DN-ADMM: Distributed Newton ADMM for Multi-agent Optimization

Yichuan Li\textsuperscript{1}, Nikolaos M. Freris\textsuperscript{2}, Petros Voulgaris\textsuperscript{3} and Dušan Stipanović\textsuperscript{4}

Abstract—In a multi-agent network, we consider the problem of minimizing an objective function that is expressed as the sum of private convex and smooth functions, and a (possibly) non-differentiable convex regularizer. We propose a novel distributed second-order method based on the framework of Alternating Direction Method of Multipliers (ADMM), by invoking approximate Newton iterations to the primal update corresponding to the differentiable part. In order to achieve a distributed implementation, the total Hessian matrix is split into a diagonal component (locally computable) and an off-diagonal component (that requires communication between neighboring agents). Subsequently, the Hessian inverse is approximated by a truncation of the Taylor expansion to \( K \) terms: this amounts to fully distributed updates entailing \( K \) distributed communication rounds. We establish global linear convergence to the primal-dual optimal solution under the assumption that the private functions are strongly convex and have Lipschitz continuous gradient. Numerical experiments demonstrate the merits of the approach comparatively with state-of-the-art methods.

I. INTRODUCTION

Decentralized consensus optimization problems consider an objective function consisting of a sum of cost functions which are only available at the corresponding agent of the underlying network. In many practical scenarios, a convex nonsmooth regularization function is augmented to the objective: typical examples include the \( \ell_1 \)-norm and the nuclear norm that are known to promote sparsity in the solution structure [21]. Formally, let \( \hat{x} \in \mathbb{R}^d \) denote the decision variable and consider a network containing \( n \) agents. Each agent \( i \) has access to the local private cost function \( f^i : \mathbb{R}^d \rightarrow \mathbb{R} \) and exchanges information with neighbors so as to minimize the global objective function:

\[
\hat{x}_* = \arg\min_{\hat{x} \in \mathbb{R}^d} \left\{ \sum_{i=1}^{n} f^i(\hat{x}) + g(\hat{x}) \right\}.
\]  

(1)

The global decision variable \( \hat{x}_* \) is common to all agents but the local cost function \( f^i(\cdot) \) is only available to agent \( i \). Although it is possible to aggregate information from the network and proceed to obtain an approximate solution in a centralized fashion, it is desirable to solve problem (1) through distributed exchanges. This is especially vital in view of the Big Data involved pertaining to large population count \( n \), problem dimension \( d \), and size of private data used in defining \( f^i(\cdot) \), that are typically shared via limited bandwidth communication subject to the requirement for high responsiveness and real-time decision-making in Cyber-Physical Systems (CPS). In the distributed consensus optimization framework, every agent \( i \) keeps a local copy of the decision variable \( x^i \) and communicates with their neighbors to cooperatively solve (1). The setting has found multiple applications in distributed control [1], power systems [2], machine learning [3], sensor networks [4], and controller design [5].

Proximal gradient [7] is a first-order method which generalizes the (projected) gradient method using the proximal operator to accommodate nonsmoothness. However, a distributed implementation is not possible when the nonsmooth term is not separable over the entries of the decision variable. Primal-dual methods for problem (1) introduce local copies of the decision vector \( x \) at each agent (i.e., agent \( i \) maintains \( x^i \)) and introduce equality constraints between an agent’s copy and its neighbors’, which enforces consensus in the network [22]. The Alternating Direction Method of Multipliers (ADMM) [8] splits the primal variable over the smooth and the nonsmooth part of the objective function and sequentially obtains iterates by minimizing the augmented Lagrangian function. Another closely related method is the Method of Multipliers (MM) [23] which performs joint minimization over primal variables. However, the presence of nonsmooth part renders such approach cumbersome, in fact inapplicable in our setting. By completion of squares and the use of the proximal operator, the authors in [11] derived the Proximal Augmented Lagrangian which is continuously differentiable and eliminates the primal variable corresponding to the nonsmooth part. At each iteration of ADMM (and MM), it requires to solve an optimization sub-problem to obtain the next update, thus maybe especially costly in terms of computations. To avoid exact minimization steps required by the aforementioned primal dual algorithms, several approximation schemes have been proposed in the literature [12]-[13]. Decentralized Linearized ADMM (DLM) approximates the convex objective, at each iteration, with a first-order model and yields per-iteration costs comparable with gradient methods but suffers from slow convergence speed. Newton’s method uses second-order information and features high convergence rates along with insensitivity to the condition number of the objective. However, Newton’s iterations require solving linear systems as well as performing backtracking to ensure global convergence; this makes a distributed implementation impractical.
We propose a Distributed Newton method in the framework of ADMM (DN-ADMM) and establish global linear convergence without the need for backtracking. Compared to existing algorithms such as [14,15], we support nonsmooth regularizers using second order information without computing the generalized gradient of the proximal mapping. Numerical simulations demonstrate the advantage of our algorithm over existing methods.

**Notation:** Vectors $x \in \mathbb{R}^d$ are represented as column vectors, denoted with lower case letters and matrices $A \in \mathbb{R}^{n \times n}$ with capital letters. We use superscript to denote the vector component $x^i$ and subscript to denote the variable at the $t$-th iteration $x_t$. For a matrix $A$, we denote $A^{ij}$ as the $ij$-th entry of $A$, and when a norm subscript is not provided, $||x||$ and $||A||$ denote the Euclidean norm of a vector and the corresponding induced norm of a matrix, respectively. The norm of a vector $x$ with respect to a positive definite matrix $P$ is denoted as $||x||_P = (x^TPx)^{1/2}$ and the set $\{1, 2, ..., n\}$ is abbreviated as $[n]$.

**II. PRELIMINARIES**

**A. Problem formulation**

Consider an undirected connected graph $G = (V, \mathcal{E}, \mathcal{A})$ of order $n$ with vertex set $V = \{v^1, ..., v^n\}$, edge set $\mathcal{E} \subseteq V \times V$, where each edge is denoted as $e^i = (v^i, v^j)$, and the signed incidence matrix $\mathcal{A} \in \mathbb{R}^{n \times m}$, where $m = |\mathcal{E}|$ is the number of edges. We denote the neighborhood of vertex $i$ as $\mathcal{N}^i = \{v^j \in V : (v^i, v^j) \in \mathcal{E}\}$. Problem (1) can be reformulated to the standard consensus form for distributed optimization as follows. Let $x^i \in \mathbb{R}^d$ be the local copy of the decision variable held by the agent $i$ and introduce an extra variable $\hat{z} \in \mathbb{R}^d$:

$$
\begin{align*}
\min_{x^i, \hat{z} \in \mathbb{R}^d} & \left\{ \sum_{i=1}^{n} f^i(x^i) + g(\hat{z}) \right\} \\
\text{s.t.} & \quad x^i = x^j \quad \text{for all } i, j \in \mathcal{N}^i, \\
& \quad x^i = \hat{z} \quad \text{for some } l \in [n].
\end{align*}
$$

(2)

If the underlying network graph $G$ is connected, then the constraints in (2) enforce network-wide consensus and (2) is equivalent to (1). The purpose of adding an extra variable $\hat{z}$ serves to separate the smooth and the nonsmooth part of the objective function and thus casts the problem to the ADMM framework. Note that in (2), we only enforce equality constraint between $x^i$ and $\hat{z}$ for some agent $l$ in the network to minimize computation burden. Other frameworks such as enforcing the equality constraint for all agents have been considered as well and our approach is applicable therein with few modifications. We may express the above optimization problem compactly by aggregating $F(x) = \sum_{i=1}^{n} f^i(x^i)$, where we stack local copies into a vector $x = [(x^1)^\top, ..., (x^n)^\top]^\top \in \mathbb{R}^{nd}$, and further define $G(z) = g(\hat{z})$ where $z = [0, ..., 0, \hat{z}]^\top \in \mathbb{R}^{(m+1)d}$. Therefore, (2) becomes:

$$
\begin{align*}
\min_{x, \hat{z}} & \quad F(x) + G(z) \\
\text{s.t.} & \quad B^\top x = z,
\end{align*}
$$

(3)

where we use Kronecker product to define $B = [A, E^1] \in \mathbb{R}^{nd \times (m+1)d}$, $A = A \otimes I_d$, and $E^1 = e^1 \otimes I_d$ is obtained by using the coordinate selection vector $e^1 \in \mathbb{R}^n$, which is a zero vector except the $l$-th entry being equal to 1. We proceed to state our first assumption on the objective function.

**Assumption 1.** The smooth local cost functions $\{f^i(\cdot)\}_{i=1}^{n}$ are twice differentiable with bounded Hessian as follows:

$$
\begin{align*}
m_f I \preceq \nabla^2 f^i(x^i) \preceq M_f I, \quad \forall i \in [n].
\end{align*}
$$

(4)

The lower bound above implies local cost functions are strongly convex with parameter $m_f > 0$ while the upper bound implies Lipschitz continuity of the gradient $\nabla f^i(\cdot)$ with constant $M_f > 0$. Moreover, since $F(x) = \sum_{i=1}^{n} f^i(x^i)$, $\nabla^2 F(x)$ is a block-diagonal matrix with the $i$-th block being $\nabla^2 f^i(x^i)$. Consequently, the same bounds as in (4) apply to the Hessian of the total cost function as well, i.e., $m_f I \preceq \nabla^2 F(x) \preceq M_f I$.

**B. Proximal operators**

Proximal operators can be seen as the generalization of the projection mapping for nonsmooth functions. For a closed, proper, and convex function $g : \mathbb{R}^d \to \mathbb{R}$, we define its proximal operator with parameter $\mu > 0$ as:

$$
\begin{align*}
\text{prox}_{\mu g}(v) = \arg\min_{z} \left\{ g(z) + \frac{1}{2\mu} ||v - z||^2 \right\},
\end{align*}
$$

(5)

and the associated value function, also known as the Moreau envelope,

$$
\begin{align*}
M_{\mu g}(v) = g(\text{prox}_{\mu g}(v)) + \frac{1}{2\mu} ||v - \text{prox}_{\mu g}(v)||^2.
\end{align*}
$$

(6)

The proximal mapping (5) is Lipschitz continuous with constant 1, differentiable almost everywhere, and single-valued [15]. For example, when $g(\cdot)$ is the indicator function of some convex closed set $\mathcal{X}$, its proximal operator reduces to the projection mapping of $\mathcal{X}$, i.e., $\Pi_{\mathcal{X}}(v) = \arg\min_{z \in \mathcal{X}} ||v - z||^2$.

For $g(z) = \sum_{t \in \mathcal{X}} |z_t|$, the proximal operator has a closed form expression known as the soft-thresholding, i.e., $\text{prox}_{\mu g}(v) = \text{sign}(v) \max(\|v\| - \mu, 0)$. More information on proximal operators and proximal algorithms can be found in [7].

**C. Alternating Direction Method of Multipliers**

The augmented Lagrangian associated with (3) with parameter $\mu > 0$ is obtained by augmenting a quadratic penalty term of the constraint to the Lagrangian: $L_{\mu}(x, z; y) = F(x) + G(z) + y^\top (B^\top x - z) + (1/2\mu) \|B^\top x - z\|^2$, where $y \in \mathbb{R}^{(m+1)d}$ is the dual variable associated with constraints. ADMM takes advantage of the fact that primal variables are separated among smooth and nonsmooth parts of the cost function and sequentially updates them by minimizing $L_{\mu}(x, z; y)$ as follows:

$$
\begin{align*}
x_{t+1} = \arg\min_{x} L_{\mu}(x, z_t; y_t), \\
z_{t+1} = \arg\min_{z} L_{\mu}(x_{t+1}, z; y_t), \\
y_{t+1} = y_{t} + \frac{1}{\mu} \nabla y L_{\mu}(x_{t+1}, z_{t+1}; y_t).
\end{align*}
$$

(7a, 7b, 7c)

In the following section, we propose a variant of ADMM method with an additional primal update and perform inexact
minimization for step (7a). We extend the result in [13] and [19] to include a non-smooth regularizer by building a quadratic model at each iteration with Hessian inverse approximated by truncated Taylor expansion, while invoking proximal operators to update the z-variable.

III. Second-order approximation and distributed implementation

A. Second order approximation

Newton’s method can be interpreted as minimizing a quadratic model incurred by the Hessian of the objective function, at each iteration. However, backtracking is required to ensure global convergence and damped steplizes are often used [20]. A quadratic model for \( \hat{H}_t \) can be constructed as \( \hat{L}_t(x_t, z_t; y_t) \approx L_t(x_t, z_t; y_t) + \nabla x_L_t(x_t, z_t; y_t)\frac{1}{2}(x_t - x_t) + \frac{1}{2}(x_t - y_t)\frac{1}{2}(x_t - x_t) + \frac{1}{2}\|x_t - x_t\|^2. \]

The last term \( \frac{1}{2}\|x_t - y_t\|^2 \) aims to keep the next iterate close to the current one and therefore renders a numerically robust algorithm. We obtain the primal update for \( x_{t+1} \) as:

\[
x_{t+1} = x_t - \frac{1}{\mu}H_t^{-1}\nabla F(x_t) + \frac{1}{\mu}BB^\top x_t - \frac{1}{\mu}Bz_t + By_t.
\]

The Hessian \( H_t = \nabla_{xx}\hat{L}_t \) is given by:

\[
H_t = \nabla^2 F(x_t) + \frac{1}{\mu}BB^\top + \epsilon I_{nd}, \tag{8}
\]

where \( I \) is the identity matrix of appropriate dimension.

Recall that \( e_i \) is the coordinate selection vector and since \( \nabla^2 F(x_t) \) is block-diagonal and \( BB^\top = (L + \epsilon(e_i e_i^\top)) \otimes I_d \) where \( L \) is the graph Laplacian, the Hessian \( H_t \) is distributedly computable by using local information and communication with neighbors.

However, the Hessian inverse \( H_t^{-1} \) is not necessarily distributedly computable. Motivated by [13] and [16], we employ a matrix splitting technique so that the Hessian inverse is suitable for Taylor expansion and, therefore, we can truncate the series to obtain a finite-term approximation of the Hessian inverse. We first decompose the Hessian as \( H_t = D_t \otimes N - N \) where \( D_t \) is block diagonal and \( N \) encodes off-diagonal entries. In specific, \( D_t \succ 0 \) is equal to:

\[
D_t = \nabla^2 F(x_t) + \frac{1}{\mu}(e_i e_i^\top)^\top \otimes I_d + \frac{1}{\mu}L_{dia} \otimes I_d + \epsilon I_{nd}, \tag{9}
\]

where \( e_i \in \mathbb{R}^n \) and \( N \preceq 0 \):

\[
N = \frac{1}{\mu}(L_{dia} - L_{off}) \otimes I_d, \tag{10}
\]

where \( L_{dia} \) and \( L_{off} \) are the diagonal and off-diagonal components of the graph Laplacian matrix \( L \). With this matrix splitting, we can expand the Hessian inverse as \( H_t^{-1} = D_t^{-\frac{1}{2}}(I - D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}})^{-1}D_t^{-\frac{1}{2}}, \) and use Taylor expansion to express \( (I - D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}})^{-1} = \sum_{i=0}^{\infty}(D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}})^i \). We truncate the series to a finite sum and show that the error incurred is bounded. To ensure validity of this expansion, we need to show that the spectral radius of \( D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}} \) is strictly less than 1 which we prove next.

**Lemma 1.** The spectral radius of \( D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}} \) is strictly less than 1.

**Proof:** The following proof is inspired by [13]. We first define \( C = \frac{1}{\mu}L_{dia} \otimes I \succ 0 \). We bound the spectral radius by considering upper bounds of each term as \( \|D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}}\| \leq \|D_t^{-\frac{1}{2}}C\| \leq \|C^{-\frac{1}{2}}NC^{-\frac{1}{2}}\| \). For the first term, note that \( D_t^{-\frac{1}{2}}C^\frac{1}{2}\) is a block-diagonal matrix with i-th block specified as:

\[
(D_t^{-\frac{1}{2}}C^\frac{1}{2})_{ii} = (\nabla^2 f_i(x_t^i) + \frac{1}{\mu_{ii}}((e_i^\top)^2 + 2L_{dia} + \mu_{ii})I^{-\frac{1}{2}}) \times (\frac{2}{\mu}L_{dia}^{-\frac{1}{2}})^{-\frac{1}{2}}.
\]

From Assumption 1 and the fact that \( 1 \leq (L_{dia})_{ii} \leq n - 1 \) for any i, we can bound \( (D_t^{-\frac{1}{2}}C^\frac{1}{2})_{ii} \) as:

\[
(D_t^{-\frac{1}{2}}C^\frac{1}{2})_{ii} \leq \left( \frac{2(n-1)}{(\mu_{ii} + \epsilon + 2(n-1))} \right) I. \tag{12}
\]

Since \( C^{-\frac{1}{2}}NC^{-\frac{1}{2}} = C^{-\frac{1}{2}}(NC^{-1})C^\frac{1}{2} \), we can bound the eigenvalues of \( C^{-\frac{1}{2}}NC^{-\frac{1}{2}} \) by investigating the eigenvalues of \( NC^{-1} = \frac{1}{2}(L_{dia} - L_{off})L_{dia}^{-1} \otimes I \). Note that \( L_{dia} \) is diagonal with entries being equal to the number of neighbors of the corresponding agent, while \( L_{off} \) is symmetric with \( i\_j \)-th entry being equal to \(-1 \) if and only if \((i\_j, j\_i) \in E \) and 0 otherwise.

Therefore,

\[
NC^{-1} = \frac{1}{2} \begin{bmatrix}
1 & -\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} & \ldots & -\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} \\
-\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} & 1 & \ldots & -\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} \\
\vdots & \vdots & \ddots & \vdots \\
-\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} & -\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} & \ldots & 1
\end{bmatrix} \otimes I. \tag{13}
\]

The sum of non-diagonal components of column i is \( \sum_{j=1,j\neq i}^{n} -\frac{1}{2} \frac{L_{dia}^{-\frac{1}{2}}}{L_{dia}} \leq \frac{1}{2} \), where we have used the property of graph Laplacian \( \sum_{j} L_{ij} = 0 \) for any i. Since each diagonal entry of the above matrix equals \( \frac{1}{2} \), by applying Gershgorin circle theorem, we get that all eigenvalues of the matrix shown in (13) lie in the circle centered at \( \frac{1}{2} \) with radius \( \frac{1}{2} \). Furthermore, for matrices \( A \in \mathbb{R}^{m \times n} \) and \( B \in \mathbb{R}^{m \times m} \), the eigenvalues of \( A \otimes B \) are given by \( \lambda_i(A)\lambda_j(B) \) for \( 1 \leq i \leq n \) and \( 1 \leq j \leq m \) where \( \lambda_i \) stands for eigenvalues. We conclude that \( 0 \leq \lambda_i(N^{-1}) \leq 1 \). Therefore, combining with (12), we obtain an upper bound for the spectral radius of \( D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}} \) as follows:

\[
\sigma_{max}(D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}}) \leq \frac{2(n-1)}{(\mu(m_f + \epsilon) + 2(n-1))} < 1. \tag{14}
\]

B. Distributed implementation

We proceed to develop DN-ADMM with Hessian inverse approximated by truncating the Taylor series and show its distributed implementation in Algorithm 1. Define the K-th order Hessian inverse approximation as:

\[
\hat{H}_t^{-1}(K) = D_t^{-\frac{1}{2}} \sum_{i=0}^{K}(D_t^{-\frac{1}{2}}ND_t^{-\frac{1}{2}})^iD_t^{-\frac{1}{2}}. \tag{15}
\]

Since only the last block of \( z_t \) is nonzero, step (7b) of ADMM is equivalent to:

\[
\hat{z}_{t+1} = \arg\min_{\hat{z}} \|g(\hat{z}) + (y_{mt+1}^{-1})^\top(x_t^i - \hat{z}) + \frac{1}{2\mu}\|x_t^i - \hat{z}\|^2\|.
\]

By completion of squares, this update reduces to the proximal mapping of \( g(\cdot) \) evaluated at \( (x_t^i + \mu y_{mt+1}^{-1})^\top \) with parameter \( \mu \), i.e.,

\[
\hat{z}_{t+1} = \text{prox}_{\mu g}(x_t^i + \mu y_{mt+1}^{-1}). \tag{16}
\]
Note that we have used the updated value $x_{t+1}$ and the current dual $y_{t+1}$ to obtain $\hat{z}_{t+1}$. In DN-ADMM, we perform an extra proximal mapping step before updating $x_{t+1}$ to mimic the behavior of the method of multipliers. The updates of order $K$ (number of terms kept in the Taylor expansion) of DN-ADMM are given below:

$$
\hat{\theta}_{t+1} = \text{prox}_{\mu}(x_t + \mu y_{t+1}^{m+1}),
$$  

(17a)

$$
x_{t+1} = x_t - H_t^{-1}(K)\nabla F(x_t) + \frac{1}{\mu} B(x_t - \hat{\theta}_{t+1}) + B y_t,
$$  

(17b)

$$
\hat{z}_{t+1} = \text{prox}_{\mu}(x_{t+1} + \mu y_{t+1}^{m+1}),
$$  

(17c)

$$
y_{t+1} = y_t + \frac{1}{\mu} B^T(x_{t+1} - z_{t+1}),
$$  

(17d)

where $\theta_{t+1} \in \mathbb{R}^{(m+1)d}$ is a zero vector except the last blocking being equal to $\hat{\theta}_{t+1}$, similarly for $\hat{z}_{t+1}$ and $z_{t+1}$. In contrary to [11] and [14], proximal operators do not explicitly appear in the $x$-update and therefore the approximated Hessian $H_t$ does not include the generalized gradient of proximal mapping. This results in a more efficient implementation, that avoids the need for computing (potentially cumbersome) generalized Hessians pertinent to the non-differentiable part. Moreover, $K$-th order approximation of the Hessian inverse requires $K$ rounds of communication among agents. Specifically, define $u_t(K-1) = H_t^{-1}(K-1)\nabla x_L(x_t, \theta_t, y_t)$ so that the $x$-updates is simply $x_{t+1} = x_t - u_t(K-1)$. From (15), we express the next order update direction $u_t(K)$ as:

$$
u_t(K) = D_t^{-\frac{1}{2}} \sum_{i=0}^{K-1} (D_t^{-\frac{1}{2}} N D_t^{-\frac{1}{2}}) D_t^{-\frac{1}{2}} \nabla x_L \mu + D_t^{-1} N D_t^{-\frac{1}{2}} \sum_{i=0}^{K-1} (D_t^{-\frac{1}{2}} N D_t^{-\frac{1}{2}}) D_t^{-\frac{1}{2}} \nabla x_L \mu
$$

$$
d_t^{-1} \nabla x_L \mu + d_t^{-1} N u_t(K-1),
$$

(18)

where we have suppressed the argument of $\nabla x_L \mu(x_t, \theta_t, y_t)$ for brevity. Note that since $D_t$ is block diagonal and $N$ respects the network structure, we can compute $u_t(K)$ by just one round of distributed communication once $u_t(K-1)$ is obtained. Therefore, to obtain the $K$-th order approximation of the Hessian inverse, every agent initializes with $u_t(0) = D_t^{-\frac{1}{2}} \nabla x_L \mu$, then communicates for $K$ rounds with its neighbors. Moreover, since $\theta_{t+1}, z_{t+1}$ has first $m$ blocks of entries being zero, only the $(m+1)$th block, i.e., the subvector $\theta_{t+1}, z_{t+1} \in \mathbb{R}^d$, needs to be computed, only using information held by the $l$th agent.

Updates shown in (17) follow directly from the framework of ADMM and are convenient for analysis purpose. We further present an efficient distributed implementation in Algorithm 1. From the definition of $B$, we note that in (17b), the dual variable $y_t$ is used in the update of $x_{t+1}$ in the form of $B y_t = A y_t^{[m]} + E y_t^{[m+1]}$, where $y_t^{[m]} \in \mathbb{R}^{md}$ denotes the concatenation of the first $m$ blocks of $y_t$. By defining $\phi_t = A y_t^{[m]} \in \mathbb{R}^{nd}$, we eliminate the need to exchange dual variables during the primal update and obtain a distributed implementation as follows. We let each agent hold the corresponding pair $(x, \phi_t)$ while the $l$th agent (which updates the nonsmooth variable) additionally holds $y_{t+1}^{m+1}$. At each iteration, agent $l$ begins by computing the block $D_t^{i^t}$ using local information and proceeds to compute $h_t^i := \nabla x_L \mu^i$ using values $x^i_t$ obtained from its neighbors (step 3). The $l$-th agent additionally evaluates the proximal mapping associated with the nonsmooth regularizer $g(\cdot)$ and updates $h_t^l$ accordingly in step 6. Steps 9-11 amount to $K$ rounds of communication among the network to compute $u_t(K)$ as in (18). Agents proceed to perform primal and dual updates in steps 12 and 13 respectively while the $l$-th agent additionally performs proximal mapping with updated iterates and update the additional dual variable $y_{t+1}^{m+1}$ in steps 15 and 16 respectively.

IV. CONVERGENCE ANALYSIS

In this section, we prove that iterates generated by (17) converge linearly to the optimal $x_*, z_*$ and $y_*$. We first state the KKT conditions of the problem along with an additional assumption on cost functions needed for our analysis in the following.

**KKT conditions for problem (3):**

$$
\nabla F(x_t) + B y_t = 0 \quad \text{(KKTa)}
$$

$$
\partial g(z) - y_*^{m+1} \geq 0 \quad \text{(KKTb)}
$$

$$
B^T x_* - z_* = 0 \quad \text{(KKTc)}
$$

Assumption 2. The function $g(z)$ is proper, closed, and convex, i.e., $\forall w_1 \in \partial g(z_1), w_2 \in \partial g(z_2)$,

$$
(z_1 - z_2)^T (w_1 - w_2) \geq 0.
$$

Lemma 2. Consider the dual updates specified in (17d), with zero initialization. Then $y_t$ is in the range of $B^T$ for all $t$. Moreover, there is a unique dual optimal $y_*$ in the column space of $B^T$.

*Proof:* See Appendix.
The following lemma characterizes a property of the dual iterates in relation to the nonsmooth variable \( \tilde{z}_{t+1} \).

**Lemma 3.** Consider the update \( \hat{z}_{t+1} = \text{prox}_{\mu g}[x_{t+1} + \mu y_{t+1}] \) in (17). It holds that:
\[
y_{t+1} \in \partial g(\hat{z}_{t+1}).
\]
(20)

**Proof:** See Appendix.

**Lemma 4.** Consider the updates of \( x_{t+1} \) and \( y_{t+1} \) specified in (17), and recall the approximated Hessian (15). For primal/dual optimal solutions \( x_\ast \) and \( y_\ast \), it holds that:
\[
\nabla F(x_{t+1}) - \nabla F(x_t) + B(y_{t+1} - y_t) + e_t
\]
\[
\quad + \epsilon(x_{t+1} - x_t) = 0,
\]
(21)
where \( e_t \) is defined as:
\[
e_t = \nabla F(x_t) + \nabla^2 F(x_t)(x_{t+1} - x_t) - \nabla F(x_t)
\]
\[
\quad + (H_t - H_t)(x_{t+1} - x_t) + \frac{1}{\mu}B(\hat{z}_{t+1} - \theta_{t+1}).
\]
(22)

**Proof:** See Appendix.

**Lemma 5.** Consider the error term defined in (22), it is bounded above as:
\[
\|e_t\| \leq \gamma \|x_{t+1} - x_t\|,
\]
where \( \gamma \) is defined as:
\[
\gamma = 2MF + 2\|M_f + \epsilon + \frac{1+2(a-1)}{\mu}(\mu m^{-1})^{K+1} + \eta F^2
\]
(23)

**Proof:** See Appendix.

The following vector matrix defined in terms of primal/dual variables and algorithm hyperparameters:
\[
v = \begin{bmatrix} x \\ y \end{bmatrix}, \quad H = \begin{bmatrix} I & 0 \\ 0 & \mu I \end{bmatrix}.
\]
(25)

Note that \( v \) is the concatenation of \( (x, y) \) and \( H \) is a diagonal matrix with blocks being corresponding scaling parameters. By defining \( v_\ast \), as the concatenation of the optimal variables, i.e., \( v_\ast = [x_\ast, y_\ast]^T \), we prove the convergence of the iterates to the optimality by showing that the Lyapunov function \( \|v_t - v_\ast\|_H \) converges to zero.

**Theorem 1.** Consider the iterates generated by DN-ADMM in (17). Denote the smallest positive eigenvalue of \( B^T B \) as \( \lambda_{\min} \). Let \( \beta, \eta > 1, \zeta \in \left(\frac{m_f + M_f}{2M_fM_f}, \frac{1}{2}\right) \) be arbitrary constants and choose \( \epsilon \geq \left[2M_f + \frac{m_f + 2M_f(n-1)^2}{\mu M_f} \right] \). Recall the definition of \( v \) and \( H \) in (25). If Assumptions 1, 2 hold, then the sequence \( \|v_t - v_\ast\|_H \) converges linearly:
\[
\|v_{t+1} - v_\ast\|_H^2 \leq \frac{1}{1 + \delta} \|v_t - v_\ast\|_H^2
\]
(26)
where
\[
\delta = \min \left\{ \frac{2m_fM_f}{\lambda_{\min}(m_f + M_f)} - 1, \frac{2M_f}{\lambda_{\min}(m_f + M_f)} - 1, \frac{\lambda_{\min}(m_f + M_f)}{\mu M_f} \right\}.
\]
(27)

**Proof:** Since \( F(x) \) is strongly convex with parameter \( m_f \) and the gradient \( \nabla F(x) \) is Lipschitz continuous with parameter \( M_f \), it holds that [25]:
\[
m_f \|x_{t+1} - x_t\|^2 + \frac{1}{2M_f} \|\nabla F(x_{t+1}) - \nabla F(x_t)\|^2 \leq \|x_{t+1} - x_t\|^2 (\nabla F(x_{t+1}) - \nabla F(x_t)).
\]
(28)
We substitute the expression for \( \nabla F(x_{t+1}) - \nabla F(x_t) \) from Lemma 4 and denote the left-hand-side of (28) by \( \text{LHS} \); this can be upper bounded by:
\[
\text{LHS} \leq -(x_{t+1} - x_\ast)^T B(y_{t+1} - y_\ast) - (x_{t+1} - x_\ast)^T e_t
\]
\[
\quad - \epsilon(x_{t+1} - x_\ast)^T (x_{t+1} - x_t).
\]
From the dual update (17) and KKTc, we have \(- (x_{t+1} - x_\ast)^T B = - \mu (y_{t+1} - y_\ast)^T (z_{t+1} - z_\ast)^T \). Further substituting this expression for \(- (x_{t+1} - x_\ast)^T B \) into above inequality gives:
\[
\text{LHS} \leq -\mu (y_{t+1} - y_\ast)^T (y_{t+1} - y_\ast) - (z_{t+1} - z_\ast)^T (x_{t+1} - x_\ast) - \epsilon(x_{t+1} - x_\ast)^T (x_{t+1} - x_t).
\]
(29)
Since \((z_{t+1} - z_\ast)^T (y_{t+1} - y_\ast) = (\hat{z}_{t+1} - \hat{z}_\ast)^T (y_{t+1} - y_{t+1})^m + 1 \in (\hat{z}_{t+1} - \hat{z}_\ast)^T (\partial g(\hat{z}_{t+1}) - \partial g(z_\ast)) \geq 0.
Therefore, \( \text{LHS} \) can be further bounded as:
\[
\text{LHS} \leq -\mu (y_{t+1} - y_\ast)^T (y_{t+1} - y_\ast) - (x_{t+1} - x_\ast)^T e_t
\]
We use the identity \(-2(a-b)^T (a-c) = \|b - c\|^2 - \|a-b\|^2 - \|a-c\|^2 \) with \( a = y_{t+1}, b = y_t \) and \( c = y_\ast \) and similarly, with \( x_{t+1}, x_t, x_\ast \). Multiplying the above inequality by 2 on both sides yields:
\[
\|x_{t+1} - x_t\|^2 (\frac{2m_fM_f}{\mu M_f} - 1) + \|\nabla F(x_{t+1}) - \nabla F(x_t)\|^2 \leq \mu \|y_{t+1} - y_\ast\|^2 - 2\|x_{t+1} - x_t\|^2 - \|x_{t+1} - x_\ast\|^2 - \|x_{t+1} - x_\ast\|^2 - \|x_{t+1} - x_\ast\|^2
\]
(30)
Since \(2(x_{t+1} - x_\ast)^T e_t \leq \frac{1}{2}\|x_{t+1} - x_\ast\|^2 - \zeta \|e_t\|^2 \forall \zeta > 0 \) and using the definition (25), we rewrite (30) as:
\[
\|x_{t+1} - x_\ast\|^2 (\frac{2m_fM_f}{\mu M_f} - 1) + \|\nabla F(x_{t+1}) - \nabla F(x_t)\|^2 \leq \|v_t - v_\ast\|^2_\mu - \|v_t - v_\ast\|^2_\mu.
\]
(31)
From the dual update in (17a), the first \( m \) block of the dual variable is updated as \( y_{t+1}^m = y_t^m + \frac{1}{\mu A} x_{t+1} \). After rearranging and combining with the first \( m \) block of KKTc, we have \( \|y_{t+1} - y_\ast\|^2 \geq \left\|y_{t+1}^m - y_t^m\right\|^2 = \frac{\mu}{\mu A^2} \|x_{t+1} - x_\ast\|^2 \). To prove linear convergence, we need to show that for some \( \delta > 0, \|v_t - v_\ast\|^2 \leq \frac{1}{\delta} \|v_t - v_\ast\|^2_\mu \). Therefore, along with the lower bound for \( \|y_{t+1} - y_\ast\|^2 \) substituted in, we show that the LHS of (31) is lower bounded by \( \|v_t - v_\ast\|^2_\mu \), i.e.,
\[
\|x_{t+1} - x_\ast\|^2 (\frac{2m_fM_f}{\mu M_f} - 1) + \|\nabla F(x_{t+1}) - \nabla F(x_t)\|^2 \geq \|v_t - v_\ast\|^2_\mu - \zeta \|e_t\|^2
\]
(32)
We proceed to establish such a bound by using the components of the LHS of (32). From (21), we have:
\[
B(y_{t+1} - y_\ast) = - \nabla F(x_{t+1}) - \nabla F(x_\ast) + \epsilon(x_{t+1} - x_\ast) + e_t
\]
(33)
Furthermore, it is easy to verify that when \( a = b + c \) implies
that \( \forall \beta > 1 \), it holds that \( \|a\|^2 \leq \frac{\beta}{\delta} \|b\|^2 + \beta \|c\|^2 \). Applying this formula two times with respective constants \( \beta, \eta > 1 \), we obtain an upper bound on the LHS of (33) as:

\[
\|B(y_{t+1} - y_t)\|^2 \leq \frac{\beta^2}{\delta} \|x_{t+1} - x_t\|^2 + \frac{\beta \eta}{\lambda_{\min}((q-1)B)} \|e_t\|^2 + \beta \eta \|\nabla F(x_{t+1}) - \nabla F(x_t)\|^2.
\]

From Lemma 2, it follows that dual iterates are orthogonal to the kernel of \( B \) and therefore, \( \|B(y_{t+1} - y_t)\|^2 \geq \lambda_{\min} \|y_{t+1} - y_t\|^2 \), where \( \lambda_{\min} \) is the smallest positive eigenvalue of \( B^T B \). Therefore, we have the following inequality:

\[
\|y_{t+1} - y_t\|^2 \leq \frac{\delta \beta^2 \eta}{\lambda_{\min}((q-1)B)} \|x_{t+1} - x_t\|^2 + \frac{\beta \eta}{\lambda_{\min}((q-1)B)} \|e_t\|^2 + \beta \eta \|\nabla F(x_{t+1}) - \nabla F(x_t)\|^2.
\]

(34)

Combining upper bounds for \( \|y_{t+1} - y_t\|^2 \|l_t\|^2 \) in (34) and (23) respectively, from the definition of \( \|v_{t+1} - v_t\|_H^2 = \mu \|y_{t+1} - y_t\|^2 + \epsilon \|x_{t+1} - x_t\|^2 \), we obtain:

\[
\|v_{t+1} - v_t\|_H^2 \leq (\frac{\mu \beta^2 \eta}{\lambda_{\min}((q-1)B)} + \frac{\mu \beta \eta}{\lambda_{\min}((q-1)B)}) \|x_{t+1} - x_t\|^2 + \epsilon \|x_{t+1} - x_t\|^2.
\]

Therefore, to satisfy (32), it is sufficient to show for some \( \delta > 0 \),

\[
\|x_{t+1} - x^*\|^2 \leq \frac{2 \mu^2 \eta^2}{\lambda_{\min}((q-1)B)} \|x_{t+1} - x_t\|^2 + \frac{1}{\lambda_{\min}((q-1)B)} \|x_{t+1} - x_t\|^2 + \frac{1}{\lambda_{\min}((q-1)B)} \|x_{t+1} - x_t\|^2 + \frac{\mu \beta^2 \eta}{\lambda_{\min}((q-1)B)} \|x_{t+1} - x^*\|^2 \geq 0.
\]

Inequality (35) holds if \( \delta \) is chosen to satisfy:

\[
\frac{2 \mu^2 \eta^2}{\lambda_{\min}((q-1)B)} - \delta \leq 0,
\]

\[
\delta - \frac{\mu \beta^2 \eta}{\lambda_{\min}((q-1)B)} \leq 0,
\]

Inequalities (36) are satisfied if \( \delta \) chosen as in (27) and \( \zeta, \epsilon, \gamma \) as in the statement (which guarantees that \( \delta > 0 \) in view of the first and the second quantity in (27)). Therefore, inequality (32) is satisfied and we conclude (26).

V. NUMERICAL SIMULATIONS

We apply DN-ADMM(\( K \)) to a distributed LASSO problem where each agent holds a local quadratic cost function with a common \( \gamma \)-weighted \( \ell_1 \)-norm regularizer,

\[
\minimize_{x \in \mathbb{R}^n} \left\{ \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \|A_ix - b_i\|^2 + \gamma \|x\|_1 \right\}.
\]

We generate a connected random binomial graph of \( n \) agents where an edge is drawn i.i.d Bernoulli(\( p \)) with \( p = 0.2 \). Two real datasets are considered from the LIBSVM and UCI Machine Learning Repository[2] the space\_ga dataset and Combined Cycle Power Plant dataset (CCPP). The space\_ga dataset contains observations on U.S. county votes for the 1980 presidential election with 6 features and the CCPP dataset predicts the net hourly electrical energy output of a power plant with 4 features. We took 3,080 instances from the space\_ga dataset and 9,000 instances from the CCPP dataset, and evenly distributed it among \( n \) agents. In Figs. 1 and 2 we plot the averaged relative cost \( \frac{1}{n} \sum_{i=1}^{n} l(x_{i}) - l(x_{i}^{\star}) \) versus the number of iterations and the number of communication rounds per node. We compare the performance of between DN-ADMM(\( K \)) with state-of-the-art first-order methods: P2D2 [24] and PG-EXTRA [18] with the mixing matrix generated using the Metropolis rule and the Laplacian based constant edge weight matrix, respectively. We do not compare against second order methods since existing ones do not support nonsmooth regularizers. For all simulations, \( \gamma = 0.002 \), stepsizes for PG-EXTRA and P2D2 are hand tuned to achieve fastest convergence. We observe significant speedup of DN-ADMM(\( K \)) over first order methods from Fig. 1 and Fig. 2. Moreover, we observe that by communicating with neighbors during the stage of Hessian approximation, DN-ADMM(\( K \)) is able to achieve significantly faster convergence speed. Note that, in all cases of DN-ADMM(\( K \)), we only communicate vectors exactly as first order methods while in DN-ADMM(0), no inner loop communication is required to compute the update direction once the gradient of the augmented Lagrangian is obtained.

VI. CONCLUSIONS

We have proposed a Newton variant of ADMM for distributed consensus optimization, that applies Newton steps in the primal update rule corresponding to the differentiable component of the objective. The method relies on the proximal mapping of the nonsmooth part of the cost function and approximates the Hessian inverse via truncation of Taylor’s series. Our analysis has established global linear convergence for the primal-dual iterates under the standard strong convexity and Lipschitz gradient assumptions without backtracking.
APPENDIX

A. Proof of Lemma 2

Recall the dual update in (17) and the fact that only the \((m+1)\)-th block of \(z_{t+1}\), i.e., \(\hat{z}_{t+1}^m\), is not identically zero. Therefore, it suffices to show that \(\{0, \hat{z}_{t+1}^\top\}\) is in the range of \(B^\top\). Note that since the underlying graph is connected, \(A^\top b = 0\) for any \(b \in \mathbb{R}^n\) such that all \(n\) blocks of sub-vectors of dimension \(d\) are equal. Construct such a vector \(b \in \mathbb{R}^n\) as: 
\[ b = \begin{bmatrix} \hat{z}_{t+1} & \cdots & \hat{z}_{t+1} \end{bmatrix}^\top. \]

From the definition of \(B\):
\[ B^\top b = \begin{bmatrix} A^\top \begin{bmatrix} b \\ (E')^\top b \end{bmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{z}_{t+1} \end{bmatrix} = \hat{z}_{t+1}, \quad (37) \]
which shows that \(\hat{z}_{t+1}\) is in the range of \(B^\top\). Therefore, with zero initialization, dual iterates generated by (17) stay in the range of \(B^\top\). Next we proceed to show that there exists a unique dual optimal \(y_t\) in the range of \(B^\top\). For any dual optimal \(y_t\) that satisfy the KKTa, \(\nabla F(x_t) + B y_t = 0\), its projection to the column space of \(B^\top\), denoted as \(y_{*}\), also satisfy KKTa. This is because their difference is in the kernel of \(B\), i.e., \(B(y_t - y_{*}) = 0\).

We show uniqueness of \(y_{*}\) by contradiction. Assume there are two dual optimal solutions, \(y_1 = B^\top b_1\) and \(y_2 = B^\top b_2\) and \(y_1 \neq y_2\). From KKTa (uniqueness of \(x_{*}\) is guaranteed by strong convexity of \(F(\cdot)\)), we obtain:
\[ \nabla F(x_t) + B B^\top b_1 = 0 \]
\[ \nabla F(x_t) + B B^\top b_2 = 0. \]

After taking the difference, we obtain \(BB^\top(b_1 - b_2) = 0\). Since \(BB^\top = LL + I_d + E'(E')^\top \succ 0\), we conclude \(b_1 = b_2\); contradiction.

B. Proof of Lemma 3

Rearranging the dual updates in (17) and focusing on the last block, we get:
\[ \hat{z}_{t+1}^m = x_{t+1}^m - \mu (g_{t+1}^m - y_{t+1}^m) \quad (38) \]
Moreover, since \(\hat{z}_{t+1} = \text{prox}_{\mu g}(x_{t+1}^m + \mu y_{t+1}^m) = \arg\min_{v} \{g(v) + \frac{1}{2\mu} \|x_{t+1}^m + \mu y_{t+1} - v\|_2^2\}\), the optimality condition is \(0 \in \partial g(\hat{z}_{t+1}) + \frac{1}{\mu} (\hat{z}_{t+1}^m - x_{t+1}^m - \mu y_{t+1}^m)\). Substituting the expression for \(x_{t+1}^m\) from (38) into above and canceling identical terms gives (20).

C. Proof of Lemma 4

Consider the primal updates for \(x_{t+1}\) in (17). After re-arranging, we obtain:
\[ \nabla F(x_t) + \frac{1}{\mu} B_B B^\top (x_t - \theta_{t+1}) + B y_t + \hat{H}_t(x_{t-1} - x_t) = 0. \]
Recalling the definition of the Hessian \(H_t = \nabla^2 F(x_t) + \frac{1}{\mu} B B^\top + \epsilon I\) in (8), add and subtract \(H_t(x_{t-1} - x_t)\) from the above gives,
\[ \nabla F(x_t) + \nabla^2 F(x_t)(x_{t-1} - x_t) + B y_t + \frac{1}{\mu} B B^\top x_{t+1} + (\hat{H}_t - H_t)(x_{t+1} - x_t) - \frac{1}{\mu} B y_t + \epsilon(x_{t+1} - x_t) = 0. \]

With the definition of error term (22), we rewrite the above as:
\[ \epsilon_t + \nabla F(x_{t+1}) + B y_t + \frac{1}{\mu} B B^\top x_{t+1} - \frac{1}{\mu} B z_{t+1} + \epsilon(x_{t+1} - x_t) = 0. \quad (39) \]

Substituting dual updates into above and subtract KKTa, we obtain (21).

D. Proof of Lemma 5
Recall the definition of error term in \([22]\) and apply triangle inequality to obtain:
\[
\|e_t\| \leq \left\| \nabla F(x_t) + \nabla^2 F(x_t)(x_{t+1} - x_t) - \nabla F(x_{t+1}) \right\|
\]
\[
+ \left\| (H_t - H_t)(x_{t+1} - x_t) \right\| + \frac{1}{\mu} \|B(z_{t+1} - \theta_{t+1})\|. \tag{40}
\]
We proceed to bound each component separately. From Assumption 1 and Lipschitz continuity of the gradient:
\[
\|\nabla F(x_t) + \nabla^2 F(x_t)(x_{t+1} - x_t) - \nabla F(x_{t+1})\| \leq 2M_f \|x_{t+1} - x_t\|. \tag{41}
\]
For the second term in \([40]\), we first rewrite \(H_t - H_t = H_t^{\frac{1}{2}}(I - H_t^{-\frac{1}{2}} H_t H_t^{-\frac{1}{2}}) H_t^{\frac{1}{2}}\). Therefore, we have:
\[
\|H_t - H_t\| \leq \left\| H_t^{\frac{1}{2}} \right\|^2 \left\| I - H_t^{-\frac{1}{2}} H_t H_t^{-\frac{1}{2}} \right\|. \tag{42}
\]
From properties of similar matrices, it follows that:
\[
\left\| I - H_t^{-\frac{1}{2}} H_t H_t^{-\frac{1}{2}} \right\| = \left\| I - H_t H_t^{-1} \right\|. \tag{43}
\]
Recall the matrix splitting \(H_t = D_t - N\) and the definition in \([15]\) (recall also that \(H_t\) is abbreviation for \(H_t(K)\)),
\[
H_t H_t^{-1} = (D_t - N)(D_t^{-\frac{1}{2}} \sum_{i=0}^{K} (D_t^{-\frac{1}{2}} N D_t^{-\frac{1}{2}})^i D_t^{-\frac{1}{2}})
\]
\[
= (D_t - N)(\sum_{i=0}^{K} D_t^{-1} (N D_t^{-1})^i).
\]
Therefore, we have:
\[
I - H_t H_t^{-1} = I - \sum_{i=0}^{K} (N D_t^{-1})^i + \sum_{i=0}^{K} (N D_t)^{i+1}
\]
\[
= (N D_t^{-1})^{K+1}. \tag{44}
\]
From \([14]\) we can bound this as:
\[
\left\| I - H_t H_t^{-1} \right\| = \left\| N D_t^{-1} \right\|^{K+1} = \left\| D_t^{-\frac{1}{2}} N D_t^{-\frac{1}{2}} \right\|^{K+1}
\]
\[
\leq \left( \frac{2(n-1)}{\mu(m_f + \epsilon) + 2(n-1)} \right)^{K+1}. \tag{45}
\]
Combining \([42]\), \([43]\) and \([45]\), we have:
\[
\|H_t - H_t\| \leq \left( \frac{2(n-1)}{\mu(m_f + \epsilon) + 2(n-1)} \right)^{K+1} \left\| H_t^{-\frac{1}{2}} \right\|^2. \tag{46}
\]
By definition, we know:
\[
H_t^{-\frac{1}{2}} = D_t^{-\frac{1}{2}} \sum_{i=0}^{K} (D_t^{-\frac{1}{2}} N D_t^{-\frac{1}{2}})^i D_t^{-\frac{1}{2}}
\]
\[
= D_t^{-\frac{1}{2}} + D_t^{-\frac{1}{2}} \sum_{i=1}^{K} (D_t^{-\frac{1}{2}} N D_t^{-\frac{1}{2}})^i D_t^{-\frac{1}{2}}. \tag{47}
\]
Since the second term in \([47]\) is positive semidefinite, we obtain \(\|H_t^{-\frac{1}{2}}\| \geq \|D_t^{-\frac{1}{2}}\|\) and using a term-by-term upper bound on the eigenvalues of \(D_t\) in \([9]\) we obtain:
\[
\|H_t^{-\frac{1}{2}}\| \geq \|D_t^{-\frac{1}{2}}\| \geq \frac{\mu(M_f + \epsilon) + 1 + 2(n-1)}{\mu(M_f + \epsilon) + [1 + 2(n-1)]}
\]
\[
\|H_t^{-\frac{1}{2}}\| \leq M_f + \epsilon + \frac{1+2(n-1)}{\mu}. \tag{48}
\]
Combining \([46]\) and \([48]\), we bound the second term of \([40]\) as:
\[
\|(H_t - H_t)(x_{t+1} - x_t)\| \leq (M_f + \epsilon + \frac{1+2(n-1)}{\mu})\|x_{t+1} - x_t\| \left( \frac{2(n-1)}{\mu(m_f + \epsilon) + 2(n-1)} \right)^{K+1}. \tag{49}
\]
The third term of \([40]\) can be upper bounded by considering the fact that \(\frac{1}{\mu} \|B(z_{t+1} - \theta_{t+1})\| = \frac{1}{\mu} \|\hat{z}_{t+1} - \hat{\theta}_{t+1}\| = \frac{1}{\mu} \|\text{prox}_{\mu g}(x_{t+1}^i + \mu y_{t+1}^{n+1}) - \text{prox}_{\mu g}(x_{t+1}^i + \mu y_{t}^{n+1})\|\). Since the proximal operator is nonexpansive, it follows that \(\frac{1}{\mu} \|\hat{z}_{t+1} - \hat{\theta}_{t+1}\| \leq \frac{1}{\mu} \|x_{t+1} - x_t\|.\) Combining these upper bounds, we arrive at the claim in \([23]\). \[\square\]