

Iterative Sampled Methods for Massive and Separable Nonlinear Inverse Problems

Julianne Chung¹(✉), Matthias Chung¹, and J. Tanner Slagel¹

Department of Mathematics, Virginia Tech, Blacksburg, VA 24060, USA
{j Chung, mcchung, slagelj}@vt.edu

Abstract. In this paper, we consider iterative methods based on sampling for computing solutions to separable nonlinear inverse problems where the entire dataset cannot be accessed or is not available all-at-once. In such scenarios (e.g., when massive amounts of data exceed memory capabilities or when data is being streamed), solving inverse problems, especially nonlinear ones, can be very challenging. We focus on separable nonlinear problems, where the objective function is nonlinear in one (typically small) set of parameters and linear in another (larger) set of parameters. For the linear problem, we describe a limited-memory sampled Tikhonov method, and for the nonlinear problem, we describe an approach to integrate the limited-memory sampled Tikhonov method within a nonlinear optimization framework. The proposed method is computationally efficient in that it only uses available data at any iteration to update both sets of parameters. Numerical experiments applied to massive super-resolution image reconstruction problems show the power of these methods.

Keywords: Tikhonov regularization · sampled methods · variable projection · Kaczmarz methods · super-resolution · medical imaging and other applications

1 Introduction

Advanced tools for image reconstruction are essential in many scientific applications ranging from biomedical to geophysical imaging [11]. A major challenge in many of the newer imaging systems is that the entire dataset cannot be accessed or is not available *all-at-once*. For example, faster scan speeds on recently-developed micro-tomography instruments have resulted in very large datasets [14]. Using standard image reconstruction techniques to analyze the massive amounts of data is computationally intractable. Another example arises in streaming-data problems or automated pipelines, where immediate feedback (e.g., a partial reconstruction) may be needed to inform the data acquisition process [21]. These scenarios are becoming common in many applications, thereby motivating the need for further developments on sampled iterative methods for image reconstruction.

We consider image reconstruction problems where the underlying model is separable and nonlinear. For the case where observations are available all-at-once, the data acquisition process can be modeled as,

$$\mathbf{b} = \mathbf{A}(\mathbf{y}_{\text{true}})\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}, \quad (1)$$

where $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$ contains the desired image, $\mathbf{y}_{\text{true}} \in \mathbb{R}^p$ contains the desired forward model parameters, $\mathbf{A}(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^{m \times n}$ is a nonlinear operator describing the forward model, $\boldsymbol{\epsilon} \in \mathbb{R}^m$ contains noise or measurement errors (typically treated as a realization from a Gaussian distribution with zero mean), and $\mathbf{b} \in \mathbb{R}^m$ contains the observations. It is often assumed that $\mathbf{A}(\mathbf{y}_{\text{true}})$ is known, in which case we have a linear model. However, in many realistic scenarios, parameters $\mathbf{y}_{\text{true}}$ must be estimated from the data. Here we assume that the parameterization of the model $\mathbf{A}(\cdot)$ is known and that the number of parameters in $\mathbf{y}_{\text{true}}$ is significantly smaller than the number of unknowns in $\mathbf{x}_{\text{true}}$, i.e., $p \ll n$. An example of a separable nonlinear inverse problem of this form arises in super-resolution image reconstruction, see Section 4.

Since image reconstruction problems are usually ill-posed, small errors in the data can result in very large errors in the solution. Thus regularization is needed to compute a reasonable solution, and here we consider the widely-used Tikhonov regularization method. That is, we are interested in Tikhonov-regularized optimization problems of the form,

$$\min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}, \mathbf{y}) = \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2, \quad (2)$$

where $\lambda > 0$ is a regularization parameter that balances the data-fit and the regularization term. Note that (2) is separable¹ since the objective function f is a linear function in terms of $\mathbf{x}$ and a nonlinear function in terms of $\mathbf{y}$. Previously developed numerical optimization methods for (2) have been investigated and range from fully decoupled approaches (e.g., alternating optimization) to fully coupled (e.g, nonlinear) approaches [6]. A popular alternative is the *variable projection* method [9, 19], where the linear parameters are mathematically eliminated and a nonlinear optimization scheme is used to solve the reduced optimization problem. These methods have been investigated for various image processing applications, see e.g., [2, 7, 12]. However, all of these methods require all-at-once access to the data to perform full matrix-vector multiplications with $\mathbf{A}(\mathbf{y})$, and hence they cannot be used for massive or streaming problems.

In this paper, we develop an iterative sampled method to estimate a solution for (2) in the case of massive or streaming data. The method follows a variable projection approach by first mathematically eliminating the linear variables. However, to address massive or streaming data, we use recently-developed sampled Tikhonov methods to approximate the regularized linear problem and use a sampled Gauss-Newton method to approximate the nonlinear variables. Sampled Tikhonov methods are simple and have favorable convergence properties [22]. Also, limited-memory variants can reduce computational costs.

¹ This is sometimes referred to as partially separable [18].

An outline of the paper is as follows. In Section 2 we provide an overview of sampled Tikhonov methods and provide a numerical exploration of the convergence properties of the limited-memory variants. Then in Section 3 we describe iterative sampled methods for separable nonlinear inverse problems, where the sampled Tikhonov methods from Section 2 are integrated within a nonlinear optimization framework for updating estimates of $\mathbf{x}_{\text{true}}$ and $\mathbf{y}_{\text{true}}$. Numerical results from super-resolution imaging are presented in Section 4, and conclusions and future work are presented in Section 5.

2 Sampled Tikhonov methods for linear inverse problems

Suppose $\mathbf{y}$ is fixed and consider computing the Tikhonov solution,

$$\mathbf{x}_{\text{tik}} = \arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2, \quad (3)$$

for the case where all of $\mathbf{A}$ and $\mathbf{b}$ are not available at once. For linear problems, we can use *sampled limited-memory Tikhonov* (**slimTik**) methods, in which we iteratively solve a sequence of sampled least-squares problems [22]. These sampled Tikhonov methods can be interpreted as extensions of block Kaczmarz type methods and are related to recursive least squares methods for ill-posed problems, randomized least-squares solvers, and stochastic optimization methods for convex programs, see e.g., [1, 3, 8, 16, 10, 17, 24].

Let's assume that matrix $\mathbf{A}$ and vector $\mathbf{b}$ can be partitioned into M blocks,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \vdots \\ \mathbf{A}^{(M)} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(M)} \end{bmatrix}. \quad (4)$$

For simplicity we assume that all blocks have the same dimension, i.e., $\mathbf{A}^{(i)} \in \mathbb{R}^{\ell \times n}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^{\ell}$, $i = 1, \dots, M$, with $\ell = m/M$. Then for an arbitrary initial guess $\mathbf{x}_0 \in \mathbb{R}^n$ and $\lambda > 0$, the k -th **slimTik** iterate can be written as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{s}_k$$

with

$$\mathbf{s}_k = \left(\frac{k\lambda}{M} \mathbf{I}_n + \mathbf{M}_k^\top \mathbf{M}_k + \mathbf{A}_k^\top \mathbf{A}_k \right)^{-1} \left(\mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) + \frac{\lambda}{M} \mathbf{x}_{k-1} \right),$$

where $\mathbf{A}_k = \mathbf{A}^{((k-1) \bmod M + 1)}$, $\mathbf{b}_k = \mathbf{b}^{((k-1) \bmod M + 1)}$, and matrix $\mathbf{M}_k = [\mathbf{A}_{k-r}^\top, \dots, \mathbf{A}_{k-1}^\top]^\top$ collects the previously-accessed matrix blocks up to *memory level* $r \in \mathbb{N}$. Note that mod is the modulo operation, and hence we are sampling with *cyclic control*. Thus, after all blocks in ascending order have been visited, the algorithm continues with the first block; other sampling strategies are possible, but these investigations are beyond the scope of this paper, see [4]. The memory level is assumed to be constant, and for the first r iterates, the blocks of $\mathbf{A}$ and $\mathbf{b}$

with negative indices are set to zero. Notice that the update $\mathbf{s}_k$ can be computed efficiently by solving the regularized least-squares problem,

$$\mathbf{s}_k = \arg \min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k \\ \mathbf{A}_k \\ \sqrt{\frac{k\lambda}{M}} \mathbf{I}_n \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k \\ \sqrt{\frac{\lambda}{kM}} \mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2,$$

using iterative methods such as LSQR [20]. The `slimTik` method with memory r is an approximation of the full memory method where $r = M - 1$, for which it can be shown that $\mathbf{x}_M = \mathbf{x}_{\text{tik}}$, for details see [22]. Hence, the full memory `slimTik` method converges after M iterations to the Tikhonov solution (3), with the corresponding regularization parameter λ .

We illustrate convergence for an example from the *Regularization Tools* toolbox [11]. We use the `gravity` example which provides a matrix $\mathbf{A} \in \mathbb{R}^{1000 \times 1000}$ and a vector $\mathbf{x}_{\text{true}}$. We partition $\mathbf{A}$ into $M = 100$ blocks with $\ell = 10$ and let $\lambda = 0.0196$. We simulate observed data by adding Gaussian white noise with zero mean such that the noise level is 0.01, i.e., $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}$ where $\frac{\|\boldsymbol{\epsilon}\|_2}{\|\mathbf{A}\mathbf{x}_{\text{true}}\|_2} = 0.01$. First, we run `slimTik` for one epoch ($k = M$) with memory levels $r = 0, \dots, M - 1$, and we report the relative error between the reconstructions $\mathbf{x}_M$ and the Tikhonov solution $\mathbf{x}_{\text{tik}}$ in the left panel of Fig. 1. Note that for full memory (i.e., $r = M - 1$), the relative error is within machine precision. Also, for lower memory levels, the reconstructions $\mathbf{x}_M$ are close to the Tikhonov solution. The right panel of Fig. 1 illustrates the asymptotic convergence of `slimTik` for memory levels $r = 0, 2, 4, 6$, and 8, where we also compare to a standard sampled gradient (`sg`) method without regularization. Errors are plotted after each full epoch. Empirically, we observe that the iterates $\mathbf{x}_k$ converge to $\mathbf{x}_{\text{tik}}$ as $k \rightarrow \infty$, using cyclic control. Asymptotic convergence of these methods using cyclic, random cyclic, or fully random control has not yet been studied and is current research. Asymptotic convergence for consistent systems using cyclic or random control has been shown in [23, 4, 16, 13].

3 Iterative sampled methods for separable nonlinear inverse problems

Next for separable nonlinear inverse problems of the form (2), we describe an iterative sampled approach that integrates `slimTik` within a nonlinear optimization framework so that both sets of parameters can be updated as data become available. Similar to the mathematical description in Section 2, we assume that $\mathbf{A}$ and $\mathbf{b}$ can be split into blocks as in (4), but where $\mathbf{A}^{(i)}(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^{\ell \times n}$, $i = 1, \dots, M$, with $\ell = m/M$. For an initial guess of the linear parameters $\mathbf{x}_0 \in \mathbb{R}^n$, nonlinear parameters $\mathbf{y}_0 \in \mathbb{R}^p$, and $\lambda > 0$, the k -th iterate of the separable nonlinear `slimTik` (`sn-slimTik`) method can be written as

$$\begin{aligned} \mathbf{x}_k &= \mathbf{x}_{k-1} - \mathbf{s}_k \\ \mathbf{y}_k &= \mathbf{y}_{k-1} - \alpha_k \left(\mathbf{J}_k^\top \mathbf{J}_k \right)^\dagger \mathbf{J}_k^\top \mathbf{r}_k(\mathbf{y}_{k-1}) \end{aligned} \quad (5)$$

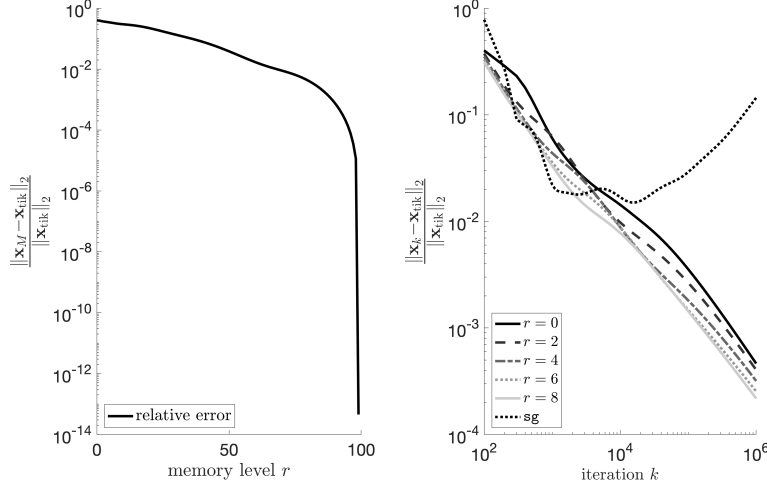


Fig. 1. Convergence of the `slimTik` method. The plot in the left panel contains the relative errors between the iterates after one epoch and the Tikhonov solution, for different memory levels. The plot in the right panel illustrates asymptotic convergence of the `slimTik` method for memory levels $r = 0, 2, 4, 6$, and 8 . For comparison we include relative errors for a sample gradient method.

with

$$\mathbf{s}_k = \arg \min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k(\mathbf{y}_{k-1}) \\ \mathbf{A}_k(\mathbf{y}_{k-1}) \\ \sqrt{\frac{k\lambda}{M}} \mathbf{I}_n \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k(\mathbf{y}_{k-1}) \mathbf{x}_{k-1} - \mathbf{b}_k \\ \sqrt{\frac{\lambda}{kM}} \mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2,$$

where $\mathbf{A}_k(\cdot) = \mathbf{A}^{(((k-1) \bmod M)+1)}(\cdot)$, $\mathbf{b}_k = \mathbf{b}^{(((k-1) \bmod M)+1)}$, and $\mathbf{M}_k(\cdot) = [\mathbf{A}_{k-r}(\cdot)^\top, \dots, \mathbf{A}_{k-1}(\cdot)^\top]^\top$ for chosen memory level $r \in \mathbb{N}$. The blocks of $\mathbf{A}$ and $\mathbf{b}$ with negative indices are set to the zero function and zero vector, respectively. Here $\mathbf{r}_k(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^{\ell(r+1)}$ is the sample residual function defined as

$$\mathbf{r}_k(\mathbf{y}) = \begin{bmatrix} \mathbf{A}_{k-r}(\mathbf{y}) \\ \vdots \\ \mathbf{A}_{k-1}(\mathbf{y}) \\ \mathbf{A}_k(\mathbf{y}) \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} \mathbf{b}_{k-r} \\ \vdots \\ \mathbf{b}_{k-1} \\ \mathbf{b}_k \end{bmatrix},$$

$\mathbf{J}_k$ is the Jacobian of $\mathbf{r}_k$ evaluated at $\mathbf{y}_{k-1}$, and α_k is the step size determined by a line search method [18]. The Jacobian can be approximated with finite differences or found analytically. Note that † represents the pseudo-inverse in (5) and is required since $\mathbf{J}_k$ might not have full column rank. Also, as with any nonlinear, nonconvex optimization method, the initial guess must be within

the basin of attraction of the desired minimizer. A summary of the **sn-slimTik** algorithm is provided below.

Algorithm 1 **sn-slimTik**

```

1: Inputs:  $\mathbf{x}_0, \mathbf{y}_0, r, \lambda, M$ 
2: for  $k = 1, 2, \dots$  do
3:   Get  $\mathbf{A}_k(\mathbf{y}_{k-1}), \mathbf{b}_k$ , and  $\mathbf{M}_k(\mathbf{y}_{k-1})$ 
4:    $\mathbf{s}_k = \arg \min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k(\mathbf{y}_{k-1}) \\ \mathbf{A}_k(\mathbf{y}_{k-1}) \\ \sqrt{\frac{k\lambda}{M}} \mathbf{I}_n \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k(\mathbf{y}_{k-1}) \mathbf{x}_{k-1} - \mathbf{b}_k \\ \sqrt{\frac{\lambda}{kM}} \mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2$ 
5:    $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{s}_k$ 
6:    $\mathbf{y}_k = \mathbf{y}_{k-1} - \alpha_k (\mathbf{J}_k^\top \mathbf{J}_k)^\dagger \mathbf{J}_k^\top \mathbf{r}_k(\mathbf{y}_{k-1})$ 
7: end for

```

4 Numerical Results

In this section, we provide numerical results for super-resolution image reconstruction, which can be represented as a separable nonlinear inverse problem [5]. Suppose we have M low-resolution images. The underlying model for super-resolution imaging can be represented as (1), where $\mathbf{x}_{\text{true}}$ contains the high-resolution (HR) image, and $\mathbf{b}$ and $\mathbf{A}(\mathbf{y}_{\text{true}})$ can be partitioned as in (4), where $\mathbf{b}^{(i)}$ contains the i -th low-resolution (LR) image and $\mathbf{A}^{(i)}(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^{\ell \times n}$. More specifically, if we assume that the deformation for each LR image is affine (e.g., can be described with at most 6 parameters) and independent of the parameters for the other images, then we can partition $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^{(1)} \\ \vdots \\ \mathbf{y}^{(M)} \end{bmatrix}$$

and have $\mathbf{A}^{(i)}(\mathbf{y}) = \mathbf{R}\mathbf{S}(\mathbf{y}^{(i)})$ where $\mathbf{R}$ is a restriction matrix that takes a HR image to a LR one and $\mathbf{S}(\mathbf{y}^{(i)})$ represents an affine transformation defined by parameters in $\mathbf{y}^{(i)}$. Then the goal is to solve (2) to estimate the HR image as well as update the transformation parameters.

We will investigate iterative sampled methods for super-resolution problems with massive or streaming data, but first we investigate a smaller problem where all of the data can be accessed at once. In Experiment 1, we compare our proposed **sn-slimTik** method with different memory levels to the results from the variable projection method. We show that with relatively modest memory levels, our approaches can achieve reconstructions with similar quality to full-memory reconstructions in comparable time. Then in Experiment 2, we consider a very large streaming super-resolution problem, where both the resolution of the images as well as the number of LR images present a computational bottleneck.

In both experiments, we initialize $\mathbf{x}_0 = \mathbf{0}$, and $\mathbf{y}_0$ is obtained by adding Gaussian white noise with zero mean to $\mathbf{y}_{\text{true}}$ where the variance is $2.45 \cdot 10^{-3}$

in Experiment 1 and $4.48 \cdot 10^{-4}$ in Experiment 2. We set the regularization parameter in advance, but mention that methods for updating the regularization parameter can be found in [22].

Algorithm 2 variable projection

```

1: Inputs:  $\mathbf{y}_0, \lambda$ 
2: for  $k = 1, 2, \dots$  do
3:    $\mathbf{x}_k = \arg \min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}_{k-1}) \\ \sqrt{\lambda} \mathbf{I}_n \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2$ 
4:    $\tilde{\mathbf{r}}_k(\mathbf{y}_{k-1}) = \mathbf{A}(\mathbf{y}_{k-1})\mathbf{x}_k - \mathbf{b}$ 
5:    $\mathbf{y}_k = \mathbf{y}_{k-1} - \alpha_k \left( \tilde{\mathbf{J}}_k^\top \tilde{\mathbf{J}}_k \right)^\dagger \tilde{\mathbf{J}}_k^\top \tilde{\mathbf{r}}_k(\mathbf{y}_{k-1})$ 
6: end for
```

4.1 Experiment 1: Comparing sn-slimTik to variable projection

Both **sn-slimTik** and variable projection are iterative methods that update $\mathbf{x}$ and $\mathbf{y}$. However, the variable projection method requires access to all data at once and thus may be infeasible for massive or streaming problems. The goal of this experiment is to show that we can achieve similar reconstructions as existing methods, but without the need to access all data and matrices at once.

For completeness, we provide in Algorithm 2 the basic variable projection algorithm [9, 19], which is a Gauss-Newton algorithm applied to the problem,

$$\min_{\mathbf{y}} f(\mathbf{x}(\mathbf{y}), \mathbf{y}).$$

Here $\tilde{\mathbf{J}}_k$ is the Jacobian of $\mathbf{A}(\mathbf{y})\mathbf{x}_k - \mathbf{b}$ with respect to $\mathbf{y}$ at $\mathbf{y}_{k-1}$, and α_k is a line search parameter. Analytical methods can be used to obtain the Jacobian, see [5]. Notice that each iteration of the variable projection algorithm requires access to the entire data set $\mathbf{b}$ as well as matrix $\mathbf{A}(\mathbf{y})$ in order to solve the linear least squares problem in step 3. For our experiments, we use the LSQR method to solve the linear Tikhonov problem, where each *iteration* of LSQR requires a matrix-vector multiplication with $\mathbf{A}(\mathbf{y}_{k-1})$ and $\mathbf{A}(\mathbf{y}_{k-1})^\top$. Each multiplication requires access to all of the data, and thus, in terms of data access, is equivalent to one epoch of **slimTik**.

For this experiment, the goal is to recover a HR image that contains 512^2 pixels from a set of $M = 100$ LR images, each containing 128^2 pixels, i.e., $\mathbf{A}(\mathbf{y}) \in \mathbb{R}^{100 \cdot 128^2 \times 512^2}$. The HR image is of an astronaut and was obtained from NASA's website [15]. The HR image and three of the simulated LR images are provided in Fig. 2. The noise level for each LR image was set to 0.01, and the regularization parameter was set to $\lambda = 8 \cdot 10^{-2}$.

In Fig. 3, we provide relative error norms for the reconstructions and relative error norms for the affine parameters,

$$\frac{\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2}{\|\mathbf{x}_{\text{true}}\|_2} \quad \text{and} \quad \frac{\|\mathbf{y}_k - \mathbf{y}_{\text{true}}\|_2}{\|\mathbf{y}_{\text{true}}\|_2}, \quad (6)$$

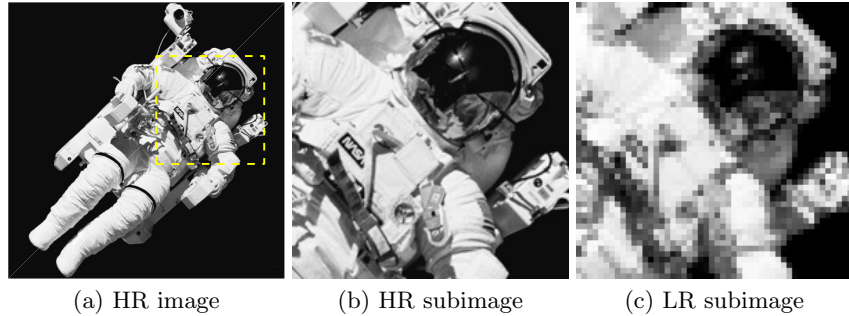


Fig. 2. Super-resolution imaging example. The high-resolution (HR) image and a subimage corresponding to the yellow box are provided in (a) and (b) respectively. The subimage of one of the low-resolution (LR) images is provided in (c).

respectively. We compare the **sn-slimTik** method with memory levels $r = 0, 1$, and 5 for 5 epochs (100 iterations correspond to one epoch), and provide results for 4 iterations of the variable projection method for comparison.

Following the discussion above, it is difficult to provide a fair comparison since each variable projection iteration requires a linear solve and here we use 20 LSQR iterations for each outer iteration. Performing one LSQR iteration requires the same memory access as 100 iterations of **sn-slimTik** with any memory level. Thus, in Fig. 3 we plot the relative reconstruction error norms for variable projection only after every 100 iterations of **sn-slimTik**. We see that for both parameters sets, **sn-slimTik** produces relative reconstruction errors that are comparable to the variable projection method. For this experiment variable projection took 644 seconds, **sn-slimTik** took 366 seconds with memory 0, 800 seconds with memory 1, and 2,570 seconds with memory 5.

Sub-images of **sn-slimTik** reconstructions at iterations $k = 100$ and 200 with memory parameters 0, 1, and 5 are provided in Fig. 4. We note that for $k = 1$ all three reconstructions are identical since all of them only have access to the first LR image. Reconstructions after 100 iterations are also similar, but after 200 iterations, we see that **sn-slimTik** with memory level 5 produces a better reconstruction. These results show that including memory in the **slimTik** algorithm may be beneficial, and results are comparable to those of variable projection.

4.2 Experiment 2: **sn-slimTik** for a massive problem

Next we consider a very large *streaming* super-resolution problem, where the goal is to reconstruct a HR image of 1024^2 pixels from 300 LR images of 64^2 pixels that are being observed in time. The HR image comes from NASA [15] and is depicted, along with three of the LR images, in Fig. 5. For this example, once all data has been accessed, $\mathbf{A} \in \mathbb{R}^{300 \cdot 64^2 \times 1,024^2}$ is too large to store in memory. Furthermore, in many streaming scenarios, we would like to be able to compute

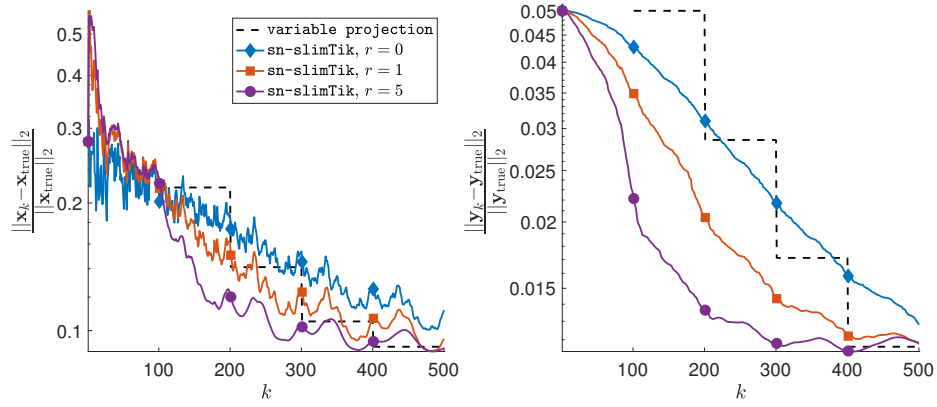


Fig. 3. Relative reconstruction error norms for the image $\mathbf{x}_k$ (left) and the nonlinear parameters $\mathbf{y}_k$ (right) for variable projection and sn-slimTik for various memory levels. Note that variable projection errors are only provided after every 100 iterations of sn-slimTik .

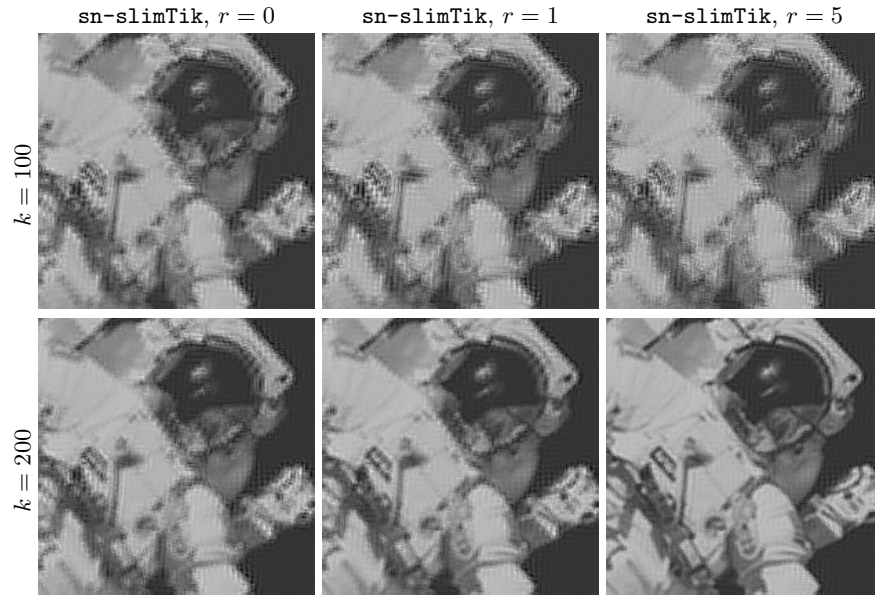


Fig. 4. Sub-images of sn-slimTik reconstructions for memory levels $r = 0, 1, \text{ and } 5$ for iterates within the first two epochs of data access.

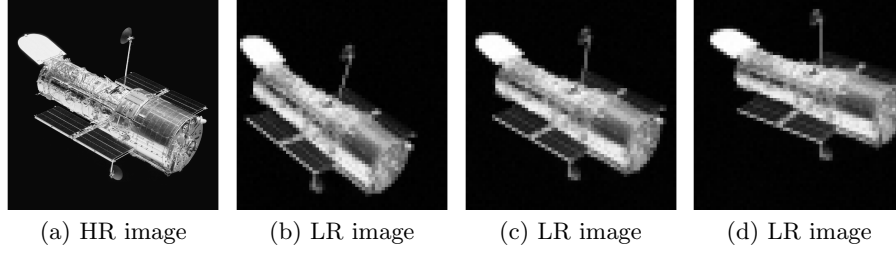


Fig. 5. Streaming super-resolution imaging example. The high-resolution ($1,024 \times 1,024$) image is provided in (a), along with three of the low-resolution (64×64) images in (b)–(d).

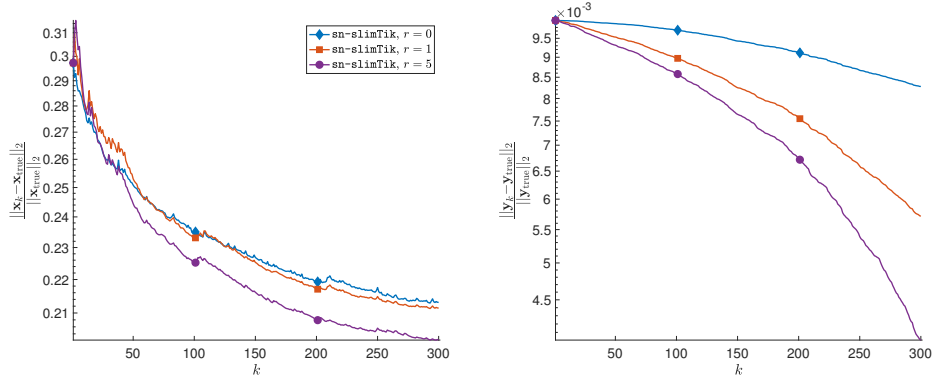


Fig. 6. Relative reconstruction errors for both the linear (left) and nonlinear (right) parameters for the streaming data super-resolution problem.

partial image reconstructions and update the nonlinear parameters during the data acquisition process, e.g., while LR images are still being streamed. Notice that the variable projection method requires us to wait until all LR images are observed, and even then it may be too costly to access all of $\mathbf{A}$ at once.

Thus, in this experiment, we consider the **sn-slimTik** method with memory levels $r = 0, 1$, and 5 . We run 300 iterations (e.g., accessing one epoch of the data) and set the noise level for each LR image to be 0.01 and $\lambda = 5 \cdot 10^{-3}$. In Fig. 6 we provide the relative reconstruction errors for $\mathbf{x}_k$ and $\mathbf{y}_k$. We observe that a higher memory level corresponds to improved estimates of the nonlinear parameters and the reconstructions. In Fig. 7, we provide sub-images of absolute errors images of the reconstructions, computed as $|\mathbf{x}_{300} - \mathbf{x}_{\text{true}}|$, in inverted colormap so that white corresponds to small absolute errors. These images show that **sn-slimTik** methods produce better reconstructions with increased memory level, but an increased memory level comes with an increase in computation time. For this

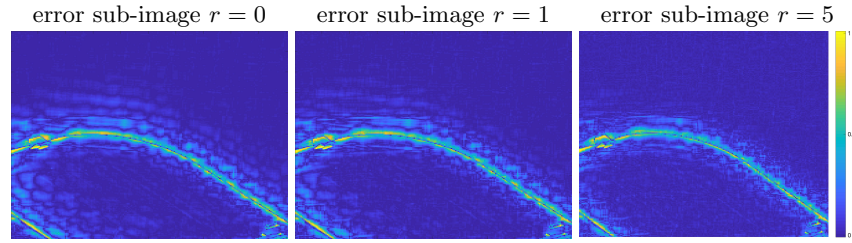


Fig. 7. Sub-image of absolute error images for **sn-slimTik** reconstructions with different memory levels.

example, the CPU times for **sn-slimTik** are 1,035, 1,954, and 5,858 seconds for memory levels of 0, 1, and 5, respectively.

5 Conclusions

In this work we introduced the **sn-slimTik** method, which is a sample based iterative algorithm to approximate the solution of a separable nonlinear inverse problem, for the case where the data cannot be accessed all-at-once. The method combines limited-memory sampled Tikhonov methods, which were developed for linear inverse problems, within a nonlinear optimization framework. Numerical results on massive super-resolution problems show that results are comparable to those from variable projection, when all data can be accessed at once. When this is not the case (e.g., streaming or massive data), the **sn-slimTik** method can effectively and efficiently update both sets of parameters.

A future area of research is to develop a theoretical analysis of the convergence properties of **sn-slimTik** and **slimTik** methods, including asymptotic convergence and a mean squared error analysis. Furthermore, future investigations should incorporate importance sampling, where the sampling strategy can be adapted as data become available.

Acknowledgements

We gratefully acknowledge support by the National Science Foundation under grants NSF DMS 1723005 and NSF DMS 1654175.

References

1. Andersen, M.S., Hansen, P.C.: Generalized row-action methods for tomographic imaging. *Numerical Algorithms* **67**(1), 121–144 (2014)
2. Berisha, S., Nagy, J.G., Plemmons, R.J.: Estimation of atmospheric PSF parameters for hyperspectral imaging. *Numerical Linear Algebra with Applications* (2015)

3. Björck, A.: Numerical Methods for Least Squares Problems. SIAM (1996)
4. Chung, J., Chung, M., Slagel, J.T., Tenorio, L.: Stochastic Newton and quasi-Newton methods for large linear least-squares problems. arXiv preprint arXiv:1702.07367 (2017)
5. Chung, J., Haber, E., Nagy, J.G.: Numerical methods for coupled super-resolution. *Inverse Problems* **22**, 1261–1272 (2006)
6. Chung, J., Nagy, J.G.: An efficient iterative approach for large-scale separable nonlinear inverse problems. *SIAM Journal on Scientific Computing* **31**(6), 4654–4674 (2010)
7. Cornelio, A., Piccolomini, E.L., Nagy, J.G.: Constrained variable projection method for blind deconvolution. In: *Journal of Physics: Conference Series*. vol. 386, p. 012005. IOP Publishing (2012)
8. Escalante, R., Raydan, M.: Alternating projection methods, vol. 8. SIAM (2011)
9. Golub, G., Pereyra, V.: Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems* **19**, R1–R26 (2003)
10. Gower, R.M., Richtárik, P.: Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications* **36**(4), 1660–1690 (2015)
11. Hansen, P.C.: Discrete Inverse Problems: Insight and Algorithms. SIAM (2010)
12. Herring, J., Nagy, J., Ruthotto, L.: LAP: a linearize and project method for solving inverse problems with coupled variables. *Sampling Theory in Signal and Image Processing* **17**(2), 127–151 (2018)
13. Kaczmarz, S.: Angenäherte Auflösung linearer Gleichungssysteme. *Bulletin International de l’Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques* pp. 355–357 (1937)
14. Marchesini, S., Krishnan, H., Daurer, B.J., Shapiro, D.A., Perciano, T., Sethian, J.A., Maia, F.R.: SHARP: a distributed GPU-based ptychographic solver. *Journal of Applied Crystallography* **49**(4), 1245–1252 (2016)
15. NASA: Images from NASA webpage. <https://www.nasa.gov>, accessed: 2019-01-10
16. Needell, D., Tropp, J.A.: Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra and its Applications* **441**, 199–221 (2014)
17. Needell, D., Zhao, R., Zouzias, A.: Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra and its Applications* **484**, 322–343 (2015)
18. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, 2 edn. (2006)
19. O’Leary, D.P., Rust, B.W.: Variable projection for nonlinear least squares problems. *Computational Optimization and Applications* **54**(3), 579–593 (2013)
20. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software* **8**(1), 43–71 (1982)
21. Parkinson, D.Y., Pelt, D.M., Perciano, T., Ushizima, D., Krishnan, H., Barnard, H.S., MacDowell, A.A., Sethian, J.: Machine learning for micro-tomography. In: *Developments in X-Ray Tomography XI*. vol. 10391, p. 103910J. International Society for Optics and Photonics (2017)
22. Slagel, J.T., Chung, J., Chung, M., Kozak, D., Tenorio, L.: Sampled Tikhonov regularization for large linear inverse problems. arXiv preprint arXiv:1812.06165 (2018)
23. Strohmer, T., Vershynin, R.: A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications* **15**(2), 262–278 (2009)
24. Zouzias, A., Freris, N.M.: Randomized extended Kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis and Applications* **34**(2), 773–793 (2013)