SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 3, pp. 61–90

Assessing Stochastic Algorithms for Large Scale Nonlinear Least Squares Problems Using Extremal Probabilities of Linear Combinations of Gamma Random Variables*

Farbod Roosta-Khorasani†, Gábor J. Székely‡, and Uri M. Ascher†

Abstract. This paper considers stochastic algorithms for efficiently solving a class of large scale nonlinear least squares (NLS) problems which frequently arise in applications. We propose eight variants of a practical randomized algorithm where the uncertainties in the major stochastic steps are quantified. Such stochastic steps involve approximating the NLS objective function using Monte Carlo methods, and this is equivalent to the estimation of the trace of corresponding symmetric positive semidefinite matrices. For the latter, we prove *tight necessary* and *sufficient* conditions on the sample size (which translates to cost) to satisfy the prescribed probabilistic accuracy. We show that these conditions are practically computable and yield small sample sizes. They are then incorporated in our stochastic algorithm to quantify the uncertainty in each randomized step. The bounds we use are applications of more general results regarding extremal tail probabilities of linear combinations of gamma distributed random variables. We derive and prove new results concerning the maximal and minimal tail probabilities of such linear combinations, which can be considered independently of the rest of this paper.

Key words. randomized algorithms, inverse problems, Monte Carlo methods, trace estimation, gamma random variable, extremal probability, large scale simulation

AMS subject classifications. 65C20, 65C05, 68W20, 60E05

DOI. 10.1137/14096311X

1. Introduction.

Large scale data fitting problems arise often in many applications in science and engineering. As the ability to gather larger amounts of data increases, the need to devise algorithms to efficiently solve such problems becomes more important. The main objective here is typically to recover some model parameters, and it is a widely accepted working assumption that having more data can only help (at worst not hurt) the model recovery.

Consider the system[1]

$$\mathbf{d}_i = \mathbf{f}_i(\mathbf{m}) + \boldsymbol{\eta}_i,\ i = 1, 2, \ldots, s, \tag{1.1}$$

where $\mathbf{d}_i \in \mathbb{R}^l$ is the measurement data obtained in the ith experiment, $\mathbf{f}_i = \mathbf{f}_i(\mathbf{m})$ is the known forward operator (or data predictor) for the ith experiment, $\mathbf{m} \in \mathbb{R}^{l_m}$ is the sought-

*Received by the editors April 1, 2014; accepted for publication (in revised form) November 24, 2014; published electronically January 15, 2015.
http://www.siam.org/journals/juq/3/96311.html
†Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (farbod@cs.ubc.ca, ascher@cs.ubc.ca). The work of these authors was partially funded by NSERC grant 84306.
‡National Science Foundation, Arlington, VA 22230, and Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, 1053 Budapest, Hungary (gszekely@nsf.gov).
[1]In this paper, we use bold lower case to denote vectors and regular capital letters to denote matrices.

after parameter vector,[2] and $\boldsymbol{\eta}_i$ is the noise incurred in the ith experiment. The total number of experiments, or data sets, is assumed large: $s \gg 1$. The goal is to find (or infer) the unknown model, $\mathbf{m}$, from the measurements $\mathbf{d}_i$, $i = 1, 2, \ldots, s$. Generally, this problem can be ill-posed. Various approaches, including different regularization techniques, have been proposed to alleviate this ill-posedness; see, e.g., [33, 13].

In this paper we assume that the forward operators have the form

$$\mathbf{f}_i(\mathbf{m}) = \mathbf{f}(\mathbf{m}, \mathbf{q}_i), \quad i = 1, \ldots, s, \tag{1.2}$$

where $\mathbf{q}_i$ are inputs such that the ith data set, $\mathbf{d}_i$, is measured after injecting the ith input (or source) $\mathbf{q}_i$ into the system. Thus, for an input $\mathbf{q}_i$, $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$ predicts the ith measurement, given the underlying model $\mathbf{m}$. We only consider a special case where $\mathbf{q}_i \in \mathbb{R}^{l_q}$ for all i and $\mathbf{f}$ is linear in $\mathbf{q}$, i.e., $\mathbf{f}(\mathbf{m}, w_1\mathbf{q}_1 + w_2\mathbf{q}_2) = w_1\mathbf{f}(\mathbf{m}, \mathbf{q}_1) + w_2\mathbf{f}(\mathbf{m}, \mathbf{q}_2)$. Alternatively, we write $\mathbf{f}(\mathbf{m}, \mathbf{q}) = G(\mathbf{m})\mathbf{q}$, where $G \in \mathbb{R}^{l \times l_q}$ is a matrix that depends nonlinearly on the sought-after $\mathbf{m}$. We also assume that the task of evaluating $\mathbf{f}$ for each input, $\mathbf{q}_i$, is computationally expensive. Examples of such a situation arise frequently in partial differential equation (PDE) constrained inverse problems with many data sets; see, e.g., [18, 10, 30] and references therein.

Under the further assumption that the independent noise satisfies[3] $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma\mathbb{I})$ for all i, where $\mathcal{N}$ denotes normal distribution, $\mathbb{I} \in \mathbb{R}^{l \times l}$ denotes the identity matrix, and $\sigma > 0$, the standard maximum likelihood (ML) approach leads to minimizing the ℓ_2 misfit function

$$\phi(\mathbf{m}) := \sum_{i=1}^{s} \|\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i\|_2^2. \tag{1.3}$$

However, since the above inverse problem is typically ill-posed, a regularization functional, $R(\mathbf{m})$, is often added to the above objective, thus minimizing instead

$$\phi_{R,\alpha}(\mathbf{m}) := \phi(\mathbf{m}) + \alpha R(\mathbf{m}), \tag{1.4}$$

where α is a regularization parameter [13]. In general, this regularization term can be chosen using a priori knowledge of the desired model. The objective functional (1.4) coincides with the maximum a posteriori (MAP) formulation. Implicit regularization also exists in which there is no explicit term $R(\mathbf{m})$ in the objective [20, 10]. Various optimization techniques can be used to decrease the value of the above objective functionals, (1.3) or (1.4), to a desired level (determined, e.g., by a given tolerance which depends on the noise level), thus recovering the sought-after model.

Algorithms that rely on efficiently approximating the misfit function $\phi(\mathbf{m})$ have been proposed and studied in [18, 10, 30, 29, 28]. In effect, they draw upon estimating the trace of an implicit[4] symmetric positive semidefinite (SPSD) matrix. To see this, rewrite (1.3) as

$$\phi(\mathbf{m}) = \|F(\mathbf{m}) - D\|_F^2, \tag{1.5}$$

[2]The parameter vector $\mathbf{m}$ often arises from a parameter function in several space variables projected onto a discrete grid and reshaped into a vector.

[3]For notational simplicity, we do not distinguish between a random vector (e.g., noise) and its realization, as they are clear within the context in which they are used.

[4]By "implicit matrix" we mean that the matrix of interest is not available explicitly: only information in the form of matrix-vector products for any appropriate vector is available.

where $F(\mathbf{m})$ and D are $l \times s$ matrices whose ith columns are, respectively, $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$ and $\mathbf{d}_i$, and $\|\cdot\|_F$ stands for the Frobenius norm. Now, letting $B = B(\mathbf{m}) := F(\mathbf{m}) - D$, it can be shown that

$$\phi(\mathbf{m}) = \|B\|_F^2 = \operatorname{tr}(B^T B) = \mathbb{E}(\|B\mathbf{w}\|_2^2), \tag{1.6}$$

where $\mathbf{w}$ is a random vector drawn from any distribution satisfying $\mathbb{E}(\mathbf{w}\mathbf{w}^T) = \mathbb{I}$, $\operatorname{tr}(A)$ denotes the trace of the matrix A, and $\mathbb{E}$ denotes the expectation. Hence, approximating the misfit function $\phi(\mathbf{m})$ in (1.3) or in (1.4) is equivalent to approximating the corresponding matrix trace (or, equivalently, approximating the above expectation). The standard approach for doing this is based on a Monte Carlo method, where one generates n random vector realizations, $\mathbf{w}_j$, from a suitable probability distribution and computes the empirical mean

$$\widehat{\phi}(\mathbf{m}, n) := \frac{1}{n} \sum_{j=1}^{n} \|B(\mathbf{m})\mathbf{w}_j\|_2^2 \approx \phi(\mathbf{m}). \tag{1.7}$$

Note that $\widehat{\phi}(\mathbf{m}, n)$ is an *unbiased estimator* of $\phi(\mathbf{m})$, as we have $\phi(\mathbf{m}) = \mathbb{E}(\widehat{\phi}(\mathbf{m}, n))$. For the special case of the forward operators (1.2) considered in this paper, if $n \ll s$, then this procedure yields a very efficient algorithm for approximating the misfit (1.3) because

$$\sum_{i=1}^{s} \mathbf{f}(\mathbf{m}, \mathbf{q}_i) w_i = \mathbf{f}\left(\mathbf{m}, \sum_{i=1}^{s} \mathbf{q}_i w_i\right),$$

which can be computed with a single evaluation of $\mathbf{f}$ per realization of the random vector $\mathbf{w} = (w_1, \ldots, w_s)^T$.

Our assumption regarding the noise distribution leading to the ordinary least squares misfit function (1.3), although standard, is quite simplistic. Fortunately, however, it can be readily generalized in one of the following two ways:

1. The noise is independent and identically distributed (i.i.d.) as $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \Sigma)$ for all i, where $\Sigma \in \mathbb{R}^{l \times l}$ is the symmetric positive definite covariance matrix. In this case, the ML approach leads to minimizing the ℓ_2 misfit function

$$\phi_{(1)}(\mathbf{m}) := \sum_{i=1}^{s} \|C^{-1}\big(\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i\big)\|_2^2, \tag{1.8}$$

where $C \in \mathbb{R}^{l \times l}$ is any invertible matrix such that $\Sigma = CC^T$ (e.g., C can be the Cholesky factor of Σ). Thus,

$$\phi_{(1)}(\mathbf{m}) = \|C^{-1}\big(F(\mathbf{m}) - D\big)\|_F^2 = \|B(\mathbf{m})\|_F^2,$$

with $B(\mathbf{m}) := C^{-1}\big(F(\mathbf{m}) - D\big)$. The Monte Carlo approximation $\widehat{\phi}_{(1)}(\mathbf{m}, n)$ is then precisely as in (1.7) but with the newly defined $B(\mathbf{m})$.

2. The noise is independent but *not* identically distributed, satisfying instead $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma_i^2 \mathbb{I})$, $i = 1, 2, \ldots, s$, where $\sigma_i > 0$ are the standard deviations. Under this assumption, the ML approach yields the *weighted least squares* misfit function

$$\phi_{(2)}(\mathbf{m}) := \sum_{i=1}^{s} \frac{1}{\sigma_i^2} \|\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i\|_2^2. \tag{1.9}$$

We can further write this equation as

$$\phi_{(2)}(\mathbf{m}) = \|\big(F(\mathbf{m}) - D\big)C^{-1}\|_F^2,$$

where $C \in \mathbb{R}^{s\times s}$ denotes the diagonal matrix whose ith diagonal element is σ_i. Thus, with $B(\mathbf{m}) = (F(\mathbf{m}) - D)C^{-1}$ we can again apply (1.7) to obtain a similar Monte Carlo approximation $\widehat{\phi}_{(2)}(\mathbf{m}, n)$.

Now, if $n \ll s$, then the unbiased estimators $\widehat{\phi}_{(1)}(\mathbf{m}, n)$ and $\widehat{\phi}_{(2)}(\mathbf{m}, n)$ are obtained with an efficiency similar to that of $\widehat{\phi}(\mathbf{m}, n)$. In what follows, for notational simplicity, we just concentrate on $\phi(\mathbf{m})$ and $\widehat{\phi}(\mathbf{m}, n)$, but all the results hold almost verbatim also for (1.8) and (1.9).

Hence, the objective is to be able to generate as few realizations of $\mathbf{w}$ as possible for achieving acceptable approximations to the misfit function. Estimates on how large n must be to achieve a prescribed accuracy in a probabilistic sense have been derived in [1, 3, 34, 28]. However, the obtained bounds are typically not sufficiently tight to be practically useful. In this paper, we prove *tight* bounds for tail probabilities for such Monte Carlo approximations employing the standard normal distribution. These tail bounds are then used to obtain *necessary* and *sufficient* bounds on n, and we demonstrate that these bounds can be practically small and computable. Furthermore, using these results, we are able to better quantify the uncertainties in the highly efficient randomized algorithms proposed in [10, 30, 29]. Variants of such algorithms with better uncertainty quantification are derived.

This paper is organized as follows. In section 2, we develop and state theorems regarding the tight tail bounds promised above. The theory in this section relies upon some novel results regarding the extremal probabilities (i.e., maxima and minima of the tail probabilities) of nonnegative linear combinations of gamma random variables, which are proved in Appendix B.

In section 3, we present our stochastic algorithm variants for approximately minimizing (1.3) or (1.4) and discuss their novel elements. Subsequently, in section 4, the efficiency of the proposed algorithm variants is demonstrated using an important class of problems that arise often in practice. This is followed by conclusions and further thoughts in section 5.

2. Matrix trace estimation.

Let the matrix $A = B^T B \in \mathbb{R}^{s\times s}$ be implicit SPSD, and denote its trace by $\mathrm{tr}(A)$. As described in section 1, we approximate $\mathrm{tr}(A)$ by

$$\mathrm{tr}_n(A) := \frac{1}{n}\sum_{j=1}^{n} \mathbf{w}_j^T A \mathbf{w}_j, \tag{2.1}$$

where $\mathbf{w}_j \in \mathbb{R}^s \sim \mathcal{N}(0, \mathbb{I})$.

Now, given a pair of small positive real numbers (ε, δ), consider finding an appropriate sample size n such that

$$\Pr\Big(\mathrm{tr}_n(A) \geq (1-\varepsilon)tr(A)\Big) \geq 1-\delta, \tag{2.2a}$$

$$\Pr\Big(\mathrm{tr}_n(A) \leq (1+\varepsilon)tr(A)\Big) \geq 1-\delta. \tag{2.2b}$$

In [28] we showed that the inequalities (2.2) hold if

$$n > 8c, \quad \text{where } c = c(\varepsilon, \delta) = \varepsilon^{-2}\ln(1/\delta). \tag{2.3}$$

However, this bound on n can be rather pessimistic. Theorems 2.3 and 2.4 and Corollary 2.5 below provide tighter and hopefully more useful bounds on n. In order to prove these we require two additional theorems, Theorems 2.1 and 2.2, whose nontrivial and more technical proofs are deferred to Appendix B. Let $X \sim Gamma(\alpha, \beta)$ denote a gamma distributed random variable (r.v.) parametrized by shape α and rate β.[5]

Theorem 2.1 (monotonicity of cumulative distribution function of gamma r.v.). *Given parameters $0 < \alpha_1 < \alpha_2$, let $X_i \sim Gamma(\alpha_i, \alpha_i)$, $i = 1, 2$, be independent r.v.'s, and define $\Delta(x) := \Pr(X_2 < x) - \Pr(X_1 < x)$. Then we have that*

(i) *there is a unique point $x(\alpha_1, \alpha_2)$ such that $\Delta(x) < 0$ for $0 < x < x(\alpha_1, \alpha_2)$ and $\Delta(x) > 0$ for $x > x(\alpha_1, \alpha_2)$, and*

(ii) $1 \le x(\alpha_1, \alpha_2) \le \frac{2\sqrt{\alpha_1(\alpha_2-\alpha_1)}+1}{2\sqrt{\alpha_1(\alpha_2-\alpha_1)}}$.

Theorem 2.2 (extremal probabilities of linear combination of gamma r.v.'s). *Given shape and rate parameters $\alpha, \beta > 0$, let $X_i \sim Gamma(\alpha, \beta)$, $i = 1, 2, \ldots, n$, be i.i.d. gamma r.v.'s, and define $\Theta := \{\boldsymbol{\lambda} = \left(\lambda_1, \lambda_2, \ldots, \lambda_n\right)^T \mid \lambda_i \ge 0\ \forall i,\ \sum_{i=1}^n \lambda_i = 1\}$. Then we have*

$$m_n(x) := \min_{\boldsymbol{\lambda}\in\Theta} \Pr\left(\sum_{i=1}^n \lambda_i X_i < x\right) = \begin{cases} \Pr\left(\frac{1}{n}\sum_{i=1}^n X_i < x\right), & x < \frac{\alpha}{\beta}, \\ \Pr\left(X_1 < x\right), & x > \frac{2\alpha+1}{2\beta}, \end{cases}$$

$$M_n(x) := \max_{\boldsymbol{\lambda}\in\Theta} \Pr\left(\sum_{i=1}^n \lambda_i X_i < x\right) = \begin{cases} \Pr\left(X_1 < x\right), & x < \frac{\alpha}{\beta}, \\ \Pr\left(\frac{1}{n}\sum_{i=1}^n X_i < x\right), & x > \frac{2\alpha+1}{2\beta}. \end{cases}$$

Next we state and prove the results that are directly relevant to this section. Let us define

$$Q(n) := \frac{1}{n} Q_n,$$

where $Q_n \sim \chi_n^2$ denotes a chi-squared r.v. of degree n. Note that $Q(n) \sim Gamma(n/2, n/2)$. In case of several i.i.d. gamma r.v.'s of this sort, we refer to the jth r.v. by $Q_j(n)$.

Theorem 2.3 (necessary and sufficient condition for (2.2a)). *Given an SPSD matrix A of rank r and tolerances (ε, δ) as above, the following hold:*

(i) Sufficient condition: *There exists some integer $n_0 \ge 1$ such that*

$$\Pr\left(Q(n_0) < (1-\varepsilon)\right) \le \delta. \tag{2.4}$$

Furthermore, (2.2a) holds for all $n \ge n_0$.

(ii) Necessary condition: *If (2.2a) holds for some $n_0 \ge 1$, then for all $n \ge n_0$,*

$$P_{\varepsilon,r}^{-}(n) := \Pr\left(Q(nr) < (1-\varepsilon)\right) \le \delta. \tag{2.5}$$

[5]Recall that the probability density function of such an r.v. is

$$f(x) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, & x \ge 0, \\ 0, & x < 0. \end{cases}$$

(iii) Tightness: *If the r positive eigenvalues of A are all equal (note that this always happens if $r = 1$), then there is a positive integer n_0 satisfying (2.5) such that (2.2a) holds iff $n \geq n_0$.*

Proof. Since A is SPSD, it can be diagonalized by a unitary similarity transformation as $A = U^T \Lambda U$, where Λ is the diagonal matrix of eigenvalues sorted in nonincreasing order. Consider n random vectors $\mathbf{w}_i$, $i = 1, \ldots, n$, whose components are i.i.d. and drawn from the standard normal distribution, and define $\mathbf{z}_i = U\mathbf{w}_i$ for each i. Note that since U is unitary, the entries of $\mathbf{z}_i$ are i.i.d. standard normal variables, like the entries of $\mathbf{w}_i$. We have

$$\begin{aligned}\frac{\mathrm{tr}_n(A)}{\mathrm{tr}(A)} &= \frac{1}{n\ \mathrm{tr}(A)} \sum_{i=1}^{n} \mathbf{w}_i^T A \mathbf{w}_i = \frac{1}{n\ \mathrm{tr}(A)} \sum_{i=1}^{n} \mathbf{z}_i^T \Lambda \mathbf{z}_i = \frac{1}{n\ \mathrm{tr}(A)} \sum_{i=1}^{n} \sum_{j=1}^{r} \lambda_j z_{ij}^2 \\ &= \sum_{j=1}^{r} \frac{\lambda_j}{n\ \mathrm{tr}(A)} \sum_{i=1}^{n} z_{ij}^2 = \sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} Q_j(n),\end{aligned}$$

where the λ_j's appearing in the sums are positive eigenvalues of A. Now, noting that $\sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} = 1$, Theorem 2.2 yields

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} Q_j(n) \leq (1-\varepsilon)\right) \leq \Pr\big(Q(n) \leq (1-\varepsilon)\big) = P_{\varepsilon,1}^{-}(n), \tag{2.6a}$$

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} Q_j(n) \leq (1-\varepsilon)\right) \geq \Pr\big(Q(nr) \leq (1-\varepsilon)\big) = P_{\varepsilon,r}^{-}(n). \tag{2.6b}$$

In addition, for any given $r > 0$ and $\varepsilon > 0$, the function $P_{\varepsilon,r}^{-}(n)$ is monotonically decreasing on integers $n \geq 1$. This can be seen by Theorem 2.1 using the sequence $\alpha_i = (n_0 + (i-1))r/2$, $i \geq 1$. The claims now easily follow by combining (2.6) and this decreasing property. ■

Theorem 2.4 (necessary and sufficient condition for (2.2b)). *Given an SPSD matrix A of rank r and tolerances (ε, δ) as above, the following hold:*

(i) Sufficient condition: *If the inequality*

$$\Pr\big(Q(n_0) \leq (1+\varepsilon)\big) \geq 1 - \delta \tag{2.7}$$

is satisfied for some $n_0 > \varepsilon^{-1}$, then (2.2b) holds with $n = n_0$. Furthermore, there is always an $n_0 > \varepsilon^{-2}$ such that (2.7) is satisfied and, for such n_0, it follows that (2.2b) holds for all $n \geq n_0$.

(ii) Necessary condition: *If (2.2b) holds for some $n_0 > \varepsilon^{-1}$, then*

$$P_{\varepsilon,r}^{+}(n) := \Pr\big(Q(nr) \leq (1+\varepsilon)\big) \geq 1 - \delta, \tag{2.8}$$

with $n = n_0$. Furthermore, if $n_0 > \varepsilon^{-2} r^{-2}$, then (2.8) holds for all $n \geq n_0$.

(iii) Tightness: *If the r positive eigenvalues of A are all equal, then there is a smallest $n_0 > \varepsilon^{-2} r^{-2}$ satisfying (2.8) such that for any $n \geq n_0$, (2.2b) holds, and for any $\varepsilon^2 r^{-2} < n < n_0$, (2.2b) does not hold. If δ is small enough so that (2.8) does not hold for any $n \leq \varepsilon^2 r^{-2}$, then n_0 is both necessary and sufficient for (2.2b).*

Proof. The same unitary diagonalization argument as in the proof of Theorem 2.3 shows that

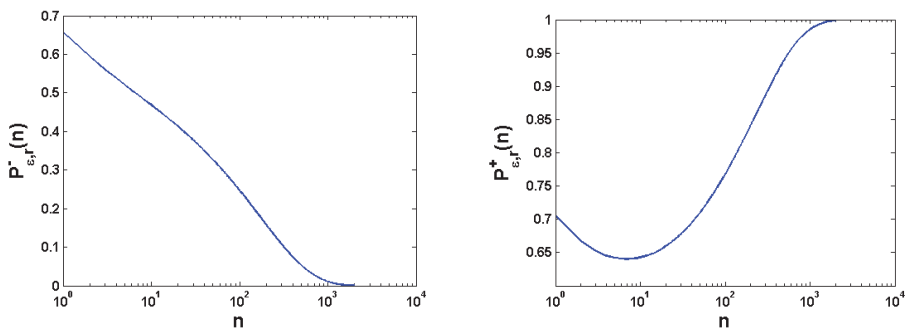
$$\Pr\left(\mathrm{tr}_n(A) < (1+\varepsilon)\mathrm{tr}(A)\right) = \Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} Q_j(n) < (1+\varepsilon)\right).$$

Now we see that if $n > \varepsilon^{-1}$, Theorem 2.2 with $\alpha = n/2$ yields

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} Q_j(n) \le (1+\varepsilon)\right) \ge \Pr\left(Q(n) \le (1+\varepsilon)\right) = P^{+}_{\varepsilon,1}(n), \tag{2.9a}$$

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{\mathrm{tr}(A)} Q_j(n) \le (1+\varepsilon)\right) \le \Pr\left(Q(nr) \le (1+\varepsilon)\right) = P^{+}_{\varepsilon,r}(n). \tag{2.9b}$$

In addition, for any given $r > 0$ and $\varepsilon > 0$, the function $P^{+}_{\varepsilon,r}(n)$ is monotonically increasing on integers $n > \varepsilon^{-2}r^{-2}$. This can be seen by Theorem 2.1 using the sequence $\alpha_i = (n_0 + (i-1))r/2$, $i \ge 1$. The claims now easily follow by combining (2.9) and this increasing property. ■



(a) (b)

Figure 1. *The curves of $P^{-}_{\varepsilon,r}(n)$ and $P^{+}_{\varepsilon,r}(n)$, defined in (2.5) and (2.8), for $\varepsilon = 0.1$ and $r = 1$: (a) $P^{-}_{\varepsilon,r}(n)$ decreases monotonically for all $n \ge 1$; (b) $P^{+}_{\varepsilon,r}(n)$ increases monotonically only for $n \ge n_0$, where $n_0 > 1$: according to Theorem 2.4, $n_0 = 100$ is safe, and this value does not disagree with the plot.*

Remarks.

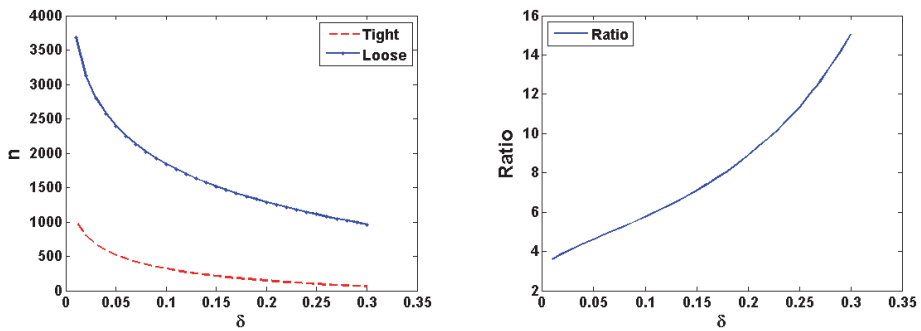
(i) Part (iii) of Theorem 2.4 states that if δ is not small enough, then n_0 might not be a necessary and sufficient sample size for the special matrices mentioned there, i.e., matrices with $\lambda_1 = \lambda_2 = \cdots = \lambda_r$. This can be seen from Figure 1(b): for $r = 1$, $\varepsilon = 0.1$, if $\delta = 0.33$, say, there is an integer $10 < n \le 100$ such that (2.2b) holds, so $n = 101$ is no longer a necessary sample size (although it is still sufficient).

(ii) Simulations show that the sufficient sample size obtained using Theorems 2.3 and 2.4 amounts to bounds of the form $\mathcal{O}\left(c(\varepsilon,\delta)g(\delta)\right)$, where $g(\delta) < 1$ is a decreasing function of δ and $c(\varepsilon,\delta)$ is as defined in (2.3). As such, for larger values of δ, i.e., when larger uncertainty is allowed, one can obtain significantly smaller sample sizes than the one

predicted by (2.3); see Figures 2 and 3. In other words, the difference between the above tighter conditions and (2.3) is increasingly more prominent as δ gets larger.

(iii) Note that the results in Theorems 2.3 and 2.4 are independent of the size of the matrix. In fact, the first items (i) in both theorems do not require any a priori knowledge about the matrix, other than it being SPSD. In order to compute the necessary sample sizes, though, one is required to also know the rank of the matrix.

(iv) The conditions in our theorems, despite their potentially ominous look, are actually simple to compute. Appendix C contains a short MATLAB code which calculates these necessary or sufficient sample sizes to satisfy the probabilistic accuracy guarantees (2.2), given a pair (ε, δ) (and the matrix rank r in case of necessary sample sizes). This code was used for generating Figures 2 and 3.



(a) (b)

Figure 2. *Comparing, as a function of δ, the sample size obtained from (2.4) and denoted by "tight" with that of (2.3) and denoted by "loose" for $\varepsilon = 0.1$ and $0.01 \le \delta \le 0.3$: (a) sufficient sample size, n, for (2.2a); (b) ratio of sufficient sample size obtained from (2.3) over that of (2.4). When δ is relaxed, our new bound is tighter than the older one by an order of magnitude.*

Combining Theorems 2.3 and 2.4, we can easily state conditions on the sample size n for which the condition

$$\Pr\left(|\mathrm{tr}_n(A) - \mathrm{tr}(A)| \le \varepsilon\ \mathrm{tr}(A)\right) \ge 1 - \delta \tag{2.10}$$

holds. We have the following immediate corollary.

Corollary 2.5 (necessary and sufficient condition for (2.10)). *Given an SPSD matrix A of rank r and tolerances (ε, δ) as above, the following hold:*

(i) Sufficient condition: *If the inequality*

$$\Pr\left((1-\varepsilon) \le Q(n_0) \le (1+\varepsilon)\right) \ge 1 - \delta \tag{2.11}$$

is satisfied for some $n_0 > \varepsilon^{-1}$, then (2.10) holds with $n = n_0$. Furthermore, there is always an $n_0 > \varepsilon^{-2}$ such that (2.11) is satisfied and, for such n_0, it follows that (2.10) holds for all $n \ge n_0$.

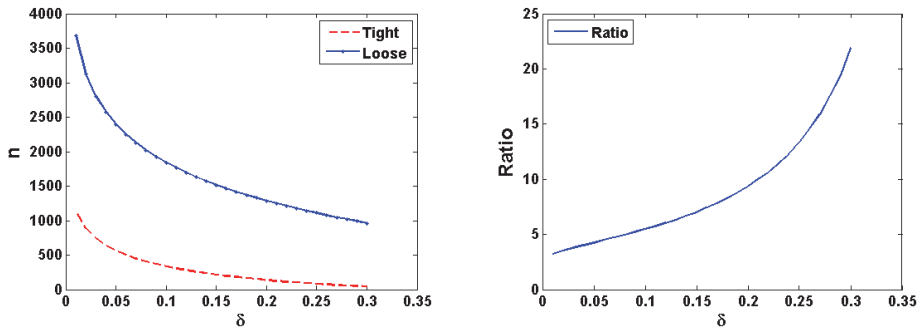
(ii) Necessary condition: *If (2.10) holds for some $n_0 > \varepsilon^{-1}$, then*

$$\Pr\left((1-\varepsilon) \le Q(nr) \le (1+\varepsilon)\right) \ge 1 - \delta, \tag{2.12}$$

with $n = n_0$. Furthermore, if $n_0 > \varepsilon^{-2}r^{-2}$, then (2.12) holds for all $n \geq n_0$.

(iii) Tightness: *If the r positive eigenvalues of A are all equal, then there is a smallest $n_0 > \varepsilon^{-2}r^{-2}$ satisfying (2.12) such that for any $n \geq n_0$, (2.10) holds, and for any $\varepsilon^{-2}r^{-2} < n < n_0$, (2.10) does not hold. If δ is small enough so that (2.12) does not hold for any $n \leq \varepsilon^{-2}r^{-2}$, then n_0 is both necessary and sufficient for (2.10).*

Remark. The necessary condition in Corollary 2.5(ii) is valid only for $n > \varepsilon^{-1}$ (this is a consequence of the condition (2.12) being tight, as shown in part (iii)). In [28], an "almost tight" necessary condition is given that works for all $n \geq 1$.



(a) (b)

Figure 3. *Comparing, as a function of δ, the sample size obtained from (2.7) and denoted by "tight" with that of (2.3) and denoted by "loose" for $\varepsilon = 0.1$ and $0.01 \leq \delta \leq 0.3$: (a) sufficient sample size, n, for (2.2b); (b) ratio of sufficient sample size obtained from (2.3) over that of (2.7). When δ is relaxed, our new bound is tighter than the older one by an order of magnitude.*

3. Randomized algorithms for solving large scale nonlinear least squares problems.

Consider the problem of decreasing the value of the original objective (1.3) to a desired level (e.g., satisfying a given tolerance) to recover the sought-after model, $\mathbf{m}$. With the sensitivity matrices

$$J_i(\mathbf{m}) = \frac{\partial \mathbf{f}(\mathbf{m}, \mathbf{q}_i)}{\partial \mathbf{m}}, \quad i = 1, \ldots, s,$$

we have the gradient

$$\nabla \phi(\mathbf{m}) = 2 \sum_{i=1}^{s} J_i^T(\mathbf{m})(\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i).$$

An iterative method such as modified Gauss–Newton (GN), L-BFGS, or nonlinear conjugate gradient is typically designed to decrease the value of the objective function using repeated calculations of the gradient. In this paper we follow [30] and employ variants of stabilized GN throughout, thus achieving a context in which to focus our attention on the new aspects of this work. In the kth iteration of such a method, having the current iterate $\mathbf{m}_k$, an update direction, $\delta\mathbf{m}_k$, is calculated. Then the iterate is updated as $\mathbf{m}_{k+1} \leftarrow \mathbf{m}_k + \alpha_k \delta\mathbf{m}_k$ for some appropriate step length α_k.

What is special in our context here is that the update direction, $\delta\mathbf{m}_k$, is calculated using the approximate misfit, $\widehat{\phi}(\mathbf{m}_k, n_k)$, defined as described in (1.7) (n_k is the sample size used for

this approximation in the kth iteration). Thus, we need to check or assess whether the value of the original objective is also decreased using this new iterate. The challenge is to do this as well as check for termination of the iteration process with a minimal number of evaluations of the prohibitively expensive original misfit function ϕ.

In this section, we extend the algorithms introduced in [30, 29] in the context of the more general nonlinear least squares (NLS) formulation (1.8) or (1.9), assuming that their corresponding noise distributions hold, although, as promised in section 1, we stick to the simpler notation (1.3), (1.7). Variants of modified stochastic steps in the original algorithms are presented, and using Theorems 2.3 and 2.4, the uncertainties in these steps are quantified. More specifically, in the main algorithm introduced in [30], following a stabilized GN iteration on the approximated objective function using the approximated misfit, the iterate is updated, and some (or all) of the following steps are performed:

(i) *Cross validation:* An approximate assessment of this iterate is performed in terms of sufficient decrease in the objective function using a control set of random combinations of measurements. More specifically, at the kth iteration with the new iterate $\mathbf{m}_{k+1}$, we test whether the condition

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \kappa \widehat{\phi}(\mathbf{m}_k, n_k) \tag{3.1}$$

(cf. (1.7)) holds for some $\kappa \leq 1$, employing an independent set of weight vectors used in both approximations of ϕ.

(ii) *Uncertainty check:* Upon success of cross validation, an inexpensive plausible termination test is performed where, given a tolerance ρ, we check for the condition

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \rho \tag{3.2}$$

using a fresh set of random weight vectors.

(iii) *Stopping criterion:* Upon success of the uncertainty check, an additional independent and potentially more rigorous termination test against the given tolerance ρ is performed (possibly using the original misfit function).

The role of the cross validation step within an iteration is to assess whether the true objective function at the current iterate has (sufficiently) decreased compared to the previous one. If this test fails, we deem that the current sample size is not sufficiently large to yield an update that decreases the original objective, and the fitting step needs to be repeated using a larger sample size; see [10]. In [30], this step was used heuristically, so the amount of uncertainty in such validation of the current iterate was not quantified. Consequently, there was no handle on the amount of false positives/negatives in such approximate evaluations (e.g., a sample size could be deemed too small while the stabilized GN iteration has in fact produced an acceptable iterate). In addition, in [30] the sample size for the uncertainty check was heuristically chosen. So this step was also performed with no control over the amount of uncertainty.

For the stopping criterion step in [30, 10], the objective function was accurately evaluated using all s experiments, which is clearly a very expensive choice for an algorithm termination check. This was a judicious decision made in order to be able to have a fairer comparison of the new and different methods proposed there. Replacement of this termination criterion by another independent heuristic "uncertainty check" is experimented with in [29].

In this section, we address the issues of quantifying the uncertainty in the validation, uncertainty check, and stopping criterion steps within a nonlinear iteration. In what follows, we assume for simplicity that the iterations are performed on the objective (1.3) using dynamic regularization (or iterative regularization [20, 9, 10]) where the regularization is performed implicitly. Extension to the case (1.4) is straightforward. Throughout, we assume to be given a pair of positive and small probabilistic tolerance numbers, (ε, δ).

3.1. Cross validation step with quantified uncertainty. The condition (3.1) is an independent, unbiased indicator of

$$\phi(\mathbf{m}_{k+1}) \leq \kappa\phi(\mathbf{m}_k),$$

which indicates sufficient decrease in the objective. If (3.1) is satisfied, then the current sample size, n_k, is considered sufficiently large to capture the original misfit well enough to produce a valid iterate, and the algorithm continues using the same sample size. Otherwise, the sample size is deemed insufficient and is increased. Using Theorems 2.3 and 2.4, we can now remove the heuristic characteristic as to *when* this sample size increase has been performed hitherto and present two variants of (3.1) where the uncertainties in the validation step are quantified.

Assume we have a sample size n_c such that

$$\Pr\left(\widehat{\phi}(\mathbf{m}_k, n_c) \leq (1+\varepsilon)\phi(\mathbf{m}_k)\right) \geq 1-\delta, \tag{3.3a}$$

$$\Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \geq (1-\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta. \tag{3.3b}$$

If in the procedure outlined above, after obtaining the updated iterate $\mathbf{m}_{k+1}$, we verify that

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \leq \kappa\left(\frac{1-\varepsilon}{1+\varepsilon}\right)\widehat{\phi}(\mathbf{m}_k, n_c), \tag{3.4}$$

then it follows from (3.3) that $\phi(\mathbf{m}_{k+1}) \leq \kappa\phi(\mathbf{m}_k)$ with a probability of at least $(1-\delta)^2$. In other words, success of (3.4) indicates that the updated iterate decreases the value of the original misfit (1.3) with a probability of at least $(1-\delta)^2$.

Alternatively, suppose that we have

$$\Pr\left(\widehat{\phi}(\mathbf{m}_k, n_c) \geq (1-\varepsilon)\phi(\mathbf{m}_k)\right) \geq 1-\delta, \tag{3.5a}$$

$$\Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \leq (1+\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta. \tag{3.5b}$$

Now, if instead of (3.4) we check whether or not

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \leq \kappa\left(\frac{1+\varepsilon}{1-\varepsilon}\right)\widehat{\phi}(\mathbf{m}_k, n_c), \tag{3.6}$$

then it follows from (3.5) that if the condition (3.6) is *not* satisfied, then $\phi(\mathbf{m}_{k+1}) > \kappa\phi(\mathbf{m}_k)$ with a probability of at least $(1-\delta)^2$. In other words, failure of (3.6) indicates that the updated iterate results in an insufficient decrease in the original misfit (1.3) with a probability of at least $(1-\delta)^2$.

We can replace (3.1) with either of the conditions (3.4) or (3.6) and use the conditions (2.4) or (2.7) to calculate the cross validation sample size, n_c. If the relevant check (3.4) or (3.6) fails, we deem the sample size used in the fitting step, n_k, to be too small to produce an iterate which decreases the original misfit (1.3) and consequently consider increasing the sample size, n_k. Note that since $\frac{1-\varepsilon}{1+\varepsilon} < 1 < \frac{1+\varepsilon}{1-\varepsilon}$, the condition (3.4) results in a more aggressive strategy for increasing the sample size used in the fitting step than the condition (3.6). Figure 8 in section 4 demonstrates this within the context of an application.

Remarks.

(i) Larger values of ε result in more aggressive (or relaxed) descent requirement by the condition (3.4) (or (3.6)).

(ii) As the iterations progress and we get closer to the solution, the decrease in the original objective could be less than what is imposed by (3.4). As a result, if ε is too large, we might never successfully pass the cross validation test. One useful strategy to alleviate this is to start with a larger ε, decreasing it as we get closer to the solution. A similar strategy can be adopted for the case when the condition (3.6) is used as a cross validation: as the iterations get closer to the solution, one can make the condition (3.6) less relaxed by decreasing ε.

3.2. Uncertainty check with quantified uncertainty and efficient stopping criterion. The usual test for terminating the iterative process is to check whether

$$\phi(\mathbf{m}_{k+1}) \leq \rho \tag{3.7}$$

for a given tolerance ρ. However, this can be very expensive in our current context; see section 4.1 and Tables 1 and 2 for examples of a scenario where one misfit evaluation using the entire data set can be as expensive as the entire cost of an efficient, complete algorithm. In addition, if the exact value of the tolerance ρ is not known (which is usually the case in practice), one should be able to reflect such uncertainty in the stopping criterion and perform a softer version of (3.7). Hence, it could be useful to have an algorithm which allows one to adjust the cost and accuracy of such an evaluation in a quantifiable way and find the balance that is suitable to particular objectives and computational resources.

Regardless of the issues of cost and accuracy, this evaluation should be carried out as rarely as possible and only when deemed timely. In [30], we addressed this by employing an "uncertainty check" (3.2) as described earlier in this section, heuristically. Using Theorems 2.3 and 2.4, we now devise variants of (3.2) with quantifiable uncertainty. Subsequently, again using Theorems 2.3 and 2.4, we present a much cheaper stopping criterion than (3.7) which, at the same time, reflects our uncertainty in the given tolerance.

Assume that we have a sample size n_u such that

$$\Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \geq (1-\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta. \tag{3.8}$$

If the updated iterate, $\mathbf{m}_{k+1}$, successfully passes the cross validation test, then we check for

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \leq (1-\varepsilon)\rho. \tag{3.9}$$

If this holds too, then it follows from (3.8) that $\phi(\mathbf{m}_{k+1}) \leq \rho$ with a probability of at least $(1-\delta)$. In other words, success of (3.9) indicates that the misfit is likely to be below the tolerance with a probability of at least $(1-\delta)$.

Alternatively, suppose that

$$\Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \leq (1+\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta, \tag{3.10}$$

and instead of (3.9) we check for

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \leq (1+\varepsilon)\rho. \tag{3.11}$$

Then it follows from (3.10) that if the condition (3.11) is *not* satisfied, then $\phi(\mathbf{m}_{k+1}) > \rho$ with a probability of at least $(1-\delta)$. In other words, failure of (3.11) indicates that using the updated iterate, the misfit is likely to still be above the desired tolerance with a probability of at least $(1-\delta)$.

We can replace (3.2) with the condition (3.9) (or (3.11)) and use the condition (2.4) (or (2.7)) to calculate the uncertainty check sample size, n_u. If the test (3.9) (or (3.11)) fails, then we skip the stopping criterion check and continue iterating. Note that since $(1-\varepsilon) < 1 < (1+\varepsilon)$, the condition (3.9) results in fewer false positives than the condition (3.11). On the other hand, the condition (3.11) is expected to result in fewer false negatives than the condition (3.9). The choice of either alternative is dependent on one's requirements and resources and the application on hand.

The stopping criterion step can be performed in the same way as the uncertainty check but potentially with higher certainty in the outcome. In other words, for the stopping criterion we can choose a smaller δ, resulting in a larger sample size n_t satisfying $n_t > n_u$, and check for satisfaction of either

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_t) \leq (1-\varepsilon)\rho \tag{3.12a}$$

or

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_t) \leq (1+\varepsilon)\rho. \tag{3.12b}$$

Clearly the condition (3.12b) is softer than (3.12a): a successful (3.12b) is only necessary and not sufficient for concluding that (3.7) holds with the prescribed probability.

In practice, when the value of the stopping criterion threshold, ρ, is not *exactly* known (it is often crudely estimated using the measurements), one can reflect such uncertainty in ρ by choosing an appropriately large δ. Smaller values of δ reflect a higher certainty in ρ and a more rigid stopping criterion.

Remarks.

(i) If ε is large, then using (3.12a), one might run the risk of overfitting. Similarly, using (3.12b) with large ε, there is a risk of underfitting. Thus, appropriate values of ε need to be considered in accordance with the application and one's computational resources and experience.

(ii) The same issues regarding large ε arise when employing the uncertainty check condition (3.9) (or (3.11)): large ε might increase the frequency of false negatives (or positives).

3.3. Algorithm. We now present an efficient, stochastic, iterative algorithm for approximately solving NLS formulations of (1.3) or (1.4). By performing cross validation, an uncertainty check, and a stopping criterion as described in sections 3.1 and 3.2, we can devise eight variants of Algorithm 1 below. Depending on the application, the variant of choice can be selected appropriately. More specifically, cross validation, an uncertainty check, and a stopping criterion can, respectively, be chosen to be one of the following combinations (referring to their equation numbers):

(i) (3.4 - 3.9 - 3.12a)	(ii) (3.4 - 3.9 - 3.12b)	(iii) (3.4 - 3.11 - 3.12a)	(iv) (3.4 - 3.11 - 3.12b)
(v) (3.6 - 3.9 - 3.12a)	(vi) (3.6 - 3.9 - 3.12b)	(vii) (3.6 - 3.11 - 3.12a)	(viii) (3.6 - 3.11 - 3.12b)

Remarks.

(i) The sample size, n_k, used in the fitting step of Algorithm 1 could in principle be determined by Corollary 2.5, using a pair of tolerances $(\varepsilon_f, \delta_f)$. If cross validation (3.4) (or (3.6)) fails, the tolerance pair $(\varepsilon_f, \delta_f)$ is reduced to obtain, in the next iteration, a larger fitting sample size, n_{k+1}. This would give a sample size which yields a quantifiable approximation with a desired relative accuracy. However, in the presence of all the added safety steps described in this section, we have found in practice that Algorithm 1 is capable of producing a satisfying recovery, even with a significantly smaller n_k than that predicted by Corollary 2.5. Thus, the "*how*" of the fitting sample size increase is left to be heuristic (as opposed to its "*when*," which is quantified as described in section 3.1).

(ii) In the algorithm below, we consider only fixed values (i.e., independent of k) for ε and δ. One can easily modify Algorithm 1 to incorporate nonstationary values which adapt to the iteration process, as mentioned in the closing remark of section 3.1.

In Algorithm 1, when we draw vectors $\mathbf{w}_i$ for some purpose, we always draw them independently from the standard normal distribution.

4. A practical application. In this section, we demonstrate the efficacy of Algorithm 1 by applying it to an important class of problems that arise often in practice: large scale PDE inverse problems with many measurements. We show below the capability of our method by applying it to such examples in the context of the direct current resistivity/electrical impedance tomography problem, as in [10, 30, 29].

4.1. PDE inverse problems with many measurements. The context considered here is one where each evaluation of $\mathbf{f}_i(\mathbf{m})$ in (1.2) is computationally expensive. The evaluation of the misfit function $\phi(\mathbf{m})$ is especially costly when many experiments, involving different combinations of sources and receivers, are employed in order to obtain reconstructions of acceptable quality. The sought-after model $\mathbf{m}$ is a discretization of the function $m(\mathbf{x})$ as described in section 1, and

$$\mathbf{f}_i(\mathbf{m}) = P_i\mathbf{u}_i = P_i L(\mathbf{m})^{-1}\mathbf{q}_i. \tag{4.1a}$$

Here we write the PDE system in discretized form as

$$L(\mathbf{m})\mathbf{u}_i = \mathbf{q}_i, \quad i = 1, \ldots, s, \tag{4.1b}$$

where $\mathbf{u}_i \in \mathbb{R}^{l_q}$ is the ith field, $\mathbf{q}_i \in \mathbb{R}^{l_q}$ is the ith source, and L is a square matrix discretizing the PDEs plus appropriate side conditions. Furthermore, the given projection matrices P_i are

Algorithm 1. Solve NLS formulation of (1.3) (or (1.4)) using an uncertainty check, cross validation, and a cheap stopping criterion.

Given: sources $\mathbf{q}_i$, $i = 1, \ldots, s$, measurements $\mathbf{d}_i$, $i = 1, \ldots, s$, stopping criterion level ρ, objective function sufficient decrease factor $\kappa \leq 1$, pairs of small numbers $(\varepsilon_c, \delta_c)$, $(\varepsilon_u, \delta_u)$, $(\varepsilon_t, \delta_t)$, and initial guess $\mathbf{m}_0$.

Initialize:

- $\mathbf{m} = \mathbf{m}_0$, $n_0 = 1$
- Calculate the cross validation sample size, n_c, as described in section 3.1 with $(\varepsilon_c, \delta_c)$.
- Calculate the sample sizes for the uncertainty check, n_u, and the stopping criterion, n_t, as described in section 3.2 with $(\varepsilon_u, \delta_u)$ and $(\varepsilon_t, \delta_t)$, respectively.

for $k = 0, 1, 2, \ldots$ until termination **do**

  Fitting:

  - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_k$.
  - Approximate the misfit term and potentially its gradient in (1.3) or (1.4) using (1.7) with the above weights and $n = n_k$.
  - Find an update for the objective function using the approximated misfit (1.7).

  Cross Validation:

  - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_c$.

  if (3.4) (or (3.6)) holds **then**

    Uncertainty Check:

    - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_u$.

    if (3.9) (or (3.11)) holds **then**

      Stopping Criterion:

      - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_t$.

      if (3.12a) (or (3.12b)) holds **then**

        - Terminate

      end if

    end if

    - Set $n_{k+1} = n_k$.

  else

    - **Sample Size Increase**: for example, set $n_{k+1} = \min(2n_k, s)$.

  end if

end for

such that $\mathbf{f}_i(\mathbf{m})$ predicts the ith data set. Note that the notation (4.1b) reflects an assumption of linearity in $\mathbf{u}$ but not in $\mathbf{m}$ [30].

If the locations where data are measured do not change from one experiment to another, i.e., $P = P_i$ for all i, then we get

$$\mathbf{f}(\mathbf{m}, \mathbf{q}_i) = PL(\mathbf{m})^{-1}\mathbf{q}_i, \tag{4.2}$$

and the linearity assumption of $\mathbf{f}(\mathbf{m}, \mathbf{q})$ in $\mathbf{q}$ is satisfied. Thus, we can use Algorithm 1 to efficiently recover $\mathbf{m}$ and be quantifiably confident in the recovered model. If the P_i's are different across experiments, there are methods to extend the existing data set to one where

all sources share the same receivers; see [29, 17]. Using these methods (when they apply!), one can effectively transform the problem (4.1a) to (4.2), for which Algorithm 1 can be employed.

There are several problems of practical interest in the form (1.3), (4.1), where the use of many experiments, resulting in a large number s, is crucial for obtaining credible reconstructions in practical situations. These include electromagnetic data inversion in mining exploration (see, e.g., [24, 12, 16, 25]), seismic data inversion in oil exploration (see, e.g., [14, 21, 27]), diffuse optical tomography (DOT) (see, e.g., [2, 4]), quantitative photo-acoustic tomography (QPAT) (see, e.g., [15, 35]), direct current (DC) resistivity (see, e.g., [31, 26, 19, 18, 10]), and electrical impedance tomography (EIT) (see, e.g., [5, 8, 11]).

Our examples are performed in the context of solving the DC resistivity problem. The PDE has the form

$$\nabla \cdot (\mu(\mathbf{x})\nabla u) = q(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{4.3a}$$

where $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, and $\mu(\mathbf{x})$ is a conductivity function which may be rough[6] (e.g., discontinuous). However, the PDE is coercive: there is a constant $\mu_0 > 0$ such that $\mu(\mathbf{x}) \geq \mu_0$ for all $\mathbf{x} \in \Omega$. It is possible to inject some a priori information on μ, when available, via a parametrization of $\mu(\mathbf{x})$ in terms of $m(\mathbf{x})$ using an appropriate transfer function ψ as $\mu(\mathbf{x}) = \psi(m(\mathbf{x}))$. For example, ψ can be chosen so as to ensure that the conductivity stays positive and bounded away from 0, as well as to incorporate bounds, which are often known in practice, on the sought-after conductivity function. Some possible choices of function ψ are described in [30, Appendix A]. Here we take Ω to be the unit square in two dimensions and assume the homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial n} = 0, \quad \mathbf{x} \in \partial\Omega. \tag{4.3b}$$

The inverse problem is then to recover m in Ω from sets of measurements of u on the domain's boundary for different sources q. Details of the numerical methods employed here, both for defining the predicted data $\mathbf{f}$ and for solving the inverse problem in appropriately transformed variables, can be found in [30, Appendix A].

4.2. Numerical experiments. Below we consider two examples, each having a piecewise constant "exact solution," or "true model," used to synthesize data:

(E.1) In our simpler model a target object with conductivity $\mu_t = 1$ has been placed in a background medium with conductivity $\mu_b = 0.1$ (see Figure 4(a)).

(E.2) In a slightly more complex setting a conductive object with conductivity $\mu_c = 0.01$, as well as a resistive one with conductivity $\mu_r = 1$, have been placed in a background medium with conductivity $\mu_b = 0.1$ (see Figure 6(a)). Note that the recovery of the model in example (E.2) is more challenging than example (E.1) since here the dynamic range of the conductivity is much larger.

Details of the numerical setup for the following examples are given in Appendix A.

4.2.1. Example (E.1). We carry out the eight variants of Algorithm 1 for the parameter values $(\varepsilon_c, \delta_c) = (0.05, 0.3)$, $(\varepsilon_u, \delta_u) = (0.1, 0.3)$, $(\varepsilon_t, \delta_t) = (0.1, 0.1)$, and $\kappa = 1$. The resulting

[6] In theory, the conductivity function is defined so that $\mu \in L_\infty(\Omega)$, and hence it can be very rough.

Table 1

Example (E.1). *Work in terms of the number of PDE solves for all variants of Algorithm* 1*, described in section* 3.3 *and indicated here by* (i)–(viii). *The "vanilla" count is also given as a reference.*

Vanilla	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
436,590	4,058	4,028	3,764	3,282	4,597	3,850	3,734	3,321

Table 2

Example (E.2). *Work in terms of the number of PDE solves for all variants of Algorithm* 1*, described in section* 3.3 *and indicated here by* (i)–(viii). *The "vanilla" count is also given as a reference.*

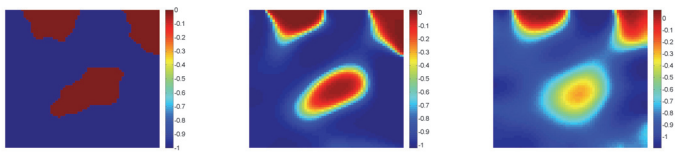
Vanilla	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
476,280	5,631	5,057	5,011	3,990	6,364	4,618	4,344	4,195

total count of PDE solves, which is the main computational cost of the iterative solution of such inverse problems, is reported in Tables 1 and 2. As a point of reference, we also include the total PDE count using the "plain vanilla" stabilized GN method which employs the entire set of s experiments at every iteration and misfit estimation task. The recovered conductivities are displayed in Figures 5 and 7, demonstrating that employing Algorithm 1 can drastically reduce the total work while obtaining equally acceptable reconstructions.

For the calculations displayed here we have employed *dynamical regularization* [9, 10]. In this method, there is no explicit regularization term $R(\mathbf{m})$ in (1.4) and the regularization is done implicitly and iteratively.

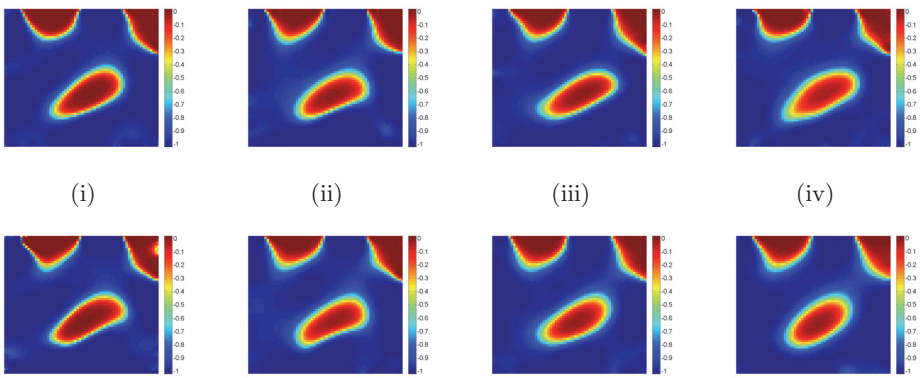
The quality of reconstructions obtained by the various variants in Figure 5 is comparable to that of the "vanilla" with $s = 3,969$ in Figure 4(b). In contrast, employing only $s = 49$ data sets corresponding to similar experiments distributed over a coarser grid yields an inferior reconstruction in Figure 4(c). The cost of this latter run is $5,684$ PDE solves, which is more expensive than our randomized algorithms for the much larger s. Furthermore, comparing Figures 4(b) and 5 to Figures 3 and 4 of [29], which shows similar results for $s = 961$ data sets, we again see a relative improvement in reconstruction quality. All of this goes to show that a large number of measurements s can be crucial for better reconstructions. Thus, it is not the case that one can dispense with a large portion of the measurements and still expect the same quality reconstructions. Hence, it is indeed useful to have algorithms such as Algorithm 1 that, while taking advantage of the entire available data, can efficiently carry out the computations and yet obtain credible reconstructions.

We have resisted the temptation to make comparisons between values of $\phi(\mathbf{m}_{k+1})$ and $\hat{\phi}(\mathbf{m}_{k+1})$ for various iterates. There are two major reasons for that. The first is that $\hat{\phi}$ values in bounds such as (3.4), (3.6), (3.9), (3.11), and (3.12) are different and are always compared against tolerances in context that are based on noise estimates. In addition, the sample sizes that we used for uncertainty checks and stopping criteria, since they are given by Theorems 2.3 and 2.4, already determine how far the estimated misfit is from the true misfit. The second (and more important) reason is that in such a highly diffusive forward problem as DC resistivity, misfit values are typically far closer to one another than the resulting reconstructed models $\mathbf{m}$ are. A good misfit is merely a necessary condition, which can fall significantly short of being sufficient, for a good reconstruction [16, 29].



(a) (b) (c)

Figure 4. *Example* (E.1)*. Plots of log-conductivity:* (a) *true model;* (b) *vanilla recovery with* $s = 3,969$*;* (c) *vanilla recovery with* $s = 49$*. The vanilla recovery using only* 49 *measurement sets is clearly inferior, showing that a large number of measurement sets can be crucial for better reconstructions.*



(i) (ii) (iii) (iv)

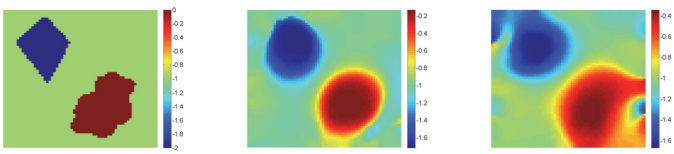
(v) (vi) (vii) (viii)

Figure 5. *Example* (E.1)*. Plots of log-conductivity of the recovered model using the* 8 *variants of Algorithm* 1*, described in section* 3.3 *and indicated here by* (i)–(viii)*. The quality of reconstructions is generally comparable to that of plain vanilla with* $s = 3,969$ *and across variants.*

4.2.2. Example (E.2). Here we have imposed prior knowledge on the *discontinuous* model in the form of total variation (TV) regularization [11, 7, 6]. Specifically, $R(\mathbf{m})$ in (1.4) is the discretization of the TV functional $\int_\Omega |\nabla m(\mathbf{x})|$. For each recovery, the regularization parameter, α, has been chosen by trial and error within the range $[10^{-6}, 10^{-3}]$ to visually yield the best quality recovery.

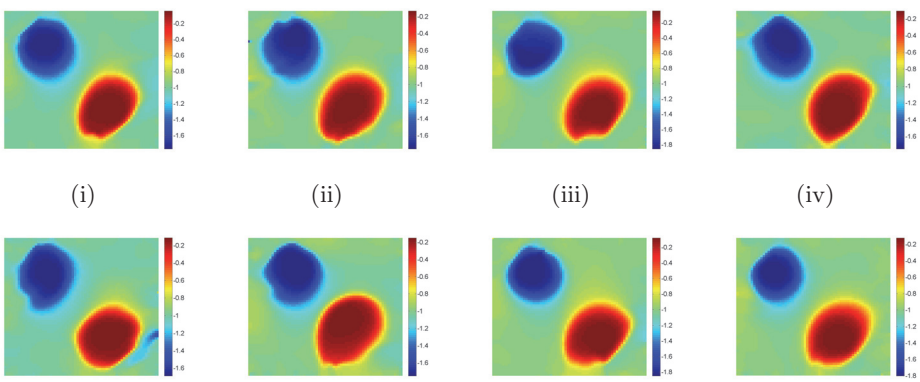
Table 2 and Figures 6 and 7 tell a story similar to that in example (E.1). The quality of reconstructions with $s = 3,969$ by the various variants, displayed in Figure 7, is comparable to that of the "vanilla" version in Figure 6(b), yet it is obtained at only a fraction (about 1%) of the cost. The "vanilla" solution for $s = 49$ displayed in Figure 6(c) costs 5,978 PDE solves, which again is a higher cost for an inferior reconstruction compared to our Algorithm 1.

It is clear from Tables 1 and 2 that for most of these examples, variants (i)–(iv), which use the more aggressive cross validation (3.4), are at least as efficient as their respective counterparts, namely, variants (v)–(viii), which use (3.6). This suggests that, sometimes, a more aggressive sample size increase strategy may be a better option; see also the numerical examples in [30]. Notice that for all variants, the entire cost of the algorithm is comparable to one single evaluation of the misfit function $\phi(\mathbf{m})$ using the full data set!



(a) (b) (c)

Figure 6. *Example* (E.2). *Plots of log-conductivity:* (a) *true model;* (b) *vanilla recovery with* $s = 3,969$*;* (c) *vanilla recovery with* $s = 49$*. The vanilla recovery using only* 49 *measurement sets is clearly inferior, showing that a large number of measurement sets can be crucial for better reconstructions.*



(i) (ii) (iii) (iv)

(v) (vi) (vii) (viii)

Figure 7. *Example* (E.2). *Plots of log-conductivity of the recovered model using the eight variants of Algorithm* 1*, described in section* 3.3 *and indicated here by* (i)–(viii)*. The quality of reconstructions is generally comparable to each other and that of plain vanilla with* $s = 3,969$*.*

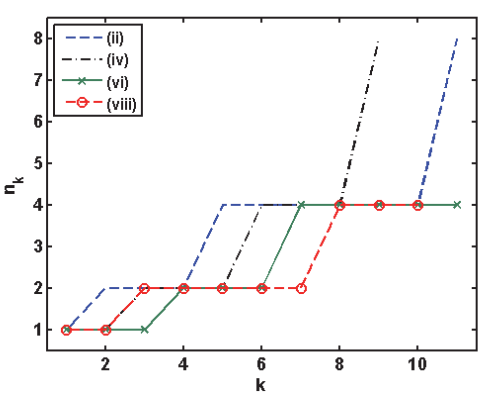


Figure 8. *Example* (E.2). *Growth of the fitting sample size,* n_k*, as a function of the iteration* k*, upon using cross validation strategies* (3.4) *and* (3.6)*. The graph shows the fitting sample size growth for variants* (ii) *and* (vi) *of Algorithm* 1*, as well as their counterparts, namely, variants* (vi) *and* (viii)*. Observe that for variants* (ii) *and* (iv) *where* (3.4) *is used, the fitting sample size grows at a more aggressive rate than for variants* (vi) *and* (viii) *where* (3.6) *is used.*

5. Conclusions. In this paper we have proved tight necessary and sufficient conditions for the sample size, n, required to reach, with a probability of at least $1-\delta$, (one-sided) approximations for $\mathrm{tr}(A)$ to within a relative tolerance ε. All of the sufficient conditions are computable in practice and do not assume any a priori knowledge about the matrix. If the rank of the matrix is known, then the necessary bounds can also be computed in practice.

Subsequently, using these conditions, we have presented eight variants of a general purpose algorithm for solving an important class of large scale NLS problems. These algorithms can be viewed as an extended version of those in [30, 29], where the uncertainty in most of the stochastic steps is quantified. Such uncertainty quantification allows one to have better control over the behavior of the algorithm and have more confidence in the recovered solution. The resulting algorithm is presented in section 3.3.

Furthermore, we have demonstrated the performance of our algorithm using an important class of problems which arise often in practice, namely, PDE inverse problems with many measurements. By examining our algorithm in the context of the DC resistivity problem as an instance of such a class of problems, we have shown that Algorithm 1 can recover solutions with remarkable efficiency. This efficiency is comparable to similar heuristic algorithms proposed in [30, 29]. The added advantage here is that with the uncertainty being quantified, the user can have more confidence in the approximate solution obtained by our algorithm.

Tables 1 and 2 show the amount of work (in PDE solves) of the eight variants of our algorithm. Compared to a similar algorithm which uses the entire data set, an efficiency improvement by two orders of magnitude is observed. For most of the examples considered, the same tables also show that the more aggressive cross validation strategy (3.4) is at least as efficient as the more relaxed strategy (3.6). A thorough comparison of the behavior of these cross validation strategies (and all of the variants in general) on different examples and model problems is left for future work.

Appendix A. Numerical experiment setup. The experimental setting we use in section 4.1 is as follows: for each experiment i, there is a positive unit point source at $\mathbf{x}_1^i$ and a negative sink at $\mathbf{x}_2^i$, where $\mathbf{x}_1^i$ and $\mathbf{x}_2^i$ denote two locations on the boundary $\partial\Omega$. Hence in (4.3a) we must consider sources of the form $q_i(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{x}_1^i) - \delta(\mathbf{x}-\mathbf{x}_2^i)$, i.e., a difference of two δ-functions, and $\mathbf{q}_i$ is the discretization of q_i over the grid.

For our experiments, when we place a source on the left boundary, we place the corresponding sink on the right boundary in every possible combination. Hence, having p locations on the left boundary for the source would result in $s = p^2$ experiments. The receivers are located at the top and bottom boundaries. No source or receiver is placed at the corners.

We then generate data $\mathbf{d}_i$ by using a chosen true model (or ground truth) and a source-receiver configuration as described above. This is followed by peppering these values with 2% additive Gaussian noise to create the data $\mathbf{d}_i$ used in our experiments. Specifically, for an additive noise of 2%, denoting the "clean data" $l \times s$ matrix by D^*, we reshape this matrix into a vector $\mathbf{d}^*$ of length sl, calculate the standard deviation $\sigma = .02\|\mathbf{d}^*\|/\sqrt{sl}$, and define $D = D^* + \sigma * \mathtt{randn}(\mathtt{l}, \mathtt{s})$ using the MATLAB random generator function `randn`. Following the celebrated Morozov *discrepancy principle* [33, 13, 23, 22], the stopping tolerance is set to be $\rho = \tau\sigma^2 sl$. As in [30], we choose $\tau = 1.2$.

For all numerical experiments, in order to avoid committing "inverse crime," the "true

field" is calculated on a grid that is twice as fine as the one used to reconstruct the model. For the two-dimensional examples, the reconstruction is done on a uniform grid of size 64^2 with $s = 3,969$ experiments in the setup described above.

As for an iterative method to decrease the value of the objective function, we employ variants of stabilized GN; see [30, Appendix A] for more details. At each iteration of such a method, an update direction needs to be calculated. Usually another iterative scheme is used to calculate the update. We employ preconditioned conjugate gradient (PCG) as our inner iterative solver. The PCG iteration limit is set to 20, and the PCG tolerance was chosen to be 10^{-3}. We again refer the reader to [30, Appendix A] for more details. The initial guess for GN iterations is $\mathbf{m}_0 = \mathbf{0}$.

For the transfer function ψ described in section 4.1, we use the formulation [30, eq. (6.3)] with $\mu_{\max} = 1.2 \max \mu(\mathbf{x})$ and $\mu_{\min} = .83 \min \mu(\mathbf{x})$.

Appendix B. Extremal probabilities of linear combinations of gamma random variables.

In this appendix we prove Theorems 2.1 and 2.2. Such results were obtained in [32] for the special case where the X_i's are chi-squared r.v,'s of degree 1 (corresponding to $\alpha = \beta = 1/2$). Here we extend those results to arbitrary gamma random variables, including chi-squared of arbitrary degree, exponential, Erlang, etc.

In what follows, for a gamma r.v. $X \sim Gamma(\alpha, \beta)$, we use the notation f_X for its probability density function (PDF) and F_X for its cumulative distribution function (CDF).

The objective in the proof of Theorem 2.2 is to find the extrema (with respect to $\boldsymbol{\lambda} \in \Theta$) of the CDF of r.v. $\sum_{i=1}^n \lambda_i X_i$. This is mainly achieved by perturbation arguments, employing a key identity which is derived using Laplace transforms. Using our perturbation arguments with this identity and employing Lemma B.2, we obtain that at any extremum, we must have either $\lambda_1, \lambda_2 > 0$ and $\lambda_3 = \cdots = \lambda_n = 0$ or for some $i \leq n$ we must get $\lambda_1 = \cdots = \lambda_i > 0$ and $\lambda_{i+1} = \cdots = \lambda_n = 0$. (Note that this latter case covers the "corners" as well.) In the former case, Lemma B.3 is used to distinguish between the minima and maxima for different values of x. These results along with Theorem 2.1 are then used to prove Theorem 2.2.

Three lemmas are used in the proofs of our two theorems. Lemma B.1 describes some properties of the PDF of nonnegative linear combinations of arbitrary gamma r.v.'s, such as analyticity and vanishing derivatives at zero. Lemma B.2 describes the monotonicity property of the mode of the PDF of nonnegative linear combinations of a *particular* set of gamma r.v.'s, which is useful for the proof of Theorem 2.2. Lemma B.3 gives some properties regarding the mode of the PDF of convex combinations of two *particular* gamma r.v.'s, which is used in proving Theorems 2.1 and 2.2.

B.1. Lemmas.

We next state and prove the lemmas summarized above.

Lemma B.1 (generalization of [32, Lemma A]). *Let $X_i \sim Gamma(\alpha_i, \beta_i)$, $i = 1, 2, \ldots, n$, be independent r.v.'s, where $\alpha_i, \beta_i > 0$ for all i. Define $Y_n := \sum_{i=1}^n \lambda_i X_i$ for $\lambda_i > 0$ for all i and $\rho_j := \sum_{i=1}^j \alpha_i$. Then for the PDF of Y_n, f_{Y_n}, we have the following:*

(i) *$f_{Y_n} > 0$ for all $x > 0$;*
(ii) *f_{Y_n} is analytic on $\mathbb{R}^+ = \{x | x > 0\}$;*
(iii) *$f_{Y_n}^{(k)}(0) = 0$ if $0 \leq k < \rho_n - 1$, where $f_{Y_n}^{(k)}$ denotes the kth derivative of f_{Y_n}.*

Proof. The proof is done by induction on n. For $n = 2$ we have

$$f_{Y_2}(x) = \int_0^\infty f_{\lambda_1 X_1}(y) f_{\lambda_2 X_2}(x-y)dy = \frac{(\beta_1/\lambda_1)^{\alpha_1}(\beta_2/\lambda_2)^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^x y^{\alpha_1-1}(x-y)^{\alpha_2-1} e^{-\frac{\beta_1 y}{\lambda_1} - \frac{\beta_2(x-y)}{\lambda_2}} dy.$$

Now the change of variables $y \to x\cos^2\theta_1$ would yield

$$f_{Y_2}(x) = 2\frac{(\beta_1/\lambda_1)^{\alpha_1}(\beta_2/\lambda_2)^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)} x^{(\alpha_1+\alpha_2-1)} \int_0^{\frac{\pi}{2}} (\cos\theta_1)^{2\alpha_1-1}(\sin\theta_1)^{2\alpha_2-1} e^{-x(\frac{\beta_1\cos^2\theta_1}{\lambda_1} + \frac{\beta_2\sin^2\theta_1}{\lambda_2})} d\theta_1.$$

By induction on n, one can show that for arbitrary $n \geq 2$,

$$f_{Y_n}(x) = 2^{n-1}\left(\prod_{i=1}^n \frac{(\beta_i/\lambda_i)^{\alpha_i}}{\Gamma(\alpha_i)}\right) x^{\rho_n-1} \int_{D^{n-1}} P_n(\Theta_{n-1})Q_n(\Theta_{n-1})e^{-xR_n(\Theta_{n-1})}\mathbf{d}\Theta_{n-1}, \tag{B.1a}$$

where

$$P_n(\Theta_{n-1}) = \prod_{j=1}^{n-1}(\cos\theta_j)^{2\rho_j-1}, \quad Q_n(\Theta_{n-1}) = \prod_{j=1}^{n-1}(\sin\theta_j)^{2\alpha_{j+1}-1}, \tag{B.1b}$$

the function $R_n(\Theta_{n-1})$ satisfies the recurrence relation

$$R_n(\Theta_{n-1}) = \cos^2\theta_{n-1}R_{n-1}(\Theta_{n-2}) + \beta_n\lambda_n^{-1}\sin^2\theta_{n-1} \quad \forall n \geq 2, \tag{B.1c}$$

$$R_1(\Theta_0) = \beta_1/\lambda_1, \tag{B.1d}$$

and $\mathbf{d}\Theta_{n-1}$ denotes the $(n-1)$-dimensional Lebesgue measure with the domain of integration

$$D^{n-1} = (0,\pi/2) \times (0,\pi/2) \times \cdots \times (0,\pi/2) = (0,\pi/2)^{n-1} \subset \mathbb{R}^{n-1}. \tag{B.1e}$$

Now the claims in Lemma B.1 follow from (B.1). ■

Lemma B.2 (generalization of [32, Lemma 1]). *Let $X_i \sim Gamma(\alpha_i, \alpha)$, $i = 1, 2, \ldots, n$, be independent r.v.'s, where $\alpha_i > 0$ for all i and $\alpha > 0$. Also let $\psi \sim Gamma(1, \alpha)$ be another r.v. independent of all X_i's. If $\sum_{i=1}^n \alpha_i > 1$, then the mode $\bar{x}(\lambda)$ of the r.v. $W(\lambda) = Y + \lambda\psi$ is strictly increasing in $\lambda > 0$, where $Y = \sum_{i=1}^n \lambda_i X_i$ with $\lambda_i > 0$ for all i.*

Proof. The proof is almost identical to that of Lemma 1 in [32]; hence, we omit the details. ■

Lemma B.3 (generalization of [32, Lemma 2]). *For some $\alpha_2 \geq \alpha_1 > 0$, let $\xi_1 \sim Gamma(1+\alpha_1, \alpha_1)$ and $\xi_2 \sim Gamma(1+\alpha_2, \alpha_2)$ be independent gamma r.v.'s. Also let $\bar{x} = \bar{x}(\lambda)$ denote the mode of the r.v. $\xi(\lambda) = \lambda\xi_1 + (1-\lambda)\xi_2$ for $0 \leq \lambda \leq 1$. Then the following hold:*

(i) *For a given λ, $\bar{x}(\lambda)$ is unique.*

(ii) *$1 \leq \bar{x}(\lambda) \leq \frac{2\sqrt{\alpha_1\alpha_2}+1}{2\sqrt{\alpha_1\alpha_2}}$ for all $0 \leq \lambda \leq 1$, with $\bar{x}(0) = \bar{x}(1) = 1$ and, in case of $\alpha_i = \alpha_j = \alpha$, $\bar{x}(\frac{1}{2}) = \frac{2\alpha+1}{2\alpha}$; otherwise the inequalities are strict.*

(iii) *There is a $\lambda^* \in \left(\frac{\sqrt{\alpha_1}}{\sqrt{\alpha_2}+\sqrt{\alpha_1}}, 1\right)$ such that the mode $\bar{x}(\lambda)$ is a strictly increasing function of λ on $(0, \lambda^*)$ and is a strictly decreasing function on $(\lambda^*, 1)$ and, for $\alpha_1 = \alpha_2$, we have $\lambda^* = \frac{1}{2}$.*

Proof. Uniqueness claim (i) has already been proven in [32, Theorem 4]. We prove (iii) since (ii) is implied from within the proof. For $0 < \lambda < 1$, the PDF of $\xi(\lambda)$ can be written as

$$f_{\xi(\lambda)}(x) = \int_0^x f_{\lambda\xi_1}(y) f_{(1-\lambda)\xi_2}(x-y)dy.$$

Since $f_{\lambda\xi_1}(0) = f_{(1-\lambda)\xi_2}(0) = 0$, we have

$$\begin{aligned}\frac{\partial}{\partial x} f_{\xi(\lambda)}(x) &= \int_0^x f_{\lambda\xi_1}(y) \frac{\partial}{\partial x} f_{(1-\lambda)\xi_2}(x-y)dy = -\int_0^x f_{\lambda\xi_1}(y) \frac{\partial}{\partial y} f_{(1-\lambda)\xi_2}(x-y)dy \\ &= \int_0^x \frac{\partial}{\partial y}\left(f_{\lambda\xi_1}(y)\right) f_{(1-\lambda)\xi_2}(x-y)dy,\end{aligned}$$

where for the second equality we use the fact that $\frac{\partial}{\partial x} f(x-y) = -\frac{\partial}{\partial y} f(x-y)$, and for the third equality we use integration by parts. Let $\alpha = \alpha_1$ and $\alpha_2 = c\alpha$ for some $c \geq 1$. So now we have

$$\begin{aligned}\frac{\partial}{\partial x} f_{\xi(\lambda)}(x) &= \frac{(\frac{\alpha}{\lambda})^{1+\alpha}(\frac{c\alpha}{1-\lambda})^{1+\alpha c}}{\Gamma(1+\alpha)\Gamma(1+\alpha c)} \int_0^x \frac{\partial\left(y^\alpha e^{-\frac{\alpha y}{\lambda}}\right)}{\partial y}(x-y)^{\alpha c} e^{-\frac{c\alpha(x-y)}{1-\lambda}} dy \\ &= \frac{\alpha^{2+\alpha}(c\alpha)^{1+c\alpha}}{\Gamma(1+\alpha)\Gamma(1+c\alpha)} \lambda^{-2-\alpha}(1-\lambda)^{-1-\alpha c} e^{-\frac{c\alpha x}{(1-\lambda)}} \int_0^x (\lambda - y) y^{\alpha-1}(x-y)^{\alpha c} e^{-\alpha y\left(\frac{1}{\lambda} - \frac{c}{1-\lambda}\right)} dy \\ &= C(x,\lambda)A(x,\lambda),\end{aligned}$$

where

$$\begin{aligned}C(x,\lambda) &:= \frac{\alpha^{2+\alpha}(c\alpha)^{1+c\alpha}}{\Gamma(1+\alpha)\Gamma(1+c\alpha)} \lambda^{-2-\alpha}(1-\lambda)^{-1-\alpha c} e^{-\frac{c\alpha x}{(1-\lambda)}}, \\ A(x,\lambda) &:= \int_0^x (\lambda - y)\, y^{\alpha-1}(x-y)^{\alpha c} e^{-\phi(\lambda)y} dy, \\ \phi(\lambda) &:= \alpha\left(\frac{1}{\lambda} - \frac{c}{1-\lambda}\right).\end{aligned}$$

Now if $\bar{x}$ is the mode of $\xi(\lambda)$, then we have

$$\frac{\partial}{\partial x} f_{\xi(\lambda)}(\bar{x}) = C(\bar{x},\lambda)A(\bar{x},\lambda) = 0,$$

which implies that $A(\bar{x},\lambda) = 0$ since $C(\bar{x},\lambda) > 0$. Let us define the linear functional $L : \mathcal{G} \to \mathbb{R}$, where $\mathcal{G} = \{g : (0,\bar{x}) \to \mathbb{R} \mid \int_0^{\bar{x}} g(y)y^{\alpha-1} < \infty\}$, as

$$L(g) := \int_0^{\bar{x}} g(y)y^{\alpha-1}(\bar{x}-y)^{\alpha c} e^{-\phi(\lambda)y} dy.$$

We have

$$\begin{aligned}\frac{\partial}{\partial \lambda} A(x,\lambda) &= \int_0^x \left[1 - \phi'(\lambda)y(\lambda - y)\right] y^{\alpha-1}(x-y)^{\alpha c} e^{-\phi(\lambda)y} dy \\ &= \int_0^x \left[1 - \lambda\phi'(\lambda)y + \phi'(\lambda)y^2\right] y^{\alpha-1}(x-y)^{\alpha c} e^{-\phi(\lambda)y} dy,\end{aligned}$$

so

$$\left[\frac{\partial}{\partial\lambda}A(x,\lambda)\right]_{x=\bar{x}} = L\left(1-\lambda\phi'(\lambda)f+\phi'(\lambda)f^2\right), \tag{B.2}$$

where $f \in \mathcal{G}$ is such that $f(y) = y$. On the other hand, since $A(\bar{x},\lambda) = 0$, we get

$$\begin{aligned}
L(\lambda) = L(f) &= \int_0^{\bar{x}} y^\alpha(\bar{x}-y)^{\alpha c}e^{-\phi(\lambda)y}dy = \int_0^{\bar{x}} y^\alpha e^{-\phi(\lambda)y} d\left(-\frac{(\bar{x}-y)^{\alpha c+1}}{\alpha c+1}\right)\\
&= (\alpha c+1)^{-1}\int_0^{\bar{x}} (\bar{x}-y)^{\alpha c+1}\, d\left(y^\alpha e^{-\phi(\lambda)y}\right)\\
&= (\alpha c+1)^{-1}\int_0^{\bar{x}} (\bar{x}-y)\,(\alpha-\phi(\lambda)y)\, y^{\alpha-1}\,(\bar{x}-y)^{\alpha c}\, e^{-\phi(\lambda)y}dy\\
&= (\alpha c+1)^{-1}L\Big(\,(\bar{x}-f)\,(\alpha-\phi(\lambda)f)\,\Big)\\
&= (\alpha c+1)^{-1}L\Big(\alpha\bar{x}-\alpha f-\phi(\lambda)\bar{x}f+\phi(\lambda)f^2\Big),
\end{aligned}$$

where the second integral is Lebesgue–Stieltjes, and the third integral follows from Lebesgue–Stieltjes integration by parts. So, for $\lambda \in (0, \frac{1}{c+1}) \cup (\frac{1}{c+1}, 1)$, we get

$$L(f^2) = \frac{1}{\phi(\lambda)}\left[(\alpha c+1)L(f) - L\Big(\alpha\bar{x}-\alpha f-\phi(\lambda)\bar{x}f\Big)\right] = \frac{1}{\phi(\lambda)}\left[\Big((1+c)\alpha+1-\frac{c\alpha\bar{x}}{1-\lambda}\Big)L(f)\right],$$

where we used the fact that $L(\alpha\bar{x}) = \frac{\alpha\bar{x}}{\lambda}L(\lambda) = \frac{\alpha\bar{x}}{\lambda}L(f)$. Now substituting $L(f^2)$ in (B.2) yields

$$\begin{aligned}
\left[\frac{\partial}{\partial\lambda}A(x,\lambda)\right]_{x=\bar{x}} &= L\left(\frac{1}{\lambda}f-\lambda\phi'(\lambda)f+\phi'(\lambda)f^2\right)\\
&= \left(\frac{1}{\lambda}-\lambda\phi'(\lambda)+\frac{\phi'(\lambda)}{\phi(\lambda)}\left[(1+c)\alpha+1-\frac{c\alpha\bar{x}}{1-\lambda}\right]\right)L(f),
\end{aligned}$$

which after some tedious but routine computations gives

$$\left[\frac{\partial}{\partial\lambda}A(x,\lambda)\right]_{x=\bar{x}} = R(\lambda)\frac{\bar{x}-\Phi(\lambda)}{1-(c+1)\lambda}, \quad \lambda \in \left(0,\frac{1}{1+c}\right)\cup\left(\frac{1}{1+c},1\right),$$

where $R(\lambda) > 0$ for all $0 < \lambda < 1$, and

$$\Phi(\lambda) := \frac{\alpha+(1-2\alpha)\lambda+(\alpha-1+\alpha c)\lambda^2}{\alpha\Big((c+1)\lambda^2-2\lambda+1\Big)}.$$

Since $d\Phi(\lambda)/d\lambda = \big((1-c)\lambda^2-2\lambda+1\big)/\big(\alpha\big((c+1)\lambda^2-2\lambda+1\big)\big)^2$, we have that $d\Phi(\lambda)/d\lambda = 0$ at $\lambda = 1/(1+\sqrt{c})$. Note that the other root, $1/(1-\sqrt{c})$, falls outside of $(0,1)$ for any $c \geq 1$. It readily can be seen that $\Phi(\lambda)$ is increasing on $0 < \lambda < \frac{1}{1+\sqrt{c}}$ and decreasing on $\frac{1}{1+\sqrt{c}} < \lambda < 1$, and so

$$1 \leq \Phi(\lambda) \leq \frac{2\alpha\sqrt{c}+1}{2\alpha\sqrt{c}} \quad \forall 0 \leq \lambda \leq 1.$$

The differentiability of $\bar{x}(\lambda)$ with respect to λ follows from the implicit function theorem:

$$\frac{d\bar{x}(\lambda)}{d\lambda} = -\frac{\frac{\partial}{\partial\lambda}A(\bar{x},\lambda)}{\frac{\partial}{\partial\bar{x}}A(\bar{x},\lambda)},$$

and for that we need to show that $\frac{\partial A(\bar{x},\lambda)}{\partial\bar{x}} \neq 0$ for all $0 < \lambda < 1$. If we assume the contrary for some λ, we get

$$\alpha c A(\bar{x},\lambda) = \alpha c \int_0^{\bar{x}} (\lambda - y) y^{\alpha-1} (\bar{x} - y)^{\alpha c} e^{-\phi(\lambda)y} dy = 0,$$
$$(\bar{x} - \lambda)\frac{\partial}{\partial\bar{x}}A(\bar{x},\lambda) = \alpha c \int_0^{\bar{x}} (\lambda - y)(\bar{x} - \lambda) y^{\alpha-1} (\bar{x} - y)^{\alpha c - 1} e^{-\phi(\lambda)y} dy = 0,$$

which is impossible since the integrand in the first equality is strictly larger than that in the second equality: we can see this by looking at the two cases $0 < y < \lambda$ and $\lambda < y < \bar{x}$. From this we can also note that $\frac{\partial}{\partial\bar{x}}A(\bar{x},\lambda) < 0$ for all $0 < \lambda < 1$. To see this, first consider the case $\bar{x} > \lambda$, and it follows directly as above that $\frac{\partial}{\partial\bar{x}}A(\bar{x},\lambda) < [\alpha c/(\bar{x} - \lambda)]A(\bar{x},\lambda) = 0$. Now assume that $\bar{x} \leq \lambda$, but since the integrand in the first equality is strictly positive for all $0 < y < \bar{x}$, then $A(\bar{x},\lambda) > 0$, which is impossible. So we get

$$\frac{d\bar{x}(\lambda)}{d\lambda} = S(\lambda)\frac{\bar{x} - \Phi(\lambda)}{1 - (c+1)\lambda}, \quad \lambda \in [0,1], \tag{B.3}$$

where $S(\lambda) > 0$ for all $0 < \lambda < 1$. We also defined $\frac{d\bar{x}(\lambda)}{d\lambda}$ for $\lambda = 0, 1, \frac{1}{2}$ using l'Hôpital's rule (with one-sided differentiability for $\lambda = 0, 1$). It is easy to see that

$$\bar{x}(0) = \bar{x}(1) = \Phi(0) = \Phi(1) = 1 \quad \text{and} \quad \bar{x}\left(\frac{1}{c+1}\right) = \Phi\left(\frac{1}{c+1}\right) = \frac{(c+1)\alpha + 1}{(c+1)\alpha}.$$

Next we show that $\bar{x}$ is strictly increasing on $(0, \frac{1}{c+1})$. We first show that on this interval, we must have $\bar{x}(\lambda) \geq \Phi(\lambda)$; otherwise there must exist a $\hat{\lambda} \in (0, \frac{1}{c+1})$ such that $\bar{x}(\hat{\lambda}) < \Phi(\hat{\lambda})$. But this contradicts $\bar{x}(\frac{1}{c+1}) = \Phi(\frac{1}{c+1})$ by (B.3), increasing property of Φ, and continuity of $\bar{x}$. So $\bar{x}$ is nondecreasing on $(0, \frac{1}{c+1})$. We must also have that $\bar{x}(\lambda) > \Phi(\lambda)$ for $\lambda \in (0, \frac{1}{c+1})$; otherwise if there is a $\hat{\lambda} \in (0, \frac{1}{c+1})$ such that $\bar{x}(\hat{\lambda}) = \Phi(\hat{\lambda})$, then, by (B.3), it must be a saddle point of $\bar{x}$. But since Φ is strictly increasing and $\bar{x}$ is nondecreasing on this interval, this would imply that for an ε arbitrarily small, we must have $\bar{x}(\hat{\lambda} + \varepsilon) < \Phi(\hat{\lambda} + \varepsilon)$, but this would contradict the nondecreasing property of $\bar{x}$ on this interval by (B.3). The same reasoning shows that we must have $\bar{x}(\lambda) < \Phi(\lambda)$ on $(\frac{1}{c+1}, \lambda^*)$ (i.e., $\bar{x}$ is strictly increasing on $(\frac{1}{c+1}, \lambda^*)$) and $\bar{x}(\lambda) > \Phi(\lambda)$ on $(\lambda^*, 1)$ (i.e., $\bar{x}$ is strictly decreasing on $(\lambda^*, 1)$). Now we show that $\lambda^* \geq \frac{1}{1+\sqrt{c}}$. For $c = 1$ we have $\frac{1}{c+1} = \frac{1}{\sqrt{c}+1}$, and hence $\lambda^* = \frac{1}{2}$. For $c > 1$, since $\bar{x}(\lambda)$ is increasing for $0 < \lambda < \lambda^*$ and decreasing for $\lambda^* < \lambda < 1$, and $\bar{x}(\lambda^*) = \Phi(\lambda^*)$, then by (B.3), this implies that λ^* is where the maximum of $\bar{x}(\lambda)$ occurs. Now if we assume that $\lambda^* < \frac{1}{1+\sqrt{c}}$, since Φ is increasing on $(0, \frac{1}{1+\sqrt{c}})$, this would contradict $\bar{x}(\lambda) > \Phi(\lambda)$ on $(\lambda^*, 1)$. Lemma B.3 is proved. ■

B.2. Proofs of Theorems 2.1 and 2.2. We now give the detailed proofs for our main theorems stated and used in section 2.

Proof of Theorem 2.1. For proving (i), we first show that $\Delta(x) = 0$ at exactly one point on $\mathbb{R}^+ = \{x|x > 0\}$ denoted by $x(\alpha_1, \alpha_2)$. Since $\alpha_2 > \alpha_1$, let $\alpha_2 = \alpha_1 + c$ for some $c > 0$. We have

$$\frac{d\Delta(x)}{dx} = C(\alpha_2)x^{\alpha_2-1}e^{-\alpha_2 x} - C(\alpha_1)x^{\alpha_1-1}e^{-\alpha_1 x} = C(\alpha_2)x^{\alpha_1-1}e^{-\alpha_1 x}\left(x^c e^{-cx} - \frac{C(\alpha_1)}{C(\alpha_2)}\right),$$

where $C(\alpha) = (\alpha)^\alpha/\Gamma(\alpha)$. The constant $C(\alpha_1)/C(\alpha_2)$ cannot be larger than $x^c e^{-cx}$ for all $x \in \mathbb{R}^+$; otherwise $d\Delta(x)/dx$ would be negative for all $x \in \mathbb{R}^+$, and this is impossible since $\Delta(0) = \Delta(\infty) = 0$. The function $x^c e^{-cx}$ is increasing on $(0, 1)$ and decreasing on $(1, \infty)$, and since $C(\alpha_1)/C(\alpha_2)$ is constant, there must exist an interval (a, b) containing $x = 1$ such that $d\Delta(x)/dx > 0$ for $x \in (a, b)$ and $d\Delta(x)/dx < 0$ for $x \in (0, a) \cup (b, \infty)$. Now since $\Delta(x)$ is continuous and $\Delta(0) = \Delta(\infty) = 0$, then there must exist a unique $x(\alpha_1, \alpha_2) \in (0, \infty)$ such that $\Delta(x)$ crosses zero (i.e., $\Delta(x) = 0$ at the unique point $x(\alpha_1, \alpha_2)$) and that $\Delta(x) < 0$ for $0 < x < x(\alpha_1, \alpha_2)$ and $\Delta(x) > 0$ for $x > x(\alpha_1, \alpha_2)$.

We now prove (ii). The desired inequality is equivalent to $\Delta(x) < 0$ for all $x < 1$ and $\Delta(x) > 0$ for all $x > \left(2\sqrt{\alpha_1(\alpha_2 - \alpha_1)} + 1\right)/\left(2\sqrt{\alpha_1(\alpha_2 - \alpha_1)}\right)$. Without loss of generality consider $\alpha = \alpha_1$, and $\alpha_2 = (1 + c)\alpha$, for $c = (\alpha_2 - \alpha)/\alpha$. Define $\tilde{X} \sim Gamma(c\alpha, c\alpha)$, and let $Y(t) = tX_1 + (1 - t)\tilde{X}$. Note that $Y(1) = X_1$ and $Y(1/(1 + c)) = X_2$, so it suffices to show that the CDF of $Y(t)$ is increasing in $t \in [\frac{1}{1+c}, 1]$ for $x < 1$ and decreasing for $x > (2\alpha\sqrt{c} + 1)/(2\alpha\sqrt{c})$. Now we take the Laplace transform of $Y(t)$ as $\mathcal{L}[Y(t)](z) = \left(1 + \frac{tz}{\alpha}\right)^{-\alpha}\left(1 + \frac{(1-t)z}{c\alpha}\right)^{-c\alpha}$ for $Re(z) > \max\{-\alpha/t, -c\alpha/(1 - t)\}$. The Laplace transform of F_Y is $\mathcal{L}[F_Y](z) = \int_0^\infty e^{-zx}F_Y(x)dx = \frac{1}{z}\int_0^\infty e^{-zx}dF_Y(x) = \frac{1}{z}\mathcal{L}[Y](z)$. Note that in the second equality we applied integration by parts and the fact that $F_Y(0) = 0$. Defining $J(z) := \mathcal{L}[F_Y](z)$ and differentiating with respect to t gives

$$\begin{aligned}\frac{dJ}{dt} &= J\frac{d}{dt}(\ln(J)) = J\frac{d}{dt}\left(-\ln(z) - \alpha\ln\left(1 + \frac{tz}{\alpha}\right) - c\alpha\ln\left(1 + \frac{(1-t)z}{c\alpha}\right)\right)\\ &= \frac{z^2}{c\alpha}J\Big((1 + c)t - 1\Big)\left(1 + \frac{tz}{\alpha}\right)^{-1}\left(1 + \frac{(1-t)z}{c\alpha}\right)^{-1}.\end{aligned}$$

Taking the inverse transform yields

$$\frac{d}{dt}\Pr\left(Y(t) \le x\right) = \frac{(1 + c)t - 1}{c\alpha}\frac{d^2}{dx^2}\Pr\left(Y(t) + t\psi_1 + \frac{1-t}{c}\psi_2 < x\right),$$

where $\psi_i \sim Gamma(1, \alpha)$, $i = 1, 2$, are i.i.d. gamma r.v.'s which are also independent of all X_1 and X_2. Now applying Lemma B.3 yields the desired results. ■

Proof of Theorem 2.2. It is enough to prove the theorem for the special case where $\alpha = \beta$, and the general statement follows from the scaling properties of the gamma r.v.'s.

Introduce the random variable $Y := \sum_{i=1}^n \lambda_i X_i$ with CDF $F_Y(x) = \Pr(Y < x)$. As in the proof of Theorem 2.1, define $J(z) := \mathcal{L}[F_Y](z) = \frac{1}{z}\mathcal{L}[Y](z)$, where $\mathcal{L}[F_Y]$ and $\mathcal{L}[Y]$

denote the Laplace transform of F_Y and Y, respectively, and $\mathcal{L}[Y](z) = \prod_{i=1}^{n}(1+\lambda_i z/\alpha)^{-\alpha}$ for $Re(z) > -\alpha/\lambda_i,\ i = 1, 2, \ldots, n$.

Now consider a vector $\boldsymbol{\lambda} \in \Theta$ for which $\lambda_i\lambda_j \neq 0$ for some $i \neq j$. We keep all $\lambda_k,\ k \neq i, j$, fixed and vary λ_i and λ_j under the condition that $\lambda_i + \lambda_j = \text{const}$. We may assume without loss of generality that $i = 1$ and $j = 2$. Vectors for which $\lambda_i = 1$ for some i, i.e., the "corners" of Θ, are considered at the end of this proof. Differentiating J, we get

$$\frac{dJ}{d\lambda_1} = J\frac{d}{d\lambda_1}(\ln J) = J\frac{d}{d\lambda_1}\left(-\ln(z) - \alpha\sum_{i=1}^{n}\ln\left(1+\frac{\lambda_i z}{\alpha}\right)\right) = J\alpha\frac{z^2}{\alpha^2}\frac{\lambda_1-\lambda_2}{(1+\frac{\lambda_1 z}{\alpha})(1+\frac{\lambda_2 z}{\alpha})}$$

$$= \frac{1}{\alpha}(\lambda_1-\lambda_2)z\mathcal{L}[\lambda_1\psi_1](z)\mathcal{L}[\lambda_2\psi_2](z)\mathcal{L}[Y](z), \tag{B.4}$$

where $\psi_i \sim Gamma(1, \alpha),\ i = 1, 2$, are i.i.d. gamma r.v.'s which are also independent of all X_i's.

Letting $W(\lambda) = Y + \lambda_1\psi_1 + \lambda\psi_2$ with the CDF $F_{W(\lambda)}(x)$, it can be shown that since $\lambda_1\lambda_2 \neq 0$, then by Lemma B.1(iii), $F'_{W(\lambda)}(0) = 0$ for all $\lambda \geq 0$. Defining

$$L(Y, \lambda, x) := F''_{W(\lambda)} = \frac{d^2}{dx^2}\Pr\left(W(\lambda) < x\right) = \frac{d^2}{dx^2}\Pr\Big(Y + \lambda_1\psi_1 + \lambda\psi_2 < x\Big) \tag{B.5}$$

and noting that $\mathcal{L}[W(\lambda)](z) = \mathcal{L}[\lambda_1\psi_1](z)\mathcal{L}[\lambda\psi_2](z)\mathcal{L}[Y](z)$, we get

$$\mathcal{L}\big[L(Y,\lambda,.)\big](z) = \int_0^\infty e^{-zx}L(Y,\lambda,x)dx = \int_0^\infty e^{-zx}F''_{W(\lambda)}(x)dx = z\int_0^\infty e^{-zx}dF_{W(\lambda)}(x)$$
$$= z\mathcal{L}\big[W(\lambda)\big](z) = z\mathcal{L}\big[\lambda_1\psi_1\big](z)\mathcal{L}\big[\lambda\psi_2\big](z)\mathcal{L}\big[Y\big](z).$$

Inverting (B.4) yields

$$\frac{dF_Y(x)}{d\lambda_1} = \frac{1}{\alpha}(\lambda_1-\lambda_2)L(Y,\lambda_2,x). \tag{B.6}$$

So a necessary condition for the extremum of $F_Y(x)$ is either $\lambda_1\lambda_2(\lambda_1-\lambda_2) = 0$ or $L(\lambda_2, x) = 0$. Since $\lambda_1\lambda_2 \neq 0$, then by Lemma B.1, the PDF, $f_{W(\lambda)}(x)$, of the linear form $W(\lambda) = Y + \lambda_1\psi_1 + \lambda\psi_2$, for $\lambda > 0$, is differentiable everywhere and $f_{W(\lambda)}(0) = 0$. In addition, on the positive half-line, $f'_{W(\lambda)}(x) = 0$ holds at a unique point because $f_{W(\lambda)}(x)$ is a unimodal analytic function (its graph contains no line segment). The unimodality of $f_{W(\lambda)}(x)$ was already proven for all gamma random variables in [32, Thm. 4].

Now we can prove that, for any $x > 0$, if $F_Y(x)$ has an extremum, then the nonzero λ_i's can take at most two different values. Suppose that $\lambda_1\lambda_2(\lambda_1-\lambda_2) \neq 0$; then by (B.6) we have $L(Y, \lambda_2, x) = 0$. Now we show that, for every $\lambda_j \neq 0$, (B.6) implies that $\lambda_i = \lambda_1$ or $\lambda_i = \lambda_2$. For this, we assume the contrary, that $\lambda_i \neq \lambda_1$, $\lambda_i \neq \lambda_2$, and by using the same reasoning that led to (B.6), we can show that

$$L(Y, \lambda_2, x) = L(Y, \lambda_j, x) = 0$$

for every $\lambda_j \neq 0$, i.e., the point $x > 0$ is simultaneously the mode of the PDF of $W_Y^{\lambda_2}$ and $W_Y^{\lambda_j}$, which contradicts Lemma B.2. So we get that $\lambda_i = \lambda_1$ or $\lambda_2 = \lambda_j$. Thus the

extrema of $F_Y(x)$ are taken for some $\lambda_1 = \lambda_2 = \cdots = \lambda_k$, $\lambda_{k+1} = \lambda_{k+2} = \cdots = \lambda_{k+m}$, and $\lambda_{k+m+1} = \lambda_{k+m+2} = \cdots = \lambda_n = 0$, where $k + m \leq n$, i.e.,

$$\text{extremum } \Pr\left(\sum_{i=1}^{n} \lambda_i X_i \leq x\right) = \text{extremum } \Pr\left(\frac{\lambda}{k}\sum_{i=1}^{k} X_i + \frac{1-\lambda}{m}\sum_{i=k+1}^{k+m} X_i \leq x\right).$$

Here without loss of generality we can assume $k \geq m \geq 1$. Now the same reasoning as in the end of the proof of [32, Theorem 1] shows an extremum is taken either at $k = m = 1$ or at $\lambda_1 = \lambda_2 = \cdots = \lambda_{k+m}$. In the former case, by Lemma B.3, for any $x \in (0,1) \cup (\frac{2\alpha+1}{2\alpha}, \infty)$, the extremum can only be taken at $\lambda \in \{0, \frac{1}{2}, 1\}$. However, for any $x \in [1, \frac{2\alpha+1}{2\alpha}]$, in addition to $\lambda \in \{0, \frac{1}{2}, 1\}$, the extremum can be achieved for some λ^* such that $x = \bar{x}(\lambda^*)$, where $\bar{x}(\lambda)$ denotes the mode of the distribution of $\lambda X_1 + (1-\lambda)X_2 + \lambda\psi_1 + (1-\lambda)\psi_2$. But for such λ^* and x, using (B.6) and Lemma B.3(iii) with $\alpha_1 = \alpha_2 = \alpha$, one can show that $\Pr(\lambda X_1 + (1-\lambda)X_2 \leq x)$ achieves a local maximum. Now including the case where $\lambda_1 = 1$ mentioned earlier in the proof, we get

$$m_n(x) = \min_{1\leq d\leq n} \Pr\left(\frac{1}{d}\sum_{i=1}^{d} X_i < x\right) \quad \forall x > 0,$$

$$M_n(x) = \max_{1\leq d\leq n} \Pr\left(\frac{1}{d}\sum_{i=1}^{d} X_i < x\right) \quad \forall x \in \Big(0,1\Big) \cup \left(\frac{2\alpha+1}{2\alpha}, \infty\right),$$

where $m_n(x)$ and $M_n(x)$ are defined in the statement of Theorem 2.2 in section 2. Now applying Theorem 2.1 by considering the collection $\alpha_i = i\alpha$, $i = 1, 2, \ldots, n$, would yield the desired results. ■

Appendix C. MATLAB code. Here we provide a short MATLAB code, promised in section 2, to calculate the necessary or sufficient sample sizes to satisfy the probabilistic accuracy guarantees (2.2) for an SPSD matrix using the Gaussian trace estimator. This code can be easily modified to be used for (2.10) as well.

```
function [N1,N2] = getSampleSizes(epsilon,delta,maxN,r)
% INPUT:
% @ epsilon:  Accuracy of the estimation.
% @ delta:  Uncertainty of the estimation.
% @ r: Rank of the matrix (Use r = 1 for obtaining the sufficient sample sizes).
% @ maxN: Maximum allowable sample size
% OUTPUT:
% @ N1: The sufficient (or necessary) sample size for (2.2a).
% @ N2: The sufficient (or necessary) sample size for (2.2b).
Ns = 1:1:maxN;
P1 = gammainc(Ns*r*(1-epsilon)/2,Ns*r/2);
I1 = find(P1 <= delta,1,'first');
N1 = Ns(I1); % Necessary/Sufficient sample size obtained for (2.2a).
Ns = (floor(1/epsilon)+1):1:maxN;
P2 = gammainc(Ns*r*(1+epsilon)/2,Ns*r/2);
I2 = find(P2 >= 1-delta,1,'first');
N2 = Ns(I2); % Necessary/Sufficient sample size obtained for (2.2b).
end
```

Acknowledgments. We thank our anonymous referees for several valuable comments which have helped to improve the text. The first author thanks Prof. Yaming Yu for referring him to [32], which resulted in the collaboration of the authors of this paper.

REFERENCES

[1] D. Achlioptas, *Database-friendly random projections*, in Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 01), 2001, pp. 274–281.

[2] S. R. Arridge, *Optical tomography in medical imaging*, Inverse Problems, 15 (1999), pp. R41–R93.

[3] H. Avron and S. Toledo, *Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix*, J. ACM, 58 (2011), 8.

[4] D. A. Boas, D. H. Brooks, E. L. Miller, C. A. DiMarzio, M. Kilmer, R. J. Gaudette, and Q. Zhang, *Imaging the body with diffuse optical tomography*, IEEE Signal Process. Mag., 18 (2001), pp. 57–75.

[5] L. Borcea, J. G. Berryman, and G. C. Papanicolaou, *High-contrast impedance tomography*, Inverse Problems, 12 (1996), pp. 835–858.

[6] A. Borsic, B. M. Graham, A. Adler, and W. R. B. Lionheart, *Total Variation Regularization in Electrical Impedance Tomography*, MIMS EPrint 2007.92, The University of Manchester, Manchester, UK, 2007.

[7] T. Chan and X. Tai, *Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients*, J. Comput. Phys., 193 (2003), pp. 40–66.

[8] M. Cheney, D. Isaacson, and J. C. Newell, *Electrical impedance tomography*, SIAM Rev., 41 (1999), pp. 85–101.

[9] K. van den Doel and U. Ascher, *Dynamic level set regularization for large distributed parameter estimation problems*, Inverse Problems, 23 (2007), pp. 1271–1288.

[10] K. van den Doel and U. M. Ascher, *Adaptive and stochastic algorithms for electrical impedance tomography and DC resistivity problems with piecewise constant solutions and many measurements*, SIAM J. Sci. Comput., 34 (2012), pp. A185–A205.

[11] K. van den Doel, U. Ascher, and E. Haber, *The lost honour of ℓ_2-based regularization*, in Large Scale Inverse Problems, Radon Ser. Comput. Appl. Math. 13, M. Cullen, M. Freitag, S. Kindermann, and R. Scheichl, eds., de Gruyter, Berlin, 2013, pp. 181–203.

[12] O. Dorn, E. L. Miller, and C. M. Rappaport, *A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets*, Inverse Problems, 16 (2000), pp. 1119–1156.

[13] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[14] A. Fichtner, *Full Seismic Waveform Modeling and Inversion*, Springer, Berlin, Heidelberg, 2011.

[15] H. Gao, S. Osher, and H. Zhao, *Quantitative photoacoustic tomography*, in Mathematical Modeling in Biomedical Imaging II, Springer, Heidelberg, 2012, pp. 131–158.

[16] E. Haber, U. Ascher, and D. Oldenburg, *Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach*, Geophysics, 69 (2004), pp. 1216–1228.

[17] E. Haber and M. Chung, *Simultaneous Source for Non-uniform Data Variance and Missing Data*, preprint, arXiv:1404.5254, 2014.

[18] E. Haber, M. Chung, and F. Herrmann, *An effective method for parameter estimation with PDE constraints with multiple right-hand sides*, SIAM J. Optim., 22 (2012), pp. 739–757.

[19] E. Haber, S. Heldmann, and U. Ascher, *Adaptive finite volume method for distributed non-smooth parameter identification*, Inverse Problems, 23 (2007), pp. 1659–1676.

[20] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.

[21] F. Herrmann, Y. Erlangga, and T. Lin, *Compressive simultaneous full-waveform simulation*, Geophysics, 74 (2009), pp. A35–A40.

[22] J. Kaipo and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, New York, 2005.

[23] V. A. Morozov, *Methods for Solving Incorrectly Posed Problems*, Springer, New York, 1984.
[24] G. A. Newman and D. L. Alumbaugh, *Frequency-domain modelling of airborne electromagnetic responses using staggered finite differences*, Geophys. Prospecting, 43 (1995), pp. 1021–1042.
[25] D. W. Oldenburg, E. Haber, and R. Shekhtman, *Three dimensional inversion of multisource time domain electromagnetic data*, Geophysics, 78 (2013), pp. E47–E57.
[26] A. Pidlisecky, E. Haber, and R. Knight, *RESINVM3D: A MATLAB 3D resistivity inversion package*, Geophysics, 72 (2007), pp. H1–H10.
[27] J. Rohmberg, R. Neelamani, C. Krohn, J. Krebs, M. Deffenbaugh, and J. Anderson, *Efficient seismic forward modeling and acquisition using simultaneous random sources and sparsity*, Geophysics, 75 (2010), pp. WB15–WB27.
[28] F. Roosta-Khorasani and U. Ascher, *Improved bounds on sample size for implicit matrix trace estimators*, Found. Comput. Math., 2014, DOI: 10.1007/s10208-014-9220-1.
[29] F. Roosta-Khorasani, K. van den Doel, and U. Ascher, *Data completion and stochastic algorithms for PDE inversion problems with many measurements*, Electron. Trans. Numer. Anal., 42 (2014), pp. 177–196.
[30] F. Roosta-Khorasani, K. van den Doel, and U. Ascher, *Stochastic algorithms for inverse problems involving PDEs and many measurements*, SIAM J. Sci. Comput., 36 (2014), pp. S3–S22.
[31] N. C. Smith and K. Vozoff, *Two-dimensional DC resistivity inversion for dipole-dipole data*, IEEE Trans. Geosci. Rem. Sens., 22 (1984), pp. 21–28.
[32] G. J. Székely and N. K. Bakirov, *Extremal probabilities for Gaussian quadratic forms*, Probab. Theory Related Fields, 126 (2003), pp. 184–202.
[33] C. R. Vogel, *Computational Methods for Inverse Problem*, SIAM, Philadelphia, 2002.
[34] J. Young and D. Ridzal, *An application of random projection to parameter estimation in partial differential equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2344–A2365.
[35] Z. Yuan and H. Jiang, *Quantitative photoacoustic tomography: Recovery of optical absorption coefficient maps of heterogeneous media*, Appl. Phys. Lett., 88 (2006), 231101.