Combining Ordered Subsets and Momentum for Accelerated X-Ray CT Image Reconstruction

Donghwan Kim*, *Member, IEEE*, Sathish Ramani, *Senior Member, IEEE*, and Jeffrey A. Fessler, *Fellow, IEEE*

***Abstract*—Statistical X-ray computed tomography (CT) reconstruction can improve image quality from reduced dose scans, but requires very long computation time. Ordered subsets (OS) methods have been widely used for research in X-ray CT statistical image reconstruction (and are used in clinical PET and SPECT reconstruction). In particular, OS methods based on separable quadratic surrogates (OS-SQS) are massively parallelizable and are well suited to modern computing architectures, but the number of iterations required for convergence should be reduced for better practical use. This paper introduces OS-SQS-momentum algorithms that combine Nesterov's momentum techniques with OS-SQS methods, greatly improving convergence speed in early iterations. If the number of subsets is too large, the OS-SQS-momentum methods can be unstable, so we propose diminishing step sizes that stabilize the method while preserving the very fast convergence behavior. Experiments with simulated and real 3D CT scan data illustrate the performance of the proposed algorithms.**

***Index Terms*—Computed tomography (CT), momentum, ordered subsets, parallelizable iterative algorithms, relaxation, separable quadratic surrogates, statistical image reconstruction, stochastic gradient.**

I. Introduction

STATISTICAL X-ray computed tomography (CT) image reconstruction methods can provide images with improved resolution, reduced noise and reduced artifacts from lower dose scans, by minimizing regularized cost-functions [1]–[4]. However, current iterative methods require too much computation time to be used for every clinical scan. Many general iterative algorithms have been applied to statistical CT reconstruction including coordinate descent [5], [6], preconditioned conjugate gradient [7], and ordered subsets [8], [9], but these algorithms all converge slower to the minimizer than is desired for clinical CT. This paper describes new image reconstruction algorithms that require less computation time.

Recent advances on iterative algorithms in X-ray CT minimize the cost function by using splitting techniques [10]–[12] accompanied with the method-of-multipliers framework [13]. The Chambolle-Pock primal-dual algorithm [14] has been applied to tomography [15], [16]. Momentum techniques [17]–[19] that use previous update directions have been applied to X-ray CT [20]–[22], accelerating a gradient descent update using the Lipschitz constant of the gradient of the cost function.

This paper focuses on momentum techniques that have received wide attention in the optimization community. Nesterov [17], [18] developed two momentum techniques that use previous descent directions to decrease the cost function at the fast convergence rate $O(1/n^2)$, where n counts the number of iterations. The rate $O(1/n^2)$ is known to be optimal[1] for first-order[2] optimization methods [23], while ordinary gradient descent has the rate $O(1/n)$. Nesterov's momentum algorithms have been extended to handle nonsmooth convex functions [19], [24], and have been applied to image restoration problems [25], [19].

Momentum techniques in X-ray CT [20]–[22] have been used to accelerate gradient descent methods. However, these "traditional" momentum algorithms do not show significant improvement in convergence speed compared with other existing algorithms, due to the large Lipschitz constant of the gradient of the cost function [11]. Here, we propose to combine momentum techniques with ordered subsets (OS) methods [8], [9] that provide fast initial acceleration. (Preliminary results based on this idea were discussed in [26] and [27].) OS methods, an instance of incremental gradient methods [28], approximate a gradient of a cost function using only a subset of the data to reduce computational cost per image-update. Even though the approximation in the method may prevent the algorithm from converging to the optimum, OS methods are widely used in tomography problems for their M-times initial acceleration in run time, where M is the number of subsets, leading to rate $O(1/(nM))$ in early iterations. Remarkably, our proposed OS-momentum algorithms should have the rate $O(1/(nM)^2)$ in early iterations providing a promising M^2 acceleration compared to the standard Nesterov method.

Manuscript received July 25, 2014; accepted August 18, 2014. Date of publication August 22, 2014. Date of publication August 22, 2014; date of current version December 24, 2014. This work was supported in part by GE Healthcare, the National Institutes of Health under Grant R01-HL-098686 and Grant U01-EB-018753, and equipment donations from Intel Corporation. *Asterisk indicates corresponding author.*

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

*D. Kim is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105 USA (e-mail: kimdongh@umich.edu).

S. Ramani is with the GE Global Research Center, Niskayuna, NY 12309 USA (e-mail: dr.s.r@ieee.org).

J. A. Fessler is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105 USA (e-mail: fessler@umich.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2014.2350962

[1]Nesterov [23] showed that there exists at least one convex function that cannot be minimized faster than the rate $O(1/n^2)$ by any first-order optimization methods. Therefore, the rate $O(1/n^2)$ is optimal for first-order methods in convex problems.

[2]First-order optimization methods refer to a class of iterative algorithms that use only first-order information of a cost function such as its value and its gradient.

Conventional momentum methods use either the (smallest) Lipschitz constant or a backtracking scheme to ensure monotonic descent in the gradient step [19]. Here, to reduce computation, we instead use diagonal preconditioning (or majorizing) based on the separable quadratic surrogate (SQS) [9] that is used widely for designing monotonic descent algorithms for X-ray CT. Another advantage of using SQS is that we can further accelerate the algorithm using the nonuniform approach in SQS methods that provides larger updates for the voxels that are far from their respective optima [29].

Combining OS methods with Nesterov's momentum techniques directly is a practical approach for acceleration but lacks convergence theory.[3] Indeed, we observed unstable behavior in some cases. To stabilize the algorithm, we adapt the diminishing step size rule developed for stochastic gradient method with Nesterov's momentum in [33]. We view the OS-SQS methods in a stochastic sense and study the relaxation scheme of momentum approach [33]. We investigate various relaxation schemes to achieve both fast initial acceleration and stability (or convergence) in a stochastic sense. [Note that this relaxation (or diminishing step size) is not necessary if OS is not used in the proposed algorithms.]

This paper is organized as follows. Section II explains the problem of X-ray CT image reconstruction and Section III summarizes the OS-SQS algorithms. Section IV reviews momentum techniques and combines them with OS-SQS. Section V suggests a step size relaxation scheme that stabilizes the proposed algorithms. Section VI shows experimental results on simulated and real CT scans. Section VII offers conclusion and discussion.

II. Problem

We consider a (simplified) linear model for X-ray CT transmission tomography

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{N_d}$ is (post-log) measurement data, $\boldsymbol{A} \triangleq \{a_{ij}\} \in \mathbb{R}_+^{N_d \times N_p}$ is a forward projection operator [34], [35] ($a_{ij} \geq 0$ for all i, j), $\boldsymbol{x} \triangleq \{x_j\} \in \mathcal{X} = \mathbb{R}_+^{N_p}$ is an unknown (nonnegative) image (of attenuation coefficients) to be reconstructed and $\boldsymbol{\epsilon} \in \mathbb{R}^{N_d}$ is noise.

A penalized weighted least squares (PWLS) [3], [4] criterion is widely used for reconstructing an image $\boldsymbol{x}$ with a roughness penalty and a nonnegativity constraint:[4]

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \succeq \boldsymbol{0}}{\arg\min} \left\{ \Psi(\boldsymbol{x}) \triangleq \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{W}}^2 + R(\boldsymbol{x}) \right\} \tag{2}$$

[3]Some previous works combine incremental gradient methods [28] and relatively small momentum with convergence analysis [30]–[32], but our focus is to use larger momentum like Nesterov's methods with OS methods in tomography problem for fast initial convergence rate.

[4]For two vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ of the same size, the expression $\boldsymbol{x} \succeq \boldsymbol{z}$ (or $\boldsymbol{x} \succ \boldsymbol{z}$) means that $\boldsymbol{x} - \boldsymbol{z}$ is element-wise nonnegative (or element-wise positive). For two symmetric matrices $\boldsymbol{X}$ and $\boldsymbol{Z}$ of the same size, the notation $\boldsymbol{X} \succeq \boldsymbol{Z}$ (or $\boldsymbol{X} \succ \boldsymbol{Z}$) means that $\boldsymbol{X} - \boldsymbol{Z}$ is positive semidefinite (or positive definite). A weighted Euclidean seminorm is defined as $\|\boldsymbol{y}\|_{\boldsymbol{W}} \triangleq \sqrt{\sum_{i=1}^{N_d} w_i y_i^2}$ for a vector $\boldsymbol{y} \triangleq \{y_i\}$ and a positive semidefinite diagonal matrix $\boldsymbol{W} \triangleq \mathrm{diag}\{w_i\}$.

where the diagonal matrix $\boldsymbol{W} \succeq \boldsymbol{0}$ provides statistical weighting that accounts for the ray-dependent variance of the noise $\boldsymbol{\epsilon}$. Here, we focus on smooth[5] convex regularization functions:

$$R(\boldsymbol{x}) \triangleq \sum_{r=1}^{N_r} \psi_r([\boldsymbol{C}\boldsymbol{x}]_r) \tag{3}$$

where $\psi_r(t) \triangleq \beta_r \psi(t)$, the function $\psi(\cdot)$ is an edge-preserving potential function such as a Fair potential function in [36], the parameter β_r provides spatial weighting [37], and $\boldsymbol{C} \triangleq \{c_{rj}\} \in \mathbb{R}^{N_r \times N_p}$ is a finite-differencing matrix considering 26 neighboring voxels in 3D image space. The regularizer (3) makes the PWLS cost function $\Psi(\boldsymbol{x})$ in (2) to be smooth and strictly convex with a unique global minimizer $\hat{\boldsymbol{x}}$ [38]. Throughout this manuscript, we assume that the objective $\Psi(\boldsymbol{x})$ is smooth and (strictly) convex.

This nonquadratic PWLS cost function cannot be optimized analytically and requires an iterative algorithm. We propose algorithms combining OS and (relaxed) momentum approaches that minimize a smooth convex objective function $\Psi(\boldsymbol{x})$ for CT rapidly and efficiently. Although we focus on PWLS for simplicity, the methods may also apply to penalized-likelihood formulations and to cost functions with nonsmooth regularizers, which we leave as future extensions.

III. OS-SQS Algorithms

A. Optimization Transfer Method

When a cost function $\Psi(\boldsymbol{x})$ is difficult to minimize, we can replace it by a (simple) surrogate $\phi(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ at the nth iteration, and generate a sequence $\{\boldsymbol{x}^{(n)}\}$ by minimizing the surrogate as

$$\boldsymbol{x}^{(n+1)} = \underset{\boldsymbol{x} \succeq \boldsymbol{0}}{\arg\min}\, \phi\left(\boldsymbol{x}; \boldsymbol{x}^{(n)}\right). \tag{4}$$

This optimization transfer method [39] is also known as a majorization-minimization approach [40, Sec. 8.3].

To monotonically decrease $\Psi(\boldsymbol{x})$ using an optimization transfer method, a surrogate function $\phi(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ at nth iteration should satisfy the following majorization conditions:

$$\begin{cases} \Psi\left(\boldsymbol{x}^{(n)}\right) = \phi\left(\boldsymbol{x}^{(n)}; \boldsymbol{x}^{(n)}\right), \\ \Psi(\boldsymbol{x}) \leq \phi\left(\boldsymbol{x}; \boldsymbol{x}^{(n)}\right), \quad \forall \boldsymbol{x} \in \mathcal{X} = \mathbb{R}_+^{N_p}. \end{cases} \tag{5}$$

Surrogates satisfying the conditions in (5) can be constructed using a Lipschitz constant [19], quadratic surrogates [41], and SQS methods [9], [29].

B. SQS Algorithms

An optimization transfer method used widely in tomography problems is a SQS method [9] yielding the following surrogate function $\phi(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ with a diagonal Hessian (second-order derivatives) matrix $\boldsymbol{D} \triangleq \mathrm{diag}\{d_j\}$, at nth iteration:

$$\phi\left(\boldsymbol{x}; \boldsymbol{x}^{(n)}\right) \triangleq \Psi\left(\boldsymbol{x}^{(n)}\right) + \nabla\Psi\left(\boldsymbol{x}^{(n)}\right)^{\top}\left(\boldsymbol{x} - \boldsymbol{x}^{(n)}\right) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^{(n)}\|_{\boldsymbol{D}}^2 \tag{6}$$

[5]A smooth function refers to a function that is differentiable with a Lipschitz continuous gradient [19].

TABLE I
SQS METHODS

1: Initialize $\boldsymbol{x}^{(0)}$ and compute $\boldsymbol{D}$ such that (5) and (6) hold.
2: for $n = 0, 1, \cdots$
3: $\quad \boldsymbol{x}^{(n+1)} = \mathcal{P}_{\mathcal{X}} \left[\boldsymbol{x}^{(n)} - \boldsymbol{D}^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}) \right]$

for all $\boldsymbol{x}, \boldsymbol{x}^{(n)} \in \mathcal{X}$, which satisfies (5). The standard SQS algorithm [9] uses the following diagonal *majorizing* matrix:

$$\boldsymbol{D} \triangleq \operatorname{diag}\{[\boldsymbol{A}^\top \boldsymbol{W} \boldsymbol{A} + |\boldsymbol{C}|^\top \operatorname{diag}\{\ddot{\psi}_r(0)\} |\boldsymbol{C}|] \mathbf{1}\} \tag{7}$$

using the maximum curvature $\ddot{\psi}_r(0) = \max_t \ddot{\psi}_r(t)$ [41], where $|\boldsymbol{C}| \triangleq \{|c_{rj}|\} \in \mathbb{R}_+^{N_r \times N_p}$, and the vector $\mathbf{1} \in \mathbb{R}^{N_p}$ consists of N_p ones. (This $\boldsymbol{D}$ is positive definite because its diagonal entries are all positive.)

Table I gives the outline of the computationally efficient (and massively parallelizable) SQS algorithm, where the operation $\mathcal{P}_{\mathcal{X}}[\boldsymbol{x}]$ projects $\boldsymbol{x}$ onto a constraint set $\mathbb{R}_+^{N_p}$ by a (simple) element-wise clipping that replaces negative element values to zero. The sequence $\{\boldsymbol{x}^{(n)}\}$ generated from the SQS algorithm in Table I has the following convergence rate [29]:

$$\Psi\left(\boldsymbol{x}^{(n+1)}\right) - \Psi(\hat{\boldsymbol{x}}) \leq \frac{\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|_{\boldsymbol{D}}^2}{2(n+1)} \tag{8}$$

for any diagonal majorizing matrix $\boldsymbol{D}$ satisfying (5) and (6), including (7), by a simple generalization of [19, Th. 3.1]. Based on (8), the nonuniform approach [29] (see Section V-E) accelerates SQS methods by providing larger updates for the voxels that are far from their respective optima, which we use in our experiments (in Section VI).

SQS algorithms require many iterations to converge, both due to the $O(1/n)$ rate in (8) and large values in $\boldsymbol{D}$ needed for satisfying the conditions (5) and (6) in 3D X-ray CT problem. Thus we usually combine SQS algorithms with OS algorithms for faster convergence in early iterations [9], [29].

C. OS Algorithms

Iterative reconstruction requires both forward and back projection operations $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}^\top \boldsymbol{y}$ on the fly [34], [35] due to their large-scale in 3D, and thus computation of the gradient $\nabla \Psi(\boldsymbol{x}) = \boldsymbol{A}^\top \boldsymbol{W}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) + \nabla R(\boldsymbol{x})$ is very expensive. OS methods [8] accelerate gradient-based algorithms such as in Table I by grouping the projection views into M subsets evenly and using only the subset of measured data to approximate the exact gradient of the cost function.

OS methods define the subset-based cost function

$$\Psi_m(\boldsymbol{x}) \triangleq \frac{1}{2} \|\boldsymbol{y}_m - \boldsymbol{A}_m \boldsymbol{x}\|_{\boldsymbol{W}_m}^2 + \frac{1}{M} R(\boldsymbol{x}) \tag{9}$$

for $m = 0, 1, \ldots, M-1$, where $\Psi(\boldsymbol{x}) = \sum_{m=0}^{M-1} \Psi_m(\boldsymbol{x})$ and the matrices $\boldsymbol{y}_m, \boldsymbol{A}_m, \boldsymbol{W}_m$ are submatrices of $\boldsymbol{y}, \boldsymbol{A}, \boldsymbol{W}$ for the mth subset, and rely on the following "subset balance" approximation [8], [9]:

$$\nabla \Psi(\boldsymbol{x}) \approx M \nabla \Psi_0(\boldsymbol{x}) \approx \cdots \approx M \nabla \Psi_{M-1}(\boldsymbol{x}). \tag{10}$$

TABLE II
OS-SQS METHODS

1: Initialize $\boldsymbol{x}^{(0)}$ and compute $\boldsymbol{D}$.
2: for $n = 0, 1, \cdots$
3: for $m = 0, 1, \cdots, M-1$
4: $\quad k = nM + m$
5: $\quad \boldsymbol{x}^{(\frac{k+1}{M})} = \mathcal{P}_{\mathcal{X}} \left[\boldsymbol{x}^{(\frac{k}{M})} - \boldsymbol{D}^{-1} M \nabla \Psi_m(\boldsymbol{x}^{(\frac{k}{M})}) \right]$

Using (10), OS methods provide initial acceleration of about the factor of the number of subsets M in run time, by replacing $\nabla \Psi(\boldsymbol{x})$ in Table I with the approximation $M \nabla \Psi_m(\boldsymbol{x})$ that requires about $(1/M)$-times less computation, as described in Table II.

We count one iteration after all M subiterations are performed, considering the use of projection operators $\boldsymbol{A}$ and $\boldsymbol{A}^\top$ per iteration, and in practice the initial convergence rate is $O(1/(nM))$. (Using large M can slow down the algorithm in run time due to the regularizer computation [42].) OS algorithms approach a limit-cycle because (10) breaks near the optimum [43]. OS algorithms can be modified to converge to the optimum with some loss of acceleration in early iterations [28], [44].

IV. OS-SQS METHODS WITH NESTEROV'S MOMENTUM

To further accelerate OS-SQS methods, we propose to adapt two of Nesterov's momentum techniques [17], [18] that reuse previous descent directions as momentum towards the minimizer for acceleration. (One could also consider another Nesterov momentum approach [45] achieving same rate as other two [17], [18].) This section reviews both momentum approaches and combines them with OS methods.

The first momentum method [17] uses two previous iterates, while the second [18] accumulates all gradients. Without using OS methods, both Nesterov methods provide $O(1/n^2)$ convergence rate. We heuristically expect that combining momentum with OS methods will provide roughly $O(1/(nM)^2)$ rates in early iterations, by replacing n by nM. The main benefit of combining OS and Nesterov's momentum is that we have approximately M^2 times acceleration in early iterations with M subsets, yet the extra computation and memory needed (in using Nesterov's momentum approaches) are almost negligible. We discuss both proposed algorithms in more detail.

A. Proposed OS-SQS Methods With Momentum 1 (OS-mom1)

Table III illustrates the proposed combination of an OS-SQS algorithm with the momentum technique that is described in [17], where the algorithm generates two sequences $\{\boldsymbol{x}^{(k/M)}\}$ and $\{\boldsymbol{z}^{(k/M)}\}$, and line 7 of the algorithm corresponds to a momentum step with Nesterov's optimal parameter sequence t_k. Table III reduces to the ordinary OS-SQS algorithm in Table II when $t_k = 1$ for all $k \geq 0$.

The non-OS version of Table III satisfies the following convergence rate:

TABLE III
PROPOSED OS-SQS METHODS WITH MOMENTUM IN [17] (OS-MOM1)

1: Initialize $\boldsymbol{x}^{(0)} = \boldsymbol{z}^{(0)}$, $t_0 = 1$ and compute $\boldsymbol{D}$.
2: for $n = 0, 1, \cdots$
3: for $m = 0, 1, \cdots, M-1$
4: $\quad k = nM + m$
5: $\quad t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$
6: $\quad \boldsymbol{x}^{(\frac{k+1}{M})} = \mathcal{P}_{\mathcal{X}}\left[\boldsymbol{z}^{(\frac{k}{M})} - \boldsymbol{D}^{-1} M \nabla \Psi_m(\boldsymbol{z}^{(\frac{k}{M})})\right]$
7: $\quad \boldsymbol{z}^{(\frac{k+1}{M})} = \boldsymbol{x}^{(\frac{k+1}{M})} + \frac{t_k - 1}{t_{k+1}}\left(\boldsymbol{x}^{(\frac{k+1}{M})} - \boldsymbol{x}^{(\frac{k}{M})}\right)$

TABLE IV
PROPOSED OS-SQS METHODS WITH MOMENTUM IN [18] (OS-MOM2), THE NOTATION $(l)_M$ DENOTES $l \bmod M$

1: Initialize $\boldsymbol{x}^{(0)} = \boldsymbol{v}^{(0)} = \boldsymbol{z}^{(0)}$, $t_0 = 1$ and compute $\boldsymbol{D}$.
2: for $n = 0, 1, \cdots$
3: for $m = 0, 1, \cdots, M-1$
4: $\quad k = nM + m$
5: $\quad t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$
6: $\quad \boldsymbol{x}^{(\frac{k+1}{M})} = \mathcal{P}_{\mathcal{X}}\left[\boldsymbol{z}^{(\frac{k}{M})} - \boldsymbol{D}^{-1} M \nabla \Psi_m(\boldsymbol{z}^{(\frac{k}{M})})\right]$
7: $\quad \boldsymbol{v}^{(\frac{k+1}{M})} = \mathcal{P}_{\mathcal{X}}\left[\boldsymbol{z}^{(0)} - \boldsymbol{D}^{-1} \sum_{l=0}^{nM+m} t_l M \nabla \Psi_{(l)_M}(\boldsymbol{z}^{(\frac{l}{M})})\right]$
8: $\quad \boldsymbol{z}^{(\frac{k+1}{M})} = \boldsymbol{x}^{(\frac{k+1}{M})} + \frac{t_{k+1}}{\sum_{l=0}^{k+1} t_l}\left(\boldsymbol{v}^{(\frac{k+1}{M})} - \boldsymbol{x}^{(\frac{k+1}{M})}\right)$

Lemma 1: For $n \geq 0$, the sequence $\{\boldsymbol{x}^{(n)}\}$ generated by the non-OS version ($M = 1$) of Table III satisfies

$$\Psi\left(\boldsymbol{x}^{(n+1)}\right) - \Psi(\hat{\boldsymbol{x}}) \leq \frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|_{\boldsymbol{D}}^2}{(n+1)(n+2)} \tag{11}$$

where $\boldsymbol{D}$ is a diagonal majorizing matrix satisfying (5) and (6), such as (7).

Inequality (11) is a simple generalization of [19, Th. 4.4]. In practice, we expect the initial rate of OS-mom1 for $M > 1$ to be $O(1/(nM)^2)$ with the approximation (10), which is the main benefit of this approach, while the computation cost remains almost the same as that of OS-SQS algorithm in Table II. The only slight drawback of Table III over Table II is the extra memory needed to store the image $\boldsymbol{z}^{(k/M)}$.

B. Proposed OS-SQS Methods With Momentum 2 (OS-mom2)

Table IV summarizes the second proposed OS-SQS algorithm with momentum. This second method, OS-mom2, is based on [18] and uses accumulation of the past (subset) gradients in line 7. Rather than using the original choice of coefficient $t_k = (k+1)/2$ from [18], Table IV uses the t_k from [46] and [47] that gives faster convergence. Both $\boldsymbol{x}^{(k/M)}$ and $\boldsymbol{v}^{(k/M)}$ in Table IV lie in the set $\mathcal{X}$ (e.g., are nonnegative) because of the projection operation $\mathcal{P}_{\mathcal{X}}[\cdot]$. Furthermore, $\boldsymbol{z}^{(k/M)}$ is a convex combination of $\boldsymbol{x}^{(k/M)}$ and $\boldsymbol{v}^{(k/M)}$ and thus also lies in $\mathcal{X}$. This may improve the stability of OS-mom2 over OS-mom1 that lacks this property.

The sequence $\{\boldsymbol{x}^{(n)}\}$ generated by Table IV with $M = 1$ can be proven to satisfy (11), by generalizing [18, Th. 2]. While the one-subset ($M = 1$) version of Table IV provides $O(1/n^2)$, we heuristically expect from (10) for the OS version to have the rate $O(1/(nM)^2)$ in early iterations. (The convergence analysis of this algorithm in Table IV is discussed stochastically in the next section.) Compared to Table III, one additional $\mathcal{P}_{\mathcal{X}}[\cdot]$ operation per iteration and extra arithmetic operations are required in Table IV, but those are negligible.

Overall, the two proposed algorithms in Tables III and IV are expected to provide fast convergence rate $O(1/(nM)^2)$ in early iterations, which we confirm empirically in Section VI. However, the type of momentum affects the overall convergence when combined with OS. Also, the convergence behavior of two algorithms is affected by the number and ordering of subsets, as discussed in Section VI.

The proposed OS-momentum algorithms in Tables III and IV become unstable in some cases when M is too large, as predicted by the convergence analysis in [33]. To stabilize the algorithms, the next section proposes to adapt a recent relaxation scheme [33] developed for stochastic gradient methods with momentum.

V. RELAXATION OF MOMENTUM

This section relates the OS-SQS algorithm to *diagonally preconditioned* stochastic gradient methods and adapts a relaxation scheme designed for stochastic gradient algorithms with momentum. Then we investigate various choices of relaxation[6] to achieve overall fast convergence.

A. Stochastic Gradient Method

If one uses random subset orders, then one can view OS methods as stochastic gradient methods by defining $M \nabla \Psi_{S_k}(\boldsymbol{x})$ as a stochastic estimate of $\nabla \Psi(\boldsymbol{x})$, where a random variable S_k at kth iteration is uniformly chosen from $\{0, 1, \ldots, M-1\}$. In this stochastic setting, OS-SQS methods satisfy

$$\begin{cases} \mathbb{E}\left[M \Psi_{S_k}(\boldsymbol{x})\right] = \Psi(\boldsymbol{x}) \\ \mathbb{E}\left[M \nabla \Psi_{S_k}(\boldsymbol{x})\right] = \nabla \Psi(\boldsymbol{x}) \\ \mathbb{E}\left[\left(M \nabla_j \Psi_{S_k}(\boldsymbol{x}) - \nabla_j \Psi(\boldsymbol{x})\right)^2\right] \leq \sigma_j^2, \ \forall j \end{cases} \tag{12}$$

for all $\boldsymbol{x} \in \mathcal{B}$, for some finite constants $\{\sigma_j\}$, where $\mathbb{E}$ is the expectation operator over the random selection of S_k, $\nabla_j \triangleq \partial / \partial x_j$, and $\mathcal{B}$ is a bounded feasible set that includes $\hat{\boldsymbol{x}}$. The feasible set $\mathcal{B}$ can be derived based on the measurement data $\boldsymbol{y}$ [44, Sec. A.2], and we assume that the sequences generated by the algorithms are within the set $\mathcal{B}$. The last inequality in (12) is a generalized version of [33, eq. (2.5)] for (diagonally preconditioned) OS-SQS-type algorithms. The vector $\boldsymbol{\sigma} \triangleq \{\sigma_j\}$ has smaller values if we use smaller M or the subsets are balanced as (10). However, estimating the value of $\boldsymbol{\sigma}$:

$$\sigma_j \triangleq \max_{x \in \mathcal{B}} \tilde{\sigma}_j(\boldsymbol{x}) \tag{13}$$

where

$$\begin{aligned} \tilde{\sigma}_j^2(\boldsymbol{x}) &\triangleq \mathbb{E}\left[\left(M \nabla_j \Psi_{S_k}(\boldsymbol{x}) - \nabla_j \Psi(\boldsymbol{x})\right)^2\right] \\ &= M \sum_{m=0}^{M-1} \left[\boldsymbol{A}_m^\top \boldsymbol{W}_m (\boldsymbol{A}_m \boldsymbol{x} - \boldsymbol{y}_m)\right]_j^2 \\ &\quad - \left[\boldsymbol{A}^\top \boldsymbol{W}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})\right]_j^2 \end{aligned} \tag{14}$$

[6]The relaxation scheme involves couple of parameters and we provide a table of notations in the supplementary material to improve readability.

TABLE V
PROPOSED STOCHASTIC OS-SQS ALGORITHMS WITH MOMENTUM (OS-MOM3). ξ_k IS A REALIZATION OF A RANDOM VARIABLE S_k

1: Initialize $\boldsymbol{x}^{(0)} = \boldsymbol{v}^{(0)} = \boldsymbol{z}^{(0)}$, $t_0 \in (0,\ 1]$ and compute $\boldsymbol{D}$.
2: for $n = 0, 1, \cdots$
3: for $m = 0, 1, \cdots, M-1$
4: $k = nM + m$
5: Choose $\boldsymbol{\Gamma}^{(k)} \triangleq \mathsf{diag}\left\{\gamma_j^{(k)}\right\}$ s.t. $\begin{cases} \boldsymbol{\Gamma}^{(0)} \succ \boldsymbol{D}, & k = 0 \\ \boldsymbol{\Gamma}^{(k)} \succeq \boldsymbol{\Gamma}^{(k-1)}, & k > 0 \end{cases}$
6: Choose t_{k+1} s.t. $t_{k+1}^2 \boldsymbol{\Gamma}^{(k+1)} \preceq \left(\sum_{l=0}^{k+1} t_l\right) \boldsymbol{\Gamma}^{(k)}$
7: $\boldsymbol{x}^{(\frac{k+1}{M})} = \mathcal{P}_{\mathcal{X}}\left[\boldsymbol{z}^{(\frac{k}{M})} - \left[\boldsymbol{\Gamma}^{(k)}\right]^{-1} M \nabla \Psi_{\xi_k}(\boldsymbol{z}^{(\frac{k}{M})})\right]$
8: $\boldsymbol{v}^{(\frac{k+1}{M})} = \mathcal{P}_{\mathcal{X}}\left[\boldsymbol{z}^{(0)} - \left[\boldsymbol{\Gamma}^{(k)}\right]^{-1} \sum_{l=0}^{k} t_l M \nabla \Psi_{\xi_l}(\boldsymbol{z}^{(\frac{l}{M})})\right]$
9: $\boldsymbol{z}^{(\frac{k+1}{M})} = \boldsymbol{x}^{(\frac{k+1}{M})} + \frac{t_{k+1}}{\sum_{l=0}^{k+1} t_l}\left(\boldsymbol{v}^{(\frac{k+1}{M})} - \boldsymbol{x}^{(\frac{k+1}{M})}\right)$

appears to be impractical, so Section V-F provides a practical approach for approximating $\boldsymbol{\sigma}$.

B. Proposed OS-SQS Methods With Relaxed Momentum (OS-mom3)

Inspired by [33], Table V describes a generalized version of OS-SQS-momentum methods that reduces to OS-mom2 if one uses a deterministic subset ordering $S_k = (k \bmod M)$ and a fixed majorizing diagonal matrix

$$\boldsymbol{\Gamma}^{(k)} = \boldsymbol{D}. \tag{15}$$

For $M = 1$, the algorithm with these choices satisfies (11) with $n = k$. However, for $M > 1$, the analysis in [33] illustrates that using the choice (15) leads to the following:

$$\mathbb{E}\left[\Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - \Psi(\hat{\boldsymbol{x}})\right] \leq \frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|_{\boldsymbol{D}}^2}{(k+1)(k+2)} + \frac{(k+3)\boldsymbol{p}^\top \boldsymbol{\sigma}}{3} \tag{16}$$

for $k \geq 0$, where $\boldsymbol{p} \triangleq \{p_j \triangleq \max_{\boldsymbol{x},\bar{\boldsymbol{x}} \in \mathcal{B}} |x_j - \bar{x}_j|\}$ measures the diameter of the feasible set $\mathcal{B}$. This expression reveals that OS methods ($M > 1$) with momentum may suffer from error accumulation due to the last term in (16) that depends on the error bounds ($\boldsymbol{\sigma}$) in (12). To improve stability for the case $M > 1$, we would like to find a way to decrease this last term. Using a larger constant denominator, i.e., $\boldsymbol{\Gamma}^{(k)} = q\boldsymbol{D}$ for $q > 1$, would slow the accumulation of error but would not prevent eventual accumulation of error [33].

To stabilize the algorithm for $M > 1$, we adapt the *relaxed* momentum approach in [33] as described in Table V with appropriately selected $\boldsymbol{\Gamma}^{(k)}$ and t_k satisfying the conditions in lines 5 and 6 in Table V. Then, the algorithm in Table V satisfies the following convergence rate.

Lemma 2: For $k = nM + m \geq 0$, the sequence $\{\boldsymbol{x}^{((k+1)/M)}\}$ generated by Table V satisfies

$$\mathbb{E}\left[\Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - \Psi(\hat{\boldsymbol{x}})\right] \leq \frac{1}{\sum_{l=0}^{k} t_l}\left[\frac{\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|_{\boldsymbol{\Gamma}^{(k)}}^2}{2} + \sum_{l=0}^{k}\sum_{i=0}^{l} t_i \|\boldsymbol{\sigma}\|_{(\boldsymbol{\Gamma}^{(l)} - \boldsymbol{D})^{-1}}^2\right]. \tag{17}$$

Proof: See Appendix A. ∎

Lemma 2 shows that increasing $\boldsymbol{\Gamma}^{(k)}$ can help prevent accumulation of error $\boldsymbol{\sigma}$. Next we discuss the selection of parameters $\boldsymbol{\Gamma}^{(k)}$ and t_k.

C. The Choice of $\boldsymbol{\Gamma}^{(k)}$ and t_k

For any given $\boldsymbol{\Gamma}^{(k)} \triangleq \mathrm{diag}\{\gamma_j^{(k)}\}$, we use $t_0 = 1$ and the following rule:

$$t_{k+1} = \frac{1}{2\alpha_{k+1}}\left(1 + \sqrt{1 + 4t_k^2 \alpha_k \alpha_{k+1}}\right) \tag{18}$$

for all $k \geq 0$, where $\alpha_{k+1} \triangleq \max_j(\gamma_j^{(k+1)}/\gamma_j^{(k)})$ and $\alpha_0 = 1$. The choice (18) increases the fastest among all possible choices satisfying the condition in line 6 of Table V (see the proof in Appendix B).[7]

In this paper, we focus on the choice

$$\boldsymbol{\Gamma}^{(k)} = \boldsymbol{D} + (k+2)^{c_k}\boldsymbol{\Gamma} \tag{19}$$

for a nondecreasing $c_k \geq 0$ and a fixed diagonal matrix $\boldsymbol{\Gamma} \triangleq \mathrm{diag}\{\gamma_j\} \succ 0$. The choice (19) generalizes [33], enabling more flexibility in the choice of c_k. We leave other formulations of $\boldsymbol{\Gamma}^{(k)}$ that may provide better convergence as future work.

For $\boldsymbol{\Gamma}^{(k)}$ in (19), computing α_k in (18) becomes

$$\alpha_{k+1} = 1 + \frac{(k+3)^{c_{k+1}} - (k+2)^{c_k}}{\min_j(d_j/\gamma_j) + (k+2)^{c_k}} \geq 1. \tag{20}$$

Overall, the computational cost of Table V with the choices (18) and (19) remains similar to that of Table IV. Using (18) and (19), the proposed algorithm in Table V achieves the following inequality:

Corollary 1: For $k = nM + m \geq 0$, the sequence $\{\boldsymbol{x}^{((k+1)/M)}\}$ generated by Table V with the coefficients (18) and (19) satisfies

$$\mathbb{E}\left[\Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - \Psi(\hat{\boldsymbol{x}})\right] \leq \left(\max_{0\leq l\leq k}\sqrt{\alpha_l}\right)\left[\frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|_{\boldsymbol{D}}^2}{(k+1)(k+2)} + \frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|_{\boldsymbol{\Gamma}}^2}{(k+1)(k+2)^{1-c_k}} + \frac{2\sum_{l=0}^{k}(l+2)^{2-c_l}\|\boldsymbol{\sigma}\|_{\boldsymbol{\Gamma}^{-1}}^2}{(k+1)(k+2)}\right]. \tag{21}$$

Proof: Use Lemma 2 and the inequality conditions of the sequence $\{t_k\}$ in (18)

$$\frac{k+1}{2\sqrt{\alpha_k}} \leq t_k \leq \frac{k+1}{\sqrt{\alpha_k}},$$

$$\min_{0\leq l\leq k}\frac{(k+1)(k+2)}{4\sqrt{\alpha_l}} \leq \sum_{l=0}^{k} t_l \leq \max_{0\leq l\leq k}\frac{(k+1)(k+2)}{2\sqrt{\alpha_l}}$$

for $k \geq 0$, which can be easily proven by induction. ∎

There are two parameters c_k and $\boldsymbol{\Gamma}$ to be tuned in (19). Based on Corollary 1, the next two subsections explore how these parameters affect convergence rate. (We made a preliminary investigation of these two parameters in [48].)

D. The Choice of c_k

In Corollary 1, the choice of c_k controls the overall convergence rate. We first consider a constant $c_k = c$.

[7]The coefficient (18) increases faster than the choice $t_{k+1} = (k+1)/2^c$ for a constant $c_k = c \geq 1$ used in [33], so we use the choice (18) that leads to faster convergence based on Lemma 2.

Corollary 2: For $k = nM+m \geq 0$ and a fixed constant $c_k = c \in [0, 2]$, the sequence $\{\boldsymbol{x}^{((k+1)/M)}\}$ generated by Table V with the coefficients (18) and (19) satisfies

$$\mathbb{E}\left[\Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - \Psi(\hat{\boldsymbol{x}})\right] \leq \left(\max_{0\leq l\leq k} \sqrt{\alpha_l}\right)\left[\frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{D}}}{(k+1)(k+2)} + \frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{\Gamma}}}{(k+1)(k+2)^{1-c}} + \frac{2(k+3)^{3-c}\|\boldsymbol{\sigma}\|^2_{\boldsymbol{\Gamma}^{-1}}}{(3-c)(k+1)(k+2)}\right]. \quad (22)$$

Proof: Use the derivation in [33, Sec. 6.3] using

$$\sum_{l=0}^{k}(l+2)^{2-c} \leq \int_1^{k+1}(s+2)^{2-c}\,\mathrm{d}s \leq \frac{(k+3)^{3-c}}{3-c}.$$

■

In Corollary 2, the choice $c_k = 1.5$ provides the rate

$$\mathbb{E}\left[\Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - \Psi(\hat{\boldsymbol{x}})\right] \leq \left(\max_{\substack{0\leq l\leq k\\ \forall j}} \sqrt{\frac{d_j + (l+2)^{1.5}\gamma_j}{d_j + (l+1)^{1.5}\gamma_j}}\right) \times O\left(\frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{D}}}{k^2} + \frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{\Gamma}}}{\sqrt{k}} + \frac{2\|\boldsymbol{\sigma}\|^2_{\boldsymbol{\Gamma}^{-1}}}{1.5\sqrt{k}}\right) \quad (23)$$

achieving, on average, the optimal rate $O(1/\sqrt{k})$ in the presence of stochastic noise[8]. Corollary 2 shows that using $c \leq 1$ will suffer from error accumulation, using $1 < c < 1.5$ might provide faster initial convergence than $c = 1.5$ but does not achieve the optimal rate, and $c > 1.5$ will cause slower overall convergence rate than $c = 1.5$. So, we prefer $1 < c \leq 1.5$, for which we expect to eventually reach smaller cost function values than the choice $\boldsymbol{\Gamma}^{(k)} = \boldsymbol{D}$ (or $\boldsymbol{\Gamma} = 0$) in (15), since we prevent the accumulation of error from OS methods by increasing the denominator $\boldsymbol{\Gamma}^{(k)}$ as (19). In other words, the algorithm with $M > 1$ and (19) is slower than the choice of (15) initially, but eventually becomes faster and reaches the optimum on average.

In light of these trade-offs, we further consider using an increasing sequence c_k that is upper-bounded by 1.5, hoping to provide fast overall convergence rate. We investigated the following choice:

$$c_k = 1 + 0.5\left(1 - \frac{\eta}{k+\eta}\right) \quad (24)$$

for $k \geq 0$ with a parameter $\eta > 0$. This choice of c_k balances between fast initial acceleration and prevention of error accumulation. In other words, this increasing c_k can provide faster initial convergence than a constant $c_k = 1.5$ (or $\eta = 0$), yet guarantees the optimal (asymptotic) rate $O(1/\sqrt{k})$ as the case $c_k = 1.5$, based on Corollary 1. We leave further optimization of c_k as future work.

E. The Choice of $\boldsymbol{D}$ and $\boldsymbol{\Gamma}$

To optimize the choice of $\boldsymbol{\Gamma}^{(k)}$ in (19), we would like to minimize the upper bound on the right-hand side (RHS) of (22) with respect to both $\boldsymbol{D}$ and $\boldsymbol{\Gamma}$, where we consider a fixed $c_k = c$ for simplicity. In nonuniform SQS [29], to accelerate algorithms in Tables I and II, we suggested to use $\boldsymbol{D}$ that minimizes the RHS of (8) among all possible choices of $\boldsymbol{D}$ generated by a (general) SQS technique [9], [29] (details are omitted) and thus satisfies (5) and (6). Similarly in our proposed methods, we use the following diagonal majorizing matrix $\hat{\boldsymbol{D}}$ that minimizes the RHS of (22):

$$\hat{\boldsymbol{D}} \triangleq \underset{\substack{\boldsymbol{D}\succ 0,\\ \boldsymbol{D}\text{ is diagonal and it is}\\ \text{generated by SQS [9],[29].}}}{\arg\min} \left\{\frac{2\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{D}}}{(k+1)(k+2)}\right\} = \mathrm{diag}\{\mathrm{diag}\{\hat{\boldsymbol{u}}\}^{-1}[\boldsymbol{A}^\top \boldsymbol{W}\boldsymbol{A} + |\boldsymbol{C}|^\top \mathrm{diag}\{\ddot{\psi}_r(0)\}|\boldsymbol{C}|]\hat{\boldsymbol{u}}\} \quad (25)$$

instead of (7), where the vector $\hat{\boldsymbol{u}}$ is defined as

$$\hat{\boldsymbol{u}} = \left\{\hat{u}_j \triangleq |x_j^{(0)} - \hat{x}_j|\right\} \in \mathbb{R}^{N_p}_+. \quad (26)$$

To choose $\boldsymbol{\Gamma}$, we also minimize the upper bound on the RHS of (22) with respect to $\boldsymbol{\Gamma}$. For simplicity in designing $\boldsymbol{\Gamma}$, we ignore the $\max_{0\leq l\leq k}\sqrt{\alpha_l}$ term, and we ignore the "+1" and "+2" factors added to k. The optimal $\boldsymbol{\Gamma}$ for k (sub)iterations is

$$\hat{\boldsymbol{\Gamma}}_c(k) \triangleq \underset{\substack{\boldsymbol{\Gamma}\succ 0,\\ \boldsymbol{\Gamma}\text{ is diagonal.}}}{\arg\min} \left\{\frac{\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{\Gamma}}}{k^{2-c}} + \frac{\|\boldsymbol{\sigma}\|^2_{\boldsymbol{\Gamma}^{-1}}}{(3-c)k^{c-1}}\right\} = \mathrm{diag}\left\{\frac{\sigma_j}{\sqrt{3-c}\,\hat{u}_j}k^{1.5-c}\right\}. \quad (27)$$

It is usually undesirable to have to select the (sub)iteration k before iterating. The choice $c = 1.5$ cancels out the parameter k in (27) leading to the k-independent choice

$$\hat{\boldsymbol{\Gamma}}_{1.5} = \mathrm{diag}\left\{\frac{\sigma_j}{\sqrt{1.5}\hat{u}_j}\right\}. \quad (28)$$

Since we prefer the choice of c_k that eventually becomes 1.5 for the optimal rate, we focus on the choice $\hat{\boldsymbol{\Gamma}}_{1.5}$ in (28).

F. The Choice of $\boldsymbol{\sigma}$ and $\hat{\boldsymbol{u}}$

The optimized $\hat{\boldsymbol{D}}$ (25) and $\hat{\boldsymbol{\Gamma}}_{1.5}$ (28) rely on unavailable parameters $\{\sigma_j\}$ (13) and $\{\hat{u}_j\}$ (26), so we provide a practical approach to estimate them, which we used in Section VI. In practice, the sequences in Table V visit only a part of the feasible set $\mathcal{B}$, so it would be preferable to compute $\tilde{\sigma}_j(\boldsymbol{x})$ in (14) within such part of $\mathcal{B}$ for estimating σ_j, but even that is impractical. Instead, we use $\tilde{\sigma}_j(\boldsymbol{x}^{(0)})$ as a practical approximation of σ_j, which is computationally efficient. This quantity measures the variance of the stochastic estimate of the gradient at the initial image $\boldsymbol{x}^{(0)}$, and depends on the grouping and number of subsets. This estimate of σ_j may be sensitive to the choice of $\boldsymbol{x}^{(0)}$, and we leave further investigation as future work.

To save computation, we evaluate $\tilde{\sigma}_j(\boldsymbol{x}^{(0)})$ simultaneously with the computation of $\boldsymbol{D}$ in (7) or (25) using modified projectors $\boldsymbol{A}$ and $\boldsymbol{A}^\top$ (see [29, Sec. III-F]) that handle two inputs.

We further approximate $\hat{u}_j$ (26) by

$$\hat{u}_j \approx \zeta \bar{u}_j \quad (29)$$

for $\hat{\boldsymbol{\Gamma}}_{1.5}$ in (28), where $\zeta > 0$ is an (unknown) constant, and a vector $\bar{\boldsymbol{u}} \in \mathbb{R}^{N_p}_+$ is a (normalized) approximation of $\hat{u}_j$,

[8]Stochastic gradient algorithms (using only first-order information) cannot decrease the stochastic noise faster than the rate $O(1/\sqrt{k})$ [49], and the proposed relaxation scheme achieves this optimal rate [33].

which is computed by applying an edge-detector to the filtered back-projection (FBP) image that is used for the initial $\boldsymbol{x}^{(0)}$ as described in [29, Sec. III-E].[9] In low-dose clinical CT, the root mean squared difference (RMSD[10]) within the region-of-interest (ROI) between the initial FBP image $\boldsymbol{x}^{(0)}$ and the optimal image $\hat{\boldsymbol{x}}$ is about 30 [HU], i.e., $\|\hat{\boldsymbol{u}}_{\mathrm{ROI}}\|/\sqrt{N_{p,\mathrm{ROI}}} \approx 30$ [HU], so we let $\zeta = 30$ [HU] in (29) as a reasonable choice in practice[11] where $\|\bar{\boldsymbol{u}}_{\mathrm{ROI}}\|/\sqrt{N_{p,\mathrm{ROI}}} = 1$. Then, our final practical choice of $\boldsymbol{\Gamma}$ becomes

$$\hat{\boldsymbol{\Gamma}}_{1.5} \approx \mathrm{diag}\left\{ \frac{\tilde{\sigma}_j\left(\boldsymbol{x}^{(0)}\right)}{\sqrt{1.5}\zeta \bar{\boldsymbol{u}}_j} \right\}. \tag{30}$$

VI. Results

We investigate the convergence rate of the three proposed OS-momentum algorithms in Tables III, IV, and V, namely OS-mom1, OS-mom2 and OS-mom3 in this section, for PWLS reconstruction of simulated and real 3D CT scans. We implemented[12] the proposed algorithm in C and ran on a machine with two 2.27 GHz 10-core Intel Xeon E7-8860 processors using 40 threads.[13]

We use an edge-preserving potential function $\psi_r(\,\cdot\,)$ in [29, eq. (45)] with $a = 0.0558$, $b = 1.6395$, and $\delta = 10$

$$\psi(t) = \frac{\delta^2}{b^3}\left(\frac{ab^2}{2}\left|\frac{t}{\delta}\right|^2 + b(b-a)\left|\frac{t}{\delta}\right| + (a-b)\log\left(1 + b\left|\frac{t}{\delta}\right|\right)\right).$$

For simulation data, the spatial weighting β_r was chosen empirically to be

$$\beta_r \triangleq 50 \prod_{\forall j, c_{rj} \neq 0} \max\{\kappa_j, 0.01\kappa_{\max}\} \tag{31}$$

for uniform resolution properties [37], where $\kappa_j \triangleq \sqrt{(\sum_{i=1}^{N_d} a_{ij} w_i)/(\sum_{i=1}^{N_d} a_{ij})}$ and $\kappa_{\max} \triangleq \max_j \kappa_j$. We emulated β_r of the GE product "Veo" for the patient 3D CT scans. We use a diagonal majorizing matrix $\hat{\boldsymbol{D}}$ in (25) using the nonuniform approach [29] with (29) for SQS methods. We investigated 12, 24, and 48 subsets for OS algorithms.

We first use simulated data to analyze the factors that affect the stability of the proposed OS-momentum algorithms, and further study the relaxation scheme for the algorithms. Then we verify the convergence speed of the proposed algorithm using real 3D CT scans. We computed the RMSD between the current and converged[14] image within the ROI versus computation time for 30 iterations (n) to measure the convergence rate.[15]

A. Simulation Data

We simulated a $888 \times 64 \times 2934$ sinogram (number of detector columns $\times$ detector rows $\times$ projection views) from a $1024 \times 1024 \times 154$ XCAT phantom [51] scanned in a helical geometry with pitch 1.0 (see Fig. 1). We reconstructed $512 \times 512 \times 154$ images with an initial FBP image $\boldsymbol{x}^{(0)}$ in Fig. 1(a) using a (simple) single-slice rebinning method [52]. Fig. 2 shows that SQS-Nesterov's momentum methods without OS algorithms do not accelerate SQS much. OS algorithm itself can accelerate the SQS algorithm better than Nesterov's momentum. Our proposed OS-momentum algorithms rapidly decrease RMSD in early iterations (disregarding the diverging curves that we address shortly). However, the convergence behavior of OS-momentum algorithm depends on several factors such as the number of subsets, the order of subsets, and the type of momentum techniques. Thus, we discuss these in more detail based on the results in Fig. 2, and suggest ways to improve the stability of the algorithm while preserving the fast convergence rate.

1) The Number of Subsets: Intuitively, using more subsets will provide faster initial convergence but will increase instability due to errors between the full gradient and subset gradient. Also, performing many subiterations (m) can increase error accumulation per outer iteration (n). Fig. 2 confirms this behavior.

2) The Ordering of Subsets: Interestingly, the results of the proposed algorithms depend greatly on the subset ordering. Fig. 2 focuses on two deterministic orders: a sequential (OSs) order, and a bit-reversal (OSb) order [53] that selects each order-adjacent subsets to have their projection views to be far apart as described in Table VI. The ordering greatly affects the build-up of momentum in OS-momentum algorithms, whereas ordering is less important for ordinary OS methods as seen in Fig. 2. The bit-reversal order provided much better stability in Fig. 2(b) compared to the results in Fig. 2(a). Apparently, the bit-reversal order can cancel out some gradient errors, because successive updates are likely to have opposite directions due to its subset ordering rule. In contrast, the sequential ordering has high correlation between the updates from two adjacent subsets, increasing error accumulation through momentum. Therefore, we recommend using the bit-reversal order. [Fig. 3(d) shows that random ordering (OSr) performed worse than the bit-reversal order.]

3) Type of Momentum: Fig. 2 shows that combining OS with two of Nesterov's momentum techniques in Tables III and IV (OS-mom1 and OS-mom2) resulted in different behaviors, whereas the one-subset versions of them behaved almost the same. Fig. 2 shows that the OS-mom2 algorithm is more stable than the OS-mom1 algorithm perhaps due to the different formulation of momentum or the fact that the momentum term

[9]We provide convergence results from using the oracle $\hat{\boldsymbol{u}}$, compared to its approximate $\zeta\bar{\boldsymbol{u}}$, in the supplementary material.

[10]$\mathrm{RMSD}_{\mathrm{ROI}}(\boldsymbol{x}^{(n)}) \triangleq \|\boldsymbol{x}_{\mathrm{ROI}}^{(n)} - \hat{\boldsymbol{x}}_{\mathrm{ROI}}\|/\sqrt{N_{p,\mathrm{ROI}}}$ [HU], where $N_{p,\mathrm{ROI}}$ is the number of voxels within the ROI.

[11]This choice worked well in our experiments, but may depend on the initial image, the cost function and the measurements, so improving the choice of ζ is future work.

[12]The matlab code of the proposed OS-momentum methods will be available through the last author's toolbox [50].

[13]Our implementation and choice of platform are likely to be suboptimal, and further exploiting the massively parallelizable nature of the proposed algorithms will provide additional speedup in run time, which we leave as future work.

[14]We ran thousands of iterations of (convergent) SQS algorithm to generate (almost) converged images $\hat{\boldsymbol{x}}$.

[15]Even though the convergence analysis in Section V is based on the cost function, we plot RMSD rather than the cost function because RMSD is more informative (see [29, Supplementary material]).

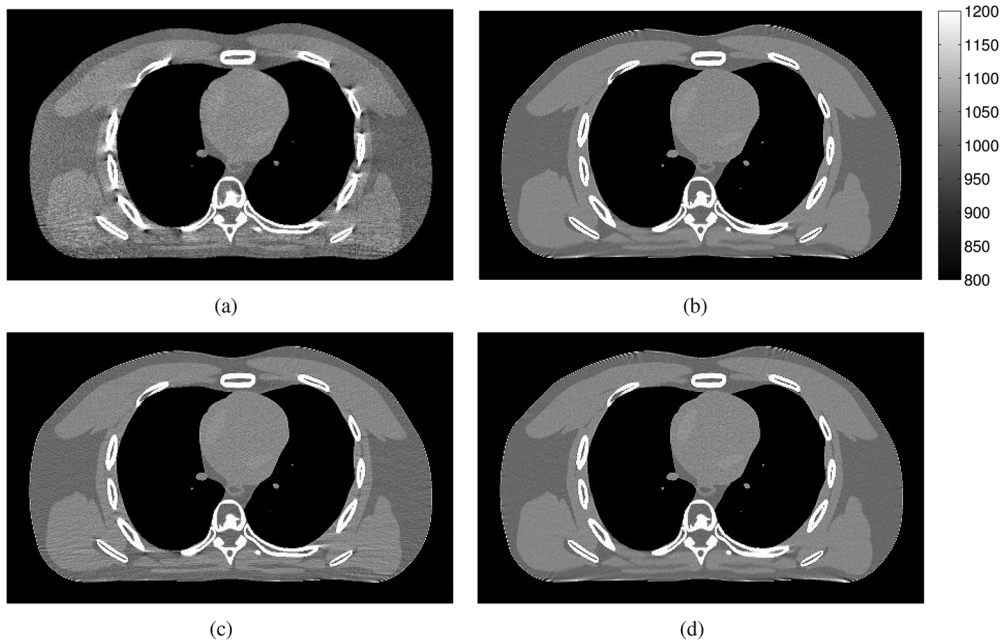

Fig. 1. Simulation data: a transaxial plane of (a) an initial FBP image $\boldsymbol{x}^{(0)}$, (b) a converged image $\hat{\boldsymbol{x}}$, and two reconstructed images $\boldsymbol{x}^{(15)}$ after 15 iterations (about 680 s) of (c) OSb(24) and (d) OSb(24)-mom2. (Images are cropped for better visualization.)

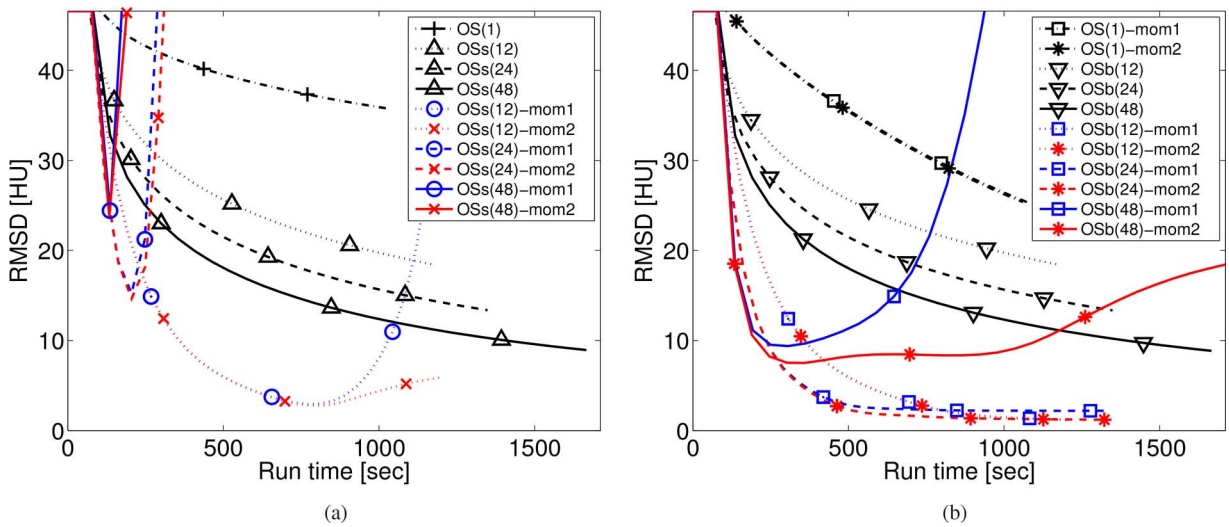

Fig. 2. Simulation data: convergence rate of OS algorithms (1, 12, 24, and 48 subsets) for 30 iterations with and without momentum for (a) sequential order and (b) bit-reversal order in Table VI. (The first iteration counts the precomputation of the denominator $\boldsymbol{D}$ in (7), and thus there are no changes during the first iteration.)

$\{\boldsymbol{z}^{(k/M)}\}$ in Table III is not guaranteed to stay within the set $\mathcal{X}$ unlike that in Table IV. Therefore, we recommend using the OS-mom2 algorithm in Table IV for better stability.

Fig. 1 shows the initial image, the converged image and reconstructed images after 15 iterations (about 680 s) of conventional OS and our proposed OS-momentum algorithm with 24 subsets and the bit-reversal ordering. The OSb(24)-mom2 reconstructed image is very similar to the converged image after 15 iterations while that of conventional OS is still noticeably different. However, even the more stable OS-mom2 algorithm becomes unstable eventually for many subsets $(M > 24)$ as seen in Fig. 2; the next subsection shows how relaxation improves stability.

4) The Choice of $\boldsymbol{\Gamma}$*:* Section V-E gives an optimized $\boldsymbol{\Gamma}$ in (19) that minimizes the right term of (22), i.e., the gradient error term, on average. However, since the right term in (22) is a worst-case loose upper-bound, we can afford to use smaller $\boldsymbol{\Gamma}$ than $\hat{\boldsymbol{\Gamma}}_{1.5}$ in (28) [or (30)]. In addition, we may use even smaller $\boldsymbol{\Gamma}$ depending on the order of subsets. Specifically, the bit-reversal ordering (OSb) appears to accumulate less gradient error than other orderings, including random subset orders (OSr), so the choice (28) [or (30)] may be too conservative. Therefore, we investigated decreasing the matrix $\hat{\boldsymbol{\Gamma}}_{1.5}$ (28) [or (30)] by a factor $\lambda \in (0, 1]$ as

$$\tilde{\boldsymbol{\Gamma}} = \lambda \hat{\boldsymbol{\Gamma}}_{1.5}. \tag{32}$$

Fig. 3 shows the effect of the parameter λ for various choices of the number and ordering of subsets. In all cases, $\lambda = 1$ is too conservative and yields very slow convergence. Smaller

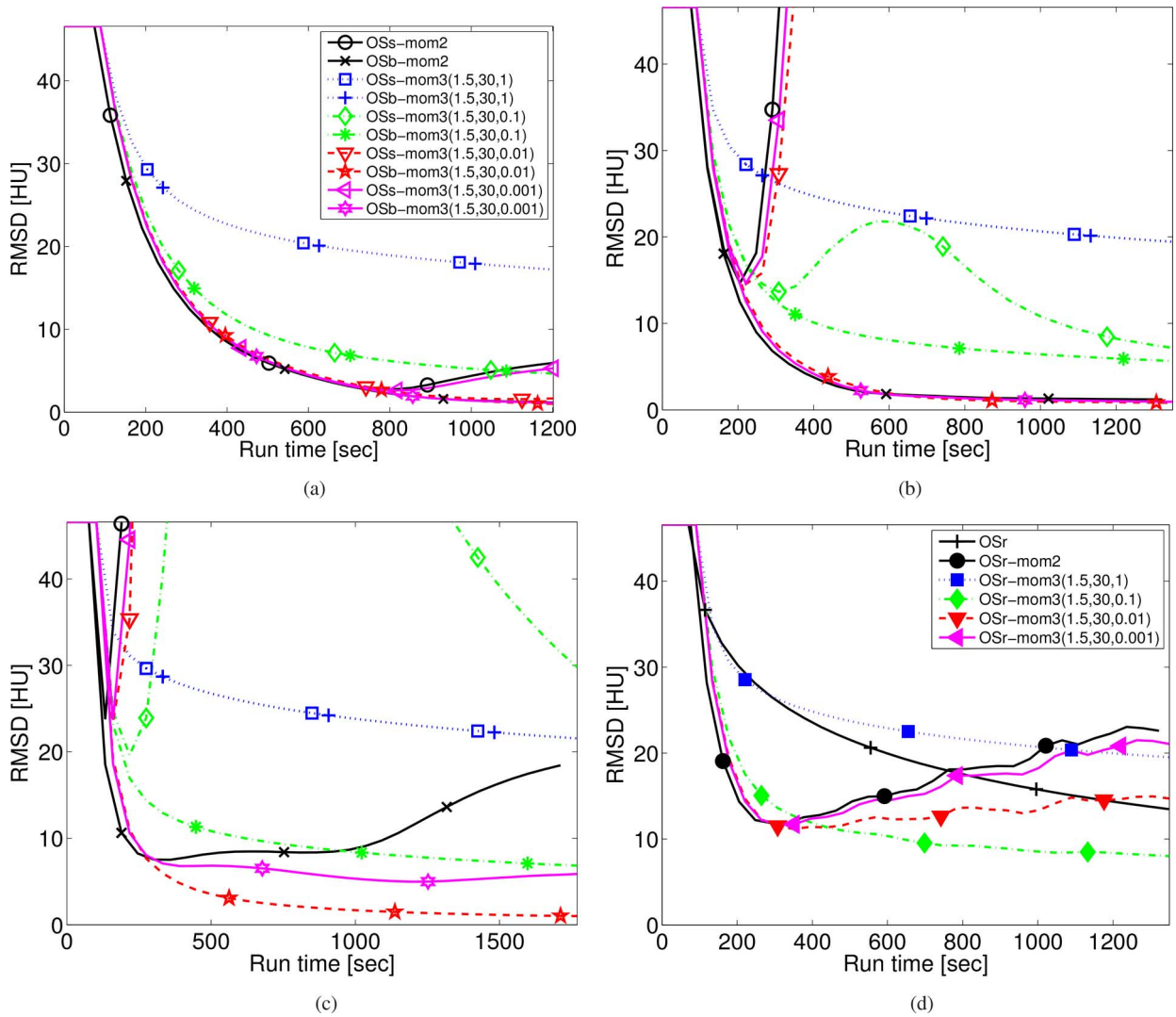

Fig. 3. Simulation data: convergence rate for various choices of the parameter λ in relaxation scheme of OS-momentum algorithms (c, ζ, λ) for (a) 12, (b) 24, (c) 48 subsets with both sequential (OSs) and bit-reversal (OSb) subset orderings in Table VI for 30 iterations. [The plot (b) and (c) share the legend of (a).] The averaged plot of five realizations of random subset ordering (OSr) is illustrated in (d) for 24 subsets.

TABLE VI
EXAMPLES OF SUBSET ORDERINGS: TWO DETERMINISTIC SUBSET ORDERING (OSS, OSB) AND ONE INSTANCE OF RANDOM ORDERING (OSR) FOR OS METHODS WITH $M = 8$ SUBSETS IN A SIMPLE GEOMETRY WITH 24 PROJECTION VIEWS DENOTED AS $(p_0, p_1, \ldots, p_{23})$, WHERE THOSE ARE REASONABLY GROUPED INTO THE FOLLOWING EIGHT SUBSETS: $S_0 = (p_0, p_8, p_{16})$, $S_1 = (p_1, p_9, p_{17}), \ldots, S_7 = (p_7, p_{15}, p_{23})$

Sequential (OSs):	$(S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7), (S_0, S_1, \cdots$
Bit-reversal (OSb):	$(S_0, S_4, S_2, S_6, S_1, S_5, S_3, S_7), (S_0, S_4, \cdots$
Random (OSr):	$S_6, S_7, S_1, S_7, S_5, S_0, S_2, S_4, S_7, S_3, \cdots$

$\lambda \leq 0.1$ values lead to faster convergence, but it failed to stabilize the case of sequential ordering for $M > 24$. However, $\lambda = 0.01$ worked well for the bit-reversal orderings in the simulation data, while the choice $\lambda = 0.001$ was too small to suppress the accumulation of error within 30 iterations for 48 subsets. Any value of $\lambda > 0$ here will eventually lead to stability as $\mathbf{\Gamma}^{(k)}$ increases with $c_k = 1.5$, based on the convergence analysis (22). Particularly, OSs-mom3 algorithm with $\lambda = 0.1$ in Fig. 3(b) and (c) illustrates this stability property, where the RMSD curve recovers from the initial diverging behavior as the algorithm proceeds.

For Fig. 3(d), we executed five realizations of the random ordering and show the average of them for each curve. Here, we found that $\lambda = 0.01$ was too small to suppress the error within 30 iterations, and $\lambda = 0.1$ worked the best. Based on Fig. 3, we recommend using the bit-reversal order with $\lambda = 0.01$ rather than random ordering.

Figs. 2 and 3 are plotted with respect to the run time of each algorithm. Using larger subsets slightly increased the run time due to extra regularizer computation, but those increases were minor compared to the acceleration given by OS methods. The additional computation required for momentum methods was almost negligible, confirming that introducing momentum approach accelerates OS algorithm significantly in run time.

Overall, the simulation study demonstrated dramatic acceleration from combining OS algorithm and momentum approach.

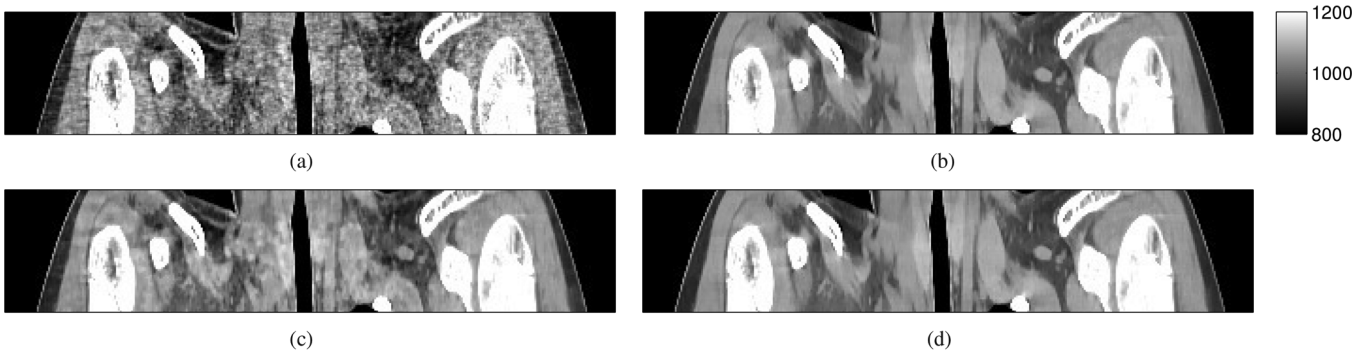

Fig. 4. Patient CT scan data: a sagittal plane of (a) an initial FBP image $\boldsymbol{x}^{(0)}$, (b) a converged image $\hat{\boldsymbol{x}}$, and two reconstructed images $\boldsymbol{x}^{(15)}$ after 15 iterations (about 865 s) from (c) OSb(24) and (d) OSb(24)-mom3 where $(c, \zeta, \lambda) = (1.5, 30, 0.01)$.

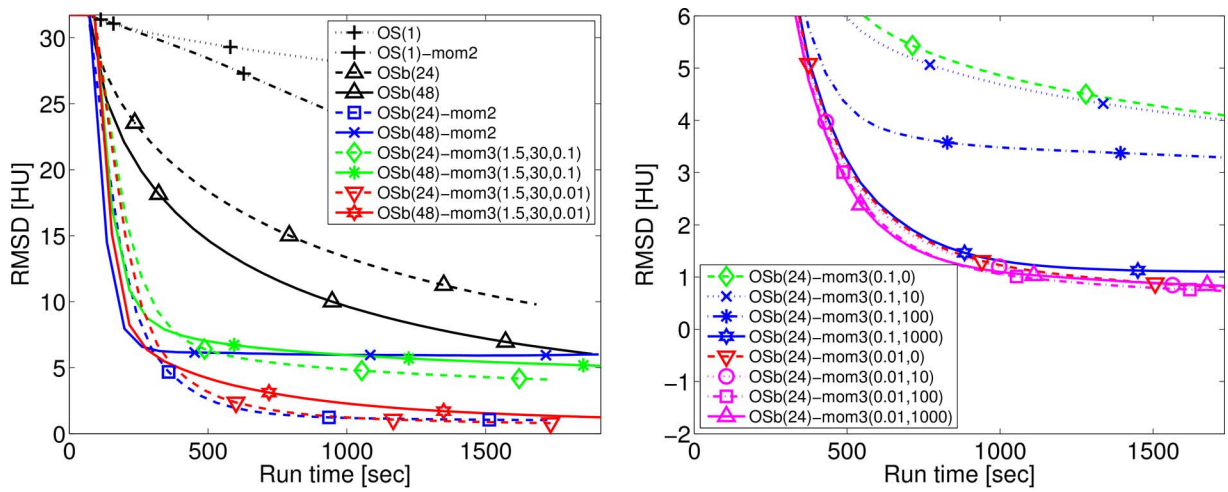

Fig. 5. Patient CT scan data: convergence rate of OSb methods (24, 48 subsets) for 30 iterations with and without momentum for (a) several choices of (c, ζ, λ) with a fixed $c_k = c = 1.5$ and (b) the choices of (λ, η) for an increasing c_k in (24) with 24 subsets and $\zeta = 30$ [HU].

Next, we study the proposed OS-momentum algorithms on patient data, and verify that the parameters tuned with the simulation data work well for real CT scans.[16]

B. Patient CT Scan Data

From a $888 \times 32 \times 7146$ sinogram measured in a helical geometry with pitch 0.5, we reconstructed a $512 \times 512 \times 109$ shoulder region image in Fig. 4. Fig. 5 shows the RMSD convergence curves for the bit-reversal subset ordering, where the results are similar to those for the simulation in Fig. 3 in terms of parameter selection. In Fig. 5(a), the parameter $\lambda = 0.01$ for both 24 and 48 subsets worked well providing overall fast convergence. Particularly for $M = 48$, the choice $\lambda = 0.01$ greatly reduced the gradient approximation error and converged faster than the unrelaxed OS-momentum algorithm.

In Fig. 5(b), we further investigate the increasing c_k (24) in (19) that starts from 1 and eventually becomes 1.5 with a tuning parameter η in (24). Larger η in (24) leads to a slowly increasing c_k, i.e., smaller c_k values in early subiterations (k), and thus, the results in Fig. 5(b) show better initial acceleration from using large η. Particularly, using large η for the choice $\lambda = 0.1$ showed a big acceleration, while that was less effective in the case $\lambda = 0.01$ due to small values of $\boldsymbol{\Gamma}$ (32) in (19).

[16]We provide results from one patient data here, and present additional results from another real CT scan in the supplementary material.

Fig. 4 shows the initial FBP image, converged, and reconstructed images from conventional OS and the proposed OS-momentum with relaxation. Visually, the reconstructed image from the proposed algorithm is almost identical to the converged image after 15 iterations.

VII. Conclusion and Discussion

We introduced the combination of OS-SQS and Nesterov's momentum techniques in tomography problems. We quantified the accelerated convergence of the proposed algorithms using simulated and patient 3D CT scans. The initial combination could lack stability for large numbers of subsets, depending on the subset ordering and type of momentum. So, we adapted a diminishing step size approach to stabilize the proposed algorithm while preserving fast convergence.

We have focused on PWLS cost function in this paper, but the proposed algorithms can be applied to any convex cost function for tomography problems, including penalized-likelihood methods based on Poisson models for pre-log sinogram data. The ideas also generalize to parallel MRI problems [54], [55]. We are further interested in studying the proof of convergence of the OS-momentum algorithm for a (bit-reversal) "deterministic" order.

The accumulating error of the proposed algorithms in Section V is hard to measure due to the computational complexity,

and thus optimizing the relaxation parameters for an increasing $\mathbf{\Gamma}^{(k)}$ in (19) remains an open issue. In our experiments, we observed that simply averaging all of the subiterations at the final iteration [29] greatly reduces RMSD, particularly when the proposed algorithm becomes unstable (depending on the relaxation parameters). One could consider this averaging technique to improve stability, or alternatively one could discard the current momentum and restart the build-up of the momentum as in [30], [56]. Such refinements could make OS-momentum a practical approach for low-dose CT.

Appendix A
Proof of Lemma 2

We extend the proof of [33, Th. 7] for *diagonally preconditioned* stochastic OS-SQS-type algorithms for the proof of Lemma 2. We first use the following lemma.

Lemma 3: For $k = nM + m \geq 0$, the sequence $\{\boldsymbol{x}^{(k/M)}\}$ generated by Table V satisfies

$$\sum_{l=0}^{k} t_l \Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - e^{(k)} \leq \min_{\boldsymbol{v}\succeq 0} \Phi^{(k)}(\boldsymbol{v})$$

$$\Phi^{(k)}(\boldsymbol{v}) \leq \sum_{l=0}^{k} t_l \Psi(\boldsymbol{v}) + \frac{\|\boldsymbol{v} - \boldsymbol{z}^{(0)}\|^2_{\mathbf{\Gamma}^{(k)}}}{2} + \bar{e}^{(k)}(\boldsymbol{v})$$

where

$$\Phi^{(k)}(\boldsymbol{v}) \triangleq \sum_{l=0}^{k} t_l \left[M\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{l}{M})}\right) + M\nabla\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{l}{M})}\right)^{\top}\left(\boldsymbol{v} - \boldsymbol{z}^{(\frac{l}{M})}\right)\right] + \frac{\|\boldsymbol{v} - \boldsymbol{z}^{(0)}\|^2_{\mathbf{\Gamma}^{(k)}}}{2}$$

that satisfies $\boldsymbol{v}^{((k+1)/M)} = \arg\min_{\boldsymbol{v}\succeq 0} \Phi^{(k)}(\boldsymbol{v})$,

$$e^{(k)} \triangleq \sum_{l=0}^{k}\sum_{i=0}^{l} t_i \left\| M\nabla\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{i}{M})}\right) - \nabla\Psi\left(\boldsymbol{z}^{(\frac{i}{M})}\right)\right\|^2_{[\mathbf{\Gamma}^{(l)} - \boldsymbol{D}]^{-1}} + \sum_{l=0}^{k} t_l \left(\Psi\left(\boldsymbol{z}^{(\frac{l}{M})}\right) - M\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{l}{M})}\right)\right) + \sum_{l=1}^{k}\sum_{i=0}^{l-1} t_i \left(\nabla\Psi\left(\boldsymbol{z}^{(\frac{i}{M})}\right) - M\nabla\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{i}{M})}\right)\right)^{\top} \times \left(\boldsymbol{z}^{(\frac{i}{M})} - \boldsymbol{x}^{(\frac{i}{M})}\right)$$

and

$$\bar{e}^{(k)}(\boldsymbol{v}) \triangleq \sum_{l=0}^{k} t_l \left[M\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{l}{M})}\right) - \Psi\left(\boldsymbol{z}^{(\frac{l}{M})}\right) + \left(M\nabla\Psi_{S_l}\left(\boldsymbol{z}^{(\frac{l}{M})}\right) - \nabla\Psi\left(\boldsymbol{z}^{(\frac{l}{M})}\right)\right)^{\top}\left(\boldsymbol{v} - \boldsymbol{z}^{(\frac{l}{M})}\right)\right].$$

Proof: Simply generalize the proof of [33, Lemma 2] using the proof of [18, Lemma 1]. ∎

Using Lemma 3 with the fact $\min_{\boldsymbol{v}\succeq 0} \Phi^{(k)}(\boldsymbol{v}) \leq \Phi^{(k)}(\hat{\boldsymbol{x}})$ leads to the following:

$$\sum_{l=0}^{k} t_l \left(\Psi\left(\boldsymbol{x}^{(\frac{k+1}{M})}\right) - \Psi(\hat{\boldsymbol{x}})\right) \leq \frac{\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2_{\mathbf{\Gamma}^{(k)}}}{2} + e^{(k)} + \bar{e}^{(k)}(\hat{\boldsymbol{x}}).$$

Finally, the expectation on the above equation provides Lemma 2, as in [33, Th. 7].

Appendix B
Choice of Coefficients t_k

Lemma 4: For any given $\{\gamma_j^{(k)}\}$ satisfying its constraint in line 5 of Table V, the $\{t_k\}$ generated by $t_0 = 1$ and

$$t_{k+1} = \frac{1}{2\alpha_{k+1}}\left(1 + \sqrt{1 + 4t_k^2\alpha_k\alpha_{k+1}}\right)$$

where $\alpha_{k+1} \triangleq \max_j(\gamma_j^{(k+1)}/\gamma_j^{(k)})$ and $\alpha_1 = 1$ tightly satisfies the following conditions:

$$t_0 \in (0, 1] \quad \text{and} \quad \alpha_{k+1}t_{k+1}^2 \leq \sum_{l=0}^{k+1} t_l, \quad \forall k \geq 0$$

which are equivalent to the conditions in line 6 of Table V.

Proof: Let t_0 have the largest possible value 1.

For $k = 0$,

$$\alpha_1 t_1^2 = t_0 + t_1 = t_0^2 + t_1$$
$$t_1 = \frac{1}{2\alpha_1}\left(1 + \sqrt{1 + 4t_0^2\alpha_0\alpha_1}\right). \tag{33}$$

For $k > 0$, we get

$$\alpha_{k+1}t_{k+1}^2 = \sum_{l=0}^{k+1} t_l = \alpha_k t_k^2 + t_{k+1}$$
$$t_{k+1} = \frac{1}{2\alpha_{k+1}}\left(1 + \sqrt{1 + 4t_k^2\alpha_k\alpha_{k+1}}\right). \tag{34}$$

∎

This rule for t_k (34) reduces to those used in Tables III and IV when $\gamma_j^{(k+1)} = \gamma_j^{(k)}$ for all $k \geq 0$ and j.

References

[1] J. A. Fessler, , M. Sonka and J. M. Fitzpatrick, Eds., "Statistical image reconstruction methods for transmission tomography," in *Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis*. Bellingham: SPIE, 2000, pp. 1–70.

[2] I. A. Elbakri and J. A. Fessler, "Statistical image reconstruction for polyenergetic X-ray computed tomography," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 89–99, Feb. 2002.

[3] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 534–548, Feb. 1993.

[4] J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh, "A three-dimensional statistical approach to improved image quality for multi-slice helical CT," *Med. Phys.*, vol. 34, no. 11, pp. 4526–4544, Nov. 2007.

[5] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange, "Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 166–175, Apr. 1997.

[6] Z. Yu, J.-B. Thibault, C. A. Bouman, K. D. Sauer, and J. Hsieh, "Fast model-based X-ray CT reconstruction using spatially non-homogeneous ICD optimization," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 161–175, Jan. 2011.

[7] J. A. Fessler and S. D. Booth, "Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction," *IEEE Trans. Image Process.*, vol. 8, no. 5, pp. 688–699, May 1999.

[8] H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601–609, Dec. 1994.

[9] H. Erdoğan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, no. 11, pp. 2835–2851, Nov. 1999.

[10] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[11] S. Ramani and J. A. Fessler, "A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction," *IEEE Trans. Med. Imag.*, vol. 31, no. 3, pp. 677–688, Mar. 2012.

[12] M. G. McGaffin, S. Ramani, and J. A. Fessler, "Reduced memory augmented Lagrangian algorithm for 3D iterative X-ray CT image reconstruction," in *Proc. SPIE 8313 Med. Imag. Phys. Med. Imag.*, 2012, p. 831327.

[13] D. P. Bertsekas, "Multiplier methods: A survey," *Automatica*, vol. 12, no. 2, pp. 133–145, Mar. 1976.

[14] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[15] E. Y. Sidky, J. H. Jorgensen, and X. Pan, "Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm," *Phys. Med. Biol.*, vol. 57, no. 10, pp. 3065–3092, May 2012.

[16] E. Y. Sidky, J. S. Jrgensen, and X. Pan, "First-order convex feasibility algorithms for X-ray CT," *Med. Phys.*, vol. 40, no. 3, p. 031115, 2013.

[17] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Dokl. Akad. Nauk. USSR*, vol. 269, no. 3, pp. 543–547, 1983.

[18] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

[19] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[20] K. Choi, J. Wang, L. Zhu, T.-S. Suh, S. Boyd, and L. Xing, "Compressed sensing based cone-beam computed tomography reconstruction with a first-order method," *Med. Phys.*, vol. 37, no. 9, pp. 5113–5125, Nov. 2010.

[21] T. L. Jensen, J. H. Jrgensen, P. C. Hansen, and S. H. Jense, "Implementation of an optimal first-order method for strongly convex total variation regularization," *BIT Numer. Math.*, vol. 52, no. 2, pp. 329–356, Jun. 2012.

[22] S. Anthoine, J.-F. Aujol, Y. Boursier, and C. Mélot, "Some proximal methods for Poisson intensity CBCT and PET," *Inverse Prob. Imag.*, vol. 6, no. 4, pp. 565–598, Nov. 2012.

[23] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. New York: Kluwer, 2004.

[24] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013.

[25] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[26] D. Kim, S. Ramani, and J. A. Fessler, "Ordered subsets with momentum for accelerated X-ray CT image reconstruction," in *Proc. IEEE Conf. Acoust. Speech Signal Process.*, 2013, pp. 920–923.

[27] D. Kim, S. Ramani, and J. A. Fessler, "Accelerating X-ray CT ordered subsets image reconstruction with Nesterov's first-order methods," in *Proc. Int. Mtg. on Fully 3D Image Recogn. in Rad. and Nucl. Med.*, 2013, pp. 22–25.

[28] S. Ahn, J. A. Fessler, D. Blatt, and A. O. Hero, "Convergent incremental optimization transfer algorithms: Application to tomography," *IEEE Trans. Med. Imag.*, vol. 25, no. 3, pp. 283–296, Mar. 2006.

[29] D. Kim, D. Pal, J.-B. Thibault, and J. A. Fessler, "Accelerating ordered subsets image reconstruction for X-ray CT using spatially non-uniform optimization transfer," *IEEE Trans. Med. Imag.*, vol. 32, no. 11, pp. 1965–1978, Nov. 2013.

[30] O. L. Mangasarian and M. V. Solodov, "Serial and parallel backpropagation convergence via nonmonotone perturbed minimization," *Optimiz. Methods Software*, vol. 4, no. 2, pp. 103–116, 1994.

[31] P. Tseng, "An incremental gradient(-projection) method with momentum term and adaptive stepsize rule," *SIAM J. Optim.*, vol. 8, no. 2, pp. 506–531, 1998.

[32] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, 2007.

[33] O. Devolder, Stochastic First Order Methods in Smooth Convex Optimization CORE, Catholic Univ. Louvain, Louvain-la-Neuve, Belgium, Tech. Rep., 2011.

[34] B. De Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Med. Biol.*, vol. 49, no. 11, pp. 2463–2475, Jun. 2004.

[35] Y. Long, J. A. Fessler, and J. M. Balter, "3D forward and back-projection for X-ray CT using separable footprints," *IEEE Trans. Med. Imag.*, vol. 29, no. 11, pp. 1839–1850, Nov. 2010.

[36] R. C. Fair, "On the robust estimation of econometric models," *Ann. Econ. Social Measure.*, vol. 2, pp. 667–677, Oct. 1974.

[37] J. A. Fessler and W. L. Rogers, "Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs," *IEEE Trans. Image Process.*, vol. 5, no. 9, pp. 1346–1358, Sep. 1996.

[38] A. H. Delaney and Y. Bresler, "Globally convergent edge-preserving regularized reconstruction: An application to limited-angle tomography," *IEEE Trans. Image Process.*, vol. 7, no. 2, pp. 204–221, Feb. 1998.

[39] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *J. Computat. Graph. Stat.*, vol. 9, no. 1, pp. 1–20, Mar. 2000.

[40] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1970.

[41] H. Erdoğan and J. A. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, no. 9, pp. 801–814, Sep. 1999.

[42] J. H. Cho and J. A. Fessler, "Accelerating ordered-subsets image reconstruction for X-ray CT using double surrogates," in *Proc. SPIE 8313 Med. Imag.: Phys. Med. Imag.*, 2012, p. 83131X.

[43] Z. Q. Luo, "On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks," *Neural Computat.*, vol. 32, no. 2, pp. 226–245, Jun. 1991.

[44] S. Ahn and J. A. Fessler, "Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms," *IEEE Trans. Med. Imag.*, vol. 22, no. 5, pp. 613–626, May 2003.

[45] Y. Nesterov, "On an approach to the construction of optimal methods of minimization of smooth convex functions," *Èkonom. I. Mat. Metody*, vol. 24, pp. 509–517, 1988.

[46] P. Tseng, On Accelerated Proximal Gradient Methods for Convex-Concave Optimization 2008 [Online]. Available: http://pages.cs.wisc.edu/~brecht/cs726docs/Tseng.APG.pdf

[47] O. Devolder, F. Glineur, and Y. Nesterov, "Intermediate gradient methods for smooth convex problems with inexact oracle," 2013, CORE discussion paper 2013/17.

[48] D. Kim and J. A. Fessler, "Ordered subsets acceleration using relaxed momentum for X-ray CT image reconstruction," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 2013, pp. 1–5.

[49] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*. New York: Wiley, 1983.

[50] J. A. Fessler, Matlab Tomography Toolbox 2004 [Online]. Available: web.eecs.umich.edu/~fessler

[51] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui, "Realistic CT simulation using the 4D XCAT phantom," *Med. Phys.*, vol. 35, no. 8, pp. 3800–3808, Aug. 2008.

[52] F. Noo, M. Defrise, and R. Clackdoyle, "Single-slice rebinning method for helical cone-beam CT," *Phys. Med. Biol.*, vol. 44, no. 2, pp. 561–570, Feb. 1999.

[53] G. T. Herman and L. B. Meyer, "Algebraic reconstruction techniques can be made computationally efficient," *IEEE Trans. Med. Imag.*, vol. 12, no. 3, pp. 600–609, Sep. 1993.

[54] S. Ramani and J. A. Fessler, "Accelerated nonCartesian SENSE reconstruction using a majorize-minimize algorithm combining variable-splitting," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2013, pp. 704–707.

[55] M. Muckley, D. C. Noll, and J. A. Fessler, "Accelerating SENSE-type MR image reconstruction algorithms with incremental gradients," in *Proc. Int. Soc. Mag. Res. Med.*, 2014, p. 4400.

[56] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Found. Computat. Math.*, 2014.