb{Z}_+ \right\}$$

equipped with the inner product

$$\left(\sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i, \cdot), \sum_{j=1}^{n} \beta_j k(\mathbf{x}_j, \cdot) \right) \mathcal{H}_k = \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) \,.$$

For some $\lambda > 0$, the KRR estimator is

$$\mathbf{f}^\star = \arg\min_{\mathbf{f} \in \mathcal{H}_k} \sum_{i=1}^{n} (\mathbf{f}(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{f}\|_{\mathcal{H}_k}^2 \,. \tag{3.5}$$

The celebrated representer theorem [39] guarantees that $\mathbf{f}^\star$ can be written as

$$\mathbf{f}^\star(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i^\star k(\mathbf{x}_i, \mathbf{x}) \tag{3.6}$$

for some $\alpha_1^\star, \ldots, \alpha_n^\star \in \mathbb{R}$ (note that $k(\mathbf{x}_i, \cdot) \in \mathcal{H}_k$ so $\sum_{i=1}^n \alpha_i^\star k(\mathbf{x}_i, \cdot) \in \mathcal{H}_k$). Simple linear algebra now implies that we can find $\alpha_1, \ldots, \alpha_n$ by solving the linear system

$$(\mathbf{K} + \lambda \mathbf{I}_n)\boldsymbol{\alpha} = \mathbf{y}, \tag{3.7}$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the matrix defined by $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{y} = [y_1 \cdots y_n]^{\mathrm{T}} \in \mathbb{R}^n$.

3.2.1. KRR as SILR. We now show how (3.5) can be written as an SILR problem. Define the $n \times \infty$ quasimatrix $\mathbf{A}$ over $\mathcal{H}_k$:

$$\mathbf{A} = \begin{bmatrix} k(\cdot, \mathbf{x}_1)^* \\ \vdots \\ k(\cdot, \mathbf{x}_n)^* \end{bmatrix} .$$

Due to the reproducing property of RKHS, $(\mathbf{f}, k(\cdot, \mathbf{x}_j))_{\mathcal{H}_k} = \mathbf{f}(\mathbf{x}_j) = (\mathbf{A}\mathbf{f})_j$ and we have

$$\mathbf{f}^\star = \arg\min_{\mathbf{f} \in \mathcal{H}_k} \|\mathbf{A}\mathbf{f} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{f}\|_{\mathcal{H}_k}^2 \,. \tag{3.8}$$

Thus, the KRR estimator is the solution to an underdetermined SILR problem. In fact, using (3.7) to solve (3.8) is an instance of a direct method for solving underdetermined SILR problems; see section 4.2.

In (3.8), the quasimatrix $\mathbf{A}$ is defined over an RKHS. In certain cases, the problem can be cast as an SILR problem with quasimatrices defined over an L_2 space, and this leads to approximation methods based on sampling. The following is based on the seminal work of Rahimi and Recht on random Fourier features [37]. Suppose that k is a shift-invariant positive definite function, that is, $k(\mathbf{x},\mathbf{z}) = k(\mathbf{x}-\mathbf{z})$ for some positive definite $k(\cdot)$ (note that we abuse notation in denoting by k both the kernel and the positive definite function that defines it). Further assume that k is normalized in the sense that $k(\mathbf{x},\mathbf{x}) = 1$. According to Bochner's theorem, there exists a probability measure μ such that

$$k(\mathbf{x},\mathbf{z}) = k(\mathbf{x}-\mathbf{z}) = \int_{\mathbb{R}^d} e^{-2\pi i(\mathbf{x}-\mathbf{z})^{\mathrm{T}}\boldsymbol{\eta}} d\mu(\boldsymbol{\eta})\,.$$

Define the function $\varphi : \mathcal{X} \times \mathbb{R}^d \to \mathbb{C}$:

$$\varphi(\mathbf{x},\boldsymbol{\eta}) = e^{2\pi i \mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}}\,.$$

For fixed $\mathbf{x},\mathbf{z} \in \mathcal{X}$ we have

$$(\varphi(\mathbf{x},\cdot),\varphi(\mathbf{z},\cdot))_{L_2(\mathbb{R}^d,d\mu)} = \int_{\mathbb{R}^d} e^{-2\pi i(\mathbf{x}-\mathbf{z})^{\mathrm{T}}\boldsymbol{\eta}} d\mu(\boldsymbol{\eta}) = \int_{\mathbb{R}^d} e^{-2\pi i(\mathbf{x}-\mathbf{z})^{\mathrm{T}}\boldsymbol{\eta}} p(\boldsymbol{\eta}) d\boldsymbol{\eta} = k(\mathbf{x},\mathbf{z})$$

so $\varphi(\mathbf{x},\cdot) \in L_2(\mathbb{R}^d, d\mu)$ for every $\mathbf{x} \in \mathcal{X}$. Let us now define the $n \times \infty$ quasimatrix $\mathbf{B}$ over $L_2(\mathbb{R}^d, d\mu)$:

$$\mathbf{B} = \begin{bmatrix} \varphi(\mathbf{x}_1,\cdot)^* \\ \vdots \\ \varphi(\mathbf{x}_n,\cdot)^* \end{bmatrix}. \tag{3.9}$$

LEMMA 3. *Assuming that* $\mathbf{K}$ *is full rank or* $\lambda > 0$*, the following holds,*

$$\mathbf{f}^\star(\mathbf{x}) = (\varphi(\mathbf{x},\cdot),\mathbf{w}^\star)_{L_2(\mathbb{R}^d,d\mu)},$$

where

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}\in L_2(\mathbb{R}^d,d\mu)} \|\mathbf{B}\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|^2_{L_2(\mathbb{R}^d,d\mu)}\,. \tag{3.10}$$

Proof. Let

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}\in L_2(\mathbb{R}^d,d\mu)} \|\mathbf{B}\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|^2_{L_2(\mathbb{R}^d,d\mu)}\,.$$

Since $\mathbf{range}\,(\mathbf{B}^*)$ is a a closed linear subspace of $\mathcal{H}$, there exists $\mathbf{v}^\star \in \mathbb{R}^n$ such that $\mathbf{w}^\star = \mathbf{B}^*\mathbf{v}^\star + \mathbf{z}$, where $\mathbf{z} \perp \mathbf{range}\,(\mathbf{B}^*)$. Since $\mathbf{B}$, viewed as an operator, is bounded, $\mathbf{null}\,(\mathbf{B}) = (\mathbf{range}\,(\mathbf{B}^*))^\perp$, so $\mathbf{B}\mathbf{z} = 0$. Now, since $\mathbf{z} \neq 0$ can only increase $\lambda\|\mathbf{w}\|^2_{L_2(\mathbb{R}^d,d\mu)}$ we conclude that $\mathbf{z} = 0$. Thus, $\mathbf{w}^\star = \mathbf{B}^*\mathbf{v}^\star$ and we can write

$$\begin{aligned}\min_{\mathbf{w}\in L_2(\mathbb{R}^d,d\mu)} \|\mathbf{B}\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|^2_{L_2(\mathbb{R}^d,d\mu)} &= \min_{\mathbf{v}\in\mathbb{R}^n} \|\mathbf{B}\mathbf{B}^*\mathbf{v}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{B}^*\mathbf{v}\|^2_{L_2(\mathbb{R}^d,d\mu)} \\ &= \min_{\mathbf{v}\in\mathbb{R}^n} \|\mathbf{K}\mathbf{v}-\mathbf{y}\|_2^2 + \lambda\mathbf{v}^{\mathrm{T}}\mathbf{K}\mathbf{v},\end{aligned}$$

where $\mathbf{K} = \mathbf{B}\mathbf{B}^* \in \mathbb{R}^{n\times n}$ is the kernel matrix previously defined. The optimal solution is $\mathbf{v}^\star = (\mathbf{K}+\lambda\mathbf{I}_n)^{-1}\mathbf{y}$, i.e., $\mathbf{v}^\star = \boldsymbol{\alpha}^\star$, so $\mathbf{w}^\star = \sum_{j=1}^n \alpha_j^\star \varphi(\mathbf{x}_j,\cdot)$. We now have

$$
\begin{aligned}
(\varphi(\mathbf{x},\cdot),\mathbf{w}^\star)_{L_2(\mathbb{R}^d,d\mu)} &= \int_{\mathbb{R}^d} \overline{\varphi(\mathbf{x},\boldsymbol{\eta})}\mathbf{w}^\star(\boldsymbol{\eta})d\mu(\boldsymbol{\eta}) \\
&= \int_{\mathbb{R}^d} e^{-2\pi i\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta}} \left(\sum_{j=1}^{n} \alpha_j^\star e^{2\pi i\mathbf{x}_j^{\mathrm{T}}\boldsymbol{\eta}}\right) d\mu(\boldsymbol{\eta}) \\
&= \sum_{j=1}^{n} \alpha_j^\star \int_{\mathbb{R}^d} e^{-2\pi i(\mathbf{x}-\mathbf{x}_j)^{\mathrm{T}}\boldsymbol{\eta}} d\mu(\boldsymbol{\eta}) \\
&= \sum_{j=1}^{n} \alpha_j^\star k(\mathbf{x},\mathbf{x}_j) = \mathbf{f}^\star(\mathbf{x})\,. \qquad \square
\end{aligned}
$$

The quasimatrix $\mathbf{B}$ is over complex-valued L_2 spaces. It is possible to actually define an equivalent SILR problem with a quasimatrix over a real-valued L_2 space. Let $\hat{\Omega} = \mathbb{R}^d \times [0, 2\pi]$ and $\hat{\mu} = \mu \times U(0, 2\pi)$, where $U(0, 2\pi)$ is the uniform measure on $[0, 2\pi]$. Now, let $L_2(\hat{\Omega}, d\hat{\mu})$ denote the space of *real-valued* square integrable functions with respect to the measure $\hat{\mu}$. Define the function $\hat{\varphi} : \mathcal{X} \times \hat{\Omega} \to \mathbb{R}$:

$$
\hat{\varphi}(\mathbf{x}, (\boldsymbol{\eta}, b)) = \sqrt{2}\cos(\mathbf{x}^{\mathrm{T}}\boldsymbol{\eta} + b)\,.
$$

Now, let

$$
\mathbf{C} = \begin{bmatrix} \hat{\varphi}(\mathbf{x}_1,\cdot)^* \\ \vdots \\ \hat{\varphi}(\mathbf{x}_n,\cdot)^* \end{bmatrix}.
$$

Then,

$$
\mathbf{f}^\star(\mathbf{x}) = (\hat{\varphi}(\mathbf{x},\cdot),\mathbf{u}^\star)_{L_2(\hat{\Omega},d\hat{\mu})},
$$

where

$$
\mathbf{u}^\star = \arg\min_{\mathbf{u}\in L_2(\hat{\Omega},d\hat{\mu})} \|\mathbf{C}\mathbf{u} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{u}\|_{L_2(\hat{\Omega},d\hat{\mu})}^2\,.
$$

See [37].

3.2.2. Approximating KRR using quasimatrix sampling. Computing the exact KRR estimator is costly (since $\mathbf{K}$ is typically dense, finding $\boldsymbol{\alpha}$ in (3.7) costs $O(n^3)$ using direct methods; computing $\mathbf{f}^\star(\mathbf{x})$ for some $\mathbf{x}$ using (3.6) costs $O(nd)$; since computing $\mathbf{f}^\star$ requires storing the entire training set, storage requirements for holding a representation of $\mathbf{f}^\star$ are $O(nd)$), which motivates looking for some approximation schemes. In this section we show how to perform approximate KRR by sampling the quasimatrix $\mathbf{B}$ defined in the previous subsection. The resulting method is actually identical to approximating KRR using random Fourier features, one of the most popular approximations of KRR, though the presentation as a sampling method for finding an approximate solution to an SILR problem is new.

Consider the wide quasimatrix $\mathbf{B}$ defined in the previous subsection. A coordinate representation of $\mathbf{B}$ is

$$
\mathbf{z}(\boldsymbol{\eta}) = \begin{bmatrix} \overline{\varphi(\mathbf{x}_1,\boldsymbol{\eta})} \\ \vdots \\ \overline{\varphi(\mathbf{x}_n,\boldsymbol{\eta})} \end{bmatrix}.
$$

This allows us to discuss column sampling of $\mathbf{B}$. For $s \leq n$, consider the matrix $\mathbf{B}_{\boldsymbol{\eta}} \in \mathbb{C}^{n \times s}$ obtained by column sampling $\mathbf{B}$ according to μ. That is, we sample $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ according to μ and define the matrix

$$\mathbf{B}_{\boldsymbol{\eta}} = \begin{bmatrix} \mathbf{z}(\boldsymbol{\eta}_1) & \mathbf{z}(\boldsymbol{\eta}_2) & \cdots & \mathbf{z}(\boldsymbol{\eta}_s) \end{bmatrix} = \begin{bmatrix} \overline{\varphi(\mathbf{x}_1, \boldsymbol{\eta}_1)} & \cdots & \overline{\varphi(\mathbf{x}_1, \boldsymbol{\eta}_s)} \\ \vdots & & \vdots \\ \overline{\varphi(\mathbf{x}_n, \boldsymbol{\eta}_1)} & \cdots & \overline{\varphi(\mathbf{x}_n, \boldsymbol{\eta}_s)} \end{bmatrix}.$$

Let

$$\mathbf{w}^\star_{\boldsymbol{\eta}} = \arg\min_{\mathbf{w} \in \mathbb{C}^s} \|\mathbf{B}_{\boldsymbol{\eta}}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 .$$

Finding $\mathbf{w}^\star_{\boldsymbol{\eta}}$ amounts to solving a finite linear least squares problem, and can be accomplished using $O(ns^2)$ arithmetic operations (and, notably, without performing any FUNOPs). The approximate KRR estimator is

$$\mathbf{f}_{\boldsymbol{\eta}}(\mathbf{x}) = \sum_{i=1}^{s} \overline{\varphi(\mathbf{x}, \eta_i)} (w^\star_{\boldsymbol{\eta}})_i,$$

where $(w^\star_{\boldsymbol{\eta}})_i$ denotes entry i of $\mathbf{w}^\star_{\boldsymbol{\eta}}$. In a sense, the vector $\mathbf{w}^\star_{\boldsymbol{\eta}}$ is an approximation of the function $\mathbf{w}^\star$ that is obtained by solving a sampled version of (3.10), and $\mathbf{f}_{\boldsymbol{\eta}}$ approximates the inner product $(\varphi(\mathbf{x}, \cdot), \mathbf{w}^\star)_{L_2(\mathcal{X}, d\mu)}$.

3.3. Stretching a finite linear least squares problem. Since numerical computing is typically done with numbers and not with functions, it is natural to find an approximate solution to SILR problems by sampling the quasimatrix. Here, we show that it is also possible to go the other way, and "stretch" a finite linear least squares problem to an SILR problem. This process is interesting since it yields a novel interpretation to the use of the Johnson–Lindenstrauss sketch in order to approximately solve a linear regression problem.

Suppose that $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a full rank matrix with $n \gg d$, and that $\mathbf{y} \in \mathbb{R}^n$. Consider finding $\mathbf{w}^\star$ that minimizes $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$. Define the function

$$\varphi(\mathbf{x}, \boldsymbol{\eta}) = \boldsymbol{\eta}^{\mathrm{T}} \mathbf{x},$$

and let p denote the standard Gaussian density over $\mathbb{R}^n$. We have

$$\int_{\mathbb{R}^n} \varphi(\mathbf{x}, \boldsymbol{\eta})^2 p(\boldsymbol{\eta}) d\boldsymbol{\eta} = \mathbf{x}^{\mathrm{T}} \left(\int_{\mathbb{R}^n} \boldsymbol{\eta}\boldsymbol{\eta}^{\mathrm{T}} p(\boldsymbol{\eta}) d\boldsymbol{\eta} \right) \mathbf{x} = \mathbf{x}^{\mathrm{T}} \mathbf{x},$$

so for $\mathbf{x} \in \mathbb{R}^n$ it holds that $\varphi(\mathbf{x}, \cdot) \in L_2(\mathbb{R}^n, d\mu)$, where $d\mu$ denotes the standard Gaussian distribution. Denote by $\mathbf{x}_1, \ldots, \mathbf{x}_d$ the columns of $\mathbf{X}$. Define the $\infty \times d$ quasimatrix $\mathbf{A}$ over $L_2(\mathbb{R}^n, d\mu)$: $\mathbf{A} = [\, \varphi(\mathbf{x}_1, \cdot) \quad \cdots \quad \varphi(\mathbf{x}_d, \cdot) \,]$. We now show that

$$\mathbf{w}^\star = \arg\min_{\mathbf{w} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{w} - \varphi(\mathbf{y}, \cdot)\|^2_{L_2(\mathbb{R}^n, d\mu)} . \tag{3.11}$$

Indeed, for every $\mathbf{w} \in \mathbb{R}^d$ we have

$$\begin{aligned} \|\mathbf{A}\mathbf{w} - \varphi(\mathbf{y}, \cdot)\|^2_{L_2(\mathcal{X}, d\mu)} &= \int_{\mathbb{R}^n} \left(\sum_{i=1}^{d} (\boldsymbol{\eta}^{\mathrm{T}} \mathbf{x}_i) w_i - \boldsymbol{\eta}^{\mathrm{T}} \mathbf{y} \right)^2 p(\boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \int_{\mathbb{R}^n} \left(\boldsymbol{\eta}^{\mathrm{T}} (\mathbf{X}\mathbf{w} - \mathbf{y}) \right)^2 p(\boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2. \end{aligned}$$

Thus, we have converted the finite linear least squares problem to an SILR problem.

Let us now consider approximately solving (3.11) by sampling "rows" from $\mathbf{A}$ and the corresponding entries from $\varphi(\mathbf{y}, \cdot)$. Since $\mathbf{A}$ is a quasimatrix over an L_2 space, we need a coordinate representation to meaningfully talk about sampling rows from $\mathbf{A}$. A coordinate representation of $\mathbf{A}$ is $\mathbf{z}(\boldsymbol{\eta}) = \mathbf{X}^{\mathrm{T}}\boldsymbol{\eta}$, where the index set is $\Omega = \mathbb{R}$. We can now sample (3.11) as follows. We sample $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s \in \mathbb{R}^n$ independently according to p and form the matrix

$$\mathbf{A}_{\boldsymbol{\eta}} = \begin{bmatrix} \mathbf{z}(\boldsymbol{\eta}_1)^{\mathrm{T}} \\ \mathbf{z}(\boldsymbol{\eta}_2)^{\mathrm{T}} \\ \vdots \\ \mathbf{z}(\boldsymbol{\eta}_s)^{\mathrm{T}} \end{bmatrix}$$

and the vector $\mathbf{y}_{\boldsymbol{\eta}} = (\boldsymbol{\eta}_i^{\mathrm{T}}\mathbf{y})_{i=1}^s$. The sampled problem (which is again a finite linear least squares problem) is

$$\hat{\mathbf{w}} = \|\mathbf{A}_{\boldsymbol{\eta}}\mathbf{w} - \mathbf{y}_{\boldsymbol{\eta}}\|_2^2 .$$

One may ask whether $\hat{\mathbf{w}}$ is close to being a minimizer of $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$. Let $\mathbf{S}$ be the $s \times n$ matrix whose rows are $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$, then $\mathbf{A}_{\boldsymbol{\eta}} = \mathbf{S}\mathbf{X}$ and $\mathbf{y}_{\boldsymbol{\eta}} = \mathbf{S}\mathbf{y}$. Using known results on subspace embedding [47] we conclude that if $s = \Omega(d/\epsilon^2)$ then with high probability

$$\|\mathbf{X}\hat{\mathbf{w}} - \mathbf{y}\|_2 \leq (1+\epsilon)\|\mathbf{X}\mathbf{w}^\star - \mathbf{y}\|_2 .$$

The random matrix $\mathbf{S}$ is a Johnson–Lindenstrauss sketching matrix, and we have demonstrated that applying the Johnson–Lindenstrauss sketch corresponds to stretching the linear least squares problem and then applying plain row sampling. A surprising aspect of this observation is the fact that only $O(d/\epsilon^2)$ samples are sufficient. Indeed, standard techniques used for analyzing sampled linear least squares problems, which are based on matrix tail inequalities, can be used to derive results that require $\Omega(d \log d/\epsilon^2)$ samples at the very least; see section 6.

4. Direct methods. Direct methods attempt to compute the solutions of SILR problems using quasimatrix operations. This mostly involves FUNOPs (under the chebfun model), but in some cases the computation can be reduced to algorithms that operate in the standard model (without FUNOPs). For simplicity, we assume that the quasimatrix involved is either full rank or $\lambda > 0$.

4.1. Overdetermined SILR. Let $\mathbf{A}$ be a tall quasimatrix with n columns. We can solve the SILR problem in (3.1) using the normal equations. The development is essentially the same as that for the finite linear least squares case. Let $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}\|_2^2$ be the objective function. We have,

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}\|_2^2 = \mathbf{x}^{\mathrm{T}}(\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I}_n)\mathbf{x} - 2\mathbf{Re}\left(\mathbf{x}^{\mathrm{T}}\mathbf{A}^*\mathbf{b}\right) + \|\mathbf{b}\|_{\mathcal{H}}^2 .$$

Thus the optimum value is obtained as the solution of the following linear system:

$$(\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I}_n)\mathbf{x} = \mathbf{A}^*\mathbf{b}$$

(easily verified by computing the gradient of $f(\mathbf{x})$ and equating to zero). Note that since we assumed that either $\mathbf{A}$ is full rank or $\lambda > 0$, $\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I}_n$ is invertible.

Thus, we can find the optimal $\mathbf{x}^\star$ in the chebfun model by first computing $\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I}_n$ and $\mathbf{A}^*\mathbf{b}$ ($n(n+3)/2$ FUNOPs), and then solving an $n \times n$ linear system ($O(n^3)$ FLOPs). However, using the Gram matrix $\mathbf{A}^*\mathbf{A}$ entails a squaring of the condition number, so the use of a factorization is preferred numerically. It is simple algebra to show that if $\mathbf{A} = \mathbf{Q}\mathbf{R}$ is a reduced QR factorization of $\mathbf{A}$, then $\mathbf{x}^\star = (\mathbf{R} +$

$\lambda \mathbf{R}^{-*})^{-1}\mathbf{Q}^*\mathbf{b}$, so n FUNOPs and $O(n^3)$ FLOPs ($O(n^2)$ if $\lambda = 0$) are needed once we have a QR factorization. However, this entails explicitly inverting $\mathbf{R}^*$, and that matrix might be ill conditioned.

If $\mathbf{A}$ was a matrix, explicitly inverting $\mathbf{R}$ can be avoiding by factorizing the augmented matrix

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix}.$$

However, if $\mathbf{A}$ is a quasimatrix over $\mathcal{H}$, then $\hat{\mathbf{A}}$ is a quasimatrix over $\mathcal{H}\times\mathbb{C}^n$ (if column j of $\mathbf{A}$ is $\mathbf{a}_j$, then column j of the augmented quasimatrix is the tuple $(\mathbf{a}_j, \sqrt{\lambda}\mathbf{e}_j)$, where $\mathbf{e}_j$ is the jth identity vector in $\mathbb{C}^n$), possibly making computations more cumbersome.[3] We now show how it is possible to find $\mathbf{x}^\star$ using a QR factorization of $\mathbf{A}$ without explicitly inverting $\mathbf{R}$. This involves fairly standard linear algebra tricks. Suppose we have a QR factorization $\mathbf{A} = \mathbf{Q}_\mathbf{A}\mathbf{R}_\mathbf{A}$. Let $\mathbf{C}$ be the matrix obtained by augmenting $\mathbf{R}_\mathbf{A}$ with the matrix $\sqrt{\lambda}\mathbf{I}_n$, and form a QR factorization of it. That is,

$$\mathbf{C} = \begin{bmatrix} \mathbf{R}_\mathbf{A} \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} = \mathbf{Q}_\mathbf{C}\mathbf{R}_\mathbf{C}.$$

Let $\mathbf{Q}_{\mathbf{C},1}$ denote the top n rows of $\mathbf{Q}_\mathbf{C}$, and $\mathbf{Q}_{\mathbf{C},2}$ the bottom. Now,

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_\mathbf{A} & \\ & \mathbf{I}_n \end{bmatrix} \mathbf{C} = \begin{bmatrix} \mathbf{Q}_\mathbf{A} & \\ & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\mathbf{C},1} \\ \mathbf{Q}_{\mathbf{C},2} \end{bmatrix} \mathbf{R}_\mathbf{C} = \begin{bmatrix} \mathbf{Q}_\mathbf{A}\mathbf{Q}_{\mathbf{C},1} \\ \mathbf{Q}_{\mathbf{C},2} \end{bmatrix} \mathbf{R}_\mathbf{C}.$$

Also, since

$$\begin{bmatrix} \mathbf{Q}_{\mathbf{C},1}^*\mathbf{Q}_\mathbf{A}^* & \mathbf{Q}_{\mathbf{C},2}^* \end{bmatrix} \begin{bmatrix} \mathbf{Q}_\mathbf{A}\mathbf{Q}_{\mathbf{C},1} \\ \mathbf{Q}_{\mathbf{C},2} \end{bmatrix} = \mathbf{I}_n,$$

we have a QR factorization of $\hat{\mathbf{A}}$. This implies that $\mathbf{x}^\star$ is the solution of the triangular system

$$\mathbf{R}_\mathbf{C}\mathbf{x} = \mathbf{Q}_{\mathbf{C},1}^*\mathbf{Q}_\mathbf{A}^*\mathbf{b}.$$

As is the case of finite linear least squares, a reduced SVD can be used to solve SILR problems as well. If $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$ is a reduced SVD factorization, then simple algebra reveals that $\mathbf{x}^\star = \mathbf{V}(\Sigma^2 + \lambda\mathbf{I}_n)^{-1}\Sigma\mathbf{U}^*\mathbf{b}$, so n FUNOPs and $O(n^2)$ FLOPs are needed once we have an SVD factorization.

In certain cases it might be possible to compute $\mathbf{A}^*\mathbf{A}$ and $\mathbf{A}^*\mathbf{b}$ analytically, without resorting to FUNOPs. We give a concrete example later, when we discuss the underdetermined case.

4.2. Underdetermined SILR. Let $\mathbf{A}$ be a wide quasimatrix with n rows. Again, the following argument follows closely the one used for finite linear least squares. The space $\mathbf{range}\,(\mathbf{A}^*)$ is a closed linear subspace of $\mathcal{H}$, so we can write $\mathbf{x}^\star = \mathbf{A}^*\mathbf{y}^\star + \mathbf{z}^\star$, where $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{z} \perp \mathbf{range}\,(\mathbf{A}^*)$. Since $\mathbf{A}$, viewed as an operator, is bounded, $\mathbf{null}\,(\mathbf{A}) = (\mathbf{range}\,(\mathbf{A}^*))^\perp$, so $\mathbf{A}\mathbf{z} = 0$. Thus, the objective at $\mathbf{x}^\star$ is

$$\|\mathbf{A}\mathbf{x}^\star - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}^\star\|_\mathcal{H}^2 = \|\mathbf{A}\mathbf{A}^*\mathbf{y}^\star - \mathbf{b}\|_2^2 + \lambda\|\mathbf{A}^*\mathbf{y}^\star\|_\mathcal{H}^2 + \lambda\|\mathbf{z}^\star\|_\mathcal{H}^2,$$

where we used the fact that $\mathbf{z}^\star \perp \mathbf{A}^*\mathbf{y}^\star$. Obviously, $\mathbf{z}^\star = 0$, otherwise the objective can be reduced. Denoting $\mathbf{K} = \mathbf{A}\mathbf{A}^* \in \mathbb{R}^{n\times n}$, we find that $\mathbf{y}^\star$ is the minimizer of

$$f(\mathbf{y}) = \|\mathbf{K}\mathbf{y} - \mathbf{b}\|_2^2 + \lambda\mathbf{y}^{\mathrm{T}}\mathbf{K}\mathbf{y}.$$

[3] We remark that the chebfun library does support hybrids of quasimatrices and a matrix (and calls such objects by the name "chebmatrix").

This can be written as a determined (for $\lambda = 0$) or overdetermined ($\lambda > 0$) finite linear least squares problems, from which we find that $\mathbf{y}^\star$ solves the equation

$$(\mathbf{K}^2 + \lambda\mathbf{K})\mathbf{y} = \mathbf{K}\mathbf{b}.$$

Since we assumed that either $\mathbf{A}$ is full rank or $\lambda > 0$, $\mathbf{K} + \lambda\mathbf{I}_n$ is invertible, and the vector $(\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{b}$ solves the equation. The solution is unique, thus $\mathbf{y}^\star = (\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{b}$. We find that

$$\mathbf{x}^\star = \mathbf{A}^*(\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{b}\,.$$

Thus, in the chebfun model we can find the optimal $\mathbf{x}^\star$ by first computing $\mathbf{K} + \lambda\mathbf{I}_n$ ($n(n+1)/2$ FUNOPs), solving for $\mathbf{y}^\star$ ($O(n^3)$ FLOPs), and finally computing $\mathbf{A}^*\mathbf{y}^\star$ (n FUNOPs).

We can avoid forming the potentially ill conditioned matrix $\mathbf{K}$ using a QR factorization of $\mathbf{A}^*$ in a way similar to the previous subsection, where now we have a QR factorization $\mathbf{A}^* = \mathbf{Q}_\mathbf{A}\mathbf{R}_\mathbf{A}$. Similarly, we can factorize

$$\begin{bmatrix} \mathbf{A}^* \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_\mathbf{A}\mathbf{Q}_{\mathbf{C},1} \\ \mathbf{Q}_{\mathbf{C},2} \end{bmatrix} \mathbf{R}_\mathbf{C}\,.$$

This implies that

$$\begin{bmatrix} \mathbf{A} & \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} = \mathbf{R}_\mathbf{C}^* \begin{bmatrix} \mathbf{Q}_{\mathbf{C},1}^*\mathbf{Q}_\mathbf{A}^* & \mathbf{Q}_{\mathbf{C},2}^* \end{bmatrix}$$

is an LQ factorization of the augmented matrix. Hence, $\mathbf{x}^\star$ is equal to

$$\mathbf{x}^\star = \mathbf{Q}_\mathbf{A}\mathbf{Q}_{\mathbf{C},1}(\mathbf{R}_\mathbf{C}^*)^{-1}\mathbf{b}\,.$$

An SVD factorization can be used as well: if $\mathbf{A}^* = \mathbf{U}\Sigma\mathbf{V}^*$ is an SVD factorization, then $\mathbf{x}^\star = \mathbf{U}(\Sigma^2 + \lambda\mathbf{I}_n)^{-1}\Sigma\mathbf{V}^*\mathbf{b}$.

In certain cases, it is possible to compute $\mathbf{K}$ analytically. As an example, consider again KRR (section 3.2) in the RKHS formulation (3.8). Due to the definition of $\mathcal{H}_k$, the ij entry of $\mathbf{K}$ is

$$\mathbf{K}_{ij} = (k(\mathbf{x}_i, \cdot), k(\mathbf{x}_j, \cdot))_{\mathcal{H}_k} = k(\mathbf{x}_i, \mathbf{x}_j)\,,$$

thus we can form $\mathbf{K} + \lambda\mathbf{I}_n$ and compute $\alpha = (\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{y}$ using $O(n^2d + n^3)$ FLOPs. The solution to (3.8) is then

$$\mathbf{f}^\star(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

and all computations are done in the standard model.

5. Iterative methods. In this section we discuss solving SILR problems using iterative methods. First, we consider using the classical approach of Krylov subspace methods. We show how methods such as LSMR or LSQR can be rather naturally generalized to quasimatrices. Next, we propose a novel method based on stochastic optimization which requires considerably fewer FUNOPs, but depends on the ability to sample the quasimatrix. We also show that this algorithm can, in certain cases, be applied in the standard model (without FUNOPs).

5.1. Krylov subspace methods. Krylov subspace methods are one of the most important classes of iterative methods in numerical linear algebra. Many of the most widely used iterative linear solvers are Krylov subspace methods. One important benefit of Krylov subspace methods is that they only use matrix-vector operations. For SILR, when working in the chebfun model, this implies that each iteration does only $O(n)$ FUNOPs.

It was already observed by several authors that it is possible to generalize Krylov subspace methods to operator equations. For example, Olver suggested the use of GMRES with the differentiation operator [32], and the chebfun library implements GMRES for operator equations. Continuous analogues of CG, GMRES, and MINRES appear in [19] in the context of differential operators. In the same vein, we can adapt Krylov subspace algorithms for finite linear least squares, such as LSQR [34] and LSMR [18], to solve SILR problems. Here we describe the LSMR algorithm for quasimatrices. The development is a rather straightforward generalization of the matrix case, but we show it for concreteness.

5.1.1. Golub–Kahan bidiagonalization process for quasimatrices. LSMR and LSQR are based on Golub–Kahan bidiagonalization [20]. The goal of Golub–Kahan bidiagonalization is to iteratively find a decomposition $\mathbf{U}^*\mathbf{A}\mathbf{V} = \mathbf{B}$ where $\mathbf{U}$ and $\mathbf{V}$ have orthogonal columns, and $\mathbf{B}$ is a bidiagonal matrix. When $\mathbf{A}$ is a quasimatrix over $\mathcal{H}$, one of $\mathbf{U}$ and $\mathbf{V}$ is a quasimatrix over $\mathcal{H}$ and the other one is a matrix. The algorithm remains essentially unchanged, and is given in Algorithm 5.1 for a tall quasimatrix (for a wide quasimatrix the algorithm remains the same under the corresponding changes of norms).

After k steps of the algorithm, we have $\mathbf{A}\mathbf{V}_k{=}\mathbf{U}_{k+1}\mathbf{B}_k$ and $\mathbf{A}^*\mathbf{U}_{k+1}{=}\mathbf{V}_{k+1}\mathbf{L}_{k+1}^*$. Note that, in the overdetermined case, where $\mathbf{A}$ is an $\infty \times n$ quasimatrix, $\mathbf{V}_k$ is an $n \times k$ matrix and $\mathbf{U}_k$ is an $\infty \times k$ tall quasimatrix over $\mathcal{H}$. In the underdetermined case, $\mathbf{U}_k$ is an $n \times k$ matrix and $\mathbf{V}_k$ is an $\infty \times k$ tall quasimatrix over $\mathcal{H}$ such that $\mathbf{V}_k^*\mathbf{V}_k = \mathbf{U}_k^*\mathbf{U}_k = \mathbf{I}_k$. The algorithm also defines a $(k{+}1) \times k$ lower bidiagonal matrix $\mathbf{B}_k$.

5.1.2. LSMR for overdetermined SILR. Recall that the solution $\mathbf{x}^\star$ of an overdetermined SILR problem solves the normal equations $(\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I}_n)\mathbf{x}^\star = \mathbf{A}^*\mathbf{b}$. LSMR is equivalent to applying MINRES to the normal equations, i.e., in each iteration the minimizer of $\mathbf{A}^*\mathbf{b} - (\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I}_n)\mathbf{x}$ is found under the constraint that $\mathbf{x}$ belongs to the Krylov subspace. Thus, defining $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ at iteration k, LSMR minimizes $\|\mathbf{A}^*\mathbf{r}_k - \lambda\mathbf{x}_k\|_2$ subject to $\mathbf{x}_k \in \mathcal{K}_k(\mathbf{A}^*\mathbf{A}, \mathbf{b})$, where $\mathcal{K}_k(\mathbf{A}^*\mathbf{A}, \mathbf{b}) :=$ span $\{\mathbf{b}, \mathbf{A}^*\mathbf{A}\mathbf{b}, \ldots, (\mathbf{A}^*\mathbf{A})^{k-1}\mathbf{b}\}$ is the kth order Krylov subspace generated by $\mathbf{A}^*\mathbf{A}$ and $\mathbf{b}$.

To find x_k, LSMR uses the Golub–Kahan bidiagonalization. After k iterations we have $\mathbf{A}\mathbf{V}_k = \mathbf{U}_{k+1}\mathbf{B}_k$ and $\mathbf{A}^*\mathbf{U}_{k+1} = \mathbf{V}_{k+1}\mathbf{L}_{k+1}^*$, where $\mathbf{V}_k$ is an $n \times k$ matrix and $\mathbf{U}_k$ is an $\infty \times k$ tall quasimatrix over $\mathcal{H}$. Thus,

$$\mathbf{A}^*(\mathbf{A}\mathbf{V}_k) = (\mathbf{A}^*\mathbf{U}_{k+1})\mathbf{B}_k = \mathbf{V}_{k+1}\mathbf{L}_{k+1}^{\mathrm{T}}\mathbf{B}_k = \mathbf{V}_{k+1}\begin{bmatrix} \mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k \\ \alpha_{k+1}\beta_{k+1}\mathbf{e}_{k+1}^{\mathrm{T}} \end{bmatrix},$$
$$\mathbf{A}^*\mathbf{b} = \beta_1\mathbf{A}^{\mathrm{T}}\mathbf{u}_1 = \alpha_1\beta_1\mathbf{V}_{k+1}\mathbf{e}_1\,.$$

(These equations are the same as in the matrix case; the quasimatrix algebra defined in section 2 allows us to write essentially the same derivations). Since $\mathbf{x}_k$ is in the Krylov subspace, we can write $\mathbf{x}_k = \mathbf{V}_k\mathbf{y}_k$ for some $\mathbf{y}_k \in \mathbb{C}^k$. Thus, we can write

Algorithm 5.1. (Tall) Quasimatrix Golub-Kahan bidiagonalizaion.

1: **Inputs:** Tall $\infty \times n$ quasimatrix $\mathbf{A}$ over $\mathcal{H}$, and $\infty \times 1$ quasimatrix $\mathbf{b}$.

2: **set:**

$$\beta_1 = \|\mathbf{b}\|_{\mathcal{H}}, \quad \mathbf{u}_1 = \mathbf{b}/\beta_1, \quad \alpha_1 = \|\mathbf{A}^{\mathbf{T}}\mathbf{u}_1\|_2, \quad \mathbf{v}_1 = \mathbf{A}^{\mathbf{T}}\mathbf{u}_1/\alpha_1 .$$

3: **for** $k = 1, 2, \ldots,$

$$\beta_{k+1} = \|\mathbf{A}\mathbf{v}_k - \alpha_k\mathbf{u}_k\|_{\mathcal{H}}, \mathbf{u}_{k+1} = (\mathbf{A}v_k - \alpha_k\mathbf{u}_k)/\beta_{k+1}$$
$$\alpha_{k+1} = \|\mathbf{A}^*\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{v}_k\|_2, \mathbf{v}_{k+1} = (\mathbf{A}^*\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{v}_k)/\alpha_{k+1} .$$

4: **Denote:**

$$\mathbf{V}_k = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_k \end{bmatrix}, \mathbf{U}_k = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_k \end{bmatrix}$$
$$\mathbf{B}_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}, \mathbf{L}_K = \begin{bmatrix} \mathbf{B}_k & \alpha_{k+1}\mathbf{e}_{k+1} \end{bmatrix} .$$

$$\begin{aligned} \|\mathbf{A}^*(\mathbf{A}\mathbf{x}_k - \mathbf{b})\|_2^2 + \lambda\|\mathbf{x}_k\|_2^2 &= \|\mathbf{A}^*(\mathbf{A}\mathbf{V}_k\mathbf{y}_k - \mathbf{b})\|_2^2 + \lambda\|\mathbf{V}_k\mathbf{y}_k\|_2^2 \\ &= \left\| \mathbf{V}_{k+1} \left(\begin{bmatrix} \mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k \\ \alpha_{k+1}\mathbf{e}_{k+1}^{\mathrm{T}} \end{bmatrix} \mathbf{y}_k - \alpha_1\beta_1\mathbf{e}_1 \right) \right\|_2^2 + \lambda\|\mathbf{y}_k\|_2^2 . \end{aligned}$$

So, finding $\mathbf{y}_k$ and $\mathbf{x}_k$ has been reduced to the solution of a finite linear least squares problem. An algorithm for finding these vectors efficiently and iteratively is described in [18].

Stopping criteria: the Golub–Kahan process terminates whenever $\alpha_{k+1} = 0$ or $\beta_{k+1} = 0$, which implies that the last equation is zero. However, we can use one of the stopping criteria originally presented for the LSQR algorithm, involving the predetermined parameters ATOL, BTOL, and CONLIM:

$$S_1 \text{ :Stop if } \quad \sqrt{\|\mathbf{r}_k\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}_k\|_2^2} \le \mathrm{BTOL}\|\mathbf{b}\|_{\mathcal{H}} + \mathrm{ATOL}\sqrt{\sigma_{\max}(\mathbf{B}_k)^2 + \lambda}\|\mathbf{x}_k\|_2,$$

$$S_2 \text{ :Stop if } \quad \|\mathbf{A}^*\mathbf{r}_k - \lambda\mathbf{x}_k\|_2 \le \mathrm{ATOL}\sqrt{\sigma_{\max}(\mathbf{B}_k)^2 + \lambda}\sqrt{\|\mathbf{A}\mathbf{x}_k - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda},$$

$$S_3 \text{ :Stop if } \quad \sqrt{\frac{\sigma_{\max}(\mathbf{B}_k)^2 + \lambda}{\sigma_{\min}(\mathbf{B}_k)^2 + \lambda}} \ge \mathrm{CONLIM} .$$

The motivation for these stopping rules is the fact that $\sigma_{\max}(\mathbf{B}_k)$ and $\sigma_{\min}(\mathbf{B}_k)$ provide estimates for $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$. This follows from the fact that $\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k = \mathbf{V}_k^*\mathbf{A}^*\mathbf{A}\mathbf{V}_k$. See [18] for more details.

Complexity: when compared to the matrix version of LSMR, the quasimatrix version trades each matrix-vector product with n FUNOPs. Thus, in terms of

FUNOPs, $2n$ FUNOPs are required per iteration. Since the number of iterations is $O(\sqrt{\kappa(\mathbf{A}^*\mathbf{A}+\lambda\mathbf{I}_n)}$, overall complexity is $O(n\sqrt{\kappa(\mathbf{A}^*\mathbf{A}+\lambda\mathbf{I}_n)}$ FUNOPs.

5.1.3. Numerical example. We illustrate the use of LSMR for the problem of approximating the Runge function $f(x) = 1/(1+25x^2)$ on $[-1,1]$ using a polynomial of degree 300. We can write the approximation as the solution of an overdetermined SILR problem, where $\mathbf{A}$ is any quasimatrix whose columns span the space of degree 300 polynomials. However, we want $\mathbf{A}$ to be reasonably well conditioned so that LSMR will converge quickly, so we use Chebyshev polynomials as the columns of $\mathbf{A}$ (we empirically observed that when the columns are the Chebyshev polynomials, $\mathbf{A}$ is well conditioned, though we are unaware of any analytical result showing this; note that taking the normalized Legendre polynomials instead would have resulted in an orthogonal $\mathbf{A}$, which would have made for an uninteresting numerical example). Thus, we solve the SILR problem, where

$$\mathbf{A} = \begin{bmatrix} T_0 & T_1 & \cdots & T_{299} \end{bmatrix}, \quad \mathbf{b} = \left[\frac{1}{1+25x^2}\right].$$

In the above, T_j is the jth Chebyshev polynomial. We use $\lambda = 0$ (no regularization) and parameters $\mathrm{ATOL} = \mathrm{BTOL} = 10^{-7}$. Convergence plots are shown in Figure 1.

5.2. Stochastic variance reduced gradient. Recent literature on convex optimization advocated the use of stochastic methods. Even for the specialized cases of solving linear equations or linear least squares, such methods have been shown to be beneficial [27, 21]. In this section, we propose a method for solving SILR problems using SVRG [23, 48]. SVRG is a stochastic optimization method for minimizing objective functions that have a finite sum structure, i.e., of the form

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}).$$

For such objective functions, we can compute stochastic gradients by sampling an index of the sum. SVRG's main benefit comes from the fact that it combines such stochastic gradients with a small amount of full gradients (i.e., exact gradients of f). For strongly convex functions, the number of such full gradients we need to compute is independent of the condition number (however, the number of stochastic gradients does depend on the average condition number).

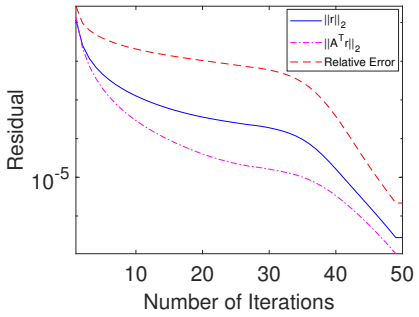

FIG. 1. *Numerical illustration: using LSMR to solve an SILR related to approximating the Runge function using a polynomial.*

For SILR, full gradients correspond to products of a quasimatrix $\mathbf{A}$ with a vector or function, but this is the only operation that assumes the chebfun model and requires FUNOPs. Thus, by using SVRG we remove the condition number dependence for the number of FUNOPs required for convergence, which is a major improvement over Krylov methods. However, stochastic gradients in SVRG correspond to sampling objective functions, and for SILR this translates to sampling a row from a tall quasimatrix or a column from a wide quasimatrix. Thus, the quasimatrix must be an quasimatrix over an L_2 space, and must have a coordinate representation.

One obstacle in applying SVRG to SILR problems is that such problems cannot be written as a finite sum, but rather can be written as an integral of simpler functions, i.e.,

$$f(\mathbf{x}) = \int_\Omega f_{\boldsymbol{\eta}}(\mathbf{x}) d\mu(\boldsymbol{\eta}) \,.$$

We generalize SVRG and its analysis to handle such functions. The generalization might be of independent interest, and appears in Appendix A.

5.2.1. Overdetermined SILR. Consider the overdetermined SILR problem (3.1), where the quasimatrix $\mathbf{A}$ is over $\mathcal{H} = L_2(\Omega, d\mu)$ for some index set Ω. We further assume we have a coordinate representation $\mathbf{z}_\mathbf{A} : \Omega \to \mathbb{R}^n$ for $\mathbf{A}$ and $\mathbf{z}_\mathbf{b} : \Omega \to \mathbb{R}$ for $\mathbf{b}$. We further assume there exists an M such that for every $\boldsymbol{\eta} \in \Omega$ we have $\|\mathbf{z}_\mathbf{A}(\boldsymbol{\eta})\|_2^2 \le M$. We can write the objective function in (3.1) as an integral,

$$\begin{aligned}
\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{L_2(\Omega,d\mu)}^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2 &= \frac{1}{2}\left\|\sum_{i=1}^n x_i \overline{\mathbf{z}_\mathbf{A}(\cdot)_i} - \mathbf{b}\right\|_{L_2(\Omega,d\mu)}^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2 \\
&= \frac{1}{2}\|\mathbf{z}_\mathbf{A}(\cdot)^*\mathbf{x} - \mathbf{b}\|_{L_2(\Omega,d\mu)}^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2 \\
&= \frac{1}{2}\int_\Omega \left(\mathbf{z}_\mathbf{A}(\boldsymbol{\eta})^*\mathbf{x} - \mathbf{z}_\mathbf{b}(\boldsymbol{\eta})\right)^2 + \lambda\|\mathbf{x}\|_2^2 \, d\mu(\boldsymbol{\eta}) \\
&= \int_\Omega f_{\boldsymbol{\eta}}(\mathbf{x}) d\mu(\boldsymbol{\eta}),
\end{aligned}$$

where

$$f_{\boldsymbol{\eta}}(\mathbf{x}) \coloneqq \frac{1}{2}\left(\mathbf{z}_\mathbf{A}(\boldsymbol{\eta})^*\mathbf{x} - \mathbf{z}_\mathbf{b}(\boldsymbol{\eta})\right)^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2 \,.$$

We can now apply the aforementioned variant of SVRG [23] (see Appendix A), which is adapted for objective integrable functions. To do so, the following assumptions need to be verified.

Assumption 4. For all $\boldsymbol{\eta} \in \Omega$, $\nabla f_{\boldsymbol{\eta}}(\mathbf{x})$ is Lipschitz continuous, i.e., there exists $L_{\boldsymbol{\eta}} > 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$\|\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{y})\| \le L_{\boldsymbol{\eta}}\|\mathbf{x} - \mathbf{y}\| \,.$$

Assumption 5. Suppose that $f(\mathbf{x})$ is strongly convex, i.e., there exists $\gamma > 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{x}) - f(\mathbf{y}) \ge \frac{\gamma}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \nabla f(\boldsymbol{y})^{\mathrm{T}}(\mathbf{x} - \mathbf{y}) \,.$$

Assumption 6. The equality $\nabla f(\mathbf{x}) = \int_\Omega \nabla f_{\boldsymbol{\eta}}(\mathbf{x}) d\mu(\boldsymbol{\eta})$ holds.

Assumption 7. $L_{\sup} \coloneqq \sup_{\boldsymbol{\eta}\in\Omega} L_{\boldsymbol{\eta}} < \infty$.

We begin by writing

$$f(\mathbf{x}) \coloneqq \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2_{L_2(\Omega,d\mu)} + \frac{\lambda}{2}\|\mathbf{x}\|_2^2 = \frac{1}{2}\mathbf{x}^{\mathrm{T}}(\mathbf{K}+\lambda\mathbf{I}_n)\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^*\mathbf{b} + \frac{1}{2}\|\mathbf{b}\|^2_{L_2(\Omega,d\mu)},$$

where $\mathbf{K} = \mathbf{A}^*\mathbf{A} \in \mathbb{R}^{n\times n}$. Thus,

$$\nabla f(\mathbf{x}) = \mathbf{A}^*(\mathbf{A}\mathbf{x}-\mathbf{b}) + \lambda\mathbf{x}\,.$$

It can be seen that Assumption 5 holds with $\gamma = \lambda + \lambda_{\min}(\mathbf{K})$. We also have

$$\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) = \mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\left(\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})^*\mathbf{x} - \mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta})\right) + \lambda\mathbf{x}$$

with

$$\begin{aligned}\int_\Omega \nabla f_{\boldsymbol{\eta}}(\mathbf{x})d\mu(\boldsymbol{\eta}) &= \int_\Omega \mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\left(\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})^*\mathbf{x} - \mathbf{z}_{\mathbf{b}}(\eta)\right) + \lambda\mathbf{x}d\mu(\boldsymbol{\eta})\\ &= \left(\int_\Omega \mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})^* d\mu(\boldsymbol{\eta})\right)\mathbf{x} - \int_\Omega \mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta})d\mu(\boldsymbol{\eta}) + \lambda\mathbf{x}\\ &= \mathbf{K}\mathbf{x} - \mathbf{A}^*\mathbf{b} + \lambda\mathbf{x} = \nabla f(\mathbf{x}),\end{aligned}$$

so Assumption 6 holds as well. Note that for every $\boldsymbol{\eta} \in \mathcal{X}$

$$\begin{aligned}\|\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{y})\|_2 &= \|(\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})^* + \lambda\mathbf{I}_n)(\mathbf{x}-\mathbf{y})\|_2\\ &\le \left(\|\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\|_2^2 + \lambda\right)\|\mathbf{x}-\mathbf{y}\|_2,\end{aligned}$$

so each $\nabla f_{\boldsymbol{\eta}}$ is Lipschitz continuous with Lipschitz constant $L_{\boldsymbol{\eta}} = \|\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\|_2^2 + \lambda$. Thus, Assumptions 4 and 7 hold with $L_{\sup} = M + \lambda$.

Therefore, according to Theorem 22 (in Appendix A), if we set

$$m = 50\cdot\kappa,\ \kappa = \frac{M+\lambda}{\gamma^2+\lambda},\ \ \alpha = \frac{\theta}{M+\lambda},\ 0<\theta<\frac{1}{4},$$

where γ is any lower bound on $\sigma_{\min}(\mathbf{A})$ (if $\lambda > 0$ we can take $\gamma = 0$), then taking $\theta = 1/5$ and assuming we start with $\mathbf{x} = 0$ yields

$$\mathbb{E}\left[f(\tilde{\mathbf{x}}_s)\right] - f(\mathbf{x}^\star) \le \left(\frac{5}{6}\right)^s\left(\frac{1}{2}\|\mathbf{b}\|^2_{L_2(\Omega,d\mu)} - f(\mathbf{x}^\star)\right).$$

Overall, to reduce (in expectation) by a factor of ϵ we need to do $O(\log(1/\epsilon))$ outer iterations, each requiring $2n$ FUNOPs. Each outer iteration requires $O(\kappa)$ inner iterations, each requiring $O(n+T)$ FLOPS, where T is the cost of computing $\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})$ and $\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta})$ for a given $\boldsymbol{\eta}$, so in total we need $O((n+T)\cdot\kappa\cdot\log(1/\epsilon))$ FLOPs. We see that in contrast with Krylov subspace methods, the number of FUNOPs does *not* depend on the condition number. The proposed algorithm is summarized in Algorithm 5.2.

5.2.2. Underdetermined SILR. We now consider the case that $\mathbf{A}$ is a wide $n\times\infty$ quasimatrix over $L_2(\Omega,d\mu)$ of full rank, and $\mathbf{b}\in\mathbb{C}^n$. As explained in subsection 4.2, the optimal solution $\mathbf{x}^\star$ has the form $\mathbf{x}^\star = \mathbf{A}^*\mathbf{y}^\star$ for $\mathbf{y}^\star\in\mathbb{C}^n$. In addition, we have $\mathbf{y}^\star = (\mathbf{K}+\lambda\mathbf{I}_n)^{-1}\mathbf{b}$, where $\mathbf{K} = \mathbf{A}\mathbf{A}^*$. Hence,

$$\mathbf{y}^\star = \arg\min_{\mathbf{y}\in\mathbb{C}^n}\frac{1}{2}\mathbf{y}^*(\mathbf{K}+\lambda\mathbf{I}_n)\mathbf{y} - \mathbf{y}^*\mathbf{b}\,.$$

Algorithm 5.2. SVRG for overdetermined SILR.

Inputs:
- Tall $\infty \times n$ quasimatrix $\mathbf{A}$ over $L_2(\Omega, d\mu)$, along with coordinate representation $\mathbf{z_A} : \Omega \to \mathbb{C}^n$
- $\mathbf{b}$ with coordinate representation $\mathbf{z_b} : \Omega \to \mathbb{C}^n$, $\lambda > 0$
- M such that for all $\boldsymbol{\eta} \in \Omega$ we have $\|\mathbf{z_A}(\boldsymbol{\eta})\|_2^2 \leq M$
- γ such that $0 \leq \gamma \leq \sigma_{\min}(\mathbf{A})$
- Accuracy parameter $\epsilon > 0$

$\tilde{\mathbf{x}}_0 \leftarrow 0$

$\alpha \leftarrow \frac{1}{5(M+\lambda)}$, $m \leftarrow \frac{50(M+\lambda)}{\gamma^2+\lambda}$

$s_{\max} \leftarrow \left(\log\left(\frac{6}{5}\right)\right)^{-1} \cdot \log\left(\frac{\|\mathbf{b}\|_{L_2(d\mu)}^2}{2\epsilon}\right)$

Iterate: for $s = 1, 2, \ldots, s_{\max}$

  $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_{s-1}$

  $\tilde{\mathbf{g}} = \mathbf{A}^*(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}) + \lambda\tilde{\mathbf{x}}$

  $\mathbf{x}_0 = \tilde{\mathbf{x}}$

  Iterate: for $k = 1, 2, \ldots, m$

    sample $\boldsymbol{\eta}_k$ from the distribution μ

    $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha\left(\mathbf{z_A}(\boldsymbol{\eta}_k)\mathbf{z_A}(\boldsymbol{\eta}_k)^*\left(\mathbf{x}_{k-1} - \tilde{\mathbf{x}}\right) + \lambda\left(\mathbf{x}_{k-1} - \tilde{\mathbf{x}}\right) + \tilde{\mathbf{g}}\right)$

  end

  option I: set $\tilde{\mathbf{x}}_s = \mathbf{x}_m$

  option II: set $\tilde{\mathbf{x}}_s = \frac{1}{m}\sum_{k=1}^m \mathbf{x}_k$

end

return $\tilde{\mathbf{x}}_{s_{\max}}$

Thus, we can find approximate solutions to the regression problem by optimizing

$$f(\mathbf{y}) \coloneqq \frac{1}{2}\mathbf{y}^*(\mathbf{K} + \lambda\mathbf{I}_n)\mathbf{y} - \mathbf{y}^*\mathbf{b}$$

and returning $\tilde{\mathbf{x}} = \mathbf{A}^*\tilde{\mathbf{y}}$ for the $\tilde{\mathbf{y}}$ found by the optimization process. Note that if we find a $\tilde{\mathbf{y}}$ such that $f(\tilde{\mathbf{y}}) \leq f(\mathbf{y}^\star) + \epsilon$, then for $\tilde{\mathbf{x}} = \mathbf{A}^*\tilde{\mathbf{y}}$ we have

$$\|\tilde{\mathbf{x}} - \mathbf{x}^\star\|_{L_2(\Omega,d\mu)}^2 \leq \frac{2\lambda_{\max}(\mathbf{K})}{\lambda_{\min}(\mathbf{K}) + \lambda} \cdot \epsilon\,.$$

We can again use SVRG (with the specific variant described in Appendix A) to minimize $f(\mathbf{y})$. Since the assumptions are the same as in the previous section, and the developments are almost identical, we do not repeat them. The algorithm is almost identical to Algorithm 5.2, with two small differences: the equation for $s_{\max}$ is replaced by $(\log(6/5))^{-1} \cdot \log(\|\mathbf{b}\|_2^2 + \|\mathbf{b}\|_2^2/(\epsilon\sqrt{\lambda_{\min}(\mathbf{K}) + \lambda}))$ and $\tilde{\mathbf{g}}$ is $\mathbf{A}(\mathbf{A}^*\tilde{\mathbf{y}}) + \lambda\tilde{\mathbf{y}} - \mathbf{b}$.

5.2.3. SVRG for KRR. Recall that KRR can be recast as an underdetermined SILR problem (subsection 3.2). We can use the algorithm from the previous subsection to solve this SILR problem. However, since we can compute $\mathbf{K}$ via the kernel function without assuming the chebfun model, we can avoid performing FUNOPs when computing $\tilde{\mathbf{g}}$. That is, we can apply SVRG under the standard model. For this case, the assumptions hold with $M = d$.

We illustrate the performance of this algorithm on a small scale experiment. The goal is to learn a one dimensional dataset generated by noisily sampling the function

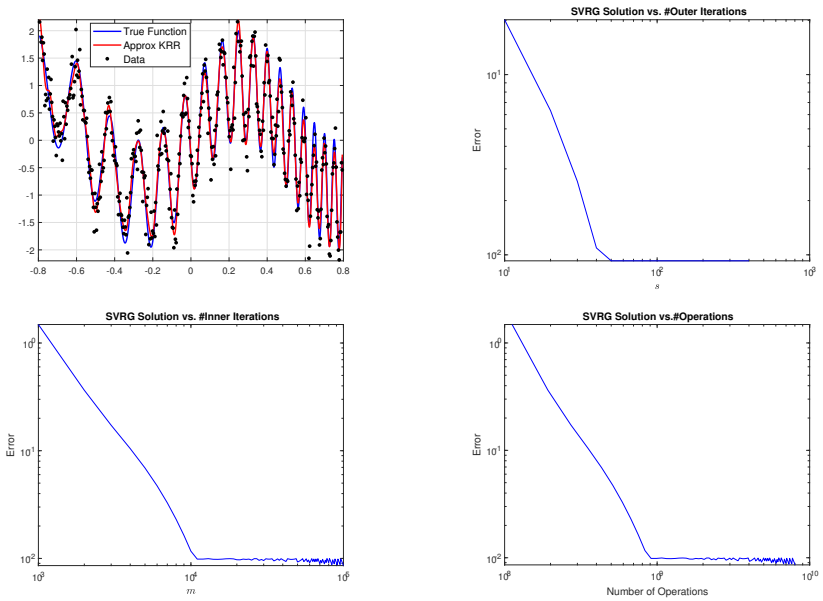

FIG. 2. *Experiment with SVRG for KRR. The target function and the samples are shown in the top left graph. The top right and bottom left graph show the test error when varying s or m (respectively). In the bottom right we show the test error as a function of the number of operations.*

$f^\star(x) = \sin(6x) + \sin(60e^x)$, i.e., $y_i = f^\star(x_i) + \epsilon_i$ with $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, 0.3^2)$. The training set consists of 400 equispaced examples on $[-1, 1]$, and we use the Gaussian kernel. The goal was to reach error $\epsilon = 10^{-2}$. The experiment was run with fixed step size of $\alpha = 10^{-4} < 1/2L_{\text{sup}}$. We varied both the value of m and s. Results are reported in Figure 2.

6. Converting SILR to finite linear least squares via sampling. All the previous algorithms we presented for solving SILR problems either assumed the chebfun model, or relied on the ability to compute the Gram matrix using an analytic formula (e.g., KRR). In practice, such formulas are not always available, and the chebfun model is implemented only in software, and even then only when the columns/rows of the quasimatrix are univariate functions.[4] Thus, a different technique is needed in order to solve SILR problems that violate these constraints. One natural approach for approximately solving a SILR problem is to discretize the infinite dimension via sampling.

For brevity, let us focus on overdetermined SILR (3.1). In order to discuss sampling, we need a coordinate representation of $\mathbf{A}$. Thus, we assume that $\mathcal{H} = L_2(\Omega, d\mu)$ for some measurable index set $\Omega \subseteq \mathbb{R}^d$ and that we have a coordinate representation $\mathbf{z}_\mathbf{A} : \Omega \to \mathbb{C}^n$ for $\mathbf{A}$ and $\mathbf{z}_\mathbf{b} : \Omega \to \mathbb{C}$ for $\mathbf{b}$. A generic approach is as follows. We first select s coordinates $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s \in \Omega$, and associated weights $w_1, \ldots, w_s \in \mathbb{R}$. We then form the row sampled matrix $\mathbf{A}_{\boldsymbol{\eta}}$ and row sampled vector $\mathbf{b}_{\boldsymbol{\eta}}$ as follows:

[4] While chebfun does support bivariate functions, it does not seem to support quasimatrices of bivariate functions.

$$\mathbf{A}_{\boldsymbol{\eta}} = \begin{bmatrix} w_1 \mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta}_1)^* \\ w_2 \mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta}_2)^* \\ \vdots \\ w_s \mathbf{z}(\boldsymbol{\eta}_s)^* \end{bmatrix}, \qquad \mathbf{b}_{\boldsymbol{\eta}} = \begin{bmatrix} w_1 \overline{\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta}_1)} \\ w_2 \overline{\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta}_2)} \\ \vdots \\ w_s \overline{\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta}_s)} \end{bmatrix}.$$

We now solve the sampled problem (which is a finite linear least squares problem):

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{A}_{\boldsymbol{\eta}}\mathbf{x} - \mathbf{b}_{\boldsymbol{\eta}}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 .$$

Solving this sampled problem can be considered as an approximation to the SILR problem, and, as explained in section 3, is the scheme used in least squares approximation of functions [11, 12] and random Fourier features [37].

To make the method concrete we need to address a couple of related questions. Given $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$, can we relate $\tilde{\mathbf{x}}$ to $\mathbf{x}^\star$? How can we select $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s \in \Omega$ and $w_1, \ldots, w_s \in \mathbb{R}$ so that $\tilde{\mathbf{x}}$ is a good enough approximate solution? Similar questions have been asked, and answered, for finite linear least squares [47, 1, 2], and various structural conditions have been suggested. The following result is similar to ones that appear in the literature on sampling finite linear least squares problems.

PROPOSITION 8. *Consider the overdetermined SILR*

$$\min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}\|_2^2$$

along with a full rank tall quasimatrix $\mathbf{A}$ *over* $\mathcal{H}$ *with* n *columns and* $\lambda \geq 0$. *Assume that* $(\|\mathbf{A}\mathbf{x}^\star - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}^\star\|_2^2)/2 \geq \lambda$. *Also assume that we are given a matrix* $\mathbf{A}_{\boldsymbol{\eta}} \in \mathbb{R}^{s\times n}$ *and a vector* $\mathbf{b}_{\boldsymbol{\eta}} \in \mathbb{R}^s$ *such that*

$$\begin{aligned} (1-\epsilon)\left(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}\|_2^2 + \lambda\right) &\leq \|\mathbf{A}_{\boldsymbol{\eta}}\mathbf{x} - \mathbf{b}_{\boldsymbol{\eta}}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 + \lambda \\ &\leq (1+\epsilon)\left(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}\|_2^2 + \lambda\right) \end{aligned} \tag{6.1}$$

for all $\mathbf{x} \in \mathbb{C}^n$. *Then,*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\tilde{\mathbf{x}}\|_2^2 \leq \frac{1+2\epsilon}{1-\epsilon}\left(\|\mathbf{A}\mathbf{x}^\star - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}^\star\|_2^2\right).$$

Proof. We have

$$\begin{aligned} \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\tilde{\mathbf{x}}\|_2^2 &\leq \frac{1}{1-\epsilon}\left(\|\mathbf{A}_{\boldsymbol{\eta}}\tilde{\mathbf{x}} - \mathbf{b}_{\boldsymbol{\eta}}\|_2^2 + \lambda\|\tilde{\mathbf{x}}\|_2^2 + \lambda\right) - \lambda \\ &\leq \frac{1}{1-\epsilon}\left(\|\mathbf{A}_{\boldsymbol{\eta}}\mathbf{x}^\star - \mathbf{b}_{\boldsymbol{\eta}}\|_2^2 + \lambda\|\mathbf{x}^\star\|_2^2 + \lambda\right) - \lambda \\ &\leq \frac{1+\epsilon}{1-\epsilon}\left(\|\mathbf{A}\mathbf{x}^\star - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}^\star\|_2^2 + \lambda\right) - \lambda \\ &\leq \frac{1+2\epsilon}{1-\epsilon}\left(\|\mathbf{A}\mathbf{x}^\star - \mathbf{b}\|_{\mathcal{H}}^2 + \lambda\|\mathbf{x}^\star\|_2^2\right), \end{aligned}$$

where the first and third inequalities use (6.1), the second inequality follows from $\tilde{\mathbf{x}}$ being the minimizer of the sampled SILR, and the last inequality uses the first assumption. ∎

Note that Proposition 8 does not require $\mathbf{A}_{\boldsymbol{\eta}}$ and $\mathbf{b}_{\boldsymbol{\eta}}$ to actually be row samples of $\mathbf{A}$ and $\mathbf{b}$.

6.1. Randomized sampling. One approach for selecting $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ and $w_1, \ldots, w_s$ is to sample $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ randomly from Ω and set the weights accordingly. The question is what distribution on Ω to use, and how to set the weights? To answer these questions, we show a general result on the number of samples s required to ensure (6.1) holds given some distribution on Ω and a specific way to set the weights. The result is based on the concept of ridge leverage scores [17, 13], which we generalize to quasimatrices (the generalization is similar to the one used in [3, 4]).

DEFINITION 9. *Let* $\mathbf{A}$ *be a quasimatrix over* $L_2(\Omega, d\mu)$ *equipped with a coordinate representation* $\mathbf{z}$*, and* $\lambda \geq 0$*. Further assume that* μ *is a probability measure for which a corresponding density* p *exists. The* λ-leverage function *of* $\mathbf{A}$ *is*

$$\tau_\lambda : \Omega \to \mathbb{R}, \quad \tau_\lambda(\boldsymbol{\eta}) \coloneqq p(\boldsymbol{\eta})\mathbf{z}(\boldsymbol{\eta})^*(\mathbf{K} + \lambda \mathbf{I}_n)^{-1}\mathbf{z}(\boldsymbol{\eta}),$$

where $\mathbf{K} = \mathbf{A}^*\mathbf{A}$ *if* $\mathbf{A}$ *is a tall quasimatrix, or* $\mathbf{K} = \mathbf{A}\mathbf{A}^*$ *if* $\mathbf{A}$ *is a wide quasimatrix.*

PROPOSITION 10 (similar to Proposition 5 in [3]). *Under the same conditions in Definition* 9*,*

$$\int_\Omega \tau_\lambda(\boldsymbol{\eta}) d\boldsymbol{\eta} = \mathbf{Tr}\left((\mathbf{K} + \lambda \mathbf{I}_n)^{-1}\mathbf{K}\right) =: s_\lambda(\mathbf{A}),$$

($s_\lambda(\mathbf{A})$ *is called the* statistical dimension *of* $\mathbf{A}$*).*

LEMMA 11 (similar to Lemma 8 in [3]). *Consider the overdetermined SILR*

$$\min \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2_{L_2(\Omega, d\mu)} + \lambda \|\mathbf{x}\|_2^2,$$

where $\mathbf{A}$ *is a tall quasimatrix* $\mathbf{A}$ *with* n *columns, and* $\lambda \geq 0$*. If* $\lambda = 0$*, further assume that* $\mathbf{A}$ *is full rank. Assume we have coordinate representation* $\mathbf{z}_\mathbf{A} : \Omega \to \mathbb{C}^n$ *for* $\mathbf{A}$ *and* $\mathbf{z}_\mathbf{b} : \Omega \to \mathbb{C}$ *for* $\mathbf{b}$*. Assume that*

$$\left\| \begin{bmatrix} \mathbf{A}^* \\ \mathbf{b}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \right\|_2 \geq \lambda\,.$$

Let $\tau_\lambda(\boldsymbol{\eta})$ *be the* λ*-leverage function of* $[\ \mathbf{A} \quad \mathbf{b}\ ]$*. Let* $\tilde{\tau} : \Omega \to \mathbb{R}$ *be a measurable function such that* $\tilde{\tau}(\boldsymbol{\eta}) \geq \tau_\lambda(\boldsymbol{\eta})$ *for all* $\boldsymbol{\eta} \in \Omega$*, and assume that* $s_{\tilde{\tau}} = \int_\Omega \tilde{\tau}(\boldsymbol{\eta}) d\boldsymbol{\eta} < \infty$*. Also, denote* $p_{\tilde{\tau}}(\boldsymbol{\eta}) = \tilde{\tau}(\boldsymbol{\eta})/s_{\tilde{\tau}}$*. Suppose we sample* $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ *using* $p_{\tilde{\tau}}$ *and set* $w_j = \sqrt{\frac{p(\boldsymbol{\eta}_j)}{s p_{\tilde{\tau}}(\boldsymbol{\eta}_j)}}$*. Given* $\epsilon \leq 1/2$ *and* $0 < \delta < 1$*, if* $s \geq \frac{8}{3} s_{\tilde{\tau}} \epsilon^{-2} \ln(16 s_\lambda([\ \mathbf{A} \quad \mathbf{b}\ ])/\delta)$*, then*

$$\begin{aligned} (1-\epsilon)\left(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2_{L_2(\Omega, d\mu)} + \lambda\|\mathbf{x}\|_2^2 + \lambda\right) &\leq \|\mathbf{A}_{\boldsymbol{\eta}}\mathbf{x} - \mathbf{b}_{\boldsymbol{\eta}}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 + \lambda \\ &\leq (1+\epsilon)\left(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2_{L_2(\Omega, d\mu)} + \lambda\|\mathbf{x}\|_2^2 + \lambda\right) \end{aligned} \tag{6.2}$$

holds with probability of at least $1 - \delta$*.*

Proof Sketch. The proof is very similar to the proof of [3, Lemma 8], so we give only a sketch of the proof. Denote

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \sqrt{\lambda}\mathbf{I}_{n+1} & \end{bmatrix}, \hat{\mathbf{A}}_{\boldsymbol{\eta}} = \begin{bmatrix} \mathbf{A}_{\boldsymbol{\eta}} & \mathbf{b}_{\boldsymbol{\eta}} \\ \sqrt{\lambda}\mathbf{I}_{n+1} & \end{bmatrix}, \hat{\mathbf{x}} = \frac{1}{\sqrt{1 + \|\mathbf{x}\|_2^2}} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix}.$$

Then, the inequality (6.2) is equivalent to

$$(1-\epsilon)\|\hat{\mathbf{A}}\hat{\mathbf{x}}\|^2_{L_2(\mathcal{X}, d\mu)} \leq \|\hat{\mathbf{A}}_{\boldsymbol{\eta}}\hat{\mathbf{x}}\|_2^2 \leq (1+\epsilon)\|\hat{\mathbf{A}}\hat{\mathbf{x}}\|^2_{L_2(\mathcal{X}, d\mu)},$$

i.e.,

$$-\epsilon\hat{\mathbf{A}}^*\hat{\mathbf{A}} \preceq \hat{\mathbf{A}}_{\boldsymbol{\eta}}^*\hat{\mathbf{A}}_{\boldsymbol{\eta}} - \hat{\mathbf{A}}^*\hat{\mathbf{A}} \preceq \epsilon\hat{\mathbf{A}}^*\hat{\mathbf{A}}\,.$$

We write $\hat{\mathbf{A}}^*\hat{\mathbf{A}} = \mathbf{V}^*\boldsymbol{\Sigma}^2\mathbf{V}$. The claim is now equivalent to

$$-\epsilon\mathbf{I}_d \preceq \boldsymbol{\Sigma}^{-1}\mathbf{V}^\star\hat{\mathbf{A}}_{\boldsymbol{\eta}}^*\hat{\mathbf{A}}_{\boldsymbol{\eta}}\mathbf{V}\boldsymbol{\Sigma}^{-1} - \mathbf{I}_d \preceq \epsilon\mathbf{I}_d\,.$$

Notice that

$$\mathbf{A}_{\boldsymbol{\eta}}^*\mathbf{A}_{\boldsymbol{\eta}} = \sum_{j=1}^{s} w_j^2\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta}_j)\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta}_j)^*,\ \mathbf{b}_{\boldsymbol{\eta}}^*\mathbf{b}_{\boldsymbol{\eta}} = \sum_{j=1}^{s} w_j^2\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta}_j)\mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta}_j)^*\,.$$

It can be seen that $\mathbf{z}(\boldsymbol{\eta}) = \left[\begin{smallmatrix}\mathbf{z}_{\mathbf{A}}(\boldsymbol{\eta})\\ \mathbf{z}_{\mathbf{b}}(\boldsymbol{\eta})\end{smallmatrix}\right]$ is a coordinate representation for the quasimatrix part of $\hat{\mathbf{A}}$. Let

$$\mathbf{S}_j = \frac{p(\boldsymbol{\eta}_j)}{p_{\tilde{\tau}}(\boldsymbol{\eta}_j)}\boldsymbol{\Sigma}^{-1}\mathbf{V}^*\mathbf{z}(\boldsymbol{\eta}_j)\mathbf{z}(\boldsymbol{\eta}_j)^*\mathbf{V}\boldsymbol{\Sigma}^{-1}\,.$$

It is possible to show that $\mathbb{E}[\mathbf{S}_j^2] \preceq s_{\tilde{\tau}}\mathbb{E}[\mathbf{S}_j]$ and $\mathbf{Tr}\left(\mathbb{E}[\mathbf{S}_j]\right) = s_{\tilde{\tau}} \cdot s_\lambda(\hat{\mathbf{K}})$. The claim follows from [46, Corollary 7.3.3]. ∎

A similar result appears in [12] for truncated and conditioned least squares approximations of functions, however, without any ridge term. The ridge leverage function can be viewed as a variant of the Christoffel function [35] from the literature on orthogonal polynomials and approximation theory [35, 30, 42, 9].

One natural strategy for selecting the $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ is to sample them using the distribution μ. We call this strategy *natural sampling*. Using Lemma 11 we can give a bound on the number of samples needed when sampling $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ using this strategy and setting all the weights to $\sqrt{1/s}$.

PROPOSITION 12. *Let $\tau_\lambda(\boldsymbol{\eta})$ be the λ-leverage function of $[\ \mathbf{A} \quad \mathbf{b}\ ]$. Suppose that $M_\lambda = M_\lambda([\ \mathbf{A} \quad \mathbf{b}\ ]) \coloneqq \sup_{\boldsymbol{\eta}\in\Omega}\tau_\lambda(\boldsymbol{\eta})/p(\boldsymbol{\eta})$ is finite. Suppose we sample $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ using μ, and set $w_j = \sqrt{1/s}$ for $j = 1, \ldots, s$. If*

$$s \geq \frac{8}{3}M_\lambda\epsilon^{-2}\ln(16s_\lambda\left(\left[\ \mathbf{A} \quad \mathbf{b}\ \right]\right)/\delta),$$

then (6.2) *holds with probability of at least $1-\delta$.*

Proof. Let us define $\tilde{\tau}(\boldsymbol{\eta}) = M_\lambda p(\boldsymbol{\eta})$. Notice that $s_{\tilde{\tau}} = M_\lambda$ and that $p_{\tilde{\tau}}(\boldsymbol{\eta}) = p(\boldsymbol{\eta})$. Thus, the conditions of Proposition 11 hold if we sample using $p(\cdot)$ and set the weights to $\sqrt{1/s}$, and the claim follows. ∎

The quantity M_λ is a generalization of the concept of matrix coherence [5] to quasimatrices. A similar quantity appears in [11] in the context of function approximation using sampling. When using natural sampling, the number of samples required for (6.2) to hold with high probability depends on the coherence of the quasimatrix, which can be large. Sampling using the ridge leverage scores, often referred to as *leverage score sampling*, yields a better bound since $s_\lambda([\ \mathbf{A} \quad \mathbf{b}\ ]) \leq M_\lambda([\ \mathbf{A} \quad \mathbf{b}\ ])$.

Of course, it is not simple to sample using the ridge leverage function. Cohen and Migliorati suggested a method from leverage score sampling when $\lambda = 0$ [12]. Their method is based on sequential conditional sampling, where individual coordinates are sampled using either rejection sampling or inversion transform sampling. An alternative approach is to find some simple and easy way to sample the upper bound on τ_λ. For this to be worthwhile, the bound has to be tighter than the bound $\tau_\lambda(\boldsymbol{\eta}) \leq M_\lambda p(\boldsymbol{\eta})$ used in Proposition 12. This approach is used in [3, 4].

6.2. Quadrature sampling. In this section, we discuss deterministic sampling using quadrature formulas. For simplicity, we assume that $\Omega = [-1, 1]$ and that μ is the Lebesgue measure on $[-1, 1]$. Accordingly, the sampling scheme is based on the Gauss–Legendre quadrature. Higher dimensional domains can be handled via tensoring the quadrature. We also assume that $\lambda > 0$. Let $\mathbf{z_A} : \mathbb{R} \to \mathbb{C}^n$ be a coordinate representation of $\mathbf{A}$. We can write

$$\mathbf{A}^*\mathbf{A} = \int_{-1}^{1} \mathbf{z_A}(\eta)\mathbf{z_A}(\eta)^* d\eta\,.$$

Furthermore, for every $\mathbf{x} \in \mathbb{R}^n$

$$\|\mathbf{Ax}\|^2_{L_2([-1,1],d\mu)} = \int_{-1}^{1} \mathbf{x}^{\mathrm{T}}\mathbf{z_A}(\eta)\mathbf{z_A}(\eta)^*\mathbf{x} d\eta = \int_{-1}^{1} \left|\mathbf{z_A}(\eta)^*\mathbf{x}\right|^2 d\eta\,.$$

Let $\mathbf{z_b} : \mathbb{R} \to \mathbb{C}$ be a coordinate representation of $\mathbf{b}$. Then,

$$\mathbf{A}^*\mathbf{b} = \int_{-1}^{1} \mathbf{z_A}(\eta)\overline{\mathbf{z_b}(\eta)} d\eta, \quad \|\mathbf{b}\|^2_{L_2([-1,1],d\mu)} = \int_{-1}^{1} \left|\mathbf{z_b}(\eta)\right|^2 d\eta\,.$$

We conclude that the overdetermined SILR can be written as an integral form

$$\int_{-1}^{1} f_{\mathbf{x}}(\eta) d\eta = \|\mathbf{Ax} - \mathbf{b}\|^2_{L_2([-1,1],d\mu)} + \lambda\|\mathbf{x}\|_2^2, \tag{6.3}$$

where

$$f_{\mathbf{x}}(\eta) = \left|\mathbf{z_A}(\eta)^*\mathbf{x} - \mathbf{z_b}(\eta)\right|^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2\,.$$

The underlying idea is to approximate the integral in (6.3) using the Gauss–Legendre quadrature. For a given $\epsilon \in (0, 1)$, our algorithm sets the nodes $\eta_1, \ldots, \eta_s \in [-1, 1]$ to be the Gauss–Legendre quadrature nodes, and sets the weights $w_1, \ldots, w_s > 0$ so that their squares are the Gauss–Legendre quadrature weights. We set s to be large enough so that

$$\frac{\left|\int_{-1}^{1} f_{\mathbf{x}}(\eta) d\eta - \sum_{j=1}^{s} w_j^2 f_{\mathbf{x}}(\eta_j)\right|}{\int_{-1}^{1} f_{\mathbf{x}}(\eta) d\eta + \lambda} \le \epsilon\,. \tag{6.4}$$

Once $\eta_1, \ldots, \eta_s$ and the weights $w_1, \ldots, w_s$ are computed, we can define $\mathbf{A}_{\boldsymbol{\eta}}$ and $\mathbf{b}_{\boldsymbol{\eta}}$ as before. We have

$$\sum_{j=1}^{s} w_j^2 f_{\mathbf{x}}(\eta_j) = \|\mathbf{A}_{\boldsymbol{\eta}}\mathbf{x} - \mathbf{b}_{\boldsymbol{\eta}}\|_2^2 + \lambda\|\mathbf{x}\|_2^2,$$

so if (6.4) holds then (6.1) holds (with $\mathcal{H} = L_2([-1, 1], d\mu)$), and we can apply Proposition 8.

To determine how many quadrature nodes s are needed so that (6.4) holds, we can apply the following theorem, which is a modified version of [40, Theorem 11] for the function $g_{\mathbf{x}}(\eta) \coloneqq f_{\mathbf{x}}(\eta)/(\int_{-1}^{1} f_{\mathbf{x}}(\eta)d\eta + \lambda)$. Since the proof is a simple modification of the proof [40, Theorem 11], we omit it.

THEOREM 13. *Let E be the (Bernstein) ellipse in the complex plane with foci ± 1 that passes through i, and let $\rho = 1 + \sqrt{2}$. Assume that both real and imaginary parts of $\mathbf{z_A}(\cdot)_i$, $i = 1, \ldots, n$, and $\mathbf{z_b}(\cdot)$ are analytic on $\mathbb{R}$, and denote their analytic continuations by $\hat{\mathbf{z}}_\mathbf{A}(\cdot)$ and $\hat{\mathbf{z}}_\mathbf{b}(\cdot)$ correspondingly. Denote*

$$M_\mathbf{A} \coloneqq \sup_{\eta \in E} \|\hat{\mathbf{z}}_\mathbf{A}(\eta)\|_\infty, \quad M_\mathbf{b} \coloneqq \sup_{\eta \in E} \|\hat{\mathbf{z}}_\mathbf{b}(\eta)\|_\infty .$$

Then, given a small ϵ, for

$$s \geq \frac{\ln\left(8(\lambda^{-1}(nM_\mathbf{A}^2 + M_\mathbf{b}^2) + 1)\right) - \ln \epsilon - \ln \sqrt{2}}{2 \ln(1 + \sqrt{2})} + 1$$

we have

$$\left| \int_{-1}^{1} g_\mathbf{x}(\eta) d\eta - \sum_{j=1}^{s} w_j g_\mathbf{x}(\eta_j) \right| \leq \epsilon,$$

where $\eta_1, \ldots \eta_s$ are chosen to be the Gauss–Legendre quadrature nodes, and $w_1^2, \ldots, w_s^2$ are the Gauss–Legendre quadrature weights.

Remark 14. For $\eta \in E$, we denote $c_\mathbf{x} = \|\mathbf{Ax} - \mathbf{b}\|^2_{L_2([-1,1], d\mu)} + \lambda \|\mathbf{x}\|_2^2 + \lambda$ and bound $g_\mathbf{x}(\eta)$ as follows,

$$\begin{aligned}
g_\mathbf{x}(\eta) &= \frac{1}{c_\mathbf{x}} \left(|\mathbf{z_A}(\eta)^* \mathbf{x} - \mathbf{z_b}(\eta)|^2 + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \right) \\
&= \frac{1}{c_\mathbf{x}} \left(\left| \begin{bmatrix} \mathbf{z_A}(\eta)^* & \mathbf{z_b}(\eta) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right|^2 + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \right) \\
&= \frac{1}{c_\mathbf{x}} \left(\left| \begin{bmatrix} \mathbf{z_A}(\eta)^* & \mathbf{z_b}(\eta) \end{bmatrix} \hat{\mathbf{K}}^{-1/2} \hat{\mathbf{K}}^{1/2} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right|^2 + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \right) \\
&\leq \frac{1}{c_\mathbf{x}} \left(\begin{bmatrix} \mathbf{z_A}(\eta)^* & \mathbf{z_b}(\eta) \end{bmatrix} \hat{\mathbf{K}}^{-1} \begin{bmatrix} \mathbf{z_A}(\eta) \\ \mathbf{z_b}(\eta)^* \end{bmatrix} \cdot \left(\begin{bmatrix} \mathbf{x} & -1 \end{bmatrix} \hat{\mathbf{K}} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right) + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \right) \\
&= \begin{bmatrix} \mathbf{z_A}(\eta)^* & \mathbf{z_b}(\eta) \end{bmatrix} \hat{\mathbf{K}}^{-1} \begin{bmatrix} \mathbf{z_A}(\eta) \\ \mathbf{z_b}(\eta)^* \end{bmatrix} + \frac{1}{c_\mathbf{x}} \cdot \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \\
&\leq \lambda^{-1} (\|\mathbf{z_A}(\eta)\|_2^2 + |\mathbf{z_b}(\eta)|^2) + \frac{1}{2} \\
&\leq \lambda^{-1} (nM_\mathbf{A}^2 + M_\mathbf{b}^2) + \frac{1}{2},
\end{aligned}$$

where $\hat{\mathbf{K}} = \left[\begin{smallmatrix} \mathbf{A}^*\mathbf{A} & \mathbf{A}^*\mathbf{b} \\ \mathbf{b}^*\mathbf{A} & \mathbf{b}^*\mathbf{b} \end{smallmatrix} \right] + \lambda \mathbf{I}_{n+1}$ and in the first inequality we use the Cauchy–Schwarz inequality. Theorem 13 yields

$$\frac{\left| \int_{-1}^{1} f_\mathbf{x}(\eta) d\eta - \sum_{j=1}^{s} w_j^2 f_\mathbf{x}(\eta_j) \right|}{\int_{-1}^{1} f_\mathbf{x}(\eta) d\eta + \lambda} = \left| \int_{-1}^{1} g_\mathbf{x}(\eta) d\eta - \sum_{j=1}^{s} w_j^2 g_\mathbf{x}(\eta_j) \right| \leq \epsilon .$$

We can generalize the above theorem, which is specific for $\Omega = [-1, 1]$, to complex sets and/or high dimensional sets with a variety of probability measures on them, as done in [40].

6.3. Numerical example. We illustrate both sampling approaches, randomized and quadrature, on a small numerical example. Consider trying to approximate on $[-1, 1]$ the Runge function using a polynomial of degree 39. We use the Chebyshev basis, i.e.,

$$\mathbf{A} = \begin{bmatrix} T_0 & T_1 & \cdots & T_{39} \end{bmatrix}, \quad \mathbf{b} = \left[\frac{1}{1 + 25x^2} \right]$$

with $\lambda = 10^{-4}$.

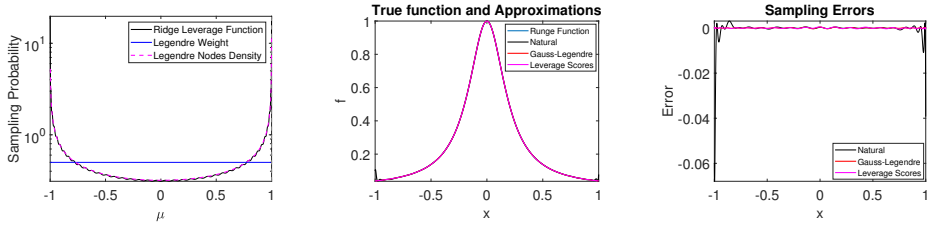

FIG. 3. *Numerical illustration: approximating the Runge function using various sampling methods.*

The leftmost graph in Figure 3 shows the ridge leverage density of $[\ \mathbf{A} \quad \mathbf{b}\ ]$, and compares it to the uniform density. We also plot the density of the limiting distribution of Legendre nodes. We see very close alignment between the ridge leverage score density and the density of the Legendre nodes. We note that in this case $s_\lambda = 39.99$. In contrast $M_\lambda = 798.28$, and thus we will need about 95% fewer samples when using leverage score sampling when compared to natural sampling. However, even for $\epsilon = 0.01$, the number of samples required for randomized sampling is huge. In contrast, for $\epsilon = 0.01$ only $s = 73$ features are required using quadrature features. Nevertheless, in the experiments we use $s = 100$ for both randomized and quadrature sampling.

The middle and rightmost graphs in Figure 3 show the function approximation (on the left), and the error in approximating the function (on the right). We use both natural sampling and leverage score sampling, where we used inverse transform sampling for leverage score sampling. With $s = 100$, using quadrature sampling and leverage score sampling we get small errors: the maximum absolute error is 4.48×10^{-4} for quadrature sampling, and 9.82×10^{-4} for leverage score sampling. Natural sampling has a large error near the boundary of $[-1, 1]$ (as expected), and the maximum absolute error is 0.0581.

7. Conclusions and future work. In this paper, we gave an algebraic framework for working with quasimatrices and explored the use of this framework to solve SILR problems, i.e., regression problems where the system's matrix has an infinite number of rows or columns. We discussed various applications, such as function approximation and supervised learning (using KRR). We offered several classes of algorithms for solving SILR problems: direct methods and iterative methods (generalizing known iterative methods such as LSMR as an example of a Krylov subspace method and SVRG as an example of a stochastic optimization method). Finally, motivated by recent research on RandNLA methods for solving finite linear least squares problems, we explored the use of sampling techniques to approximate the solution of an SILR, where sampling can be either randomized or deterministic. Possible future directions are to further leverage advanced randomized linear algebra methods, such as sketching, whereas the main challenge is in how to generate a random quasimatrix from the correct distribution. Another interesting idea is to generalize the Batson–Spielman–Srivastava process for iteratively building a spectral approximation of a matrix using columns samples [6] to quasimatrices.

Appendix A. SVRG with integrable sums. The usual SVRG algorithm [23, 48] is defined for objective functions that have a finite sum structure, i.e.,

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})\,. \tag{A.1}$$

Here we propose a variant of the algorithm designed for objective functions that can be written as an integral. Let μ be some probability measure on a measurable index set, Ω. Our variant of SVRG is designed for functions that can be written as

$$f(\mathbf{x}) = \int_{\Omega} f_{\boldsymbol{\eta}}(\mathbf{x}) d\mu(\boldsymbol{\eta}), \tag{A.2}$$

where the integral should be interepted as a Lebesgue integral. Notice that (A.1) is a special case of (A.2): $\Omega = \{1, \ldots, n\}$ and $\mu(A) = \frac{|A|}{n}$. The proposed algorithm is summarized in Algorithm A.1.

Algorithm A.1. SVRG for integrable objective functions.

Inputs: initial $\tilde{\mathbf{x}}_0$, learning rate α, frequency m
Iterate: for $s = 1, 2, \ldots$
  $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_{s-1}$
  $\tilde{\mu} = \nabla \int_{\Omega} f_{\boldsymbol{\eta}}(\tilde{\mathbf{x}}) d\mu(\boldsymbol{\eta}) = \nabla f(\tilde{\mathbf{x}})$
  $\mathbf{x}_0 = \tilde{\mathbf{x}}$
  Iterate: for $k = 1, 2, \ldots, m$
    sample $\boldsymbol{\eta}_k$ according to the probability of $\boldsymbol{\eta}$ and update
    $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha\left(\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) + \tilde{\mu}\right)$
  end
option I: set $\tilde{\mathbf{x}}_s = \mathbf{x}_m$
option II: set $\tilde{\mathbf{x}}_s = \frac{1}{m}\sum_{k=1}^{m} \mathbf{x}_k$
end

As in common convex optimization, certain assumptions must be made in order for the algorithm to converge. We prove that Algorithm A.1 converges and analyze the convergence rate, when the following assumptions hold. We start with assumptions that are analogous to the assumptions in finite sum SVRG, which we already mentioned in section 5.2.

Assumption 15. For all $\boldsymbol{\eta} \in \Omega$, $\nabla f_{\boldsymbol{\eta}}(\mathbf{x})$ is Lipschitz continuous, i.e., there exists $L_{\boldsymbol{\eta}} > 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$\|\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{y})\| \le L_{\boldsymbol{\eta}} \|\mathbf{x} - \mathbf{y}\|\,.$$

Assumption 16. Suppose that $f(\mathbf{x})$ is strongly convex, i.e., there exist $\gamma > 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{x}) - f(\mathbf{y}) \ge \frac{\gamma}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \nabla f(\boldsymbol{y})^{\mathrm{T}}(\mathbf{x} - \mathbf{y})\,.$$

Next, we list assumptions that trivially hold for the finite case but are required for the continuous case.

Assumption 17. The equality $\nabla f(\mathbf{x}) = \int_{\Omega} \nabla f_{\boldsymbol{\eta}}(\mathbf{x}) d\mu(\boldsymbol{\eta})$ holds.

Suppose $\Omega = \mathbb{R}^d$ and $f_{\boldsymbol{\eta}}(\mathbf{x}), \nabla f_{\boldsymbol{\eta}}(\mathbf{x}) \in L_1(\Omega)$ with respect to η. Then, Assumption 17 holds from the Leibniz integral rule.

Assumption 18. $L_{\sup} \coloneqq \sup_{\boldsymbol{\eta}\in\Omega} L_{\boldsymbol{\eta}} < \infty$.

Assumptions 17 and 18 imply that $\nabla f(\mathbf{x})$ is Lipschitz continuous with Lipschitz constant $L \le L_{\sup}$. Note that for the finite sum case, Assumptions 17 and 18 hold trivially, but this is no longer the case in the integrable case.

COROLLARY 19. *If Assumptions* 17, 18 *hold, then we can make Assumptions* 15, 16 *hold for the continuous case.*

We now analyze Algorithm 17. The analysis follows the analysis in [23, 48] quite closely, making adjustments where necessary for integrals instead of sums, and using the additional assumptions when needed.

LEMMA 20. *Suppose Assumptions* 15, 17, 18 *hold. Let* $\mathbf{x}^\star = \arg\min_{\mathbf{x}} f(\mathbf{x})$ *and* $L_{\sup} = \sup_{\boldsymbol{\eta}\in\Omega} L_{\boldsymbol{\eta}}$. *Then*

$$\int_\Omega \|\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)\|_2^2 d\mu(\boldsymbol{\eta}) \le 2L_{\sup}\left(f(\mathbf{x}) - f(\mathbf{x}^\star)\right).$$

Proof. Given any $\boldsymbol{\eta} \in \Omega$, let

$$g_{\boldsymbol{\eta}}(\mathbf{x}) = f_{\boldsymbol{\eta}}(\mathbf{x}) - f_{\boldsymbol{\eta}}(\mathbf{x}^\star) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)^T(\mathbf{x} - \mathbf{x}^\star).$$

It can be seen that $\nabla g_{\boldsymbol{\eta}}(\mathbf{x}^\star) = 0$ and, hence, $\mathbf{x}^\star = \arg\min_{\mathbf{x}} g_{\boldsymbol{\eta}}(\mathbf{x})$. Moreover, from Assumption 15, $\nabla g_{\boldsymbol{\eta}}(\mathbf{x}) = \nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)$ is Lipschitz continuous with constant $L_{\boldsymbol{\eta}}$. This yields

$$g_{\boldsymbol{\eta}}(\mathbf{x}) - g_{\boldsymbol{\eta}}(\mathbf{y}) \le \frac{L_{\boldsymbol{\eta}}}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \nabla g_{\boldsymbol{\eta}}(\mathbf{y})^{\mathrm{T}}(\mathbf{x} - \mathbf{y})$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ (see [29, Lemma 1.2.3]). Replacing $\mathbf{x}$ with $\mathbf{x} - \frac{1}{L_{\boldsymbol{\eta}}}\nabla g_{\boldsymbol{\eta}}(\mathbf{x})$ and $\mathbf{y}$ with $\mathbf{x}$, gives

$$g_{\boldsymbol{\eta}}\left(\mathbf{x} - \frac{1}{L_{\boldsymbol{\eta}}}\nabla g_{\boldsymbol{\eta}}(\mathbf{x})\right) \le g_{\boldsymbol{\eta}}(\mathbf{x}) - \frac{1}{2L_{\boldsymbol{\eta}}}\|\nabla g_{\boldsymbol{\eta}}(\mathbf{x})\|_2^2.$$

Since $\min_{\mathbf{x}} g_{\boldsymbol{\eta}}(\mathbf{x}) = g_{\boldsymbol{\eta}}(\mathbf{x}^\star) = 0$, we have $0 \le g_{\boldsymbol{\eta}}(\mathbf{x} - \nabla g_{\boldsymbol{\eta}}(\mathbf{x})/L_{\boldsymbol{\eta}})$, which implies

$$\frac{1}{2L_{\boldsymbol{\eta}}}\|\nabla g_{\boldsymbol{\eta}}(\mathbf{x})\|_2^2 \le g_{\boldsymbol{\eta}}(\mathbf{x}). \tag{A.3}$$

Substituting the definition of g gives

$$\|\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)\|_2^2 \le 2L_{\boldsymbol{\eta}}\left(f_{\boldsymbol{\eta}}(\mathbf{x}) - f_{\boldsymbol{\eta}}(\mathbf{x}^\star) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)^T(\mathbf{x} - \mathbf{x}^\star)\right).$$

Now, by taking an integral over Ω, we have

$$\begin{aligned}\int_\Omega \|\nabla f_{\boldsymbol{\eta}}(\mathbf{x}) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)\|_2^2 d\mu(\boldsymbol{\eta}) &\le 2L_{\sup}\int_\Omega f_{\boldsymbol{\eta}}(\mathbf{x}) - f_{\boldsymbol{\eta}}(\mathbf{x}^\star) - \nabla f_{\boldsymbol{\eta}}(\mathbf{x}^\star)^{\mathrm{T}}(\mathbf{x} - \mathbf{x}^\star) d\mu(\boldsymbol{\eta}) \\ &= 2L_{\sup}\left(f(\mathbf{x}) - f(\mathbf{x}^\star) - \nabla f(\mathbf{x}^\star)^{\mathrm{T}}(\mathbf{x} - \mathbf{x}^\star)\right) = 2L_{\sup}\left(f(\mathbf{x}) - f(\mathbf{x}^\star)\right),\end{aligned}$$

where in the first inequality we use Assumption 18, in the second equality we use Assumption 17, and the last equality is due to the fact that $\nabla f(\mathbf{x}^\star) = 0$. ∎

COROLLARY 21. *Denote* $\mathbf{v}_k = \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) + \tilde{\mu}$. *Then, conditioned on* $\mathbf{x}_{k-1}$ *we have*

$$\mathbb{E}\|\mathbf{v}_k\|_2^2 \le 4L_{\sup}\left(f(\mathbf{x}_{k-1}) - 2f(\mathbf{x}^\star) + f(\tilde{\mathbf{x}})\right).$$

Proof. Conditioned on $\mathbf{x}_{k-1}$, taking expectation with respect to $\boldsymbol{\eta}_k$ gives $\mathbb{E}[\nabla f_{\eta_k}(\mathbf{x}_{k-1})] = \nabla f(\mathbf{x}_{k-1})$. Similarly, $\mathbb{E}[\nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}})] = \nabla f(\tilde{\mathbf{x}})$. Therefore

$$\mathbb{E}\left[\mathbf{v}_k\right] = \mathbb{E}\left[\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) + \tilde{\mu}\right] = \nabla f(\mathbf{x}_{k-1})\,. \tag{A.4}$$

Now,

$$\begin{aligned}
\mathbb{E}\|\mathbf{v}_k\|_2^2 &= \mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) + \tilde{\mu} + \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)\|_2^2 \\
&\le 2\mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)\|_2^2 + 2\mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star) - \nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) + \tilde{\mu}\|_2^2 \\
&= 2\mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)\|_2^2 \\
&\quad + 2\mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star) - \mathbb{E}\left[\nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)\right]\|_2^2 \\
&\le 2\mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}_{k-1}) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)\|_2^2 + 2\mathbb{E}\|\nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}}) - \nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)\|_2^2 \\
&\le 4L_{\sup}\left(f(\mathbf{x}_{k-1}) - 2f(\mathbf{x}^\star) + f(\tilde{\mathbf{x}})\right),
\end{aligned}$$

where in the first inequality we use $\|a+b\|^2 \le 2(\|a\|^2 + \|b\|^2)$. The second equality uses $\mathbb{E}[\nabla f_{\boldsymbol{\eta}_k}(\tilde{\mathbf{x}})] = \tilde{\mu}$, $\mathbb{E}[\nabla f_{\boldsymbol{\eta}_k}(\mathbf{x}^\star)] = \nabla f(\mathbf{x}^\star) = 0$. The second inequality uses the fact that for any $\xi \in \mathbb{R}^d$: $\mathbb{E}\|\xi - \mathbb{E}\xi\|_2^2 = \mathbb{E}\|\xi\|_2^2 - \|\mathbb{E}\xi\|_2^2 \le \mathbb{E}\|\xi\|_2^2$. In the last inequality we use Lemma 20. ∎

Now we can proceed to prove the main theorem.

THEOREM 22. *Suppose Assumptions* 16, 17, 18 *hold, and let* $\mathbf{x}^\star = \arg\min_{\mathbf{x}} f(\mathbf{x})$ *and* $L_{\sup} = \sup_{\boldsymbol{\eta}\in\Omega} L_{\boldsymbol{\eta}}$. *In addition, assume that there exists* $0 < \alpha < \frac{1}{2L_{\sup}}$, *a sufficiently large* m *such that*

$$\rho = \frac{1}{\gamma\alpha(1-2L_{\sup}\alpha)m} + \frac{2L_{\sup}\alpha}{(1-2L_{\sup}\alpha)} < 1\,.$$

Then SVRG (Algorithm A.1*) with option* II *has geometric convergence in expectation:*

$$\mathbb{E}\left[f(\tilde{\mathbf{x}}_s)\right] - f(\mathbf{x}^\star) \le \rho^s\left(f(\tilde{\mathbf{x}}_0) - f(\mathbf{x}^\star)\right)\,.$$

Proof. From Assumption 16, and using (A.4), we have

$$\begin{aligned}
f(\mathbf{x}^\star) - f(\mathbf{x}_{k-1}) &\ge -\nabla f(\mathbf{x}_{k-1})^{\mathrm{T}}(\mathbf{x}_{k-1} - \mathbf{x}^\star), \\
f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) &\ge -\alpha\mathbb{E}\left[\mathbf{v}_k\right]^{\mathrm{T}}\mathbf{v}_k\,.
\end{aligned} \tag{A.5}$$

Subtracting these inequalities yields

$$-\nabla f(\mathbf{x}_{k-1})^{\mathrm{T}}(\mathbf{x}_{k-1} - \mathbf{x}^\star) \le f(\mathbf{x}^\star) - f(\mathbf{x}_k) - \alpha\mathbb{E}\left[\mathbf{v}_k\right]^{\mathrm{T}}\mathbf{v}_k\,. \tag{A.6}$$

Thus, we have

$$\begin{aligned}
\mathbb{E}\|\mathbf{x}_k - \mathbf{x}^\star\|_2^2 &= \mathbb{E}\|\mathbf{x}_{k-1} - \alpha\mathbf{v}_k - \mathbf{x}^\star\|_2^2 \\
&= \|\mathbf{x}_{k-1} - \mathbf{x}^\star\|_2^2 - 2\alpha\nabla f(\mathbf{x}_{k-1})^{\mathrm{T}}(\mathbf{x}_{k-1} - \mathbf{x}^\star) + \alpha^2\mathbb{E}\|\mathbf{v}_k\|_2^2 \\
&\le \|\mathbf{x}_{k-1} - \mathbf{x}^\star\|_2^2 - 2\alpha\left(f(\mathbf{x}_k) - f(\mathbf{x}^\star)\right) - 2\alpha^2\mathbb{E}\left[\mathbf{v}_k\right]^{\mathrm{T}}\mathbf{v}_k \\
&\quad + 4L_{\sup}\alpha^2\left(f(\mathbf{x}_{k-1}) - 2f(\mathbf{x}^\star) + f(\tilde{\mathbf{x}})\right),
\end{aligned}$$

where the equality uses (A.4) and the inequality uses (A.6) and Corollary 21.

Now, consider a fixed stage s, such that $\mathbf{x}_0 = \tilde{\mathbf{x}} = \tilde{\mathbf{x}}_{s-1}$ and $\tilde{\mathbf{x}}_s = \frac{1}{m}\sum_{k=1}^{m} \mathbf{x}_k$. By summing the previous inequality over $k = 1, \ldots, m$ and taking expectation with respect to the history of the random variables $\eta_1, \ldots, \eta_m$, we obtain

$$
\begin{aligned}
\mathbb{E}\|\mathbf{x}_m - \mathbf{x}^\star\|^2 &\le \|\mathbf{x}_0 - \mathbf{x}^\star\|_2^2 - 2\alpha \sum_{k=1}^{m} \left(\mathbb{E}\left[f(\mathbf{x}_k)\right] - f(\mathbf{x}^\star)\right) - 2\alpha^2 \sum_{k=1}^{m} \|\mathbb{E}\left[\mathbf{v}_k\right]\|_2^2 \\
&\quad + 4L_{\sup}\alpha^2 \sum_{k=1}^{m} \left(\mathbb{E}\left[f(\mathbf{x}_{k-1})\right] - f(\mathbf{x}^\star)\right) + 4L_{\sup}\alpha^2 m \left(f(\tilde{\mathbf{x}}) - f(\mathbf{x}^\star)\right) \\
&\le \|\tilde{\mathbf{x}} - \mathbf{x}^\star\|_2^2 - 2\alpha \sum_{k=1}^{m} \left(\mathbb{E}\left[f(\mathbf{x}_k)\right] - f(\mathbf{x}^\star)\right) - 2\alpha^2 \sum_{k=1}^{m} \|\mathbb{E}\left[\mathbf{v}_k\right]\|_2^2 \\
&\quad + 4L_{\sup}\alpha^2 \sum_{k=1}^{m} \left(\mathbb{E}\left[f(\mathbf{x}_k)\right] - f(\mathbf{x}^\star)\right) + 4L_{\sup}\alpha^3 \sum_{k=1}^{m} \|\mathbb{E}\left[\mathbf{v}_k\right]\|_2^2 + 4L_{\sup}\alpha^2 m \left(f(\tilde{\mathbf{x}}) - f(\mathbf{x}^\star)\right) \\
&\le \|\tilde{\mathbf{x}} - \mathbf{x}^\star\|_2^2 - 2\alpha\left(1 - 2L_{\sup}\alpha\right) \sum_{k=1}^{m} \left(\mathbb{E}\left[f(\mathbf{x}_k)\right] - f(\mathbf{x}^\star)\right) + 4L_{\sup}\alpha^2 m \left(f(\tilde{\mathbf{x}}) - f(\mathbf{x}^\star)\right) \\
&\le \left(\frac{2}{\gamma} + 4L_{\sup}\alpha^2 m\right) \left(f(\tilde{\mathbf{x}}_{s-1}) - f(\mathbf{x}^\star)\right) - 2\alpha\left(1 - 2L_{\sup}\alpha\right) \sum_{k=1}^{m} \left(\mathbb{E}\left[f(\mathbf{x}_k)\right] - f(\mathbf{x}^\star)\right) .
\end{aligned}
$$

The second inequality is due to the strong convexity in (A.5), and the third inequality uses the assumption $2L_{\sup}\alpha < 1$ such that $(4L_{\sup}\alpha^3 - 2\alpha^2)\|\mathbb{E}[\mathbf{v}_k]\|_2^2 \le 0$. The last inequality uses Assumption 16 with $\mathbf{x}$ replaced by $\tilde{\mathbf{x}}$ and $\mathbf{y}$ replaced by $\mathbf{x}^\star$. In addition, $f(\tilde{\mathbf{x}}_s) \le \frac{1}{m}\sum_{k=1}^{m} f(\mathbf{x}_k)$ due to the convexity of f. Therefore, we obtain

$$
2\alpha\left(1 - 2L_{\sup}\alpha\right) m \left(\mathbb{E}\left[f(\tilde{\mathbf{x}}_s)\right] - f(\mathbf{x}^\star)\right) \le \left(\frac{2}{\gamma} + 4L_{\sup}\alpha^2 m\right) \left(f(\tilde{\mathbf{x}}_{s-1}) - f(\mathbf{x}^\star)\right) .
$$

Dividing both sides of the above inequality by $2\alpha(1 - 2L_{\sup}\alpha)m$ gives

$$
\mathbb{E}\left[f(\tilde{\mathbf{x}}_s)\right] - f(\mathbf{x}^\star) \le \rho^s \left(f(\tilde{\mathbf{x}}_0) - f(\mathbf{x}^\star)\right) . \qquad \square
$$

REFERENCES

[1] H. Avron, K. L. Clarkson, and D. P. Woodruff, *Faster kernel ridge regression using sketching and preconditioning*, SIAM J. Matrix. Anal. Appl., 38 (2017), pp. 1116–1138.

[2] H. Avron, K. L. Clarkson, and D. P. Woodruff, *Sharper bounds for regularized data fitting*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017), K. Jansen, J. D. P. Rolim, D. Williamson, and S. S. Vempala, eds., LIPIcs Leibniz Int. Proc. Inform., 81, Wadern Germany, 2017, 27, https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2017.27.

[3] H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh, *Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees*, in Proceedings of the 34th International Conference on Machine Learning, Vol. 70, International Machine Learning Society, Stroudsburg, PA, 2017, pp. 253–262.

[4] H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh, *A universal sampling method for reconstructing signals with simple Fourier transforms*, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, ACM, New York, 2019, pp. 1051–1063.

[5] H. Avron, P. Maymounkov, and S. Toledo, *Blendenpik: Supercharging LAPACK's least-squares solver*, SIAM J. Sci. Comput., 32 (2010), pp. 1217–1236, https://doi.org/10.1137/090767911.

[6] J. Batson, D. A. Spielman, and N. Srivastava, *Twice-Ramanujan sparsifiers*, SIAM Rev., 56 (2014), pp. 315–334, https://doi.org/10.1137/130949117.

[7] Z. Battles, *Numerical Linear Algebra for Continuous Functions*, PhD thesis, University of Oxford, Oxford, 2005.
[8] Z. Battles and L. N. Trefethen, *An extension of MATLAB to continuous functions and operators*, SIAM J. Sci. Comput., 25 (2004), pp. 1743–1770.
[9] P. Borwein and T. Erdélyi, *Polynomials and Polynomial Inequalities*, Grad. Texts in Math. 161, Springer, New York, 1995.
[10] K. L. Clarkson, P. Drineas, M. Magdon-Ismail, M. W. Mahoney, X. Meng, and D. P. Woodruff, *The fast Cauchy transform and faster robust linear regression*, SIAM Journal Comput., 45 (2016), pp. 763–810, https://doi.org/10.1137/140963698.
[11] A. Cohen, M. A. Davenport, and D. Leviatan, *On the stability and accuracy of least squares approximations*, Found. Comput. Math., 13 (2013), pp. 819–834.
[12] A. Cohen and G. Migliorati, *Optimal weighted least-squares methods*, SMAI J. Comput. Math., 3 (2017), pp. 181–203.
[13] M. B. Cohen, C. Musco, and C. Musco, *Input sparsity time low-rank approximation via ridge leverage score sampling*, in Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2017, pp. 1758–1777.
[14] F. Cucker and S. Smale, *On the mathematical foundations of learning*, Bull. Amer. Math. Soc., 39 (2002), pp. 1–49.
[15] C. de Boor, *An alternative approach to (the teaching of) rank, basis, and dimension*, Linear Algebra App., 146 (1991), pp. 221–229.
[16] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, *Faster least squares approximation*, Numer. Math., 117 (2011), pp. 219–249, https://doi.org/10.1007/s00211-010-0331-6.
[17] A. El Alaoui and M. W. Mahoney, *Fast Randomized Kernel Methods with Statistical Guarantees*, preprint, arXiv:1411.0306, 2014.
[18] D. C.-L. Fong and M. Saunders, *LSMR: An iterative algorithm for sparse least-squares problems*, SIAM J. Sci. Comput., 33 (2011), pp. 2950–2971.
[19] M. A. Gilles and A. Townsend, *Continuous analogues of Krylov subspace methods for differential operators*, SIAM J. Numer. Anal., 57 (2019), pp. 899–924.
[20] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. SIAM Ser. B Numer. Anal., 2 (1965), pp. 205–224.
[21] A. Gonen, F. Orabona, and S. Shalev-Shwartz, *Solving ridge regression using sketched preconditioned SVRG*, in Proc. Mach. Learn. Res. (PMLR), 48 (2016), pp. 1397–1405, http://proceedings.mlr.press/v48/gonen16.html.
[22] B. Hashemi and Y. Nakatsukasa, *On the spectral problem for trivariate functions*, BIT, 58 (2018), pp. 981–1008.
[23] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems, Curran Associates, Red Hooks, NY, 2013, pp. 315–323.
[24] N. Kamel, I. Kajo, and Y. Ruichek, *On visual periodicity estimation using singular value decomposition*, J. Math. Imaging Vision, 61 (2019), pp. 1135–1153.
[25] K. Kandasamy, M. Al-Shedivat, and E. P. Xing, *Learning HMMs with nonparametric emissions via spectral decompositions of continuous matrices*, in Advances in Neural Information Processing Systems, Vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds., Curran Associates, Red Hook, NY, 2017, https://proceedings.neurips.cc/paper/2016/file/afe434653a898da20044041262b3ac74-Paper.pdf.
[26] Y. Koyano, K. Yatabe, and Y. Oikawa, *Infinite-dimensional SVD for analyzing microphone array*, in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Piscataway, NJ, 2017, pp. 176–180.
[27] Y. T. Lee and A. Sidford, *Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems*, in 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, 2013, pp. 147–156, https://doi.org/10.1109/FOCS.2013.24.
[28] X. Meng, M. A. Saunders, and M. W. Mahoney, *LSRN: A parallel iterative solver for strongly over- or underdetermined systems*, SIAM J. Sci. Comput., 36 (2014), pp. C95–C118, https://doi.org/10.1137/120866580.
[29] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, Boston, 2003.
[30] P. Nevai, *Géza Freud, orthogonal polynomials and Christoffel functions. A case study*, J. Approx. Theory, 48 (1986), pp. 3–167.
[31] S. Olver, *Preconditioned GMRES for Oscillatory Integrals*. Technical report NA–08/19, Oxford University Computing Laboratory, University of Oxford, Oxford, 2008.
[32] S. Olver, *GMRES for the differentiation operator*, SIAM J. Numer. Anal., 47 (2009), pp. 3359–3373, https://doi.org/10.1137/080724964.

[33] S. Olver, *Shifted GMRES for oscillatory integrals*, Numer. Math., 114 (2010), pp. 607–628.
[34] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[35] E. Pauwels, F. Bach, and J.-P. Vert, *Relating leverage scores and density using regularized Christoffel functions*, in NIPS2018, Curran Associates, Red Hook, NY, 2018.
[36] M. Pilanci and M. J. Wainwright, *Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares*, J. Mach. Learn. Res., 17 (2016), pp. 1842–1879.
[37] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, in Advances in Neural Information Processing Systems, Springer, Berlin, 2008, pp. 1177–1184.
[38] V. Rokhlin and M. Tygert, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proc. Nat. Acad. Sci., 105 (2008), pp. 13212–13217, https://doi.org/10.1073/pnas.0804869105.
[39] B. Schölkopf, R. Herbrich, and A. J. Smola, *A Generalized Representer Theorem*, Springer, Berlin, 2001, pp. 416–426, https://doi.org/10.1007/3-540-44581-1_27.
[40] P. F. Shustin and H. Avron, *Gauss-Legendre features for Gaussian process regression*, preprint, arXiv:2101.01137, 2021.
[41] G. Stewart, *Afternotes Goes to Graduates School*, SIAM, Philadelphia, 1998, https://doi.org/10.1137/1.9781611971422.
[42] V. Totik, *Asymptotics for Christoffel functions for general measures on the real line*, J. Anal. Math., 81 (2000), pp. 283–303.
[43] A. Townsend and L. N. Trefethen, *Continuous analogues of matrix factorizations*, Proc, A, 471 (2015), 20140585.
[44] L. N. Trefethen, *Householder triangularization of a quasimatrix*, IMA J. Numer. Anal., 30 (2010), pp. 887–897, https://doi.org/10.1093/imanum/drp018.
[45] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, Siam, Philadelphia, 1997.
[46] J. A. Tropp, *An introduction to matrix concentration inequalities*, Found. Trends Mach, Learn., 8 (2015), pp. 1–230, https://doi.org/10.1561/2200000048.
[47] D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci., 10 (2014), pp. 1–157, https://doi.org/10.1561/0400000060.
[48] L. Xiao and T. Zhang, *A proximal stochastic gradient method with progressive variance reduction*, SIAM J. Optim., 24 (2014), pp. 2057–2075.