INNER-LOOP-FREE ADMM FOR CRYO-EM

Laurène Donati, Emmanuel Soubies, Michael Unser

Biomedical Imaging Group, EPFL, Lausanne, Switzerland

ABSTRACT

Thanks to recent advances in signal processing, the interest for fast ℓ_1-regularized reconstruction algorithms in cryo-electron microscopy (cryo-EM) has intensified. The approaches based on the alternating-direction of multipliers method (ADMM) are particularly well-suited due to the prime convergence speed and flexibility of use of this algorithm. Yet, the standard ADMM scheme still relies on a nested conjugate gradient (CG) to solve the linear step in its alternating-minimization procedure, which can be costly when handling large-scale problems. In this work, we present an inner-loop-free ADMM algorithm for 3D reconstruction in cryo-EM. By using an appropriate splitting scheme, we are able to avoid the use of CG for solving the linear step. This leads to a substantial increase in algorithmic speed, as demonstrated by our experiments.

Index Terms— cryo-EM reconstruction, inverse problem, regularization, alternating-direction of multipliers method (ADMM), splitting, inner-loop-free, convergence speed.

1. INTRODUCTION

Cryo-electron microscopy (cryo-EM) aims at characterizing the three-dimensional (3D) structure of proteins at the atomic level [1]. The process starts with the imaging of numerous replicates of a macromolecule with nearly parallel electron rays at cryogenic temperatures. Software packages are then used to process the set of acquired two-dimensional (2D) measurements and produce a high-resolution 3D structure [2, 3, 4, 5, 6].

In most packages, the reconstruction is performed using direct methods based on the central-slice theorem [7, 8]. Direct approaches work adequately when the projections are sufficiently numerous and weakly degraded. Their speed is also a key advantage. Unfortunately, the use of direct methods is less appropriate when dealing with strongly ill-posed imaging conditions such as heavy noise, few projection measurements, or inaccurately known projection angles.

This drawback has led to a surge of robuster iterative schemes in the past years [9]. Among those, ℓ_1-regularized reconstruction algorithms perform particularly well in challenging imaging situations. However, as with most iterative schemes, they come with a prohibitive computational cost if not carefully engineered, which was up to recently hindering their wider use in cryo-EM.

Several works have improved this situation by proposing methods with increased speed. In particular, a breakthrough came when a costly step of many reconstruction algorithms was shown to be quickly computable as a discrete convolution [10, 11, 12, 13].

This development was essential to the deployment of the alternating-direction of multipliers method (ADMM) for iterative reconstruction in cryo-EM [13]. ADMM is a powerful splitting-based minimization algorithm with prime convergence speed [14]. Moreover, the alternating scheme decouples the physical aspects of the problem from the imposition of prior constraints on the signal, which permits a particularly modular implementation [15]. Yet, the standard ADMM scheme still relies on inner conjugate-gradient (CG) loops to solve the linear step in its minimization procedure. This leads to an algorithmic cost that grows with the number of inner CG loops required for reconstruction, which can be prohibitive.

In this work, we present an inner-loop-free ADMM algorithm for cryo-EM inspired by [16]. By using an appropriate splitting scheme, we are able to directly solve the linear step in ADMM without using CG. Experiments show that this improvement leads to a noticeable increase in algorithmic speed.

The paper is structured as follows: In Section 2, we describe the imaging model $\mathbf{H}$ and the fast formulation for the costly $\mathbf{H}^T\mathbf{H}$. In Section 3, we present our inner-loop-free ADMM algorithm, derived from a suitable splitting of the objective function. Finally, we provide in Section 4 results on the convergence speed of the proposed algorithm compared to the standard ADMM formulation.

1.1. Notations

Depending on the context, we write a continuous function f, $f(\cdot)$, or $f(\boldsymbol{x})$, where $\boldsymbol{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$. In this work, we shall either consider $d = 2$ or $d = 3$. Sequences are denoted by c or $c[\boldsymbol{k}]$ with $\boldsymbol{k} = (k_1, \dots, k_d) \in \mathbb{Z}^d$. Vectors are denoted by bold lowercase letters (*e.g.*, $\mathbf{c}$) and matrices by bold uppercase letters (*e.g.*, $\mathbf{H}$). The ℓ_1 and ℓ_2 norms of $\mathbf{c} = (\mathrm{c}_1, \dots, \mathrm{c}_N) \in \mathbb{R}^N$ are defined as $\|\mathbf{c}\|_1 = \sum_{n=1}^N |\mathrm{c}_n|$ and $\|\mathbf{c}\|_2 = \big(\sum_{n=1}^N |\mathrm{c}_n|^2\big)^{\frac{1}{2}}$, respectively. The spaces of finite-energy sequences and functions are denoted by $\ell_2(\mathbb{Z}^d)$ and $L_2(\mathbb{R}^d)$, respectively. The continuous convolution is written as $(f * g)(\boldsymbol{x}) = \int_{\mathbb{R}^d} f(\boldsymbol{\tau}) g(\boldsymbol{x} - \boldsymbol{\tau}) \mathrm{d}\boldsymbol{\tau}$. We make the distinction with the discrete convolution, denoted by $(c \star d)[\boldsymbol{m}] = \sum_{\boldsymbol{n} \in \mathbb{Z}^d} c[\boldsymbol{n}] d[\boldsymbol{m} - \boldsymbol{n}]$. We denote the reflection of a function as $f^\vee(\boldsymbol{x}) = f(-\boldsymbol{x})$. The proximal mapping associated to the convex functional $\mathcal{R}$ is defined as $\mathrm{prox}_{\mathcal{R}}(\mathbf{z}; \mu) = \operatorname{argmin}_{\mathbf{s}} \left\{ \frac{1}{2}\|\mathbf{s} - \mathbf{z}\|_2^2 + \mu \mathcal{R}(\mathbf{s}) \right\}$, with $\mu \in \mathbb{R}$.

2. IMAGING MODEL

Let $\mathcal{P}_{\boldsymbol{\theta}_p}\{f\}(\boldsymbol{y})$ with $\boldsymbol{y} \in \mathbb{R}^2$ denote the x-ray transform of the atomic density f for a pth 3D particle copy with orientation $\boldsymbol{\theta}_p$ (Euler angles) [17]. In cryo-EM, this entity is usually blurred by the point-spread function (PSF) w^p. We thus model the noiseless 2D cryo-EM measurements $\tilde{b}^p(\boldsymbol{y})$ of the pth particle as

$$\tilde{b}^p(\boldsymbol{y}) = (\mathcal{P}_{\boldsymbol{\theta}_p}\{f\} * w^p)(\boldsymbol{y}). \tag{1}$$

To discretize our object f, we use a generalized sampling scheme [18] where the reconstruction space is defined as

$$V(\varphi) = \left\{ \sum_{\boldsymbol{k} \in \mathbb{Z}^3} c[\boldsymbol{k}] \varphi(\cdot - \boldsymbol{k}) : c \in \ell_2\big(\mathbb{Z}^3\big) \right\}. \tag{2}$$

This work was funded by an ERC grant (ERC-692726-GlobalBioIm).

The coefficient sequence c thus corresponds to the discrete representation of the object $f = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} c[\boldsymbol{k}]\varphi(\cdot - \boldsymbol{k})$ in the space $V(\varphi)$.

In practice, as the object f is compactly supported, the sequence c is restricted to a finite number of nonzero coefficients. We write this vector of coefficients as $\mathbf{c} = \big(c[\boldsymbol{k}]\big)_{\boldsymbol{k}\in\Omega_{3\mathrm{D}}}$, where the set $\Omega_{3\mathrm{D}} \subset \mathbb{Z}^3$ corresponds to the support of the coefficients required to represent the object f. The number of elements in $\Omega_{3\mathrm{D}}$ is then denoted by $N = \#\Omega_{3\mathrm{D}}$, such that $\mathbf{c} \in \mathbb{R}^N$.

An appropriate choice for φ in (2) is the optimized Kaiser-Bessel window function (KBWF) [19, 20]. A KBWF represents functions very effectively when using specific parameter values (see [20]). Its isotropic property also allows for a significant reduction in computational costs.

Then, combining (1) and (2) and using the linearity and pseudo-shift-invariant properties of the x-ray transform [17], we obtain

$$\tilde{b}^p(\boldsymbol{y}) = \sum_{\boldsymbol{k}\in\Omega_{3\mathrm{D}}} c[\boldsymbol{k}]\big(\mathcal{P}_{\boldsymbol{\theta}_p}\{\varphi\} * w^p\big)(\boldsymbol{y} - \mathbf{M}_{\boldsymbol{\theta}_p^\perp}\boldsymbol{k}), \tag{3}$$

where the hyperplane projection matrix $\mathbf{M}_{\boldsymbol{\theta}_p^\perp} \in \mathbb{R}^{2\times 3}$ has rows that specify the normal basis of the hyperplane perpendicular to the direction $\boldsymbol{\theta}_p$ of integration. The measurements $\tilde{b}^p(\boldsymbol{y})$ for the pth particle are assumed to be acquired at the sampled points $\boldsymbol{y}_{\boldsymbol{j}} = \boldsymbol{j}\Delta$ for $\boldsymbol{j} \in \Omega_{2\mathrm{D}}$. Here, the set $\Omega_{2\mathrm{D}} \subset \mathbb{Z}^2$ denotes the support of the projection of f. The number of elements in $\Omega_{2\mathrm{D}}$ is denoted by $M = \#\Omega_{2\mathrm{D}}$. For the sake of clarity, we consider $\Delta = 1$ and we note $\tilde{\mathbf{b}}^p \in \mathbb{R}^M$ the discrete noiseless measurement vector for the pth particle. This gives us the entries of the imaging matrix $\mathbf{H}^p \in \mathbb{R}^{M\times N}$ as

$$[\mathbf{H}^p]_{\boldsymbol{j},\boldsymbol{k}} = \big(\mathcal{P}_{\boldsymbol{\theta}_p}\{\varphi\} * w^p\big)(\boldsymbol{j} - \mathbf{M}_{\boldsymbol{\theta}_p^\perp}\boldsymbol{k}). \tag{4}$$

In practice, each measurement $\tilde{\mathbf{b}}^p$ is corrupted by substantial additive Gaussian noise $\mathbf{n}^p$ [21]. Therefore, we finally obtain the discrete formulation of the complete forward model with P particles as

$$\mathbf{b} = \mathbf{H}\mathbf{c} + \mathbf{n}, \tag{5}$$

with

$$\mathbf{b} = \begin{bmatrix}\mathbf{b}^1\\ \mathbf{b}^2\\ \vdots\\ \mathbf{b}^P\end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix}\mathbf{H}^1\\ \mathbf{H}^2\\ \vdots\\ \mathbf{H}^P\end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix}\mathbf{n}^1\\ \mathbf{n}^2\\ \vdots\\ \mathbf{n}^P\end{bmatrix}, \tag{6}$$

where $P \in \mathbb{N}^*$, $\mathbf{b} \in \mathbb{R}^{MP}$, $\mathbf{H} \in \mathbb{R}^{MP\times N}$, and $\mathbf{n} \in \mathbb{R}^{MP}$.

2.1. Fast $\mathbf{H}^T\mathbf{H}$ Formulation

When the imaging model is described as in (4), we have a fast formulation for the product $\mathbf{H}^T\mathbf{H}$ [13].

Theorem 1. *Let $\varphi(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^3$ be such that $\widehat{\varphi}(\boldsymbol{\omega}) = 0$ for all $\|\boldsymbol{\omega}\| \geq \pi$. Moreover, let the imaging matrix $\mathbf{H}$ be as defined in (6) and let P denote the number of particles. Then, the discrete product $\mathbf{H}^T\mathbf{H}\mathbf{c}$ can be computed as the discrete convolution*

$$[\mathbf{H}^T\mathbf{H}\mathbf{c}]_{\boldsymbol{k}} = (c \star r)[\boldsymbol{k}] \tag{7}$$

for $\boldsymbol{k} \in \Omega_{3\mathrm{D}}$ and with kernel

$$r[\boldsymbol{k}] = \sum_{p=1}^{P} \big(\mathcal{P}_{\boldsymbol{\theta}_p}\{\varphi\} * \mathcal{P}_{\boldsymbol{\theta}_p}\{\varphi^\vee\} * q^p\big)(\mathbf{M}_{\boldsymbol{\theta}_p\perp}\boldsymbol{k}), \tag{8}$$

where the function $q^p(\boldsymbol{y}) = (w^p * (w^p)^\vee)(\boldsymbol{y})$ with $\boldsymbol{y} \in \mathbb{R}^2$ is the autocorrelation function of the PSF $w^p(\boldsymbol{y})$.

Computation	$\mathbf{H}\mathbf{c}$	$\mathbf{H}^T\mathbf{b}$	$\mathbf{H}^T\mathbf{H}\mathbf{c}$	fast $\mathbf{H}^T\mathbf{H}\mathbf{c}$ (7)
Runtime $[s]$	127.70	132.08	257.41	0.81

Table 1. Runtimes for applying the projection operator $\mathbf{H}$ and its variants $\mathbf{H}^T$, $\mathbf{H}^T\mathbf{H}$, and fast $\mathbf{H}^T\mathbf{H}$ (7) in a given experimental setup.

The benefit is that the costly matrix-vector product $\mathbf{H}^T\mathbf{H}\mathbf{c}$ can now be computed quickly as a pointwise multiplication in the Fourier domain (Table 1). This is an extremely valuable result as many optimization algorithms rely on computing this product at every iteration. Note also that the cost of the convolution in (7) does not depend on the number of projection directions, which is another significant advantage in cryo-EM.

In practice, the discrete convolution $c \star r$ in (7) only needs to be computed for $\boldsymbol{k} \in \Omega_{3\mathrm{D}}$. This convolution is computed in the Fourier domain using periodic boundary conditions. We do this by convolving a padded $\mathbf{c}$ with a kernel $\mathbf{r}$ of finite support. This kernel is obtained by first convolving the autocorrelation function of $\mathcal{P}_{\boldsymbol{\theta}_p}\{\varphi\}$ with the autocorrelation function of the PSF w^p, and then interpolating its value at the sampling points $\mathbf{M}_{\boldsymbol{\theta}_p\perp}\boldsymbol{k}$.

3. FAST ADMM-BASED ALGORITHMS

For cryo-EM reconstruction, the task is to minimize

$$\widehat{\mathbf{c}} = \underset{\mathbf{c}\in\mathbb{R}^N}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{H}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda\mathcal{R}(\mathbf{L}\mathbf{c}) + i_{\geq 0}(\mathbf{c}) \right\} \tag{9}$$

where the convex functional $\mathcal{R}$ associated to the regularization operator $\mathbf{L} \in \mathbb{R}^{N\times N}$ injects prior knowledge into the reconstruction process. A popular prior is the edge-preserving total-variation (TV) regularization ($\mathcal{R} = \|\cdot\|_{1,2}$, $\mathbf{L} = \nabla$) [22]. The constraint $i_{\geq 0}$ imposes nonnegativity of the reconstruction, while the regularization parameter $\lambda > 0$ sets the strength of the regularization.

With our fast formulation (7) for $\mathbf{H}^T\mathbf{H}$ in hand, the goal is then to favor optimization algorithms that solve (9) without applying $\mathbf{H}$ or $\mathbf{H}^T$ at every iteration. To support this reasoning, a comparison of the runtime of these operators is provided in Table 1. There, the forward operator was constructed from (4) as to compute 500 projections of a $(128\times128\times128)$ volume, with a KBWF as basis function. The time needed for computing the kernel (8) for the fast $\mathbf{H}^T\mathbf{H}$ is $118.07s$ (computed only once per reconstruction). For comparison purposes, we considered the cost of the projector without CTF correction.

A powerful algorithm that leverages (7) is ADMM [14]. Yet, as explained in Section 3.1, the standard ADMM still has a bottleneck since it relies on inner CG loops to solve the linear step at each iteration. To avoid these costly CG procedures, we present in Section 3.2 a splitting scheme that permits to directly compute the linear step.

3.1. ADMM with Inner CG Loops (Standard)

To solve (9) using ADMM, the problem is split by introducing two auxiliary variables $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$ [13] so that

$$\begin{aligned} &\widehat{\mathbf{c}} = \underset{\mathbf{c}\in\mathbb{R}^N}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{H}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda\mathcal{R}(\mathbf{u}) + i_{\geq 0}(\mathbf{v}) \right\} \\ &s.t. \quad \mathbf{u} = \mathbf{L}\mathbf{c}, \quad \mathbf{v} = \mathbf{c}. \end{aligned} \tag{10}$$

Algorithm 1 ADMM with inner CG loops (standard)

Require: $\mathbf{b} \in \mathbb{R}^{MP}$, $\rho > 0$

1: $\rho_{\mathrm{u}} = \rho_{\mathrm{v}} = \rho$
2: $k = 1$
3: **while** (ADMM not converged) **do**
4: $\quad \mathbf{u}^{k+1} = \mathrm{prox}_{\mathcal{R}}\big(\mathbf{L}\mathbf{c}^k - \frac{\widetilde{\mathbf{u}}^k}{\rho_{\mathrm{u}}}; \frac{\lambda}{\rho_{\mathrm{u}}}\big)$
5: $\quad \mathbf{v}^{k+1} = \mathrm{prox}_{i_{\geq 0}}\big(\mathbf{c}^k - \frac{\widetilde{\mathbf{v}}^k}{\rho_{\mathrm{v}}}; \frac{1}{\rho_{\mathrm{v}}}\big)$
6: $\quad$ **while** (CG not converged) **do**
7: $\quad\quad \mathbf{c}^{k+1} \leftarrow$ solve (11) using CG and fast $\mathbf{H}^T\mathbf{H}$
8: $\quad \widetilde{\mathbf{u}}^{k+1} = \widetilde{\mathbf{u}}^k + \rho_{\mathrm{u}}\big(\mathbf{u}^{k+1} - \mathbf{L}\mathbf{c}^{k+1}\big)$
9: $\quad \widetilde{\mathbf{v}}^{k+1} = \widetilde{\mathbf{v}}^k + \rho_{\mathrm{v}}\big(\mathbf{v}^{k+1} - \mathbf{c}^{k+1}\big)$
10: $\quad k \leftarrow k + 1$
11: **return** $\mathbf{c}^{k+1}$

In our case, the ADMM scheme for solving (9) consists of 5 alternating steps, as summarized in Algorithm 1. The proximal operators at Lines 4 and 5 admit closed-form expressions that can be computed efficiently [23]. Lines 8 and 9 correspond to simple updates of the dual variables $\widetilde{\mathbf{u}}, \widetilde{\mathbf{v}} \in \mathbb{R}^N$. The $\rho_{\mathrm{u}}, \rho_{\mathrm{v}} > 0$ are penalty parameters.

The bottleneck in Algorithm 1 is the quadratic minimization (or "linear step") at Line 7, which requires one to solve

$$\begin{aligned}\big(\rho_{\mathrm{u}}\mathbf{L}^T\mathbf{L} + \rho_{\mathrm{v}}\mathbf{I} + \mathbf{H}^T\mathbf{H}\big)\mathbf{c}^{k+1} =& \mathbf{H}^T\mathbf{b} + \rho_{\mathrm{v}}(\mathbf{v}^{k+1} + \widetilde{\mathbf{v}}^k/\rho_{\mathrm{v}}) \\ &+ \rho_{\mathrm{u}}\mathbf{L}^T(\mathbf{u}^{k+1} + \widetilde{\mathbf{u}}^k/\rho_{\mathrm{u}}) \quad (11)\end{aligned}$$

in terms of $\mathbf{c}$. The difficulty is that one cannot explicitly build the inverse of the matrix $\big(\rho_{\mathrm{u}}\mathbf{L}^T\mathbf{L} + \rho_{\mathrm{v}}\mathbf{I} + \mathbf{H}^T\mathbf{H}\big)$. Moreover, because the fast computation of $\mathbf{H}^T\mathbf{H}$ in (7) requires padding of the object prior to the convolution, (11) cannot be inverted in the Fourier domain. Hence, the standard ADMM scheme must rely on an iterative algorithm—classically CG—to solve (11), which can be costly if multiple inner CG loops are needed.[1]

3.2. Inner-Loop-Free ADMM (Proposed)

To avoid the use of CG for solving the linear step, we propose a novel splitting of (9) inspired by [16]. The idea is to add a tailored auxiliary variable $\mathbf{w} \in \mathbb{R}^N$ to (10) that simplifies the standard algorithm and makes the linear step directly solvable.

We formulate this splitting scheme as

$$\begin{aligned}&\widehat{\mathbf{c}} = \underset{\mathbf{c}\in\mathbb{R}^N}{\mathrm{argmin}}\left\{\frac{1}{2}\|\mathbf{H}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda\mathcal{R}(\mathbf{u}) + i_{\geq 0}(\mathbf{v})\right\} \quad (12)\\ &s.t. \quad \mathbf{u} = \mathbf{L}\mathbf{c}, \quad \mathbf{v} = \mathbf{c}, \quad \mathbf{w} = \mathbf{T}\mathbf{c},\end{aligned}$$

where $\mathbf{T} = (\mathbf{A} - \mathbf{H}^T\mathbf{H})^{1/2}$ with $\mathbf{A} \succeq \mathbf{H}^T\mathbf{H}$. (We impose that the matrix $\mathbf{A}$ dominates the term $\mathbf{H}^T\mathbf{H}$). In this work, we set $\mathbf{A} = \|\mathbf{H}\|^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

[1] Note that, although the discrete product $\mathbf{H}^T\mathbf{b}$ in (11) only needs to be computed once during the whole optimization procedure, it is also costly in its own rights. Recent works have shown that it was also possible to formulate it as a convolution [13, 24].

The augmented Lagrangian for (12) is then given by

$$\begin{aligned}\mathcal{L}\big(\mathbf{c}, \mathbf{u}, \widetilde{\mathbf{u}}, \mathbf{v}, \widetilde{\mathbf{v}}, \mathbf{w}, \widetilde{\mathbf{w}}\big) =& \frac{1}{2}\|\mathbf{H}\mathbf{c} - \mathbf{b}\|_2^2 \\ &+ \lambda\mathcal{R}(\mathbf{u}) + \langle\widetilde{\mathbf{u}}, \mathbf{u} - \mathbf{L}\mathbf{c}\rangle + \frac{\rho_{\mathrm{u}}}{2}\Big\|\mathbf{u} - \mathbf{L}\mathbf{c}\Big\|_2^2 \\ &+ i_{\geq 0}(\mathbf{v}) + \langle\widetilde{\mathbf{v}}, \mathbf{v} - \mathbf{c}\rangle + \frac{\rho_{\mathrm{v}}}{2}\|\mathbf{v} - \mathbf{c}\|_2^2 \\ &+ \langle\widetilde{\mathbf{w}}, \mathbf{w} - \mathbf{T}\mathbf{c}\rangle + \frac{\rho_{\mathrm{w}}}{2}\Big\|\mathbf{w} - \mathbf{T}\mathbf{c}\Big\|_2^2, \quad (13)\end{aligned}$$

where $\widetilde{\mathbf{w}}$ is the additional dual variable and ρ_{w} the associated penalty parameter. The ADMM alternating scheme is thus (temporarily) composed of 7 steps.

- Minimization *w.r.t.* the auxiliary variables $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$:

$$\mathbf{u}^{k+1} = \mathrm{prox}_{\mathcal{R}}\Big(\mathbf{L}\mathbf{c}^k - \frac{\widetilde{\mathbf{u}}^k}{\rho_{\mathrm{u}}}; \frac{\lambda}{\rho_{\mathrm{u}}}\Big) \quad (14)$$

$$\mathbf{v}^{k+1} = \mathrm{prox}_{i_{\geq 0}}\Big(\mathbf{c}^k - \frac{\widetilde{\mathbf{v}}^k}{\rho_{\mathrm{v}}}; \frac{1}{\rho_{\mathrm{v}}}\Big) \quad (15)$$

$$\begin{aligned}\mathbf{w}^{k+1} =& \underset{\mathbf{w}\in\mathbb{R}^N}{\arg\min}\left\{\frac{\rho_{\mathrm{w}}}{2}\Big\|\mathbf{w} - \mathbf{T}\mathbf{c}^k + \frac{\widetilde{\mathbf{w}}^k}{\rho_{\mathrm{w}}}\Big\|_2^2\right\} \\ =& \mathbf{T}\mathbf{c}^k - \frac{\widetilde{\mathbf{w}}^k}{\rho_{\mathrm{w}}}. \quad (16)\end{aligned}$$

- Minimization *w.r.t.* to the object $\mathbf{c}$:

$$\begin{aligned}\mathbf{c}^{k+1} = \underset{\mathbf{c}\in\mathbb{R}^N}{\arg\min}\Bigg\{&\frac{1}{2}\|\mathbf{H}\mathbf{c} - \mathbf{b}\|_2^2 + \frac{\rho_{\mathrm{v}}}{2}\Big\|\mathbf{v}^{k+1} - \mathbf{c} + \frac{\widetilde{\mathbf{v}}^k}{\rho_{\mathrm{v}}}\Big\|_2^2 \\ &+ \frac{\rho_{\mathrm{u}}}{2}\Big\|\mathbf{u}^{k+1} - \mathbf{L}\mathbf{c} + \frac{\widetilde{\mathbf{u}}^{k+1}}{\rho_{\mathrm{u}}}\Big\|_2^2 \\ &+ \frac{\rho_{\mathrm{w}}}{2}\Big\|\mathbf{w}^{k+1} - \mathbf{T}\mathbf{c} + \frac{\widetilde{\mathbf{w}}^k}{\rho_{\mathrm{w}}}\Big\|_2^2\Bigg\}. \quad (17)\end{aligned}$$

- Update of the dual variables $\widetilde{\mathbf{u}}$, $\widetilde{\mathbf{v}}$ and $\widetilde{\mathbf{w}}$:

$$\widetilde{\mathbf{u}}^{k+1} = \widetilde{\mathbf{u}}^k + \rho_{\mathrm{u}}\big(\mathbf{u}^{k+1} - \mathbf{L}\mathbf{c}^{k+1}\big) \quad (18)$$

$$\widetilde{\mathbf{v}}^{k+1} = \widetilde{\mathbf{v}}^k + \rho_{\mathrm{v}}\big(\mathbf{v}^{k+1} - \mathbf{c}^{k+1}\big) \quad (19)$$

$$\widetilde{\mathbf{w}}^{k+1} = \widetilde{\mathbf{w}}^k + \rho_{\mathrm{w}}\big(\mathbf{w}^{k+1} - \mathbf{T}\mathbf{c}^{k+1}\big). \quad (20)$$

Operations (14), (15), (18) and (19) are identical to Lines 4, 5, 8 and 9 in the standard ADMM algorithm, respectively. As currently stated, the proposed algorithm necessitates two steps more than the standard ADMM. The trick is that we can put (16) in (17), which then simplifies as:

$$\begin{aligned}\mathbf{c}^{k+1} = \underset{\mathbf{c}\in\mathbb{R}^N}{\arg\min}\Bigg\{&\frac{1}{2}\|\mathbf{H}\mathbf{c} - \mathbf{b}\|_2^2 + \frac{\rho_{\mathrm{u}}}{2}\Big\|\mathbf{u}^{k+1} - \mathbf{L}\mathbf{c} + \frac{\widetilde{\mathbf{u}}^{k+1}}{\rho_{\mathrm{u}}}\Big\|_2^2 \\ &+ \frac{\rho_{\mathrm{v}}}{2}\Big\|\mathbf{v}^{k+1} - \mathbf{c} + \frac{\widetilde{\mathbf{v}}^k}{\rho_{\mathrm{v}}}\Big\|_2^2 + \frac{\rho_{\mathrm{w}}}{2}\Big\|\mathbf{T}(\mathbf{c}^k - \mathbf{c})\Big\|_2^2\Bigg\}. \quad (21)\end{aligned}$$

Operations (16) and (20) can then be set aside as they are not used in any other alternating step. Moreover, if one sets $\rho_{\mathrm{w}} = 1$, the minimizer of (21) satisfies

$$\begin{aligned}\big(\rho_{\mathrm{u}}\mathbf{L}^T\mathbf{L} + \rho_{\mathrm{v}}\mathbf{I} + \mathbf{A}\big)\mathbf{c}^{k+1} =& \mathbf{H}^T\mathbf{b} + \rho_{\mathrm{u}}\mathbf{L}^T(\mathbf{u}^{k+1} + \widetilde{\mathbf{u}}^k/\rho_{\mathrm{u}}) \\ &+ \rho_{\mathrm{v}}(\mathbf{v}^{k+1} + \widetilde{\mathbf{v}}^k/\rho_{\mathrm{v}}) \\ &+ (\mathbf{A} - \mathbf{H}^T\mathbf{H})\mathbf{c}^k. \quad (22)\end{aligned}$$

Algorithm 2 Inner-loop-free ADMM (proposed)

Require: $\mathbf{b} \in \mathbb{R}^{MP}$, $\rho > 0$

$\rho_{\mathrm{u}} = \rho_{\mathrm{v}} = \rho$
$\mathbf{A} = \|\mathbf{H}\|^2 \mathbf{I}$
$k = 1$
while (not converged) **do**
  $\mathbf{u}^{k+1} = \mathrm{prox}_{\mathcal{R}}\left(\mathbf{L}\mathbf{c}^k - \frac{\widetilde{\mathbf{u}}^k}{\rho_{\mathrm{u}}}; \frac{\lambda}{\rho_{\mathrm{u}}}\right)$
  $\mathbf{v}^{k+1} = \mathrm{prox}_{\iota_{\geq 0}}\left(\mathbf{c}^k - \frac{\widetilde{\mathbf{v}}^k}{\rho_{\mathrm{v}}}; \frac{1}{\rho_{\mathrm{v}}}\right)$
  $\mathbf{c}^{k+1} \leftarrow$ solve (22) directly using fast $\mathbf{H}^T\mathbf{H}$
  $\widetilde{\mathbf{u}}^{k+1} = \widetilde{\mathbf{u}}^k + \rho_{\mathrm{u}}\left(\mathbf{u}^{k+1} - \mathbf{L}\mathbf{c}^{k+1}\right)$
  $\widetilde{\mathbf{v}}^{k+1} = \widetilde{\mathbf{v}}^k + \rho_{\mathrm{v}}\left(\mathbf{v}^{k+1} - \mathbf{c}^{k+1}\right)$
  $k \leftarrow k + 1$
return $\mathbf{c}^{k+1}$

The elegance of this formulation is that (22) can be solved directly without relying on a nested CG. Indeed, the matrix $\left(\rho_{\mathrm{u}}\mathbf{L}^T\mathbf{L} + \rho_{\mathrm{v}}\mathbf{I} + \mathbf{A}\right)$ is easily invertible in the Fourier domain.

For a volume of size N, this new scheme leads to a complexity of $\mathcal{O}(2N\log(N))$ per ADMM iteration. By comparison, the standard ADMM with K inner CG loops has a complexity of $\mathcal{O}(2(K+1)N\log(N))$ per iteration.

A pseudo-code of our inner-loop-free ADMM is presented in Algorithm 2.

4. EXPERIMENTS

We compared the convergence speed of the two algorithms by reconstructing synthetic data. We used as ground-truth a $(256\times256\times256)$ β-galactosidase volume [25]. We put ourselves in challenging imaging conditions, such as those faced at the beginning of the so-called "refinement procedure" in cryo-EM (*i.e.*, few data[2], fairly noisy projections, multiple angular misassignments). From the ground-truth, we computed 30 equi-distributed projections using the imaging model in (4) with a KBWF as basis function. We then degraded those projections with Gaussian noise, setting their SNR to 1, and added error on the projection directions prior to reconstruction. For the sake of simplicity, we did not consider CTF effects.

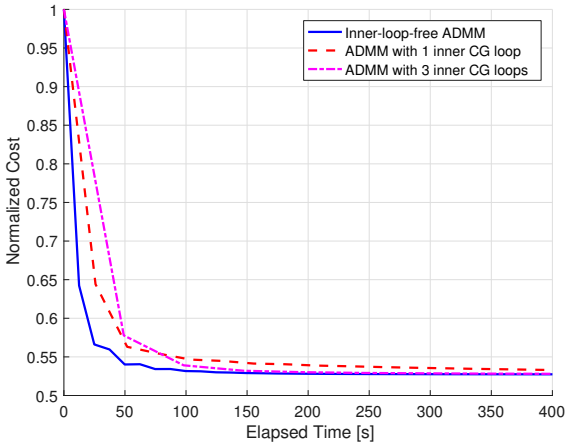
The reconstruction algorithms were implemented using the `GlobalBioIm Library`, an open-source library[3] for solving inverse problems [15]. The library contains generic modules that facilitate the implementation of imaging models and optimization algorithms.

We compared three ADMM configurations: two standard ADMM with 1 and 3 inner CG loops, and the inner-loop-free ADMM. The algorithm parameters (*i.e.*, the regularization parameter λ and the penalty parameter ρ with $\rho_{\mathrm{u}} = \rho_{\mathrm{v}} = \rho$) were tuned empirically to get the best reconstruction with the fastest convergence. We ran each algorithm for 200 iterations and recorded their convergence with respect to the elapsed time (Figure 1.a) and the number of iterations (Figure 1.b).

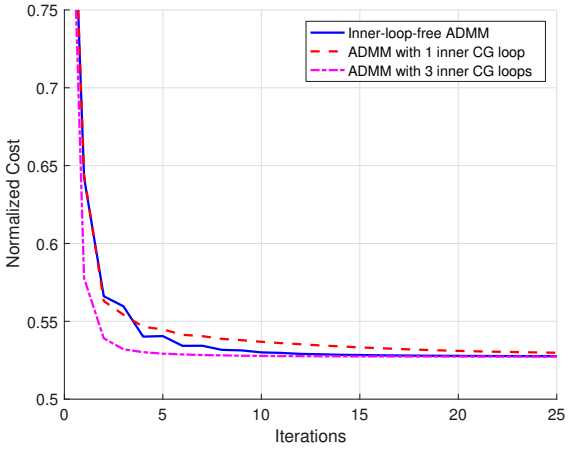
Overall, the results show that the proposed inner-loop-free ADMM provides a significant gain in algorithmic speed compared to the standard approach. As expected, the convergence of the standard ADMM accelerates with the number of CG iterations, but this also severely increases the execution time of the algorithm. The cost per iteration of the proposed ADMM algorithm is significantly lower, which permits faster computations.

[2]Most packages reconstruct high-resolution volumes by gradually adding information to a rough initial volume computed from very few class averages.

[3]`http://bigwww.epfl.ch/algorithms/globalbioim`



(a) Convergence with respect to time.



(b) Convergence with respect to iterations.

Fig. 1. Comparisons of the empirical convergence of the objective function (9) for three ADMM schemes: the standard ADMM with inner CG loops (resp. 1 and 3) and the proposed inner-loop-free ADMM. The proposed algorithm provides a substantial gain in algorithmic speed for cryo-EM reconstruction.

5. CONCLUSION

We presented an inner-loop-free ADMM algorithm for 3D reconstruction in cryo-EM. By using an appropriate splitting scheme, we eliminated the need for a nested CG scheme to solve the linear step in ADMM. We showed experimentally that this leads to a noticeable increase in algorithmic speed compared to the standard ADMM formulation.

6. REFERENCES

[1] J Dubochet, M Adrian, J Chang, J Homo, J Lepault, AW McDowall, and P Schultz, "Cryo-electron microscopy of vitrified specimens," *Quarterly Reviews of Biophysics*, vol. 21, no. 2, pp. 129–228, 1988.

[2] J Frank, B Shimkin, and H Dowse, "SPIDERA modular software system for electron image processing," *Ultramicroscopy*, vol. 6, no. 4, pp. 343–357, 1981.

[3] G Tang, L Peng, PR Baldwin, DS Mann, W Jiang, I Rees, and SJ Ludtke, "EMAN2: An extensible image processing suite for electron microscopy," *Journal of Structural Biology*, vol. 157, no. 1, pp. 38–46, 2007.

[4] SHW Scheres, "RELION: Implementation of a Bayesian approach to cryo-EM structure determination," *Journal of Structural Biology*, vol. 180, no. 3, pp. 519–530, 2012.

[5] JM De la Rosa-Trevín, A Quintana, L Del Cano, A Zaldivar, I Foche, J Gutiérrez, J Gómez-Blanco, J Burguet-Castell, J Cuenca-Alba, and V Abrishami, "Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy," *Journal of Structural Biology*, vol. 195, no. 1, pp. 93 – 99, 2016.

[6] A Punjani, JL Rubinstein, DJ Fleet, and MA Brubaker, "cryoSPARC: Algorithms for rapid unsupervised cryo-EM structure determination," *Nature Methods*, vol. 14, no. 3, pp. 290, 2017.

[7] PA Penczek, R Renka, and H Schomberg, "Gridding-based direct Fourier inversion of the three-dimensional ray transform," *Journal of the Optical Society of America A*, vol. 21, no. 4, pp. 499–509, 2004.

[8] V Abrishami, JR Bilbao-Castro, J Vargas, R Marabini, JM Carazo, and COS Sorzano, "A fast iterative convolution weighting approach for gridding-based direct Fourier three-dimensional reconstruction with correction for the contrast transfer function," *Ultramicroscopy*, vol. 157, pp. 79–87, 2015.

[9] Carlos Oscar S. Sorzano, J Vargas, J Otón, JM de la Rosa-Trevín, JL Vilas, M Kazemi, R Melero, L Del Caño, J Cuenca, P Conesa, and J Gomez-Blanco, "A survey of the use of iterative reconstruction algorithms in electron microscopy," *BioMed Research International*, vol. 2017, 2017.

[10] AH Delaney and Y Bresler, "A fast and accurate Fourier algorithm for iterative parallel-beam tomography," *IEEE Transactions on Image Processing*, vol. 5, no. 5, pp. 740–753, 1996.

[11] C Vonesch, L Wang, Y Shkolnisky, and A Singer, "Fast wavelet-based single-particle reconstruction in cryo-EM," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011*. IEEE, 2011, pp. 1950–1953.

[12] L Wang, Y Shkolnisky, and A Singer, "A Fourier-based approach for iterative 3d reconstruction from cryo-EM images," *arXiv preprint arXiv:1307.5824*, 2013.

[13] L Donati, COS Nilchian, Mand Sorzano, and M Unser, "Fast multiscale reconstruction for cryo-EM," *Journal of Structural Biology*, 2018.

[14] S Boyd, N Parikh, E Chu, B Peleato, and J Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[15] M Unser, E Soubies, F Soulez, M McCann, and L Donati, "GlobalBioIm: A unifying computational framework for solving inverse problems," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2017, pp. CTu1B–1.

[16] Y Zhu, "An augmented ADMM algorithm with application to the generalized lasso problem," *Journal of Computational and Graphical Statistics*, vol. 26, no. 1, pp. 195–204, 2017.

[17] F Natterer, *The Mathematics of Computerized Tomography*, Society for Industrial and Applied Mathematics, jan 2001.

[18] M Unser, "Sampling—50 years after Shannon," *Proceedings of IEEE*, vol. 88, no. 4, pp. 569–587, 2000.

[19] RM Lewitt, "Multidimensional digital image representations using generalized Kaiser–Bessel window functions," *Journal of the Optical Society of America A*, vol. 7, no. 10, pp. 1834–1846, 1990.

[20] M Nilchian, JP Ward, C Vonesch, and M Unser, "Optimized Kaiser–Bessel window functions for computed tomography," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3826–3833, 2015.

[21] COS Sorzano, LG De La Fraga, R Clackdoyle, and JM Carazo, "Normalizing projection images: A study of image normalizing procedures for single particle three-dimensional electron microscopy," *Ultramicroscopy*, vol. 101, no. 2-4, pp. 129–138, 2004.

[22] LI Rudin, S Osher, and E Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[23] PL Combettes and JC Pesquet, "A proximal decomposition method for solving convex variational inverse problems," *Inverse Problems*, vol. 24, no. 6, pp. 065014, 2008.

[24] J Andén and A Singer, "Structural variability from noisy tomographic projections," *SIAM Journal on Imaging Sciences*, vol. 11, no. 2, pp. 1441–1492, 2018.

[25] A Bartesaghi, D Matthies, S Banerjee, A Merk, and S Subramaniam, "Structure of β-galactosidase at 2.2Å resolution obtained by cryo-electron microscopy," *Proceedings of the National Academy of Sciences*, vol. 111, no. 32, pp. 11709–11714, 2014.