Convergent Incremental Optimization Transfer Algorithms: Application to Tomography

Sangtae Ahn*, Member, IEEE, Jeffrey A. Fessler, Senior Member, IEEE, Doron Blatt, Student Member, IEEE, and Alfred O. Hero, Fellow, IEEE

Abstract—No convergent ordered subsets (OS) type image reconstruction algorithms for transmission tomography have been proposed to date. In contrast, in emission tomography, there are two known families of convergent OS algorithms: methods that use relaxation parameters [1], and methods based on the incremental expectation-maximization (EM) approach [2]. This paper generalizes the incremental EM approach [3] by introducing a general framework, “incremental optimization transfer.” The proposed algorithms accelerate convergence speeds and ensure global convergence without requiring relaxation parameters. The general optimization transfer framework allows the use of a very broad family of surrogate functions, enabling the development of new algorithms [4]. This paper provides the first convergent OS-type algorithm for (nonconcave) penalized-likelihood (PL) transmission image reconstruction by using separable paraboloidal surrogates (SPS) [5] which yield closed-form maximization steps. We found it desirable to achieve fast initial convergence rates by starting with an OS algorithm with a large number of subsets and switching to the new “transmission incremental optimization transfer” algorithm. Results show that TRIOT is faster in increasing the PL convergence without requiring relaxation parameters. The general optimization transfer framework allows the use of a very broad family of surrogate functions, enabling the development of new algorithms [4]. This paper provides the first convergent OS-type algorithm for (nonconcave) penalized-likelihood (PL) transmission image reconstruction by using separable paraboloidal surrogates (SPS) [5] which yield closed-form maximization steps. We found it is very effective to achieve fast convergence rates by starting with an OS algorithm with a large number of subsets and switching to the new “transmission incremental optimization transfer” algorithm. Results show that TRIOT is faster in increasing the PL objective than nonincremental ordinary SPS and even OS-SPS yet is convergent.

Index Terms—Incremental optimization transfer, maximum-likelihood estimation, penalized-likelihood estimation, statistical image reconstruction, transmission tomography.

I. INTRODUCTION

ORDERED SUBSETS (OS) algorithms, also known as block iterative or incremental gradient methods, have been very popular in the medical imaging community for tomographic image reconstruction due to their remarkably fast “convergence” rates [1], [5]–[16]. For example, ordered subsets expectation-maximization (OS-EM) provides an order-of-magnitude acceleration over its non-OS counterpart, EM in the context of emission tomography [6]; and OS convex (OSC) algorithms achieve extremely high acceleration factors in X-ray CT imaging where the number of projection views is very large [13], [16]. The incremental gradient type algorithms are also found in convex programming [17]–[20]. The ordered subsets (or incremental) idea is to perform the update iteration incrementally by sequentially (or sometimes randomly [18], [19]) using a subset of the data. Row-action methods [21] including algebraic reconstruction techniques (ART) [22], [23] can also be viewed as OS type algorithms in which each subset corresponds to a single measurement.

The OS algorithms apply successfully to problems where an objective function of interest is a sum of a large number of component functions. Because of the assumed statistical independence of tomographic data, such sums arise in statistical tomographic reconstruction problems including penalized-likelihood (PL) [equivalently, maximum a posteriori (MAP)] or maximum-likelihood (ML) reconstruction, Typically, the OS methods partition the component functions into several subsets, each corresponding to a subset of the projection views, and each subset defines a subobjective function.

Roughly speaking, if the subset gradients are suitably balanced, then the gradient approximation can be quite reasonable when the iterates are far from a maximizer. Thus, OS methods initially accelerate convergence in the sense that less computation is required to achieve nearly the same level of objective increase. However, ordinary (unrelaxed) OS algorithms such as OS-EM [6], RBI-EM [8], and OS-SPS (or OSTR in a context of transmission tomography) [5] generally do not converge to an optimal solution but rather approach a suboptimal limit cycle that consists of as many points as there are subsets. (An “optimal” solution means a maximizer of the ML or PL objective function throughout this paper.) In fact, due to their subset-dependent scaling (or preconditioning) matrices [1], OS-EM and RBI-EM in their original forms [6], [8] usually do not converge to the optimal point even if relaxed.

Convergence to an optimal solution is important for any algorithm for optimization problems, particularly in medical applications where reliability and stability are essential. It is more critical for PL (or MAP) reconstruction than for ML because one usually does not run ML reconstruction algorithms until convergence; therefore, for popular OS-EM in emission imaging, convergence may not be a practical issue. However, for PL reconstruction, the image shown in Fig. 4(e), which corresponds to one point of a limit cycle generated by an OS algorithm, looks different from the PL solution image shown in Fig. 4(a) (see Section IV-A for details). Nonconvergent OS algorithms can lead to higher variance as the number of subsets increases as shown in Section IV-B (see also [13], [16]). It is desirable to achieve both fast initial convergence rates (typical of OS algorithms) and global convergence. There have been
three known families of convergent incremental (or OS type) algorithms: methods that use relaxation parameters, methods based on the incremental EM approach, and incremental aggregated gradient (IAG) methods.

Relaxation parameters are used widely to render OS algorithms convergent [1], [7], [10]–[12], [14], [17]–[19], [24]–[26]. Suitably relaxed algorithms can be shown to converge to an optimal solution under certain regularity conditions1 [1]. However, since relaxation parameters should be scheduled to converge to zero to ensure global convergence, relaxed OS algorithms have slow asymptotic convergence rates. Finding good relaxation parameters (in terms of convergence rates) may require some experimentation and trial-and-error; as a rule of thumb, for properly scaled OS algorithms such as modified \textbf{<Au: Define acronym BSREM?>} (BSREM) and relaxed OS-SPS, one should initialize the relaxation parameter close to unity and decrease it gradually as convergence to a limit cycle nears [1]. One may optimize a few initial relaxation parameters by training when a training set is available for a particular task [7], [23]. One could use the dynamic stepsize rule in [18], [19], but the method needs to compute the objective value at every update, which is computationally expensive in tomographic reconstruction problems. Alternatively, to achieve convergence, one could decrease the number of subsets as iterations proceed or could use hybrid methods that combine OS and non-OS algorithms [28]. However, the schedules for decreasing the number of subsets and the parameters for the hybrid algorithms are as inconvenient to determine as relaxation parameters for relaxed OS algorithms.

Incremental EM algorithms do not require user-specified relaxation parameters [3]. They are convergent yet faster than ordinary EM algorithms although slower initially than nonconvergent OS-EM type algorithms [2], [29], [30]. Such incremental EM algorithms have been applied to emission tomography [2], [15], [30], [31].

Recently, Blatt et al. proposed a convergent incremental gradient method, called incremental aggregated gradient (IAG), that does not require relaxation parameters [32]. The IAG method computes a single subset gradient for each update but aggregates it with the stored subset gradients that were computed in previous iterations. The use of the aggregated gradient to approximate the full gradient of the objective function leads to convergence.

In this paper, we generalize the incremental EM algorithms by introducing an approach called “incremental optimization transfer”; this is akin to the generalization of the EM algorithms [33] by the optimization transfer principles [4]. In fact, the broad family of “incremental optimization transfer algorithms” includes the ordinary optimization transfer algorithms (e.g., EM), also referred to as MM (minimize-maximize or majorize-minimize) algorithms in [34], as a special case where the objective function consists of only one subobjective function. Incremental optimization transfer algorithms show faster convergence rates than their nonincremental counterparts like EM [2], [3], [30].

Incremental optimization transfer is a general framework in which one can develop many different algorithms by using a very broad family of application-dependent surrogate functions. These methods are particularly useful for large-scale problems where the objective function is expressed as a sum of several subobjective functions. In this paper, we focus on the PL image reconstruction for transmission tomography, which is a challenging nonconcave maximization problem. We propose a particular incremental optimization transfer algorithm that uses separable paraboloidal surrogates (SPS) [5]. Such quadratic surrogates simplify the maximization. In contrast, the standard EM surrogates for transmission tomography do not have a closed-form maximizer in the “M-step” [35].

The proposed “transmission incremental optimization transfer (TRIOT)” algorithm is convergent yet converges faster than ordinary SPS [5]; it can be further accelerated by the enhancement method in [36] or by initializing through a few iterations of OS-SPS (see Section III for details). It is parallelizable, and the nonnegativity constraint is naturally enforced. In addition, it is easily implemented for systems models that use factored system matrices [37], [38] whereas pixel-grouped coordinate ascent based methods require column access of the system matrix [39]–[42].

Section II describes the incremental optimization transfer algorithms in a general framework and discusses their convergence properties. Section III develops incremental optimization transfer algorithms for transmission tomography, and addresses acceleration methods. Section IV provides simulation and real PET data results, and Section V gives conclusions.

II. INCREMENTAL OPTIMIZATION TRANSFER

A. Incremental Optimization Transfer Algorithms

Most objective functions of interest in image reconstruction can be expressed as a sum of subobjective functions:2

$$\Phi(x) = \sum_{m=1}^{M} \Phi_m(x)$$  \hspace{1cm} (1)

where \(\Phi_m : \mathcal{X} \subset \mathbb{R}^p \rightarrow \mathbb{R}\) is a continuously differentiable function whose domain \(\mathcal{X}\) is a nonempty, convex and closed set. The \(p \times 1\) vector \(x\) represents an image, e.g., attenuation coefficients or radioactivity, depending on applications. Consider the following optimization problem:

$$\text{maximize } \Phi(x) \text{ subject to } x \in \mathcal{X}. \hspace{1cm} (2)$$

Usually, there exists no closed-form solution to the above problem, so one must apply iterative algorithms. Assume that for each subobjective function \(\Phi_m\), we find a surrogate function \(\phi_m(x)\) which satisfies the following “minimization” conditions [27], [44]:

$$\phi_m(x, z) = \Phi_m(x), \quad \forall x \in \mathcal{X}$$

$$\phi_m(x, z) \leq \Phi_m(x), \quad \forall x, z \in \mathcal{X} \hspace{1cm} (3)$$

Such functions are said to be additive-separable in [17]; and to be partially separable [43] when each \(\Phi_m(x)\) is a function of fewer components of \(x \in \mathbb{R}^p\) than \(p\).
where $\tilde{x}$ is an augmented variable that is called an expansion point in [42]. It follows from the above conditions that

$$
\Phi_m(x) - \Phi_m(\tilde{x}) \geq \phi_m(x; \tilde{x}) - \phi_m(\tilde{x}; \tilde{x}), \quad \forall x, \tilde{x} \in \mathcal{X}. \quad (4)
$$

In other words, choosing $x$ such that $\phi_m(x; \tilde{x}) \geq \psi_m(\tilde{x}; \tilde{x})$ ensures that $\Phi_m(x) \geq \Phi_m(\tilde{x})$, which is called a “monotonicity” property. When there is only one subobjective function ($M = 1$, $\Phi = \phi_1$, and $\phi = \phi_1$), one can construct the following “optimization transfer” algorithm:

$$
x^{n+1} = \arg \max_{x \in \mathcal{X}} \phi(x; x^n).
$$

Then, we have $\Phi(x^{n+1}) \geq \Phi(x^n)$ due to the monotonicity property. The reader who would like more details on optimization transfer principles is referred to [4].

Now we construct an incremental version of the optimization transfer algorithm. First, define the following “augmented” objective function:

$$
F(x; \tilde{x}_1, \ldots, \tilde{x}_M) = \sum_{m=1}^{M} \phi_m(x; \tilde{x}_m), \quad (5)
$$

Noting that $\arg \max_{x \in \mathcal{X}} \phi_m(x; \tilde{x}) = \Phi_m(x)$ due to (3), one can rewrite the optimization problem in (2) as follows:

maximize $F(x; \tilde{x}_1, \ldots, \tilde{x}_M)$

subject to $(x; \tilde{x}_1, \ldots, \tilde{x}_M) \in \mathcal{X}^{M+1}$. \quad (6)

By construction, $x^* \in \mathcal{X}$ is an optimal solution of (2) if and only if $(x^*; \tilde{x}_1^*, \ldots, \tilde{x}_M^*) \in \mathcal{X}^{M+1}$ is an optimal solution of (6) for some $(\tilde{x}_1, \ldots, \tilde{x}_M) \in \mathcal{X}^M$. Therefore, we can find a solution to problem (2) by maximizing $F$ with respect to $(x; \tilde{x}_1, \ldots, \tilde{x}_M)$. By alternating between updating $x$ and one of the $\tilde{x}_m$’s, we obtain an “incremental optimization transfer algorithm” outlined in Table I.

In many applications where (T-1) has a closed-form solution, the computational cost for computing $x^{\text{new}}$ is smaller, usually by a factor of $M$, when only one of $\tilde{x}_m$’s has been incrementally updated than when all $\tilde{x}_m$’s have been simultaneously updated. For example, in tomographic reconstruction problems we focus on, computing $x^{\text{new}}$ usually involves forward projections of $\tilde{x}_m$’s, and we have only to project one of $\tilde{x}_m$’s and use previously computed projections for other $\tilde{x}_m$’s.

The incremental optimization transfer algorithm shown in Table I can be viewed as a block coordinate ascent algorithm for maximizing $F$ with respect to $(x; \tilde{x}_1, \ldots, \tilde{x}_M)$ [45, p. 270]. It monotonically increases the augmented objective function $F$, but not necessarily the original objective function $\Phi$ [46]. The incremental approach ($M > 1$) usually leads to faster convergence rates than nonincremental methods ($M = 1$) [3]. The incremental EM algorithms [3], [31] including COSEM [2], [30] are a special case where the surrogates $\phi_m$ are constructed by EM principles as described in Appendix A.

If one were to maximize only one of the $\phi_m$’s instead of the sum shown in (5), then one would have ordinary OS type algorithms. Although this greedy approach usually yields fast initial convergence rates, the OS type algorithms are not monotonic in $F$ nor in $\Phi$. Since OS algorithms consider only a part of the objective function for each update and an optimal point is characterized by the whole objective function, OS algorithms usually cannot converge to the optimal point. In contrast, the incremental optimization transfer methods involve the augmented objective function that reflects the whole objective function yet is updated incrementally.

Although we focus on monoenergetic transmission tomography in Section III, the incremental optimization transfer is a general method which can be applied to a variety of problems where an objective function is a sum of functions as in (1) and the OS approach applies: for example, polychromatic transmission tomography [47], confocal microscopy [48], and emission tomography [49].

For incremental optimization transfer algorithms one must store $M$ vectors $(\tilde{x}_m)_{m=1}^{M}$, so one needs more memory compared to ordinary OS algorithms. This can be a practical limitation when $M$ is very large for large-sized problems.

### B. Convergence Properties

Since incremental optimization transfer algorithms monotonically increase the augmented objective $F$, the sequence of augmented objective values converges to some value in the usual case where $F$ has an upper bound. The question of whether the algorithms really converge to a maximizer of (2) is addressed next.

Define a solution set as the collection of stationary points of (2)

$$
\Gamma = \{ x^* \in \mathcal{X} : \nabla \Phi(x^*)(x - x^*) \leq 0, \quad \forall x \in \mathcal{X} \} \quad (7)
$$

<table>
<thead>
<tr>
<th>Table I: Outline for Incremental Optimization Transfer Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize $x^0, \tilde{x}_1^0, \ldots, \tilde{x}_M^0 \in \mathcal{X}$</td>
</tr>
<tr>
<td>for $n = 0, \ldots, n_{\text{iter}} - 1$</td>
</tr>
<tr>
<td>for $m = 1, \ldots, M$</td>
</tr>
<tr>
<td>$x^{\text{new}} = \arg \max_{x \in \mathcal{X}} \left( x; \tilde{x}_1^{n+1}, \ldots, \tilde{x}_m^{n+1}, \tilde{x}_m^n, \ldots, \tilde{x}_M^n \right)$ \quad (T-1)</td>
</tr>
<tr>
<td>$\tilde{x}<em>m^{n+1} = x^{\text{new}} = \arg \max</em>{x \in \mathcal{X}} \left( x; \tilde{x}_1^{n+1}, \ldots, \tilde{x}_m^{n+1}, \tilde{x}_m^n, \ldots, \tilde{x}_M^n \right)$ \quad (T-2)</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>$x^{n+1} = x_M^{n+1}$ \quad (T-3)</td>
</tr>
</tbody>
</table>
where \( \Gamma \) denotes matrix or vector transpose, and we assume \( \Gamma \neq 0 \). Each element of the solution set \( \Gamma \) satisfies the first-order necessary condition for a local maximizer of \( \Phi \) over \( \mathcal{X} \) [45, p. 194]. We want algorithms to converge to some point in \( \Gamma \). If the objective function \( \Phi \) is concave, then the condition defining \( \Gamma \) is sufficient for optimality, and \( \Gamma \) is the set of (possibly multiple) global maximizers of \( \Phi \) over \( \mathcal{X} \) [45, p. 194]. If \( \Phi \) is strictly concave, then \( \Gamma \) is the singleton set whose element is a unique global maximizer [45, p. 685]. On the other hand, for a nonconcave objective function \( \Phi \) (as in Section III), the solution set \( \Gamma \) could contain local maximizers and even local minimizers. It is difficult to guarantee finding a global maximizer of a nonconcave objective function that may have multiple local maxima. However, the hope is that, with an initial point reasonably close to a global maximizer, the iterates generated by a monotonic algorithm will approach the global maximizer (see [42] for discussion about convergence to a globally optimal point).

In Appendix B, we show that every limit point \( \Gamma \) of the sequence generated by an incremental optimization transfer algorithm is an element of the solution set \( \Gamma \) of stationary points regardless of initial points \( \Gamma \) when the following general sufficient conditions hold: 1) each \( \Phi_m \) is differentiable, and each \( \phi_m(x) \) is differentiable with respect to the first argument and is continuous with respect to the second argument; 2) the iterates are bounded, e.g., \( \mathcal{X} \) is a bounded set; 3) the surrogates \( \phi_m \) satisfy the minorization conditions in (3); 4) the gradients of \( \Phi_m \) and \( \phi_m(x) \) with respect to the first argument match; 5) the maximizer in (T-1) is unique, e.g., \( \phi_m(x) \) is strictly concave for any \( \mathcal{X}_m \); 6) there exists a unique maximizer in (T-2); 7) the maximizer of the augmented objective function is bounded. Consequently, if the objective function \( \Phi \) is strictly concave, then the algorithm converges to the global maximizer. For a nonconcave objective function \( \Phi \), if the points in \( \Gamma \) are isolated, the algorithm will still converge to some stationary point in \( \Gamma \) that we hope is a global maximizer or at least a local maximizer (see Appendix B). It is an open question whether optimization transfer algorithms converge to nonisolated stationary points (see [42] for a discussion of this issue).

Although we focus on transmission tomography in the next section, we briefly discuss the applicability of our convergence proofs to the emission case. The convergence proofs in Appendix B do not apply to classical ML-EM and COSEM for the emission case in their original forms in [35], [52] and [2], [30] respectively since the EM surrogates used in those algorithms blow up to (negative) infinity on the boundary of the nonnegativity constraint set and, therefore, they violate the aforementioned sufficient conditions. The readers are referred to [53] and [54] for convergence proofs for ML-EM and COSEM respectively for the emission case. However, to avoid the boundary problem one can use a slightly modified ML surrogate in [41, Eq. (20)] for the usual case where there are nonzero contributions from background events such as scatter and randoms. Our convergence proofs apply to those modified ML-EM (called “ML-EM-3” in [41]) and COSEM algorithms. Moreover, the modified EM surrogate is known to accelerate convergence rates [41].

See [55, Appendix F] for an asymptotic local convergence rate analysis and an illustrative one-parameter example for a comparison of the convergence rates of incremental and non-incremental algorithms.

### III. APPLICATION TO TRANSMISSION TOMOGRAPHY

In this section we develop a particular incremental optimization transfer algorithm for transmission tomographic reconstruction. We use quadratic surrogates [5], [27] rather than EM surrogates in (26) in Appendix A because the standard complete-data proposed in [35] for transmission tomography does not yield a closed-form solution for M-step [56]. Using quadratic surrogates is not limited to the transmission case [57]–[59].

#### A. Problem

We assume the following Poisson statistical model for (monoenergetic) transmission measurements

\[
y_i \sim \text{Poisson} \left\{ b_i e^{-|\mathbf{A} x_i|} + r_i \right\}, \quad i = 1, \ldots, N
\]

where \( y_i \) denotes the transmission measurement of the \( i \)th detector, \( b_i \) denotes the blank scan counts of the \( i \)th detector, \( r_i \) denotes the mean number of background counts, and \( |\mathbf{A} x_i| = \sum_{j=1}^{n} a_{ij} x_j \) represents the \( i \)th line integral of the attenuation map in which \( x_j \) is the unknown attenuation coefficient in the \( j \)th pixel, \( \mathbf{A} = \{a_{ij}\} \) is the system matrix, and \( N \) and \( p \) are the number of detectors and pixels, respectively. We assume that \( \{b_i\}, \{a_{ij}\} \), and \( \{r_i\} \) are known nonnegative constants. We focus on PL, also known as MAP, estimation for the attenuation map reconstruction. Our goal is to compute a PL estimate \( \hat{\mathbf{x}}_{\text{PL}} \) which is defined by

\[
\hat{\mathbf{x}}_{\text{PL}} = \arg \max_{\mathbf{x} \in \mathbb{R}^N} \Phi(\mathbf{x}), \quad \Phi(\mathbf{x}) = I(\mathbf{x}) - \beta R(\mathbf{x})
\]

where the objective function \( \Phi \), which can be nonconcave when \( r_i \neq 0 \) [27], includes the log-likelihood

\[
I(\mathbf{x}) = \sum_{i=1}^{N} b_i e^{-|\mathbf{A} x_i|}
\]

and a roughness penalty

\[
R(\mathbf{x}) = \frac{1}{2} \sum_{j \neq k \in N} \omega_{jk} \psi(|x_j - x_k|).
\]

The box constraint set is defined by

\[
\mathcal{X} = \{ \mathbf{x} \in \mathbb{R}^N : 0 \leq x_j \leq U, \quad \forall j \}.
\]
The nonnegativity restriction is imposed on physical grounds, and the upper bound $U > 0$ is set by the user to be a value that is larger than the maximum attenuation coefficient conceivable for the object being scanned. The reason for using the box constraint rather than the usual nonnegativity constraint is that the convergence proofs in Appendix B need the iterates to be bounded. However, imposing upper bounds is not overly restrictive in a sense that one can choose a physically meaningful upper bound for attenuation coefficients, and the image estimate $\hat{\phi}$ is unlikely to be affected by $U$ if one chooses an arbitrarily large $U$. In practice, if the upper bound happens to be active by some iterate, then the user could re-run the algorithm with a larger bound.

In the penalty function (10), the function $\psi$ is a symmetric and convex potential function, $\mathcal{N}_j$ represents a neighborhood of the $j$th pixel, $\beta$ is a regularization parameter that controls the smoothness in reconstructed images, and $w_{jk}$ are weights (ordinarily, $w_{jk} = 1$ for horizontal and vertical neighboring pixels, and $w_{jk} = 1/\sqrt{2}$ for diagonal neighboring pixels). We assume the potential function $\psi$ satisfies some conditions given in [27], [60, p. 184]. We used the following edge-preserving nonquadratic potential function in our PL reconstruction results [61]

$$\psi(t) = \delta^2 \left[ \frac{t}{\delta} - \log \left( 1 + \frac{t}{\delta} \right) \right]$$

for some $\delta > 0$. We assume that appropriate $\beta$ and $\delta$ are pre-specified. To design an optimal regularization function is application-dependent and is beyond the scope of this paper. See [62]–[67] for quadratic penalty design.

**B. Transmission Incremental Optimization Transfer (TRIOT)**

We decompose the objective function $\Phi$ into the following subobjective functions:

$$\Phi_m(\mathbf{z}) = \sum_{i \in S_m} h_i(\mathbf{A}\mathbf{z})_i - \frac{\beta}{M} R(\mathbf{z}), \quad m = 1, \ldots, M$$

where $\{ S_m \}_{m=1}^M$ is a partition of $\{1, \ldots, N\}$. We use the usual subsets corresponding to downsampled projection angles [6]. Consider the following *separable* quadratic surrogate $\phi_m$ for the subobjective function $\Phi_m$:

$$\phi_m(\mathbf{z}; \mathbf{x}) = \frac{1}{2}(\mathbf{z} - \mathbf{x})^T \mathbf{C}_m(\mathbf{x})(\mathbf{z} - \mathbf{x})$$

$$+ \nabla \Phi_m(\mathbf{z})(\mathbf{z} - \mathbf{x}) + \Phi_m(\mathbf{x})$$

with

$$\mathbf{C}_m(\mathbf{x}) = \text{diag}_j \{ \tilde{c}_{mj}(\mathbf{x}) \}$$

where $\tilde{c}_{mj}(\cdot) > 0$ and $\text{diag}_j$ denotes a diagonal matrix appropriately formed. The surrogates $\phi_m$ in (13) satisfy Conditions 2 and 3 in Appendix B.

To make $\phi_m$ additionally satisfy the minimization conditions in (3), one has at least two choices for $\tilde{c}_{mj}$: “optimum curvature” (OC) and “maximum curvature” (MC). Those curvatures $\tilde{c}_{mj}$ have the following form:

$$\tilde{c}_{mj}(\mathbf{x}) = \max \{ \tilde{c}_{mj}(\mathbf{x}), \epsilon \}$$

Note that $\psi$ is twice differentiable: $\psi(t) = \delta \cdot t / (\delta + |t|)$ and $\psi(t) = \delta^2 / (\delta + |t|)^2$.

$$\tilde{c}_{mj}(\mathbf{x}) = \sum_{i \in S_m} a_{ij} a_{kj} (\mathbf{A}\mathbf{z})_i$$

$$+ \frac{2\beta}{M} \sum_{k \in N_j} w_{jk} \omega_{jk}(x_j - x_k)$$

for some small value $\epsilon > 0$ where $a_{ij} \triangleq \sum_{j=1}^{J} a_{ij}$ and $\omega_{jk}(\cdot)$ is the wavelet transform of $x_k$. The use of $\epsilon$ keeps the curvatures $\tilde{c}_{mj}$ positive and so the augmented objective function defined in (5) has a unique maximizer with respect to the first argument due to strict concavity. The functions $c_i(\cdot)$ are defined as follows. For OC, we define

$$c_i^{OC}(l) \triangleq \left\{ \begin{array}{ll} -2h_i(0) - \hat{h}_i(l) & l > 0 \\ \hat{h}_i(0) & l = 0 \end{array} \right\}_{+}$$

and for MC

$$c_i^{MC}(l) \triangleq \left\{ \begin{array}{ll} -\hat{h}_i(0) & l = 0 \end{array} \right\}_{+}$$

where $[x]_+ = \max(x, 0)$. On the right side in (15), the first term corresponds to the curvature of a quadratic surrogate for the log-likelihood part, and the second term for the penalty part. The optimum curvature $c_i^{OC}$ in (17) is computed as the lowest curvature of 1D quadratic surrogates, satisfying the minorization conditions, for the marginal log-likelihood $h_i(l)$. A low curvature of a surrogate implies a wide paraboloid which in turn implies a large stepsize, that is, fast convergence rate [27]. Therefore, the optimal curvature is “optimal” in a sense that it leads to the fastest convergence rate (per iteration). However, one needs an “extra” backprojection for computing the first summation in (15). On the other hand, the $c_i^{MC}$ is the MC of the marginal log-likelihood $h_i(l)$ over $l \geq 0$. Therefore, $c_i^{MC}$ is a constant and the first summation in (15) can be precomputed and stored. Detailed derivations of (15)–(18) can be found in [27]. We leave the second summation in (15) as a function of $\mathbf{x}$ even for MC since its computation is usually cheap compared to projection and backprojection operations unless $M$ is too large.

The augmented objective function $F$ defined in (5) with (13) is readily maximized with respect to $\mathbf{x}$ over the box constraint $\mathcal{X}$ as follows:

$$\mathbf{x}_{\text{new}} = P_Y(\mathbf{x}_{\text{new}})$$

$$\mathbf{x}_{\text{new}} = \left[ \sum_{m=1}^{M} \mathbf{C}_m(\mathbf{z}_m) \right]^{-1} \times \sum_{m=1}^{M} \left[ \mathbf{C}_m(\mathbf{z}_m) \mathbf{z}_m + \nabla \Phi_m(\mathbf{z}_m) \right]$$

where $P_Y(\mathbf{x})$ is the orthogonal projection of $\mathbf{x} \in \mathbb{R}^p$ onto $\mathcal{X}$ and is easily computed componentwise as follows: $[P_Y(\mathbf{x})]_j = \text{median}(0, x_j, U)$ for all $j$. Using (19) in the step (T-1) in Table I leads to a new TRIOT algorithm, which is outlined in Table II. When $M = 1$, TRIOT reduces to ordinary SPS [5]. The TRIOT update begins after $\eta_{OS}^{SPS} (\geq 1)$ iteration(s) of OS-SPS [5] (see the next subsection for OS-SPS in detail).
The strategy to switch from OS-SPS to TRIOT is discussed in Section III-D. Running initially (at least) one iteration of OS-SPS is more effective than initializing all $\bar{x}_m$’s to be the same image (e.g., a FBP or uniform image) because both cases require nearly the same computation yet one can take advantage of fast initial convergence rates of OS-SPS.

In Table II, a TRIOT using MC in (18), called TRIOT-MC8, is outlined; however, the OC case in (17) can be easily included. The two steps (T-1) and (T-2) in Table I are combined in Table II. In (T-5), one can avoid the summation $\sum_{l=1}^{M}$ at every subiteration by maintaining that sum as a state vector that is updated incrementally as in [2], [30], [36]. One iteration, indexed by $n$, of TRIOT-MC requires one projection and one backprojection operation while TRIOT-OC needs an extra backprojection [see (15) and (17)].

The discussion and proofs for global convergence given in Section II-B and Appendix B apply to TRIOT. When $r_i \equiv 0$ for all $i$, the algorithm converges to the optimal solution under mild conditions\(^8\) since the PL objective for transmission tomography is strictly concave [69]. In the case where $r_i \neq 0$, the objective function is not necessarily concave [27], and we have a weaker conclusion that every limit point of a sequence generated by TRIOT is a stationary point. However, in our practical experience, we obtained the same limit in all experiments with different initializations, suggesting that suboptimal local maxima are rare, or are far from reasonable starting images.

### C. OS-SPS

Since we use OS-SPS in initializing and accelerating TRIOT, we briefly review OS-SPS [5] for completeness. For each subiteration, indexed by $m$, maximizing the $m$th subobjective $\phi_m(\bar{\mathbf{z}}, \bar{\mathbf{m}})$ in (13) instead of the augmented objective $F(\cdot; \bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_M)$ in (5) leads to the following OS-SPS update:

$$\bar{\mathbf{z}}^{\text{new}}_{(m \mod M)+1} = \mathcal{P}_X \left( \bar{\mathbf{z}}_m + \left[ \hat{\mathbf{C}}_m(\bar{\mathbf{m}}) \right]^{-1} \nabla \Phi_m(\bar{\mathbf{z}}_m) \right)$$

for $m = 1, \ldots, M$. This greedy approach does not ensure monotonicity in the augmented objective $F$ nor the PL objective $\Phi$, so we need not insist that the curvatures satisfy the minorization conditions. A natural choice for $c_m(\cdot)$ is the Newton’s curvature $-\hat{\mathbf{h}}_k(\cdot)$; this can be approximated as follows:

$$-\hat{\mathbf{h}}_k(l) \approx c_k^{\text{PC}} \triangleq \hat{\mathbf{h}}_k \left( \arg \max_{l \geq 0} h_k(l) \right)$$

### TABLE II

**OUTLINE FOR TRIOT ALGORITHM USING MC**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialize:</strong></td>
<td>$\bar{x} = \bar{x}^0$</td>
</tr>
<tr>
<td><strong>Precompute:</strong></td>
<td>$d_{m \cdot}^\text{MC} = \sum_{i \in S_m} a_{ij} a_i N_{i \cdot} \left( \frac{1 - \frac{y_i r_i}{(b_i + r_i)^2}}{b_i e^{-i r_i}} \right)$ and $d_{m \cdot}^\text{PC} = \frac{1}{M} \sum_{i=1}^N a_{ij} c_i^{\text{PC}}, \forall m, j$</td>
</tr>
<tr>
<td><strong>for each iteration $n = 1, \ldots, n_{\text{iter}}$:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>for each subset (subiteration) $m = 1, \ldots, M$:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\hat{y}<em>i = \frac{p}{j=1} a</em>{ij} \hat{x}_j, \quad \hat{h}_i = \left( 1 - \frac{y_i}{b_i e^{-i r_i} + r_i} \right) b_i e^{-i r_i}, \forall i \in S_m$</td>
</tr>
<tr>
<td></td>
<td>$\hat{L}<em>{m \cdot j} = \sum</em>{i \in S_m} a_{ij} \hat{h}<em>i, \quad r</em>{m \cdot j} = \frac{2 \gamma}{M} \sum_{k \in N_i} w_{jk} \psi(\hat{x}_j - \hat{x}_k), \forall j$</td>
</tr>
<tr>
<td></td>
<td>$\bar{x}_{m \cdot j} = \hat{x}_j, \forall j$</td>
</tr>
<tr>
<td><strong>if</strong></td>
<td>$n &lt; n_{\text{iter}}^{\text{OS}}$, perform the following OS-SPS update:</td>
</tr>
<tr>
<td></td>
<td>$\hat{x}<em>j = \left[ \bar{x}</em>{m \cdot j} + \frac{\hat{L}<em>{m \cdot j} - \frac{2 \gamma}{M} \sum</em>{k \in N_i} w_{jk} \psi(\bar{x}<em>{m \cdot j} - \bar{x}<em>k)}{\max \left{ d</em>{m \cdot j}^{\text{PC}} + r</em>{m \cdot j}, \epsilon \right}} \right]^{+}, \forall j$</td>
</tr>
<tr>
<td><strong>else, perform the following TRIOT-MC update:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\hat{x}<em>j = \left[ \sum</em>{l=1}^{M} \left[ \hat{y}<em>j \max \left{ d</em>{l \cdot j}^{\text{MC}} \right} + r_{l \cdot j}, \epsilon \right] \right]^{+}, \forall j$</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td></td>
</tr>
<tr>
<td><strong>if</strong></td>
<td>$n = n_{\text{iter}}^{\text{OS}}$ (the last iteration of OS-SPS), perform (T-5)</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td></td>
</tr>
<tr>
<td><strong>end</strong></td>
<td>$\bar{x}^{n_{\text{iter}}} = \bar{x}$</td>
</tr>
</tbody>
</table>

Here $\epsilon$ is some small positive value; $c_i^{\text{PC}}$ is defined in (22); and $[x]^{+} \triangleq \max\{0, x, U\}$, which should not be confused with $[x]_{+}^{\triangle} \triangleq \max\{x, 0\}$.  

\(^8\)The second part denotes a specific curvature used (e.g., SPS-OC).

\(^9\)The potential function $\psi$ is strictly convex, and $A'g \neq 0$. 

---

**Note:** This text is a snippet from a medical imaging research paper, discussing the TRIOT algorithm and its subroutines for image reconstruction. The focus is on showing how the algorithm converges to a solution, and how it can be initialized and accelerated using OS-SPS. The table outlines a specific algorithmic procedure for using Monte Carlo methods within the TRIOT framework. The text includes mathematical expressions and algorithmic updates, aiming to provide a clear understanding of the computational steps involved in the optimization process.
This choice is called “precomputed curvature (PC)” [5], [27]. For OS-SPS, the following subset-independent preconditioning matrix using PC is usually used in place of $\mathbf{C}_m(\mathbf{x}_m)$ in (21):

$$
\mathbf{C}^{PC}(\mathbf{x}) = \text{diag} \left\{ c^PC_j(\mathbf{x}) \right\}
$$

$$
c^PC_j(\mathbf{x}) = \max \left\{ c^PC_j(\mathbf{x})_k \varepsilon \right\}
$$

(23)

where

$$
c^PC_j(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{N} a_{ij} k^PC_i + \frac{2\beta}{M} \sum_{k \in N_j} w_{jk} \phi_i(x_j - x_k),
$$

(24)

The first term on the right side in (24) can be precomputed and stored. The benefit of using PC is that it leads to faster convergence rates than MC since $\gamma_i^{MC} \geq \gamma_i^{PC}$. The update for OS-SPS is shown in (T-4) in Table II.

The OS-SPS shows very fast initial convergence rates but becomes eventually stuck at a limit cycle. Using more subsets leads to a faster initial convergence rate but causes the points in the limit cycle to be farther from the optimal solution.

It is worth noting that, for each update, OS-SPS uses the gradient and curvature for only one subobjective function at the previous subiterate in (21) whereas TRIOT uses the gradients and curvatures for all subobjective functions at previous $M$ subiterates in (20). When the number of subobjective functions is $M = 1$, then both OS-SPS and TRIOT reduce to SPS.

### D. Acceleration

Usually, TRIOT-OC/MC is initially slower than the non-convergent OS-SPS. Here, we discuss methods to accelerate TRIOT.

1) Switch From OS-SPS to TRIOT: It is a popular idea to switch from a nonconvergent yet initially fast OS type algorithm to a convergent non-OS algorithm at some point to take advantage of both fast initial convergence rates of OS methods and global convergence of non-OS methods.

We observed that it is very effective to switch to TRIOT from OS-SPS at the point where the OS-SPS algorithm nearly gets to a limit cycle; even one single subiteration of TRIOT moves the iterate from the limit cycle to some point very close to the optimal solution. The reason is as follows: a group of the points in the limit cycle would be roughly centered around the optimal point and the update for TRIOT includes a weighted average of the points [see the first term on the right side in (20) or (T-5)].

To obtain further insight into this property, consider a simple unconstrained quadratic problem where the objective function and the subobjective functions are

$$
\Phi(\mathbf{x}) = \frac{1}{2} \mathbf{x}'Q\mathbf{x} + b'\mathbf{x}, \quad \Phi_m(\mathbf{x}) = \frac{1}{2} \mathbf{x}'Q_m\mathbf{x} + b_m'\mathbf{x}
$$

for $m = 1, \ldots, M$ where $\sum_{m=1}^{M} Q_m = Q$ and $\sum_{m=1}^{M} b_m = b$.

Assume that each surrogate function $\phi_m(\mathbf{x}; \mathbf{z})$ is equal to its corresponding subobjective $\Phi_m(\mathbf{x})$ so it has a closed-form maximizer $\hat{x}_m = Q_m^{-1}b_m$ where we assume each $Q_m$ is invertible.

Then, the OS approach will generate a limit cycle that consists of those $\hat{x}_1, \ldots, \hat{x}_M$. Now applying just one iteration of the incremental optimization transfer method as in (19) leads to

$$
\hat{x} = \left( \sum_{m=1}^{M} Q_m \right)^{-1} \sum_{m=1}^{M} Q_m \hat{x}_m = Q^{-1} \sum_{m=1}^{M} Q_m (Q_m^{-1} b_m)
$$

which is the maximizer of the original objective $\Phi$ [the second term on the right side in (20) equals zero]. This example suggests that the built-in averaging operation in TRIOT helps escape from a limit cycle, generated by nonconvergent OS algorithms, toward the optimal solution.

However, in the early iterations, when OS-SPS is still far from the limit cycle and is making progress toward the optimal point, TRIOT is usually slower than OS-SPS due to the averaging of the past subiterates because the incremental optimization transfer approach updates the surrogates incrementally, that is, conservatively to ensure monotonicity. So it is desirable to get to a limit cycle quickly using OS-SPS with many subsets and then switch to TRIOT. In a 2D reconstruction case in Section IV, the use of 64 subsets is sufficient to reach a limit cycle within a couple of iterations.

2) Precomputed Curvatures: Forgoing monotonicity (in the augmented objective) and accordingly provable convergence, one can use for TRIOT the “precomputed curvatures (PC)” in (22). TRIOT-PC is faster than provably convergent TRIOT-OC/MC. It is an open question whether TRIOT-PC converges to an optimal solution. However, in our experiments, TRIOT-PC yielded the same limit as convergent algorithms like SPS-OC within numerical precision!

### IV. RESULTS

#### A. Algorithms Performance Evaluation: Real Data Study

To assess the performance of the proposed algorithms, we performed 2D attenuation map reconstructions from real PET data.

We acquired PET data using a Siemens/CTI ECAT EXACT 921 PET scanner with rotating rod transmission sources [70]. We used an anthropomorphic thorax phantom (Data Spectrum, Chapel Hill, NC). The sinogram had 160 radial bins and 192 angles, and the reconstructed images were $128 \times 128$ with 4.2-mm pixels. The system geometry was approximated with $3.375$-mm-wide strip integrals and $3.375$-mm ray spacing; the system matrix was generated using ASPIRE [71]. The total counts amounted to $9.2 \times 10^5$. We used the edge-preserving nonquadratic penalty (12) with $\delta = 4 \times 10^{-4}$ mm$^{-1}$ and $\beta = 218.5$, chosen by visual inspection. A uniform image was used as a starting image. The results obtained by using a FBP reconstruction as a starting image were similar and are not shown here. We set $U$ in (11) to be $7$ mm$^{-1}$, which was much higher than reconstructed values ($< 0.02$ mm$^{-1}$), and the upper bound was never active.

Images were reconstructed using SPS-MC/PC, OS-SPS, and TRIOT-MC/PC. For OS-SPS and TRIOT algorithms, we used 16 subsets (a moderate number) and 64 subsets (a little larger
number than usual). For SPS and TRIOT, the performance (objective value or distance from the optimal image) with the optimum curvature (OC) in (17), which requires an extra backprojection per iteration, was between those with MC and PC (see [55, pp. 85–86] for the results with OC).

Fig. 1 shows normalized $\Phi$ difference versus iteration number for different algorithms using 16 subsets. The normalized $\Phi$ difference is defined as $(\Phi(x_{PL}^\text{opt}) - \Phi(x^*))/(\Phi(x_{PL}^\text{opt}) - \Phi(x_0^*))$ where $x_{PL}^\text{opt}$ is a maximizer of the PL objective and $x_0^*$ is the initial uniform image; a small value of the normalized difference means the image is closer to the optimal image $x_{PL}^\text{opt}$. The optimal image $x_{PL}^\text{opt}$ [shown in Fig. 4(a)] was estimated by 30 iterations of OS-SPS-16 (where “16” means the number of subsets) followed by 800 iterations of the SPS-OC algorithm that is monotonic and convergent (to a stationary point). For TRIOT, as described in Section III-B, we ran one iteration of OS-SPS, and then performed TRIOT iterations. Only for the last iteration of OS-SPS (in this case, iteration 1), we performed an additional TRIOT update in (T-6) in Table II that takes negligible computation. To see the effects of “this” TRIOT update, we plotted normalized $\Phi$ difference values both before and after (T-6) at iteration 1, which yields the vertical lines at iteration 1 for TRIOT algorithms in Fig. 1. For TRIOT, the jump in the unfavorable direction is due to the fact that built-in averaging slows down the algorithm when a limit cycle has not yet been reached as discussed in Section III-D. For a fair comparison, the SPS algorithms were performed after one iteration of OS-SPS was run. That is, for all methods shown in Fig. 1, $x^*$ denotes the reconstructed image obtained using one iteration of OS-SPS. Although OS-SPS showed a fast initial convergence rate, it became stuck at a suboptimal point whereas other methods continued to improve in terms of objective values. The TRIOT algorithms were outperformed by other algorithms in early iterations. However, TRIOT-MC and TRIOT-PC eventually outperform SPS-MC and SPS-PC, respectively. Although global convergence is not provably ensured for TRIOT-PC, the limit of TRIOT-PC (say, obtained by 1000 iterations) was the same as that of SPS-OC (obtained similarly) within numerical precision, which suggests TRIOT-PC has desirable convergence properties.

To investigate the performance of TRIOT algorithms after OS-SPS reaches a limit cycle, we performed 6 iterations of OS-SPS, which is sufficient to get close to a limit cycle, and then applied TRIOT (and SPS as well). Fig. 2 shows that TRIOT yielded considerable improvement at iteration 6 where TRIOT was first applied. TRIOT-MC and TRIOT-PC converge faster than SPS-MC and SPS-PC, respectively. (SPS-PC outperforms TRIOT-MC from iteration 16 on.) This shows that it is effective to switch from OS-SPS to TRIOT, as described in Section III-D, when OS-SPS almost reaches a limit cycle. However, it is still inconvenient to predict how many iterations are required for OS-SPS to arrive at a limit cycle.

Fig. 3 shows normalized $\Phi$ difference versus iteration number when 64 subsets are used. As the number of subsets increased to 64, the initial convergence rate of OS-SPS became faster (even a couple of iterations led to a limit cycle) but OS-SPS stagnated at a worse image. Meanwhile, for the TRIOT algorithms, a significant improvement was obtained at iteration 2 when OS-SPS was switched to TRIOT, and the TRIOT algorithms outperformed the SPS algorithms. The switching point (iteration 2) was determined by some experimentation. In light of the effectiveness of the built-in averaging in TRIOT, to make SPS a stronger competitor, prior to switching to SPS (at iteration 2), we averaged the 64 previous subiterates that approximately comprise the limit cycle. As shown in Fig. 3, this averaging yielded improvements for SPS algorithms. However, convergence rates of TRIOT were still faster than those of SPS with such averaging. A plot of the normalized distance from the optimal image, $||x^* - x_{PL}^\text{opt}||/||x_{PL}^\text{opt}||$, versus iteration number showed a similar trend (not shown here).

Figs. 4(c), (d), and (e) show the images to which OS-SPS with 16, 32, and 64 subsets converged, respectively. They represent one point of the limit cycle generated by the OS-SPS. For 64
The total counts amounted to 1M, and the known objects were obtained using 30 iterations of OS-SPS algorithm. This figure also shows the performance of SPS algorithms that include averaging 64 subiterates after 2 iterations of OS-SPS.

subsets, the reconstructed image looks different from the true PL optimal image $\hat{\mathbf{x}}_{\text{PC}}$ in Fig. 4(a), whereas the reconstructed image for 16 subsets looks similar to the PL image. In contrast, 18 iterations of TRIOT-PC initialized by 2 iterations of OS-SPS yielded the image in Fig. 4(f) which is indistinguishable from the optimal image in Fig. 4(a). TRIOT-MC and -OC (18 iterations with initial 2 iterations of OS-SPS) in Figs. 4(g) and (h) also yielded images very similar to the optimal image. The normalized distance (ND) $\|\hat{\mathbf{x}}_{\text{PL}} - \mathbf{x}\|/\|\mathbf{x}_{\text{PL}}\|$ from the optimal image is shown in Fig. 4; TRIOT-PC showed the minimum ND from the PL image. Fig. 5 shows the horizontal profile through the reconstructed images in Fig. 4 for TRIOT-PC, OS-SPS-16, and OS-SPS-64. Again, the TRIOT-PC profile agreed very well with the PL image profile, and OS-SPS-64 showed a noticeable deviation from the PL image.

B. Convergence Really Matters?: Simulation Study

Generally, ordered subsets or incremental algorithms converge to a limit cycle that consists of suboptimal solutions. In this paper we focus on developing incremental algorithms that converge to a true optimal solution. To examine how “suboptimal” the limit cycle solutions are, we compare nonconvergent OS algorithms (OS-SPS) and convergent TRIOT through a simulation study for 2D attenuation map reconstruction.

We used the same scanner geometry as in the previous subsection. The synthetic digital phantom shown in Fig. 6 was used; the attenuation coefficients of the warm background, left cold disc, and right hot disc were 0.004 mm$^{-1}$, 0.009 mm$^{-1}$, and 0.001 mm$^{-1}$. The total counts amounted to 1M, and the known background contribution $r_i$ corresponded to a uniform field of 10%. We used the penalty in (12) with $\delta = 5 \times 10^{-4}$ mm$^{-1}$ and $\beta = 2^{1/2}$.

The aim of the simulation is to compare the bias and variance of the suboptimal images obtained by nonconvergent OS-SPS algorithms and the (nearly) optimal image obtained by TRIOT. We generated 400 realizations of pseudorandom transmission
measurements according to (8). For each realization, we obtained a limit cycle image for OS-SPS-8 and OS-SPS-16 by 50 iterations, and for OS-SPS-32 and OS-SPS-64 by 20 iterations (note that it takes more iterations for OS algorithms with fewer subsets to reach a limit cycle); we also obtained a (nearly) optimal image by 18 iterations of TRIOT-PC with 64 subsets after 2 iterations of OS-SPS-64.

A limit cycle consists of as many images as the number of subsets. We took only one image, say the first one, from the limit cycle for each OS method and computed the sample mean and variance over 400 realizations. Taking another image (e.g., the second one) from the limit cycle led to similar results. Fig. 7 shows the profiles through the sample mean images for different methods; all images were nearly free of systematic biases. For each pixel, we computed the ratios of the sample standard deviation of OS methods to the sample standard deviation of TRIOT, and Fig. 8 shows the histogram of the ratios. Overall, OS-SPS-8 showed almost the same level of standard deviation as TRIOT; and OS-SPS-16, -32, and -64 yielded larger standard deviation than TRIOT by about 5%, 20%, and 58%, respectively. The noise increase can be attributed to the limit cycle generated by OS algorithms. As the number of subsets decreases, the variance decreases, but the (initial) convergence rate also decreases and it takes longer to converge to a limit cycle. We also tried averaging the limit cycle images; the resulting sample variance decreased but they were still larger than or equal to that of TRIOT depending on the number of subsets.

This example illustrates the importance of convergence for regularized methods. To summarize, as the number of subsets increases, a nonconvergent OS algorithm leads to higher variance although it does not increase bias significantly; therefore, it gives lower contrast-to-noise ratios. It was also observed in [13], [16] that the image noise increases as the number of subsets increases. Although using a small number of subsets leads to an image very similar to the one obtained by a convergent non-OS algorithm, its convergence acceleration factor is smaller than that for a large number of subsets. Therefore, there is a trade-off between speed acceleration and image quality depending on the number of subsets. In this paper, we achieve both fast convergence rates and image quality (convergence) by using the incremental optimization transfer. In [13], the image degradation caused by a large number of subsets was corrected by subsequently using a reduced number of subsets. Which method is preferable is an open question.

V. CONCLUSION

We presented a broad family of incremental optimization transfer algorithms by generalizing the incremental EM family. The incremental optimization transfer algorithms usually show faster convergence rates than ordinary optimization transfer methods like EM, and they are globally convergent.

We also developed a particular incremental optimization transfer algorithm for transmission tomography by using separable quadratic surrogates: TRIOT algorithms. We found that it is very effective to switch from OS-SPS to TRIOT when OS-SPS nearly reaches a limit cycle. The switching idea is also found in [72]. But we need to determine when to switch from
OS-SPS to TRIOT as we must determine the relaxation parameters for relaxed OS algorithms such as RAMLA, BSREM, and relaxed OS-SPS. However, from our experience, it seems more convenient to determine when to switch than to choose relaxation parameters. In fact, when reasonably many subsets were used, as few as a couple of iterations of OS-SPS were sufficient to get close to a limit cycle for our 2D reconstruction. This switching strategy seems robust since we obtained similar results from a 2D simulation study using a different digital phantom [55]. Recently, Li et al. have proposed a method to determine automatically the switching point by fitting an exponential function to (sub)objective function values computed at previous iterates and by determining how close to a limit cycle a current iterate is [49].

TRIOT may be preferable to reducing the number of subsets with iteration when the consistent data flow is beneficial since the number of subsets remains unchanged over iterations for the method using OS-SPS initially and TRIOT later. And it might be easier to determine the point when to switch from OS-SPS to TRIOT than to determine an optimal schedule for reducing the number of subsets. However, the question of which method is better remains open, and further investigation will be needed.

One iteration of TRIOT-MC/PC or OS-SPS requires computing one projection and one backprojection plus the penalty related gradients and curvatures (the use of OC needs an extra computing one projection and one backprojection plus the penalty part being updated while keeping TRIOT available particularly for large-sized problems. It will be an interesting challenging problem to resolve the memory issues for each update it uses previous updates (where TRIOT may be preferable to reducing the number of subsets). However, the question of which method is better remains open, and further investigation will be needed.

Assumption 3: Consider Assumption 2: if

\[ \Delta \hat{\theta} \]

is a map from \( \hat{\theta} \) to \( \hat{\theta} \) for a nonempty, closed, and convex set \( \mathcal{X} \subset \mathbb{R}^p \) for all \( \hat{\theta} \in \mathcal{X} \), and each \( \phi_m(\hat{\theta}) \) is continuous with respect to \( \hat{\theta} \) on \( \mathcal{X} \) for all \( \hat{\theta} \in \mathcal{X} \). The feasible set \( \mathcal{X} \) is the true value of the unknown parameter. Assume that we identify an admissible complete-data random vector \( \mathbf{Z} \) for \( f(y; \mathbf{x}) \). Then, the following EM surrogate function satisfies the minimization conditions in (3) [33]:

\[
\phi(\mathbf{x}; \bar{\mathbf{z}}) = E[\log f(\mathbf{z}; \mathbf{x}) | Y = y; \bar{\mathbf{z}}]
\]

for all \( \bar{\mathbf{z}} \in \mathcal{X} \). But in many applications including imaging problems, the observed data is independent so the log-likelihood objective is additive-separable, that is

\[
\phi(\mathbf{x}) = \sum_{m=1}^{M} \phi_m(\mathbf{x})
\]

and the complete data is conditionally independent, so for each \( \phi_m(\mathbf{x}) \), one can obtain the following EM surrogate:

\[
\phi_m(\mathbf{x}; \bar{\mathbf{z}}) = E[\log f(\mathbf{z}; \mathbf{x}) | Y_m = y_m; \bar{\mathbf{z}}]
\]

which also satisfies the minimization conditions in (3) where \( Y = (Y_1, \ldots, Y_M) \) and \( \mathbf{Z} = (Z_1, \ldots, Z_M) \) are some decompositions of the incomplete data and the complete data, respectively. Defining the augmented objective function as in (5) and then alternating between updating \( \mathbf{x} \) and one of the \( \bar{\mathbf{z}}_m \)'s as in Table I leads to the incremental EM algorithms [3], [31]. The COSEM algorithm [2], [30], a special case of the incremental EM for emission tomography, can be readily derived.

APPENDIX B

GLOBAL CONVERGENCE PROOF

In this appendix we prove the convergence of the incremental optimization transfer algorithm given in Table I. Define \( \mathbf{z} \equiv (\mathbf{x}; \bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_M) \in \mathcal{X}^{M+1} \), and define a map \( \mathcal{M} : \mathcal{X}^{M+1} \rightarrow \mathcal{X}^{M+1} \) such that \( \mathbf{z}^{n+1} = \mathcal{M}(\mathbf{z}^n) \) where \( \mathbf{z}^{n+1} = (x^{n+1}; \bar{\mathbf{z}}_1^{n+1}, \ldots, \bar{\mathbf{z}}_M^{n+1}) \) is computed by (T-1)–(T-3) for \( \mathbf{z}^n = (x^n; \bar{\mathbf{z}}_1^n, \ldots, \bar{\mathbf{z}}_M^n) \). Suppose that the algorithm generates a sequence \( \{\mathbf{z}^n\} \) (or a sequence \( \{\mathbf{z}^n\} \) by taking the first component of \( \mathbf{z}^n \), given some initial point \( \mathbf{z}^0 \in \mathcal{X}^{M+1} \). Define an augmented solution set as follows:

\[
\Lambda \equiv \{ \mathbf{z} = (\mathbf{x}; \bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_M) \in \mathcal{X}^{M+1} : \mathbf{x} \in \Gamma \} \tag{27}
\]

where \( \Gamma \) is defined in (7). We impose the following assumptions.

**Assumption 1:** Each \( \phi_m(\mathbf{x}) \) and \( \phi_m(\mathbf{x}; \bar{\mathbf{z}}) \) is differentiable with respect to \( \mathbf{x} \) on a nonempty, closed, and convex set \( \mathcal{X} \subset \mathbb{R}^p \) for all \( \mathbf{x} \in \mathcal{X} \) and each \( \phi_m(\mathbf{x}; \bar{\mathbf{z}}) \) is continuous with respect to \( \bar{\mathbf{z}} \) on \( \mathcal{X} \) for all \( \mathbf{x} \in \mathcal{X} \).

**Assumption 2:** The solution set \( \Gamma \) of stationary points defined in (7) is nonempty and, therefore, so is the augmented solution set \( \Lambda \) defined in (27).

**Assumption 3:** The iterates \( \{\mathbf{z}^n\} \) are bounded where \( \mathbf{z}^n = (x^n; \bar{\mathbf{z}}_1^n, \ldots, \bar{\mathbf{z}}_M^n) \).

Assumption 3 is ensured by either of the following sufficient conditions.

**Assumption 3’:** A level set defined by \( \{ \mathbf{z} \in \mathcal{X}^{M+1} : F(\mathbf{z}) \geq F(\mathbf{z}^0) \} \) is bounded where \( F \) is the augmented objective function defined in (5).

**Assumption 3’’:** The feasible set \( \mathcal{X} \) is bounded.
Note that Assumptions 1 and 3 imply Assumption 2 by the Weierstrass’ Theorem [45, p. 654]. We assume that the surrogates \( \phi_M \) satisfy the following conditions.

**Condition 1:** The functionals \( \phi_M \) satisfy the minorization conditions in (3).

**Condition 2:** The following derivatives match for all \( m \) and \( x \in X \):

\[
\nabla \phi_M(x) = \nabla^{10} \phi_M(x; x)
\]

(28)

where \( \nabla^{10} \) is the column gradient operator with respect to the first argument (see [42] for less restrictive conditions).

**Condition 3:** There exists a unique maximizer in (T-1).

The following is sufficient for Condition 3.

**Condition 3’:** Each \( \phi_M(\cdot; \xi_m) \) is strictly concave for all \( \xi_m \in X \), and there exists a maximizer of \( F(\cdot; \xi_1, \ldots, \xi_M) \) over \( X \), for all \( \xi_1, \ldots, \xi_M \in X \).

**Condition 4:** There exists a unique maximizer in (T-2).

The following is sufficient for Condition 4.

**Condition 4’:** For all \( x, \xi \in X \), \( \phi_M(x) \geq \phi_M(x; \xi) \) where the equality holds if and only if \( x = \xi \).

Even if there exist \( x, \xi \in X \) such that \( x \neq \xi \) and \( \phi_M(x) = \phi_M(x; \xi) \), using a modified surrogate \( \phi_M(x; \xi; \varepsilon) = \phi_M(x; \xi) - \varepsilon \| x - \xi \|^2 \) for a fixed \( \varepsilon > 0 \) will lead to Conditions 4’ and 4.

**Condition 5:** If \( A \) is a bounded subset of \( X \), then \( \{ \max_{x \in X} F(\cdot; \xi_1, \ldots, \xi_M) : (\xi_1, \ldots, \xi_M) \in A \} \) is also bounded.

Note that Assumption 3’ implies Condition 5. Using the above assumptions and conditions, we prove a series of lemmas necessary for proving convergence.

**Lemma 1:** The map \( M \) such that \( z^{n+1} = M(z^n) \) is continuous.

**Proof:** The map \( M \) is the composition of \( 2M \) maps: \( M = S^M T S^{M-1} T \cdots S^1 T \) where \( S^m(x; \xi_1, \ldots, \xi_M) \triangleq \chi(x; \xi_1, \ldots, \xi_m-1, x, \xi_{m+1}, \ldots, \xi_M) \),

\[ T(x; \xi_1, \ldots, \xi_M) \triangleq \left( u_1, \ldots, u_M \right) \text{ with } u = \max_{x \in X} F(x; \xi_1, \ldots, \xi_M) \] and \( u^n = \max_{x \in X} F(x; \xi_1, \ldots, \xi_M) \). Since each \( S^m \) is continuous, it is only necessary to show that \( T \) is continuous [50, p. 187]. Let \( \{ \xi^n \} \) be a sequence from \( X \) with \( \xi^n \rightarrow \xi^m \) for all \( m \). Suppose that \( u^k = \max_{x \in X} F(x; \xi_1, \ldots, \xi_M) \) and \( u^n = \max_{x \in X} F(x; \xi_1, \ldots, \xi_M) \). It suffices to show that \( u^k \rightarrow u^n \), and suppose that \( u^n \) does not converge to \( u^k \). Then, there exists a subsequence \( \{ u^{n_k} \} \) of \( \{ u^n \} \) such that \( u^{n_k} \rightarrow u^* \) for some \( u^* \neq u^n \) by Condition 5. By the definition of \( u^k, F(u^{n_k}; \xi_1, \ldots, \xi_M) \geq F(u^k; \xi_1, \ldots, \xi_M) \). By letting \( l \rightarrow \infty \), we have \( F(u^m; \xi_1, \ldots, \xi_M) \geq F(u^k; \xi_1, \ldots, \xi_M) \) since \( F \) is continuous by Assumption 1. But this is a contradiction in view of the definition of \( u^m \) and Condition 3. Therefore, it must be the case that \( u^k \rightarrow u^n \), and the conclusion follows.

**Lemma 2:** The iterates \( \{ x^n \} \) generated by (T-1)-(T-3) yield monotonic increases in \( F \), that is, \( F(x^{n+1}) \geq F(x^n) \) for all \( n \).

**Proof:** It follows from the cyclic block coordinate ascent updates in (T-1) and (T-2).

---

11\( \text{If } x \text{ is an interior point of } X \), Condition 2 is implied by Condition 1 [55, Lemma 3.3].
REFERENCES


