Anderson Acceleration for Nonconvex ADMM Based on Douglas-Rachford Splitting

Wenqing Ouyang[1]   Yue Peng[1,2]   Yuxin Yao[1]   Juyong Zhang[1†]   Bailin Deng[2]

[1]University of Science and Technology of China   [2]Cardiff University

Abstract

The alternating direction multiplier method (ADMM) is widely used in computer graphics for solving optimization problems that can be nonsmooth and nonconvex. It converges quickly to an approximate solution, but can take a long time to converge to a solution of high-accuracy. Previously, Anderson acceleration has been applied to ADMM, by treating it as a fixed-point iteration for the concatenation of the dual variables and a subset of the primal variables. In this paper, we note that the equivalence between ADMM and Douglas-Rachford splitting reveals that ADMM is in fact a fixed-point iteration in a lower-dimensional space. By applying Anderson acceleration to such lower-dimensional fixed-point iteration, we obtain a more effective approach for accelerating ADMM. We analyze the convergence of the proposed acceleration method on nonconvex problems, and verify its effectiveness on a variety of computer graphics including geometry processing and physical simulation.

CCS Concepts

• *Mathematics of computing → Solvers; Mathematical optimization; Numerical analysis;*

1. Introduction

Numerical optimization is commonly used in computer graphics, and finding a suitable solver is often instrumental to the performance of the algorithm. For an unconstrained problem with a simple smooth target function, gradient-based solvers such as gradient descent or the Newton method are popular choices [NW06]. On the other hand, for more complex problems, such as those with a nonsmooth target function or with nonlinear hard constraints, it is often necessary to employ more sophisticated optimization solvers to achieve the desired performance. For example, proximal splitting methods [CP11] are often used to handle nonsmooth optimization problems with or without constraints. The basic idea is to introduce auxiliary variables to replace some of the original variables in the target function, while enforcing consistency between the original variables and the auxiliary variables with a soft or hard constraint. This often allows to problem to be solved via alternating update of the variables, which reduces to simple sub-problems that can be solved efficiently. One example of such proximal splitting methods is the local-global solvers commonly used for geometry processing and physical simulation [SA07, LZX*08, BDS*12, LBOK13, BML*14].

Another popular type of proximal splitting methods, the alternating direction method of multipliers (ADMM) [BPC*11], is designed

---

† Corresponding author: juyong@ustc.edu.cn (Juyong Zhang)

for the following form of optimization:

$$\min_{\mathbf{x},\mathbf{z}} \; \Phi(\mathbf{x},\mathbf{z}) \qquad \text{s.t. } \mathbf{Ax} - \mathbf{Bz} = \mathbf{c}, \qquad (1)$$

where $\mathbf{x},\mathbf{z}$ are the original variable and the auxiliary variable, and the linear hard constraint $\mathbf{Ax} - \mathbf{Bz} = \mathbf{c}$ enforces their compatibility. ADMM computes a stationary point of the augmented Lagrangian function $L(\mathbf{x},\mathbf{z},\mathbf{y}) = \Phi(\mathbf{x},\mathbf{z}) + \langle \beta\mathbf{y}, \mathbf{Ax} - \mathbf{Bz} - \mathbf{c} \rangle + \frac{\beta}{2}\|\mathbf{Ax} - \mathbf{Bz} - \mathbf{c}\|^2$ via the following iterations [BPC*11]:

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}} \; L(\mathbf{x}_k,\mathbf{z},\mathbf{y}_k), \qquad (2)$$

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} \; L(\mathbf{x},\mathbf{z}_{k+1},\mathbf{y}_k), \qquad (3)$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{Ax}_{k+1} - \mathbf{Bz}_{k+1} - \mathbf{c}, \qquad (4)$$

where $\mathbf{y}$ is the dual variable, and $\beta \in \mathbb{R}^+$ is a penalty parameter. This formulation is general enough to represent a large variety of optimization problems. For example, any additional hard constraint can be incorporated into the target function using an indicator function that vanishes if the constraint is satisfied and has value $+\infty$ otherwise. The above iteration often has a low computational cost, where each sub-problem can be solved in parallel and/or in a closed form. The solver can handle nonsmooth problems, and typically converges to an approximate solution in a small number of iterations [BPC*11]. Moreover, although ADMM was initially designed for convex problems, it has proved to be also effective for many noncovex problems [WYZ19]. Such properties make it a popular solver for large-scale optimization in computer graphics [NVW*13,

NVT*14, OBLN17], computer vision [LFYL18, WG19], and image processing [FB10, AF13, HDN*16].

Despite its popularity, a major drawback of ADMM is that it can take a long time to converge to a solution of high accuracy. This limitation has motivated various work on accelerating ADMM with a focus on convex problem [GOSB14, KCSB15, ZW18]. For nonconvex ADMM, an acceleration technique was proposed recently in [ZPOD19]. By treating the steps (2)–(4) as a fixed-point iteration of the variables $(\mathbf{x}, \mathbf{y})$, it speeds up the convergence using Anderson acceleration [And65], a well-known acceleration technique for fixed-point iterations. It is also shown in [ZPOD19] that for problems with a separable target function that satisfies certain assumptions, ADMM can be treated as a fixed-point iteration on a reduced set of variables, which further reduces the overhead of Anderson acceleration.

In this paper, we propose a novel acceleration technique for nonconvex ADMM from a different perspective. We note that if the target function is separable in $\mathbf{x}$ and $\mathbf{z}$, then ADMM is equivalent to Douglas-Rachford (DR) splitting [DR56], a classical proximal splitting method. Such equivalence enables us to interpret ADMM using its equivalent DR splitting form, which turns out to be a fixed-point iteration for a linear transformation of the ADMM variables, with the same dimensionality as the dual variable $\mathbf{y}$. As a result, we can apply Anderson acceleration to such alternative form of fixed-point iteration, often with a much lower dimensionality than the fixed-point iteration of $(\mathbf{x}, \mathbf{y})$ that is utilized in [ZPOD19] for the general case and with a lower computational overhead. Moreover, compared to the other acceleration techniques in [ZPOD19] based on reduced variables, our new approach has the same dimensionality for the fixed-point iteration but requires a much weaker assumption on the optimization problem. To achieve stability of the Anderson acceleration, we propose two merit functions for determining whether an accelerated iterate can be accepted: 1) the DR envelope, with a strong guarantee for global convergence of the accelerated solver, and 2) the primal residual norm, which provides fewer theoretical guarantees but incurs lower computational overhead. As far as we are aware of, this is the first global convergence proof for Anderson acceleration on nonconvex ADMM. We evaluate our method on a variety of nonconvex ADMM solvers used in computer graphics and other domains. Thanks to its low dimensionality and strong theoretical guarantee, our method achieves more effective acceleration than [ZPOD19] on many of the experiments.

To summarize, our main contributions include:

- We propose an acceleration technique for nonconvex ADMM solvers, by utilizing their equivalence to DR splitting and applying Anderson acceleration to the fixed-point iteration form of DR splitting. We also propose two types of merit functions that can be used to verify the effectiveness of an accelerated iterate, as well as acceptance criteria for the iterate based on the merit functions.
- We prove the convergence of our accelerated solver under appropriate assumptions on the problem and the algorithm parameters.

2. Related Works

ADMM. ADMM is a variant of the augmented Lagrangian scheme that uses partial updates for the dual variables, and is commonly used for optimization problems with separable target functions and

linear side constraints [BPC*11]. Its ability to handle nonsmooth and constrained problems and its fast convergence to an approximate solution makes it a popular choice for large-scale optimization in various problem domains. In computer graphics, ADMM has been applied for geometry processing [BTP13, NVW*13, ZDL*14, XZZ*14, NVT*14], image processing [HDN*16], computational photography [WFDH18], and physical simulation [GITH14, PM17, OBLN17]. It is well known that ADMM suffers from slow convergence to a high-accuracy solution, and different strategies have been proposed in the past to speed up its convergence, e.g., using Nesterov's acceleration [GOSB14, KCSB15] or GMRES [ZW18]. However, these acceleration methods focus on convex problems, while many problems in computer graphics are nonconvex.

Anderson Acceleration. Anderson acceleration [And65, WN11] is an established method for accelerating fixed-point iterations, and has been applied successfully to numerical solvers in different domains, such as numerical linear algebra [Ste12, PSP16, SPP19], computational physics [LSV13, WTK14, AJW17, MST*18], and robotics [POD*18]. The key idea of Anderson acceleration is to utilize m previous iterates to construct a new iterate that converges faster to the fixed point. It has been noted that such an approach is indeed a quasi-Newton method [Eye96, FS09, RS11]. Other research works have investigated its local convergence [TK15, TEE*17] as well as its effectiveness in acceleration [EPRX20]. Recently, it has been applied in [PDZ*18a] to improve the convergence of local-global solvers in computer graphics. Later, Zhang et al. [ZPOD19] proposed to speed up the convergence of nonconvex ADMM solvers in computer graphics using Anderson acceleration.

DR Splitting. DR splitting was originally proposed in [DR56] to solve differential equations for heat conduction problems, and has been primarily used for solving separable convex problems. In recent years, there is a growing research interest in its application on nonconvex problems [ABT14, LP16, Pha16, HL13, HLN14]. The convergence of DR splitting in such scenarios has only been studied very recently [LP16, TP20]. In this paper, we will work with the same assumption as in [TP20] to analyze the convergence of our algorithm.

Similar to ADMM, DR splitting also needs a large number of iterations to converge to a solution of high accuracy [FZB19]. This has motivated research works on acceleration techniques for DR splitting, such as adaptive synchronization [BKW*19] and momentum acceleration [ZUMJ19]. Anderson acceleration and similar adaptive acceleration strategies have also been used to accelerate DR splitting [FZB19, PL19]. However, these works consider convex problems only, and their convergence proofs rely heavily on the convexity. Thus they are not applicable to the nonconvex problems considered in this paper.

The equivalence between ADMM and DR splitting is well known for convex problems [Glo83]. Some existing methods utilize this connection to accelerate ADMM [PJ16, PL19], but they are only applicable to convex problems. Our method is based on the equivalence between ADMM and DR splitting for nonconvex problems, which has only been established very recently [BK15, YY16, TP20].

3. Algorithm

In this section, we first introduce the background for ADMM, DR splitting, and Anderson acceleration. Then we discuss the equivalence between ADMM and DR splitting on nonconvex problems, and derive an Anderson acceleration technique for ADMM based on its equivalent DR splitting form.

3.1. Preliminary

ADMM.   In this paper, we focus on ADMM for the following optimization problem with a separable target function:

$$\min_{\mathbf{x},\mathbf{z}} \ f(\mathbf{x}) + g(\mathbf{z}) \qquad \text{s.t. } \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{z} = \mathbf{c}, \tag{5}$$

with the ADMM steps given by:

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} \left(f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{z}_k + \mathbf{y}_k - \mathbf{c}\|^2 \right), \tag{6}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{A}\mathbf{x}_{k+1} - \mathbf{B}\mathbf{z}_k - \mathbf{c}, \tag{7}$$

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}_{k+1} - \mathbf{B}\mathbf{z} + \mathbf{y}_{k+1} - \mathbf{c}\|^2 \right), \tag{8}$$

Throughout this paper, we assume that the solutions to sub-problems (6) and (8) always exist. Note that for each sub-problem, it is possible that there exist multiple solutions. Like [ZPOD19], we assume that the solver for each sub-problem is deterministic and always returns the same solution if given the same input, so that the operator argmin is single-valued. Although the order of steps here appears different from the standard scheme in Eqs. (3)–(4), they are actually equivalent since they have the same relative order between the steps. We adopt this notation instead of the standard scheme, because it facilitates our discussion about the equivalence with DR splitting. A commonly used convergence criterion for ADMM is that both the *primal residual* $\mathbf{r}_p^k$ and the dual residual $\mathbf{r}_d^k$ vanish [BPC*11]:

$$\mathbf{r}_p^k = \mathbf{A}\mathbf{x}_k - \mathbf{B}\mathbf{z}_{k-1} - \mathbf{c}, \quad \mathbf{r}_d^k = \beta\mathbf{B}^T\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1}). \tag{9}$$

The primal and dual residuals measure the violation of the linear side constraint and the dual feasibility condition of problem (5), respectively [BPC*11]. An alternative criterion is a vanishing *combined residual* [GOSB14]:

$$r_c^k = \beta\|\mathbf{A}\mathbf{x}_k - \mathbf{B}\mathbf{z}_{k-1} - \mathbf{c}\|^2 + \beta\|\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1})\|^2, \tag{10}$$

which is a sufficient condition for vanishing primal and dual residuals. Moreover, the combined residual decreases monotonically for convex problems [GOSB14].

DR splitting.   DR splitting has been used to solve optimization problems of the following form:

$$\min_{\mathbf{u}} \ \varphi_1(\mathbf{u}) + \varphi_2(\mathbf{u}), \tag{11}$$

with an iteration scheme:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \mathbf{v}_k - \mathbf{u}_k, \tag{12}$$

$$\mathbf{u}_{k+1} = \operatorname{prox}_{\gamma\varphi_1}(\mathbf{s}_{k+1}), \tag{13}$$

$$\mathbf{v}_{k+1} = \operatorname{prox}_{\gamma\varphi_2}(2\mathbf{u}_{k+1} - \mathbf{s}_{k+1}), \tag{14}$$

where $\gamma \in \mathbb{R}^+$ is a constant and prox_h denotes the proximal mapping of function h, i.e.,

$$\operatorname{prox}_h(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{y} \in \mathbb{R}^n} \left(h(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 \right). \tag{15}$$

Similar to our treatment of ADMM, we assume that there always exists a solution to the minimization problem above, and its solver always return the same result if given the same input, so that the proximal operator is single-valued. Although DR splitting has been primarily used on convex optimization, recent results show that it is also effective for noncovex problems [LP16]. Later in Section 3.2, we will show that the ADMM steps (6)–(8) are equivalent to the DR splitting scheme (12)–(14) for two functions φ_1, φ_2 derived from the target function and the linear constraint in Eq. (5).

Anderson Acceleration.   Given a fixed-point iteration

$$\mathbf{x}_{k+1} = G(\mathbf{x}_k),$$

Anderson acceleration [And65, WN11] aims at speeding up its convergence to a fixed point where the residual

$$F(\mathbf{x}) = G(\mathbf{x}) - \mathbf{x}$$

vanishes. Its main idea is to use the residuals of the latest step $\mathbf{x}_k$ and its previous m steps $\mathbf{x}_{k-1}, ..., \mathbf{x}_{k-m}$ to find a new step $\mathbf{x}_{k+1}^{AA}$ with a small residual. This is achieved via an affine combination of the images of $\mathbf{x}_k, \mathbf{x}_{k-1}, ..., \mathbf{x}_{k-m}$ under the fixed-point mapping G:

$$\mathbf{x}_{k+1} = G(\mathbf{x}_k) - \sum_{j=1}^{m} \theta_j^* \left(G(\mathbf{x}_{k-j+1}) - G(\mathbf{x}_{k-j}) \right),$$

where the coefficients are found by solving a least-squares problem:

$$(\theta_1^*, \dots, \theta_m^*) = \operatorname*{argmin}_{\theta_1, \dots, \theta_m} \left\| F(\mathbf{x}_k) - \sum_{j=1}^{m} \theta_j \left(F(\mathbf{x}_{k-j+1}) - F(\mathbf{x}_{k-j}) \right) \right\|^2 .$$

3.2. Anderson Acceleration Based on DR Splitting

The derivation of our acceleration method relies on the equivalence between ADMM and DR splitting from [TP20], which we will review in the following. To facilitate the presentation, we first introduce a notation from [TP20]:

Definition 3.1. Given $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $\mathbf{A} \in \mathbb{R}^{p \times n}$, the *image function* $f_{\mathbf{A}} : \mathbb{R}^p \to [-\infty, +\infty]$ is defined as

$$f_{\mathbf{A}}(\mathbf{x}) = \begin{cases} \inf_{\mathbf{y}}\{f(\mathbf{y}) \mid \mathbf{A}(\mathbf{y}) = \mathbf{x}\} & \text{if } \mathbf{x} \text{ is in the range of } \mathbf{A}, \\ +\infty & \text{otherwise.} \end{cases}$$

Note that we adopt a different symbol for image function than the one used in [TP20] to improve readability. The equivalence between ADMM and DR splitting is given as follows:

Proposition 3.2. *([TP20, Theorem 5.5]) Suppose* $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^p$, *and let* $(\mathbf{x}^+, \mathbf{y}^+, \mathbf{z}^+)$ *be generated by the ADMM iteration* (6)–(8) *from* $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. *Define*

$$\begin{cases} \mathbf{s} = \mathbf{A}\mathbf{x} - \mathbf{y} \\ \mathbf{u} = \mathbf{A}\mathbf{x} \\ \mathbf{v} = \mathbf{B}\mathbf{z} + \mathbf{c} \end{cases}, \qquad \begin{cases} \mathbf{s}^+ = \mathbf{A}\mathbf{x}^+ - \mathbf{y}^+ \\ \mathbf{u}^+ = \mathbf{A}\mathbf{x}^+ \\ \mathbf{v}^+ = \mathbf{B}\mathbf{z}^+ + \mathbf{c} \end{cases} . \tag{16}$$

Then we have:

$$\mathbf{s}^+ = \mathbf{s} + \mathbf{v} - \mathbf{u}, \tag{17}$$

$$\mathbf{u}^+ = \text{prox}_{\gamma\varphi_1}(\mathbf{s}^+), \tag{18}$$

$$\mathbf{v}^+ = \text{prox}_{\gamma\varphi_2}(2\mathbf{u}^+ - \mathbf{s}^+), \tag{19}$$

where $\gamma = 1/\beta$, and

$$\varphi_1(\mathbf{u}) = f_{\mathbf{A}}(\mathbf{u}), \quad \varphi_2(\mathbf{u}) = g_{\mathbf{B}}(\mathbf{u} - \mathbf{c}). \tag{20}$$

Proposition 3.2 shows that for the optimization problem (5), we can find the functions φ_1 and φ_2 in the problem (11) such that the DR splitting steps (12)–(14) are related to the ADMM steps (6)–(8) via the transformation defined in Eq. (16).

According to the DR splitting steps (13) and (14), both $\mathbf{u}_k$ and $\mathbf{v}_k$ are functions of $\mathbf{s}_k$. Then the step (12) indicates that $\mathbf{s}_{k+1}$ can be written as a function of $\mathbf{s}_k$ only:

$$\mathbf{s}_{k+1} = \mathcal{G}(\mathbf{s}_k) := \frac{1}{2}\Big((2\text{prox}_{\gamma\varphi_2} - \mathbf{I}) \circ (2\text{prox}_{\gamma\varphi_1} - \mathbf{I}) + \mathbf{I}\Big)(\mathbf{s}_k), \tag{21}$$

where $\mathbf{I}$ denote the identity operator. In other words, the DR splitting steps can be considered as a fixed-point iteration of $\mathbf{s}$, which is a transformation of the variables $\mathbf{x}$ and $\mathbf{y}$ for its equivalent ADMM solver. Therefore, we can apply Anderson acceleration to the $\mathbf{s}$ variable in DR splitting to speed up the convergence. One tempting approach is to compute the value of $\mathbf{s}$ according to Eq. (16) after each ADMM iteration and apply Anderson acceleration. This would not work in general, however, because from an accelerated value of $\mathbf{s}$ we cannot recover the values of $\mathbf{x}$ and $\mathbf{y}$ to carry on the subsequent ADMM steps. Instead, we perform Anderson acceleration on DR splitting, and derive the ADMM solution $\mathbf{x}, \mathbf{y}, \mathbf{z}$ based on the final values of the DR splitting variables $\mathbf{s}, \mathbf{u}, \mathbf{v}$. To implement this idea, we still need to resolve a few problems. First, we need to determine the specific forms of the proximal operators $\text{prox}_{\gamma\varphi_1}$ and $\text{prox}_{\gamma\varphi_1}$ used in DR splitting. Second, similar to [ZPOD19], we need to define criteria for the acceptance of an accelerated iterate, to improve the stability of Anderson acceleration. Finally, we need to find a way to recover the ADMM variables $\mathbf{x}, \mathbf{y}, \mathbf{z}$ after the termination of DR splitting. These problems will be discussed in the following.

3.2.1. Proximal Operators for $\gamma\varphi_1$ and $\gamma\varphi_2$

In general, given the functions f and g from the optimization problem (5), it is difficult to find an explicit formula for the image functions φ_1 and φ_2 given in Eq. (20). On the other hand, the proximal operators $\text{prox}_{\gamma\varphi_1}$ and $\text{prox}_{\gamma\varphi_2}$ have rather simple forms, as we will show below. Here and in the remaining parts of the paper, we will make frequent use of the following proposition from [TP20]:

Proposition 3.3. *([TP20, Proposition 5.2]) Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $\mathbf{A} \in \mathbb{R}^{p \times n}$. Suppose that for some $\beta > 0$ the set-valued mapping $\mathcal{X}_\beta(\mathbf{s}) := \underset{\mathbf{x} \in \mathbb{R}^n}{\text{argmin}}\{f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{s}\|^2\}$ is nonempty for all $\mathbf{s} \in \mathbb{R}^p$. Then*

(i) the image function $f_{\mathbf{A}}$ is proper;

(ii) $f_{\mathbf{A}}(\mathbf{A}\mathbf{x}_\beta) = f(\mathbf{x}_\beta)$ for all $\mathbf{s} \in \mathbb{R}^p$ and $\mathbf{x}_\beta \in \mathcal{X}_\beta(\mathbf{s})$;

(iii) $\text{prox}_{f_{\mathbf{A}}/\beta} = \mathbf{A}\mathcal{X}_\beta$.

Then from Proposition 3.3, it is easy to derive the following:

Proposition 3.4. *The proximal operators $\text{prox}_{\gamma\varphi_1}, \text{prox}_{\gamma\varphi_2}$ defined in Eqs. (18) and (19) can be evaluated as follows:*

$$\text{prox}_{\gamma\varphi_1}(\mathbf{s}) = \mathbf{A}\bar{\mathbf{x}}, \qquad \text{prox}_{\gamma\varphi_2}(2\mathbf{u} - \mathbf{s}) = \mathbf{B}\bar{\mathbf{z}} + \mathbf{c}, \tag{22}$$

where

$$\bar{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}}\Big(f(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{A}\mathbf{x} - \mathbf{s}\|^2\Big), \tag{23}$$

$$\bar{\mathbf{z}} = \underset{\mathbf{z}}{\text{argmin}}\Big(g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{B}\mathbf{z} + \mathbf{c} - (2\mathbf{u} - \mathbf{s})\|^2\Big). \tag{24}$$

3.2.2. Criteria for Accepting Accelerated Iterate

Classical Anderson acceleration can be unstable with slow convergence or stagnate at a wrong solution [WN11, PE13, PDZ*18b]. To improve stability, in [ZPOD19] an accelerated iterate is accepted only if it decreases a certain quantity that will converge to zero with effective iterations, such as the combined residual. Adopting a similar approach, we define a merit function ψ whose decrease indicates the effectiveness of an iteration. At the k-th iteration, we evaluate the un-accelerated iterate $\mathcal{G}(\mathbf{s}_{k-1})$ as well as the accelerated iterate $\mathbf{s}_{\text{AA}}$, and evaluate the decrease of the merit function from $\mathbf{s}_{k-1}$ to $\mathbf{s}_{\text{AA}}$:

$$d = \psi(\mathbf{s}_{\text{AA}}) - \psi(\mathbf{s}_{k-1}).$$

We choose $\mathbf{s}_{\text{AA}}$ as the new iterate if d meets a certain criterion, and revert to the un-accelerated iterate $\mathcal{G}(\mathbf{s}_{k-1})$ otherwise.

One choice of the merit function is

$$\psi_{\text{P}}(\mathbf{s}) := \|\mathbf{v}(\mathbf{s}) - \mathbf{u}(\mathbf{s})\|, \tag{25}$$

where $\mathbf{u}(\mathbf{s})$ and $\mathbf{v}(\mathbf{s})$ denote the $\mathbf{u}$ and $\mathbf{v}$ values produced by the DR splitting steps (13) and (14) from $\mathbf{s}$, i.e.,

$$\mathbf{u}(\mathbf{s}) = \text{prox}_{\gamma\varphi_1}(\mathbf{s}), \qquad \mathbf{v}(\mathbf{s}) = \text{prox}_{\gamma\varphi_2}(2\mathbf{u}(\mathbf{s}) - \mathbf{s}). \tag{26}$$

Note that according to Eq. (12), $\mathbf{v}(\mathbf{s}) - \mathbf{u}(\mathbf{s})$ measures the change in variable $\mathbf{s}$ between two consecutive iterations. Therefore, if $\mathbf{s}$ converges to a value $\mathbf{s}^*$, then $\psi_{\text{P}}(\mathbf{s})$ must converge to zero. Moreover, Proposition 3.2 indicates that $\|\mathbf{v} - \mathbf{u}\| = \|\mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{z} - \mathbf{c}\|$, which is the norm of the primal residual for the equivalent ADMM problem (5) [BPC*11]. We call $\psi_{\text{P}}(\mathbf{s})$ the *primal residual norm*, and accept an accelerated iterate if its primal residual norm is no larger than the previous iterate. Thus the decrease criterion is:

$$d \le 0. \tag{27}$$

An alternative merit function is the *DR envelope*:

$$\psi_{\text{E}}(\mathbf{s}) := \min_{\mathbf{w}}\Big(\varphi_1(\mathbf{u}(\mathbf{s})) + \varphi_2(\mathbf{w}) + \langle\nabla\varphi_1(\mathbf{u}(\mathbf{s})), \mathbf{w} - \mathbf{u}(\mathbf{s})\rangle + \frac{1}{2\gamma}\|\mathbf{w} - \mathbf{u}(\mathbf{s})\|^2\Big), \tag{28}$$

where $\mathbf{u}(\mathbf{s})$ is defined in Eq. (26). It is shown in [TP20, Theorem 4.1] that $\psi_{\text{E}}(\mathbf{s})$ decreases monotonically during DR splitting iterations under the following assumptions:

(A.1) φ_1 is L-smooth, σ-hypoconvex with $\sigma \in [-L, L]$.

(A.2) φ_2 is lower semicontinuous and proper.

(A.3) Problem (11) has a solution.

Here a function F is said to be L-smooth if it is differentiable and $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|^2 \ \forall \mathbf{x}, \mathbf{y}$. F is said to be σ-hypoconvex if it is differentiable and $\langle\nabla F(\mathbf{x}) - \nabla F(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle \ge \sigma\|\mathbf{x} - \mathbf{y}\|^2 \ \forall \mathbf{x}, \mathbf{y}$. F is said to be lower semicontinuous if $\liminf_{\mathbf{x} \to \mathbf{x}_0} F(\mathbf{x}) \ge F(\mathbf{x}_0) \ \forall \mathbf{x}_0$. F is said to be proper if $F(\mathbf{x}) > -\infty \ \forall \mathbf{x}$ and $F \not\equiv +\infty$. Under Assumptions (A.1)–(A.3), the DR envelope has a more simple form:

Proposition 3.5. *If Assumptions (A.1)–(A.3) hold, then*

$$\psi_{\mathrm{E}}(\mathbf{s}) = f(\bar{\mathbf{x}}) + g(\bar{\mathbf{z}}) + \frac{1}{\gamma}\langle \mathbf{s} - \mathbf{u}(\mathbf{s}), \mathbf{v}(\mathbf{s}) - \mathbf{u}(\mathbf{s})\rangle + \frac{1}{2\gamma}\|\mathbf{v}(\mathbf{s}) - \mathbf{u}(\mathbf{s})\|^2,$$
(29)

where $\bar{\mathbf{x}}, \bar{\mathbf{z}}$ *are defined in* (23) *and* (24) *respectively, and* $\mathbf{u}(\mathbf{s}), \mathbf{v}(\mathbf{s})$ *are defined in* (26).

A proof is given in Appendix B. Note that the values $\bar{\mathbf{x}}, \bar{\mathbf{z}}, \mathbf{u}(\mathbf{s}), \mathbf{v}(\mathbf{s})$ are already evaluated during the DR splitting iteration. Therefore, the actual cost for computing $\psi_{\mathrm{E}}(\mathbf{s})$ is the evaluation of functions f and g as well as two inner products, which only incurs a small overhead in many cases. Using the DR envelope as the merit function, we can enforce a more sophisticated decrease criterion that provides a stronger guarantee of convergence. Specifically, we require that $\mathbf{s}_{\mathrm{AA}}$ decreases the DR envelope sufficiently compared to $\mathbf{s}_{k-1}$:

$$d \leq -\nu_1\|\mathcal{G}(\mathbf{s}_{k-1}) - \mathbf{s}_{k-1}\|^2 - \nu_2\|\mathbf{s}_{\mathrm{AA}} - \mathbf{s}_{k-1}\|^2,$$
(30)

where ν_1, ν_2 are nonnegative constants. The convergence of our solver using such acceptance criterion is discussed in Theorems 4.4 and 4.6 in Section 4.

In this paper, unless stated otherwise, we use the DR envelope as the merit function to benefit from its convergence guarantee if the optimization problem satisfies the conditions given Theorems 4.4 or 4.6, and use the primal residual norm otherwise as it is an effective heuristic with lower overhead according to our experiments.

3.2.3. Recovery of $\mathbf{x}, \mathbf{y}, \mathbf{z}$

After the variable $\mathbf{s}$ converges to a fixed point $\mathbf{s}^*$ for the mapping $\mathcal{G}$, it is easy to recover the corresponding stationary point $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ for the ADMM problem. Before presenting the method, we first introduce the definition for the stationary points.

Definition 3.6. $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ is said to be a stationary point of (5) if

$$\mathbf{A}\mathbf{x}^* - \mathbf{B}\mathbf{z}^* = \mathbf{c}, \quad -\beta\mathbf{A}^T\mathbf{y}^* \in \partial f(\mathbf{x}^*), \quad \beta\mathbf{B}^T\mathbf{y}^* \in \partial g(\mathbf{z}^*),$$

where ∂f and ∂g denote the generalized subdifferentials of f and g [RW09, Definition 8.3], respectively. Our method for recovering $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ is based on the following:

Proposition 3.7. *Let* $\mathbf{s}^*$ *be a fixed point of* $\mathcal{G}$. *Define*

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\left(f(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{A}\mathbf{x} - \mathbf{s}^*\|^2\right)$$

$$\mathbf{u}^* = \mathbf{A}\mathbf{x}^*,$$

$$\mathbf{y}^* = \mathbf{u}^* - \mathbf{s}^*,$$

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}}\left(g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{B}\mathbf{z} + \mathbf{c} - (2\mathbf{u}^* - \mathbf{s}^*)\|^2\right).$$

Then $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ *is a stationary point of the problem* (5).

A proof is given in Appendix C. Note that the evaluation of $\mathbf{x}^*, \mathbf{z}^*$ has the same form as the intermediate values $\bar{\mathbf{x}}, \bar{\mathbf{z}}$ in Proposition 3.4 for evaluating the proximal operators in DR splitting. Therefore, during the DR splitting, we store the values of $\bar{\mathbf{x}}$ and $\bar{\mathbf{z}}$ when evaluating the proximal operators. When the variable $\mathbf{s}$ converges, we simply return the latest values of $\bar{\mathbf{x}}, \bar{\mathbf{z}}$ as the solution to the ADMM problem. Algorithm 1 summarizes our acceleration method.

Algorithm 1: Anderson Acceleration for ADMM based on DR splitting.

Data: $\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0$: initial values;
$m \in \mathbb{N}$: number of previous iterates used for acceleration;
$k_{\max}$: maximum number of iterations;
ε: convergence threshold.

1 $\mathbf{x}_{\mathrm{default}} = \mathbf{x}_0$; $\mathbf{z}_{\mathrm{default}} = \mathbf{z}_0$;
2 $\mathbf{s}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{y}_0$; $\mathbf{u}_0 = \mathbf{v}_0 = \mathbf{0}$; $\mathbf{s}_{\mathrm{default}} = \mathbf{s}_0$;
3 $k = 0$; $\psi_{\mathrm{prev}} = r = +\infty$; reset = TRUE;
4 **while** *TRUE* **do**
    // Perform one iteartion of DR splitting to evaluate merit function for $\mathbf{s}_k$
5 $\quad$ $\bar{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}}\left(f(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{A}\mathbf{x} - \mathbf{s}_k\|^2\right)$;
6 $\quad$ $\bar{\mathbf{u}} = \mathbf{A}\bar{\mathbf{x}}$;
7 $\quad$ $\bar{\mathbf{z}} = \operatorname{argmin}_{\mathbf{z}}\left(g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{B}\mathbf{z} + \mathbf{c} - (2\bar{\mathbf{u}} - \bar{\mathbf{s}})\|^2\right)$;
8 $\quad$ $\bar{\mathbf{v}} = \mathbf{B}\bar{\mathbf{z}} + \mathbf{c}$;
9 $\quad$ Compute ψ using Eq. (25) (or Eq. (28));
10 $\quad$ $d = \psi - \psi_{\mathrm{prev}}$;
   // Acceptance check for $\mathbf{s}_k$
11 $\quad$ **if** *reset == TRUE **OR** d satisfies condition* (27) *(or* (30)*)* **then**
   // Record the accepted iterate
12 $\quad\quad$ $\mathbf{x}_k = \mathbf{x}_{\mathrm{default}} = \bar{\mathbf{x}}$; $\mathbf{z}_k = \mathbf{z}_{\mathrm{default}} = \bar{\mathbf{z}}$; $\mathbf{u}_k = \mathbf{u}_{\mathrm{default}} = \bar{\mathbf{u}}$;
13 $\quad\quad$ $\mathbf{v}_k = \mathbf{v}_{\mathrm{default}} = \bar{\mathbf{v}}$; $\mathbf{s}_{\mathrm{default}} = \mathbf{s}_k$;
14 $\quad\quad$ $\psi_{\mathrm{prev}} = \psi$; reset = FALSE;
   // Compute accelerated iterate
15 $\quad\quad$ $\mathbf{g}_k = \mathbf{s}_k + \bar{\mathbf{v}} - \bar{\mathbf{u}}$; $\mathbf{f}_k = \mathbf{g}_k - \mathbf{s}_k$; $r = \|\mathbf{f}_k\|$; $\bar{m} = \min(m, k)$;
16 $\quad\quad$ $(\theta_1^*, \ldots, \theta_{\bar{m}}^*) = \underset{\theta_1, \ldots, \theta_{\bar{m}}}{\operatorname{argmin}} \left\|\mathbf{f}_k - \sum_{j=1}^{\bar{m}}\theta_j(\mathbf{f}_{k-j+1} - \mathbf{f}_{k-j})\right\|^2$;
17 $\quad\quad$ $\mathbf{s}_{\mathrm{AA}} = \mathbf{g}_k - \sum_{j=1}^{\bar{m}}\theta_j^*(\mathbf{g}_{k-j+1} - \mathbf{g}_{k-j})$;
   // Use $\mathbf{s}_{\mathrm{AA}}$ for next acceptance check
18 $\quad\quad$ $\mathbf{s}_{k+1} = \mathbf{s}_{\mathrm{AA}}$; $k = k+1$;
19 $\quad$ **else**
   // Revert to last accepted iterate
20 $\quad\quad$ $\mathbf{s}_k = \mathbf{s}_{\mathrm{default}}$; $\mathbf{u}_k = \mathbf{u}_{\mathrm{default}}$; $\mathbf{v}_k = \mathbf{v}_{\mathrm{default}}$;
21 $\quad\quad$ $\mathbf{x}_k = \mathbf{x}_{\mathrm{default}}$; $\mathbf{z}_k = \mathbf{z}_{\mathrm{default}}$; reset = TRUE;
22 $\quad$ **end if**
   // Check convergence
23 $\quad$ **if** $k \geq k_{\max}$ *OR* $r < \varepsilon$ **then**
24 $\quad\quad$ **return** $\mathbf{x}_{\mathrm{default}}, \mathbf{z}_{\mathrm{default}}$;
25 $\quad$ **end if**
26 **end while**

3.3. Discussion

3.3.1. Choice of Parameter m

As pointed out in [FS09], Anderson acceleration can be considered as a quasi-Newton method to find the root of the residual function, utilizing the m previous iterates to approximate the inverse Jacobian. Similar to other Anderson acceleration based methods such as [HS16, PDZ*18b, ZPOD19], we observe that a larger m leads to more reduction in the number of iterations required for convergence, but also increases the overhead per iteration. We empirically set $m = 6$ in all our experiments.

3.3.2. Comparison with [ZPOD19]

[ZPOD19] also proposed an Anderson acceleration approach for ADMM. In the general case, they treat the ADMM iteration (6)–(8) as a fixed-point iteration of $(\mathbf{x}, \mathbf{y})$. In comparison, Proposition 3.2 shows that our approach is based on a fixed-point iteration of $\mathbf{s} = \mathbf{Ax} - \mathbf{y}$, with a dimensionality up to 50% lower than $(\mathbf{x}, \mathbf{y})$. A main computational overhead for Anderson acceleration is $2m$ inner products between vectors with the same dimensionality as the fixed-point iteration variables [PDZ*18a]. Therefore, our approach incurs a lower overhead per iteration. The lower dimensionality of our formulation also indicates that it describes the inherent structure of ADMM in a more essential way. And we observe in experiments that such lower-dimensional representation can be more effective in reducing the number of iterations required for convergence. Together with the lower overhead per iteration, this often leads to faster convergence than the general approach from [ZPOD19].

It is also shown in [ZPOD19] that if there is a special structure in the problem (5), ADMM can be represented as a fixed-point iteration of $\mathbf{x}$ or $\mathbf{y}$ alone, which would have the same dimensionality as the fixed-point mapping we use in this paper. In this case, besides the general approach mentioned in the previous paragraph, Anderson acceleration can also be applied to $\mathbf{x}$ or $\mathbf{y}$ alone, often with similar performance to our approach. However, this formulation requires one of the two target function terms in (5) to be a strongly convex quadratic function, which is a strong assumption that limits its applicability. In comparison, our method imposes no special requirements on functions f and g, making it a more versatile approach for effective acceleration.

4. Convergence Analysis

If we utilize the DR envelope as the merit function in Algorithm 1, and use condition (30) to determine acceptance for an accelerated iterate, then it can be shown that Algorithm 1 converges to a stationary point to the optimization problem. In the following, we will discuss the conditions for such convergence. Unless stated otherwise, we assume that all the functions are lower semicontinuous and proper. In contrast to Section 3, we will write $\in$ instead of $=$ for the evaluation of proximal mappings and minimization subproblems, to indicate that our results are still applicable when these operators are multi-valued. We first introduce some definitions:

Definition 4.1. A point $\mathbf{s}^*$ is said to be a fixed point of the mapping $\mathcal{G}$ if $\mathbf{s}^* \in \mathcal{G}(\mathbf{s}^*)$.

Definition 4.2. A point $\mathbf{u}^*$ is said to be a stationary point of (11) if

$$0 \in \partial \varphi_1(\mathbf{u}^*) + \partial \varphi_2(\mathbf{u}^*).$$

Definition 4.3. A function F is said to be level-bounded if the set $\{\mathbf{x} : F(\mathbf{x}) \le \alpha\}$ is bounded for any $\alpha \in \mathbb{R}$.

Our first convergence result requires the following assumptions:

(B.1) The constants ν_1, ν_2 in condition (30) satisfy $\nu_1 > 0, \nu_2 \ge 0$.
(B.2) $\varphi_1 + \varphi_2$ is level-bounded.
(B.3) The constant $\gamma = 1/\beta$ satisfies $\gamma < \min\{\frac{1}{2\max\{-\sigma,0\}}, \frac{1}{L}\}$, where L and σ are defined in Assumption (A.1).
(B.4) The function $\overline{g}(\mathbf{z}) := g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{Bz} + \mathbf{c} - \mathbf{s}\|^2$ is level-bounded and bounded from below for any given $\mathbf{s}$.

Our first convergence result is then given as follows:

Theorem 4.4. *Suppose Assumptions (A.1)–(A.3) and (B.1)–(B.3) hold. Let $\{(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)\}$ be the sequence generated by Algorithm 1 using Eq. (30) as the acceptance condition. Then*

(a) $\{\psi_E(\mathbf{s}_k)\}$ *is monotonically decreasing and* $\|\mathbf{v}_k - \mathbf{u}_k\| \to 0$.
(b) The sequence $(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)$ *is bounded. If any subsequence* $\{\mathbf{s}_{k_i}\}$ *converges to a point* $\mathbf{s}^*$*, then* $\mathbf{s}^*$ *is a fixed point of* $\mathcal{G}$ *and* $\mathbf{u}^* = \text{prox}_{\gamma\varphi_1}(\mathbf{s}^*)$ *is a stationary point of* (11)*. Moreover, such a convergent subsequence must exist.*
(c) Suppose Assumption (B.4) is also satisfied. For any convergent subsequence $\{\mathbf{s}_{k_i}\}$ *in (b), let* $\{\mathbf{z}_{k_i}\}$ *be the corresponding subsequence generated by Algorithm 1, i.e.,*

$$\mathbf{z}_{k_i} \in \underset{\mathbf{z}}{\arg\min}\left(g(\mathbf{z}) + \frac{1}{2\gamma}\left\|\mathbf{Bz} + \mathbf{c} - (2\mathbf{u}(\mathbf{s}_{k_i}) - \mathbf{s}_{k_i})\right\|^2\right).$$

Then $\{\mathbf{z}_{k_i}\}$ *is bounded. Let* $\mathbf{z}^*$ *be a cluster point of* $\{\mathbf{z}_{k_i}\}$*, and define*

$$\mathbf{x}^* \in \underset{\mathbf{x}}{\arg\min} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{Ax} - \mathbf{s}^*\|^2, \qquad \mathbf{y}^* = \mathbf{u}^* - \mathbf{s}^*.$$

Then $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ *is a stationary point of* (5).

A proof is given in Appendix D.

Remark 4.5. Given a fixed point $\mathbf{s}^*$ of $\mathcal{G}$, we can also compute a stationary point (5) without the assumptions used in Theorem 4.4. The reader is referred to Appendix E for further discussion.

Theorem 4.4 shows the subsequence convergence of $\{(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)\}$ to a value corresponding to a stationary point. Next, we consider the global convergence of the whole sequence. We define

$$\mathcal{D}_\gamma(\mathbf{s}, \mathbf{u}, \mathbf{v}) = \varphi_1(\mathbf{u}) + \varphi_2(\mathbf{v}) + \frac{1}{\gamma}\langle \mathbf{s} - \mathbf{u}, \mathbf{v} - \mathbf{u}\rangle + \frac{1}{2\gamma}\|\mathbf{v} - \mathbf{u}\|^2.$$

Our global convergence results rely on the following assumptions:

(C.1) The constants ν_1, ν_2 used in condition (30) are positive.
(C.2) Function $\mathcal{D}_\gamma$ is sub-analytic.

The definition of a sub-analytic function can be found in [XY13]. Then we can show the following:

Theorem 4.6. *Suppose assumptions (A.1)–(A.3), (B.1)–(B.3) and (C.1)–(C.2) hold. Let $\{(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)\}$ be the sequence generated by Algorithm 1 using Eq. (30) as the acceptance condition. Then $\{(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)\}$ converges to $(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*)$, where $\mathbf{s}^*$ is a fixed-point of $\mathcal{G}$, and $\mathbf{v}^* = \mathbf{u}^* = \text{prox}_{\gamma\varphi_1}(\mathbf{s}^*)$.*

A proof is given in Appendix F.

Remark 4.7. A sufficient condition for Assumption (C.2) is that f and g are both semi-algebraic functions. In this case, φ_1 and φ_2 will both be semi-algebraic [TP20], thus $\mathcal{D}_\gamma$ is also semi-algebraic. Since a semi-algebraic function is also sub-analytic [XY13], $\mathcal{D}_\gamma$ will be a sub-analytic function. As noted in [ZPOD19], a large variety of functions used in computer graphics are semi-algebraic. Interested readers are referred to [ZPOD19] and [LP15] for further discussion.

Remark 4.8. If the functions f and g satisfy some further conditions, it can be shown that the convergence rate of $(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)$ is r-linear. The discussion relies on the KL property [ABS13] and is rather technical, so we leave it to Appendix F.

Remark 4.9. Assumption (A.1) requires the function f in (5) to be globally Lipschitz differentiable. When f is only locally Lipschitz differentiable, it is still possible to prove the convergence of Algorithm 1. One such example is given in Appendix (I).

4.1. Assumptions on f and g

Assumptions (A.1), (A.2) and (B.2) impose conditions on the functions φ_1 and φ_2 in (11). As there is no closed-form expression for φ_1 and φ_2 in general, these conditions can be difficult to verify. For practical purposes, we provide some conditions on the functions f and g that can ensure Assumptions (A.1), (A.2) and (B.2). These conditions are based on the results in [TP20, Section 5.4].

Proposition 4.10. *Suppose the problem* (5) *and the ADMM subproblems in* (6) *and* (8) *have a solution. Then the following conditions are sufficient for Assumptions (A.1), (A.2) and (B.2):*

(D.1) *f and g are proper and lower semicontinuous.*

(D.2) *One of the functions f and g is level-bounded, and the other is bounded from below.*

(D.3) *$\mathbf{A}$ is surjective.*

(D.4) *f satisfies one of the following conditions:*

1. *f is Lipschitz differentiable, and $\operatorname{argmin}_{\mathbf{x}}\{f(\mathbf{x}) \mid \mathbf{A}\mathbf{x} = \mathbf{s}\}$ is single-valued and Lipschitz continuous;*
2. *f is Lipschitz differentiable and convex;*
3. *f is differentiable, and $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{A}(\mathbf{x} - \mathbf{y})\|^2$ for any $\mathbf{x}, \mathbf{y}$ if $\nabla f(\mathbf{x})$ and $\nabla f(\mathbf{y})$ are in the range of $\mathbf{A}^T$.*

(D.5) *The function $\mathcal{Z}(\mathbf{s}) := \operatorname{argmin}_{\mathbf{z}}\{g(\mathbf{z}) \mid \mathbf{B}\mathbf{z} + \mathbf{c} = \mathbf{s}\}$ is locally bounded on the set $\mathcal{S} = \{\mathbf{B}\mathbf{z} + \mathbf{c} \mid g(\mathbf{z}) < +\infty\}$, i.e., for any $\mathbf{s} \in \mathcal{S}$ there exists a neighborhood O such that $\mathcal{Z}$ is bounded on O.*

A proof is given in Appendix G.

5. Numerical Experiments

We apply our method to a variety of problems to validate its effectiveness, focusing mainly on nonconvex problems in computer graphics. We describe each problem using the same variable names as in (5), so that its ADMM solver can be described by the steps (6)–(8). Different solvers are run using the same initialization. For each problem, we compare the convergence speed between the original ADMM solver, the accelerated solver (AA-ADMM) proposed in [ZPOD19], and our method. For each method we plot the combined residual (10) with respect to the iteration count and the computational time respectively, where a faster decrease of the combined residual indicates faster convergence. For ADMM and AA-ADMM, the combined residual is evaluated according to Eq. (10). For DR splitting, it can be evaluated using the values of $\mathbf{s}, \mathbf{u}, \mathbf{v}$ without recovering their corresponding ADMM variables. Using the notations and results from Proposition 3.2, we have

$$\mathbf{u}^+ - \mathbf{v} = \mathbf{A}\mathbf{x}^+ - \mathbf{B}\mathbf{z} - \mathbf{c}, \qquad \mathbf{u}^+ - \mathbf{u} = \mathbf{A}(\mathbf{x}^+ - \mathbf{x}).$$

Therefore, given an DR splitting iterate $(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)$, we evaluate the combined residual r_c^k by performing a partial iteration

$$\mathbf{s}' = \mathbf{s}_k + \mathbf{v}_k - \mathbf{u}_k, \qquad \mathbf{u}' = \operatorname{prox}_{\gamma\varphi_1}(\mathbf{s}')$$

and computing

$$r_c^k = \frac{1}{\gamma}\left(\|\mathbf{u}' - \mathbf{v}_k\|^2 + \|\mathbf{u}' - \mathbf{u}_k\|^2\right).$$

Similar to [ZPOD19], we normalize all combined residual values as follows to factor out the influence from the dimensionality and the value range of the variables:
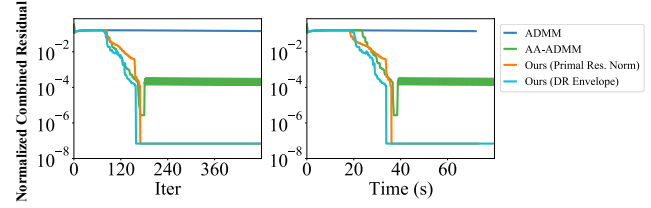
$$R = \sqrt{r_c / (N_{\mathbf{A}} \cdot a^2)}, \tag{31}$$

Figure 1: *Comparison between ADMM, AA-ADMM, and our method with different merit functions, using the ℓ_q-regularized logistic regression problem* (32). *The two variants of our method have similar performance. Both accelerate the convergence of ADMM and perform better than AA-ADMM.*

where $N_{\mathbf{A}}$ is the number of rows of matrix $\mathbf{A}$, and a is a scalar that indicates the typical range of variable values. For both AA-ADMM and our method, we use $m = 6$ previous iterates for Anderson acceleration. We adopt the implementation of Anderson acceleration from [PDZ*18a][†]. All experiments are run on a desktop PC with a hexa-core CPU at 3.7GHz and 16GB of RAM. The source codes for the examples are available at https://github.com/YuePengUSTC/AADR.

ℓ_q-Regularized Logistic Regression. First, we consider a sparse logistic regression problem from the ADMM demo code for [WYZ19][‡]:

$$\min_{\mathbf{x},\mathbf{z}} \quad p \cdot \lambda \cdot \Omega(\mathbf{z}^1) + \sum_{i=1}^{p} \log(1 + \exp(-b_i(\mathbf{a}_i^T\mathbf{w} + v))) \quad \text{s.t. } \mathbf{x} = \mathbf{z}. \tag{32}$$

Here $\mathbf{x} = (\mathbf{w}, v)$ are the parameters to be optimized, with $\mathbf{w} \in \mathbb{R}^n$ and $v \in \mathbb{R}$. $\mathbf{z} = (\mathbf{z}^1, z_2)$ is an auxiliary variable, with $\mathbf{z}^1 \in \mathbb{R}^n$ and $z^2 \in \mathbb{R}$. $\{(\mathbf{a}_i, b_i) \mid i = 1, \ldots, p\}$ is a set of input data pairs each consisting of a feature vector $\mathbf{a}_i \in \mathbb{R}^n$ and a label $b_i \in \{-1, 1\}$. $\Omega(\mathbf{z}^1) = \sum_{i=1}^{n} |z_i^1|^{1/2}$ is an ℓ_q sparsity regularization term with $q = \frac{1}{2}$. To test the performance, we use the data generator in the code to randomly generate $p = 1000$ pairs of data with feature vector dimension $n = 1000$. We test the problem with a weight parameter $\lambda = 10^{-4}$ and a penalty parameter $\beta = 10^5$. It can be verified that problem (32) satisfy the assumptions for Theorem 4.6 (see Appendix H). Thus we use the DR envelope as the merit function for Algorithm 1, with parameter $v_1 = v_2 = 10^{-3}$ for the acceptance condition (30). For comparison, we also run the algorithm using the primal residual norm as the merit function. We run AA-ADMM using the general approach in [ZPOD19] that accelerates $\mathbf{x}$ and the dual variable $\mathbf{y}$ simultaneously, since the problem does not meet their requirement for reduced-variable acceleration. Fig. 1 shows the comparison between the four solvers. We can see that both AA-ADMM and our methods can accelerate the convergence, while our methods achieve better performance thanks to the lower dimensionality of its accelerated variables. In addition, there is no significant difference between the performance of the two variants of our method, which verifies the effectiveness of the primal residual norm as the merit function despite its lack of convergence guarantee in theory.

[†] https://github.com/bldeng/AASolver
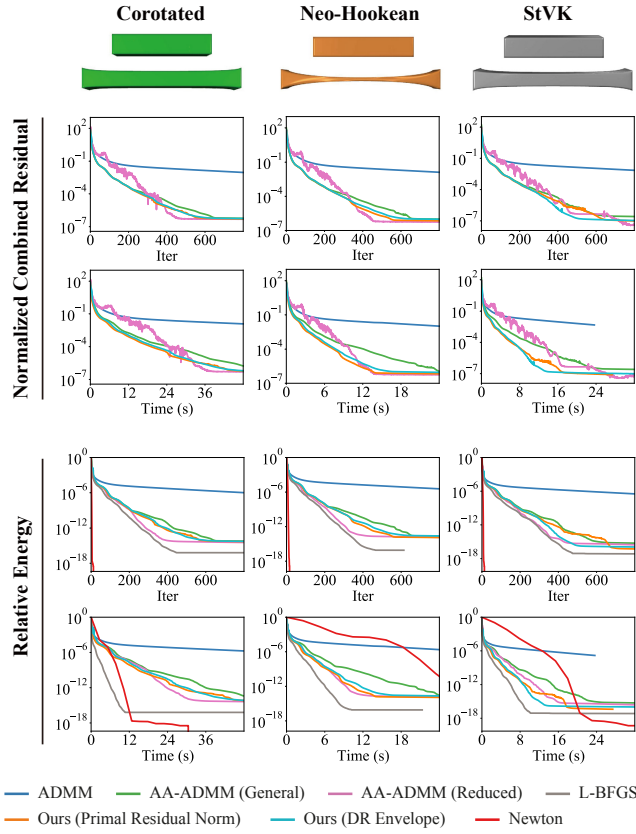[‡] https://github.com/shifwang/Nonconvex_ADMM_Demos

Figure 2: *Comparison using* (33) *for computing a frame in physical simulation of a stretched elastic bar with 6171 vertices and 25000 tetrahedrons, using three types of hyperelastic energy and a high stiffness parameter ('rubber' in the source code of [OBLN17]). The normalized combined residual plots (the top two rows) show that both variants of our method achieve similar acceleration results as the reduced-variable scheme of AA-ADMM. All three approaches perform better than the general scheme of AA-ADMM. The bottom two rows plot the relative energy* (35) *and include a Newton solver [SB12] and an L-BFGS solver for [LBK17] for comparison.*

Figure 3: *Computation of compressed manifold basis via problem* (36). *Our method achieves similar reduction of iterations as AA-ADMM, but outperforms AA-ADMM in computational time thanks to its lower overhead.*

Physical Simulation. Next, we consider the ADMM solver used in [OBLN17] for the following optimization for physical simulation:

$$\min_{\mathbf{x},\mathbf{z}} \; f(\mathbf{x}) + g(\mathbf{z}) \quad \text{s.t.} \; \mathbf{W}(\mathbf{x} - \mathbf{Dz}) = 0, \quad (33)$$

where $\mathbf{z}$ is the node positions to be optimized, $\mathbf{x}$ is an auxiliary variable that represents the absolute or relative node positions for the elements according to the selection matrix $\mathbf{D}$, $\mathbf{W}$ is a diagonal weight matrix, f is an elastic potential energy, and g is a quadratic momentum energy. In Appendix I, we use the StVK model as an example to prove the convergence of Algorithm 1 on problem (33). AA-ADMM can be applied to this problem to accelerate the variable $\mathbf{x}$ alone [ZPOD19], and we include both the general approach and the reduced-variable approach for comparison. For our method, we include the implementation using each merit function into the comparison, and choose parameter $\nu_1 = \nu_2 = 0$ for the acceptance condition (30). Fig. 2 shows the performance of the five solver
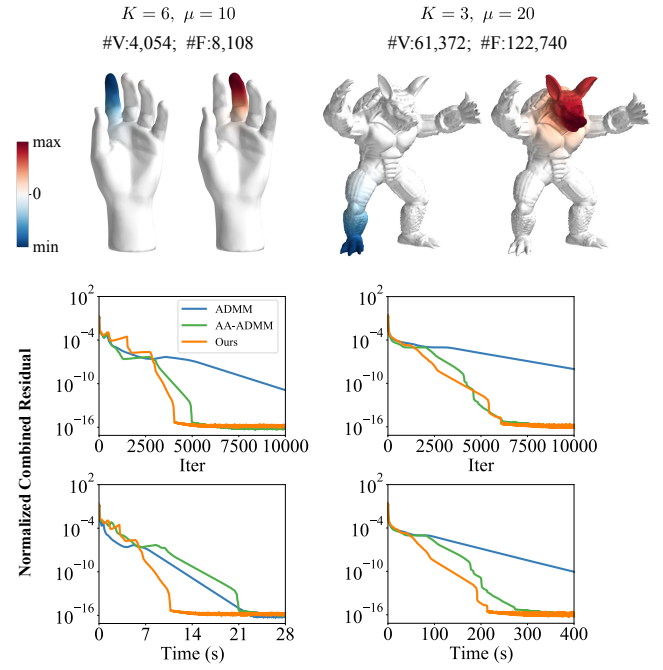
variants on the simulation of a stretched hyperelastic bar with a high stiffness parameter, using three types of hyperelastic energy. We adapt the source codes from [OBLN17][§] and [ZPOD19][¶] for the implementation of ADMM and AA-ADMM, respectively. The normalized combined residual plots (the top two rows) show that all accelerated variants achieve better performance than the ADMM solver. Overall, the general AA-ADMM takes a long time than other accelerated variants for full convergence, potentially due to the larger number of variables involved in the fixed-point iteration and the higher overhead they induce. For a more complete evaluation, we also compare the solvers with a Newton method [SB12] and an L-BFGS method [LBK17], neither of which suffers from slow final convergence. Specifically, we use them to minimize the following energy equivalent to the target function of (33):

$$F(\mathbf{z}) = f(\mathbf{Dz}) + g(\mathbf{z}). \quad (34)$$

In the bottom two rows of Fig. 2, we compare all methods by plotting their relative energy

$$E = (F - F^*)/(F_0 - F^*), \quad (35)$$

with respect to the iteration count and computational time, where F_0 and F^* are the initial value and the minimum of the energy F, respectively. We can see that although the Newton method requires the fewest iterations to convergence, it is one of the slowest methods

---

§ https://github.com/mattoverby/admm-elastic
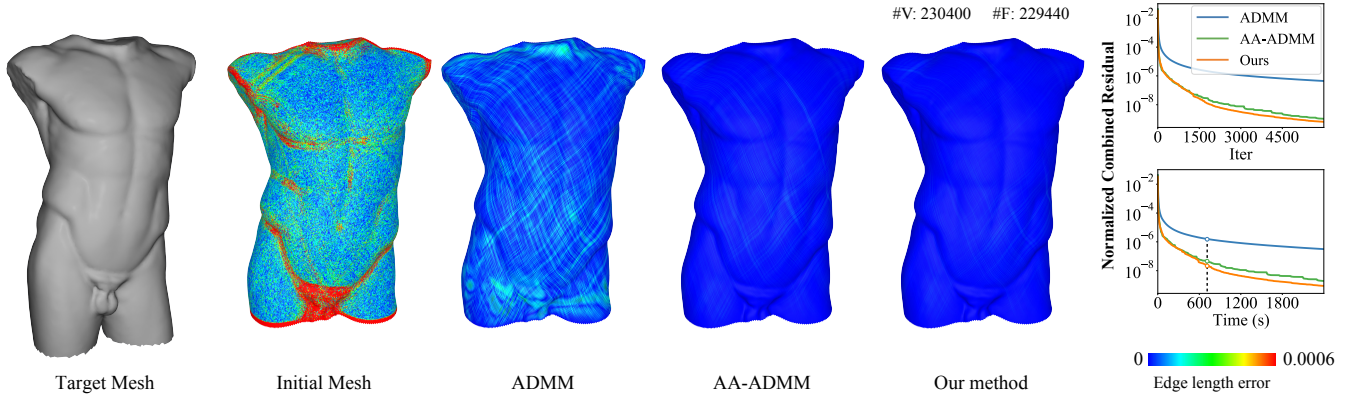
¶ https://github.com/bldeng/AA-ADMM

Figure 4: *Comparison between ADMM and accelerated methods on a wire mesh optimization problem* (38). *The normalized combined residual plots show faster convergence using the accelerated solvers and better performance with our method. The color-coding visualizes the edge length error ξ defined in* (39) *on meshes computed by the three methods within the same computational time (see the bottom-right plot).*

in terms of actual computational time, due to its high computational cost per iteration. L-BFGS achieves the best performance in terms of computational time, followed by the accelerated ADMM solvers. Note, however, that classical Newton and L-BFGS are intended for smooth unconstrained optimization problems, and they are often not applicable if the problem is nonsmooth or constrained — the type of problems that ADMM is popular for.

Geometry Processing. Nonconvex ADMM solvers have also been used in geometry processing. In Fig. 3, we compare the performance between different methods on the following optimization problem from [NVT*14] for compressed manifold modes on a triangle mesh with N vertices:

$$\min_{\mathbf{X},\mathbf{Z}} \text{Tr}((\mathbf{X}^1)^T \mathbf{L} \mathbf{X}^1) + \mu \|\mathbf{X}^2\|_1 + \iota(\mathbf{Z}) \quad \text{s.t. } \mathbf{Z} = \mathbf{X}^1, \mathbf{Z} = \mathbf{X}^2, \quad (36)$$

where $\mathbf{Z} \in \mathbb{R}^{N \times K}$ denotes a set of basis functions to be optimized, $\mathbf{X}^1, \mathbf{X}^2 \in \mathbb{R}^{N \times K}$ are auxiliary variables, $\mathbf{L} \in \mathbb{R}^{N \times N}$ is a Laplacian matrix, and ι is an indicator function of $\mathbf{Z}$ for enforcing the orthogonality condition if $\mathbf{Z}^T \mathbf{D} \mathbf{Z} = \mathbf{I}$ with respect to a mass matrix $\mathbf{D}$. We apply our method with the primal residual norm as the merit function. We use the source code released by the authors[||] for the ADMM solver, and modify it to implement AA-ADMM and our method. We use the general approach of AA-ADMM that accelerates $\mathbf{X}$ together with the dual variable, as the problem does not meet the requirement for reduced-variable acceleration. Fig. 3 shows the combined residual plots for the three methods on two models as well as the parameter settings for each problem instance. Our method achieves a similar effect in reducing the number of iterations as AA-ADMM, but outperforms AA-ADMM in terms of computational time thanks to its lower computational overhead.

We also apply our method to a problem proposed in [DBD*15] for optimizing the vertex positions $\mathbf{x} \in \mathbb{R}^{3n}$ of a mesh model subject to a set of soft constraints $\mathbf{A}_i \mathbf{x} \in C_i$ ($i \in \mathcal{S}$) and hard constraints $\mathbf{A}_j \mathbf{x} \in C_j$ ($j \in \mathcal{H}$), where matrices $\mathbf{A}_i$ and $\mathbf{A}_j$ select the relevant

vertices for the constraints and compute their differential coordinates where appropriate, and C_i and C_j represent the feasible sets. This is formulated in [DBD*15] as the following optimization:

$$\min_{\mathbf{x},\mathbf{z}} \quad \frac{1}{2} \|\mathbf{L}(\mathbf{x} - \tilde{\mathbf{x}})\|^2 + \sum_{i \in \mathcal{S}} \left(\frac{w_i}{2} \|\mathbf{A}_i \mathbf{x} - \mathbf{z}_i\|^2 + \sigma_{C_i}(\mathbf{z}_i) \right) + \sum_{j \in \mathcal{H}} \sigma_{C_j}(\mathbf{z}_j)$$

$$\text{s.t.} \quad \mathbf{A}_j \mathbf{x} - \mathbf{z}_j = \mathbf{0} \quad \forall j \in \mathcal{H}. \quad (37)$$

where $\mathbf{z}_i$ ($i \in \mathcal{S}$) and $\mathbf{z}_j$ ($j \in \mathcal{H}$) are auxiliary variables, σ_{C_i} and σ_{C_j} are indicator functions for the feasible sets, and w_i are user-specified weights. The first term of the target function is an optional Laplacian smoothness energy, whereas the second term measures the violation of the soft constraints using the squared Euclidean distance to the feasible sets. This problem is solved using ADMM and AA-ADMM in [ZPOD19]. However, since its target function is not separable, our accelerated ADMM solver is not applicable. To apply our method, we reformulate the problem as follows:

$$\min_{\mathbf{x},\mathbf{z}} \quad \frac{1}{2} \|\mathbf{L}(\mathbf{x} - \tilde{\mathbf{x}})\|^2 + \sum_{i \in \mathcal{S}} \frac{w_i}{2} \left(D_{C_i}(\mathbf{z}_i) \right)^2 + \sum_{j \in \mathcal{H}} \sigma_{C_j}(\mathbf{z}_j)$$

$$\text{s.t.} \quad \mathbf{A}_i \mathbf{x} = \mathbf{z}_i \quad \forall i \in \mathcal{S}, \quad \mathbf{A}_j \mathbf{x} = \mathbf{z}_j \quad \forall j \in \mathcal{H}, \quad (38)$$

where $D_{C_i}(\cdot)$ denotes the Euclidean distance to C_i. This problem has a separable target function, and we derive its ADMM solver in Appendix A. We compare the performance of ADMM, AA-ADMM and our method on problem (38) for wire mesh optimization [GSD*14]: we optimize a regular quad mesh subject to the soft constraints that each vertex lies on a target shape, and the hard constraints that (1) each edge has the same length l and (2) all angles of each quad face are within the range $[\pi/4, 3\pi/4]$. In Fig. 4, We solve the problem on a mesh with 230K vertices, using $\mathbf{L} = \mathbf{0}$, $w_i = 1$, and penalty parameter $\beta = 10000$. The combined residual plots show that both AA-ADMM and our method and our method achieve faster convergence than ADMM, with a slightly better performance from our method. To illustrate the benefit of such acceleration, we take the results generated by each method within the same computational time, and use color-coding to visualize the edge-length error
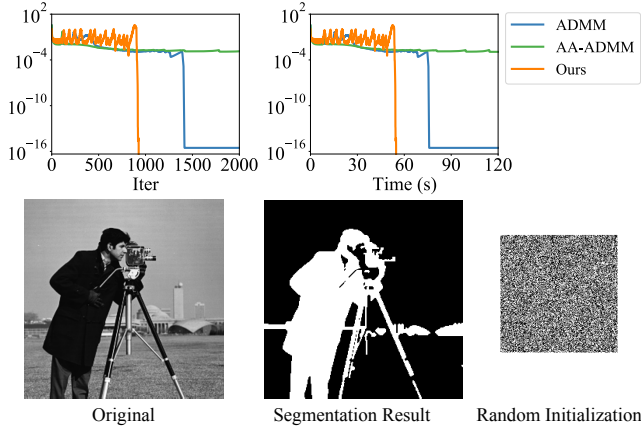
$$\xi(e) = |e - l|/l \quad (39)$$

[||] https://github.com/tneumann/cmm

Original   Segmentation Result   Random Initialization

Figure 5: *Comparison on the image segmentation problem* (40) *with a re-formulated binary constraint. Our method reduces the iteration count and computational time required for convergence, while AA-ADMM fails to achieve acceleration.*

Figure 6: *Comparison on a convex problem* (41) *with λ = 2, for computing local mesh deformation components from an input mesh sequence and given weights. The methods are tested using two mesh sequences constructed from the facial expression dataset of [RBSB18], with 100 frames and 250 frames, respectively. We set the penalty parameter to β = 10 for both problem instances. Our method have similar acceleration performance as AA-ADMM in reducing the number of iterations, and outperforms AA-ADMM in actual computational time.*

where e is the actual length for each edge. We can see that the two accelerated solvers lead to notably smaller edge-length errors than ADMM within the same computational time. The acceleration brings significant savings in computational time needed for a high-accuracy solution, which is required for the physical fabrication of the design [GSD*14].

Image Processing. In Fig. 5, we test our method on the non-convex ADMM solver for the following image segmentation problem [WG19]:

$$\min_{\mathbf{x},\mathbf{z}} \mathbf{x}^T \mathbf{L} \mathbf{x} + \mathbf{d}^T \mathbf{x} + \iota_1(\mathbf{z}^1) + \iota_2(\mathbf{z}^2) \quad \text{s.t. } \mathbf{z}^1 = \mathbf{x}, \ \mathbf{z}^2 = \mathbf{x}, \quad (40)$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the pixel-wise labels to be optimized, $\mathbf{L}$ is a Laplacian matrix based on the similarity between adjacent pixels, $\mathbf{d}$ is a unary cost vector, and $\mathbf{z} = (\mathbf{z}^1, \mathbf{z}^2)$ is an auxiliary variable with $\mathbf{z}^1, \mathbf{z}^2 \in \mathbb{R}^n$. ι_1 and ι_2 are indicator functions for the feasible sets $\mathcal{S}_1 = [0,1]^n$ and $\mathcal{S}_2 = \{\mathbf{p} \in \mathbb{R}^n \mid \sum_{i=1}^n (p_i - \frac{1}{2})^2 = \frac{n}{4}\}$ respectively, which together with the linear constraint between $\mathbf{x}$ and $\mathbf{z}$ induces a binary constraint for the labels $\mathbf{x}$. Fig. 5 uses the cameraman image to compare ADMM, AA-ADMM, and our method with the primal residual norm as the merit function, using the same random initialization. We use the python source code released by the authors** for the ADMM implementation, and modify it to implement AA-ADMM and our method. We use the general approach of AA-ADMM since the problem does not meet the reduced-variable conditions. The released code gradually changes the penalty parameter β, starting with $\beta = 5$ and increasing it by 3% every five iterations until it reaches the upper bound 1000. Since a different value of β will lead to a different fixed-point iteration, for both AA-ADMM and our method we reset the history of Anderon acceleration when β changes. We observe an interesting behavior of the ADMM solver: initially it maintains a relatively high value of the combined residual norm until the variable $\mathbf{z}$ converges to its value $\mathbf{z}^*$ in the solution; afterwards,
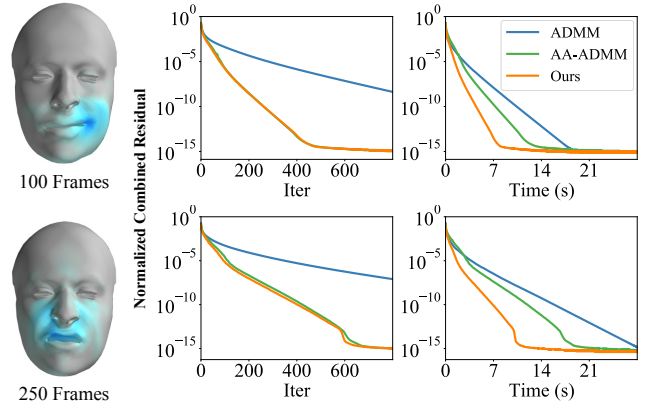
$\mathbf{z}$ remains close to $\mathbf{z}^*$, and the ADMM iteration effectively reduces to an affine transformation for the variables $\mathbf{x}$ and $\mathbf{y}$ with a rapid decrease of the combined residual norm. In comparison, our method shows more oscillation of the combined residual norm in the initial stage but accelerates the convergence of $\mathbf{z}$ towards $\mathbf{z}^*$, followed by a similar rapid decrease of the combined residual norm, thus outperforming ADMM in both iteration count and computational time. On the other hand, AA-ADMM fails to achieve acceleration.

Convex Problems. Although our method is designed with non-convex problems in mind, it can be naturally applied to convex problems. In Fig. 6, we apply our method to the ADMM solver in [NVW*13] for computing mesh deformation components given a mesh animation sequence and component weights:

$$\operatorname*{argmin}_{\mathbf{X},\mathbf{Z}} \|\mathbf{V} - \mathbf{W}\mathbf{Z}\|_F^2 + \lambda \cdot \Omega_1(\mathbf{X}) \quad \text{s.t. } \mathbf{X} = \mathbf{Z}, \quad (41)$$

where matrix $\mathbf{Z}$ represents the deformation components to be optimized, $\mathbf{V}$ is the input mesh sequence, $\mathbf{W}$ represents the given weights for the components, $\mathbf{X}$ is an auxiliary variable, and $\Omega_1(\mathbf{X})$ is a weighted ℓ_1/ℓ_2-norm to induce local support for the deformation components. In Fig. 7, we accelerate the ADMM solver in [HDN*16] for image deconvolution:

$$\operatorname*{argmin}_{\mathbf{x},\mathbf{z}} \|\mathbf{x}_1 - \mathbf{f}\|^2 + \lambda \cdot \Omega_2(\mathbf{x}_2) \quad \text{s.t. } \mathbf{K}\mathbf{z} = \mathbf{x}_1, \ \mathbf{G}\mathbf{z} = \mathbf{x}_2, \quad (42)$$

where $\mathbf{z}$ represents the image to be recovered, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ are auxiliary variables, matrix $\mathbf{K}$ represents the convolution operator, $\mathbf{G}$ is the image gradient matrix, and Ω_2 is the ℓ_1/ℓ_2-norm for regularizing the image gradients. Both problems (41) and (42) are convex, and AA-ADMM can only be applied using the general approach due to the problem structures. For both problems, we apply
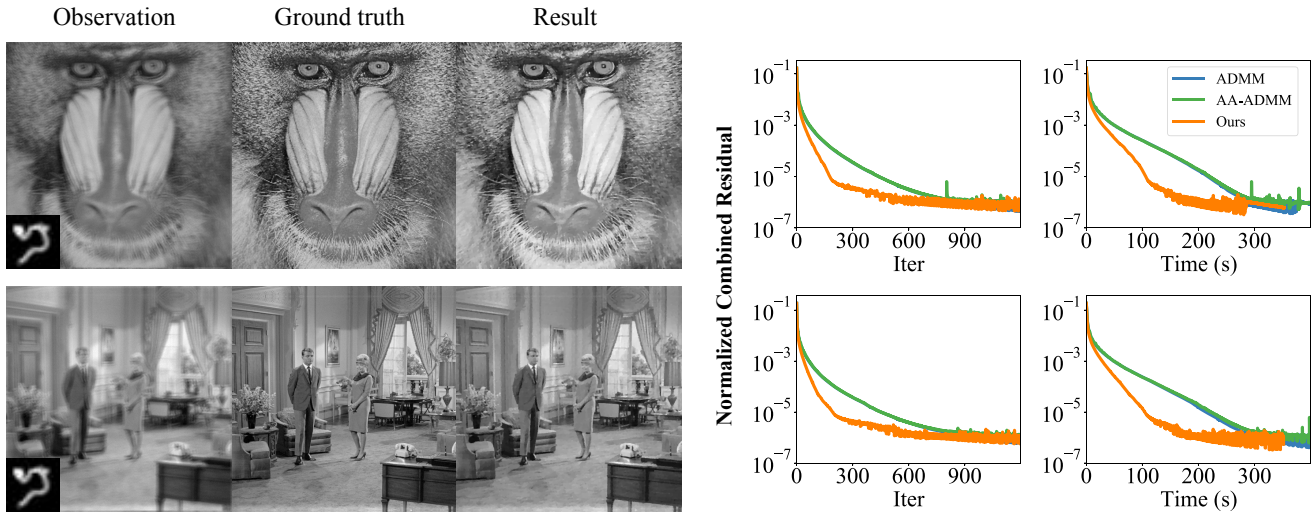

** https://github.com/wubaoyuan/Lpbox-ADMM

Observation   Ground truth   Result

Figure 7: *Comparison on the convex problem* (42) *for image deconvolution. We choose* $\lambda = 400$ *in problem* (42)*, and set the penalty parameter to* $\beta = 100$*. ADMM and AA-ADMM have fairly similar performance. Both are outperformed by our method.*

our method using the primal residual norm as the merit function. We use the source codes released by the authors[††][‡‡] to implement the ADMM solver and their accelerated versions. For both problems, our method accelerates the convergence of ADMM and outperforms AA-ADMM in the computational time.

Limitation. Similar to [ZPOD19], our method may not be effective for ADMM solvers with very low computational cost per iteration. Fig. 8 shows the performance of our method and AA-ADMM on the ADMM solver from [TZD*19] for recovering a geodesic distance function on a mesh surface from a unit tangent vector field. The two methods achieve almost the same effect in reducing the amount of iterations required for convergence. Our method requires a shorter computational time than AA-ADMM to achieve the same value of combined residual, because we can only apply the general approach of AA-ADMM to this problem and its overhead is higher than our method. On the other hand, both approaches take a longer time than the original ADMM solver to achieve convergence, because the very low computational cost per iteration of the original solver means high relative overhead for both acceleration techniques.

6. Concluding Remarks

In this paper, we propose an acceleration method for ADMM by applying Anderson Acceleration on its equivalent DR splitting formulation. Based on a fixed-point interpretation of DR splitting, we accelerate one of its variables that is not explicitly available in ADMM but can be derived from a linear transformation of the ADMM variables. Our strategy consistently outperforms the general Anderson acceleration approach in [ZPOD19] due to the lower dimensionality of the accelerated variable. Compared to the
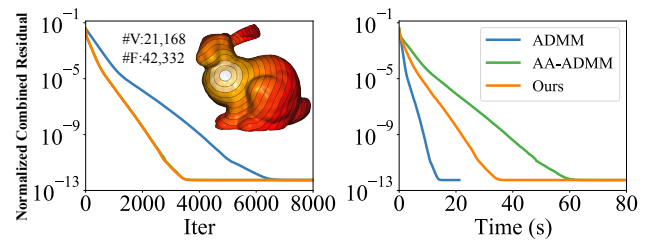

Figure 8: *Comparison on the ADMM solver in [TZD*19] for recovering geodesic distance on meshes. Both AA-ADMM and our method can reduce the number of iterations required for convergence, but their actual computational time is higher due to the very low computational cost per iteration for the ADMM solver. Our method takes a shorter time than AA-ADMM thanks to its lower overhead.*

reduced-variable approach in [ZPOD19], our method has the same dimensionality for the accelerated variable and achieves similar performance, but imposes no special requirements on the problem except for the separability of its target function. This makes our approach applicable to a much wider range of problems. In addition, we analyze the convergence of the proposed algorithm, and show that it converges to a stationary point of the ADMM problem under appropriate assumptions. Various ADMM solvers in computer graphics and other domains have been tested to verify the effectiveness and efficiency of our algorithm.

There are still some limitations for our approach. First, the equivalence between ADMM and DR splitting relies on a separable target function for the ADMM problem. As a result, our method is not applicable to problems where the target function is not separable. However, as far as we are aware of, the majority of ADMM problems in computer graphics, computer vision, and image processing have a separable target function. Moreover, as shown in the geometry

optimization example in Section 5, it is possible to reformulate the problem to make the target function separable. Therefore, this issue does not hinder the practical application of our method. Another limitation is that there is no theoretical guarantee that the method can always accelerate the convergence even locally. Recently, [EPRX20] provide theoretical results showing that Anderson Acceleration can improve the convergence rate, but their proofs require the original iteration to be contractive or converge q-linearly. For nonconvex DR splitting, to the best of our knowledge, local q-linear convergence can only be shown in very special cases that is too restrictive in practice. Further investigation of the theoretical property of Anderson Acceleration and nonconvex DR splitting is needed to provide a theoretical guarantee for acceleration.

References

[ABS13] ATTOUCH H., BOLTE J., SVAITER B. F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program. 137*, 1-2, Ser. A (2013), 91–129. 6, 16

[ABT14] ARTACHO F. J. A., BORWEIN J. M., TAM M. K.: Recent results on Douglas–Rachford methods for combinatorial optimization problems. *Journal of Optimization Theory and Applications 163*, 1 (2014), 1–30. 2

[AF13] ALMEIDA M. S. C., FIGUEIREDO M.: Deconvolving images with unknown boundaries using the alternating direction method of multipliers. *IEEE Transactions on Image Processing 22*, 8 (2013), 3074–3086. 2

[AJW17] AN H., JIA X., WALKER H. F.: Anderson acceleration and application to the three-temperature energy equations. *Journal of Computational Physics 347* (2017), 1–19. 2

[And65] ANDERSON D. G.: Iterative procedures for nonlinear integral equations. *J. ACM 12*, 4 (1965), 547–560. 2, 3

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum 31*, 5 (2012), 1657–1667. 1

[BK15] BAUSCHKE H. H., KOCH V. R.: Projection methods: Swiss army knives for solving feasibility and best approximation problems with halfspaces. *Contemp. Math 636* (2015), 1–40. 2

[BKW*19] BANSODE P., KOSARAJU K., WAGH S., PASUMARTHY R., SINGH N.: Accelerated distributed primal-dual dynamics using adaptive synchronization. *IEEE Access 7* (2019), 120424–120440. 2

[BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph. 33*, 4 (2014), 154:1–154:11. 1

[BPC*11] BOYD S., PARIKH N., CHU E., PELEATO B., ECKSTEIN J., ET AL.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning 3*, 1 (2011), 1–122. 1, 2, 3, 4

[BST14] BOLTE J., SABACH S., TEBOULLE M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program. 146*, 1-2, Ser. A (2014), 459–494. 16

[BTP13] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Sparse iterative closest point. *Computer Graphics Forum 32*, 5 (2013), 113–123. 2

[CP11] COMBETTES P. L., PESQUET J.-C.: Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Bauschke H. H., Burachik R. S., Combettes P. L., Elser V., Luke D. R., Wolkowicz H., (Eds.). 2011, pp. 185–212. 1

[DBD*15] DENG B., BOUAZIZ S., DEUSS M., KASPAR A., SCHWARTZBURG Y., PAULY M.: Interactive design exploration for constrained meshes. *Computer-Aided Design 61*, Supplement C (2015), 13–23. 9

[DR56] DOUGLAS J., RACHFORD H. H.: On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society 82*, 2 (1956), 421–439. 2

[EPRX20] EVANS C., POLLOCK S., REBHOLZ L. G., XIAO M.: A proof that anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically). *SIAM Journal on Numerical Analysis 58*, 1 (2020), 788–810. 2, 12

[Eye96] EYERT V.: A comparative study on methods for convergence acceleration of iterative vector sequences. *Journal of Computational Physics 124*, 2 (1996), 271–285. 2

[FB10] FIGUEIREDO M. A. T., BIOUCAS-DIAS J. M.: Restoration of poissonian images using alternating direction optimization. *IEEE Transactions on Image Processing 19*, 12 (2010), 3133–3145. 2

[FS09] FANG H.-R., SAAD Y.: Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications 16*, 3 (2009), 197–221. 2, 5

[FZB19] FU A., ZHANG J., BOYD S.: Anderson accelerated Douglas-Rachford splitting. *arXiv preprint arXiv:1908.11482* (2019). 2

[GITH14] GREGSON J., IHRKE I., THUEREY N., HEIDRICH W.: From capture to simulation: Connecting forward and inverse problems in fluids. *ACM Trans. Graph. 33*, 4 (2014), 139:1–139:11. 2

[Glo83] GLOWINSKI R.: *Augmented Lagrangian Methods: Applications to the numerical solution of boundary-value problems*. North-Holland, 1983. 2

[GOSB14] GOLDSTEIN T., O'DONOGHUE B., SETZER S., BARANIUK R.: Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences 7*, 3 (2014), 1588–1623. 2, 3

[GSD*14] GARG A., SAGEMAN-FURNAS A. O., DENG B., YUE Y., GRINSPUN E., PAULY M., WARDETZKY M.: Wire mesh design. *ACM Trans. Graph. 33*, 4 (2014), 66:1–66:12. 9, 10

[HDN*16] HEIDE F., DIAMOND S., NIESSNER M., RAGAN-KELLEY J., HEIDRICH W., WETZSTEIN G.: ProxImaL: Efficient image optimization using proximal algorithms. *ACM Trans. Graph. 35*, 4 (2016), 84:1–84:15. 2, 10

[HL13] HESSE R., LUKE D. R.: Nonconvex notions of regularity and convergence of fundamental algorithms for feasibility problems. *SIAM Journal on Optimization 23*, 4 (2013), 2397–2419. 2

[HLN14] HESSE R., LUKE D. R., NEUMANN P.: Alternating projections and Douglas-Rachford for sparse affine feasibility. *IEEE Transactions on Signal Processing 62*, 18 (2014), 4868–4881. 2

[HS16] HIGHAM N. J., STRABIĆ N.: Anderson acceleration of the alternating projections method for computing the nearest correlation matrix. *Numerical Algorithms 72*, 4 (2016), 1021–1042. 5

[KCSB15] KADKHODAIE M., CHRISTAKOPOULOU K., SANJABI M., BANERJEE A.: Accelerated alternating direction method of multipliers. KDD '15, pp. 497–506. 2

[LBK17] LIU T., BOUAZIZ S., KAVAN L.: Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans. Graph. 36*, 3 (2017), 23:1–23:16. 8

[LBOK13] LIU T., BARGTEIL A. W., O'BRIEN J. F., KAVAN L.: Fast simulation of mass-spring systems. *ACM Trans. Graph. 32*, 6 (2013), 214:1–214:7. 1

[LFYL18] LU C., FENG J., YAN S., LIN Z.: A unified alternating direction method of multipliers by majorization minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 40*, 3 (2018), 527–541. 2

[LP15] Li G., Pong T. K.: Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization 25*, 4 (2015), 2434–2460. 6

[LP16] Li G., Pong T. K.: Douglas–Rachford splitting for nonconvex optimization with application to nonconvex feasibility problems. *Mathematical programming 159*, 1-2 (2016), 371–401. 2, 3

[LSV13] Lipnikov K., Svyatskiy D., Vassilevski Y.: Anderson acceleration for nonlinear finite volume scheme for advection-diffusion problems. *SIAM Journal on Scientific Computing 35*, 2 (2013), A1120–A1136. 2

[LZX*08] Liu L., Zhang L., Xu Y., Gotsman C., Gortler S. J.: A local/global approach to mesh parameterization. *Computer Graphics Forum 27*, 5 (2008), 1495–1504. 1

[MST*18] Matveev S., Stadnichuk V., Tyrtyshnikov E., Smirnov A., Ampilogova N., Brilliantov N. V.: Anderson acceleration method of finding steady-state particle size distribution for a wide class of aggregation–fragmentation models. *Computer Physics Communications 224* (2018), 154–163. 2

[Nes18] Nesterov Y.: *Lectures on convex optimization*, vol. 137. Springer, 2018. 19

[NVT*14] Neumann T., Varanasi K., Theobalt C., Magnor M., Wacker M.: Compressed manifold modes for mesh processing. *Computer Graphics Forum 33*, 5 (2014), 35–44. 1, 2, 9

[NVW*13] Neumann T., Varanasi K., Wenger S., Wacker M., Magnor M., Theobalt C.: Sparse localized deformation components. *ACM Trans. Graph. 32*, 6 (2013), 179:1–179:10. 1, 2, 10

[NW06] Nocedal J., Wright S. J.: *Numerical Optimization*, 2nd ed. Springer-Verlag New York, 2006. 1

[OBLN17] Overby M., Brown G. E., Li J., Narain R.: ADMM ⊇ projective dynamics: Fast simulation of hyperelastic models with dynamic constraints. *IEEE Transactions on Visualization and Computer Graphics 23*, 10 (2017), 2222–2234. 1, 2, 8

[PDZ*18a] Peng Y., Deng B., Zhang J., Geng F., Qin W., Liu L.: Anderson acceleration for geometry optimization and physics simulation. *ACM Trans. Graph. 37*, 4 (2018), 42:1–42:14. 2, 6, 7

[PDZ*18b] Peng Y., Deng B., Zhang J., Geng F., Qin W., Liu L.: Anderson acceleration for geometry optimization and physics simulation. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 42. 4, 5

[PE13] Potra F. A., Engler H.: A characterization of the behavior of the anderson acceleration on linear problems. *Linear Algebra and its Applications 438*, 3 (2013), 1002–1011. 4

[Pha16] Phan H. M.: Linear convergence of the Douglas–Rachford method for two closed sets. *Optimization 65*, 2 (2016), 369–385. 2

[PJ16] Pejcic I., Jones C. N.: Accelerated ADMM based on accelerated Douglas-Rachford splitting. In *2016 European Control Conference (ECC)* (2016), Ieee, pp. 1952–1957. 2

[PL19] Poon C., Liang J.: Trajectory of alternating direction method of multipliers and adaptive acceleration. In *Advances in Neural Information Processing Systems* (2019), pp. 7355–7363. 2

[PM17] Pan Z., Manocha D.: Efficient solver for spacetime control of smoke. *ACM Trans. Graph. 36*, 5 (2017). 2

[POD*18] Pavlov A. L., Ovchinnikov G. V., Derbyshev D. Y., Tsetserukou D., Oseledets I. V.: AA-ICP: iterative closest point with anderson acceleration. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018* (2018), IEEE, pp. 1–6. 2

[PSP16] Pratapa P. P., Suryanarayana P., Pask J. E.: Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems. *Journal of Computational Physics 306* (2016), 43–54. 2

[RBSB18] Ranjan A., Bolkart T., Sanyal S., Black M. J.: Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)* (2018), Springer International Publishing, pp. 725–741. 10

[RS11] Rohwedder T., Schneider R.: An analysis for the DIIS acceleration method used in quantum chemistry calculations. *Journal of Mathematical Chemistry 49*, 9 (2011), 1889–1914. 2

[RW09] Rockafellar R. T., Wets R. J.-B.: *Variational analysis*, vol. 317. Springer Science & Business Media, 2009. 5, 15, 18

[SA07] Sorkine O., Alexa M.: As-rigid-as-possible surface modeling. SGP '07, pp. 109–116. 1

[SB12] Sifakis E., Barbič J.: FEM simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (2012), pp. 20:1–20:50. 8

[SPP19] Suryanarayana P., Pratapa P. P., Pask J. E.: Alternating anderson-richardson method: An efficient alternative to preconditioned krylov methods for large, sparse linear systems. *Computer Physics Communications 234* (2019), 278–285. 2

[Ste12] Sterck H. D.: A nonlinear GMRES optimization algorithm for canonical tensor decomposition. *SIAM Journal on Scientific Computing 34*, 3 (2012), A1351–A1379. 2

[TEE*17] Toth A., Ellis J. A., Evans T., Hamilton S., Kelley C. T., Pawlowski R., Slattery S.: Local improvement results for anderson acceleration with inaccurate function evaluations. *SIAM Journal on Scientific Computing 39*, 5 (2017), S47–S65. 2

[TK15] Toth A., Kelley C. T.: Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis 53*, 2 (2015), 805–819. 2

[TP20] Themelis A., Patrinos P.: Douglas-Rachford splitting and ADMM for nonconvex optimization: Tight convergence results. *SIAM Journal on Optimization 30*, 1 (2020), 149–181. 2, 3, 4, 6, 7, 15, 16, 17, 18, 19

[TZD*19] Tao J., Zhang J., Deng B., Fang Z., Peng Y., He Y.: Parallel and scalable heat methods for geodesic distance computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). 11

[WCX18] Wang F., Cao W., Xu Z.: Convergence of multi-block bregman admm for nonconvex composite problems. *Science China Information Sciences 61*, 12 (2018), 122101. 18

[WFDH18] Wang C., Fu Q., Dun X., Heidrich W.: Megapixel adaptive optics: Towards correcting large-scale distortions in computational cameras. *ACM Trans. Graph. 37*, 4 (2018), 115:1–115:12. 2

[WG19] Wu B., Ghanem B.: ℓ_p-box ADMM: A versatile framework for integer programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*, 7 (2019), 1695–1708. 2, 10

[WN11] Walker H. F., Ni P.: Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis 49*, 4 (2011), 1715–1735. 2, 3, 4

[WTK14] Willert J., Taitano W. T., Knoll D.: Leveraging Anderson acceleration for improved convergence of iterative solutions to transport systems. *Journal of Computational Physics 273* (2014), 278–286. 2

[WYZ19] Wang Y., Yin W., Zeng J.: Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing 78*, 1 (2019), 29–63. 1, 7

[XY13] Xu Y., Yin W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences 6*, 3 (2013), 1758–1789. 6, 18

[XZZ*14] Xiong S., Zhang J., Zheng J., Cai J., Liu L.: Robust surface reconstruction via dictionary learning. *ACM Trans. Graph. 33*, 6 (2014), 201:1–201:12. 2

[YY16] Yan M., Yin W.: Self equivalence of the alternating direction method of multipliers. In *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016, pp. 165–194. 2

[ZDL*14] Zhang J., Deng B., Liu Z., Patanè G., Bouaziz S., Hormann K., Liu L.: Local barycentric coordinates. *ACM Trans. Graph. 33*, 6 (2014), 188:1–188:12. 2

[ZPOD19] ZHANG J., PENG Y., OUYANG W., DENG B.: Accelerating ADMM for efficient simulation and optimization. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–21. 2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19

[ZUMJ19] ZHANG J., URIBE C. A., MOKHTARI A., JADBABAIE A.: Achieving acceleration in distributed optimization via direct discretization of the heavy-ball ODE. In *2019 American Control Conference (ACC)* (2019), IEEE, pp. 3408–3413. 2

[ZW18] ZHANG R. Y., WHITE J. K.: GMRES-accelerated ADMM for quadratic objectives. *SIAM Journal on Optimization 28*, 4 (2018), 3025–3056. 2

Appendix A: Derivation of ADMM for Problem (38)

In this section, we derive an ADMM solver for the geometry optimization problem (38) using the scheme (6)–(8). We first write the problem in matrix form as

$$\min_{\mathbf{x},\mathbf{z}} \ \frac{1}{2}\|\mathbf{L}(\mathbf{x}-\tilde{\mathbf{x}})\|^2 + \sum_{i\in\mathcal{S}}\frac{w_i}{2}\left(D_{C_i}(\mathbf{z}_i)\right)^2 + \sum_{j\in\mathcal{H}}\sigma_{C_j}(\mathbf{z}_j)$$

$$\text{s.t.} \ \ \mathbf{A}\mathbf{x}-\mathbf{z}=\mathbf{0},$$

where matrix $\mathbf{A}$ stacks all matrices $\{\mathbf{A}_i \mid i \in \mathcal{S}\}$ and $\{\mathbf{A}_j \mid j \in \mathcal{H}\}$. In the following, $\mathbf{y}$ denotes the dual variable that consists of $\{\mathbf{y}_i \mid i \in \mathcal{S}\}$ and $\{\mathbf{y}_j \mid j \in \mathcal{H}\}$ corresponding to the soft constraints $\mathcal{S}$ and hard constraints $\mathcal{H}$, respectively. We will use superscripts to indicate iteration counts, to avoid conflict with subscripts that indicate the constraints. Then the step (6) reduces to the problem

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{L}(\mathbf{x}-\tilde{\mathbf{x}})\|^2 + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}-\mathbf{z}^k+\mathbf{y}^k\|^2, \tag{43}$$

which can be solved via the linear system

$$(\mathbf{L}^T\mathbf{L}+\beta\mathbf{A}^T\mathbf{A})\mathbf{x}^{k+1} = \mathbf{L}^T\mathbf{L}\tilde{\mathbf{x}} + \beta\mathbf{A}^T(\mathbf{z}^k-\mathbf{y}^k). \tag{44}$$

The step (7) is simply written as

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{A}\mathbf{x}^{k+1} - \mathbf{z}^k. \tag{45}$$

The step (8) reduces to separable subproblems:

$$\min_{\mathbf{z}_i} \ \frac{w_i}{2}\left(D_{C_i}(\mathbf{z}_i)\right)^2 + \frac{\beta}{2}\|\mathbf{A}_i\mathbf{x}^{k+1}-\mathbf{z}_i+\mathbf{y}_i^{k+1}\|^2 \quad \text{for } i\in\mathcal{S}, \tag{46}$$

$$\min_{\mathbf{z}_j} \ \sigma_{C_j}(\mathbf{z}_j) + \frac{\beta}{2}\|\mathbf{A}_j\mathbf{x}^{k+1}-\mathbf{z}_j+\mathbf{y}_j^{k+1}\|^2 \quad \text{for } j\in\mathcal{H}. \tag{47}$$

The solution to (47) is

$$\mathbf{z}_j^{k+1} = P_{C_j}(\mathbf{A}_j\mathbf{x}^{k+1}+\mathbf{y}_j^{k+1}), \tag{48}$$

where $P_{C_j}(\cdot)$ is a projection operator onto the C_j. The solution to (46) is

$$\mathbf{z}_i^{k+1} = \frac{w_i \cdot P_{C_i}(\mathbf{A}_i\mathbf{x}^{k+1}+\mathbf{y}_i^{k+1}) + \beta\cdot(\mathbf{A}_i\mathbf{x}^{k+1}+\mathbf{y}_i^{k+1})}{w_i+\beta}. \tag{49}$$

Appendix B: Proof for Proposition 3.5

Proof. By Proposition 3.3 we have

$$\mathbf{u}(\mathbf{s}) = \mathbf{A}\bar{\mathbf{x}} = \text{prox}_{\gamma\varphi_1}(\mathbf{s}).$$

doing a simple change of variables, similar result for $\mathbf{v}(\mathbf{s})$ can be attained as

$$\mathbf{v}(\mathbf{s}) = \mathbf{B}\bar{\mathbf{z}} + \mathbf{c} = \text{prox}_{\gamma\varphi_2}(2\mathbf{u}(\mathbf{s})-\mathbf{s}),$$

For the expression of DR envelope, we utilize the optimality condition of $\text{prox}_{\gamma\varphi_1}$

$$\nabla\varphi_1(\mathbf{u}(\mathbf{s})) + \frac{1}{\gamma}(\mathbf{u}(\mathbf{s})-\mathbf{s})=0 \quad\Rightarrow\quad 2\mathbf{u}(\mathbf{s})-\mathbf{s}=\mathbf{u}(\mathbf{s})-\gamma\nabla\varphi_1(\mathbf{u}(\mathbf{s})). \tag{50}$$

We then rewrite ψ_{E} as

$$\psi_{\mathrm{E}}(\mathbf{s}) = \min_{\mathbf{w}}\left\{\varphi_1(\mathbf{u}(\mathbf{s})) + \varphi_2(\mathbf{w}) + \frac{1}{2\gamma}\|\mathbf{w}-(\mathbf{u}(\mathbf{s})-\gamma\nabla\varphi_1(\mathbf{u}(\mathbf{s})))\|^2 \right.$$
$$\left. -\frac{1}{2\gamma}\|\nabla\varphi_1(\mathbf{u}(\mathbf{s}))\|^2\right\}.$$

The definition of $\mathbf{v}(\mathbf{s})$ indicates that $\mathbf{v}(\mathbf{s})$ is the solution of minization problem in the definition of ψ_{E}, so

$$\psi_{\mathrm{E}}(\mathbf{s})$$
$$= \varphi_1(\mathbf{u}(\mathbf{s})) + \varphi_2(\mathbf{v}(\mathbf{s})) + \langle\nabla\varphi_1(\mathbf{u}(\mathbf{s})),\mathbf{v}(\mathbf{s})-\mathbf{u}(\mathbf{s})\rangle + \frac{1}{2\gamma}\|\mathbf{v}(\mathbf{s})-\mathbf{u}(\mathbf{s})\|^2$$
$$= \varphi_1(\mathbf{u}(\mathbf{s})) + \varphi_2(\mathbf{v}(\mathbf{s})) + \frac{1}{\gamma}\langle\mathbf{s}-\mathbf{u}(\mathbf{s}),\mathbf{v}(\mathbf{s})-\mathbf{u}(\mathbf{s})\rangle + \frac{1}{2\gamma}\|\mathbf{v}(\mathbf{s})-\mathbf{u}(\mathbf{s})\|^2.$$

By Proposition 3.3 we have

$$\varphi_1(\mathbf{u}(\mathbf{s})) = f(\bar{\mathbf{x}}), \quad \varphi_2(\mathbf{v}(\mathbf{s})) = g(\bar{\mathbf{z}}),$$

which completes the proof.   □

Appendix C: Proof for Proposition 3.7

Proof. By the definition of $\mathbf{s}^*$, we know that for $\mathbf{v}^* = \mathbf{B}\mathbf{z}^* + \mathbf{c}$. The definition of fixed-point indicates $\mathbf{v}^* = \mathbf{u}^*$. Hence $\mathbf{A}\mathbf{x}^* - \mathbf{B}\mathbf{z}^* - \mathbf{c} = 0$. We then utilize the definitions of $\mathbf{x}^*$ and $\mathbf{z}^*$ and the optimality conditions of the associated minimization problems:

$$-\frac{1}{\gamma}\mathbf{A}^T(\mathbf{A}\mathbf{x}^*-\mathbf{s}^*) \in \partial f(\mathbf{x}^*),$$
$$-\frac{1}{\gamma}\mathbf{B}^T(\mathbf{B}\mathbf{z}^*+\mathbf{c}-(2\mathbf{u}^*-\mathbf{s}^*)) \in \partial g(\mathbf{z}^*).$$

We note that $\frac{1}{\gamma}=\beta$, which means

$$-\beta\mathbf{A}^T\mathbf{y}^* = \frac{1}{\gamma}\mathbf{A}^T(\mathbf{u}^*-\mathbf{s}^*) \in \partial f(\mathbf{x}^*),$$
$$\beta\mathbf{B}^T\mathbf{y}^* = -\frac{1}{\gamma}\mathbf{B}^T(\mathbf{s}^*-\mathbf{u}^*) \in \partial g(\mathbf{z}^*).$$

This completes the proof.   □

Appendix D: Proof for Theorem 4.4

Let us first prove two lemmata:

Lemma D.1. *Assume that $\mathbf{s}^*$ is a fixed point of $\mathcal{G}$, φ_1 is differentiable, and $\text{prox}_{\gamma\varphi_1}$ is single-valued. Then $\mathbf{u}^* = \text{prox}_{\gamma\varphi_1}(\mathbf{s}^*)$ is a stationary point of* (11).

Proof. By the definition of $\mathcal{G}$

$$\mathbf{u}^* \in \text{prox}_{\gamma\varphi_2}(2\mathbf{u}^*-\mathbf{s}^*).$$

By the optimality condition of $\text{prox}_{\gamma\varphi_2}$

$$\frac{1}{\gamma}(\mathbf{u}^*-\mathbf{s}^*) \in \partial\varphi_2(\mathbf{u}^*).$$

By the definition of $\mathbf{u}^*$ and the optimality condition of $\text{prox}_{\gamma\varphi_1}$

$$0 = \nabla\varphi_1(\mathbf{u}^*) + \frac{1}{\gamma}(\mathbf{u}^* - \mathbf{s}^*),$$

which means $0 \in \nabla\varphi_1(\mathbf{u}^*) + \partial\varphi_2(\mathbf{u}^*)$. $\qquad\square$

Lemma D.2. *Let $\mathbf{s}^*$ and $\mathbf{u}^*$ be defined in Lemma D.1 and assume the conditions in Lemma D.1 hold. Moreover, define*

$$\mathbf{Z}_\beta(\mathbf{s}) = \underset{\mathbf{z}\in\mathbb{R}^n}{\operatorname{argmin}}\left(g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{Bz} + \mathbf{c} - \mathbf{s}\|^2\right).$$

If $\mathbf{u}^* \in \mathbf{BZ}_\beta(2\mathbf{u}^* - \mathbf{s}^*) + \mathbf{c}$, and $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ satisfies

$$\mathbf{x}^* \in \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{Ax} - \mathbf{s}^*\|^2,$$

$$\mathbf{y}^* = \mathbf{u}^* - \mathbf{s}^*,$$

$$\mathbf{u}^* = \mathbf{Bz}^* + \mathbf{c}, \quad \mathbf{z}^* \in \mathbf{Z}_\beta(2\mathbf{u}^* - \mathbf{s}^*),$$

then $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ is a stationary point of (5).

Proof. By Proposition 3.3 we have $\mathbf{Ax}^* \in \text{prox}_{\gamma\varphi_1}(\mathbf{s}^*)$. Since $\text{prox}_{\gamma\varphi_1}$ is single-valued we have $\mathbf{u}^* = \mathbf{Ax}^*$. By (50)

$$\frac{1}{\gamma}(\mathbf{s}^* - \mathbf{u}^*) = \nabla\varphi_1(\mathbf{u}^*).$$

By [TP20, Proposition 5.3] we have

$$\mathbf{A}^T\nabla\varphi_1(\mathbf{u}^*) = \mathbf{A}^T\hat\partial\varphi_1(\mathbf{u}^*) \subset \hat\partial f(\mathbf{x}^*),$$

where we have $\hat\partial\varphi_1(\mathbf{u}^*) = \{\nabla\varphi_1(\mathbf{u}^*)\}$ by [RW09, Exercis 8.8]. Notice that $\beta = \frac{1}{\gamma}$ we then have

$$-\beta\mathbf{A}^T\mathbf{y}^* \in \hat\partial f(\mathbf{x}^*) \subset \partial f(\mathbf{x}^*).$$

Similarly we have

$$\beta\mathbf{y}^* = \frac{1}{\gamma}(\mathbf{u}^* - \mathbf{s}^*) \in \hat\partial\varphi_2(\mathbf{u}^*).$$

By Proposition 3.3 and [RW09, Exercise 8.8]

$$\beta\mathbf{B}^T\mathbf{y}^* \in \mathbf{B}^T\hat\partial\varphi_2(\mathbf{u}^*) \subset \hat\partial g(\mathbf{z}^*) \subset \partial g(\mathbf{z}^*).$$

Finally, we have

$$\mathbf{Ax}^* = \mathbf{u}^* = \mathbf{Bz}^* + \mathbf{c} \quad \Rightarrow \quad \mathbf{Ax}^* - \mathbf{Bz}^* - \mathbf{c} = 0. \qquad\square$$

Finally, we give the main proof for Theorem 4.4.

Proof. The proof here is similar to the proof for [TP20, Theorem 4.1]. Let $\eta = \min\{\nu_1, \frac{c}{(1+\gamma L)^2}\}$, where c is the constant defined in [TP20, Theorem 4.1], then by algorithmic construction we have

$$\psi_E(\mathbf{s}_k) - \psi_E(\mathbf{s}_{k+1}) \geq \eta\|\mathbf{s}_k - \mathbf{G}(\mathbf{s}_k)\|^2 = \eta\|\mathbf{v}_k - \mathbf{u}_k\|^2.$$

Due to the definition of ψ_E and the fact that φ_1, φ_2 are both proper, we have $\psi_E(\mathbf{s}_0) < \infty$. By Assumption (A.3) we know $\varphi = \varphi_1 + \varphi_2$ is bounded from below and then by [TP20, Proposition 3.4] ψ_E is also bounded from below. Hence

$$\eta\sum_{k=0}^{\infty}\|\mathbf{u}_k - \mathbf{v}_k\|^2 < \infty \quad \Rightarrow \quad \|\mathbf{u}_k - \mathbf{v}_k\| \to 0,$$

which proves (a). For (b) we first note that since $\gamma < \frac{1}{L}$, by [TP20, Theorem 3.1] ψ_E is level-bounded provided Assumption (B.2) holds.

So by (a) we know $\{\mathbf{s}_k\}$ is bounded. Then by [TP20, Proposition 2.3] we know $\text{prox}_{\gamma\varphi_1}$ is Lipschitz continuous. Therefore $\{\mathbf{u}_k\}$ is also bounded. The boundedness of $\{\mathbf{v}_k\}$ follows from $\|\mathbf{v}_k - \mathbf{u}_k\| \to 0$. Therefore $\mathbf{u}_{k_i} \to \mathbf{u}^*$. Next, to prove that $\mathbf{s}^*$ is a fixed-point of $\mathcal{G}$, it suffices to show $\mathbf{u}^* \in \text{prox}_{\gamma\varphi_2}(2\mathbf{u}^* - \mathbf{s}^*)$. By the continuity of $\text{prox}_{\gamma\varphi_1}$ we know $\mathbf{u}_{k_i} \to \mathbf{u}^*$. Then since $\|\mathbf{u}_k - \mathbf{v}_k\| \to 0$, we also have $\mathbf{v}_{k_i} \to \mathbf{u}^*$. Notice that $\mathbf{v}_{k_i} \in \text{prox}_{\gamma\varphi_2}(2\mathbf{u}_{k_i} - \mathbf{s}_{k_i})$, by Assumption (B.2) and [RW09, Theorem 1.25] we know that $\text{prox}_{\gamma\varphi_2}$ is outer semicontinuous(osc), then by [RW09, Exercise 5.30]

$$\mathbf{u}^* = \lim_{i\to\infty}\mathbf{v}_{k_i} \subset \limsup_{i\to\infty}\text{prox}_{\gamma\varphi_2}(2\mathbf{u}_{k_i} - \mathbf{s}_{k_i}) \subset \text{prox}_{\gamma\varphi_2}(2\mathbf{u}^* - \mathbf{s}^*),$$

which proves that $\mathbf{s}^*$ is fixed-point of $\mathcal{G}$. The stationarity of $\mathbf{u}^*$ follows from Lemma D.1.

For (c), notice that if Assumption (B.3) holds, then $\mathbf{Z}_\beta$ is locally bounded and osc by [RW09, Theorem 1.17]. Since $\mathbf{z}_{k_i} \in \mathbf{Z}_\gamma(2\mathbf{u}_{k_i} - \mathbf{s}_{k_i})$ and $2\mathbf{u}_{k_i} - \mathbf{s}_{k_i} \to 2\mathbf{u}^* - \mathbf{s}^*$, $\{\mathbf{z}_{k_i}\}$ is bounded. So the cluster point of $\{\mathbf{z}_{k_i}\}$, $\mathbf{z}^*$ must exists. Without loss of generality, we can assume $\mathbf{z}_{k_i} \to \mathbf{z}^*$. By [RW09, Exercise 5.30]

$$\mathbf{z}^* = \lim_{i\to\infty}\mathbf{z}_{k_i} \subset \limsup_{i\to\infty}\mathbf{Z}_\beta(2\mathbf{u}_{k_i} - \mathbf{s}_{k_i}) \subset \mathbf{Z}_\beta(2\mathbf{u}^* - \mathbf{s}^*) + \mathbf{c}.$$

Next, push to the limit on both sides of $\mathbf{v}_{k_i} = \mathbf{Bz}_{k_i} + \mathbf{c}$

$$\mathbf{v}^* = \lim_{i\to\infty}\mathbf{v}_{k_i} = \lim_{i\to\infty}\mathbf{Bz}_{k_i} + \mathbf{c} = \mathbf{Bz}^* + \mathbf{c}.$$

The stationarity of $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ follows from Lemma D.2. $\qquad\square$

Appendix E: Further Discussion for Generating the Stationary Point of ADMM

For the most general case where $\text{prox}_{\gamma\varphi_1}$ and $\text{prox}_{\gamma\varphi_2}$ are both set-valued, we need much more sophisticated techniques to generate the stationary point of ADMM.

We note the definition of $\mathbf{s}^* \in \mathcal{G}(\mathbf{s}^*)$ means there exist $\mathbf{u}^*$ such that $\mathbf{u}^* \in \text{prox}_{\gamma\varphi_1}(\mathbf{s}^*)$ and $\mathbf{u}^* \in \text{prox}_{\gamma\varphi_2}(2\mathbf{u}^* - \mathbf{s}^*)$. We assume $\mathbf{u}^*$ is known since $\mathbf{u}^*$ is explicitly available from Algorithm 1. This is because proximal mapping is outer semi-continuous, which means if a subsequence $\mathbf{s}_{k_i} \to \mathbf{s}^*$, then it suffice to choose a cluster point of $\{\mathbf{u}_{k_i}\}$ to generate such a $\mathbf{u}^*$. Our goal is to generate the stationary point of ADMM from $(\mathbf{s}^*, \mathbf{u}^*)$.

We first need a technical lemma:

Lemma E.1. *Let $h: \mathbb{R}^n \to \bar{\mathbb{R}}$ and $\mathbf{C} \in \mathbb{R}^{p\times n}$. Suppose for some β the set-valued mapping $\mathbf{X}_\beta(\mathbf{s}) := \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}}\{h(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{Cx} - \mathbf{s}\|^2\}$ is nonempty for any $\mathbf{s} \in \mathbb{R}^p$. Let $\gamma = 1/\beta$ and $\varphi = h_\mathbf{C}$. If $\mathbf{u}^* \in \text{prox}_{\gamma\varphi}(\mathbf{s}^*)$ and*

$$\mathbf{x}^* \in \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}}\{h(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{Cx} - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{Cx} - \mathbf{u}^*\|^2\},$$

where $\alpha > 0$. Then $\mathbf{x}^* \in \mathbf{X}_\beta(\mathbf{s}^*)$ and $\mathbf{u}^* = \mathbf{Cx}^*$.

Proof. We first prove $\mathbf{Cx}^* = \mathbf{u}^*$. Let $\mathbf{u}^+ = \mathbf{Cx}^*$. By the definition of $\mathbf{x}^*$ we know $\forall\mathbf{e} \in \ker(\mathbf{C})$, we have $h(\mathbf{x}^*) \leq h(\mathbf{x}^* + \mathbf{e})$, which means

$\varphi(\mathbf{u}^+) = h(\mathbf{x}^*)$. Then we have:

$$\varphi(\mathbf{u}^+) + \frac{\beta}{2}\|\mathbf{u}^+ - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{u}^+ - \mathbf{u}^*\|^2$$

$$= h(\mathbf{x}^*) + \frac{\beta}{2}\|\mathbf{C}\mathbf{x}^* - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{C}\mathbf{x}^* - \mathbf{u}^*\|^2$$

$$= \inf_{\mathbf{u}}\{ \inf_{\mathbf{x}:\mathbf{C}\mathbf{x}=\mathbf{u}} h(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{u} - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{u} - \mathbf{u}^*\|^2 \}$$

$$= \inf_{\mathbf{u}}\{\varphi(\mathbf{u}) + \frac{\beta}{2}\|\mathbf{u} - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{u} - \mathbf{u}^*\|^2 \}.$$

But by the definition of $\mathbf{u}^*$, we know

$$\{\mathbf{u}^*\} = \underset{\mathbf{u}}{\operatorname{argmin}}\{\varphi(\mathbf{u}) + \frac{\beta}{2}\|\mathbf{u} - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{u} - \mathbf{u}^*\|^2\},$$

which means $\mathbf{u}^+ = \mathbf{u}^*$ and hence demonstrates that $\mathbf{u}^* = \mathbf{C}\mathbf{x}^*$. Now assume that $\mathbf{x}^* \notin \mathbf{X}_\beta(\mathbf{s}^*)$, which means there exists another $\mathbf{x}^+$ such that

$$h(\mathbf{x}^+) + \frac{\beta}{2}\|\mathbf{C}\mathbf{x}^+ - \mathbf{s}^*\|^2 < h(\mathbf{x}^*) + \frac{\beta}{2}\|\mathbf{C}\mathbf{x}^* - \mathbf{s}^*\|^2.$$

However, we have

$$h(\mathbf{x}^*) + \frac{\beta}{2}\|\mathbf{C}\mathbf{x}^* - \mathbf{s}^*\|^2 = \varphi(\mathbf{u}^*) + \frac{\beta}{2}\|\mathbf{u}^* - \mathbf{s}^*\|^2,$$

$$\leq \varphi(\mathbf{C}\mathbf{x}^+) + \frac{\beta}{2}\|\mathbf{C}\mathbf{x}^+ - \mathbf{s}^*\|^2,$$

$$\leq h(\mathbf{x}^+) + \frac{\beta}{2}\|\mathbf{C}\mathbf{x}^+ - \mathbf{s}^*\|^2,$$

which yields contradiction. Hence $\mathbf{x}^* \in \mathbf{X}_\beta(\mathbf{s}^*)$. $\qquad\square$

Then we are able to prove the general transition theorem:

Theorem E.1. *Suppose $\mathbf{s}^*$ is the fixed-point of $\mathcal{G}$, and $\mathbf{u}^* \in$ prox$_{\gamma\varphi_1}(\mathbf{s}^*) \cap$ prox$_{\gamma\varphi_2}(2\mathbf{u}^* - \mathbf{s}^*)$. Define:*

$$\mathbf{x}^* \in \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{A}\mathbf{x} - \mathbf{s}^*\|^2 + \frac{\alpha}{2}\|\mathbf{A}\mathbf{x} - \mathbf{u}^*\|^2,$$

$$\mathbf{y}^* = \mathbf{s}^* - \mathbf{u}^*$$

$$\mathbf{z}^* \in \underset{\mathbf{z}}{\operatorname{argmin}} g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{B}\mathbf{z} + \mathbf{c} - (2\mathbf{u}^* - \mathbf{s}^*)\|^2 + \frac{\alpha}{2}\|\mathbf{B}\mathbf{z} + \mathbf{c} - \mathbf{u}^*\|^2,$$

where $\alpha > 0$, then $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ is a stationary point of (5).

Proof. Lemma E.1 means that

$$\mathbf{u}^* = \mathbf{A}\mathbf{x}^*, \mathbf{u}^* = \mathbf{B}\mathbf{z}^* + \mathbf{c},$$

$$\mathbf{x}^* \in \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{A}\mathbf{x} - \mathbf{s}^*\|^2,$$

$$\mathbf{z}^* \in \underset{\mathbf{z}}{\operatorname{argmin}} g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{B}\mathbf{z} + \mathbf{c} - (2\mathbf{u}^* - \mathbf{s}^*)\|^2.$$

Next, we utilize the optimality conditions of the proximity operator prox$_{\gamma\varphi_1}(\mathbf{s}^*)$

$$\frac{1}{\gamma}(\mathbf{s}^* - \mathbf{u}^*) \in \hat{\partial}\varphi_1(\mathbf{u}^*).$$

Then by [TP20, Proposition 5.3]

$$-\beta\mathbf{A}^T\mathbf{y}^* = \frac{1}{\gamma}\mathbf{A}^T(\mathbf{s}^* - \mathbf{u}^*) \in \hat{\partial}f(\mathbf{x}^*) \in \partial f(\mathbf{x}^*).$$

Similarly we have

$$\beta\mathbf{B}^T\mathbf{y}^* \in \partial g(\mathbf{z}^*).$$

The last condition follows from $\mathbf{u}^* = \mathbf{A}\mathbf{x}^* = \mathbf{B}\mathbf{z}^* + \mathbf{c}$. $\qquad\square$

Appendix F: Proof for Theorem 4.6 and Remark 4.8

Our proof will utilize the Kurdyka-Łojasiewicz (KL) inequality [ABS13]. We will introduce several notations for the definition of KL property. Let C_η be the set consisting of all the concave and continuous function $\rho : [0, \eta) \to \mathbb{R}_+$ satisfying that

$$\rho \in C^1((0, \eta)), \quad \rho(0) = 0, \quad \rho'(x) > 0, \forall x \in (0, \eta).$$

We also consider a subclass of C_η, called Łojasiewicz functions

$$\mathcal{L} := \{\rho : \mathbb{R}_+ \to \mathbb{R}_+, \exists\, m > 0, \theta \in [0, 1) : \rho(x) = qx^{1-\theta}\}.$$

Next we give the definition of the KL property:

Definition F.1. Let ψ be a proper, lower semicontinuous function. We say that ψ has the KL property at $\bar{\mathbf{x}} \in \operatorname{dom}\partial\psi$ if there exists $\eta \in (0, \infty]$, a neighborhood U of $\bar{\mathbf{x}}$, and a function $\rho \in C_\eta$ such that for all $\mathbf{x} \in U \cap \{\mathbf{x} \in \mathbb{R}^n : 0 < \psi(\mathbf{x}) - \psi(\bar{\mathbf{x}}) < \eta\}$ the KL-inequality holds, i.e.,

$$\rho'(\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})) \cdot \operatorname{dist}(0, \partial\psi(\mathbf{x})) \geq 1. \qquad (51)$$

If the mapping ρ can be chosen from $\mathcal{L}$ and satisfies $\rho(x) = qx^{1-\theta}$ for some $q > 0$ and $\theta \in [0, 1)$, then we say that ψ has the KL-property at $\bar{x}$ with exponent θ.

It is known that a variety of functions, which contains the subanalytic function [ABS13], have the KL property. So we will directly work with the KL property.

We note that by the definition of $\mathcal{D}_\gamma$ it is clear that $\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) = \psi_E(\mathbf{s}_k)$.

Assume $\{\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k\}$ is generated by Algorithm 1, then we define $\mathcal{U}$ to be the set consisting of all the cluster points of $\{\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k\}$. Several structural properties of U are listed in the next proposition.

Proposition F.2. *Assume $\{\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k\}$ is bounded. Then*

(a) $\mathcal{U}$ is nonempty and compact.
(b) $\operatorname{dist}((\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k), \mathcal{U}) \to 0$.
(c) If the assumptions in Theorem 4.4 hold, then $\mathcal{D}_\gamma$ is constant and finite on $\mathcal{U}$.

Proof. For statement (a) and (b), see [BST14, Lemma 5(iii)]. For (c), by Theorem 4.4 we can assume $\psi_E(\mathbf{s}_k) \to l^*$ where l^* is finite. Now assume $(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*) \in \mathcal{U}$, then the proof in Theorem 4.4 has already shown that $\mathbf{v}^* = \mathbf{u}^* \in$ prox$_{\gamma\varphi_2}(2\mathbf{u}^* - \mathbf{s}^*)$. And we clearly have $\mathbf{u}^* =$ prox$_{\gamma\varphi_1}(\mathbf{s}^*)$ by the continuity of prox$_{\gamma\varphi_1}$. Hence $\mathcal{D}_\gamma(\mathbf{s}^*, \mathbf{u}^*, \mathbf{v}^*) = \psi_E(\mathbf{s}^*)$. Notice that ψ_E is strictly continuous [TP20, Proposition 3.2], so we have $\psi_E(\mathbf{s}^*) = l^*$, which completes the proof. $\qquad\square$

In the following we provide the main proof of global and r-linear convergence stated in Theorem 4.6 and Remark 4.8.

Proof. These two conclusions trivially hold if Algorithm 1 terminates after finite steps, so in the rest of the proof we assume Algorithm 1 generates infintely many steps. Let δ, η be the constants

appearing in the definition of KL property. Choose k' sufficiently large such that for any $k \geq k'$ we have

$$\text{dist}((\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k), \mathcal{U}) < \delta, \quad 0 < \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\bar{\mathbf{s}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}) < \eta$$

where $(\bar{\mathbf{s}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}) \in \mathcal{U}$. Such a k' exists due to Proposition F.2. Define $\delta_k = \rho(\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\bar{\mathbf{s}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}))$. For $k \geq k'$ we utilize the concavity of ρ

$$\delta_k - \delta_{k+1} \geq \rho'(\delta_k)(\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\mathbf{s}_{k+1}, \mathbf{u}_{k+1}, \mathbf{v}_{k+1}))$$
$$\geq \frac{\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\mathbf{s}_{k+1}, \mathbf{u}_{k+1}, \mathbf{v}_{k+1})}{\text{dist}(0, \partial \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k))}. \tag{52}$$

Next, we estimate $\text{dist}(0, \partial \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k))$:

$$\nabla_s \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) = \frac{1}{\gamma}(\mathbf{v}_k - \mathbf{u}_k),$$

$$\nabla_u \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) = \nabla \varphi_1(\mathbf{u}_k) - \frac{1}{\gamma}(\mathbf{s}_k - \mathbf{u}_k) - \frac{1}{\gamma}(\mathbf{v}_k - \mathbf{u}_k) + \frac{1}{\gamma}(\mathbf{u}_k - \mathbf{v}_k),$$

$$= \frac{2}{\gamma}(\mathbf{u}_k - \mathbf{v}_k)$$

$$\partial_v \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) = \partial \varphi_2(\mathbf{v}_k) + \frac{1}{\gamma}(\mathbf{s}_k - \mathbf{u}_k) + \frac{1}{\gamma}(\mathbf{v}_k - \mathbf{u}_k) \ni 0,$$

where we have used (50) for the second equality and the optimality condition of $\text{prox}_{\gamma \varphi_2}$ for the third equality. These means

$$\text{dist}(0, \partial \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k)) \leq \frac{\sqrt{5}}{\gamma}\|\mathbf{v}_k - \mathbf{u}_k\|. \tag{53}$$

We now consider the next two cases

Case 1: $\mathbf{s}_{k+1} = \mathbf{s}_k^{AA}$ then

$$\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\mathbf{s}_{k+1}, \mathbf{u}_{k+1}, \mathbf{v}_{k+1}) = \psi_E(\mathbf{s}_k) - \psi_E(\mathbf{s}_{k+1})$$
$$\geq \nu_1 \|\mathbf{v}_k - \mathbf{u}_k\|^2 + \nu_2 \|\mathbf{s}_{k+1} - \mathbf{s}_k\|^2.$$

Moreover, by Young's inequality

$$2\sqrt{\nu_1 \nu_2}\|\mathbf{s}_{k+1} - \mathbf{s}_k\| = 2\frac{\sqrt{\nu_2}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|}{\sqrt{\|\mathbf{v}_k - \mathbf{u}_k\|}}\sqrt{\nu_1}\sqrt{\|\mathbf{v}_k - \mathbf{u}_k\|}$$
$$\leq \frac{\nu_2 \|\mathbf{s}_{k+1} - \mathbf{s}_k\|^2}{\|\mathbf{v}_k - \mathbf{u}_k\|} + \nu_1 \|\mathbf{v}_k - \mathbf{u}_k\|.$$

Then by (52) and (53)

$$\delta_k - \delta_{k+1} \geq \frac{\gamma}{\sqrt{5}}\left(\frac{\nu_2 \|\mathbf{s}_{k+1} - \mathbf{s}_k\|^2}{\|\mathbf{v}_k - \mathbf{u}_k\|} + \nu_1 \|\mathbf{v}_k - \mathbf{u}_k\|\right)$$
$$\geq \frac{2\gamma\sqrt{\nu_1 \nu_2}}{\sqrt{5}}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|.$$

Case 2: $\mathbf{s}_{k+1} = \mathbf{G}(\mathbf{s}_k)$. Then $\mathbf{s}_{k+1} - \mathbf{s}_k = \mathbf{v}_k - \mathbf{u}_k$ and by [TP20, Theorem 4.1]

$$\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\mathbf{s}_{k+1}, \mathbf{u}_{k+1}, \mathbf{v}_{k+1}) = \psi_E(\mathbf{s}_k) - \psi_E(\mathbf{s}_{k+1})$$
$$\geq \frac{c}{(1 + \gamma L)^2}\|\mathbf{v}_k - \mathbf{u}_k\|^2.$$

where c is the constant defined in [TP20, Theorem 4.1]. By (52) and (53)

$$\delta_k - \delta_{k+1} \geq \frac{c\gamma}{\sqrt{5}(1 + \gamma L)^2}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|.$$

Let $\bar{a} = \min\{\frac{2\gamma\sqrt{\nu_1 \nu_2}}{\sqrt{5}}, \frac{c\gamma}{\sqrt{5}(1 + \gamma L)^2}\}$, then we have

$$\delta_k - \delta_{k+1} \geq \bar{a}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|. \tag{54}$$

Notice that δ_k is positive and monotone decreasing, summing (54) from k' to ∞

$$\delta_{k'} \geq \bar{a}\sum_{k=k'}^{\infty}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|. \tag{55}$$

which means that $\{\mathbf{s}_k\}$ is a Cauchy sequence and hence converges to some point $\mathbf{s}^*$. By the continuity of $\text{prox}_{\gamma \varphi_1}$ we know $\mathbf{u}_k \to \mathbf{u}^* = \text{prox}_{\gamma \varphi_1}(\mathbf{s}^*)$. $\mathbf{v}^* = \mathbf{u}^*$ follows from $\|\mathbf{v}_k - \mathbf{u}_k\| \to 0$. $\mathbf{u}^*$ is fixed-point of $\mathcal{G}$ follows from Theorem 4.4. For Remark 4.8, we need the condition that $\mathcal{D}_\gamma$ has the KL property at $\mathcal{U}$ with exponent $\theta \in (0, \frac{1}{2}]$. Now assume $\rho(x) = qx^{1-\theta}$. By the definition of KL property (51) and (53)

$$\frac{\sqrt{5}q(1-\theta)}{\gamma}\|\mathbf{v}_k - \mathbf{u}_k\| \geq q(1-\theta)\text{dist}(0, \partial \mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k))$$
$$\geq (\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\bar{\mathbf{s}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}))^\theta.$$

Hence we have

$$\delta_k = q(\mathcal{D}_\gamma(\mathbf{s}_k, \mathbf{u}_k, \mathbf{v}_k) - \mathcal{D}_\gamma(\bar{\mathbf{s}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}))^{1-\theta} \leq q\left(\frac{\sqrt{5}q(1-\theta)}{\gamma}\|\mathbf{v}_k - \mathbf{u}_k\|\right)^{\frac{1-\theta}{\theta}}.$$

By elementary calculus, one can show that $(\frac{\sqrt{5}q(1-\theta)}{\gamma}\|\mathbf{v}_k - \mathbf{u}_k\|)^{\frac{1-\theta}{\theta}}$ is monotone increasing on $\theta \in (0, \frac{1}{2}]$ provided that $\frac{\sqrt{5}q\|\mathbf{v}_k - \mathbf{u}_k\|}{\gamma} < 1$. Since $\|\mathbf{v}_k - \mathbf{u}_k\| \to 0$, we can assume that k' is sufficiently large such that for any $k \geq k'$ we have $\frac{\sqrt{5}q\|\mathbf{v}_k - \mathbf{u}_k\|}{\gamma} < 1$. Then

$$\delta_k \leq a_1\|\mathbf{v}_k - \mathbf{u}_k\|.$$

where a_1 is some constant. Similar to the previous proof, we can show

$$\delta_k - \delta_{k+1} \geq a_2\|\mathbf{v}_k - \mathbf{u}_k\|. \tag{56}$$

where $a_2 = \frac{\sqrt{5}}{\gamma}\min\{\nu_1, \frac{c}{(1+\gamma L)^2}\}$. Summing (56) from k' to ∞

$$\delta_{k'} \geq a_2\sum_{k=k'}^{\infty}\|\mathbf{v}_k - \mathbf{u}_k\|.$$

Therefore

$$a_1\|\mathbf{v}_{k'} - \mathbf{u}_{k'}\| \geq a_2\sum_{k=k'}^{\infty}\|\mathbf{v}_k - \mathbf{u}_k\|.$$

Define $H_k = \sum_{i=k}^{\infty}\|\mathbf{v}_i - \mathbf{u}_i\|$ we have

$$a_1(H_{k'} - H_{k'+1}) \geq a_2 H_{k'} \quad \Rightarrow \quad H_{k'+1} \leq \frac{a_1 - a_2}{a_1}H_{k'}.$$

Similarly, we can show for any $l \geq k'$ we have

$$H_{l+1} \leq \frac{a_1 - a_2}{a_1}H_l,$$

which means that $\{H_k\}$ converges q-linearly and since $H_k \geq \|\mathbf{v}_k - \mathbf{u}_k\|$ we get the r-linear convergence of $\|\mathbf{v}_k - \mathbf{u}_k\|$. Then

$$a_1\|\mathbf{v}_k - \mathbf{u}_k\| \geq \delta_k \geq \bar{a}\sum_{i=k}^{\infty}\|\mathbf{s}_{i+1} - \mathbf{s}_i\| \geq \|\mathbf{s}_{k+1} - \mathbf{s}_k\|,$$

which proves the r-linear convergence of $\{\|\mathbf{s}_{k+1} - \mathbf{s}_k\|\}$ and further implies the r-linear convergence of $\{\mathbf{s}_k\}$. □

Appendix G: Proof for Proposition 4.10

Proof. The properness of φ_2 are given in Proposition 3.3(i) since we assume all the ADMM subproblems has solution. The lower semicontinuity of φ_2 are given by [TP20, Proposition 5.10] by assuming (D.4). Hence assumption (A.2) is satisfied. Assumption (A.1) comes from [TP20, Theorem 5.13] by assuming (D.2) and (D.3). For Assumption (B.2), without loss of generality, we can assume f is bounded from below and g is level-bounded. Then it is clear that φ_1 is bounded from below and φ_2 is level-bounded, and hence $\varphi_1 + \varphi_2$ is level-bounded. □

Appendix H: Verification of Assumptions for ℓ_q Regularized Logistic Regression Problem

In this section we will verify Assumption (A.1) to (A.3), (B.2) and (B.4), (C.2) for the ℓ_q regularized logistic regression problem. In this problem, we have

$$f(\mathbf{x}) = \sum_{i=1}^{p} \log(1 + \exp(-b_i(\mathbf{a}_i^T \mathbf{w} + v))), g(\mathbf{z}) = p \cdot \lambda \cdot \sum_{i=1}^{n} |z_i|^q$$

where $\mathbf{x} = (\mathbf{w}, v) \in \mathbb{R}^{n+1}$. For this problem matrices $\mathbf{A}$ and $\mathbf{B}$ are all identity, so image functions have rather simple form, i.e., $\varphi_1 = f$ and $\varphi_2 = g$. It is well known that f Lipschitz differentiable, so (A.1) is satisfied. Moreover, g is continuous and hence lower semicontinuous, so (A.2) is satisfied. To prove (A.3), since $f + g$ is continuous, it suffices to show (B.2) hold, because (A.1) then follows by [RW09, Theorem 1.9]. For the level-boundedness of $f + g$, we have:

Proposition H.1. *Assume b_i are not all 1 or -1, then $f + g$ is level-bounded.*

Proof. Without loss of generality, we can assume $b_1 = 1$ and $b_2 = -1$. Let $\alpha \in \mathbb{R}$, and $S = \{\mathbf{x} \in \mathbb{R}^{n+1} : f(\mathbf{x}) + g(\mathbf{x}) \leq \alpha\}$. Since if $\alpha \leq 0$, then it is easy to show S is bounded, we assume $\alpha > 0$ in the following. We need to prove that S is bounded. Now suppose $\mathbf{z} = (z_1, ..., z_{n+1}) \in S$, since $f(\mathbf{z}) \geq 0$, we have $\mathbf{g}(\mathbf{z}) \leq \alpha$. Then there exists some constant M which only depends on α such that $\|\mathbf{w}\| \leq M$, where $\mathbf{w} = (z_1, ..., z_n)$. Notice that $g(\mathbf{z}) \geq 0$ and $\forall 1 \leq i \leq p$, $\log(1 + \exp(-b_i(\mathbf{a}_i^T \mathbf{w} + z_{n+1}))) \geq 0$, we have

$$\log(1 + \exp(\mathbf{a}_1^T \mathbf{w} + z_{n+1})) \leq \alpha,$$
$$\log(1 + \exp(-\mathbf{a}_2^T \mathbf{w} - z_{n+1})) \leq \alpha.$$

This means

$$\exp(z_{n+1}) \leq (\exp(\alpha) - 1)\exp(-\mathbf{a}_1^T \mathbf{w}) \leq (\exp(\alpha) - 1)\exp(M\|\mathbf{a}_1\|),$$
$$\exp(-z_{n+1}) \leq (\exp(\alpha) - 1)\exp(\mathbf{a}_2^T \mathbf{w}) \leq (\exp(\alpha) - 1)\exp(M\|\mathbf{a}_2\|),$$

which proves the boundedness of z_{n+1} and completes the proof. □

(B.4) comes from the fact that g is bounded from below and $\mathbf{B} = \mathbf{I}$. For (C.2), first by [WCX18, Section 2.2], we know $f(\mathbf{u}) + g(\mathbf{v})$ is subanalytic. Moreover $\frac{1}{\gamma}\langle \mathbf{s} - \mathbf{u}, \mathbf{v} - \mathbf{u}\rangle + \frac{1}{2\gamma}\|\mathbf{v} - \mathbf{u}\|^2$ is subanalytic and maps bounded set to bounded set. Hence $\mathcal{D}_\gamma$ is subanalytic as the sum of these two functions by [XY13].

Appendix I: Convergence for Physical Simulation Problem

In this section, we analyze the convergence of Algorithm 1 on the physical simulation problem (33). In some cases, Assumption (A.1) would fail to hold. But since (A.1)–(A.3) are only used to prove the decrease of DR envelope, we would show that DR envelope is decreasing even if (A.1) is replaced by weaker assumption. Specifically, it is noted in [ZPOD19] that for physical simulation problem (33), if g is set to be the hyperelastic energy of StVK material, then g is only locally Lipschitz differentiable and hence doesn't satisfy (A.1). However, due to the monotone decreasing of DR envelope, we can except that [TP20, Theorem 4.1] still holds in this case, so that the convergence theorem in this paper remains valid.

In the following, we replace (A.1) by a weaker assumption:

(A.1)' φ_1 is Lipschitz differentiable on any bounded set.

Along with this assumption, we further assume:

(A.4) φ_1 is level-bounded and $\varphi_2 \geq 0$.

To simplify the notation, we define:
Definition I.1. We define $\text{lev}_{\leq \alpha}\varphi$ to be the set:

$$\text{lev}_{\leq \alpha}\varphi := \{\mathbf{x} \in \mathbb{R}^n : \varphi(\mathbf{x}) \leq \alpha\}.$$

We need the next initial value assumption:

(A.5) Let $\mathbf{Ax}_0 - \mathbf{Bz}_0 = \mathbf{c}$. $\mathbf{y}_0$ is chosen such that the augmented Lagrangian function $L(\mathbf{x}_0, \mathbf{z}_0, \mathbf{y}_0) = T_0 := f(\mathbf{x}_0) + g(\mathbf{z}_0) < \infty$ and $L(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_1) \leq L(\mathbf{x}_0, \mathbf{z}_0, \mathbf{y}_0)$. Assume $\mathbf{s}_0 = \mathbf{Ax}_1 - \mathbf{y}_1$.

Moreover, we need γ to be sufficiently small as the next assumption required:

(A.6) γ is sufficiently small such that $c_0 \leq 1$, where

$$c_0 = \sup_{\text{lev}_{\leq T_0+1}\varphi_1} \frac{\gamma}{2}\|\nabla \varphi_1(\mathbf{x})\|^2.$$

Here we note that such a γ must exist due to (A.4) and the fact that T_0 is independent of the choice of γ. In the following, we assume L_1 to be the Lipschitz modulus of $\nabla \varphi_1$ on the convexhull of the set $\text{lev}_{\leq T_0+1}\varphi_1$.

Lemma I.2. *Suppose that (A.5) holds. Then we have $\psi_E(s_0) \leq T_0$.*

Proof. Let $\mathbf{u}_0 = \text{prox}_{\gamma\varphi_1}(\mathbf{s}_0)$, then by the definition of DR envelope and Proposition 3.2 we can obtain that:

$$\psi_E(\mathbf{s}_0) = L(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_1) \leq T_0. \qquad □$$

Lemma I.3. *Assume (A.1)', (A.2)–(A.6) hold and $\gamma < \frac{1}{L_1}$. If it holds that*

$$\varphi_1(\mathbf{u}_k) \leq T_0 + 1, \quad \varphi_1(\mathbf{u}_{k+1}) \leq T_0 + 1,$$

then

$$\|\mathbf{u}_{k+1} - \mathbf{u}_k\| \leq \frac{1}{1 - \gamma L_1}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|, \|\mathbf{u}_{k+1} - \mathbf{u}_k\| \geq \frac{1}{1 + \gamma L_1}\|\mathbf{s}_{k+1} - \mathbf{s}_k\|.$$

Proof. By the optimality condition of $\mathbf{u}_k$ we know

$$\gamma \nabla \varphi_1(\mathbf{u}_k) + \mathbf{u}_k = \mathbf{s}_k.$$

Hence we can infer that

$$\|\mathbf{s}_{k+1} - \mathbf{s}_k\| \geq \|\mathbf{u}_{k+1} - \mathbf{u}_k\| - \gamma \|\nabla\varphi_1(\mathbf{u}_{k+1}) - \nabla\varphi_1(\mathbf{u}_k)\|,$$
$$\geq (1 - \gamma L_1)\|\mathbf{u}_{k+1} - \mathbf{u}_k\|,$$

The proof for second part is similar. This completes the proof. □

Lemma I.4. *Assume (A.1)', (A.2)–(A.6) hold and γ is sufficiently small. If for $\mathbf{s}_k$ it holds that:*

$$\psi_{\mathrm{E}}(\mathbf{s}_k) \leq T_0, \quad \varphi_1(\mathbf{u}_k) \leq T_0 + 1,$$

then the it also holds for $\mathbf{s}_{k+1}$.

Proof. Utilizing the definition of $\mathbf{u}_{k+1}$, we obtain that:

$$\varphi_1(\mathbf{u}_{k+1}) + \frac{1}{2\gamma}\|\mathbf{u}_{k+1} - \mathbf{s}_{k+1}\|^2 \leq \varphi_1(\mathbf{u}_k) + \frac{1}{2\gamma}\|\mathbf{u}_k - \mathbf{s}_{k+1}\|^2.$$

By the definition of $\mathbf{s}_{k+1}$ and (A.4) we have:

$$\varphi_1(\mathbf{u}_{k+1}) + \frac{1}{2\gamma}\|\mathbf{u}_{k+1} - \mathbf{s}_{k+1}\|^2$$

$$\leq \varphi_1(\mathbf{u}_k) + \varphi_2(\mathbf{v}_k) + \frac{1}{2\gamma}\|\mathbf{v}_k - (2\mathbf{u}_k - \mathbf{s}_k)\|^2,$$

$$= \psi_{\mathrm{E}}(\mathbf{s}_k) + \frac{1}{2\gamma}\|\mathbf{s}_k - \mathbf{u}_k\|^2,$$

$$= \psi_{\mathrm{E}}(\mathbf{s}_k) + \frac{\gamma}{2}\|\nabla\varphi_1(\mathbf{u}_k)\|^2,$$

$$\leq \psi_{\mathrm{E}}(\mathbf{s}_k) + c_0 \leq \psi_{\mathrm{E}}(\mathbf{s}_k) + 1,$$

where we have used the definition of ψ_{E} for the first equation, the optimality condition of $\mathbf{u}_k$ for the second equation, the definition of c_0 for the second inequality. Hence we have proved that

$$\varphi_1(\mathbf{u}_{k+1}) \leq \psi_{\mathrm{E}}(\mathbf{s}_k) + 1 \leq T_0 + 1.$$

For the estimation of $\psi_{\mathrm{E}}(\mathbf{s}_{k+1})$, we have:

$$\psi_{\mathrm{E}}(\mathbf{s}_{k+1})$$

$$\leq \varphi_1(\mathbf{u}_{k+1}) + \varphi_2(\mathbf{v}_k) + \langle \nabla\varphi_1(\mathbf{u}_{k+1}), \mathbf{v}_k - \mathbf{u}_{k+1}\rangle + \frac{1}{2\gamma}\|\mathbf{v}_k - \mathbf{u}_{k+1}\|^2,$$

where we have used the definition of DR envelope. We then utilize the definition of L_1 and [Nes18, Lemma 1.2.3] to obtain that

$$\varphi_1(\mathbf{u}_{k+1}) + \langle \nabla\varphi_1(\mathbf{u}_{k+1}), \mathbf{u}_k - \mathbf{u}_{k+1}\rangle \leq \varphi_1(\mathbf{u}_k) + \frac{L_1}{2}\|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2.$$

Moreover, we have:

$$\frac{1}{2\gamma}\|\mathbf{v}_k - \mathbf{u}_{k+1}\|^2$$

$$= \frac{1}{2\gamma}(\|\mathbf{v}_k - \mathbf{u}_k\|^2 + 2\langle \mathbf{v}_k - \mathbf{u}_k, \mathbf{u}_k - \mathbf{u}_{k+1}\rangle + \|\mathbf{u}_k - \mathbf{u}_{k+1}\|^2).$$

Combing all these three estimation together, we can obtain that:

$$\psi_{\mathrm{E}}(\mathbf{s}_{k+1}) \leq \psi_{\mathrm{E}}(\mathbf{s}_k) - (\frac{1}{2\gamma} - \frac{L_1}{2} - \gamma L_1^2)\|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2.$$

If $2\gamma^2 L_1^2 + \gamma L_1 < 1$, then we have:

$$\psi_{\mathrm{E}}(\mathbf{s}_{k+1}) \leq \psi_{\mathrm{E}}(\mathbf{s}_k) \leq T_0 + 1. \qquad \square$$

By induction and Lemma I.4 we can prove the next theorem:

Theorem I.5. *Assume (A.1)', (A.2)–(A.6) hold and γ is sufficiently small. Then we have:*

$$\psi_{\mathrm{E}}(\mathbf{s}_{k+1}) \leq \psi_{\mathrm{E}}(\mathbf{s}_k) - \left(\left(\frac{1}{2\gamma} - \frac{L_1}{2} - \gamma L_1^2\right)/(1 + L_1\gamma)\right)\|\mathbf{s}_{k+1} - \mathbf{s}_k\|^2.$$

Then all the convergence theorems in this paper can be stated based on Theorem I.5. We now verify (A.1)', (A.2)–(A.6) for the physical simulation problem (33). In this case, $\varphi_1(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{x}) = f(\mathbf{W}^{-1}\mathbf{x})$. So for the case where f is the hyperelastic energy of StVK material, then φ_1 satisfies (A.1)' because f satisfies (A.1)'. Notice that g is bounded from below and level-bounded, so φ_2 is lsc and proper by [TP20, Theorem 5.11]. Moreover, it can be verified that φ_2 is also level-bounded. So (A.2) is satisfied. (A.3) comes from the lower semi-continuity and level-boundedness of φ. (A.4) is trivial. (A.5) holds for the choice in [ZPOD19, Assumption 3.5]. (A.6) holds for for sufficiently small γ. Moreover, the aforementioned analysis also shows that (B.2) and (B.4) hold. Finally, since f, g are polynomial and hence semi-algebraic, so $\mathcal{D}_\gamma$ is also semi-algebraic, and then (C.2) follows from Remark 4.7.