Abstract. Active subspaces can effectively reduce the dimension of high-dimensional parameter studies enabling otherwise infeasible experiments with expensive simulations. The key components of active subspace methods are the eigenvectors of a symmetric, positive semi-definite matrix whose elements are the average of products of partial derivatives of the simulation’s input/output map. We study a random sampling method for approximating the eigenpairs of this matrix. We offer both theoretical results based on recent non-asymptotic random matrix theory and a practical approach based on the bootstrap. We extend this analysis to the case when the gradients are approximated, for example, with finite differences. Our goal is to provide guidance for three questions that arise in active subspaces: (i) How many gradient samples does one need to accurately approximate the eigenvalues? (ii) What can be said about the accuracy of the estimated subspace? (iii) How can one be confident in the estimated eigenpairs? We test the approach on both simple quadratic functions where the active subspace is known and a parameterized PDE with 100 variables characterizing the coefficients of the differential operator.

Key words. active subspace methods

AMS subject classifications.

1. Introduction. Engineering models typically contain several input parameters that must be specified to produce a set of model outputs that contains one or more quantities of interest. The goal is to study the behavior of the quantities of interest as a function of the model’s inputs. However, parameter studies like optimization and uncertainty quantification are challenging when the number of inputs is large—especially when the model is evaluated with an expensive computer simulation. In such cases, one may analyze the sensitivity with respect to individual parameters to identify a subset of inputs whose variation contributes the most to changes in the outputs. In the best case, parameter studies can be limited to key parameters, and thus reduce the dimension of the parameter study. This approach is appropriate when the model’s variability can be attributed or explained by a few key parameters. However, a model’s output may depend on all the parameters through certain linear combinations, which generalizes the idea of seeking key parameters to seeking key directions in the parameter space.

The active subspace gives combinations of the model inputs, or a subspace, constructed so that varying the inputs within the active subspace yields greater change in the outputs than varying the inputs within the complement (i.e., the inactive subspace). The active subspace is defined by a set of eigenvectors corresponding to large eigenvalues of the average of the outer product of the gradient with itself. These eigenpairs are properties of the multivariate map between model inputs and outputs—like Fourier coefficients or the Lipschitz constant. To determine if a function admits an active subspace—and thus reduce the dimension of the parameter studies—we must estimate these eigenpairs.

In what follows we analyze a random sampling method for estimating the eigenpairs that uses independent samples of the function’s gradient. After formally defining the active subspace in Section 2, we employ results from Tropp and Gittens and Tropp to bound the probability that the estimated eigenvalues deviate from the...
true eigenvalues, which yields lower bounds on the number of samples needed for accurate estimation. We extend these results to the case where samples are approximate gradients (e.g., finite difference approximations). In Section 4 we discuss a practical bootstrap-based method to examine the variability in the estimated eigenvalues, and we demonstrate these procedures numerically in Section 5.

Notation. We use bold lower case letters to denote vectors and bold upper case letters to denote matrices. Finite sample estimates are denoted with hats, e.g., \( \hat{C} \approx C \). The functional \( \lambda_k(\cdot) \) denotes the \( k \)th eigenvalue of its argument, ordered from algebraically largest to smallest. A \( \lambda \) on its own is an eigenvalue. Norms of vectors and matrices are 2-norms; the matrix 2-norm is the operator induced norm given by the largest singular value.

2. Active subspaces. We abstractly represent the map from simulation inputs to simulation outputs by a function \( f: \mathcal{X} \rightarrow \mathbb{R} \), where \( \mathcal{X} \subseteq \mathbb{R}^m \) represents the space of simulation inputs, which we assume is centered at the origin. Let \( \mathcal{X} \) be equipped with a weight function \( \rho: \mathbb{R}^m \rightarrow \mathbb{R}_+ \) that is smooth, strictly positive over the domain \( \mathcal{X} \), and zero outside of \( \mathcal{X} \). In the context of uncertainty quantification, this weight function represents a given probability density function on the inputs; examples in this context include Gaussian, uniform, or data-derived Bayesian posterior density functions. Denote an element \( x \in \mathcal{X} \), so \( f(x) = f(x) \).

We are interested in the following matrix, denoted \( C \) and defined as

\[
C = \int (\nabla_x f)(\nabla_x f)^T \rho \, dx. \quad (2.1)
\]

Samarov [17] studied this matrix as one of several average derivative functionals in the context of regression functions. The matrix \( C \) is symmetric and positive semi-definite, so it has a real eigenvalue decomposition

\[
C = W\Lambda W^T, \quad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_1 \geq \cdots \geq \lambda_m \geq 0. \quad (2.2)
\]

Partition the eigenpairs,

\[
W = [W_1 \quad W_2], \quad \Lambda = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \end{bmatrix}, \quad (2.3)
\]

where \( W_1 \) contains the first \( n < m \) eigenvectors, and \( \Lambda_1 \) contains the \( n \) largest eigenvalues. The eigenvalues define new coordinates

\[
y = W_1^T x, \quad z = W_2^T x. \quad (2.4)
\]

We call the subspace defined by \( W_1 \) the active subspace, and the corresponding \( y \) the active variables. This characterization is justified by the following two lemmas, proved in [4].

**Lemma 2.1.** The mean-squared directional derivative of \( f \) with respect to the eigenvector \( w_i \) is equal to the corresponding eigenvalue,

\[
\int ((\nabla_x f)^T w_i)^2 \rho \, dx = w_i^T C w_i = \lambda_i. \quad (2.5)
\]
Lemma 2.2. The mean-squared gradients of $f$ with respect to the coordinates $y$ and $z$ satisfy
\[
\int (\nabla_y f)^T (\nabla_y f) \rho \, dx = \text{trace} \left( W_1^T C W_1 \right) = \lambda_1 + \cdots + \lambda_n, \\
\int (\nabla_z f)^T (\nabla_z f) \rho \, dx = \text{trace} \left( W_2^T C W_2 \right) = \lambda_{n+1} + \cdots + \lambda_m.
\]
\hspace{0.5cm} (2.6)

The eigenvalues $\Lambda$ and eigenvectors $W$ are properties of $f$. If the $m - n$ trailing eigenvalues $\Lambda_2$ are exactly zero, then $f$ is constant along the directions corresponding to $W_2$, i.e., with perturbations in $z$. If $\Lambda_2$ is not exactly zero but significantly smaller than $\Lambda_1$, then $f$ changes less on average along $z$ than along $y$. If $f$ admits such a property, we would like to discover it and exploit it. Since $f$’s variation along $y$ will be greater than along $z$ on average, we can study the parametric dependence by focusing on the variables $y$. In other words, we can reduce the dimension of the parameter studies from $m$ to $n < m$. For parameter studies whose work depends exponentially on the number of parameters—e.g., high-dimensional integration or response surface construction—such reduction can enable otherwise intractable studies.

The tremendous potential benefits of the dimension reduction motivate us to study methods for estimating the eigenpairs of $C$. The elements of $C$ are defined as integrals over the $m$-dimensional space $X$. We might be discouraged since computing the desired $W$ and $\Lambda$ nominally requires approximating high-dimensional integrals; deterministic numerical integration rules are impractical beyond a handful of variables, especially if the integrand is costly to evaluate. We therefore focus on a random sampling approach to approximate the eigenpairs of $C$, where we take advantage of recent theoretical results that bound the number of samples needed to approximate the spectrum of sums of random matrices.

2.1. Examples. Two special cases illustrate the active subspace. The first class of functions are index models that have the form $f(x) = h(a_1^T x, \ldots, a_k^T x)$, where $h : \mathbb{R}^k \to \mathbb{R}$, and $a_i$ are constant $m$-vectors. In this case, $C$ has rank at most $k$, and the active subspace is contained in span$(a_1, \ldots, a_k)$. If $k = 1$, then the active subspace can be discovered with a single evaluation of the gradient anywhere in $X$.

The second special case is a function of the form $f(x) = h(x^T A x)/2$, where $h : \mathbb{R} \to \mathbb{R}$, and $A$ is a symmetric $m \times m$ matrix. In this case
\[
C = A \left( \int (h')^2 x x^T \rho \, dx \right) A^T,
\]
where $h' = h'(x^T A x)$ is the derivative of $h$. This implies that the null space of $C$ is the null space of $A$ provided that $h'$ is non-degenerate. We will study the example where $h(t) = t$ in Section 5.

2.2. Related literature. The idea of studying the eigenpairs of the average outer product of the gradients arose in statistics as average derivative functionals [17] for exploring structure in multivariate regression functions. When the form of the regression function is not known, kernel-based constructions can be used to estimate the gradients from a collection of predictor/response pairs [20] and subsequently the eigenpairs. In our case, the function is a map between the inputs and outputs of an engineering simulation, where the assumptions on the function differ (e.g., the
function is deterministic). The problem set up in Russi’s Ph.D. thesis [15] is closer to ours. He applies the methods to physical simulations of chemical kinetics; this work is where we encountered the term active subspace. Recent work in approximation theory by Fornasier [6] attempts to discover the parameters of the active subspace solely through queries of the function, and guarantees on reconstruction follow from recent compressed sensing results under the assumption that \( f \) is an index model.

3. Computing active subspaces. If drawing independent samples from the density \( \rho \) is cheap and simple, then a straightforward and easy-to-implement random sampling method to approximate the eigenvalues \( \Lambda \) and eigenvectors \( W \) proceeds as follows.

1. Draw \( N \) samples \( x_j \) independently from the measure \( \rho \).
2. For each \( x_j \), compute \( \nabla_x f_j = \nabla_x f(x_j) \).
3. Approximate
\[
C \approx \hat{C} = \frac{1}{N} \sum_{j=1}^{N} (\nabla_x f_j)(\nabla_x f_j)^T.
\] (3.1)

4. Compute the eigendecomposition \( \hat{C} = \hat{W} \hat{\Lambda} \hat{W}^T \).

The last step is equivalent to computing the SVD of the matrix
\[
\frac{1}{\sqrt{N}} \begin{bmatrix} \nabla_x f_1 & \cdots & \nabla_x f_N \end{bmatrix} = \hat{W} \sqrt{\hat{\Lambda} \hat{V}},
\] (3.2)

where standard manipulations show that the singular values are the square roots of the eigenvalues, and the left singular vectors are the eigenvectors. The SVD perspective was developed by Russi [15] as the method to discover the active subspace. It shows that the active subspace can be interpreted as the principal components of a collection of gradients.

For many simulations, the number \( m \) of input parameters is small enough (e.g., tens to thousands) so that computing the full eigendecomposition or singular value decomposition is negligible compared to the cost of computing the gradient \( N \) times; we consider this to be our case of interest. We are therefore concerned with understanding the number of gradient samples needed so that the estimates \( \hat{\Lambda} \) and \( \hat{W} \) are close to the true \( \Lambda \) and \( W \).

We apply recent work by Tropp [19] and Gittens and Tropp [8] on the spectrum of sums of random matrices to answer these questions. Note that the gradient vector \( \nabla_x f(x) \) as a function of \( x \) is completely deterministic. Randomness appears by virtue of the independent samples of \( x \) from the density \( \rho \)—a standard interpretation of Monte Carlo techniques for integration.

**Theorem 3.1.** Assume that \( \| \nabla_x f \| \leq L \) for all \( x \in \mathcal{X} \). Then for \( 0 < \varepsilon \leq 1 \),
\[
P \left\{ \hat{\lambda}_k \geq (1 + \varepsilon)\lambda_k \right\} \leq (m - k + 1) \exp \left( -\frac{N\lambda_k \varepsilon^2}{4L^2} \right),
\] (3.3)
and
\[
P \left\{ \hat{\lambda}_k \leq (1 - \varepsilon)\lambda_k \right\} \leq k \exp \left( -\frac{N\lambda_k^2 \varepsilon^2}{4\lambda_1 L^2} \right). \quad (3.4)
\]
The key to establishing Theorem 3.1 is a matrix Bernstein inequality from Theorem 5.3 from Gittens and Tropp [8]. When we apply this concentration result, we set:

\[ X_j = \nabla_x f_j \nabla_x f_j^T, \quad (3.5) \]

so that \( X_j \) is a random draw of a gradient vector. Thus, each \( X_j \) is an independent random sample of a matrix from the same distribution. Under this notion of randomness,

\[ \mathbb{E}[X_j] = \int \nabla_x f_j \nabla_x f_j^T \rho \, dx = \int \nabla_x f \nabla_x f^T \rho \, dx = C. \quad (3.6) \]

For completeness, we restate Theorem 5.3 from [8].

**Theorem 3.2 (Eigenvalue Bernstein Inequality for Subexponential Matrices, Theorem 5.3 [8]).** Consider a finite sequence \( \{X_j\} \) of independent, random, self-adjoint matrices with dimension \( n \), all of which satisfy the subexponential moment growth condition

\[ \mathbb{E}[X_m^j] \preceq m^{1/2} B^{m-2} \Sigma_j^2 \]

for \( m = 2, 3, 4, \ldots \), where \( B \) is a positive constant and \( \Sigma_j^2 \) are positive-semidefinite matrices. Given an integer \( k \leq n \), set

\[ \mu_k = \lambda_k \left( \sum_j \mathbb{E}[X_j] \right). \]

Choose \( V_+ \) as an orthogonal matrix of size \( n \times (n-k+1) \) that satisfies

\[ \mu_k = \lambda_{\max} \left( \sum_j V_+^T (\mathbb{E}X_j) V_+ \right), \]

and define

\[ \sigma_k^2 = \lambda_{\max} \left( \sum_j V_+^T \Sigma_j^2 V_+ \right). \]

Then, for any \( t \geq 0 \),

\[ \mathbb{P} \left\{ \lambda_k \left( \sum_j X_j \right) \geq \mu_k + t \right\} \leq \begin{cases} (n-k+1) \cdot \exp\left(-\frac{t^2}{4 \sigma_k^2}\right), & t \leq \sigma_k^2 / B, \\ (n-k+1) \cdot \exp\left(-\frac{t}{4 B}\right), & t \geq \sigma_k^2 / B. \end{cases} \quad (3.7) \]

**Proof.** (Theorem 3.1.) We begin with the upper estimate (3.3). First note that

\[ \mathbb{P} \left\{ \lambda_k(C) \geq \lambda_k(C) + t \right\} = \mathbb{P} \left\{ \lambda_k \left( \sum_{j=1}^N \nabla_x f_j \nabla_x f_j^T \right) \geq N \lambda_k + N t \right\}. \quad (3.7) \]

In this form we can apply Theorem 3.2. We check that the bound on the gradient’s norm implies that the matrix \( \nabla_x f \nabla_x f^T \) satisfies the subexponential growth condition:

\[ \int (\nabla_x f \nabla_x f^T)^p \rho \, dx = \int (\nabla_x f \nabla_x f^T)^{p-1} \nabla_x f \nabla_x f^T \rho \, dx \]

\[ \leq (L^2)^{p-1} \int \nabla_x f \nabla_x f^T \rho \, dx \]

\[ \leq \frac{B^2}{2} (L^2)^{p-2} (L^2 C). \quad (3.8) \]
Next we set
\[ \mu_k = \lambda_k \left( \sum_{j=1}^{N} \int \nabla_x f_j \nabla_x f_j^T \rho \, dx \right) = N\lambda_k, \quad (3.9) \]
where we simplified using the identically distributed samples of \( x_j \). Choose \( W_+ = W(:, k:m) \) to be the last \( m-k+1 \) eigenvectors of \( C \), and note that
\[ \lambda_{\max} \left( \sum_{j=1}^{N} W_+^T \left( \int \nabla_x f \nabla_x f^T \rho \, dx \right) W_+ \right) = N\lambda_{\max}(W_+^T CW_+) = N\lambda_k = \mu_k, \quad (3.10) \]
as required by Theorem 3.2. Define
\[ \sigma_k^2 = \lambda_{\max} \left( \sum_{j=1}^{N} W_+^T (L^2 C) W_+ \right) = NL^2 \lambda_{\max}(W_+^T CW_+) = NL^2 \lambda_k. \quad (3.11) \]
With these quantities, the subgaussian branch of Theorem 3.2 states,
\[ \mathbb{P} \left\{ \lambda_k \left( \sum_{j=1}^{N} \nabla_x f_j \nabla_x f_j^T \right) \geq N\lambda_k + Nt \right\} \leq (m-k+1) \exp \left( \frac{-(Nt)^2}{4\sigma_k^2} \right), \quad Nt \leq \sigma_k^2/L^2. \quad (3.12) \]
Applying this theorem with \( t = \varepsilon \lambda_k \) and the computed quantities produces the upper estimate (3.3).

For the lower estimate,
\[ \mathbb{P} \left\{ \lambda_k(C) \leq \lambda_k(C) - t \right\} \]
\[ = \mathbb{P} \left\{ -\lambda_k(C) \geq -\lambda_k(C) + t \right\} \]
\[ = \mathbb{P} \left\{ -\lambda_k \left( \sum_{j=1}^{N} \nabla_x f_j \nabla_x f_j^T \right) \geq -N\lambda_k(C) + Nt \right\} \]
\[ = \mathbb{P} \left\{ \lambda_{m-k+1} \left( \sum_{j=1}^{N} \nabla_x f_j \nabla_x f_j^T \right) \geq N\lambda_{m-k+1}(-C) + Nt \right\} \quad (3.13) \]
\[ = \mathbb{P} \left\{ \lambda_{k'} \left( \sum_{j=1}^{N} \nabla_x f_j \nabla_x f_j^T \right) \geq N\lambda_{k'}(-C) + Nt \right\} , \]
for \( k' = m-k+1 \). We can now apply Theorem 3.2 again. The subexponential growth condition is satisfied since
\[ \int \left( -\nabla_x f \nabla_x f^T \right)^p \rho \, dx \leq \int \left( \nabla_x f \nabla_x f^T \right)^p \rho \, dx \leq \frac{p!}{2} \left( L^2 \right)^{p-2} \left( L^2 C \right). \quad (3.14) \]
Set
\[ \mu_{k'} = \lambda_{k'} \left( \sum_{j=1}^{N} \int (-\nabla_x f_j \nabla_x f_j^T) \rho \, dx \right) = N\lambda_{k'}(-C). \quad (3.15) \]
Set $W_+ = W(:, 1 : k)$ to be the first $k$ eigenvectors of $C$, and note that

$$
\lambda_{\text{max}} \left( \sum_{j=1}^{N} \begin{pmatrix} W^T_+ \left( \int \left( -\nabla_x f_j \nabla_x f_j^T \right) \rho \, dx \right) W_+ \end{pmatrix} \right) = N\lambda_{\text{max}} (-W^T_+ CW_+)
$$

$$
= N(-\lambda_k(C))
= N\lambda_{m-k+1}(-C)
= N\lambda_{k'}(-C),
$$

as required by \[3.2\]

Set

$$
\sigma^2_{k'} = \lambda_{\text{max}} \left( \sum_{j=1}^{N} W^T_+ (L^2C) W_+ \right) = NL^2\lambda_{\text{max}} (W^T_+ CW_+) = NL^2\lambda_1. \tag{3.17}
$$

The subgaussian branch of Theorem \[3.2\] states

$$
P \left\{ \left( \sum_{j=1}^{N} (-\nabla_x f_j \nabla_x f_j^T) \right) \geq N\lambda_{k'}(-C) + Nt \right\}
\leq k' \exp \left( \frac{-(Nt)^2}{4\sigma^2_{k'}} \right), \quad \text{when } Nt \leq \frac{\sigma^2_{k'}}{L^2}. \tag{3.18}
$$

Plug in the computed quantities with $t = \varepsilon\lambda_{k'}$ to achieve the lower estimate \[3.4\]. Note that the conditions on the estimate are that $\varepsilon \leq (\lambda_1/\lambda_{k'}) \leq 1$.

Next we use this result to derive a lower bound on the number of gradient samples needed for relative accuracy of $\varepsilon$.

**Corollary 3.3.** Let $\kappa_k = \lambda_1/\lambda_k$. Then for $\varepsilon \in (0, 1]$,

$$
N = \Omega \left( \frac{L^2\kappa_k^2}{\lambda_1\varepsilon^2} \log(m) \right) \tag{3.19}
$$

implies $|\hat{\lambda}_k - \lambda_k| \leq \varepsilon\lambda_k$ with high probability.

**Proof.** Starting with the upper estimate from Theorem \[3.1\] if

$$
N \geq \frac{4L^2}{\lambda_k\varepsilon^2} (\beta + 1) \log(m) \geq \frac{4L^2}{\lambda_k\varepsilon^2} (\beta \log(m) + \log(m - k + 1)), \tag{3.20}
$$

then

$$
P \left\{ \hat{\lambda}_k \geq (1 + \varepsilon)\lambda_k \right\} \leq m^{-\beta}. \tag{3.21}
$$

Similarly for the lower estimate from Theorem \[3.1\] if

$$
N \geq \frac{4L^2\lambda_1}{\lambda_k^2\varepsilon^2} (\beta + 1) \log(m) \geq \frac{4L^2\lambda_1}{\lambda_k^2\varepsilon^2} (\beta \log(m) + \log(k)), \tag{3.22}
$$

then

$$
P \left\{ \hat{\lambda}_k \leq (1 - \varepsilon)\lambda_k \right\} \leq m^{-\beta}. \tag{3.23}
$$
Setting \( \kappa_k = \lambda_1 / \lambda_k \) and taking
\[
N \geq (\beta + 1) \frac{4L^2 \kappa^2}{\lambda_1 \varepsilon^2} \log(m)
\] (3.24)
satisfies both conditions as required. \( \square \)

We can combine results from Golub and Van Loan \[9, Chapter 8\] with results from Tropp \[19\] to obtain an estimate of the distance between the true active subspace defined by the eigenvectors \( \mathbf{W}_1 \) and the estimated active subspace defined by the columns \( \hat{\mathbf{W}}_1 \). This requires a different matrix Bernstein inequality, now in the form of Theorem 6.1 from Tropp \[19\]. This theorem is restated below. When we apply it, \( \mathbf{X}_j = \nabla_x f_j \nabla_x f_j^T - \mathbf{C} \), that is, the random matrix samples are the deviance of the \( j \)th sampled gradient outer product from the true matrix \( \mathbf{C} \).

**Theorem 3.4 (Matrix Bernstein: bounded case, Theorem 6.1 \[19\]).** Consider a finite sequence \( \{\mathbf{X}_j\} \) of independent, random, self-adjoint matrices with dimension \( n \).

Assume that \( \mathbb{E}[\mathbf{X}_j] = 0 \) and \( \lambda_{\text{max}}(\mathbf{X}_j) \leq R \) almost surely.

Compute the norm of the total variance,
\[
\sigma^2 := \left\| \sum_j \mathbb{E}[\mathbf{X}_j^2] \right\|.
\]

Then the following inequality holds for all \( t \geq 0 \):
\[
\mathbb{P}\left\{ \lambda_{\text{max}}\left( \sum_j \mathbf{X}_j \right) \geq t \right\} \leq \begin{cases} n \exp(-3t^2/8\sigma^2), & t \leq \sigma^2/R, \\ n \exp(-3t/8R), & t \geq \sigma^2/R. \end{cases}
\]

**Theorem 3.5.** Assume \( \|\nabla_x f\| \leq L \) for all \( x \in \mathcal{X} \). Then for \( \varepsilon \in (0, 1] \),
\[
\mathbb{P}\left\{ \|\hat{\mathbf{C}} - \mathbf{C}\| \geq \varepsilon\|\mathbf{C}\| \right\} \leq 2m \exp\left( \frac{-3N\lambda_1 \varepsilon^2}{8L^2} \right).
\] (3.25)

**Proof.** Observe that
\[
\mathbb{P}\left\{ \|\hat{\mathbf{C}} - \mathbf{C}\| \geq t \right\} = \mathbb{P}\left\{ \lambda_{\text{max}}(\hat{\mathbf{C}} - \mathbf{C}) \geq t \right\}
\geq \mathbb{P}\left\{ \lambda_{\text{max}}(\mathbf{C} - \hat{\mathbf{C}}) \geq t \right\} + \mathbb{P}\left\{ \lambda_{\text{max}}(\mathbf{C} - \mathbf{C}) \geq t \right\}
= \mathbb{P}\left\{ \lambda_{\text{max}}\left( \sum_{j=1}^N (\nabla_x f_j \nabla_x f_j^T - \mathbf{C}) \right) \geq Nt \right\}
+ \mathbb{P}\left\{ \lambda_{\text{max}}\left( \sum_{j=1}^N (\mathbf{C} - \nabla_x f_j \nabla_x f_j^T) \right) \geq Nt \right\}
\leq 2\theta \text{ where } \theta \text{ upper-bounds both probabilities.}
\]

The final result of the proof is the upper-bound \( \theta \). Note that both
\[
\int (\nabla_x f \nabla_x f^T - \mathbf{C}) \rho \, dx = \int (\mathbf{C} - \nabla_x f \nabla_x f^T) \rho \, dx = 0.
\] (3.27)
Now, $C$ being positive semidefinite and $\|\nabla_x f\| \leq L$ imply

$$\lambda_{\text{max}}(\nabla_x f \nabla_x f^T - C) = \max_{\|v\|=1} v^T (\nabla_x f \nabla_x f^T - C) v \\ \leq \max_{\|v\|=1} v^T (\nabla_x f \nabla_x f^T) v \leq L^2,$$

(3.28)

and this also holds for $\lambda_{\text{max}}(C - \nabla_x f \nabla_x f^T)$, giving us the upper-bound $R$ for Theorem 3.4. We can bound the variance parameter $\sigma^2$ as

$$\sigma^2 = \left\| \sum_{j=1}^N \int (\nabla_x f_j \nabla_x f_j^T - C)^2 \rho \, dx \right\| \\ = N \left\| \int (\nabla_x f \nabla_x f^T - C)^2 \rho \, dx \right\| \\ = N \|L^2 C - C^2\| \leq N \|C\| \|L^2 I - C\| \leq N\lambda_1 L^2. \quad (3.29)$$

The last line follows from the fact that $\lambda_1 \leq L^2$. Again, this bound holds for $C - \nabla_x f \nabla_x f^T$. The subgaussian branch of Theorem 3.4 holds for an upper bound on $\sigma^2$, which yields an upper bound on $\theta$. Plugging in the computed quantities with $t = \varepsilon \|C\| = \varepsilon \lambda_1$ yields the desired result, since $\varepsilon \in (0,1]$ satisfies the subgaussian bound. \qed

We use this result to produce a lower bound on the number of samples needed for $\varepsilon$ relative accuracy.

**Corollary 3.6.** For $\varepsilon \in (0,1]$, $\varepsilon \leq (\lambda_k - \lambda_{k+1})/(5\lambda_1)$, and choose $N$ according to Corollary 3.6. Then with high probability

$$N = \Omega\left( \frac{L^2}{\lambda_1 \varepsilon^2 \log(m)} \right) \quad (3.30)$$

implies that $\|\hat{C} - C\| \leq \varepsilon \|C\|$ with high probability.

We can combine Corollary 3.6 with [9, Corollary 8.1.11] to control the error in the estimated subspace defined by $\hat{W}_1$. We quantify this error by the distance between the subspace defined by $W_1$ and the subspace defined by $\hat{W}_1$. Recall the definition of the distance between subspaces [18],

$$\text{dist}(W_1, \hat{W}_1) = \|W_1 W_1^T - \hat{W}_1 \hat{W}_1^T\| = \|W_1^T W_2\|. \quad (3.31)$$

**Corollary 3.7.** Let $\varepsilon > 0$ be such that

$$\varepsilon \leq (\lambda_k - \lambda_{k+1})/(5\lambda_1), \quad (3.32)$$

and choose $N$ according to Corollary 3.6. Then with high probability

$$\text{dist}(W_1, \hat{W}_1) \leq 4\lambda_1 \varepsilon / (\lambda_k - \lambda_{k+1}). \quad (3.33)$$

**Proof.** Let $E = \hat{C} - C$. For $\varepsilon$ in (3.32) with $N$ chosen according to Corollary 3.6 we have with high probability

$$\|E\| \leq \varepsilon \|C\| = \varepsilon \lambda_1 \leq (\lambda_k - \lambda_{k+1})/5. \quad (3.34)$$
Under this condition, Corollary 8.1.11 states
\[
\text{dist} \left( W_1, \hat{W}_1 \right) \leq 4\| E \| / (\lambda_k - \lambda_{k+1}) \leq 4\lambda_1 \varepsilon / (\lambda_k - \lambda_{k+1}), \tag{3.35}
\]
as required.

Corollary 3.7 shows that control of the eigenvalues implies control of the subspace generated by the eigenvectors. However, the error in the estimated subspace is inversely proportional to the corresponding gap in the eigenvalues. This implies, for example, if the gap between the second and third eigenvalues is larger than the gap between the first and second, then estimates of a two-dimensional active subspace are more accurate than estimates of a one-dimensional active subspace.

3.1. Approximate gradients. Large-scale simulations often lack simple-to-evaluate expressions for the gradients of the outputs with respect to the inputs. Nevertheless, there are a variety of methods to evaluate gradients including adjoint methods \cite{3,2} and automatic differentiation \cite{10}. When there is no existing code for a derivative, finite difference approximations are possible when \( m \) is not too large and \( f \) is neither too expensive nor too noisy. Recent work even characterizes the gradient when function evaluations contain noise \cite{13,14}.

Next we extend the bounds on errors in the estimated eigenpairs to the case when the gradients are computed with some error. The model for the error that we analyze is generic. Let \( g(x) \) denote the approximate gradient computed at \( x \in \mathcal{X} \). We assume that
\[
\| g(x) - \nabla_x f(x) \| \leq \sqrt{m\gamma_h}, \quad x \in \mathcal{X}, \tag{3.36}
\]
where
\[
\lim_{h \to 0} \gamma_h = 0. \tag{3.37}
\]
The parameter \( h \) may be, for example, a finite difference parameter or grid spacing in an adjoint computation.

Define the symmetric positive semidefinite matrix \( G \) and its eigenvalue decomposition as
\[
G = \int g g^T \rho \, dx = U \Theta U^T, \quad \Theta = \text{diag} (\theta_1, \ldots, \theta_m), \tag{3.38}
\]
and its random sample approximation
\[
\hat{G} = \frac{1}{N} \sum_{j=1}^N g_j g_j^T = \hat{U} \hat{\Theta} \hat{U}^T, \quad \hat{\Theta} = \text{diag} (\hat{\theta}_1, \ldots, \hat{\theta}_m), \tag{3.39}
\]
where \( g_j = g(x_j) \) for \( x_j \) drawn independently from \( \rho \). With these quantities defined, we have the following lemma.

**Lemma 3.8.** The norm of the difference between \( \hat{C} \) and \( \hat{G} \) is bounded by
\[
\| \hat{C} - \hat{G} \| \leq \sqrt{m\gamma_h} + 2L. \tag{3.40}
\]

**Proof.** Let \( g = g(x) \) and \( \nabla_x f = \nabla_x f(x) \). First observe
\[
\| g + \nabla_x f \| = \| g - \nabla_x f + 2\nabla_x f \| \leq \| g - \nabla_x f \| + 2\| \nabla_x f \| \leq \sqrt{m\gamma_h} + 2L. \tag{3.41}
\]
Next,
\[
\|\mathbf{g} g^T - \nabla_x f \nabla_x f^T\| = \frac{1}{2} \|\mathbf{g} + \nabla_x f\| (\mathbf{g} - \nabla_x f)^T + (\mathbf{g} - \nabla_x f\| (\mathbf{g} + \nabla_x f)^T
\]
\[
\leq \|\mathbf{g} + \nabla_x f\| (\mathbf{g} - \nabla_x f)^T
\]
\[
\leq (\sqrt{m \gamma_h} + 2L) \sqrt{m \gamma_h}.
\] (3.42)

Then,
\[
\|\hat{G} - \tilde{C}\| = \left\| \frac{1}{N} \sum_{j=1}^{N} \mathbf{g}_j \mathbf{g}_j^T - \frac{1}{N} \sum_{j=1}^{N} \nabla_x f_j \nabla_x f_j^T \right\|
\]
\[
\leq \frac{1}{N} \sum_{j=1}^{N} \|\mathbf{g}_j \mathbf{g}_j^T - \nabla_x f_j \nabla_x f_j^T\|
\]
\[
\leq \sqrt{m \gamma_h} (\sqrt{m \gamma_h} + 2L).
\] (3.43)

We combine Lemma 3.8 with Corollary 3.3 to study the error in the eigenvalues of the random sample estimate with approximate gradients.

**Theorem 3.9.** For \(\varepsilon \in (0,1]\), if \(N\) is chosen as (3.19), then the difference between the true eigenvalue \(\lambda_k\) and the eigenvalue \(\hat{\theta}_k\) of the random sample estimate with approximate gradients is bounded as
\[
|\lambda_k - \hat{\theta}_k| \leq \varepsilon \lambda_k + \sqrt{m \gamma_h} (\sqrt{m \gamma_h} + 2L).
\] (3.44)

**Proof.** Observe that
\[
|\lambda_k - \hat{\theta}_k| \leq |\lambda_k - \hat{\lambda}_k| + |\hat{\lambda}_k - \hat{\theta}_k|.
\] (3.45)

Apply Corollary 3.3 to the first term. The second term follows from [9, Corollary 8.1.6] combined with Lemma 3.8 since
\[
|\hat{\theta}_k - \hat{\lambda}_k| = |\lambda_k(\hat{G}) - \lambda_k(\hat{C})| \leq \|\hat{G} - \hat{C}\| \leq \sqrt{m \gamma_h} (\sqrt{m \gamma_h} + 2L).
\] (3.46)

The error in the finite sample eigenvalue estimates goes to zero at the same rate as the error in the approximate gradient. Next we attend to the error in the active subspace computed with random samples and approximate gradients.

**Theorem 3.10.** Choose \(\varepsilon\) and \(N\) according to Corollary 3.4 and also \(N\) so that Corollary 3.3 applies to the \(k + 1\) th eigenvalue. Choose \(h\) small enough so that
\[
\sqrt{m \gamma_h} (\sqrt{m \gamma_h} + 2L) \leq \frac{1}{5} (\hat{\lambda}_k - \hat{\lambda}_{k+1}).
\] (3.47)

Then
\[
\text{dist}(U_1, W_1) \leq \frac{4}{\lambda_k - \lambda_{k+1}} \left( \frac{\sqrt{m \gamma_h} (\sqrt{m \gamma_h} + 2L)}{1 - 2\varepsilon} + \varepsilon \lambda_1 \right),
\] (3.48)

with high probability.
Proof. First
\[
\text{dist}\left(\hat{U}_1, W_1\right) \leq \text{dist}\left(\hat{U}_1, \hat{W}_1\right) + \text{dist}\left(\hat{W}_1, W_1\right). \tag{3.49}
\]
The second term is bounded in Corollary 3.7 under the assumption on \(N\) and \(\varepsilon\). By the condition (3.47) on \(h\), [9, Corollary 8.1.11] implies the first term is bounded by
\[
\text{dist}\left(\hat{U}_1, \hat{W}_1\right) \leq \frac{4}{\lambda_k - \lambda_{k+1}} \|\hat{G} - \hat{C}\|. \tag{3.50}
\]
The condition on \(N\) implies \(|\hat{\lambda}_k - \lambda_k| \leq \varepsilon\lambda_k\) and \(|\hat{\lambda}_{k+1} - \lambda_{k+1}| \leq \varepsilon\lambda_{k+1}\) with high probability due to Corollary 3.3. Then observe
\[
\lambda_k - \lambda_{k+1} = |\lambda_k - \lambda_{k+1}|
\leq |\lambda_k - \hat{\lambda}_k| + |\hat{\lambda}_{k+1} - \lambda_{k+1}| + (\hat{\lambda}_k - \hat{\lambda}_{k+1}) \tag{3.51}
\leq \varepsilon\lambda_k + \varepsilon\lambda_{k+1} + (\hat{\lambda}_k - \hat{\lambda}_{k+1}).
\]
Rearranging this inequality yields
\[
\hat{\lambda}_k - \hat{\lambda}_{k+1} \geq \lambda_k - \lambda_{k+1} - \varepsilon\lambda_k - \varepsilon\lambda_{k+1}
\geq (1 - 2\varepsilon)(\lambda_k - \lambda_{k+1}). \tag{3.52}
\]
Combining this with the bound from Lemma 3.8 yields the result. \(\square\)

In summary, the eigenvalues and the active subspace approximate \(d\) with random sampling and approximate gradients are well behaved. The error bounds include a term that goes to zero like the error in the approximate gradient and a term that behaves like the random sample approximation with exact gradients.

4. Practical approach to computation. The bounds we present in Section 3 provide a solid theoretical foundation for understanding the behavior of the random sampling estimates. However, many of the quantities in the bounds may not be known a priori—such as the maximum norm of the gradient \(L\) and the true eigenvalues of the matrix \(C\). In this section we offer a practical recipe guided by the insights from the theory.

4.1. First estimate the eigenvalues. The first objective is to estimate the eigenvalues including a measure of the variability from the finite samples. If one wishes to approximate the first \(k\) eigenvalues from a matrix \(C\) with dimension \(m \times m\), we recommend choosing the number \(N\) of independent gradient samples as
\[
N = \alpha k \log(m), \tag{4.1}
\]
where \(\alpha\) is a multiplier between 2 and 10. The \(\log(m)\) term follows from the bounds in Theorem 3.1. We have plotted this heuristic for \(N\) as a function of \(m\) with \(k = 6\) in Figure 4.1.

We form \(\hat{C}\) using the samples of the gradient as in \(3.1\), and then compute its eigenvalue decomposition. We expect that computing the full eigendecomposition will be much cheaper than computing the gradient samples. A function of a thousand variables produces \(\hat{C}\) with dimension thousand by thousand. Full eigendecompositions for matrices this size are computed in seconds on modern laptops.

We further suggest computing bootstrap confidence intervals for the eigenvalues, which involves computing the eigendecompositions of several matrices the size of \(\hat{C}\).
The bootstrap creates replicates by (i) sampling with replacement from the set of gradient samples, (ii) computing the replicate $\hat{C}^*$, and (iii) computing its eigenvalue decomposition. The collection of eigenvalue replicates is used to estimate confidence intervals. Further details on the bootstrap can be found in Efron’s well known text [5].

As seen in Corollary 3.7, the error in the estimated subspace depends inversely on the gap between the eigenvalues scaled by the largest eigenvalue. The key to accurately approximating the subspace is to look for gaps in the eigenvalues; this is consistent with standard perturbation theory for eigenvector computations [18]. For example, if there is a larger gap between the third and fourth eigenvalue than between the second and third, then estimates of the three-dimensional subspace will be more accurate than estimates of the two-dimensional subspace. This contrasts heuristics for deciding the dimension of the subspace in (i) model reduction based on the proper orthogonal decomposition [11] and (ii) dimension reduction based on principal component analysis [12]. In these cases, one chooses the dimension of the subspace by a threshold on the magnitude of the eigenvalues—e.g., so that the sum of retained eigenvalues exceeds some proportion of the sum of all eigenvalues. To accurately approximate the active subspace, the most important quantity is the spectral gap, which indicates a separation between scales.

To tease out the spectral gap, plot the estimated eigenvalues and their respective upper and lower bootstrap confidence bounds; a gap between subsequent intervals offers confidence of a spectral gap, and hence, the presence of an active subspace. In Section 5 we show several examples of such plots (Figures 5.1, 5.2, 5.4, 5.5).

**4.2. Next approximate the subspace.** To study the variability in the active subspace due to finite sampling, we again turn to the bootstrap. In particular for each replicate $\hat{W}^*$ of the eigenvectors, we compute $\text{dist}(\hat{W}_1, \hat{W}^*_1)$. One can examine the confidence intervals of this quantity to study the stability of the subspace. Recall
that the distance between subspaces is bounded above by 1, so a confidence interval whose values are close to 1 indicates a poorly approximated active subspace. Figures 5.1, 5.3, 5.4, and 5.5 show examples of plotting this metric for the stability of the subspace; the first two Figures also compare the measure of stability to the true error in the active subspace.

4.3. A step-by-step procedure. We summarize the practical approach to approximating the active subspace with confidence intervals provided by the bootstrap. What follows is a modification of the procedure outlined at the beginning of Section 3 including our suggestions for parameter values. This procedure assumes the user as decided on the number $k$ of eigenvalues to examine.

1. Choose $N = \alpha k \log(m)$, where $\alpha$ is a multiplier between 2 and 10.
2. Draw $N$ samples $x_j$ independently from $\rho$.
3. For each $x_j$, compute $\nabla_x f_j = \nabla_x f(x_j)$.
4. Compute
   \[ \hat{C} = \frac{1}{N} \sum_{j=1}^{N} (\nabla_x f_j)(\nabla_x f_j)^T = \hat{W}\hat{\Lambda}\hat{W}^T. \]  
   (4.2)
5. Use the bootstrap sampling with replacement strategy to compute replicates $\hat{C}^* = \hat{W}^*\hat{\Lambda}^*(\hat{W}^*)^T$, and use these replicates to compute confidence intervals on the eigenvalue estimates.
6. Plot the eigenvalue confidence intervals and look for large gaps.
7. Choose the dimension $n$ of the active subspace corresponding to the largest eigenvalue gap. If there is no perceivable gap, then an active subspace may not be present in the first $k - 1$ dimensions.
8. Use the bootstrap replicates of the eigenvectors to compute $\text{dist}(\hat{W}_1, \hat{W}^*_1)$, and compute confidence intervals of this quantity. If the subspace associated with the gap is not stable, then the active subspace may not be reliable.

A few comments are in order. First, we assume the dimension $m$ of $C$ is small enough so that the eigendecompositions of $C$ and its bootstrap replicates are much cheaper than the samples of the gradient. Such is the case when $m$ is in the thousands (i.e., $f$ depends on thousands of input variables), and $f$ and $\nabla_x f$ are the result of a high-fidelity engineering simulation. Second, we choose the bootstrap to examine the variability because we assume that sampling more gradients is not feasible. If this is not the case, i.e., if one can cheaply evaluate many more gradient samples, then one can compute simple Monte Carlo estimates and central limit theorem confidence intervals of the eigenvalues in place of the bootstrap estimates. Lastly, we note that the elements of $C$ are multivariate integrals. If $m$ is small enough (2 or 3) and evaluating $f$ is cheap enough, then more accurate numerical quadrature rules will perform better than the random sampling. However, practical estimates of the error are much more difficult to compute, since the error is due to bias instead of variance.

5. Experiments. We apply the procedures described in Section 4 to two models: (i) a quadratic function and (ii) a linear functional of the solution of a parameterized PDE. The quadratic model is simple enough to analytically derive the eigenpairs of the active subspace for thorough evaluation of the method. We study the same PDE model in [4, Section 5]. Gradients are available through adjoints, but the true active
subspaces are not available. Codes for the experiments in this section can be found at [https://bitbucket.org/paulcon/computing-active-subspaces](https://bitbucket.org/paulcon/computing-active-subspaces).

The PDE example uses the Random Field Simulation code [http://www.mathworks.com/matlabcentral/fileexchange/2761](http://www.mathworks.com/matlabcentral/fileexchange/2761) as well as the MATLAB PDE Toolbox.

### 5.1. A quadratic model.

Consider a quadratic function of $m = 10$ variables,

$$ f(x) = \frac{1}{2} x^T A x, \quad x \in [-1,1]^{10}, \quad (5.1) $$

where $A$ is symmetric and positive definite. We take $\rho = 2^{-10}$ on the hypercube $[-1,1]^{10}$ and zero elsewhere. The gradient is $\nabla_x f(x) = A x$, so

$$ C = A \left( \int xx^T \rho \, dx \right) A^T = \frac{1}{3} A^2. \quad (5.2) $$

The eigenvalues of $C$ are the squared eigenvalues of $A$, and the eigenvectors of $C$ are the eigenvectors of $A$.

We will study three different $A$'s constructed from three choices for the eigenvalues: (i) exponential decay with a constant rate, (ii) like the first but with a larger gap between the first and second eigenvalue, and (iii) like the first with a larger gap between third and fourth eigenvalue. The three cases of eigenvalues for $A$ are shown in the top row of Figure 5.1. Each $A$ has the same eigenvectors, which we generate as an orthogonal basis from a random $10 \times 10$ matrix. In the following figures, we denote the three $A$’s as $A_1$, $A_2$, and $A_3$ with corresponding $C_1$, $C_2$, and $C_3$. The eigenvectors defining the active subspace for each case are $W_{1,1}$, $W_{1,2}$, and $W_{1,3}$.

To estimate the eigenvalues, we choose $N$ as in (4.1) with the multiplier $\alpha = 2$ and $k = 6$ eigenvalues of interest, which yields $N = 104$ evaluations of the gradient. The middle row of Figure 5.1 shows the bootstrap 99% confidence intervals for the first six eigenvalues along with the true eigenvalues of $C$. The small confidence intervals suggest confidence in the estimates. The gaps are apparent in the last two cases. The bottom row of Figure 5.1 shows bootstrap confidence bounds on the distance between the true $k$-dimensional active subspace and the subspace estimated with the $N$ samples; the true distance is shown with the blue circles. Notice that subspaces corresponding to the larger eigenvalue gap are much better approximated than the others. For example, the three-dimensional subspace is better approximated than even the one- and two-dimensional subspaces for the third case.

Next we repeat the study using finite difference approximations of the gradient with $h = 10^{-1}$, $10^{-3}$, and $10^{-5}$. The first of these step sizes is larger than would normally be used for such a model. We chose this large value to study the interplay between inaccurate gradients and the finite sample approximations of the eigenpairs. Figure 5.2 shows the true eigenvalues, their estimates, and the 99% confidence intervals for all three cases and all three values of $h$; the black horizontal lines show the value of $h$. Eigenvalues that are smaller than $h$ are estimated less accurately than those larger than $h$. Also the gaps are much less noticeable in the estimates when finite difference parameter is not small enough to resolve the smaller eigenvalue in the pair defining the gap. In particular, if the smaller eigenvalue in a gap is below the finite difference approximation level, this causes a phantom eigenvalue to appear; see Figure 5.2(b,c) for example of this phenomenon. Our theory implicitly describes how gaps need to be resolved from all the approximations involved to be trusted.
Fig. 5.1: The top row shows the eigenvalues of the three choices for $A$, which we denote $A_1$, $A_2$, and $A_3$. The second row shows the true and estimated eigenvalues along with the 99% bootstrap confidence intervals; eigenvalues are well approximated for all three cases. The third row shows the distance between the estimated subspace and the true subspace. In practice we do not have the true subspace, but we can estimate the distance with a bootstrap procedure as described in Section 4. There is a strong bias in the sample estimate with approximate gradients (blue circles). We use the bootstrap to estimate the error in the subspace as in Section 4. There is a strong bias in the estimates of the subspace error when phantom eigenvalues appear. For instance, in Figure 5.3 the estimates of the error for subspaces of dimension 4 through 6 are biased for $h = 10^{-5}$ and significantly biased for $h = 10^{-1}$. Compare this to the error
in the last three eigenvalues for the smallest $h = 10^{-5}$ in Figure 5.2.

![Eigenvalue plots](image)

Fig. 5.2: Eigenvalues, estimates, and 99% bootstrap confidence intervals using finite difference gradients with $h = 10^{-1}$ (top row), $h = 10^{-3}$ (middle row), and $h = 10^{-5}$ (bottom row). The horizontal black lines indicate the value of $h$ in each plot. In general, estimates of eigenvalues smaller than $h$ are less accurate than those larger than $h$. (Colors are visible in the electronic version.)

5.2. A parameterized PDE model. In previous work [4], we exploited the active subspace in the following parameterized PDE model to efficiently construct a kriging surface. Here we perform a more careful study of the variation in the active subspace estimated with finite samples of the gradient. Consider the following linear elliptic PDE with parameterized, variable coefficients. Let $u = u(s, x)$ satisfy

$$-\nabla_s \cdot (a \nabla_s u) = 1, \quad s \in [0, 1]^2.$$  

(5.3)
We set homogeneous Dirichlet boundary conditions on the left, top, and bottom of the spatial domain; denote this boundary by $\Gamma_1$. The right side of the spatial domain denoted $\Gamma_2$ has a homogeneous Neumann boundary condition. The log of the coefficients $a = a(s, x)$ of the differential operator is given by a truncated Karhunen-
Loeve (KL) type expansion

$$\log(a(s,x)) = \sum_{i=1}^{m} x_i \gamma_i \phi_i(s), \quad (5.4)$$

where the $x_i$ are independent, identically distributed standard normal random variables, and the $\{\phi_i(s), \gamma_i\}$ are the eigenpairs of the correlation operator

$$C(s,t) = \exp(-\beta^{-1}||s-t||_1). \quad (5.5)$$

We will study the quality of the active subspace approximation for two correlation lengths, $\beta = 1$ and $\beta = 0.01$. These correspond to long and short correlation lengths, respectively, for the random field defining the log of the coefficients. We choose a truncation of the field $m = 100$, which implies that the parameter space $\mathcal{X} = \mathbb{R}^{100}$ with $\rho$ a standard Gaussian density function. Define the linear function of the solution

$$f(x) = \frac{1}{|T_2|} \int_{T_2} u(s,x) \, ds. \quad (5.6)$$

This is the function we will study with the active subspace method. Given a value for the input parameters $x$, we discretize the PDE with a standard linear finite element method using MATLAB’s PDE Toolbox. The discretized domain has 34320 triangles and 17361 nodes; the eigenfunctions $\phi_i$ from (5.4) are approximated on this mesh. We compute the gradient of the quantity of interest (5.6) using a discrete adjoint formulation. Further details appear in our previous work [4].

The top row of Figure 5.4 shows the estimates of the eigenvalues of $C$ along with the 99% bootstrap confidence intervals for $\beta = 1$ in (5.5). The gap between the first and second eigenvalues is apparent and supported by the gap in the corresponding confidence intervals. We exploited this gap in [4] to construct an accurate univariate kriging surface of the active variable. The bottom row of Figure 5.4 shows the variance in the estimated subspace as computed with the bootstrap including the 99% confidence intervals. The left column of Figure 5.4 uses the multiplier $\alpha = 2$ when choosing the number $N$ of gradient samples; the right column uses $\alpha = 10$. Notice the overall decrease in both the range of the confidence interval and the subspace error as we include more samples. Figure 5.5 shows the identical study with the short correlation length $\beta = 0.01$ from (5.5).

6. Summary and conclusions. The average outer product of the gradient with itself is the central matrix in the development of active subspaces for dimension reduction. Its dominant eigenvectors define the directions along which a given multivariate function varies the most on average. We have analyzed a random sampling method for approximating this matrix and its eigenpairs. We use recent theory developed for the eigenvalues of sums of random matrices to analyze the probability that the finite sample eigenvalue estimates deviate from the true eigenvalues, and we combine this analysis with results from matrix computations to derive results for the subspace. We extend this analysis to quantities computed with samples of approximate gradients, e.g., finite differences. We also provide a practical computational approach that employs the bootstrap to reveal the error in the eigenvalues and the stability of the subspace.

Our analysis offers answers to the following important questions. First, how many gradient samples does one need for an accurate approximation of the first $k$
Fig. 5.4: The top row shows estimates of the eigenvalues of $C$ along with the 99% bootstrap confidence intervals for the quantity of interest (5.6) from the parameterized PDE model with the long correlation length $\beta = 1$ from (5.5). The bottom row shows the estimates and 99% bootstrap confidence intervals on the distance between the estimated active subspace and the true active. The left column is computed with the multiplier $\alpha = 2$ when choosing $N$; the right column uses $\alpha = 10$. The gap between the first and second eigenvalue is significant as judged by the gap between the bootstrap confidence intervals. (Colors are visible in the electronic version.)

eigenvalues? Precise theoretical bounds motivate a heuristic that chooses a number proportional to $k$ times the log of the dimension $m$. Second, what can be said about the accuracy of the estimated subspace? The accuracy of the estimated subspace is directly related to gaps in the eigenvalues. Third, how does one judge the stability of the computed quantities? We propose to use bootstrap confidence intervals for the eigenvalues and the stability of the subspace. Finally, how does this analysis change when gradients are not exact but approximate? Our theory shows that approximate gradients introduce a bias term in the error bounds that goes to zero as the approxi-
mate gradients become more accurate. The numerical examples suggest that this bias can produce inaccurate subspaces when the gradients are not well resolved.

REFERENCES

[1] G Berkooz, P Holmes, and J L Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual Review of Fluid Mechanics, 25 (1993), pp. 539–575.

[2] Alfio Borzi and Volker Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2012.
[3] Arther E. Bryson and Yu-Chi Ho, Applied Optimal Control: Optimization, Estimation, and Control, Hemisphere Publishing Corporation, 1975.
[4] P. Constantine, E. Dow, and Q. Wang, Active subspace methods in theory and practice: Applications to kriging surfaces, SIAM Journal on Scientific Computing, 36 (2014), pp. A1500–A1524.
[5] Bradley Efron and Robert J Tibshirani, An Introduction to the Bootstrap, vol. 57, CRC press, 1994.
[6] Massimo Fornasier, Karin Schnass, and Jan Vybiral, Learning functions of few arbitrary linear parameters in high dimensions, Foundations of Computational Mathematics, 12 (2012), pp. 229–262.
[7] Kenji Fukumizu and Chenlei Leng, Gradient-based kernel dimension reduction for regression, Journal of the American Statistical Association, 109 (2014), pp. 359–370.
[8] Alex Gittens and Joel A Tropp, Tail bounds for all eigenvalues of a sum of random matrices, arXiv preprint arXiv:1104.4513, (2011).
[9] Gene H Golub and Charles F Van Loan, Matrix Computations, Johns Hopkins University Press, 3 ed., 1996.
[10] Andreas Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Society for Industrial and Applied Mathematics, 2000.
[11] Marian Hristache, Anatoli Juditsky, Jorg Polzehl, and Vladimir Spokoiny, Structure adaptive approach for dimension reduction, The Annals of Statistics, 29 (2001), pp. 1537–1566.
[12] Ian Jolliffe, Principal Component Analysis, Wiley Online Library, 2005.
[13] Jorge J. Moré and Stefan M. Wild, Estimating derivatives of noisy simulations, ACM Trans. Math. Softw., 38 (2012), pp. 19:1–19:21.
[14] Jorge J. Moré and Stefan M. Wild, Do you trust derivatives or differences?, Journal of Computational Physics, 273 (2014), pp. 268–277.
[15] Trent M. Russi, Uncertainty Quantification with Experimental Data and Complex System Models, PhD thesis, UC Berkeley, 2010.
[16] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, Global Sensitivity Analysis: The Primer, John Wiley & Sons, 2008.
[17] A.M. Samarov, Exploring regression structure using nonparametric functional estimation, Journal of the American Statistical Association, 88 (1993), pp. 836–847.
[18] G. Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems, SIAM Review, 15 (1973), pp. 727–764.
[19] Joel A Tropp, User-friendly tail bounds for sums of random matrices, Foundations of Computational Mathematics, 12 (2012), pp. 389–434.
[20] Yingcun Xia, A constructive approach to the estimation of dimension reduction directions, The Annals of Statistics, 35 (2007), pp. 2654–2690.
Eigenvalues

Index

True
Est CI
