SIAM J. SCI. COMPUT.
Vol. 40, No. 6, pp. A4131–A4157

© 2018 Society for Industrial and Applied Mathematics

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

A SEMISMOOTH NEWTON METHOD FOR SEMIDEFINITE PROGRAMS AND ITS APPLICATIONS IN ELECTRONIC STRUCTURE CALCULATIONS[*]

YONGFENG LI[†], ZAIWEN WEN[†], CHAO YANG[‡], AND YA-XIANG YUAN[§]

Abstract. The well-known interior point method for semidefinite progams can only be used to tackle problems of relatively small scales. First-order methods such as the the alternating direction method of multipliers (ADMM) have much lower computational cost per iteration. However, their convergence can be slow, especially for obtaining highly accurate approximations. In this paper, we present a practical and efficient second-order semismooth Newton type method based on solving a fixed-point mapping derived from an equivalent form of the ADMM. We discuss a number of techniques that can be used to improve the computational efficiency of the method and achieve global convergence. Then we further consider the application in electronic structure calculations. The ground state energy of a many-electron system can be approximated by an variational approach in which the total energy of the system is minimized with respect to one- and two-body reduced density matrices instead of many-electron wavefunctions. This problem can be formulated as a semidefinite programming problem. Extensive numerical experiments show that our approach is competitive to the state-of-the-art methods in terms of both accuracy and speed.

Key words. semidefinite programming, ADMM, semismooth Newton method, electronic structure calculation, two-body reduced density matrix

AMS subject classifications. 15A18, 65F15, 47J10, 90C22

DOI. 10.1137/18M1188069

1. Introduction.

In this paper, we consider the following generic semidefinite programming (SDP) problem. Let $\mathcal{S}^n = \{X \in \mathbb{R}^{n\times n} \mid X^\top = X\}$. For two matrices $C, X \in \mathcal{S}^n$, the inner product between them is defined as $\langle C, X\rangle = \mathrm{tr}(CX)$. For a set of given matrices $A_1, \cdots, A_m \in \mathcal{S}^n$, we define a linear operator $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ by $\mathcal{A}X = (\langle A_1, X\rangle, \cdots, \langle A_m, X\rangle)^T$. The conjugate operator of $\mathcal{A}$ is defined by $\mathcal{A}^*y = \sum_{p=1}^m A_p y_p$ for $y \in \mathbb{R}^m$. Using these notations for a given $b \in \mathbb{R}^m$, we can formulate a primal SDP as

$$\begin{aligned} \max_{X\in\mathcal{S}^n} \quad & \langle C, X\rangle \\ \text{such that (s.t.)} \quad & \mathcal{A}X = b, \\ & X \succeq 0. \end{aligned} \tag{1.1}$$

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section May 17, 2018; accepted for publication (in revised form) September 10, 2018; published electronically December 18, 2018.

http://www.siam.org/journals/sisc/40-6/M118806.html

Funding: The second author's work was supported in part by NSFC grants 11831002, 11421101, and 91330202 and by the National Basic Research Project under grant 2015CB856002. The third author's work was supported by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (and Basic Energy Sciences) and by the Center for Applied Mathematics for Energy Research Applications (CAMERA) under award number DE-SC0008666. The fourth author's work was supported in part by NSFC grants 11331012 and 11461161005.

[†] Beijing International Center for Mathematical Research and Center for Data Science, Peking University, Beijing, China (YongfengLi@pku.edu.cn, wenzw@pku.edu.cn).

[‡] Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (cyang@lbl.gov).

[§] State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China (yyx@lsec.cc.ac.cn).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The corresponding dual SDP is

$$\begin{aligned} \min_{y \in \mathbb{R}^m, S \in \mathcal{S}^n} \quad & b^T y \\ \text{s.t.} \quad & S = \mathcal{A}^* y - C, \\ & S \succeq 0. \end{aligned} \tag{1.2}$$

We assume that m is significantly smaller than n^2.

Although the SDP problems are solvable in polynomial time by interior point methods [29, 30, 32], the computational cost is typically high due to the cost associated with assembling the Newton system and performing the Cholesky factorization to obtain search directions. First-order methods, which have much lower complexity per iteration, have gained wide acceptance in recent years. The well-known alternating direction multiplier method (ADMM) has been used to solve general SDPs in [31]. Although ADMM has relatively low complexity per iteration, it may converge slowly and take thousands or tens of thousands iterations to reach high accuracy. Recently, some new methods have been developed to speed up the solution of general SDPs. An example is the Newton-CG augmented Lagrangian method (SDPNAL) proposed in [35]. There are two loops in SDPNAL. The outer loop provides the augmented Lagrangian framework and the Lagrangian multipliers are updated at each iteration. In the inner loop, the semismooth Newton method is applied to minimize the augmented Lagrangian function up to certain accuracy. An enhanced version of SDPNAL called SDPNAL+ is developed in [34], which can further treat nonnegative SDP matrices.

1.1. Our contribution. In this paper, we first review the ADMM method since it serves as the foundation of the second-order method to be introduced below. Applying the ADMM to the dual SDP formulation is equivalent to applying the Douglas Rachford splitting (DRS) [11, 18, 13] method to the primal SDP formulation of the problem. The DRS method can be viewed as a fixed-point iteration that yields a solution of a system of semismooth and monotone nonlinear equations that coincides with the solution of the corresponding SDP. The generalized Jacobian of this system of nonlinear equations is positive semidefinite and bounded. Hence, our strategy is to use the semismooth Newton method to solve this system of nonlinear equations. Although an abstract form of the semismooth method has appeared in [33], the critical algorithmic framework for ensuring global convergence as well as several major components are significantly different. Our main contributions are as follows.

- A new scheme is proposed to ensure the global convergence of the semismooth Newton method. Compared with the methods shown in [33], the projection step is removed. Moreover, by exploring the special structure of the generalized Jacobian, we can compute the Newton step practically and efficiently.
- We also take advantage of the connection between the ADMM and DRS to modify the updates of the DRS and semismooth Newton method. The adjustments of the parameters in DRS and semismooth Newton method should be constructed carefully. Otherwise, the performance may not be satisfactory. A strategy for switching between ADMM and semismooth Newton steps is designed to combine the strengths of first-order and second-order methods and guarantee the global convergence.
- Our method solves a single system of nonlinear equations. It is different from SDPNAL [35] and SDPNAL+ [34] which minimizes a sequence of augmented

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Lagrangian functions for the dual SDP by a semismooth Newton-CG method. Based on the same key implementation details and subroutines of SDPNAL [35], SDPNAL+ [34] and ADMM+ [28], we demonstrate that our semismooth algorithm is competitive with SDPNAL and SDPNAL+ in terms of both computational time and accuracy on electronic structure calculation.

Another contribution is the application to electronic structure calculations. The molecular Schrödinger's equation, which is a many-body eigenvalue problem, is a fundamental problem to solve in quantum chemistry. An alternative way to approximate the ground state energy (i.e., the smallest eigenvalue), which does not involve approximating the many-body eigenfunction directly, is to express the ground state energy in terms of the so called one-body reduced density matrix (1-RDM) and two-body reduced density matrix (2-RDM) that satisfy a number of linear constraints. The process leads to a SDP and it is often referred to as the variational 2-RDM (v2-RDM) or 2-RDM method in short. In this paper, a detailed derivation of the SDP formulation is provided. We review how the ADMM method is used to solve the v2-RDM given in [36]. To improve the computational efficiency for solving v2-RDM problem, we exploit the special structures of matrices resulting from the 1-RDM and 2-RDM constraints. The block diagonal and low rank structures of these matrices are related to spin and spatial symmetry of the molecular orbitals [17, 24, 36]. We show how they can be used to significantly reduce the computational costs in the semismooth Newton method. Extensive numerical experiments on examples taken from [23] show that our semismooth Newton method can indeed achieve higher accuracy than the ADMM method. It can obtain accurate solutions similar to these reported in [23].

1.2. Organization.

The rest of this paper is organized as follows. In section 2, we review the DRS and the ADMM for solving general SDP problems. Our semismooth Newton method is presented in section 3. In section 4, we provide some background on electronic structure calculation, establish the notation, and introduce the v2-RDM formulation. Numerical results are reported in section 5. Finally, we conclude the paper in section 6.

2. The DRS and ADMM method.

We now discuss the usage of the DRS and the ADMM to solve the SDP formulations (1.1) and (1.2) since both of them will play important roles in the design of our hybrid approach in section 3.

The DRS method, first introduced to solve nonlinear partial differential equations [11, 18, 13], can be used to solve the primal SDP. To describe the DRS method, we first establish some notations and terminologies. Given a convex function f and a scalar $t > 0$, the proximal mapping of f is defined by

$$\mathbf{prox}_{tf}(X) := \arg\min_{U} f(U) + \frac{1}{2t}\|U - X\|_F^2, \tag{2.1}$$

where $\|\cdot\|_F$ is the Frobenius norm and $U, X \in \mathbb{R}^{n\times n}$. We also define an indicator function on a convex set Ω as

$$1_\Omega(X) := \begin{cases} 0 & \text{if } X \in \Omega, \\ +\infty, & \text{otherwise.} \end{cases}$$

To use the DRS method to solve (1.1), we let

$$f(X) = -\langle C, X\rangle + 1_{\{\mathcal{A}X=b\}}(X) \quad \text{and} \quad h(X) = 1_K(X), \tag{2.2}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $K = \{X : X \succeq 0\}$. Then each iteration of the DRS procedure for solving (1.1) can be described by the following sequences of steps:

$$\begin{aligned} X^{k+1} &= \mathbf{prox}_{th}(Z^k), \\ U^{k+1} &= \mathbf{prox}_{tf}(2X^{k+1} - Z^k), \\ Z^{k+1} &= Z^k + U^{k+1} - X^{k+1}, \end{aligned} \tag{2.3}$$

where $\{U^k\}$ and $\{Z^k\}$ are two sets of auxiliary variables. It follows from some simple algebraic rearrangements that the variables X and U can be eliminated in (2.3) to yield a fixed-point iteration of the form

$$Z^{k+1} = T_{\mathrm{DRS}}(Z^k), \tag{2.4}$$

where

$$T_{\mathrm{DRS}} := I + \mathbf{prox}_{tf} \circ (2\mathbf{prox}_{th} - I) - \mathbf{prox}_{th}. \tag{2.5}$$

The ADMM can be applied to the dual formulation of the SDP (1.2). Let X be the Lagrangian multiplier associated with the linear equality constraints of (1.2). The augmented Lagrangian function is

$$L_\sigma(y, S, X) = b^T y + \langle X, S - \mathcal{A}^* y + C\rangle + \frac{\sigma}{2}||S - \mathcal{A}^* y + C||_F^2. \tag{2.6}$$

Applying the ADMM [5, 31] with $\rho \in (0, \frac{1+\sqrt{5}}{2})$ to (1.2) yields the following sequence of steps in the kth iteration:

$$\begin{aligned} y^{k+1} &= \arg\min{}_y L_\sigma(y, S^k; X^k), \\ S^{k+1} &= \arg\min{}_{S\succeq 0} L_\sigma(y^{k+1}, S; X^k), \\ X^{k+1} &= X^k + \rho\sigma(S^{k+1} - \mathcal{A}^* y^{k+1} + C). \end{aligned} \tag{2.7}$$

We next make the following assumption.

Assumption 2.1. The operator $\mathcal{A}$ in (1.1) satisfies $\mathcal{A}\mathcal{A}^* = I$, and the Slater condition holds. That is, there exists $y \in \mathbb{R}^m$ and $S \succ 0$ such that $\mathcal{A}^* y - S = C$.

The first part of the assumption implies that $\mathcal{A}$ has full row rank. It is satisfied in many SDPs after a suitable transformation of $\mathcal{A}$. The convergence of DRS and ADMM has been studied in [15, 12, 2, 5, 31, 9, 28]. The following theorem provides the convergence properties of the ADMM and DRS.

THEOREM 2.2. *Suppose the Assumption* 2.1 *holds.*

(i) [5, Theorem 4.1] *If $\rho \in (0, \frac{1+\sqrt{5}}{2})$, then the sequence of variables (X^k, S^k, y^k) generated from the ADMM converge to a solution (X^*, S^*, y^*) of* (1.1)–(1.2) *from any starting point.*

(ii) *If $\rho = 1$, the ADMM iteration for the dual SDP* (1.2) *is equal to the DRS for the primal SDP* (1.1)*. The sequence of variables Z^k generated from the DRS converge to a fixed-point solution of $I - T_{\mathrm{DRS}}$ and the residual $\|Z^k - T_{\mathrm{DRS}}(Z^k)\|_F$ goes to* 0 [9, Theorem 1].

3. The semismooth Newton method.

Although the ADMM (and consequently the DRS method due to its equivalence to the ADMM) converges from any starting point on the SDP, the convergence can be slow, especially towards a highly

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

accurate approximation to the solution of the SDP. In practice, we often observe a rapid reduction in the objective function, infeasibility, and duality gap in the first few iterations. However, the reduction levels off after the first tens or hundreds of iterations. To accelerate convergence and obtain a more accurate approximation, we consider a second-order method.

The DRS can be characterized as a fixed-point iteration (2.4) for solving a system of nonlinear equations

$$F(Z) = \mathbf{prox}_{th}(Z) - \mathbf{prox}_{tf}(2\mathbf{prox}_{th}(Z) - Z) = 0, \tag{3.1}$$

where $Z \in \mathcal{S}^n$. Moreover, the solution of (3.1) is also an optimal solution to (1.1) and vice versa. Hence, we will focus on more efficient ways to solve (3.1).

3.1. Generalized Jacobian.

Before we discuss how to solve (3.1), let us first examine the structure of the generalized Jacobian of $F(Z)$. Using the definition of $f(x)$ and $h(x)$ given in (2.2), we can write down the explicit forms of $\mathbf{prox}_{tf}(Y)$ and $\mathbf{prox}_{th}(Z)$ as

$$\begin{aligned} \mathbf{prox}_{tf}(Y) &= (Y + tC) - \mathcal{A}^*(\mathcal{A}Y + t\mathcal{A}C - b), \\ \mathbf{prox}_{th}(Z) &= Q_\alpha \Sigma_\alpha Q_\alpha^T, \end{aligned}$$

where

$$Q\Sigma Q^T = \begin{pmatrix} Q_\alpha & Q_{\bar\alpha} \end{pmatrix} \begin{pmatrix} \Sigma_\alpha & 0 \\ 0 & \Sigma_{\bar\alpha} \end{pmatrix} \begin{pmatrix} Q_\alpha^T \\ Q_{\bar\alpha}^T \end{pmatrix}$$

is the spectral decomposition of the matrix Z with $\Sigma = \operatorname{diag}(\lambda_1, \cdots, \lambda_n)$, $\alpha = \{i | \lambda_i > 0\}$, and $\bar\alpha = \{1, \ldots, n\} \setminus \alpha$ is the set of the indices of the positive and nonpositive eigenvalues of Z.

Since F is locally Lipschitz continuous, it can be verified that F is almost differentiable everywhere. We next introduce the concepts of generalized subdifferential.

DEFINITION 3.1. *Let F be locally Lipschitz continuous at $X \in \mathcal{O}$, where $\mathcal{O}$ is an open set. Let D_F be the set of differentiable points of F in $\mathcal{O}$. The* B-subdifferential *of F at X is defined by*

$$\partial_B F(X) := \left\{ \lim_{k\to\infty} F'(X^k) | X^k \in D_F, X^k \to X \right\}.$$

The set $\partial F(x) = \operatorname{co}(\partial_B F(x))$ is called Clarke's generalized Jacobian*, where* co *denotes the convex hull.*

Let $\mathcal{I}$ be an identity operator. It can be shown that

$$\mathcal{D} = \mathcal{I} - \mathcal{A}^*\mathcal{A} \tag{3.2}$$

is the Jacobian matrix $\partial\mathbf{prox}_{tf}(\cdot)$ at the point $(2\mathbf{prox}_{th}(Z) - Z)$ associated with the second term of $F(Z)$ in (3.1). Similar to the convention used in [35], we define a generalized Jacobian operator $\mathcal{M}(Z) \in \partial\mathbf{prox}_{th}(Z)$ in terms of its application to an $n \times n$ matrix S that yields

$$\mathcal{M}(Z)[S] = Q(\Omega \circ (Q^T S Q))Q^T \text{ for all } S \succeq 0, \tag{3.3}$$

where the $\circ$ symbol denotes a Hadamard product and

$$\Omega = \begin{bmatrix} E_{\alpha\alpha} & k_{\alpha\bar\alpha} \\ k_{\alpha\bar\alpha}^T & 0 \end{bmatrix}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

with $E_{\alpha\alpha}$ being a matrix of ones and $k_{ij} = \frac{\lambda_i}{\lambda_i - \lambda_j}, i \in \alpha, j \in \bar{\alpha}$. Usually, it is nontrival to describe the generalized Jacobian $\partial F(Z)$ exactly. We can define an alternative form

$$\hat{\partial} F(Z) = \partial \mathbf{prox}_{th}(Z) + \mathcal{D}(I - 2\partial \mathbf{prox}_{th}(Z)) \tag{3.4}$$

and choose an element $\mathcal{J}(Z) \in \hat{\partial} F(Z)$:

$$\mathcal{J}(Z) = \mathcal{M}(Z) + \mathcal{D}(I - 2\mathcal{M}(Z)). \tag{3.5}$$

It follows from [6, Corollory, page 75] that

$$\hat{\partial} F(Z)[S] = \partial F(Z)[S] \text{ for all } S \succeq 0. \tag{3.6}$$

We now introduce the definition of the semismoothness and monotoneness below.

Definition 3.2. *Let F be a locally Lipschitz continuous function in a domain $\mathcal{O}$. We say that F is semismooth at $x \in \mathcal{O}$ if* (i) *F is directionally differentiable at x;* (ii) *for any $z \in \mathcal{O}$ and $\mathcal{J} \in \partial F(x+z)$,*

$$\|F(x+z) - F(x) - \mathcal{J}[z]\|_2 = o(\|z\|_2) \quad \text{as } z \to 0. \tag{3.7}$$

The function F is said to be strongly semismooth if $o(\|z\|_2)$ in (3.7) *is replaced by $O(\|z\|_2^2)$. It is called monotone if $\langle x - y, F(x) - F(y)\rangle \geq 0$ for all $x, y \in \mathbb{R}^n$.*

The next lemma characterizes the fixed-point map given in (3.1) and its generalized Jacobian matrix.

Lemma 3.3. *The function F in* (3.1) *is strongly semismooth and monotone. Each element of Clarke's generalized Jacobian $\partial F(x)$ of F is positive semidefinite.*

Proof. The strongly semismoothness of F follows from the derivation given in [26, 27] to establish the semismoothness of proximal mappings. In fact, the projection over a polyhedral set is strongly semismooth [26, Example 12.31], and the projections over symmetric cones are proved to be strongly semismooth in [27]. Hence, $\mathbf{prox}_{tf}(\cdot)$ and $\mathbf{prox}_{th}(\cdot)$ are strongly semismooth. Since strongly semismoothness is closed under scalar multiplication, summation, and composition, the function F is strongly semismooth.

It has been shown in [18] that the operator T_{DRS} is firmly nonexpansive. Therefore, F is firmly nonexpansive, hence monotone [1, Proposition 4.2]. The positive semidefiniteness simply follows from Lemma 3.5 in [33]. □

3.2. Computing the Newton direction.

Using the expression given in (3.5), we can now discuss how to compute the Newton direction efficiently. At a given iterate Z^k, we compute a Newton direction S^k by solving the equation

$$(\mathcal{J}_k + \mu_k \mathcal{I})[S^k] = -F^k, \tag{3.8}$$

where $\mathcal{J}_k \in \hat{\partial} F(Z^k)$, $F^k = F(Z^k)$. The regularized parameter is defined as

$$\mu_k = \kappa_k \|F^k\|_2 \tag{3.9}$$

with $\kappa_k > 0$. Equation (3.8) is well-defined since each element of B-subdifferential $\partial_B F(x)$ of F is positive semidefinite and the regularization term $\mu_k I$ is chosen such that $\mathcal{J}_k + \mu_k \mathcal{I}$ is invertible. From a computational view, it is not practical to solve

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

the linear system (3.8) exactly. Therefore, we seek an approximate step S^k by solving (3.8) approximately so that

$$\|r^k\|_F \leq \tau \min\{1, \kappa \|F^k\|_F \|S^k\|_F\}, \tag{3.10}$$

where

$$r^k := (\mathcal{J}_k + \mu_k \mathcal{I})[S^k] + F^k \tag{3.11}$$

is the residual and $0 < \tau < 1$ is some positive constant.

Since $\mathcal{J}_k$ in (3.8) is nonsymmetric and its dimension is large, we apply the Sherman–Morrison–Woodbury (SMW) formula to transform (3.8) into a smaller symmetric system. If we vectorize the matrix S, then the operators $\mathcal{M}(Z)$ and $\mathcal{D}$ can be expressed as matrices

$$M(Z) = \tilde{Q}\Lambda\tilde{Q}^T \quad \text{and} \quad D = I - A^T A,$$

respectively, where $\tilde{Q} = Q \otimes Q$, $\Lambda = \operatorname{diag}(\operatorname{vec}(\Omega))$, I is the identity matrix and A is the matrix form of $\mathcal{A}$. Let $W = I - 2M(Z) = \tilde{Q}(I - 2\Lambda)\tilde{Q}^T$ and $H = \tilde{Q}((\mu_k + 1)I - \Lambda)\tilde{Q}^T$. Then the matrix form of $\mathcal{J}_k + \mu_k I$ can be written as $J_k + \mu_k I = H - A^T A W$. It follows from the SMW formula that

$$\begin{aligned}(J_k + \mu_k I)^{-1} &= (H - A^T A W)^{-1} \\ &= H^{-1} + H^{-1}A^T(I - AWH^{-1}A^T)^{-1}AWH^{-1}.\end{aligned}$$

Define

$$T = \tilde{Q} L \tilde{Q}^T, \tag{3.12}$$

where L is a diagonal matrix with diagonal entries $L_{ii} = \frac{\Lambda_{ii}\mu_k}{\mu_k + 1 - \Lambda_{ii}}$, where Λ_{ii} is the ith diagonal entry of Λ. By using the identities $H^{-1} = \frac{1}{\mu_k+1} I + \frac{1}{\mu_k(\mu_k+1)} T$ and $WH^{-1} = \frac{1}{1+\mu_k} I - (\frac{1}{\mu_k} + \frac{1}{\mu_k+1})T$, we can further obtain

$$\begin{aligned}&(J_k + \mu_k I)^{-1} \\ &= \frac{\mu_k I + T}{\mu_k(\mu_k+1)} \left(I + A^T \left(\frac{\mu_k^2}{2\mu_k + 1} I + ATA^T \right)^{-1} A \left(\frac{\mu_k}{2\mu_k+1} I - T \right) \right).\end{aligned} \tag{3.13}$$

As a result, the solution of (3.8) can be obtained by first solving the following symmetric linear equation

$$\left(\frac{\mu_k^2}{2\mu_k + 1} I + ATA^T \right) d_s = a, \tag{3.14}$$

where $a = -A(\frac{\mu_k}{2\mu_k+1} I - T)\operatorname{vec}(F^k)$ and vec is the vectorized operator that transforms a matrix to a verctor, by an iterative method such as the CG method or the symmetric QMR method. Note that the size of the coefficient matrix of (3.14) is $m \times m$ while that of (3.8) is $n^2 \times n^2$, where m usually is much smaller than n^2. Then we use the following expression to recover

$$S^k = \frac{1}{\mu_k(\mu_k+1)}(\mu_k \mathcal{I} + \mathcal{T})[-F^k + \mathcal{A}^* d_s],$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 1. Solving the linear system (3.8).

```
Compute $a = -\mathcal{A}(\frac{\mu_k}{2\mu_k+1}\mathcal{I} - \mathcal{T})F^k$ ;
Use the CG or symmetric QMR method to solve $(\frac{\mu_k^2}{2\mu_k+1}\mathcal{I} + \mathcal{A}\mathcal{T}\mathcal{A}^*)d_s = a$
    inexactly, where the matrix-vector multiplication is computed by (3.15) ;
Compute the Newton direction $S^k = \frac{1}{\mu_k(\mu_k+1)}(\mu_k\mathcal{I} + \mathcal{T})(-F^k + \mathcal{A}^* d_s)$.
```

where $\mathcal{T}$ is the operator form of T in (3.12). Specifically, applying $\mathcal{T}$ to a matrix S yields

$$\mathcal{T}(Z)[S] = Q(\Omega_0 \circ (Q^T SQ))Q^T \quad \forall\ S \succeq 0,$$

where

$$\Omega_0 = \begin{bmatrix} E_{\alpha\alpha} & l_{\alpha\bar{\alpha}} \\ l_{\alpha\bar{\alpha}}^T & 0 \end{bmatrix} \text{ and } l_{ij} = \frac{\mu_k k_{ij}}{\mu_k + 1 - k_{ij}}.$$

Let $\Upsilon = \mathcal{T}(Z)[S]$. We can then use the same techniques used in [35] to express Υ as multiplication:

$$\Upsilon = [Q_\alpha Q_{\bar{\alpha}}] \begin{bmatrix} Q_\alpha^T SQ_\alpha & l_{\alpha\bar{\alpha}} \circ Q_\alpha^T SQ_{\bar{\alpha}} \\ l_{\alpha\bar{\alpha}}^T \circ Q_{\bar{\alpha}}^T SQ_\alpha & 0 \end{bmatrix} \begin{bmatrix} Q_\alpha^T \\ Q_{\bar{\alpha}}^T \end{bmatrix} = G + G^T, \tag{3.15}$$

where $G = Q_\alpha(\frac{1}{2}(UQ_\alpha^T) + l_{\alpha\bar{\alpha}} \circ (UQ_{\bar{\alpha}}))$ with $U = Q_\alpha^T S$. The number of floating point operations (flops) required to compute Υ is $8|\alpha|n^2$. If $|\alpha|$ is large, we can compute Υ via the equivalent expression $\Upsilon = S - Q((E - \Omega_0) \circ (Q^T SQ))Q^T$, which requires $8|\bar{\alpha}|n^2$ flops.

Therefore, using the expression (3.15) allows us to obtain an approximate solution to (3.8) efficiently whenever $|\alpha|$ or $|\bar{\alpha}|$ is small. We summarize the procedure for solving the Newton equation (3.8) approximately in Algorithm 1.

3.3. Switching between ADMM and semismooth Newton steps.

A few safeguard strategies are developed in order to stabilize the semismooth Newton step and maintain global convergence. Let $U^k = Z^k + S^k$ be a new trial point from the Newton step. Choose $0 < \nu < 1$ and a fixed integer $\zeta > 0$. If the residual $\|F(U^k)\|_F$ is sufficiently decreased with respect to the last few steps, i.e.,

$$\|F(U^k)\|_F \le \nu \max_{\max(1,k-\zeta+1)\le j\le k} \|F(Z^j)\|_F, \tag{3.16}$$

then we execute a Newton step, i.e., $Z^{k+1} = U^k$. Otherwise, we let $Z^{k+1} = Z^k$ and mark it as a failed step. When the number N_f of failed steps reaches a fixed count $\bar{N}_f$, we go to perform the ADMM iterations.

The parameter κ_k in (3.9) is an important parameter to control the quality of S^k. When κ_k is large, S_k is close to the DRS direction $F(Z^k)$ and usually leads to a reduction of $\|F(U^k)\|_F$. However, it may also lead to a slow convergent rate. When κ_k is small, the convergence may be fast, but S_k may be a bad direction. We define the ratio

$$\rho_k = \frac{-\left\langle F(U^k), S^k \right\rangle}{\|S^k\|_F^2} \tag{3.17}$$

to decide how to update κ_k. If ρ_k is small, it is usually a signal of a bad Newton step. Then we increase κ_k. Otherwise, we decrease it.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

In summary, we set

$$Z^{k+1} = \begin{cases} U^k & \text{if } U_k \text{ satisfies (3.16) [Newton step]}, \\ Z^k, & \text{otherwise [failed step]}. \end{cases} \tag{3.18}$$

Then the parameters κ_{k+1} are updated as

$$\kappa_{k+1} = \begin{cases} \max\{\underline{\kappa}, \kappa_0 \kappa_k\} & \text{if } \rho_k \geq \eta_2, \\ \gamma_1 \kappa_k & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ \min\{\bar{\kappa}, \gamma_2 \kappa_k\}, & \text{otherwise}, \end{cases} \tag{3.19}$$

where $\gamma_0 < 1$, γ_1, $\gamma_2 > 1$ are chosen parameters and $\underline{\kappa}$, $\bar{\kappa}$ are two positive constants.

We next show our strategies for incorporating the ADMM and semismooth Newton steps. The basic idea is to combine the advantages of first-order and second-order methods. Note that we choose the ADMM rather than the DRS as our first-order method since the primal and dual variables are explicilt available during the ADMM iterations and some strategies in ADMM can be used to accelerate the convergent speed, such as choosing $\rho = 1.618$. When the ADMM method converges slowly, we switch to the semismooth Newton step. We mainly examine the reduced ratios of primal and dual infeasibilities of the last few steps defined by

$$\omega^k_{\eta_p} = \frac{\text{mean}_{k-5 \leq j \leq k} \eta^j_p}{\text{mean}_{k-25 \leq j \leq k-20} \eta^j_p} \text{ and } \omega^k_{\eta_q} = \frac{\text{mean}_{k-5 \leq j \leq k} \eta^j_q}{\text{mean}_{k-25 \leq j \leq k-20} \eta^j_q} \tag{3.20}$$

where the primal infeasibility η_p and the dual infeasibility η_d are defined by

$$\eta_p = \frac{\|\mathcal{A}(X) - b\|_2}{1 + \|b\|_2} \quad \text{and} \quad \eta_d = \frac{\|\mathcal{A}^* y - C - S\|_F}{1 + \|C\|_F}. \tag{3.21}$$

If these reduced ratios are larger than some given constants, we say that the ADMM performs bad and go to the semismooth Newton steps. Of course, the parameters $5, 20, 25$ in (3.20) can be tuned to be other values as well. Similarly, we go from the semismooth Newton steps to the ADMM steps if the reduced ratios are too large, and if the number of iterations of solving the system of linear equations (3.14) is large but there is not much progress.

In practice, the penalty parameter σ of the ADMM is often updated adaptively to achieve faster convergence. One strategy is to tune σ to balance the primal infeasibility η_p and the dual infeasibility η_d. In particular, the η^j_p and η^j_d denote the primal and dual infeasibilities at the jth iteration. If the mean of $\{\eta^j_p / \eta^j_q\}$ in a few steps is larger (or smaller) than a constant δ, we decrease (or increase) the penalty parameter σ by a multiplicative factor γ (or $1/\gamma$) with $0 < \gamma < 1$. To prevent σ from becoming excessively large or small, an upper and lower bound are often imposed on σ. This strategy has been demonstrated to be effective in [31].

Due to the switching between the ADMM and the semismooth Newton steps based on the DRS, we next describe the recovery of the variables of the DRS (also semismooth Newton steps) from these of the ADMM with $\rho = 1$ and vice versa. In fact, the X variable produced in the kth step of DRS applied to (1.1) is exactly the X variable produced in the kth step of ADMM applied to (1.2). The other variables (Z and U) and the parameter t produced in DRS are related to the variables y, S and parameter σ produced in the ADMM via

$$\begin{cases} t = \sigma; \\ Z^k = X^{k+1} - \sigma S^{k+1}. \end{cases} \tag{3.22}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

If the DRS (2.3) is first executed, we can obtain the following relationship for the ADMM as

$$\begin{cases} \sigma = t; \\ X^k = \mathbf{prox}_{th}(Z^{k-1}); \\ S^k = \dfrac{1}{t}(X^k - Z^{k-1}); \\ \mathcal{A}^* y^k = \dfrac{1}{t}(X^k - X^{k-1}) - S^k + C. \end{cases} \tag{3.23}$$

The variable y^k can be further computed from the last equation if the operator $\mathcal{A}$ is of full row rank. Consequently, the strategies of the ADMM for updating σ can be used in the DRS and Newton step for modifying t and vice versa. However, one should be careful on computing the primal and dual infeasibilities of the DRS when the parameter t is changed from t_1 to t_2 after one loop of the DRS (2.3). In this case, the next immediate update of the DRS should be

$$\begin{aligned} X^{k+1} &= \mathbf{prox}_{t_1 h}\left(Z^k\right), \\ U^{k+1} &= \mathbf{prox}_{t_2 f}\left(X^{k+1} - \frac{t_2}{t_1}(Z^k - X^{k+1})\right), \\ Z^{k+1} &= \frac{t_2}{t_1}(Z^k - X^{k+1}) + U^{k+1}. \end{aligned} \tag{3.24}$$

Thereafter, the original iterations (2.3) can still be used for the fixed t_2.

The complete approach to solve the SDP is summarized in Algorithm 2.

The following theorem establishes the global convergence of Algorithm 2.

Algorithm 2. A semismooth Newton method for SDP (SSNSDP).

```
Give $\zeta, \bar{N}_f > 0$, $0 < \tau, \nu < 1$, $0 < \eta_1 \le \eta_2 < 1$ and $\gamma_0 < 1$, $\gamma_1$, $\gamma_2 > 1$ ;
Choose $Z^0$ and $\varepsilon > 0$. Set $k = 0$, $N_f = 0$ and doSSN = 0;
while not "converged" do
    if doSSN == 1 then
        Select $J_k \in \hat{\partial} F(Z^k)$;
        Solve the linear system (3.8) approximately such that $S^k$ satisfies
          (3.10);
        Compute $U^k = Z^k + S^k$ ;
        Update $Z^{k+1}$ and $\kappa_{k+1}$ according to (3.18) and (3.19), respectively;
        If (3.16) is not satisfied, set $N_f = N_f + 1$;
        if $N_f \ge \bar{N}_f$ or the Newton step performs bad then
            Set doSSN = 0; Set the parameters for the ADMM steps;
    if doSSN == 0 then
        Perform an ADMM step. Equivalently, it defines $Z^{k+1} = Z^k - F(Z^k)$;
        if the ADMM step performs bad then
            Set doSSN = 1 and $N_f = 0$; Set the parameters of the Newton
              steps;
    Update the penalty parameter $t$;
    Set $k = k + 1$;
```

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 3.4. *Suppose that Assumption* 2.1 *holds and* $\{Z^k\}$ *is a sequence generated by Algorithm* 2 *with a fixed penalty parameter* σ *and* $\rho = 1$ *in the ADMM and a fixed parameter* t *in* (3.1). *Then the residuals of* $\{Z^k\}$ *converge to* 0*, i.e.,* $\lim_{k\to\infty} ||F(Z^k)||_F = 0$.

Proof. For notational convenience, a consecutive ADMM (semismooth Newton) steps between two Newton (ADMM) steps is called an ADMM (Newton) epoch. Then in Algorithm 2, ADMM epochs and Newton epochs perform alternatively. If all steps in a Newton epoch are failed steps, this epoch can be deleted directly. Without loss of generality, we can assume that there exists at least one successful Newton step in each Newton epoch.

If the number of Newton steps is finite, then the whole sequence is generated from the ADMM after some iterations. Hence, the convergence holds due to the convergence of the ADMM. Therefore, we only need to consider the case that the number of Newton steps is infinite. Since the number of failed Newton steps is less than $\bar{N}_f$ and the number of successful Newton steps is at least one in each Newton epoch, there must exists infinite successful Newton steps.

Firstly, we assert that the residual are nonincreasing for all steps in ADMM epochs and all failed steps in Newton epochs, i.e., $\|F(Z^{k+1})\|_F \le \|F(Z^k)\|_F$. This fact is trivial for failed steps, and it follows from the firmly nonexpansiveness of T_{DRS} [18] and Theorem 1 in [8] for the ADMM steps. Define

$$\bar{F}^k = \max_{k-\zeta+1\le j\le k} \|F(Z^j)\|_F \text{ for all } k \ge \zeta.$$

For the failed steps and ADMM steps, we have $\|F(Z^{k+1})\|_F \le \|F(Z^k)\|_F \le \bar{F}^k$. For successful Newton steps in the Newton epochs, it holds $\|F(Z^{k+1})\|_F \le v\bar{F}^k$. Thus, we prove that $\bar{F}^k$ is nondecreasing since

$$\bar{F}^{k+1} \le \max\{\|F(Z^{k+1})\|_F, \bar{F}_k\} \le \max\{v, 1\}\bar{F}^k = \bar{F}^{k+1}.$$

Finally, we prove the fact that $\bar{F}^{k+\zeta} \le v\bar{F}^k$ if a successful Newton step is performed at the kth step. We only need to show that

$$\|F(Z^{k+j})\|_F \le v\bar{F}^k \text{ for all } 1 \le j \le \zeta. \tag{3.25}$$

When $j = 1$, the assertion (3.25) holds by the conditions of successful Newton steps. When $j = 2$, we consider the following two cases. If a ADMM step or failed step is performed at the $(k+1)$th step, then

$$\|F(Z^{k+2})\|_F \le \|F(Z^{k+1})\|_F \le v\bar{F}^k.$$

If a success Newton step is executed at $(k+1)$th step, then

$$\|F(Z^{k+2})\|_F \le v\bar{F}^{k+1} \le v\bar{F}^k,$$

where the last inequality is due to the nonincreasingness of $\bar{F}^k$. This concludes that the claim (3.25) holds for $j = 2$. Continuing this argument for all $2 < j \le \zeta$, the assertion (3.25) holds. Combining with infiniteness of the successful Newton steps and the nonincreasingness of $\bar{F}^k$, we have that $\lim_{k\to\infty} \bar{F}^k = 0$, which impiles $\lim_{k\to\infty} \|F(Z^k)\|_F = 0$. □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.4. Comparisons with SDPNAL and SDPNAL+. In this subsection, we compare our algorithm with the semismooth Newton type methods SDPNAL [35] and SDPNAL+ [34]. Starting from X^0, the augmented Lagrangian method solves the dual problem (1.2) by

$$\begin{aligned} (y^{k+1}, S^{k+1}) &= \arg \min_{S\succeq 0, y\in\mathbb{R}^m} L_\sigma(y, S, X^k), \\ X^{k+1} &= X^k + \sigma(S^{k+1} - \mathcal{A}^* y^{k+1} + C), \end{aligned} \tag{3.26}$$

where $L_\sigma(y, S, X)$ is defined in (2.6). In SDPNAL, the variable S is eliminated due to $S^{k+1} = \Pi_{\mathcal{S}^n_+}(\mathcal{A}^* y^{k+1} - C - X^k/\sigma)$, where $\Pi_{\mathcal{S}^n_+}$ is the projection on semidefinite matrix cone. Consequently, SDPNAL solves an equivalent form of (3.26) as

$$y^{k+1} = \arg\min \tilde{L}_{\sigma^k}(y, X^k), \tag{3.27}$$

$$X^{k+1} = \Pi_{\mathcal{S}^n_+}(X^k - \sigma(\mathcal{A}^* y^{k+1} - C)), \tag{3.28}$$

where

$$\tilde{L}_\sigma(y, X) = b^T y + \frac{1}{2\sigma}(||\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C))||_F^2 - ||X||_F^2). \tag{3.29}$$

Then the subproblem (3.27) is minimized by using a semismooth Newton method to certain accuracy. The gradient and an alternative element of the generalized Hessian of (3.29) with respect to y is

$$\nabla_y \tilde{L}_\sigma(y, X) = b - \mathcal{A}\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C)), \tag{3.30}$$

$$V \in \sigma\mathcal{A}\partial\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C))\mathcal{A}^*. \tag{3.31}$$

For fixed y and X, the corresponding semismooth Newton step is

$$(V + \epsilon I)d = \nabla_y L_\sigma(y, X), \tag{3.32}$$

where ϵ is a small constant. The detailed computation in [35] shows that the left hand sides of both (3.32) and (3.14) share certain similar structures but they are not the same due to certain subtle differences. Their right hand sides are also different. Hence, the two systems usually lead to different search directions. The linear system of SDPNAL corresponds to the dual variable y, while ours relates to the variable Z. These two linear systems are connected through the SMW formula. Therefore, the core iteration processes of SDPNAL and SSNSDP are indeed different.

SDPNAL+ [34] is a much enhanced version of SDPNAL for SDPs with nonnegative constraints. For standard SDPs, the Newton systems of SDPNAL and SDPNAL+ are essentially the same but some specific implementation details are different. We should point out that switching rules between ADMM and Newton steps are also implemented in SDPNAL+ and our rules are inspired by them. However, the most recent rules of SDPNAL+ [34] are unknown since the source codes are not publicly available.

4. Electronic structure calculations.

We now discuss how the algorithm presented above can be used to approximate ground state energy of a molecular system.

4.1. The variational 2-RDM formulation.

The electronic structure of a molecule can be determined by the solution to an N-electron Schrödinger equation

$$H\Psi = E\Psi, \tag{4.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $\Psi : \mathbb{R}^{3N} \otimes \{\pm\frac{1}{2}\}^{3N} \to \mathbb{C}$ is an N-electron antisymmetric wave function that obeys the Pauli exclusion principle, E represent the total energy of the N-electron system, and H is the molecular Hamiltonian operator defined by

$$H = \underbrace{\sum_{i=1}^{N} -\frac{1}{2}\triangle_i - \sum_{i=1}^{N}\sum_{k=1}^{K}\frac{Z_k}{|R_k - r_i|}}_{\text{one-body term}} + \underbrace{\frac{1}{2}\sum_{i,j=1,i\neq j}^{N}\frac{1}{|r_i - r_j|}}_{\text{two-body term}}. \tag{4.2}$$

Here $\triangle_i$ denotes a Laplace operator with respect to the spatial coordinate of the ith electrons, $R_k, k = 1, \ldots, K$, gives the coordinates of the kth nuclei with charge Z_k, and $r_i, i = 1, \ldots, N$, gives the coordinates of the ith electron. The smallest eigenvalue of H, often denoted by E_0, is called the ground state energy (4.1).

It is now well known that an alternative expression of the total energy of the molecular system is

$$E = \sum_{i,j}^{d} T_{i,j}\gamma_{i,j} + \sum_{i,j,k,l=1}^{d} V_{ij,kl}\Gamma_{ij,kl}, \tag{4.3}$$

where $\gamma_{i,j}$ and $\Gamma_{ij,kl}$ are elements of the so-called 1-RDM γ and 2-RDM Γ, respectively. They are the unknowns to be determined. The matrix elements $T_{i,j}$ and $V_{ij,kl}$ are one- and two-body integrals of molecular orbitals associated with the one-body and two-body terms in (4.2).

This alternative formulation of the many-body problem and the method developed to solve the reformuated problem is known as the 2-RDM method. The development of the 2-RDM method dates back to 1950s. Mayer [20] showed how the energy of a many-body problem can be represented in terms of 1-RDM and 2-RDM, which can be writen as a matrix and a 4-order tensor. However, since not all matrices or tensors are RDMs associated with an N-electron wavefunction, one must add some constraints to guarantee that the matrices and tensors satisfy the so called *N-representability condition*, which was first proposed by Coleman [7] in 1963 and has been investigated for nearly 50 years. The N-representability condition for the 1-RDM in the variational problem has been solved in [7]. In 1964, Garrod and Percus [16] showed a sufficient and necessary condition for the 2-RDM N-representability problem. It is theoretically meaningful but computationally intractable. In 2007, Liu, Christandl, and Verstraete showed that the N-representability problem of 2-RDM is QMA-complete [19]. Since then a number of approximation conditions, including the P, Q, R, T1, T2, T2' conditions, have been proposed in [7, 16, 14, 36, 21, 3]. All these conditions are formulated by keeping matrices whose elements are linear combinations of the components of the 1-RDM and 2-RDM matrices positive semidefinite. As a result, the constrained minimization of the total energy with respect to 1-RDM and 2-RDM becomes an SDP.

The practical use of the v2-RDM approach to solving the ground state electronic structure is enabled, to some extent, by the recent advances in numerical methods for solving large-scale SDPs. In [24], Nakata et al. solved the v2-RDM problem by an interior point method. Zhao et al. reformulated the 2-RDM using the dual SDP formalism and also applied the interior point method in [36]. The problem size of the SDP formulation in [36] is usually smaller than the ones given in [24]. The boundary point method (similar to the ADMM method) is developed by Mazziotti to solve the v2-RDM in [22]. Rigorous error bounds for approximate solutions obtained from the v2-RDM approach are discussed in [4].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Note that the dimensions of γ and Γ are $d \times d$, and $d^2 \times d^2$, respectively, where d is proportional to the number of electrons N. By treating the total energy E as a function of γ and Γ, we can obtain an approximation to the ground state energy by solving an optimization problem with $O(N^4)$ variables instead of an eigenvalue problem of a dimension that grows exponentially with respect to N.

However, γ and Γ are not arbitrary matrices. They are said to be N-representible if they can be obtained from some many-body wavefunction Ψ. N-representible matrices are known to have a number of properties [21, 36] that can be used to constrain the set of matrices over which the objective function (4.3) is minimized. These properties include

$$\gamma_{i,j} = \gamma_{j,i}, \Gamma_{ij,kl} = \Gamma_{kl,ij}; \qquad \text{Hermitian} \tag{4.4}$$

$$\Gamma_{ij,kl} = -\Gamma_{ji,kl} = -\Gamma_{ij,lk}; \qquad \text{antisymmetric} \tag{4.5}$$

$$\operatorname{tr}(\gamma) = N \;\text{ and }\; \operatorname{tr}(\Gamma) = \frac{N(N-1)}{2}; \qquad \text{trace} \tag{4.6}$$

$$\sum_k \Gamma_{ik,jk} = \frac{N-1}{2}\gamma_{ij}. \qquad \text{partial trace} \tag{4.7}$$

However, the above conditions are not sufficient to guarantee γ and Γ to be N-representible. A significant amount of effort has been devoted in the last few decades to develop additional conditions that further constrain γ and Γ to be N-representible [21, 36] without making use of Ψ explicitly. These conditions are collectively called the *N-representability conditions.*

4.2. *N*-representability conditions. The N-representability conditions were first introduced in [7]. It has been shown in [7] that γ is N-representable if and only if $0 \preceq \gamma \preceq I$. For 2-RDM, it is more difficult to write down a complete set of the conditions under which Γ is N-representable. Liu, Christandl, and Verstraete showed that the N-representability problem is QMA-complete in [19]. There has been efforts to derive approximation conditions that are useful in practice. The well known approximation conditions in [7, 16, 14, 36, 21, 3] define the so-called $P, Q, G, T1, T2$ variables whose elements can be expressed as a linear function with respect to the elements of γ and Γ as follows:

$$P_{ij,i'j'} = \Gamma_{ij,i'j'}, \tag{4.8}$$

$$\begin{aligned} Q_{ij,i'j'} = {} & (\delta_{ii'}\delta_{jj'} - \delta_{ij'}\delta_{ji'}) - (\delta_{ii'}\gamma_{jj'} + \delta_{jj'}\gamma_{ii'}) \\ & +(\delta_{ij'}\gamma_{ji'} + \delta_{ji'}\gamma_{ij'}) + \Gamma_{ij,i'j'}, \end{aligned} \tag{4.9}$$

$$G_{ij,i'j'} = \delta_{jj'}\gamma_{ii'} - \Gamma_{ij',i'j}, \tag{4.10}$$

$$T1_{ijk,i'j'k'} = \mathcal{A}[ijk]\mathcal{A}[i'j'k']\left(\frac{1}{6}\delta_{ii'}\delta_{jj'}\delta_{kk'} - \frac{1}{2}\delta_{ii'}\delta_{jj'}\gamma_{k,k'} + \frac{1}{4}\delta_{ii'}\Gamma_{jk,j'k'}\right), \tag{4.11}$$

$$T2_{ijk,i'j'k'} = \mathcal{A}[jk]\mathcal{A}[j'k']\left(\frac{1}{2}\delta_{jj'}\delta_{k,k'}\gamma_{ii'} + \frac{1}{4}\delta_{ii'}\Gamma_{j'k',jk} - \delta_{jj'}\Gamma_{ik',i'k}\right), \tag{4.12}$$

where δ is the Kronecker delta symbol and $\mathcal{A}[ijk]f(i,j,k) = f(i,j,k) + f(j,k,i) + f(k,i,j) - f(i,k,j) - f(j,i,k) - f(k,j,i)$. The $T2$ variable can be strengthened to yield the $T2'$ variable described in [3, 21]. We should point out that each of (4.8)–(4.12) is in fact a set of equations enumerating all possible indices i, j, k, i', j', and k'. Since Γ is a Four-dimensional tensor satisfying (4.4) and (4.5), one can convert it to a two-dimensional matrix $\tilde{\Gamma}$, i.e.,

$$\Gamma_{ij,i'j'} = \tilde{\Gamma}_{j-i+(2d-i)(i-1)/2,j'-i'+(2d-i')(i'-1)/2}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Similar properties hold for Q. Hence, Γ and Q can be transformed into $\frac{d(d-1)}{2} \times \frac{d(d-1)}{2}$ matrices. Because (4.5) is not satisfied on the Four-dimensional tensor G, it can only be transformed into a $d^2 \times d^2$ matrix. By the antisymmetric properties of the six-dimensional tensors $T1$, $T2$ and $T2'$ [36, 3, 21], they can be transformed into $\frac{d(d-1)(d-2)}{6} \times \frac{d(d-1)(d-2)}{6}$, $\frac{d^2(d-1)}{2} \times \frac{d^2(d-1)}{2}$, and $\frac{d^2(d-1)+2d}{2} \times \frac{d^2(d-1)+2d}{2}$ matrices, respectively. For simplicity, we still use the notations $\Gamma, P, Q, G, T1, T2$, and $T2'$ to represent the matrices translated from these tensors. Finally, the corresponding N-representability condition of (4.8)–(4.12) is to require each matrix to be positive semidefinite.

4.3. The SDP formulations. Let $b = (\text{svec}(T), \text{svec}(V))^T \in \mathbb{R}^m$ and $y = (\text{svec}(\gamma), \text{svec}(\Gamma))^T \in \mathbb{R}^m$ be vectorized integral and reduced density matrices that appear in (4.3), respectively, where svec is used to turn a symmetric matrix U into a vector according to

$$\text{svec}(U) = (U_{11}, \sqrt{2}U_{12}, U_{22}, \sqrt{2}U_{13}, \sqrt{2}U_{23}, U_{33}, \dots, U_{nn}).$$

To simplify notations later, we rename matrices as $S_1 = \gamma$, $S_2 = P$, $S_3 = Q$, $S_4 = G$, $S_5 = T1$ and $S_6 = T2$, and treat both y and $\{S_j\}$ as variables in the SDP formulation. Using the definition of y, we can rewrite the equation $S_1 = \gamma$ as a system of linear equations

$$S_1 = \mathcal{A}_1^* y + C_1, \tag{4.13}$$

where $\mathcal{A}_1^* y = \sum_{p=1}^m A_{1p} y_p \in \mathbb{R}^{s_1 \times s_1}$ with $A_{1p} \in \mathbb{R}^{s_1 \times s_1}$ and $C_1 \in \mathbb{R}^{s_1 \times s_1}$. Obviously, $s_1 = d$ and C_1 is a zero matrix. Similarly, each of (4.8)–(4.12) can be written succinctly as

$$S_j = \mathcal{A}_j^* y - C_j, \; j = 2, \dots, l = 6, \tag{4.14}$$

where $\mathcal{A}_j^* y = \sum_{p=1}^m A_{jp} y_p$ with $A_{jp} \in \mathbb{R}^{s_j \times s_j}$ and $C_j \in \mathbb{R}^{s_j \times s_j}$. The integer s_j is equal to the matrix size of S_j. The matrices A_{jp} are coefficients matrices of y_p, and C_j are constant matrices in the corresponding equation of (4.8)–(4.12).

Using these notations, we can formulate the constrained minimization of (4.3) subject to N-representability conditions as an SDP:

$$\begin{aligned} \min_{y, S_j} \; & b^T y \\ \text{s.t.} \; & S_j = \mathcal{A}_j^* y - C_j, j = 1, \dots, l, \\ & B^T y = c, \\ & 0 \preceq S_1 \preceq I, \\ & S_j \succeq 0, j = 2, \dots, l, \end{aligned} \tag{4.15}$$

where the linear constraints $B^T y = c$ follows from the conditions (4.6)–(4.7) and other equality conditions introduced in [36]. If some of conditions in (4.8)–(4.12) are not considered, then (4.15) can be adjusted accordingly. If the condition on $T2$ is replaced by that of $T2'$, then we set $S_6 = T2'$.

The SDP problem given in (4.15) is often known as the dual formulation. The corresponding primal SDP of (4.15) is

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$$
\begin{aligned}
&\max_{X_j, U} \sum_{j=1}^{l} \langle C_j, X_j \rangle + \langle c, x \rangle - \langle C_1 + I, U \rangle \\
&\text{s.t. } \sum_{j=1}^{l} \mathcal{A}_j(X_j) + Bx - \mathcal{A}_1(U) = b, \\
&\quad X_j \succeq 0, j = 1, \ldots, l, \\
&\quad U \succeq 0,
\end{aligned} \tag{4.16}
$$

where $X_j \in \mathbb{R}^{s_j \times s_j}$, $U \in \mathbb{R}^{s_1 \times s_1}$, $\mathcal{A}_j$ is the conjugated operator of $\mathcal{A}_j^*$ and $\mathcal{A}_j(X) = (\langle A_{j1}, X \rangle, \ldots, \langle A_{jm}, X \rangle)^T$ for any matrix $X \in \mathbb{R}^{s_j \times s_j}$.

Since the largest matrix dimension of X_j and S_j is of order $O(d^3)$ and $m = O(d^4)$, (4.15) and (4.16) are large scale SDPs even for a moderate value d. However, the S_j in (4.15) are block diagonal matrices due to the spatial and spin symmetries of molecules. Hence, the computational cost for solving (4.15) can be reduced by exploiting such block diagonal structures. In Table 4.1, we list the number of diagonal blocks and their dimensions resulting from spin symmetries in each of $\gamma, \Gamma, Q, G, T1, T2, T2'$ matrices.

TABLE 4.1
The matrix dimensions of the block diagonal structures.

S_j matrix	block dimension
γ	$\frac{d}{2}$, 2 blocks;
P, Q, Γ	$\frac{d^2}{4}$, 1 blocks; $\frac{d}{4}(\frac{d}{2} - 1)$, 2 blocks;
G	$\frac{d^2}{2}$, 1 blocks; $\frac{d^2}{4}$, 2 blocks;
$T1$	$\frac{d^2}{8}(\frac{d}{2} - 1)$, 2 blocks; $\frac{d^2}{12}(\frac{d}{2} - 1)(\frac{d}{2} - 2)$, 2 blocks;
$T2$	$\frac{d^2}{8}(\frac{3d}{2} - 1)$, 2 blocks; $\frac{d^2}{8}(\frac{d}{2} - 1)$, 2 blocks;
$T2'$	$\frac{d}{2} + \frac{d^2}{8}(\frac{3d}{2} - 1)$, 2 blocks; $\frac{d^2}{8}(\frac{d}{2} - 1)$, 2 blocks.

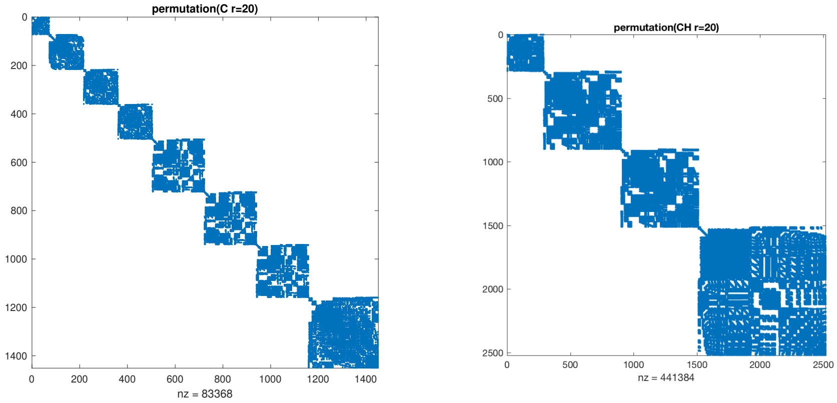
Spatial symmetry may lead to additional block diagonal structures within each spin diagonal block listed in Table 4.1. These block diagonal structures can be clearly seen within the largest spin block diagonal block of the T_2 matrices associated with the carbon atom and the CH molecules shown in Figure 4.1. These T_2 matrices are generated from spin orbitals obtained from the solution of the HF equation discretized by a double-ζ local atomic orbital basis. The block diagonal structure shown in Figure 4.1 is obtained by applying a suitable symmetric permutation to the rows and columns of the T_2 matrices. By representing the variables S_j as block diagonal matrices whose sizes are much smaller, the off-diagonal parts of S_j are no longer needed. Consequently, the length of y may be reduced and each of (4.13)–(4.14) may be split into several smaller systems. Therefore, it is possible to generate a much smaller SDP. Without loss of generality, we still consider the formulation (4.15) and our proposed algorithm can be applied to the reduced problems as well.

In addition to exploiting the block diagonal structure in the S_j matrices that appear in the dual SDP, we can also use the low rank structure of $\{X_i\}$ and U to reduce the cost for solving (4.16). The following theorem shows that $\{X_i\}$, $i = 1, 2, \ldots, l$ and U in the primal (4.16) are indeed low rank as long as d is sufficiently large.

THEOREM 4.1. *Assume that there exists matrices $\hat{X}_j \succ 0$ and $\hat{U} \succ 0$ such that the linear equality constraints of* (4.16) *are satisfied with them and the basis size d is larger than* 3. *Then there exists an optimal solution $\{X_1, \ldots, X_l, U\}$ of* (4.16) *such*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) C atom

(b) CH molecule

Fig. 4.1. *The block diagonal structures within the largest spin blocks of the T_2 matrices associated with the carbon (C) atom and the CH molecule.*

that $r = \sum_{j=1}^{l} r_j + r_u \leq \frac{\sqrt{3}}{8}(d^2+6)$, *where* r_j *is the rank of* X_j *and* r_u *is the rank of* U. *Moreover,* $r_j/s_j = O(1/d)$ *for* j*'s associated with the* $T1$, $T2$, *and* $T2$*' conditions.*

Proof. We first prove that there must exist a solution such that $r \leq \sqrt{m}$, where m is the length of the dual variable y in (4.15). The primal SDP (4.16) can be written as a standard SDP in the form of (1.1), where X is a block diagonal matrix whose diagonal parts are U, X_j, and $\operatorname{diag}(x)$. Then the size of X is $\sum_{j=1}^{l} s_j + s_u + 2s$. Let the rank of X be $\tilde{r}$. It follows from the results shown in [25] that $\frac{\tilde{r}(\tilde{r}+1)}{2} \leq m$, which implies $r \leq \sqrt{m}$. Since $m = \frac{3}{64}d^4 - \frac{1}{16}d^3 + \frac{9}{16}d^2 + \frac{1}{4}d \leq (\frac{\sqrt{3}}{8}(d^2+6))^2$ when $d \geq 3$, the first statement holds. The second statement follows from Table 4.1 that the dimension of the S_j matrices associated with the T1, T2, T2' conditions are on the order of $O(d^3)$. □

The ADMM has been successfully used to solve the 2-RDM problem in [22] where the method is referred to as the boundary point method. To apply the ADMM to solve (4.15), we first write the augmented Lagrangian function as

$$\begin{aligned} L(y, S_j; X_j, x) &= b^T y + \sum_{j=1}^{l} \langle X_j, S_j - \mathcal{A}_j^* y + C_j \rangle + \langle x, c - B^T y \rangle \\ &\quad + \frac{\sigma}{2}\left(\sum_{j=1}^{l} \|S_j - \mathcal{A}_j^* y + C_j\|_F^2 + \|c - B^T y\|_2^2 \right), \end{aligned} \tag{4.17}$$

where X_j and x are Lagrangian multipliers and $\sigma > 0$ is a penalty parameter. Then the kth iteration of ADMM consists of the following sequence of steps:

$$\begin{aligned} y^{k+1} &= \arg\min\nolimits_y L(y, S_j^k; X_j^k, x^k), \\ S_1^{k+1} &= \arg\min\nolimits_{0 \preceq S_1 \preceq I} L(y^{k+1}, S_j; X_j^k, x^k), \\ S_j^{k+1} &= \arg\min\nolimits_{S_j \succeq 0} L(y^{k+1}, S_j; X_j^k, x^k), \; j = 2, \dots, l, \\ X_j^{k+1} &= X_j^k + \sigma(S_j^{k+1} - \mathcal{A}_j^* y^{k+1} + C_j), \; j = 1, \dots, l, \\ x^{k+1} &= x^k + \sigma(c^{k+1} - B^T y^{k+1}). \end{aligned} \tag{4.18}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Although our semismooth Newton method is introduced on standard SDP, it is easy to extend it to the formula (4.16) without reducing (4.16) to the standard SDP. In our implemention, this is useful to avoid increasing the cost of computation.

5. Numerical results.

5. Numerical results. In this section, we demonstrate the effectiveness of the semismooth Newton Algorithm 2 dubbed SSNSDP. We implemented the algorithm mostly in MATLAB. The auxiliary parts (such as the user interface, various operations on matrices, etc.) of our codes are built based on SDPNAL [35], SDPNAL+ [4], and ADMM+ [28], and use the key implementation details and subroutines in these solvers. Some parts of the code are written in the C Language and interfaced with MATLAB through MEX-files. All experiments are performed on a single node of a PC cluster, where each node has two Intel Xeon 2.40 GHz CPUs with 12 cores and 256 GB RAM.

The test dataset is provided by Professor Maho Nakata and Professor Mituhiro Fukuta. The detailed information about the dataset such as the basis sets used to discretize molecular orbitals, the geometries of the molecules, etc. can be found in [23]. Since the original dataset only takes into account the spin symmetry, it does not specify additional block diagonal structures introduced by spatial symmetry of the molecular orbitals within each spin matrix block of the variables. We preprocess the dataset to identify these diagonal blocks automatically through matrix reordering. Our solver takes advantage of these block diagonal structures to reduce the complexity of the computation as described in subsection 4.3. We applied the semismooth Newton algorithm to the SDP formulation of the 2-RDM minimization problem with four different groups of N-representability conditions labeled as PQG, PQGT1, PQGT1T2, PQGT1T2'. The letters and numbers in each label simply indicate the N-representability conditions included in the SDP constraints. For example, PQGT1T2' means that the P, Q, G, T1, T2' conditions are included.

We compare SSNSDP with the state-of-the-art solvers SDPNAL and SDPNAL+ (version 1.0). The interior point methods are not included in the comparison because they usually perform worse than SDPNAL and SDPNAL+. We measure accuracy by examining four criteria: the primal infeasibility η_p and the dual infeasibility η_q that are defined by (3.21), the gap η_g between the primal and dual objective functions

$$\eta_g = \frac{|b^T y - \mathrm{tr}(C^T X)|}{1 + |\mathrm{tr}(C^T X)| + |b^T y|}, \tag{5.1}$$

and the difference between the 2-RDM energy and full CI energy defined by

$$\mathrm{err} = b^T y - \mathrm{energy}_{\mathrm{fullCI}}, \tag{5.2}$$

where $\mathrm{energy}_{\mathrm{fullCI}}$ values are taken from [23]. The last criterion is often used in quantum chemistry to assess the accuracy of an approximation model. It is used here to assess the effectiveness of adding additional N-representability conditions in the 2-RDM formulation. In the following tables, we use a short notation for the exponential form. For example, -4.8-3 means -4.8×10^{-3}.

We stop SSNSDP, SDPNAL, and SDPNAL+ when $\max\{\eta_p, \eta_d, \eta_k\} < 10^{-6}$, where

$$\eta_k = \max\left\{ \frac{\|X - \Pi_{S_+^n}(X)\|_F}{1 + \|X\|_F}, \frac{\|S - \Pi_{S_+^n}(S)\|_F}{1 + \|S\|_F}, \frac{|\mathrm{tr}(S^T X)|}{1 + \|S\|_F + \|X\|_F} \right\}. \tag{5.3}$$

Note that SDPNAL and SDPNAL+ also implement a very complicated set of rules to detect stagnation of the iterations and can stop earlier whenever there is not much

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

progress. However, the comparison is fair since their performance will be worse if these stagnation rules are removed. For solving the inner linear system in Algorithm 1 for SSNSDP, we use symmetric QMR method, which is also used in SDPNAL and SDPNAL+. The stop criterion of QMR method in SSNSDP is set to $\|r^k\|_F \leq \max\{\min\{0.1, 0.1 \times \|F^k\|_F\}, 10^{-10}\} \times 10^{-3}$ for simplicity. The stopping criteria of QMR method in SDPNAL and SDPNAL+ are much more complicated in order to achieve a faster convergence speed. Since η_k is usually very small in all three solvers, we do not report it in our tables.

In Table 5.1, we compare the performance of SSNSDP when it is applied to the orginal dataset provided in [23] and our preprocessed data that take advantage of additional block diagonal structures through permutation. We can see that the CPU time measured in seconds (the column labeled by "t" in Table 5.1) can be reduced by at least a factor of three on most examples labeled with PQGT1T2 and PQGT1T2'. For the C atom and F$^-$ system that exhibit a high spatial symmetry, the CPU time can be reduced by a factor of roughly six for SDPs that include the PQGT1T2 and PQGT1T2' conditions. These experiments illustrate the importance of exploiting spatial symmetry to identify block diagonal structures in the approximate solution and consequently reduce the computational cost significantly. For the problems that only include the PQG and PQGT1 conditions, the amount of improvement is less spectacular, because the sizes of the diagonal blocks in these examples are small. These small examples does not mainly focus on large scalability. In fact, the larger the blocks in Table 4.1 is, the more significant effectiveness of the symmetry is. Thereafter, all experiments are performed on the preprocessed data.

Table 5.1

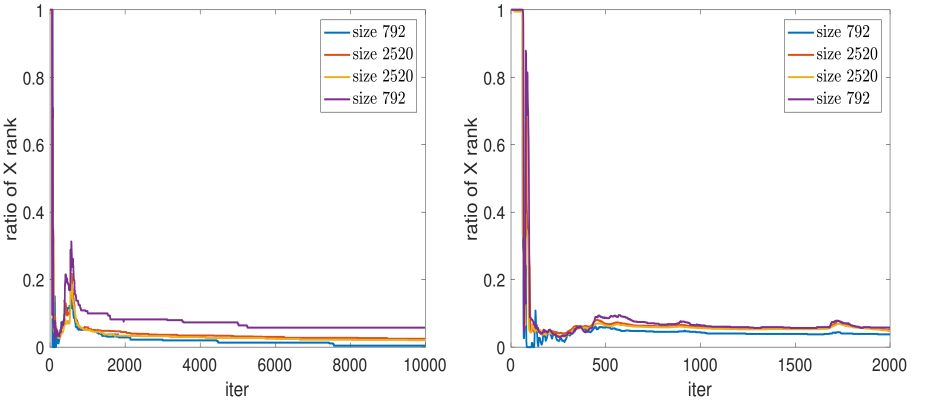
The comparison of the performance on the original and preprocessed SDPs. The number $-4.8-3$ *means* -4.8×10^{-3}.

		Preprocessed SDP						Original SDP					
System	Condition	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t
C	PQG	-4.3-3	1.3-7	3.2-7	2.4-6	1046	51	-4.3-3	7.9-7	9.0-7	2.6-6	1042	65
C	PQGT1	-3.5-3	4.4-7	5.0-7	1.8-6	614	62	-3.4-3	6.8-7	9.7-7	9.0-7	598	119
C	PQGT1T2	-7.3-4	7.3-7	6.8-7	1.4-6	656	235	-7.4-4	6.1-7	8.1-7	1.4-6	654	1265
C	PQGT1T2'	-3.4-4	4.4-7	7.7-7	2.0-6	636	214	-3.1-4	7.6-7	7.7-7	1.7-6	663	1395
CH	PQG	-1.3-2	7.3-7	4.6-7	6.0-7	585	67	-1.3-2	6.2-7	7.1-8	6.0-7	572	92
CH	PQGT1	-9.9-3	3.9-7	8.9-7	1.7-6	1546	169	-9.9-3	3.9-7	8.9-7	1.7-6	1546	377
CH	PQGT1T2	-2.0-3	7.8-7	4.0-7	3.0-6	617	1194	-2.0-3	6.7-7	4.3-7	3.0-6	614	3808
CH	PQGT1T2'	-1.0-3	5.4-7	8.2-7	5.4-6	682	1199	-1.0-3	5.8-7	8.2-7	5.7-6	686	3973
F$^-$	PQG	-1.2-2	9.8-7	3.3-7	5.2-7	983	44	-1.2-2	7.6-7	2.2-7	6.2-7	984	72
F$^-$	PQGT1	-9.6-3	6.4-7	7.2-7	8.8-6	560	137	-1.1-2	5.2-7	9.5-7	1.3-5	1697	1012
F$^-$	PQGT1T2	-1.2-3	6.0-7	3.0-7	1.9-6	659	1060	-1.2-3	6.9-7	3.4-7	1.7-6	668	9395
F$^-$	PQGT1T2'	-1.9-3	6.0-7	9.4-7	4.9-6	683	1078	-2.1-3	6.5-7	8.8-7	5.8-6	666	6922
H_2O	PQG	-1.9-2	2.9-7	1.9-7	3.8-7	987	82	-1.9-2	6.3-7	2.8-7	5.4-7	974	105
H_2O	PQGT1	-1.1-2	9.2-7	6.5-7	2.1-6	569	262	-1.1-2	9.2-7	6.5-7	2.1-6	569	719
H_2O	PQGT1T2	-2.4-3	4.7-7	6.9-7	5.8-6	653	3579	-2.4-3	6.1-7	7.4-7	6.1-6	676	12839
H_2O	PQGT1T2'	-1.5-3	8.3-7	6.4-7	3.8-6	625	3255	-1.5-3	9.5-7	8.3-7	3.7-6	625	11105

In addition to identifying block diagonal structures in the N-representibility constraints, we can further improve the efficiency of SSNSDP by taking advantage of the low rank structure of the variable matrices. Recall from Theorem 4.1 that the ratios of the rank of the X_j matrix (denoted by r_j) associated with the T2 condition over the dimension of X_j (denoted by d_j) should be bounded by $(\frac{\sqrt{3}}{8}(d^2+6))/(\frac{d^2}{8}(\frac{3d}{2}-1))$. For the C atom and CH molecule, d is 20 and 24, respectively. Thus, at the solution the ratios should be bounded by 0.06 and 0.05, respectively. In Figure 5.1, we

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) C (b) CH

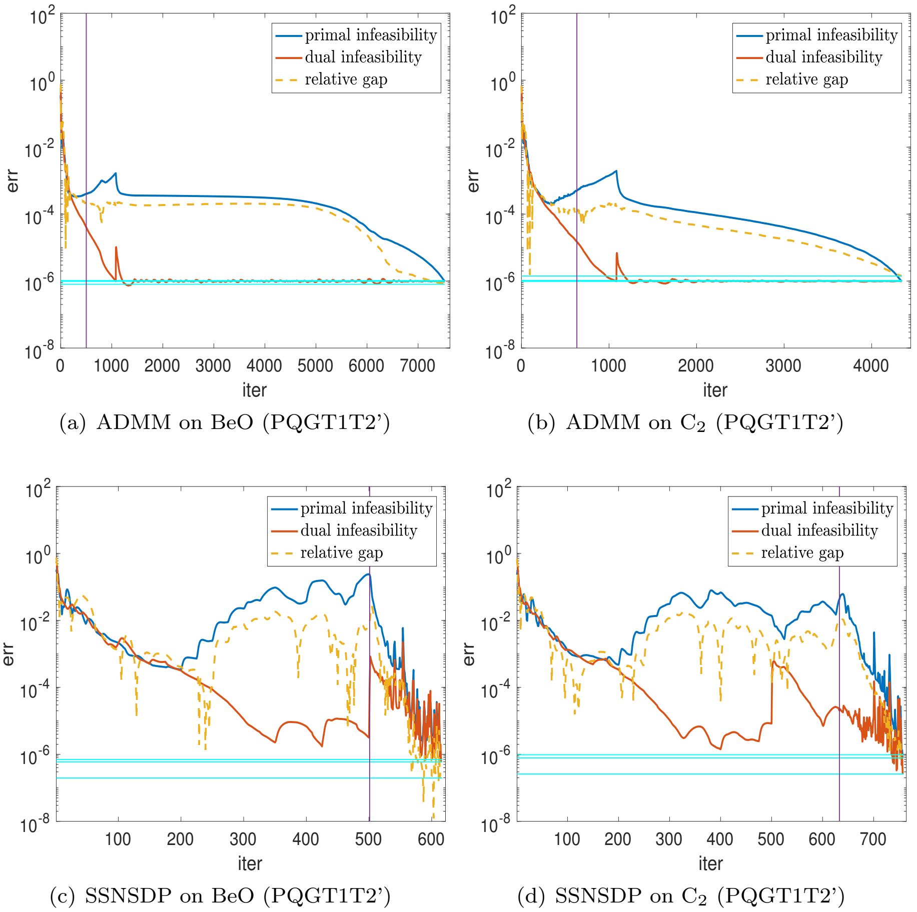
Fig. 5.1. *The percentage of the ranks of the first four largest blocks of the X in the C and CH systems.*

replace r_j by the numerical rank computed from the eigenvalue decompositions of the X_j variable and show the ratios for j's that are associated with the four largest d_j's at each DRS iteration. We observe that these ratios can be relatively high in the first few iterations. But they eventually become less than 0.1 after a few hundred iterations. This property is useful for (3.15) in the DRS and the semismooth Newton methods. It follows from (3.23) that the X variable is the projection of the Z variable to semidefinite cone and $|\alpha|$ in (3.15) is equal to the rank of X_j in the case of 2-RDM. Therefore, solving the Newton system (3.14) becomes much cheaper by using (3.15) when $|\alpha|$ is small.

Figure 5.2 shows how the relative gap, primal infeasibility, and dual infeasibility in ADMM and SSNSDP change with respect to the number of iterations when they are applied to the BeO and C_2 system. We tested both algorithms on SDPs with the PQGT1T2' N-representibility conditions. The convergence history of ADMM and SSNSDP is shown in the subfigures (a)–(b) and (c)–(d), respectively. SSNSDP is initiated from the solution produced by running some ADMM steps (501 steps in subfigure (c) and 633 steps in subfigure (d)), which is marked by vertical lines. The iteration history of ADMM in (a)–(b) and (c)–(d) is a little bit different since the parameters of ADMM steps in the pure ADMM and these in SSNSDP are tuned differently for their own performance, respectively. We can see that the ADMM can produce a moderately accurate solution in a few hundred iterations and then becomes slow. Many more iterations are required to reach a high accuracy. Starting from a point at which ADMM almost stagnates, SSNSDP is able to obtain a more accurate solution around 100 steps. Table 5.2 gives a detailed explanation on the acceleration of semismooth Newton steps on a few selected examples that ADMM method spends too many iterations. SSNSDP performs better on these examples in terms of the number of iterations (the column labeled by "it") and CPU time (the column labeled by "t"). Note that the duality gap as well as the primal and dual infeasibilities curves shown in (c)–(d) are highly oscillatory. The oscillation is due to the adaptive update of the penalty parameter μ for achieving a faster overall convergence rate. If the penalty parameter is fixed, these curves become much smoother, while more iterations are needed to reach the desired accuracy.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) ADMM on BeO (PQGT1T2')

(b) ADMM on C_2 (PQGT1T2')

(c) SSNSDP on BeO (PQGT1T2')

(d) SSNSDP on C_2 (PQGT1T2')

FIG. 5.2. *Relative gap, primal infeasibility, and dual infeasibility.*

TABLE 5.2
The comparison of the performance of ADMM and SSNSDP.

		ADMM						SSNSDP					
System	Condition	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t
B_2	PQGT1T2	-6.7-2	2.8-5	1.1-6	2.3-5	20000	10880	-6.6-2	4.5-7	4.4-7	4.9-6	724	2426
B_2	PQGT1T2'	-6.4-2	8.5-7	1.1-6	1.4-6	14510	7883	-6.5-2	3.4-7	7.3-7	1.7-6	642	1969
BH	PQGT1T2	-5.5-4	9.6-7	1.1-6	2.9-6	5758	3154	-6.5-4	9.6-7	9.4-7	3.8-6	676	1237
BH	PQGT1T2'	-3.9-4	9.6-7	1.1-6	3.2-6	4362	2481	-5.3-4	8.5-7	9.3-7	6.7-6	650	1009
BH_3O	PQGT1T2	-2.0-3	9.8-7	8.8-7	3.1-6	5036	8541	-9.1-4	4.9-7	7.4-7	9.7-7	635	3555
BH_3O	PQGT1T2'	-1.4-3	9.8-7	8.8-7	1.2-6	2148	3954	-9.9-4	7.6-7	1.7-7	6.6-7	815	3924
BeO	PQGT1T2	-1.9-3	8.6-7	1.1-6	2.2-7	10261	2003	-1.9-3	7.4-7	4.1-7	2.1-7	674	648
BeO	PQGT1T2'	-2.0-3	9.8-7	1.0-6	8.0-7	7521	1492	-1.9-3	7.0-7	6.0-7	2.0-7	615	407
C_2	PQGT1T2	1.7-2	9.3-3	1.9-6	8.1-4	20000	41694	-4.7-3	8.0-7	8.8-7	3.4-6	889	13987
C_2	PQGT1T2'	-4.0-3	9.3-7	1.3-6	2.5-6	13363	28505	-3.7-3	8.0-7	5.1-7	1.2-6	710	10645
CH	PQGT1T2	-2.0-3	9.7-7	1.1-6	1.9-6	12723	6292	-2.0-3	7.8-7	4.0-7	3.0-6	617	1194
CH	PQGT1T2'	-7.5-4	9.7-7	1.1-6	2.5-6	3975	2140	-1.0-3	5.4-7	8.2-7	5.4-6	682	1199

In Table 5.3, we compare the accuracy and efficiency of SSNSDP, SDPNAL and SDPNAL+. Due to space limitations, this table only shows the systems with

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 5.3
A summary of computational results of SSNSDP, SDPNAL, and SDPNAL+.

	SSNSDP						SDPNAL						SDPNAL+					
System	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t
AlH	-9.6-5	5.7-7	9.9-7	1.3-7	952	197	-5.3-4	4.8-6	5.1-7	2.8-6	175	410	-1.2-5	9.8-7	9.5-7	9.5-9	902	527
B_2	-6.5-2	3.4-7	7.3-7	1.7-6	642	1969	-6.5-2	1.7-5	7.2-7	5.8-6	245	2226	-6.5-2	8.8-7	9.7-7	7.1-7	2044	3066
BF	-5.8-4	5.3-7	5.9-7	6.8-7	1480	444	-7.9-4	7.7-6	6.0-7	6.7-6	195	458	-4.2-4	9.8-7	9.8-7	5.3-7	1211	643
BH^+	-5.5-5	6.9-7	9.7-7	8.1-7	1198	25	-1.2-4	2.2-6	6.8-7	1.4-6	212	87	-1.4-4	3.2-7	1.0-6	1.2-6	1538	50
BH	-5.3-4	8.5-7	9.3-7	6.7-6	650	1009	-6.1-4	8.0-5	6.9-7	5.9-5	272	1969	-6.9-4	7.8-12	1.0-6	5.0-6	4502	4347
BH_3O	-9.9-4	7.6-7	1.7-7	6.6-7	815	3924	-1.7-3	1.1-5	6.8-7	8.6-6	203	4696	-1.1-3	7.5-7	9.7-7	1.1-6	1230	6793
BN	-2.7-3	3.7-7	3.8-7	1.1-7	803	599	-3.3-3	2.2-5	7.2-7	9.2-6	234	497	-3.0-3	7.9-7	9.9-7	7.0-8	1645	675
BO	-1.2-3	8.1-7	5.6-7	5.2-7	556	397	-1.6-3	9.0-6	7.0-7	7.8-6	191	708	-1.2-3	8.8-7	7.9-7	5.7-7	1738	1252
Be(1)	-1.3-5	9.4-7	2.2-7	6.2-7	739	9	-4.7-5	2.0-7	9.1-7	7.2-7	136	20	-4.8-5	8.8-8	1.0-6	6.2-7	1216	14
Be(2)	-6.4-5	4.4-7	9.5-7	1.7-6	591	125	-1.6-4	6.9-5	6.9-7	2.9-6	241	238	-2.0-4	2.7-7	9.9-7	3.0-6	2694	344
BeF	-3.4-4	5.1-7	7.6-7	6.2-7	1278	356	-6.6-4	1.2-5	6.5-7	8.7-6	197	480	-1.9-4	7.7-7	9.9-7	7.9-7	1035	560
BeH^+	-5.8-6	9.1-7	9.6-7	1.3-7	1329	28	-7.3-5	8.5-6	5.4-7	2.2-6	218	90	-1.0-4	6.2-7	9.7-7	1.6-6	1840	56
BeH	-2.4-5	9.3-7	9.2-7	4.9-7	1153	24	-8.5-5	6.0-6	7.3-7	2.3-6	221	95	-9.4-5	2.5-7	9.9-7	1.4-6	1332	44
BeO	-1.9-3	7.0-7	6.0-7	2.0-7	615	407	-2.2-3	1.3-5	7.2-7	1.2-5	219	499	-2.1-3	7.1-7	9.1-7	8.2-7	3060	998
C(1)	-3.4-4	4.4-7	7.7-7	2.0-6	636	214	-5.5-4	3.2-5	5.8-7	1.6-5	265	432	-1.2-3	3.5-14	2.1-6	6.8-6	20081	2482
C(2)	-2.4-3	8.7-7	6.5-7	3.2-6	1463	250	-2.6-3	1.5-5	6.4-7	6.6-6	253	400	-2.8-3	3.7-7	9.9-7	5.2-6	1908	617
C_2^-	-2.1-3	5.3-7	4.8-7	1.7-7	624	273	-2.4-3	6.9-6	6.0-7	6.7-6	195	298	-2.5-3	3.3-7	9.5-7	1.3-6	1680	423
$C_2(1)$	-3.7-3	7.8-7	2.6-7	9.7-7	757	337	-4.2-3	5.1-6	7.6-7	2.6-6	204	314	-4.1-3	2.9-7	9.2-7	1.0-6	1775	426
$C_2(2)$	-3.7-3	8.0-7	5.1-7	1.2-6	710	10645	3.7-3	5.1-4	7.0-7	2.6-4	254	7246	-7.2-3	2.4-14	1.4-5	3.6-6	1965	10004
CF	-7.1-4	5.6-7	7.1-7	4.0-7	1718	488	-8.8-4	8.4-6	5.0-7	5.6-6	188	442	-7.4-5	9.9-7	9.1-7	2.2-6	1628	845
CH	-1.0-3	5.4-7	8.2-7	5.4-6	682	1199	-1.1-3	9.1-5	6.3-7	5.2-5	254	1879	-1.5-3	7.4-14	4.8-5	1.8-5	14048	10001
CH_2	-1.2-3	4.8-7	6.9-7	6.1-6	720	4070	-1.3-3	1.3-4	6.4-7	1.2-4	261	4737	-1.5-3	1.1-11	1.1-6	5.9-6	3830	10001
CH_3^+	-3.5-4	1.2-7	6.1-7	1.4-6	1196	108	-5.6-4	9.7-7	8.8-7	1.3-6	178	149	-5.0-4	7.4-7	9.5-7	9.9-7	965	144
CH_3	-8.0-4	8.1-7	3.1-7	3.9-6	640	4327	-1.4-3	3.3-5	7.7-7	2.5-5	223	7691	-1.4-3	2.7-7	1.3-6	3.3-6	3348	10124
CH_3N	-1.7-3	3.7-7	7.0-7	1.4-6	564	2271	-2.0-3	9.0-6	6.3-7	5.7-6	190	4162	-1.8-3	9.9-7	8.4-7	3.0-7	1496	7354
CH_4	-3.2-4	3.9-7	6.5-7	4.9-7	586	119	-7.5-4	9.0-7	8.8-7	1.8-6	168	165	-7.6-4	5.2-7	1.0-6	1.6-6	1328	199
CN	-1.9-3	2.8-7	4.4-7	3.2-7	653	395	-2.2-3	1.2-5	5.4-7	9.8-6	205	468	-2.0-3	1.0-6	8.9-7	5.5-7	1908	705
CO^+	-1.5-3	7.5-7	1.5-7	8.3-8	659	558	-2.0-3	1.0-5	7.8-7	8.9-6	194	445	-1.7-3	1.0-6	7.8-7	6.8-7	1786	730
CO	-8.3-4	4.0-7	7.2-7	4.3-7	1183	288	-1.3-3	1.3-5	6.7-7	9.5-6	182	403	-1.1-3	6.7-7	8.0-7	1.2-7	1376	621
F^-	-1.9-3	6.0-7	9.4-7	4.9-6	683	1078	-2.0-3	8.7-5	5.1-7	7.3-5	258	1416	-2.4-3	1.0-6	8.5-7	8.0-6	2644	2800
FH_2^+	-8.7-5	9.6-7	9.9-7	2.6-8	898	33	-2.3-4	1.1-6	6.3-7	1.1-6	166	91	-2.5-4	1.0-7	9.9-7	6.7-7	614	36
H_2O	-1.5-3	8.3-7	6.4-7	3.8-6	625	3255	-1.9-3	7.3-5	4.8-7	5.5-5	266	5663	-2.0-3	7.3-7	8.8-7	5.6-6	3420	9290
H_3	-1.1-5	6.7-7	8.7-7	4.3-7	571	32	-3.3-5	7.2-7	9.2-7	3.3-6	163	51	-1.6-5	9.6-7	8.9-7	8.1-7	1026	39
HF	-1.3-3	5.4-7	5.5-7	2.7-6	718	1534	-2.3-3	5.5-5	6.8-7	3.8-5	236	1732	-2.3-3	8.3-7	9.5-7	6.1-6	3062	3183
HLi_2	-1.5-4	6.8-7	8.9-7	2.7-6	867	983	-2.8-4	1.7-5	7.6-7	1.1-5	260	1140	-9.7-5	1.1-13	1.0-6	9.8-8	3820	1941
HN_2^+	-1.7-3	6.4-7	3.6-7	3.4-7	565	440	-2.2-3	8.3-6	7.8-7	5.4-6	187	725	-1.9-3	9.0-7	8.3-7	6.2-7	1532	883
HNO	-1.1-3	4.4-7	4.6-7	1.8-8	704	1758	-1.5-3	1.4-5	7.0-7	1.1-5	213	1562	-1.2-3	8.5-7	9.2-7	9.0-7	1286	1778
Li	-4.4-6	5.8-7	9.4-7	2.5-7	1440	18	-1.7-5	1.9-7	6.4-7	8.6-7	145	24	-2.4-5	4.5-7	1.0-6	1.1-6	1123	14
Li_2	-1.5-4	4.2-7	8.1-7	3.0-6	824	376	-2.0-4	2.3-5	6.8-7	1.2-5	262	486	-2.4-4	3.0-8	1.0-6	4.3-6	5826	1305
LiF	-3.8-4	9.8-7	8.1-7	1.5-7	614	461	-6.6-4	9.4-6	6.2-7	5.6-6	217	541	-3.5-4	1.0-6	9.0-7	7.5-7	1830	832
LiH(1)	-4.7-5	6.0-7	5.6-7	1.5-6	906	1306	-1.2-4	2.4-5	7.3-7	8.3-6	253	1773	-2.7-4	8.1-14	5.4-6	1.7-5	17870	10001
LiH(2)	-7.6-6	5.8-7	9.6-7	3.1-7	1252	26	-5.9-5	7.6-6	6.7-7	2.3-6	232	103	-7.1-5	3.8-7	9.4-7	2.1-6	1455	55
LiOH	-6.7-4	8.8-7	2.9-7	1.8-7	634	901	-1.0-3	9.9-6	5.4-7	7.5-6	203	852	-6.7-4	7.7-7	6.9-7	3.5-7	2098	1472
N	-2.2-4	8.7-7	5.7-7	5.8-7	628	223	-5.0-4	6.7-5	4.9-7	3.3-5	229	347	-1.1-3	3.7-7	1.4-6	5.3-6	20144	2735
N_2^+	-2.3-3	5.3-7	1.1-7	1.6-7	649	328	-2.8-3	5.5-6	7.6-7	5.4-6	187	282	-2.8-3	7.5-7	1.0-6	1.0-7	3939	825
N_2	-1.3-3	6.6-7	6.3-7	3.7-9	1138	209	-1.5-3	8.4-6	4.3-7	4.1-6	180	289	-2.0-3	5.1-14	1.5-6	1.6-6	20058	2587
NH(1)	-1.3-3	4.6-7	8.3-7	3.5-6	750	1611	-1.3-3	4.4-5	5.0-7	3.6-5	264	2000	-1.5-3	1.6-13	1.0-6	4.1-6	3434	3590
NH(2)	-7.1-4	3.9-7	5.9-7	2.7-6	706	1289	-9.7-4	1.1-4	5.2-7	8.1-5	253	1763	-7.8-4	6.8-13	9.4-7	1.4-6	4046	3956
$NH_2^-(1)$	-1.8-3	9.5-7	8.0-7	5.3-6	654	3654	-1.8-3	6.9-5	5.0-7	6.4-5	255	5413	-1.9-3	9.2-7	1.0-6	3.8-6	2602	8365
$NH_2^-(2)$	-8.5-5	9.3-7	9.9-7	1.9-7	1085	41	-1.6-4	1.4-6	4.8-7	9.1-7	171	93	-1.9-4	1.9-7	9.7-7	5.6-7	817	45
NH_3^+	-2.1-4	4.1-7	2.7-7	3.5-7	1174	116	-3.7-4	1.2-6	5.5-7	8.5-7	195	173	-2.7-4	8.7-7	7.9-7	2.1-7	854	129
NH_3	-1.1-3	4.6-7	7.9-7	2.7-6	686	7953	-1.6-3	9.4-6	5.8-7	8.4-6	259	13049	-2.0-3	1.3-6	3.9-6	1.1-6	307	11003
NH_4^+	-4.0-4	4.4-7	2.4-7	4.8-7	870	132	-6.1-4	1.9-6	6.2-7	9.0-7	182	186	-6.8-4	4.5-7	9.6-7	1.2-6	1228	202
Na	-2.4-4	1.0-6	9.6-7	7.9-7	1160	93	-5.2-4	4.4-6	6.4-7	1.9-6	184	174	-4.2-4	1.2-7	9.4-7	5.7-7	724	101
NaH	-2.6-4	8.1-7	6.2-7	7.0-10	1134	307	-7.9-4	5.4-6	7.2-7	3.1-6	199	481	-3.9-4	6.3-7	8.3-7	1.4-7	1161	490
Ne	-8.7-4	8.9-7	9.5-7	1.5-6	563	141	-2.5-3	2.0-5	7.6-7	1.7-5	208	314	-2.6-3	8.5-7	9.1-7	5.8-6	2370	490
O(1)	-2.0-3	5.2-7	9.6-7	3.4-6	644	216	-2.0-3	2.1-5	4.5-7	1.4-5	216	318	-2.6-3	5.1-10	1.0-6	5.0-6	2651	710
O(2)	-7.9-4	3.5-7	6.2-7	2.6-6	605	183	-1.2-3	7.3-5	5.5-7	4.5-5	217	323	-1.6-3	5.8-7	9.4-7	5.1-6	1661	532
O(3)	-2.5-3	5.7-7	8.7-7	3.3-6	784	308	-2.5-3	1.7-5	5.2-7	1.0-5	235	339	-3.0-3	8.1-7	9.9-7	5.2-6	2696	715
O_2^+	-2.2-3	5.0-7	7.1-7	1.8-7	1204	216	-2.4-3	4.4-6	5.6-7	3.3-6	172	282	-2.5-3	3.6-7	9.6-7	8.1-7	939	248
P	-4.6-4	9.1-7	9.5-7	1.8-7	674	1232	-1.1-3	6.9-6	6.3-7	3.5-6	208	1140	-6.3-4	3.5-13	1.0-6	3.6-7	640	2143
SiH_4	-2.6-4	6.4-7	2.2-7	1.5-7	768	1493	-1.0-3	5.6-6	5.1-7	2.3-6	185	1751	-3.1-4	3.5-13	1.0-6	8.8-7	817	2308

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Table 5.4
A statistic of computational results of SSNSDP, SDPNAL, and SDPNAL+.

Case	SSNSDP		SDPNAL		SDPNAL+	
	Number	Percentage	Number	Percentage	Number	Percentage
Success	276	100%	53	19.2%	265	96%
Fastest	202	73.2%	24	8.7%	50	18.1%
Fastest under success	222	80.4%	3	1.09%	51	18.5%
Not slower 1.2 times	233	84.4%	46	16.7%	78	28.3%
Not slower 1.2 times under success	244	88.4%	3	1.09%	79	28.6%

PQGT1T2' constraints, i.e., the largest problem for every system. The statistics of all examples are shown in Table 5.4. The columns labeled by "it" in Table 5.3 are the summation of the number of Newton systems solved and the number of ADMM steps. The columns labeled by "t" give the CPU time in seconds. From the table, we can observe that SSNSDP is faster than SDPNAL and SDPNAL+ on most examples, for achieving almost the same level of accuracy. In some examples, SDPNAL or SDPNAL+ may take less CPU time, but either the primal or dual infeasibilities is still larger than the order 10^{-6} and that of SSNSDP. Table 5.4 gives a clearer explanation on the comparisons. In the table, "success" means the primal and dual infeasibilities are less than 2×10^{-6} (the factor 2 is used to avoid a small numerical error). The case "fastest" means that the amount of the CPU time of the algorithm is the smallest among three methods. The case "fastest under success" means the fastest algorithm under the success condition. The case "not slower 1.2 times" means that the amount of the CPU time of the algorithm is not 1.2 times slower than the fastest algorithm. The last case corresponds to the "not slower 1.2 times" case under the success condition. It can be seen that SSNSDP and SDPNAL+ reach the stop rule on most examples, but SDPNAL is only success on around 19.2% examples. SSNSDP converges fastest on around 80.4% examples, and it is not 1.2 times slower than the two other solvers on around 88.4% examples under the success condition. The corresponding percentage of SDPNAL and SDPNAL+ seems to be further smaller than SSNSDP.

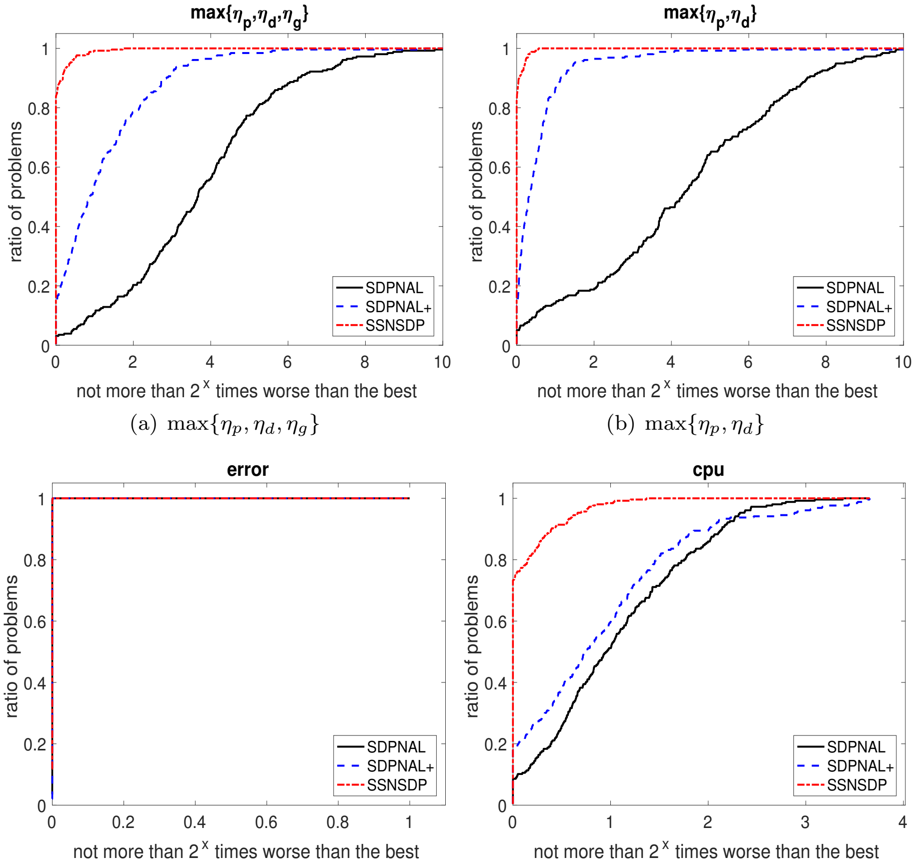
We next compare the accuracy and efficiency of SSNSDP with that of SDPNAL and SDPNAL+ using the the performance profiling method proposed in [10]. Let $t_{p,s}$ be the number of iterations or CPU time required to solve problem p by the sth solvers. Then one computes the ratio $r_{p,s}$ between $t_{p,s}$ over the smallest value obtained by n_s solvers on problem p, i.e., $r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s}:1\leq s\leq n_s\}}$. For $\tau \geq 0$, the value

$$\pi_s(\tau) := \frac{\text{number of problems where } \log_2(r_{p,s}) \leq \tau}{\text{total number of problems}}$$

indicates that solver s is within a factor $2^\tau \geq 1$ of the performance obtained by the best solver. Then the performance plot is a curve $\pi_s(\tau)$ for each solver s as a function of τ. In Figure 5.3, we show the performance profiles of four criteria $\max\{\eta_p, \eta_d, \eta_g\}$, $\max\{\eta_p, \eta_d\}$, err, and CPU time. In particular, the intercept point of the axis "ratio of problems" of each subfigure is the percentage of the fastest one among three solvers, which is also shown in the second row in Table 5.4. These figures show that the accuracy and the CPU time of SSNSDP are better than SDPNAL and SDPNAL+ on most problems.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) $\max\{\eta_p, \eta_d, \eta_g\}$ (b) $\max\{\eta_p, \eta_d\}$

(c) error (d) CPU

FIG. 5.3. *The performance profiles of SSNSDP, SDPNAL, and SDPNAL+.*

Finally, we report the accuracy of the solution produced by SSNSDP by comparing the 2-RDM ground state energy with the FCI energy and calculating their differences defined by (5.2) in Table 5.5. The last two columns labeled by n and m are the total number of dimensions of the primal variable X and the dual variable y of the standard form SDP under the PQGT1T2' condition. To compute a very accurate 2RDM energy, the stopping rule of SSNSDP is adjusted as $\eta_p < 2.5 \times 10^{-5}$ and $\eta_d < 10^{-8}$. The primal infeasibility η_p is allowed to be larger so that the algorithm converges more rapidly. The dual variables are required to be more accurate since we ultimately retrieve the desired 1-RDM and 2-RDM from the dual variables. We also choose proper parameters to make the penalty parameter μ larger so that the stopping rules can be easier satisfied. We can see that more accurate solutions are obtained from SSNSDP when more N-representability conditions are included in the constraints. These results are similar to the ones reported in [23].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Table 5.5

The error obtained by SSNSDP on various N-representability conditions: PQG, PQGT1, PQGT1T2, and PQGT1T2'.

System	State	Basis	PQG	PQGT1	PQGT1T2	PQGT1T2'	n	m
AlH	1Sigma+	STO6G	-2.3-3	-7.8-4	-2.3-5	-1.7-5	5780	7230
B_2	3Sigmag-	STO6G	-9.6-2	-8.5-2	-6.5-2	-6.4-2	5780	7230
BF	1Sigma+	STO6G	-6.7-3	-3.5-3	-3.3-4	-3.2-4	4254	4743
BH^+	2Sigma+	STO6G	-5.0-5	-3.1-5	-2.2-6	-7.9-7	5780	7230
BH	1Sigma+	DZ	-6.5-3	-4.7-3	-7.9-5	-6.0-5	5780	7230
BH_3O	1A1	STO6G	-2.8-2	-1.2-2	-7.0-4	-7.1-4	1324	948
BN	3Pi	STO6G	-2.9-2	-1.7-2	-3.0-3	-2.7-3	9848	15018
BO	2Sigma+	STO6G	-1.2-2	-6.7-3	-1.3-3	-1.0-3	12454	20709
Be(1)	1S	STO6G	-8.4-7	-5.7-7	-7.9-7	-8.0-7	5780	7230
Be(2)	1S	SV	-6.1-5	-5.6-5	-2.0-6	-2.5-7	5780	7230
BeF	2Sigma+	STO6G	-3.2-3	-1.7-3	-2.8-4	-2.0-4	790	465
BeH^+	1Sigma+	STO6G	-2.9-5	-2.8-5	-3.0-7	-2.5-7	4254	4743
BeH	2Sigma+	STO6G	-5.2-5	-2.6-5	-1.1-6	-4.0-7	5780	7230
BeO	1Sigma+	STO6G	-1.3-2	-9.5-3	-1.7-3	-1.8-3	5780	7230
C(1)	3P	DZ	-4.0-3	-3.1-3	-4.1-4	-5.0-5	1324	948
C(2)	3PSZ0	DZ	-1.7-2	-1.4-2	-2.5-3	-2.0-3	1324	948
C_2^-	2Sigmag+	STO6G	-2.6-2	-1.4-2	-2.4-3	-1.9-3	5780	7230
C_2(1)	1Sigmag+	STO6G	-4.6-2	-2.5-2	-3.5-3	-3.5-3	790	465
C_2(2)	1Sigmag+	VDZ	-5.3-2	-5.3-2	-3.2-3	-3.5-3	5780	7230
CF	2Pir	STO6G	-7.7-3	-5.8-3	-6.3-4	-4.9-4	790	465
CH	2Pir	DZ	-1.3-2	-9.6-3	-9.3-4	-3.1-4	5780	7230
CH_2	1A1	DZ	-1.9-2	-1.2-2	-3.7-4	-3.8-4	5780	7230
CH_3^+	1Ep	STO6G	-1.3-2	-3.8-3	-1.7-4	-1.6-4	5780	7230
CH_3	2A2pp	VDZ	-1.7-2	-1.0-2	-1.1-3	-3.1-4	25344	76554
CH_3N	1A1	STO6G	-3.9-2	-1.6-2	-1.0-3	-9.9-4	5780	7230
CH_4	1A1	STO6G	-1.9-2	-4.1-3	-2.2-4	-1.9-4	9848	15018
CN	2Sigma+	STO6G	-2.4-2	-1.2-2	-2.1-3	-1.7-3	15484	27888
CO^+	2Sigma+	STO6G	-1.8-2	-9.2-3	-1.8-3	-1.4-3	15484	27888
CO	1Sigma+	STO6G	-1.2-2	-7.2-3	-8.7-4	-8.7-4	3024	2964
F^-	1S	DZ+d	-1.2-2	-7.6-3	-3.8-4	-2.6-4	18970	36795
FH_2^+	1A1	STO6G	-1.1-3	-5.1-4	-2.1-5	-1.8-5	12454	20709
H_2O	1A1	DZ	-1.9-2	-1.1-2	-4.9-4	-3.9-4	4254	4743
H_3	2A1p	DZ	-8.0-4	-5.8-4	-1.5-6	-1.1-8	5780	7230
HF	1Sigma+	DZ	-1.2-2	-5.8-3	-3.6-4	-2.6-4	5780	7230
HLi_2	2A1	STO6G	-1.0-3	-6.6-4	-7.8-5	-1.1-5	5780	7230
HN_2^+	1Sigma+	STO6G	-2.5-2	-1.1-2	-1.5-3	-1.5-3	12454	20709
HNO	1Ap	STO6G	-1.9-2	-1.4-2	-8.9-4	-9.1-4	2058	1743
Li	2S	STO6G	-5.9-8	-3.7-8	-3.2-8	-1.5-8	15484	27888
Li_2	1Sigmag+	STO6G	-3.7-4	-2.9-4	-5.2-6	-4.7-6	1324	948
LiF	1Sigma+	STO6G	-1.6-3	-1.3-3	-2.4-4	-2.5-4	9848	15018
LiH(1)	1Sigma+	DZ	-3.5-4	-2.0-4	-9.4-7	-1.5-7	7634	10593
LiH(2)	1Sigma+	STO6G	-3.6-5	-2.7-5	-1.3-7	-4.0-8	7634	10593
LiOH	1Sigma+	STO6G	-8.6-3	-4.0-3	-5.8-4	-5.8-4	7634	10593
N	4S	DZ	-2.4-3	-9.1-4	-9.3-5	-1.1-5	790	465
N_2^+	2Sigmag+	STO6G	-3.1-2	-1.6-2	-2.7-3	-2.2-3	5780	7230
N_2	1Sigmag+	STO6G	-1.2-2	-8.9-3	-1.2-3	-1.2-3	5780	7230
NH(1)	1Delta	DZ	-1.7-2	-1.3-2	-5.0-4	-4.4-4	9848	15018
NH(2)	3Sigma-	DZ	-9.7-3	-5.2-3	-5.7-4	-1.4-4	1324	948
NH_2^-(1)	1A1	DZ	-2.4-2	-1.5-2	-6.6-4	-5.7-4	7634	10593
NH_2^-(2)	1A1	STO6G	-2.0-3	-1.3-3	-3.3-5	-2.6-5	5780	7230
NH_3^+	2A2pp	STO6G	-9.8-3	-1.8-3	-2.1-4	-1.1-4	5780	7230
NH_3	1A1	VDZ	-2.3-2	-1.4-2	-5.0-4	-4.7-4	5780	7230
NH_4^+	1A1	STO6G	-1.7-2	-4.2-3	-2.5-4	-2.3-4	9848	15018
Na	2S	STO6G	-1.0-3	-5.0-4	-6.4-5	-4.9-5	9848	15018
NaH	1Sigma+	STO6G	-3.5-3	-1.6-3	-9.3-5	-7.9-5	15484	27888
Ne	1S	DZ	-6.7-3	-2.6-3	-2.6-4	-1.5-4	2058	1743
O(1)	1D	DZ	-1.9-2	-1.4-2	-1.3-3	-1.2-3	3024	2964
O(2)	3P	DZ	-1.2-2	-6.3-3	-7.5-4	-2.4-4	18970	36795
O(3)	3PSZ0	DZ	-2.3-2	-1.9-2	-3.0-3	-1.6-3	4254	4743
O_2^+	2Pig	STO6G	-1.7-2	-1.5-2	-2.5-3	-2.1-3	4254	4743
P	4S	631G	-8.7-4	-3.0-4	-6.8-5	-1.6-5	5780	7230
SiH_4	1A1	STO6G	-1.9-2	-3.6-3	-1.8-4	-1.7-4	5780	7230

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

6. Conclusion.

In this paper, we consider the v2-RDM model for approximating the solution to the molecular Schrödinger equation. Instead of computing the smallest eigenvalue of the many-electron Schrödinger operator, we minimize the total energy of the many-electron system with respect to 1-RDM and 2-RDM subject to some linear constraints imposed to enhance the N-representability of the decision variables. The minimization problem to be solved is an SDP. The solution of the SDP can be obtained from the solution of a system of nonlinear equations that can be derived from a fixed-point iteration of DRS applied to the original SDP. We present a semismooth Newton type method for solving this set of nonlinear equations. A switching strategy between first-order and second-order methods is developed to improve the stability of the method and to achieve global convergence. We exploit the block diagonal structure and low rank structure of the variables in the SDP to improve the computational efficiency. The computational results show that the proposed semismooth Newton method can achieve higher accuracy, and is competitive with the Newton-CG augmented Lagrangian methods SDPNAL and SDPNAL+.

Several components of the proposed semismooth Newton method can be further improved. For example, since eigenvalue decomposition is the most expensive step in the procedure for computing the Newton direction, a more efficient eigendecomposition method needs to be investigated. A better global convergent technique is also needed to improve the overall performance.

Acknowledgments. The authors thank Prof. Nakata Maho and Prof. Mituhiro Fukuta for sharing all data sets on 2-RDM, Jinmei Zhang for helping with test problem preparation, Prof. Kim-Chuan Toh and Prof. Defeng Sun for the discussion and the sharing on SDPNAL and SDPNAL+, and Dr. Andre Milzarek for his careful proofreading and detailed comments on our manuscripts. The authors are grateful to the associate editor and the anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, New York, 2011, https://doi.org/10.1007/978-1-4419-9467-7.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found Trends Mach. Learn., 3 (2011), pp. 1–122.

[3] B. J. Braams, J. K. Percus, and Z. Zhao, *The t1 and t2 representability conditions*, in Reduced-Density-Matrix Mechanics: with Application to Many-Electron Atoms and Molecules, D. A. Mazziotti, ed., Adv. Chem. Phys. 134, Wiley, Hoboken, NJ, 2007, pp. 93–101.

[4] D. Chaykin, C. Jansson, F. Keil, M. Lange, K. T. Ohlhus, and S. M. Rump, *Rigorous results in electronic structure calculations*, Optimization online, 2016.

[5] L. Chen, D. Sun, and K.-C. Toh, *A note on the convergence of ADMM for linearly constrained convex optimization problems*, Comput. Optim. Appl., 66 (2017), pp. 327–343.

[6] F. H. Clarke, *Optimization and Nonsmooth Analysis*, vol. 5, SIAM, Philadelphia 1990.

[7] A. J. Coleman, *Structure of Fermion density matrices*, Rev. Mod. Phys., 35 (1963), p. 668.

[8] D. Davis and W. Yin, *Convergence rate analysis of several splitting schemes* http://arxiv.org/abs/1406.4834v3, preprint, arXiv:1406.4834v3 [math.OC], 2015.

[9] D. Davis and W. Yin, *Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions* http://arxiv.org/abs/1407.5210v3, preprint, arXiv:1407.5210v3 [math.OC], 2015.

[10] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[11] J. Douglas and H. H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[12] J. Eckstein and D. P. Bertsekas, *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293–318.

[13] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293–318, https://doi.org/10.1007/BF01581204.

[14] R. Erdahl, *Representability*, Int. J. Quantum Chem., 13 (1978), pp. 697–718.

[15] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[16] C. Garrod and J. K. Percus, *Reduction of the N-Particle Variational Problem*, J. Math. Phys., 5 (1964), p. 1756, https://doi.org/10.1063/1.1704098.

[17] G. Gidofalvi and D. A. Mazziotti, *Spin and symmetry adaptation of the variational two-electron reduced-density-matrix method*, Phys. Rev. A, 72 (2005), pp. 1–8, https://doi.org/10.1103/PhysRevA.72.052505.

[18] P.-L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979, https://doi.org/10.1137/0716071.

[19] Y. K. Liu, M. Christandl, and F. Verstraete, *Quantum computational complexity of the N-representability problem: QMA complete*, Phys. Rev. Lett., 98 (2007), pp. 1–4, https://doi.org/10.1103/PhysRevLett.98.110503.

[20] J. E. Mayer, *Electron correlation*, Phys. Rev., 100 (1955), p. 1579.

[21] D. A. Mazziotti, *Variational reduced-density-matrix method using three-particle n-representability conditions with application to many-electron molecules*, Phys. Rev. A, 74 (2006), p. 032501.

[22] D. A. Mazziotti, *Large-scale semidefinite programming for many-electron quantum mechanics*, Phys. Rev. Lett., 106 (2011), pp. 7–10, https://doi.org/10.1103/PhysRevLett.106.083001.

[23] M. Nakata, B. J. Braams, K. Fujisawa, M. Fukuda, J. K. Percus, M. Yamashita, and Z. Zhao, *Variational calculation of second-order reduced density matrices by strong N-representability conditions and an accurate semidefinite programming solver*, J. Chem. Phys., 128 (2008), https://doi.org/10.1063/1.2911696.

[24] M. Nakata, H. Nakatsuji, M. Ehara, M. Fukuda, K. Nakata, and K. Fujisawa, *Variational calculations of fermion second-order reduced density matrices be semidefinite programming algorithm*, J. Chem. Phys., 114 (2001), pp. 8282–8292, https://doi.org/10.1063/1.1360199.

[25] G. Pataki, *On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues*, Math. Oper. Res., 23 (1998), pp. 339–358.

[26] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, Berlin, 1998, https://doi.org/10.1007/978-3-642-02431-3.

[27] D. Sun and J. Sun, *Semismooth matrix-valued functions*, Math. Oper. Res., 27 (2002), pp. 150–169, https://doi.org/10.1287/moor.27.1.150.342.

[28] D. Sun, K.-C. Toh, and L. Yang, *A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints*, SIAM J. Optim., 25 (2015), pp. 882–915, https://doi.org/10.1137/140964357.

[29] M. J. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.

[30] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[31] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Math. Program. Comput., 2 (2010), pp. 203–230, https://doi.org/10.1007/s12532-010-0017-1.

[32] *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, H. Wolkowicz, R. Saigal, L. Vandenberghe, eds., Internat. Ser. Oper. Res. Management Sci., 27, Springer, New York, 2000.

[33] X. Xiao, Y. Li, Z. Wen, and L. Zhang, *A regularized semi-smooth Newton method with projection steps for composite convex programs*, J. Sci. Comput., (2016), pp. 1–26.

[34] L. Yang, D. Sun, and K. C. Toh, *SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints*, Math. Program. Comput., 7 (2015), pp. 331–366, https://doi.org/10.1007/s12532-015-0082-6.

[35] X.-y. Zhao, D. Sun, and K. C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2009), pp. 1737–1765.

[36] Z. Zhao, B. J. Braams, M. Fukuda, M. L. Overton, and J. K. Percus, *The reduced density matrix method for electronic structure calculations and the role of three-index representability conditions.*, J. Chem. Phys., 120 (2004), pp. 2095–104, https://doi.org/10.1063/1.1636721.

Downloaded 04/19/20 to 124.16.148.9. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.