AN INTERIOR POINT METHOD FOR NONNEGATIVE SPARSE SIGNAL RECONSTRUCTION

Xiang Huang[1], *Kuan He*[2], *Seunghwan Yoo*[2],*Oliver Cossairt*[2], *Aggelos Katsaggelos*[2], *Nicola Ferrier*[1], *Mark Hereld*[1]

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA

[2]Dept. of Electrical Eng. and Computer Science, Northwestern University, Evanston, IL 60208, USA

ABSTRACT

We present a primal-dual interior point method (IPM) with a novel preconditioner to solve the ℓ_1-norm regularized least square problem for nonnegative sparse signal reconstruction. IPM is a second-order method that uses both gradient and Hessian information to compute effective search directions and achieve super-linear convergence rates. It therefore requires many fewer iterations than first-order methods such as iterative shrinkage/thresholding algorithms (ISTA) that only achieve sub-linear convergence rates. However, each iteration of IPM is more expensive than in ISTA because it needs to evaluate an inverse of a Hessian matrix to compute the Newton direction. We propose to approximate each Hessian matrix by a diagonal matrix plus a rank-one matrix. This approximation matrix is easily invertible using the Sherman-Morrison formula, and is used as a novel preconditioner in a preconditioned conjugate gradient method to compute a truncated Newton direction. We demonstrate the efficiency of our algorithm in compressive 3D volumetric image reconstruction. Numerical experiments show favorable results of our method in comparison with previous interior point based and iterative shrinkage/thresholding based algorithms.

***Index Terms*—** ℓ_1-norm regularized optimization, primal-dual preconditioned interior point method, nonnegative sparse, compressive sensing, 3d volumetric image reconstruction

1. INTRODUCTION AND PREVIOUS WORK

We are interested in reconstructing an unknown signal $\mathbf{x} \in \mathbb{R}^N$ from its linear observation:

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{n} \in \mathbb{R}^M, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$, and $\mathbf{n} \in \mathbb{R}^M$ is the noise.

Classic least square method requires abundant measurements ($M \geq N$ and $\mathbf{A}$ has full rank N) to recover $\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. Recent compressive sensing techniques can reconstruct $\mathbf{x}$ from many fewer measurements ($M \ll N$) as long as the signal is sparse (i.e., most elements of $\mathbf{x}$ are zero or small) by solving the following basis pursuit denoising (BPDN) problem:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \quad \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 + \tau\|\mathbf{x}\|_1, \tag{BPDN}$$

where $\tau > 0$ is a given regularization weight, $\|\mathbf{x}\| := \sqrt{\sum_{i=1}^N \mathbf{x}_i^2}$ and $\|\mathbf{x}\|_1 := \sum_{i=1}^N |\mathbf{x}_i|$ denote the ℓ_2 and the ℓ_1 norms of $\mathbf{x}$, respectively.

In this paper, we only consider nonnegative signals $\mathbf{x} \geq \mathbf{0}$ for three reasons. First, image intensity is physically nonnegative, as it is proportional to the number of nonnegative photons received. Second, explicitly adding the constraint $\mathbf{x} \geq \mathbf{0}$ in Eq. (BPDN) limits the solution search space in the nonnegative orthant and can thus result in a better truly nonegative reconstruction. Third, if $\mathbf{x}$ does have negative components, we can use a common optimization technique to let $\hat{\mathbf{x}} = [\mathbf{x}^+; \mathbf{x}^-] \in \mathbb{R}^{2N} \geq \mathbf{0}$ and $\hat{\mathbf{A}} = [\mathbf{A}, -\mathbf{A}] \in \mathbb{R}^{M\times 2N}$, where $\mathbf{x}_i^+ = \max(\mathbf{x}_i, 0)$ and $\mathbf{x}_i^- = \max(-\mathbf{x}_i, 0)$, then $\mathbf{A}\mathbf{x} = \hat{\mathbf{A}}\hat{\mathbf{x}}$ and $\|\mathbf{x}\|_1 = \|\hat{\mathbf{x}}\|_1$, and hence Eq. (BPDN) can be solved with respect to $\hat{\mathbf{A}}$ and $\hat{\mathbf{x}} \geq \mathbf{0}$. Therefore, we only need to consider the following variant of Eq. (BPDN) for $\mathbf{x} \geq \mathbf{0}$,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^N} \quad & \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 + \tau\mathbf{e}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{BPDN+}$$

First-order methods with gradient-based search direction are widely used to solve Eq. (BPDN) by the compressed sensing community due to their simplicity and the low cost of each iteration, and they can be adjusted to solve Eq. (BPDN+) with a simple nonnegative thresholding step in each iteration. Popular choices include the iterative shrinkage/thresholding (ISTA) method [1] and its improved variants TwIST [2] and FISTA [3], the projected gradient method [4], and the alternating direction method [5]. First-order methods generally only achieve sub-linear convergence and require $O(1/\epsilon)$ [1] or $O(1/\sqrt{\epsilon})$ [3] iterations to achieve accuracy ϵ, thus may take long time to compute high accuracy results.

On the other hand, second-order methods have attracted much less attention in compressive sensing applications. This is not surprising because of the extra complication of adding second-order partial derivative information, and directly solving the Newton linear system at each iteration requires expensive $O(N^3)$ computations. However, second-order methods can achieve super-linear convergence (less than $O(log(1/\epsilon))$ iterations) and are suitable for high accuracy reconstruction. To mitigate the high cost of computing each Newton direction, both Kim *et al.* [6] and Fountoulakis *et al.* [7] proposed interior-point methods (IPMs) with a preconditioned conjugated gradient (PCG) algorithm to approximately solve for

the Newton direction in much less than $O(N^3)$ computations. They both converted Eq. (BPDN) to Eq. (BPDN+) by variable substitution of $\hat{\mathbf{x}} = [\mathbf{x}^+; \mathbf{x}^-] \geq \mathbf{0}$. They both used preconditioners from the approximation of the Hessian matrix (which contains $\mathbf{A}^T\mathbf{A}$ as a component) to accelerate PCG. Kim *et al.* [6] approximated $\mathbf{A}^T\mathbf{A}$ simply by its diagonal elements, which works well when $\mathbf{A}^T\mathbf{A}$ is diagonally dominant. Fountoulakis *et al.* [7] approximated $\mathbf{A}^T\mathbf{A}$ by $\frac{M}{N}\mathbf{I}$ where $\mathbf{I}$ is an identity matrix, which has good performance when $\mathbf{A}$ has orthonormal rows and satisfies the restricted isometry property (RIP) well. While Kim *et al.* [6] used a log-barrier function to push the solution away from the boundary of the nonnegative orthant and guide the search in primal interior space $\mathbf{x} > \mathbf{0}$, more recently Fountoulakis *et al.* [7] searched the optimal solution in primal-dual interior space $(\mathbf{x}, \mathbf{s}) > \mathbf{0}$ where $\mathbf{s}$ is the dual variable, and used a predictor-corrector method to move $(\mathbf{x}, \mathbf{s})$ away from the boundary.

In this paper, we first propose a primal-dual IPM with a novel preconditioner using rank-one approximation of $\mathbf{A}^T\mathbf{A}$ in Section 2. Then we compare our algorithm performance with other algorithms on a 3D volumetric image reconstruction example in Section 3.

2. THE PRIMAL-DUAL PRECONDITIONED IPM

The optimization problem of Eq. (BPDN+) is convex with only linear constraints and satisfies Slater's condition, therefore we can find its optimal solution(s) by solving its Karush-Kuhn-Tucker (KKT) system:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{s} - \mathbf{A}^T\mathbf{b} + \tau\mathbf{e} = \mathbf{0}, \tag{2a}$$

$$\mathbf{X}\mathbf{S}\mathbf{e} = \mathbf{0}, \tag{2b}$$

$$(\mathbf{x}, \mathbf{s}) \geq \mathbf{0}, \tag{2c}$$

where $\mathbf{X} \coloneqq \mathbf{Diag}(\mathbf{x})$ and $\mathbf{S} \coloneqq \mathbf{Diag}(\mathbf{s})$ denote diagonal matrices constructed from primal variable $\mathbf{x}$ and dual variable $\mathbf{s}$, respectively, and $\mathbf{0}$ and $\mathbf{e}$ denote an all zero or all one vector whose dimension shall be clear from context, respectively.

The primal-dual IPM solves a modified KKT system which simply replaces Eq. (2b) in the original KKT system by $\mathbf{X}\mathbf{S}\mathbf{e} = \sigma\mu\mathbf{e}$, where $\mu \coloneqq \mathbf{x}^T\mathbf{s}/N$ goes to 0 when converges, and $\sigma \in [0, 1]$ is a centering parameter. A σ closer to 1 will guide search direction more towards the interior $(\mathbf{x}, \mathbf{s}) > \mathbf{0}$. Starting from a point $(\mathbf{x}, \mathbf{s})$, the Newton direction for the modified KKT system can be computed as

$$\mathbf{A}^T\mathbf{A}\Delta\mathbf{x} - \Delta\mathbf{s} = \mathbf{r}_d, \tag{3a}$$

$$\mathbf{S}\Delta\mathbf{x} + \mathbf{X}\Delta\mathbf{s} = \mathbf{r}_c, \tag{3b}$$

where the stationarity residual $\mathbf{r}_d$ and complementary slackness residual $\mathbf{r}_c$ can be represented as

$$\mathbf{r}_d \coloneqq \mathbf{s} - \nabla h(\mathbf{x}), \tag{4a}$$

$$\mathbf{r}_c \coloneqq \sigma\mu\mathbf{e} - \mathbf{X}\mathbf{S}\mathbf{e}. \tag{4b}$$

Here, $\nabla h(\mathbf{x}) = \mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b} + \tau\mathbf{e}$ is the gradient of the objective function $h(\mathbf{x}) \coloneqq \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \tau\mathbf{e}^T\mathbf{x}$.

We present our primal dual preconditioned IPM with predictor-corrector steps in Algorithm 1. Similar to [7], we also use a primal-dual IPM framework as it is considered to be the most successful among various IPMs [8]. In comparison with the [7], we use new initialization, simpler yet effective σ values, new preconditioner and adaptive tolerance in PCG to achieve faster convergence. While [7] requires Eq. (2a) to be always satisfied, we allow more flexible $\mathbf{x}, \mathbf{s}$ that violate Eq. (2a) at initial setup and at the following iterations, and only requires Eq. (2a) to be satisfied at convergence.

Algorithm 1 Primal Dual Preconditioned IPM Framework

Inputs: choose $(\mathbf{x}^0, \mathbf{s}^0) > \mathbf{0}$ from Section 2.1, stop accuracy ϵ (e.g. $1e-6$), and maximum iteration number $k_{\max}$.
for $k = 1, 2, \ldots, k_{\max}$ **do**
  Perform Prediction Step: set $\sigma \leftarrow 0.01$.
  $(\mathbf{x}^k, \mathbf{s}^k, \alpha_p, \alpha_d)$ = UPDATE$(\mathbf{x}^{k-1}, \mathbf{s}^{k-1}, \sigma)$
  if $\min(\alpha_p, \alpha_d) \leq 0.1$ **then**
    Perform Correction Step: set $\sigma \leftarrow 0.99$.
    $(\mathbf{x}^k, \mathbf{s}^k, \alpha_p, \alpha_d)$ = UPDATE$(\mathbf{x}^{k-1}, \mathbf{s}^{k-1}, \sigma)$
  if $\mu_k \leq \epsilon h(\mathbf{x}^k)$ and $\|\mathbf{r}_d^k\| \leq \epsilon$ **then**
    Break
Output: $\mathbf{x}^k$
function UPDATE$(\mathbf{x}^{k-1}, \mathbf{s}^{k-1}, \sigma)$
  Compute $\Delta\mathbf{x}, \Delta\mathbf{s}$ with $\sigma, \mathbf{x}^{k-1}, \mathbf{s}^{k-1}$ use Section 2.2
  Compute α_p, α_d with $\mathbf{x}^{k-1}, \mathbf{s}^{k-1}, \Delta\mathbf{x}, \Delta\mathbf{s}$ use Section 2.3
  Update $(\mathbf{x}^k, \mathbf{s}^k) \leftarrow (\mathbf{x}^{k-1} + \alpha_p\Delta\mathbf{x}, \mathbf{s}^{k-1} + \alpha_d\Delta\mathbf{s})$.
  return $(\mathbf{x}^k, \mathbf{s}^k, \alpha_p, \alpha_d)$

2.1. Initial Point

A well chosen initial point $(\mathbf{x}^0, \mathbf{s}^0) > \mathbf{0}$ that is close to the optimal solution (satisfies Eq. (2)) can significantly reduce the total computation time of the algorithm. We use following steps to compute $(\mathbf{x}^0, \mathbf{s}^0) > \mathbf{0}$ that satisfies Eq. (2a) approximately and makes $\mathbf{x}_i$ and $\mathbf{s}_i$ balanced (not too dissimilar), which works well in practice. We first find the least squares solution of minimum norm for $\mathbf{A}\mathbf{x} = \mathbf{b}$ as $\mathbf{x}^a = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$ to keep both $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ and $\|\mathbf{x}\|$small. Then we compute $\delta^a = \max(0, -\mathbf{x}^a_{\min})$ where $\mathbf{x}^a_{\min}$ denotes the minimum element of $\mathbf{x}^a$, and adjust $\mathbf{x}^a \leftarrow \mathbf{x}^a + \delta^a\mathbf{e}$ to make $\mathbf{x}^a$ nonnegative. Then we compute $\delta^b = \max(0, (\epsilon_0\mathbf{x}^a_{\max} - \mathbf{x}^a_{\min})/(1 - \epsilon_0))$ where $\epsilon_0 = 0.001$ and $\mathbf{x}^a_{\max}$ denotes the maximum element of $\mathbf{x}^a$, and update $\mathbf{x}^b = \mathbf{x}^a + \delta^b\mathbf{e}$ so that $\mathbf{x}^b$ is strictly positive and its components are not too unbalanced: $\mathbf{x}^b_{\min}/\mathbf{x}^b_{\max} \geq \epsilon_0$. Then we compute $\delta^c = \max(0, -(\nabla h(\mathbf{x}^b)./(\mathbf{A}^T\mathbf{A}\mathbf{e}))_{\max})$ which is used to update $\nabla h(\mathbf{x}^b)$ to be nonnegative, where $./$ denotes element-wise division for two vectors, and update $\mathbf{x}^0 = \mathbf{x}^a + \delta^c\mathbf{e} > \mathbf{0}$ and $\mathbf{s}^0 = \nabla h(\mathbf{x}^0) \geq \mathbf{0}$. Finally we compute $\delta^d = \max(0, (\epsilon_0\mathbf{s}^0_{\max} - \mathbf{s}^0_{\min})/(1 - \epsilon_0))$, and adjust $\mathbf{s}^0 = \mathbf{s}^0 + \delta^d\mathbf{e}$ so that $\mathbf{s}^0 > \mathbf{0}$ is balanced: $\mathbf{s}^0_{\min}/\mathbf{s}^0_{\max} \geq \epsilon_0$.

The above computational cost is dominated by the first step $(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$, which can be solved either iteratively using PCG or directly if $\mathbf{A}$ has special structure (Section 3). In either case, the computational cost is no more than one iteration of Algorithm 1. Note our initial point may not satisfy

Eq. (2a), as $\mathbf{s}^0 - h(\mathbf{x}^0) \neq \mathbf{0}$ if $\delta^d > 0$ in the last step.

2.2. Newton Step with PCG

To compute $(\Delta\mathbf{x}, \Delta\mathbf{s})$ from Eq. (3), we first eliminate $\Delta\mathbf{s}$ and solve for $\Delta\mathbf{x}$ from

$$(\mathbf{D}^{-1} + \mathbf{A}^T\mathbf{A})\Delta\mathbf{x} = \mathbf{c}, \tag{5}$$

where

$$\mathbf{D} := \mathbf{X}\mathbf{S}^{-1}, \tag{6a}$$

$$\mathbf{c} := \mathbf{r}_d + \mathbf{X}^{-1}\mathbf{r}_c = \sigma\mu\mathbf{X}^{-1}\mathbf{e} - \nabla h(\mathbf{x}) \tag{6b}$$

Instead of directly solving Eq. (5), we use a preconditioner $\mathbf{M} = \mathbf{P}^T\mathbf{P}$ to transfer it to an equivalent equation:

$$\left(\mathbf{P}^{-T}(\mathbf{D}^{-1} + \mathbf{A}^T\mathbf{A})\mathbf{P}^{-1}\right)(\mathbf{P}\Delta\mathbf{x}) = \mathbf{P}^{-T}\mathbf{c}. \tag{7}$$

The preconditioner $\mathbf{M}$ significantly affects the efficiency of PCG. It should be easily invertible and approximate $(\mathbf{D}^{-1} + \mathbf{A}^T\mathbf{A})$ well to make $\mathbf{P}^{-T}(\mathbf{D}^{-1} + \mathbf{A}^T\mathbf{A})\mathbf{P}^{-1}$ well conditioned. The design of a good preconditioner is typically problem specific. The diagonal approximation [6] of $\mathbf{A}^T\mathbf{A}$ is good for diagonal dominant matrices, and the $\frac{M}{N}\mathbf{I}$ approximation [7] works when $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ and $\mathbf{A}$ satisfies RIP well.

In many image processing applications, convolution with a nonnegative point spread function (PSF) is used and of special interest for us. A convolution related matrix $\mathbf{A}$ has circulant structure, so that the eigenvalue decomposition of $\mathbf{A}^T\mathbf{A}$ maybe available (Section 3). Let the eigenvalue decomposition be

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^T, \tag{8}$$

where the eigenvalues $\lambda_1 \geq \lambda_2 \cdots$. If the PSF has most energy concentrated in low frequency, we may use the first few or even one columns of $\mathbf{V}$ to approximate $\mathbf{A}^T\mathbf{A}$ as $\lambda_1\mathbf{v}_1\mathbf{v}_1^T = \mathbf{v}\mathbf{v}^T$ where $\mathbf{v} := \sqrt{\lambda_1}\mathbf{v}_1$. The inverse of this preconditioner is computable by the Sherman-Morrison formula

$$\mathbf{M}^{-1} = (\mathbf{D}^{-1} + \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{D} - \frac{\mathbf{D}\mathbf{v}\mathbf{v}^T\mathbf{D}}{1 + \mathbf{v}^T\mathbf{D}\mathbf{v}} \tag{9}$$

With this preconditioner, we feed the search direction from the previous iteration (or $\mathbf{0}$ for the first iteration) as the initial point of PCG. The PCG stops when a maximum number of iterations (e.g., 100) is reached or when it finds a solution within relative tolerance $tol_{CG} = \max(.001, \min(0.1, (\mu + \|\mathbf{r}_d\|)/\|\mathbf{b}\|))$, which prefers more accurate Newton direction when approaching the optimal solution.

After getting $\Delta\mathbf{x}$, we can compute $\Delta\mathbf{s}$ from Eq. (3a) or Eq. (3b),

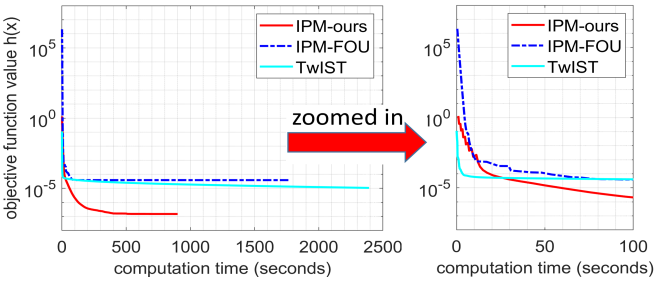
$$\Delta\mathbf{s} = \mathbf{A}^T\mathbf{A}\Delta\mathbf{x} - \mathbf{r}_d, \tag{10a}$$

$$\Delta\mathbf{s} = -\mathbf{D}^{-1}\Delta\mathbf{x} + \mathbf{X}^{-1}\mathbf{r}_c. \tag{10b}$$

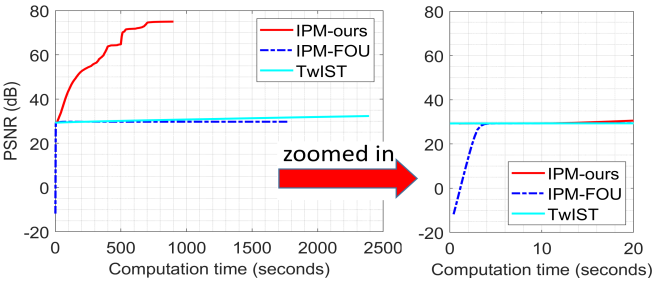
Those two equations result in different $\Delta\mathbf{s}$ as $\Delta\mathbf{x}$ only satisfies Eq. (5) approximately. We choose to compute $\Delta\mathbf{s}$ by Eq. (10b) so that Eq. (3b) is satisfied exactly to promote longer step size α_d than using Eq. (10a). By making this choice, Eq. (2a) may be violated even after a full step size $\alpha_p = 1, \alpha_d = 1$.

2.3. Step Size

The step sizes (α_p, α_d) are computed as suggested by [9]. We first find $\alpha_p^{\max}$ that makes $\mathbf{x} + \alpha_p^{\max}\Delta\mathbf{x} \geq \mathbf{0}$. If $\Delta\mathbf{x} \geq \mathbf{0}$, then we set $\alpha_p^{\max} = +\infty$; otherwise $\alpha_p^{\max} = \min_{\Delta\mathbf{x}_i < 0} -\frac{\mathbf{x}_i}{\Delta\mathbf{x}_i}$. Then we compute $\alpha_p = \min(1, \gamma\alpha_p^{\max})$, where $\gamma \in [0.9, 1.0)$ (e.g. $\gamma = 0.999$ in this paper) is used to make the updated $\mathbf{x}$ strictly positive. The computation of α_d is similar.



(a) Objective Function Value vs Time



(b) PSNR vs Time

Fig. 1: Evaluation of our algorithm. (a) Comparison of the objective function values in Eq. (BPDN+) over time. (b) Comparison of the PSNRs over time. IPM-ours performs better than either IPM-FOU or TwIST in this example.

3. APPLICATION TO 3D VOLUMETRIC IMAGE RECONSTRUCTION

In this section, we apply the proposed algorithm to reconstruct a 3D volumetric image from its single 2D observation captured by a 2D image sensor.

In 3D fluorescence microcopy, a 2D image $\mathbf{B}(u, v)$ observed at the detection plane encodes the information of the 3D volumetric fluorescence distribution: $\mathbf{B}(u, v) = \sum_{z=1}^{N_z} \mathbf{H}(u, v; z) \circledast \mathbf{X}(u, v; z)$, where we assume the focal plane is at $z = 0$ and $\circledast$ denotes 2D circular convolution between the 3D PSF $\mathbf{H}(u, v; z)$ and the object $\mathbf{X}(u, v; z)$ at depth z. Using the convolution theorem, it can be rewritten as

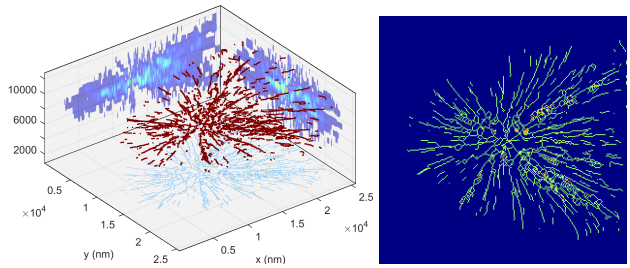
$$\mathbf{b} = \mathbf{A}\mathbf{x}, \tag{11}$$

where $\mathbf{b} \in \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^N$ are lexicographical vector representations of $\mathbf{B}$ and $\mathbf{X}$. The system model matrix $\mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2\ \ldots\ \mathbf{A}_{N_z}] \in \mathbb{R}^{M \times N}$ is the horizontal concatenation of sub-matrices $\mathbf{A}_z \in \mathbb{R}^{M \times M}$ for $z \in 1, 2, \ldots, N_z$ where

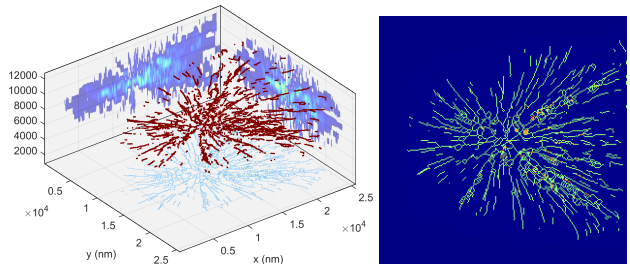
$$\mathbf{A}_z = \mathbf{F}^{-1}\mathbf{O}_z\mathbf{F}. \tag{12}$$

Here $\mathbf{F}$ denotes the 2D Fourier transform matrix, and $\mathbf{O}_z$ is a diagonal matrix whose diagonal components are the Fourier

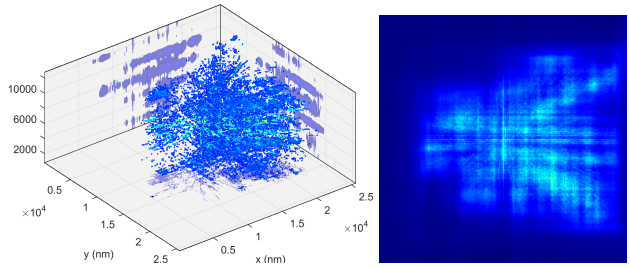
transform of $\mathbf{H}(u, v; z)$ at depth z. In our implementation, we use fast Fourier transform (FFT) so that the operations of $\mathbf{Ax}$ and $\mathbf{A}^T\mathbf{x}$ are much faster than conventional matrix vector multiplication.



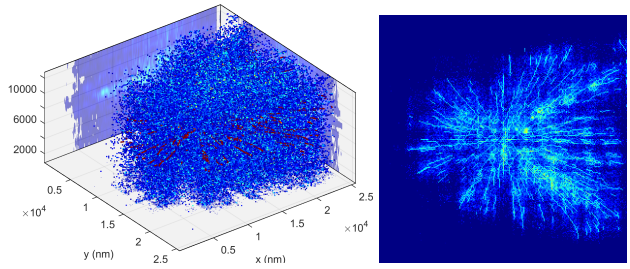
(a) Ground Truth



(b) Our Algorithm's result computed with 901 seconds



(c) IPM-FOU result computed with 1773 seconds



(d) TwIST result computed with 2392 seconds

Fig. 2: Comparison of the results for the three algorithms to the ground truth. Left column: 3D volumetric image results at the end of computation, showing as both point clouds and 2D projections to xy,yz,xz plane. Right column: the projection onto the xy plane. Intensities from low to high are mapped from blue to red.

We can show that the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are just diagonal elements of $\mathbf{OO}^H$ where ^{H} denotes conjugate transpose and $\mathbf{O} := [\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_{N_z}]$. Since a PSF is nonnegative and its Fourier transform has large energy distributed in the zero frequency (DC) component, the largest eigenvalue, i.e., the largest diagonal component of $\mathbf{OO}^H$, is usually the first one. The corresponding scaled eigenvector $\mathbf{v}$ in Eq. (9) can be computed as $[p_1\mathbf{e}_M; p_2\mathbf{e}_M; \ldots; p_{Nz}\mathbf{e}_M]$ where $\mathbf{e}_M$ is an all one M-dimensional vector and scalar $p_z = \sum_{u,v} \mathbf{H}(u, v; z)$.

In the simulation, we generate a random PSF (independent, identically distributed uniformly in $[0, 1]$) of size $256 \times 256 \times 16$, then construct the system model $\mathbf{A}$ from the PSF using FFT. We use a 3D microtubule volumetric object from [10] for testing, which is down-sampled to the same size as PSF and vectorized as $\mathbf{x}_{GT} \in \mathbb{R}^{1048576}$. We then apply the forward model to generate $\mathbf{b} = \mathbf{Ax}_{GT}$, where $\mathbf{b} \in \mathbb{R}^{65536}$ is the vectorized version of the 2D observed image of 256×256 pixels.

We reconstruct the 3D microtubule from its observation $\mathbf{b}$ by finding the optimal solution $\mathbf{x}^*$ in Eq. (BPDN+), and compare the reconstruction $\mathbf{x}^*$ with ground truth $\mathbf{x}_{GT}$. For simplicity, we do not add any noise and just use a very small regularization parameter $\tau = 10^{-6}$. We use our algorithm (referred as IPM-ours), the IPM in [7] (referred as IPM-FOU), and the TwIST algorithm in [2] to solve for Eq. (BPDN+). In our simulation, we added a thresholding step $\mathbf{x}(\mathbf{x} < \mathbf{0}) = \mathbf{0}$ in each iteration of TwIST to handle the nonnegativity constraint. With this extra step, TwIST achieved faster convergence and better result. We also removed the splitting step in IPM-FOU to work with the original nonnegative variable $\mathbf{x}$. Because IPM-FOU works better if $\mathbf{AA}^T$ is close to $\mathbf{I}$, we normalized $\mathbf{A}$ by its largest singular value, so that $\mathbf{AA}^T$ has the same spectral norm 1 as $\mathbf{I}$. In comparison, our algorithm shows faster convergence and better reconstruction comparing with the IPM-FOU and the TwIST algorithms both quantitatively (Fig. 1) and qualitatively (Fig. 2).

4. CONCLUSION AND FUTURE WORK

In this paper, we presented a primal-dual preconditioned IPM to solve the ℓ_1 regularized least square optimization problem. By (1) estimating a good starting point, (2) using a novel preconditioner for PCG, and (3) using a simple yet effective predictor-corrector protocol, our algorithms has competitive performance, as shown in a challenging 3D compressive volumetric image reconstruction example. Our algorithm is especially suitable for large scale nonnegative sparse signal reconstruction, such as sparse 3D particle tracking and microtubule reconstruction applications in biomedical microscopy.

In the future, we will perform more numerical tests. We plan to extend the framework to use a sparsity regularizer with a dictionary $\mathbf{\Psi}$, i.e., use $\|\mathbf{\Psi x}\|_1$ instead of $\|\mathbf{x}\|_1$ as the regularizer in Eq. (BPDN+). We also would like to extend the rank-one approximation to rank-k approximation for small k.

Acknowledgments

We acknowledge helpful discussions with Sven Leyffer and Todd Munson. This work was supported by funding through the Biological Systems Science Division, Office of Biological and Environmental Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

5. REFERENCES

[1] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, nov 2004. [Online]. Available: http://doi.wiley.com/10.1002/cpa.20042

[2] J. Bioucas-Dias and M. Figueiredo, "A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, dec 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4358846/

[3] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009. [Online]. Available: http://epubs.siam.org/doi/10.1137/080716542

[4] M. Figueiredo, R. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *Sel. Top. Signa*, vol. vol, no. 4, pp. 1pp586–597, 2007.

[5] J. Yang and Y. Zhang, "Alternating Direction Algorithms for L1 Problems in Compressive Sensing," *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 250–278, jan 2011. [Online]. Available: http://epubs.siam.org/doi/10.1137/090777761

[6] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale L1-Regularized Least Squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4407767

[7] K. Fountoulakis, J. Gondzio, and P. Zhlobich, "Matrix-free interior point method for compressed sensing problems," *Mathematical Programming Computation*, vol. 6, no. 1, pp. 1–31, 2014.

[8] J. Gondzio, "Interior point methods 25 years later," *European Journal of Operational Research*, vol. 218, no. 3, pp. 587–601, may 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221711008204

[9] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[10] D. Sage, L. Donati, F. Soulez, D. Fortun, G. Schmit, A. Seitz, R. Guiet, C. Vonesch, and M. Unser, "DeconvolutionLab2: An open-source software for deconvolution microscopy," *Methods*, vol. 115, pp. 28–41, feb 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1046202316305096