Direct optimization of the atomic-orbital density matrix using the conjugate-gradient method with a multilevel preconditioner

Helena Larsen, Jeppe Olsen, and Poul Jørgensen[a)]
Department of Chemistry, University of Aarhus, Langelandsgade 140, 8000 Århus C, Denmark

Trygve Helgaker
Department of Chemistry, University of Oslo, P.O.B. 1033 Blindern, N-0315 Oslo, Norway

(Received 24 July 2001; accepted 11 September 2001)

Based on a recently proposed exponential parametrization of the one-electron atomic-orbital density matrix of a single-determinant wave function [Chem. Phys. Lett. **327**, 397 (2000)], we present an implementation of the direct optimization of the atomic-orbital density matrix as an alternative to the diagonalization of the Fock–Kohn–Sham matrix when solving the Roothaan–Hall self-consistent field equations. The optimization of the density matrix is carried out by the conjugate-gradient method with a multilevel nondiagonal preconditioner and is well suited to linear scaling. Although a diagonal preconditioner may be sufficient for minimal basis sets and large highest occupied molecular orbital–lowest unoccupied molecular orbital (HOMO–LUMO) gaps, a nondiagonal preconditioner is needed in more difficult cases—that is, for basis sets with polarization and diffuse functions and for systems with small HOMO–LUMO gaps. Redundancies of the exponential parametrization of the density matrix are handled by a projection technique, thereby avoiding singular equations in the optimization of the density matrix. © *2001 American Institute of Physics.* [DOI: 10.1063/1.1415083]

I. INTRODUCTION

In recent years, much work has focused on the development of *ab initio* and density-functional methods that are able to treat large systems such as proteins and other molecules of biological interest. In particular, efforts have been made to reduce the computational cost and memory requirements so as to achieve a cost that scales linearly with the size of the system for sufficiently large systems.

For Hartree–Fock and Kohn–Sham theories, important progress has been made. In particular, the complexity of the construction of the Fock and Kohn–Sham matrices has been reduced to linear by means of the fast multipole method (FMM) and the tree-code method for long-range contributions combined with efficient prescreening techniques.[1–6] Traditionally, Hartree–Fock and Kohn–Sham theories are formulated in the molecular-orbital (MO) representation, where the MOs are obtained by diagonalizing the Fock/Kohn–Sham matrix using the Roothaan–Hall algorithm. This diagonalization scales as the third power with the size of the system. To achieve linear scaling also in this step, several alternatives have been suggested.[7–17] Most of these methods are based either on the use of localized orbitals or on the direct optimization of the density matrix in the atomic-orbital (AO) basis without calculating the MOs. In the latter approach, linear scaling is obtained by exploiting the sparsity of the Fock/Kohn–Sham, overlap, and density matrices. So far, successful calculations have been reported for basis sets such as STO-3G, 3-21G, and 6-31G**.[11–14,18] In this paper, we present an AO density-based algorithm that may be used also for larger basis sets that contain diffuse functions.

For various reasons, the optimization of the AO density matrix is a nontrivial task. First, it is not straightforward to develop a good preconditioner for the Hessian in the AO basis. Simple diagonal preconditioners have successfully been used in the canonical MO basis, where the Hessian is diagonally dominant. However, in the AO basis the Hessian is in general not diagonally dominant and a diagonal preconditioner is, therefore, not effective and has only been successfully used with small basis sets. Furthermore, as the involved equations are singular in the AO basis, it is important to solve these equations in the subspace of nonsingular parameters.

In Refs. 19 and 20 we proposed an exponential parametrization of the AO density matrix of a one-determinant wave function. The parametrization is nonredundant and ensures that the AO density matrix fulfills the idempotency, rank, and symmetry conditions of the Hartree–Fock and Kohn–Sham density matrices. Using this parametrization, we demonstrate how the Roothaan–Hall diagonalization of the Fock/Kohn–Sham matrix may be replaced by an unconstrained optimization of the AO density matrix, thereby retaining the robust convergence of the Roothaan–Hall iterations—in particular when accelerated with the direct inversion in the iterative subspace (DIIS) algorithm[21,22]—while avoiding diagonalization. Furthermore, we demonstrate how this may be achieved by using the conjugate-gradient algorithm with a multilevel nondiagonal preconditioner. With this technique tight convergence criteria may be imposed, ensuring that molecular properties such as gradients and polarizabilities may be calculated.

[a)]Electronic mail: pou@chem.au.dk

II. ATOMIC-ORBITAL DENSITY MATRIX

A. Parametrization

For a one-determinant wave function, the density matrix **D** in the AO spin–orbital basis fulfills the symmetry, rank, and idempotency conditions

$$\mathbf{D}^T=\mathbf{D}, \tag{1}$$

$$\mathrm{Tr}\mathbf{DS}=N_e, \tag{2}$$

$$\mathbf{DSD}=\mathbf{D}, \tag{3}$$

where **S** is the overlap matrix of the atomic spin orbitals and N_e is the number of electrons. Conversely, any matrix **D** that fulfills the conditions in Eqs. (1)–(3) is a valid density matrix of a one-determinant wave function. It is, therefore, of interest to develop an unconstrained parametrization that allows us to generate all valid density matrices from a valid reference density matrix. In Refs. 19 and 20 we proposed the following parametrization

$$\mathbf{D}(\mathbf{X})=\exp(-\mathbf{XS})\mathbf{D}\exp(\mathbf{SX}), \tag{4}$$

where **X** is an anti-Hermitian matrix and **D** any matrix that satisfies Eqs. (1)–(3). In the following, we assume that the density matrix is real and thus restrict the rotation matrix **X** to be antisymmetric.

B. Redundancies and nonorthogonal projectors

The exponential parametrization of the AO density matrix in Eq. (4) contains redundancies. To study these we introduce the *nonorthogonal projection matrices*

$$\mathbf{P}=\mathbf{DS}, \tag{5}$$

$$\mathbf{Q}=\mathbf{I}-\mathbf{DS}, \tag{6}$$

which fulfill the standard projection relations

$$\mathbf{P}^2=\mathbf{P}, \tag{7}$$

$$\mathbf{Q}^2=\mathbf{Q}, \tag{8}$$

$$\mathbf{PQ}=\mathbf{0}. \tag{9}$$

However, the matrices **P** and **Q** are not symmetric. Rather, since **SP** and **SQ** are symmetric, **P** and **Q** satisfy the relations

$$\mathbf{P}^T\mathbf{S}=\mathbf{SP}, \tag{10}$$

$$\mathbf{Q}^T\mathbf{S}=\mathbf{SQ}. \tag{11}$$

Equations (7) and (10) are the defining relations of a nonorthogonal projector from which all other relations follow.

Using Eqs. (7) and (9), it is easy to show that transformations of the density matrix with $\mathbf{PXP}^T$ and $\mathbf{QXQ}^T$ do not change the density matrix[19]

$$\mathbf{D}(\mathbf{PXP}^T)=\mathbf{D}(\mathbf{0}), \tag{12}$$

$$\mathbf{D}(\mathbf{QXQ}^T)=\mathbf{D}(\mathbf{0}). \tag{13}$$

The analogy with MO theory, where rotations among the occupied orbitals or among the virtual orbitals do not change the density matrix, is obvious.

An arbitrary, antisymmetric rotation matrix **X** can be separated into redundant and nonredundant parts as

$$\mathbf{X}=(\mathbf{P}+\mathbf{Q})\mathbf{X}(\mathbf{P}^T+\mathbf{Q}^T)=\mathbf{X}^{nr}+\mathbf{X}^{rd}. \tag{14}$$

where

$$\mathbf{X}^{nr}=\mathbf{PXQ}^T+\mathbf{QXP}^T, \tag{15}$$

$$\mathbf{X}^{rd}=\mathbf{PXP}^T+\mathbf{QXQ}^T. \tag{16}$$

To first order the expansions of $\mathbf{D}(\mathbf{X})$ and $\mathbf{D}(\mathbf{X}^{nr})$ are identical, and we obtain

$$\left.\frac{\partial\mathbf{D}(\mathbf{X})}{\partial X_{\mu\nu}}\right|_{\mathbf{x}=0}=\left.\frac{\partial\mathbf{D}(\mathbf{X}^{nr})}{\partial X_{\mu\nu}}\right|_{\mathbf{x}=0}. \tag{17}$$

Second and higher derivatives, by contrast, are in general different for $\mathbf{D}(\mathbf{X})$ and $\mathbf{D}(\mathbf{X}^{nr})$.

For many purposes it is convenient to introduce a different notation for the projectors. We define the projector $\mathcal{P}$ as the direct product matrix

$$\mathcal{P}=\mathbf{P}\otimes\mathbf{Q}+\mathbf{Q}\otimes\mathbf{P}, \tag{18}$$

which projects onto the nonredundant parameter space

$$(\mathcal{P}\mathbf{X})_{\mu\nu}=\sum_{\alpha\beta}\mathcal{P}_{\mu\nu,\alpha\beta}X_{\alpha\beta}=(\mathbf{PXQ}^T+\mathbf{QXP}^T)_{\mu\nu}. \tag{19}$$

Note that **X** is considered a vector in $\mathcal{P}\mathbf{X}$ but a matrix in $\mathbf{PXQ}^T$ and $\mathbf{QXP}^T$. Whether **X** is to be considered as a vector or a matrix will in the following be clear from the context. If we introduce

$$\mathcal{Q}=\mathcal{I}-\mathcal{P}, \tag{20}$$

$$\mathcal{S}=\mathbf{S}\otimes\mathbf{S}, \tag{21}$$

a set of equations for $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{S}$ similar to Eqs. (7)–(11) are obtained. For example, Eq. (10) becomes

$$\mathcal{P}^T\mathcal{S}=\mathcal{S}\mathcal{P}, \tag{22}$$

as is easily proved by using the elementary relations for direct-product matrices

$$\begin{aligned}\mathcal{P}^T\mathcal{S}&=(\mathbf{P}^T\otimes\mathbf{Q}^T+\mathbf{Q}^T\otimes\mathbf{P}^T)\mathbf{S}\otimes\mathbf{S}\\&=(\mathbf{P}^T\mathbf{S})\otimes(\mathbf{Q}^T\mathbf{S})+(\mathbf{Q}^T\mathbf{S})\otimes(\mathbf{P}^T\mathbf{S})\\&=(\mathbf{SP})\otimes(\mathbf{SQ})+(\mathbf{SQ})\otimes(\mathbf{SP})=\mathcal{S}\mathcal{P}.\end{aligned} \tag{23}$$

C. Functions of the AO density matrix

Let us consider a function of the AO density matrix $f(\mathbf{D})$. Depending on the choice of parametrization for the density matrix, we have either of two functions

$$f(\mathbf{X})=f[\mathbf{D}(\mathbf{X})], \tag{24}$$

$$f^{nr}(\mathbf{X})=f[\mathbf{D}(\mathbf{X}^{nr})]. \tag{25}$$

Expansion about $\mathbf{X}=\mathbf{0}$ gives

$$f(\mathbf{X})=f(\mathbf{0})+\mathbf{X}^T\mathbf{F}^{[1]}+\tfrac{1}{2}\mathbf{X}^T\mathbf{F}^{[2]}\mathbf{X}+\cdots, \tag{26}$$

$$f^{nr}(\mathbf{X})=f(\mathbf{0})+\mathbf{X}^T\mathcal{P}^T\mathbf{F}^{[1]}+\tfrac{1}{2}\mathbf{X}^T\mathcal{P}^T\mathbf{F}^{[2]}\mathcal{P}\mathbf{X}+\cdots, \tag{27}$$

where $\mathbf{F}^{[1]}$ and $\mathbf{F}^{[2]}$ are the first and second derivatives of $f(\mathbf{X})$ with respect to $X_{\mu\nu}$. The gradient and the Hessian of Eq. (27) include the projection matrices. As the density matrix is unaffected by a rotation $\mathcal{Q}\mathbf{X}$, we have that

$$f(\mathcal{Q}\mathbf{X}) = f(\mathbf{0}), \tag{28}$$

which may be expanded to first order to give

$$\mathcal{Q}^T\mathbf{F}^{[1]} = 0, \tag{29}$$

or equivalently

$$\mathcal{P}^T\mathbf{F}^{[1]} = \mathbf{F}^{[1]}. \tag{30}$$

The first derivatives of $f(\mathbf{X})$ and $f^{nr}(\mathbf{X})$ are therefore identical and the two functions have the same stationary points. By contrast, the second and higher derivatives of $f(\mathbf{X})$ and $f^{nr}(\mathbf{X})$ are in general different. The Hessian of $f^{nr}(\mathbf{X})$ is highly singular since all vectors $\mathcal{Q}\mathbf{X}$ are eigenvectors with zero eigenvalue, as is easily shown by using Eq. (9)

$$\mathcal{P}^T\mathbf{F}^{[2]}\mathcal{P}\mathcal{Q}\mathbf{X} = 0. \tag{31}$$

The vector space $\mathcal{V}_0$ that contains the singularities $\mathcal{Q}\mathbf{X}$ is the null space of $\mathcal{P}^T\mathbf{F}^{[2]}\mathcal{P}$.

III. OPTIMIZATION OF DENSITY-DEPENDENT FUNCTIONS

In this section we consider some aspects related to the optimization of functions of the AO density matrix. In particular, we discuss how singularities of the Hessian affect the solutions and how constraints may be imposed to make the solutions unique. The discussion in the present section is general. Details of our implementation are described in later sections.

A. The Newton method

The Newton method is a natural point of departure for the optimization of the function $f^{nr}(\mathbf{X})$. Using the gradient and Hessian of Eq. (27), the Newton equations become

$$\mathcal{P}^T\mathbf{F}^{[2]}\mathcal{P}\mathbf{X}^\star = -\mathcal{P}^T\mathbf{F}^{[1]}, \tag{32}$$

where $\mathbf{X}^\star$ is the solution vector. As the Hessian is singular for vectors in $\mathcal{V}_0$, a necessary condition for the existence of solutions to the Newton equations is that the right-hand side is orthogonal to $\mathcal{V}_0$. To demonstrate orthogonality, we write a vector in $\mathcal{V}_0$ as $\mathcal{Q}\mathbf{V}$ and use Eqs. (9) and (29)

$$(\mathcal{Q}\mathbf{V})^T\mathcal{P}^T\mathbf{F}^{[1]} = \mathbf{V}^T\mathcal{Q}^T\mathcal{P}^T\mathbf{F}^{[1]} = \mathbf{0}. \tag{33}$$

The Newton equations, therefore, have a unique solution $\mathbf{X}_\perp^\star$ in the orthogonal complement to $\mathcal{V}_0$

$$(\mathcal{Q}\mathbf{V})^T\mathbf{X}_\perp^\star = \mathbf{0}. \tag{34}$$

This requirement is equivalent to

$$\mathcal{P}^T\mathbf{X}_\perp^\star = \mathbf{X}_\perp^\star, \tag{35}$$

where we have used Eq. (20) and the fact that $\mathbf{V}$ is an arbitrary vector.

A general solution $\mathbf{X}^\star$ to the Newton equations, Eq. (32), is obtained by adding to $\mathbf{X}_\perp^\star$ an arbitrary vector $\mathbf{X}_0$ belonging to $\mathcal{V}_0$

$$\mathbf{X}^\star = \mathbf{X}_\perp^\star + \mathbf{X}_0. \tag{36}$$

The $\mathbf{X}_0$ component may be fixed by supplementing Eq. (32) with additional requirements. A natural choice is to minimize the norm of the solution with respect to the standard norm

$$\min_{\mathbf{X}_0}\|\mathbf{X}_\perp^\star + \mathbf{X}_0\|_1^2 = \min_{\mathbf{X}_0}(\mathbf{X}_\perp^\star + \mathbf{X}_0)^T(\mathbf{X}_\perp^\star + \mathbf{X}_0). \tag{37}$$

Since $\mathbf{X}_\perp^\star$ is orthogonal to $\mathbf{X}_0$, the minimizer corresponds to $\mathbf{X}_0 = 0$. In this case, therefore, the condition [see Eq. (35)]

$$\mathcal{P}^T\mathbf{X}^\star = \mathbf{X}^\star, \tag{38}$$

fixes the solution to the singular Newton equations, Eq. (32).

An alternative solution $\mathbf{X}^\star$ is obtained by minimizing the solution norm with respect to the norm defined by $\mathcal{S}$

$$\min_{\mathbf{X}_0}\|\mathbf{X}_\perp^\star + \mathbf{X}_0\|_S^2 = \min_{\mathbf{X}_0}(\mathbf{X}_\perp^\star + \mathbf{X}_0)^T\mathcal{S}(\mathbf{X}_\perp^\star + \mathbf{X}_0). \tag{39}$$

Introducing an arbitrary variation in the $\mathcal{V}_0$ space

$$\boldsymbol{\delta}_0 = \mathcal{Q}\mathbf{V}, \tag{40}$$

the solution to this minimization problem fulfills the relation

$$\boldsymbol{\delta}_0^T\mathcal{S}\mathbf{X}^\star = \mathbf{0}. \tag{41}$$

Using Eqs. (11) and (40) and the fact that $\mathcal{S}$ is nonsingular, we obtain

$$\mathcal{P}\mathbf{X}^\star = \mathbf{X}^\star. \tag{42}$$

Thus, if the solution to the Newton equations is required to minimize the standard norm, we impose Eq. (38) on the solution, whereas minimization with respect to the $\mathcal{S}$ norm gives Eq. (42). It is not obvious which norm is the better choice.

When the Newton equations are solved by iterative methods such as the conjugate-gradient method (see Sec. III B), Eq. (38) is automatically fulfilled. Once this solution has been obtained, we may project with $\mathcal{P}$ to obtain

$$\overline{\mathbf{X}}^\star = \mathcal{P}\mathbf{X}^\star, \tag{43}$$

which is the solution that minimizes the $\mathcal{S}$ norm.

B. Conjugate-gradient method

Because of the large dimension of the Hessian, we must use an iterative method to solve the Newton equations. A popular iterative method is the preconditioned conjugate-gradient (PCG) method.[23,24] The PCG method requires only the storage of the current approximation to the solution and the previous search direction, but it is (for quadratic functions) equivalent to a subspace method with the current and all previous directions included. As will be discussed later, the convergence of the PCG method is critically dependent on the choice of preconditioner.

Let us briefly review the PCG method for solving a set of linear equations

$$\mathbf{A}\mathbf{X}^\star = \mathbf{b}. \tag{44}$$

In iteration k we know the current approximation $\mathbf{X}_k$ to $\mathbf{X}^\star$, the current search direction $\mathbf{d}_k$, and the current residual

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{X}_k. \tag{45}$$

An improved approximation to $\mathbf{X}^\star$ is obtained by using the algorithm

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha_k\mathbf{d}_k, \tag{46}$$

$$\alpha_k = \frac{\mathbf{r}_k^T \overline{\mathbf{A}}^{-1} \mathbf{r}_k}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_\mathbf{k}}, \tag{47}$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{d}_k, \tag{48}$$

$$\mathbf{d}_{k+1} = \overline{\mathbf{A}}^{-1} \mathbf{r}_k + \beta_k \mathbf{d}_k, \tag{49}$$

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \overline{\mathbf{A}}^{-1} \mathbf{r}_{k+1}}{\mathbf{r}_k^T \overline{\mathbf{A}}^{-1} \mathbf{r}_k}, \tag{50}$$

where $\overline{\mathbf{A}}$ is the preconditioner. The iterations are initialized by a guess $\mathbf{X}_0$ to the solution, and by setting the initial direction equal to $\overline{\mathbf{A}}^{-1}\mathbf{r}_0$.

We here consider the PCG method for solving the Newton equations in Eq. (32). As the PCG method converges in a number of iterations that is equal to or less than the dimension of the Hessian, it is advantageous to use the projected rather than the unprojected Newton equations. Using the gradient and the Hessian of Eq. (27), we obtain the residual

$$\mathbf{r}_k = -\mathcal{P}^T \mathbf{F}^{[1]} - \mathcal{P}^T \mathbf{F}^{[2]} \mathcal{P} \mathbf{X}_k = \mathcal{P}^T \mathbf{r}_k, \tag{51}$$

which emphasizes that the residual automatically fulfills projection with $\mathcal{P}^T$. If the inverse preconditioner $\overline{\mathbf{A}}^{-1}$ is the unit matrix or more generally satisfies the equation $\mathcal{P}^T \overline{\mathbf{A}}^{-1} \mathcal{P}^T \mathbf{r}_{k+1} = \overline{\mathbf{A}}^{-1} \mathcal{P}^T \mathbf{r}_{k+1}$, all directions are unaffected by projections with $\mathcal{P}^T$. Moreover, if the initial approximation fulfills $\mathcal{P}^T \mathbf{X}_0 = \mathbf{X}_0$, we have at each iteration an approximate solution that satisfies the equation

$$\mathcal{P}^T \mathbf{X}_k = \mathbf{X}_k. \tag{52}$$

The PCG method then gives directly the solution that fulfills Eq. (38) and thereby minimizes the standard norm. Upon solution, we can project with $\mathcal{P}$ to obtain the solution that minimizes the $\mathcal{S}$ norm. Alternatively, we may project the direction with $\mathcal{P}$ in each iteration. However, a closer analysis reveals that this procedure leads to the same iteration sequence as generated by projecting with $\mathcal{P}^T$ and thus does not provide any advantages.

IV. SOLUTION OF THE HARTREE–FOCK EQUATIONS BY DIRECT ENERGY MINIMIZATION

A. Reformulation of the Roothaan–Hall scheme

The Hartree–Fock energy can be optimized efficiently by carrying out a sequence of Roothaan–Hall iterations, in particular, in conjunction with the direct inversion in the iterative subspace (DIIS) algorithm.[21,22] The Roothaan–Hall equations were originally expressed as a pseudoeigenvalue problem for the canonical MOs: From the MOs an improved AO density matrix is constructed, a new Fock matrix is generated from this density matrix and then diagonalized. However, since the cost of the diagonalization scales cubically with the size of the system, the determination of canonical MOs must be abandoned to achieve linear scaling. Instead, the AO density matrix is optimized directly.

In Refs. 19 and 20 it is described how the AO density matrix that represents the solution to the Roothaan–Hall equations can be obtained by minimizing the sum of the energies of the occupied canonical orbitals

$$E_\epsilon = \sum_I \epsilon_I = \mathrm{Tr}\mathbf{DF}. \tag{53}$$

Parametrizing the density matrix in terms of the antisymmetric matrix $\mathbf{X}$ using Eqs. (4) and (15)

$$E_\epsilon(\mathbf{X}) = \mathrm{Tr}\exp(-\mathbf{X}^{nr}\mathbf{S})\mathbf{D}\exp(\mathbf{S}\mathbf{X}^{nr})\mathbf{F}, \tag{54}$$

we may generate the AO density matrix of the Roothaan–Hall equations directly by minimizing $E_\epsilon(\mathbf{X})$. The elements $X_{\mu\nu}$ with $\mu > \nu$ are the unconstrained variational parameters and the redundancies are controlled by projections.

B. Newton method

To optimize $\mathbf{X}$ and determine the AO density matrix, we may use the Newton method. The Newton equations are obtained by expanding the energy function $E_\epsilon(\mathbf{X})$ to second order and setting the derivative with respect to $X_{\mu\nu}$ equal to zero

$$\mathcal{P}^T([\mathbf{F}\mathcal{P}(\mathbf{X})\mathbf{SDS}]^A) - \mathcal{P}^T([[\mathbf{FDS}]^S \mathcal{P}(\mathbf{X})\mathbf{S}]^A) = \mathcal{P}^T([\mathbf{FDS}]^A). \tag{55}$$

We have here introduced the notation

$$[\mathbf{M}]^A = \tfrac{1}{2}(\mathbf{M} - \mathbf{M}^T), \tag{56}$$

$$[\mathbf{M}]^S = \tfrac{1}{2}(\mathbf{M} + \mathbf{M}^T), \tag{57}$$

for the antisymmetric and symmetric parts of a matrix, respectively. The notation $\mathcal{P}(\mathbf{X})$ is used for the matrix obtained by multiplying $\mathcal{P}$ and $\mathbf{X}$. Equation (55) constitutes a set of linear equations of the form

$$\mathbf{A}(\mathbf{D})\mathbf{X} = \mathbf{G}(\mathbf{D}), \tag{58}$$

where $\mathbf{G}(\mathbf{D})$ is the negative gradient and $\mathbf{A}(\mathbf{D})$ is the Hessian.

It is instructive to express Eq. (58) in the unoptimized canonical MO basis—that is, in a basis in which the Fock matrix has diagonal occupied–occupied and virtual–virtual blocks but nonzero occupied–virtual blocks. In this basis the elements of the Hessian are given by (using A and B for virtual orbitals and I and J for occupied orbitals)

$$A^{\mathrm{MO}}_{AIBJ} = (\mathbf{C}^T \mathbf{A}(\mathbf{D}) \mathbf{C})_{AIBJ} = \delta_{AB}\delta_{IJ}(\epsilon_A - \epsilon_I), \tag{59}$$

where $\mathbf{C}$ is the MO–AO transformation matrix[19] and the virtual–occupied elements of Eq. (58) become

$$(\epsilon_A - \epsilon_I) X_{AI} = F_{AI}. \tag{60}$$

In standard MO theory the linear equations, Eq. (58), are therefore, trivially solved. In the AO basis the Hessian is neither diagonal nor diagonally dominant. The linear equations must then be solved by iterative methods such as the PCG method. In the next section we discuss how Eq. (58) may be solved in the AO basis using preconditioning.

V. PRECONDITIONING

A. Multilevel preconditioning

The convergence of the conjugate-gradient method depends critically on the condition number of the Hessian matrix $\mathbf{A}$—that is, on the absolute value of the ratio of the

numerically largest eigenvalue to the numerically smallest eigenvalue of **A**. A large condition number corresponds to a large range of eigenvalues and results in slow convergence. Conversely, a condition number close to one gives fast convergence. To accelerate convergence we may introduce a preconditioner, $\overline{\mathbf{A}}$, noting that the rate of convergence then depends on the condition number of $\overline{\mathbf{A}}^{-1}\mathbf{A}$. Thus, convergence is significantly improved by choosing a preconditioner such that

$$\overline{\mathbf{A}}^{-1}\mathbf{A}\approx\mathbf{1}. \tag{61}$$

However, the cost of introducing the preconditioner is that, in each iteration, we must solve the linear equations

$$\overline{\mathbf{A}}\tilde{\mathbf{r}}=\mathbf{r}, \tag{62}$$

to obtain a new direction—see, for example, Eqs. (46)–(50) for the PCG method. In addition to satisfying Eq. (61), a useful preconditioner should, therefore, have a structure such that the solution to Eq. (62) is easy to obtain.

In electronic-structure calculations carried out in the MO basis, **A** is often diagonally dominant. A good choice of preconditioner is then simply the diagonal or the dominant part of the diagonal of **A**. An extreme example of diagonal dominance is provided by the Hessian in the canonical MO basis [Eq. (58)], which is diagonal according to Eq. (59). By contrast, in the AO basis the Hessian is in general neither diagonal nor diagonally dominant and a diagonal preconditioner is, therefore, a poor choice. However, for small basis sets where the MO–AO transformation matrix **C** is close to a unit matrix we see from Eq. (59) that the AO Hessian is nearly diagonal. For extended basis sets, the matrix **C** contains very large elements and is not diagonal dominant. In this case the AO Hessian is not diagonal dominant with both large and small eigenvalues, leading to a large condition number. Increasing the size of the basis therefore increases the number of iterations needed to solve Eq. (58) by the conjugate gradient method.

In the following, we shall find it convenient to employ a hierarchy $\overline{\mathbf{A}},\bar{\bar{\mathbf{A}}},\overline{\overline{\mathbf{A}}},\ldots$ of preconditioners. While $\overline{\mathbf{A}}$ is the preconditioner for **A**, the matrix $\bar{\bar{\mathbf{A}}}$ is the preconditioner for the equations involving $\overline{\mathbf{A}}$, leading to a new set of linear equations involving $\bar{\bar{\mathbf{A}}}$ with preconditioner $\overline{\overline{\mathbf{A}}}$, and so on. The higher a preconditioner is placed in the hierarchy $\overline{\mathbf{A}},\bar{\bar{\mathbf{A}}},\ldots$, the easier it is to solve the equations involving the preconditioner and the poorer is the agreement between **A** and the preconditioner. The advantage of multilevel preconditioning is that we reduce the number of times the preconditioners that are low in the hierarchy are multiplied with vectors.

B. Choice of preconditioners

According to Eq. (55) the Hessian is constructed from the matrices **F**, **S**, and **D**. In our experience, it is better to introduce preconditioners by approximating these matrices than to modify the Hessian directly (e.g., by setting some of its elements equal to zero). In practice, we use distance restrictions to approximate **F**, **S**, and **D**, setting elements whose AOs are separated by more than a preset cut-off threshold equal to zero, with tighter thresholds imposed for higher preconditioners. For the highest preconditioner, the diagonal of the exact Hessian is used.

In constructing the Hessian and its products with vectors, we also need to set up the projection matrices **P** and **Q**. These matrices have hitherto been defined in terms of the exact **D** and **S**, which were used to calculate the Hessian in Eq. (55). When preconditioners with approximate **D** and **S** are introduced, **P** and **Q** may also be modified, thereby introducing different projectors at each level of the preconditioning.

Without such modified projection matrices, the operation count for the matrix-vector product at the higher levels is not significantly reduced compared with the lower levels. On the other hand, the use of modified projection matrices leads to different nonredundant spaces at the different levels. Thus, the solution vector obtained at a given level of preconditioning will not contain all components of the solution at lower levels. In the following, we shall first consider the case where the same projection matrices are used for all preconditioners. Next, we discuss the modifications that are necessary when different projection matrices are used at different levels.

VI. MULTILEVEL PCG METHOD

In Fig. 1 we have illustrated the multilevel preconditioned conjugate-gradient (MPCG) method for the optimization of the Hartree–Fock density matrix and energy. The optimization consists of a sequence of outer Fock iterations at Level 1. At this level each iteration is initiated by the construction of a Fock matrix from a given AO density matrix. The inner iterations at Level 2 are then invoked, returning an AO density matrix that is optimal for the given Fock matrix. This density matrix is identical to the one obtained in the traditional orbital-based Roothaan–Hall scheme.

At Level 2 in Fig. 1, the density matrix is determined by minimizing Eq. (54). This nonlinear optimization is carried out by the PCG method, involving the calculation of the gradient of Eq. (58) and preconditioning. In general, the preconditioner is nondiagonal and requires the solution of a linear set of equations. These equations are solved by the PCG method at Level 3. At Level 3 a coarser preconditioner is used and the resulting linear set of equations is solved at Level 4. At Level 4 we terminate the MPCG hierarchy by using a diagonal preconditioner.

An important special case of MPCG arises when the exact Hessian is used as preconditioner at Level 2. In this case, the iterations at Level 2 are essentially the Newton iterations, while the linear equations at Level 3 become the Newton equations.

In the following, we discuss our implementation of the MPCG algorithm for the solution of the Hartree–Fock equations, beginning with the PCG minimization at Level 2 in Sec. VI A. Next, we consider the linear equations at Level 3 and 4 in Sec. VI B. Finally, in Sec. VI C we discuss the use of level-dependent projection matrices.

Level 1:

Solve: $\min_{\mathbf{X}}[E(\mathbf{D}(\mathbf{X}))]$

Iteration i:

Evaluate $\mathbf{F}_i$ using the density $\mathbf{D}_i$

$\downarrow \mathbf{F}_i \qquad \uparrow \mathbf{D}(\mathbf{X}_i)$

Level 2:

Solve: $\min_{\mathbf{X}}[E_\varepsilon(\mathbf{D}(\mathbf{X}))]$

Iteration n:

Evaluate the gradient $\mathbf{G}_n$

Determine $\tilde{\mathbf{G}}_n$ through calls to Level 3 and higher

Determine the direction $\mathbf{H}_n$

Determine the length of the step α_n as $\min_\alpha[E_\varepsilon(\alpha\mathbf{H}_n)]$

$\mathbf{X}_n = \alpha_n \mathbf{H}_n$

Update the density: $\mathbf{D}_{i+1,n} = \mathbf{D}(\mathbf{X}_{in})$

$\downarrow \mathbf{G}_n \qquad \uparrow \tilde{\mathbf{G}}_n$

Level 3:

Solve: $\bar{\mathbf{A}}\tilde{\mathbf{G}}_n = \mathbf{G}_n$

Iteration m:

$\tilde{\mathbf{G}}_{nm+1} = \tilde{\mathbf{G}}_{nm} + \alpha_m \mathbf{d}_m, \quad \alpha_m = \frac{\mathbf{r}_m^T \tilde{\mathbf{r}}_m}{\mathbf{d}_m^T \bar{\mathbf{A}} \mathbf{d}_m}$

$\mathbf{r}_{m+1} = \mathbf{r}_m - \alpha_m \bar{\mathbf{A}}\mathbf{d}_m$

$\mathbf{d}_{m+1} = \bar{\bar{\mathbf{A}}}^{-1}\mathbf{r}_{m+1} + \beta_m \mathbf{d}_m, \quad \beta_m = \frac{\mathbf{r}_m^T \tilde{\mathbf{r}}_{m+1}}{\mathbf{r}_m^T \tilde{\mathbf{r}}_m}$

$\downarrow \mathbf{r}_{m+1} \qquad \uparrow \tilde{\mathbf{r}}_{m+1}$

Level 4:

Solve: $\bar{\bar{\mathbf{A}}}\tilde{\mathbf{r}}_{m+1} = \mathbf{r}_{m+1}$

$\vdots$

FIG. 1. The multilevel preconditioned conjugate-gradient method for the solution of linear equations.

A. PCG method for nonlinear minimization of E_ϵ at Level 2

At Level 2 the Fock matrix **F** and the overlap matrix **S** are given as input and the optimal density matrix for E_ϵ is determined, starting from an initial density matrix $\mathbf{D}_0$. The conjugate-gradient algorithm is initialized by calculating the gradient

$$\mathbf{G}_0 = 4(\mathbf{F}\mathbf{D}_0\mathbf{S} - \mathbf{S}\mathbf{D}_0\mathbf{F}). \tag{63}$$

Next, a linear set of equations is solved

$$\overline{\mathbf{A}}\tilde{\mathbf{G}}_0 = \mathbf{G}_0, \quad \text{where} \quad \mathcal{P}^T(\tilde{\mathbf{G}}_0) = \tilde{\mathbf{G}}_0, \tag{64}$$

to give the initial search direction

$$\mathbf{H}_0 = \tilde{\mathbf{G}}_0. \tag{65}$$

Finally, the step length α_0 is determined by minimizing $E_\epsilon(\alpha\mathbf{H}_0)$ using a one-dimensional line search, leading to the step

$$\mathbf{X}_0 = \alpha_0 \mathbf{H}_0. \tag{66}$$

The iteration procedure is then continued as

$$\mathbf{D}_{n+1} = \exp(-\mathbf{X}_n\mathbf{S})\mathbf{D}_n \exp(\mathbf{S}\mathbf{X}_n), \tag{67}$$

$$\mathbf{G}_{n+1} = 4(\mathbf{F}\mathbf{D}_{n+1}\mathbf{S} - \mathbf{S}\mathbf{D}_{n+1}\mathbf{F}), \tag{68}$$

$$\overline{\mathbf{A}}\tilde{\mathbf{G}}_{n+1} = \mathbf{G}_{n+1}, \quad \mathcal{P}^T(\tilde{\mathbf{G}}_{n+1}) = \tilde{\mathbf{G}}_{n+1}, \tag{69}$$

$$\beta_n = \frac{(\mathbf{G}_{n+1} - \mathbf{G}_n)^T \tilde{\mathbf{G}}_{n+1}}{\mathbf{G}_n \tilde{\mathbf{G}}_n}, \tag{70}$$

$$\mathbf{H}_{n+1} = \tilde{\mathbf{G}}_{n+1} + \beta_n \mathbf{H}_n, \tag{71}$$

$$\mathbf{X}_{n+1} = \alpha_{n+1}\mathbf{H}_{n+1}, \tag{72}$$

where α_{n+1} is found using a one-dimensional line search. Equation (70) is equivalent to Eq. (50) except that $\mathbf{G}_n^T\tilde{\mathbf{G}}_{n+1}$ vanishes for a quadratic function. Note that $\tilde{\mathbf{G}}_n$, $\mathbf{H}_{n+1}$, and $\mathbf{X}_n$ fulfill $\mathbf{P}^T(\tilde{\mathbf{G}}_n) = \tilde{\mathbf{G}}_n$, $\mathcal{P}^T(\mathbf{H}_{n+1}) = \mathbf{H}_{n+1}$, and $\mathbf{P}^T(\mathbf{X}_n) = \mathbf{X}_n$, respectively. Equations (64) and (69), which give $\tilde{\mathbf{G}}_n$, are solved iteratively using the PCG method.

B. PCG method for linear equations at Levels 3 and 4

At Levels 3 and 4 the PCG method is used to solve sets of linear equations. At Level 3 we solve

$$\overline{\mathbf{A}}\tilde{\mathbf{G}}_n = \mathbf{G}_n, \tag{73}$$

where $\mathbf{G}_n$ satisfies

$$\mathcal{P}^T(\mathbf{G}_n) = \mathbf{G}_n, \tag{74}$$

and we impose the condition

$$\mathcal{P}^T(\tilde{\mathbf{G}}_n) = \tilde{\mathbf{G}}_n. \tag{75}$$

The matrix $\overline{\mathbf{A}}$ is defined by its action on a trial vector

$$\overline{\mathbf{A}}\mathbf{x} = \mathcal{P}^T([\overline{\mathbf{F}}\mathcal{P}(\mathbf{x})\overline{\mathbf{S}\mathbf{D}\mathbf{S}}]^A) + \mathcal{P}^T([[\overline{\mathbf{F}\mathbf{D}\mathbf{S}}]^S\mathcal{P}(\mathbf{x})\overline{\mathbf{S}}]^A). \tag{76}$$

Note carefully that we here employ the full projection matrices **P** and **Q** constructed from the exact matrices **D** and **S**, even though we otherwise construct the Hessian from the approximate matrices $\overline{\mathbf{F}}$, $\overline{\mathbf{D}}$, and $\overline{\mathbf{S}}$, rather than from **F**, **D**, and **S**. The use of full projection matrices ensures that the solution spaces are identical at Levels 2 and 3.

A nonvanishing initial guess $\tilde{\mathbf{G}}_{n0}$ is found as

$$\tilde{\mathbf{G}}_{n0} = \mathcal{P}^T(\overline{\mathbf{A}}_{\text{dia}}^{-1}\mathbf{G}_n). \tag{77}$$

Here, the first and second indices of $\tilde{\mathbf{G}}_{n0}$ refer to the iteration numbers at Levels 2 and 3, respectively, and $\mathbf{A}_{\text{dia}}$ is the diagonal of the exact Hessian with elements given by

$$A_{\mu\nu,\mu\nu} = -4[F_{\mu\mu}(\mathbf{S}\mathbf{D}\mathbf{S})_{\nu\nu} + (\mathbf{S}\mathbf{D}\mathbf{S})_{\mu\mu}F_{\nu\nu} + 2F_{\mu\nu}(\mathbf{S}\mathbf{D}\mathbf{S})_{\mu\nu} - (\mathbf{F}\mathbf{D}\mathbf{S})_{\mu\mu}S_{\nu\nu} - S_{\mu\mu}(\mathbf{F}\mathbf{D}\mathbf{S})_{\nu\nu} + 4(\mathbf{F}\mathbf{D}\mathbf{S})_{\mu\nu}S_{\mu\nu}]. \tag{78}$$

The residual $\mathbf{r}_{n0}$ is given by

$$\mathbf{r}_{n0} = \mathbf{G}_n - \overline{\mathbf{A}}\tilde{\mathbf{G}}_{n0}. \tag{79}$$

To obtain a preconditioned direction, we next solve the Level 4 equations

Level 1:

Solve: $\min_{\mathbf{X}}[E(\mathbf{D}(\mathbf{X}))]$

Iteration i:

Evaluate $\mathbf{F}_i$ using the density $\mathbf{D}_i$

Update the density: $\mathbf{D}_{i+1}=\mathbf{D}_i(\mathbf{X}_i)$

$\downarrow \mathbf{F}_i \qquad \uparrow \mathbf{D}(\mathbf{X}_{i+1})$

Level 2:

Solve: $\min_{\mathbf{X}}[E_\varepsilon(\mathbf{D}(\mathbf{X}))]$

Iteration n:

Evaluate the gradient $\mathbf{G}_n$

Determine $\tilde{\mathbf{G}}_n$ through calls to Level 3 and higher

Determine the direction $\mathbf{H}_n$

Determine the length of the step α_n as $\min_\alpha[E_\varepsilon(\alpha\mathbf{H}_n)]$

$\mathbf{X}_n=\alpha_n\mathbf{H}_n$

Update the density matrix: $\mathbf{D}_{i+1,n}=\mathbf{D}(\mathbf{X}_{in})$

$\downarrow \mathbf{G}_n \qquad \uparrow \mathcal{P}^T(\tilde{\mathbf{G}}_n+\bar{\mathcal{Q}}^T\mathbf{G}_n)$

Level 3:

Solve: $\bar{\mathbf{A}}\tilde{\mathbf{G}}_n=\bar{\mathcal{P}}^T rm\mathbf{G}_n$

Iteration m:

$\tilde{\mathbf{G}}_{nm+1}=\tilde{\mathbf{G}}_{nm}+\alpha_m\mathbf{d}_m, \quad \alpha_m=\frac{\mathbf{r}_m^T\tilde{\mathbf{r}}_m}{\mathbf{d}_m^T rm\mathbf{A}\mathbf{d}_m}$

$\mathbf{r}_{m+1}=\mathbf{r}_m-\alpha_m\bar{\mathbf{A}}\mathbf{d}_m$

$\mathbf{d}_{m+1}=\bar{\mathbf{A}}^{-1}\mathbf{r}_m+\beta\mathbf{d}_m, \quad \beta_m=\frac{\mathbf{r}_{m+1}^T\tilde{\mathbf{r}}_{m+1}}{\mathbf{r}_m^T\tilde{\mathbf{r}}_m}$

$\downarrow \mathbf{r}_{m+1} \qquad \uparrow \bar{\mathcal{P}}^T(\tilde{\mathbf{r}}_{m+1}+\bar{\mathcal{Q}}^T\mathbf{r}_{m+1})$

Level 4:

Solve : $\bar{\bar{\mathbf{A}}}\tilde{\mathbf{r}}_{m+1}=\bar{\bar{\mathcal{P}}}^T\mathbf{r}_{m+1}$

$\vdots$

FIG. 2. The multilevel preconditioned conjugate-gradient method for the solution of linear equations using level-dependent projection matrices.

$$\bar{\bar{\mathbf{A}}}\tilde{\mathbf{r}}_{n0}=\mathcal{P}^T(\mathbf{r}_{n0}). \tag{80}$$

Although $\mathbf{r}_{n0}=\mathcal{P}^T(\mathbf{r}_{n0})$ by construction, the projection has been included for numerical stability. The initial Level 3 direction is obtained as

$$\mathbf{d}_{n0}=\tilde{\mathbf{r}}_{n0}. \tag{81}$$

The PCG method as described by Eqs. (46)–(50) may now straightforwardly be applied. Note that in iteration k at Level 3, the relations $\mathcal{P}^T(\mathbf{r}_{nk})=\mathbf{r}_{nk}$, $\mathcal{P}^T(\mathbf{d}_{nk})=\mathbf{d}_{nk}$, and $\mathcal{P}^T(\tilde{\mathbf{G}}_{nk})=\tilde{\mathbf{G}}_{nk}$ are automatically fulfilled.

At Level 4 we employ a similar method to solve Eq. (80), terminating the preconditioning hierarchy by using the diagonal of the exact Hessian, Eq. (78), as the preconditioner.

C. Use of level-dependent projections matrices

In the above formulation of the MPCG method, we used the same projection matrices at Levels 2–4, ensuring that a common nonredundant space is used at all levels. However, the use of a common projection matrix makes the linear transformations $\overline{\mathbf{A}}\tilde{\mathbf{r}}$ and $\bar{\bar{\mathbf{A}}}\tilde{\mathbf{r}}$ nearly as time-consuming as the linear transformation $\mathbf{A}\tilde{\mathbf{r}}$, thereby reducing the advantages of the proposed multilevel scheme. We now consider the use of level-dependent projection matrices.

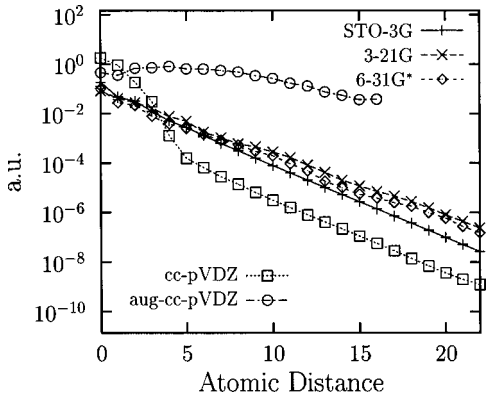

FIG. 3. The natural logarithm of the maximum of the density matrix elements plotted as a function of the distance between the atoms that are interacting for $C_{24}H_{50}$ for the STO-3G, 3-21G, 6-31G*, and cc-pVDZ basis sets, and $C_{18}H_{38}$ for the aug-cc-pVDZ basis set.

In Fig. 2 we have illustrated the MPCG method with level-dependent projection matrices. Let us consider Level 3. We introduce the modified projection matrix $\overline{\mathcal{P}}$, defined as

$$\overline{\mathcal{P}}=\overline{\mathbf{P}}\otimes\overline{\mathbf{Q}}+\overline{\mathbf{Q}}\otimes\overline{\mathbf{P}}, \tag{82}$$

where

$$\overline{\mathbf{P}}=\overline{\mathbf{D}\mathbf{S}}, \tag{83}$$

$$\overline{\mathbf{Q}}=\mathbf{1}-\overline{\mathbf{P}}. \tag{84}$$

When approximations are introduced in $\mathbf{S}$ and $\mathbf{D}$, the rank and idempotency relations Eqs. (2) and (3) are only approximately satisfied, implying that $\overline{\mathbf{P}}$ and $\overline{\mathbf{Q}}$ are not true projectors (unlike $\mathbf{P}$ and $\mathbf{Q}$). Nevertheless, we shall in the following assume that $\overline{\mathbf{P}}$ and $\overline{\mathbf{Q}}$ are true projectors, realizing some numerical noise is introduced in this manner. Since we do not need to solve the linear equations at Levels 3 and 4

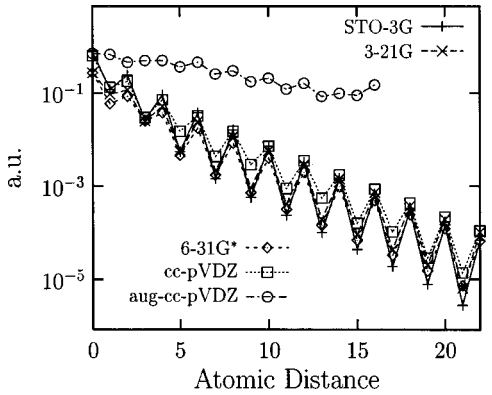

FIG. 4. The natural logarithm of the maximum of the density matrix elements plotted as a function of the distance between the atoms that are interacting for $C_{24}H_{26}$ for the STO-3G, 3-21G, 6-31G*, and cc-pVDZ basis sets, and $C_{18}H_{20}$ for the aug-cc-pVDZ basis set.

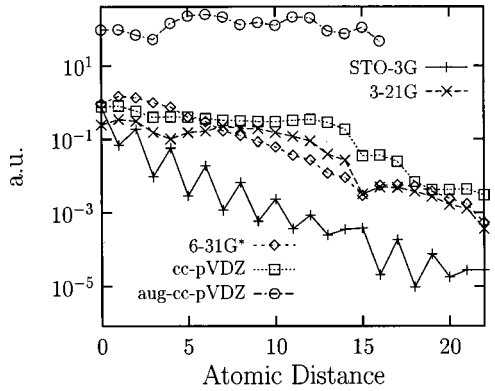

FIG. 5. The natural logarithm of the maximum of the density matrix elements plotted as a function of the distance between the atoms that are interacting for $C_{24}H_2$ for the STO-3G, 3-21G, 6-31G*, and cc-pVDZ basis sets, and $C_{18}H_2$ for the aug-cc-pVDZ basis set.

exactly, we have so far not encountered any problems arising from the approximate nature of the projectors.

Assuming that $\overline{\mathbf{P}}$ and $\overline{\mathbf{Q}}$ and thus $\overline{\mathcal{P}}$ are bona fide projectors, we shall now discuss the complications that arise from the fact that $\overline{\mathcal{P}}$ is different from $\mathcal{P}$. In the definition of $\mathbf{A}$ in Eq. (76), we used the full projection matrix $\mathcal{P}$. With the use of the simplified projector $\overline{\mathcal{P}}$, the corresponding expression for $\overline{\mathbf{A}}$ becomes

$$\overline{\mathbf{A}}\mathbf{x}=\overline{\mathcal{P}}^T([\overline{\mathbf{F}}\overline{\mathcal{P}}(\mathbf{x})\overline{\mathbf{SDS}}]^A)+\overline{\mathcal{P}}^T([[\overline{\mathbf{FDS}}]^S\overline{\mathcal{P}}(\mathbf{x})\overline{\mathbf{S}}]^A). \quad (85)$$

With this choice of preconditioner, we can solve linear equations for right-hand sides that are orthogonal to $\overline{\mathcal{Q}}\mathbf{V}$. However, the right-hand side at Level 3 ($\mathbf{G}_n$) is orthogonal to $\mathcal{Q}\mathbf{V}$ rather than $\overline{\mathcal{Q}}\mathbf{V}$. It may therefore be decomposed as

$$\mathbf{G}_n=\overline{\mathcal{P}}^T(\mathbf{G}_n)+\overline{\mathcal{Q}}^T(\mathbf{G}_n), \quad (86)$$

where the first component is large and the second small. We now solve the preconditioned problem with right-hand side $\overline{\mathcal{P}}^T(\mathbf{G}_n)$, giving a solution that fulfills the relation

$$\overline{\mathcal{P}}^T(\tilde{\mathbf{G}}_n)=\tilde{\mathbf{G}}_n, \quad (87)$$

but leave $\overline{\mathcal{Q}}^T(\mathbf{G}_n)$ unchanged, corresponding to the use of a unit preconditioner. The preconditioned residual then becomes $\tilde{\mathbf{G}}_n+\overline{\mathcal{Q}}^T(\mathbf{G}_n)$. This vector has a nonzero component in $\mathcal{V}_0$, which we remove by projecting $\tilde{\mathbf{G}}_n+\overline{\mathcal{Q}}^T(\mathbf{G}_n)$ with $\mathcal{P}^T$.

Similar considerations hold at Level 4. The use of different projectors at different levels means that a small part of the residual has been preconditioned with only a unit matrix. However, as this component by construction is small, we do not expect this to affect the convergence.

TABLE I. Number of surviving elements as a function of the distance between the interacting atoms using a neglect threshold of 10^{-3} for $C_{24}H_{50}$ using various basis sets.

Basis set	1	2	3	4	5	6	7	8
STO-3G	1910	1696	1402	624	430	300	102	
3-21G	6558	4528	3250	2020	704	560	208	12
6-31G*	9014	5728	3866	2152	838	308	72	
cc-pVDZ	22 242	19 328	10 340	2768	38			
aug-cc-pVDZ	32 094	26 716	20 682	18 028	14 964	13 036	10 068	8060

VII. DIIS ACCELERATION SCHEME

From MO theory it is well known that the convergence of the Roothaan–Hall self-consistent field iterations may be accelerated and made more robust by using the direct inversion in the iterative subspace (DIIS) algorithm. We here discuss the DIIS acceleration scheme in the context of an AO density-based optimization.[21,22]

The idea of the DIIS scheme is to utilize the information available from the preceding iterations by constructing an averaged, effective AO Fock matrix

$$(\mathbf{F}_n)_{\mathrm{av}}=\sum_{i=1}^{n}\omega_i\mathbf{F}_i. \quad (88)$$

This effective Fock matrix is then used in place of $\mathbf{F}_n$ to generate an improved set of MOs and thus an improved AO density matrix. In this manner, the Roothaan–Hall iterations are stabilized and oscillations avoided. The weights of the averaged Fock matrix are found by minimizing the norm of the averaged error vector

$$(\mathbf{e}_n)_{\mathrm{av}}=\sum_{i=1}^{n}\omega_i\mathbf{e}_i, \quad (89)$$

where the error vector equals the gradient

$$\mathbf{e}_i=4(\mathbf{SD}_i\mathbf{F}_i-\mathbf{F}_i\mathbf{D}_i\mathbf{S}), \quad (90)$$

subject to the constraint

$$\sum_{i=1}^{n}\omega_i=1. \quad (91)$$

This is achieved by minimizing the Lagrangian

$$L=\sum_{i,j=1}^{n}\omega_i\langle\mathbf{e}_i|\mathbf{e}_j\rangle\omega_j-\lambda\left(\sum_{i=1}^{n}\omega_i-1\right), \quad (92)$$

with respect to the weights ω_i and the undetermined multiplier λ, leading to a linear set of equations for the weights. For this procedure to work, we must store the previous Fock matrices and error vectors $\mathbf{F}_n$ and $\mathbf{e}_n$. However, since in practice the oldest matrices carry little information of interest, only a few Fock matrices and error vectors are stored.

The DIIS algorithm must be modified slightly to be useful in density-matrix optimizations since, in order to minimize $E_\epsilon(\mathbf{X})$ of Eq. (54), we need a valid Fock matrix—that is, a Fock matrix constructed from a valid density matrix. In the usual MO implementation of the DIIS scheme, the Fock matrix calculated according to Eq. (88) is not a valid Fock matrix. To solve this problem, we therefore average the AO density matrix rather than the Fock matrix

$$(\mathbf{D}_n)_{\mathrm{av}}=\sum_{i=1}^{n}\omega_i\mathbf{D}_i. \quad (93)$$

This averaged density matrix is then purified (using McWeeny's scheme[25]) to yield a valid, effective density matrix from which the effective Fock matrix is constructed.

TABLE II. The energy difference between the converged Hartree–Fock energy E^{conv} and the energy using the RH DIIS ($E_{\mathrm{RH\ DIIS}}$) algorithm and the MPCG (E_{MPCG}) with exact Hessian preconditioner for the alkene $C_{24}H_{26}$ using the cc-pVDZ basis set. The converged energy is −923.817 086 023 a.u.

	$E_{\mathrm{RH\ DIIS}}-E^{\mathrm{conv}}$	$E_{\mathrm{MPCG}}-E^{\mathrm{conv}}$
1	4.25	4.25
2	0.22	0.22
3	0.023	0.022
4	0.0015	0.0013
5	0.000 23	0.000 21
6	0.000 014	0.000 013
7	0.000 001 7	0.000 001 3
8	0.000 000 23	0.000 000 17
9	0.000 000 030	0.000 000 027
10	0.000 000 007	0.000 000 007
11	0.000 000 001	0.000 000 000
12	0.000 000 000	0.000 000 000
13	0.000 000 000	0.000 000 000

Thus, having obtained valid effective density and Fock matrices, we may proceed to optimize $E_\epsilon(\mathbf{X})$. Note that without purification, this procedure yields the same effective Fock matrix as Eq. (88) and that the purification introduce the need for calculating a new Fock matrix.

Thus, the density-matrix implementation of DIIS differs from the MO implementation in that we construct an averaged density matrix, Eq. (93), rather than an averaged Fock matrix, Eq. (88), storing in each iteration $\mathbf{D}_n$ and $\mathbf{e}_n$ rather than $\mathbf{F}_n$ and $\mathbf{e}_n$. Close to convergence, the averaged Fock matrix constructed according to Eq. (88) is a good approximation to the Fock matrix constructed from the (purified) density matrix in Eq. (93). We may then avoid the expensive construction of the Fock matrix by using Eq. (88) directly, storing $\mathbf{F}_n$ rather than $\mathbf{D}_n$ in each iteration.

VIII. COMPUTATIONAL DETAILS

The algorithm described above has been implemented in a local version of the DALTON program.[26] To demonstrate the capabilities of the algorithm, we report a series of test calculations. As typical examples of saturated and unsaturated systems, we have used the normal alkanes $C_{18}H_{38}$, $C_{24}H_{50}$, and $C_{30}H_{62}$, the conjugated alkenes $C_{18}H_{20}$ and $C_{24}H_{26}$, and the alkynes $C_{18}H_2$ and $C_{24}H_2$. These systems were selected to test our algorithm for different bonding situations. For alkanes the density matrix decays rapidly with increasing interaction distance. For alkenes and alkynes the density matrix decays more slowly because of the conjugation of their π-electron systems.

The bond lengths were chosen to be 1.54, 1.33, and 1.19 Å for the single, double, and triple C–C bonds, respectively, and 1.09, 1.076, and 1.06 Å for the C–H bonds of the alkanes, alkenes, and alkynes, respectively. The basis sets considered are STO-3G, 3-21G, 6-31G*, cc-pVDZ, and aug-cc-pVDZ. The largest calculation includes 756 basis functions. The energies have been converged to at least $10^{-9}E_h$.

IX. SPARSITY OF THE DENSITY MATRIX

In this section we examine the size of the elements of the AO density matrix, $D_{\mu\nu}$, as a function of the distance between the centers of AOs μ and ν. The separation between basis functions μ on carbon atom I and ν on carbon atom J is denoted R_{IJ}. The separation is one for AOs on neighboring carbon atoms, two for next neighbors, and so on. We do not consider the elements of the Fock and overlap matrices as these are sparser than the density matrix. For more thorough investigations of matrix sparsity for minimal basis sets, see Refs. 11, 27, and 28.

In Fig. 3, we have plotted the natural logarithm of the largest AO density matrix elements for a given atomic separation, R_{IJ}, as a function of R_{IJ} between the carbon atoms of the alkane—the hydrogen atoms are assigned the same atomic number, μ, as the carbon atom to which they bond. An exponential decay is clearly observed for STO-3G, 3-21G, 6-31G*, and cc-pVDZ, in accordance with previous observations.[14,27,29] However, there is no correlation between the size of the matrix elements and the size of the basis. Thus, the cc-pVDZ basis set has the smallest maximum density-matrix elements, whereas the next smallest set 3-21G has the largest elements (disregarding aug-cc-pVDZ). Because of its diffuseness, the aug-cc-pVDZ basis set behaves differently, its matrix elements decrease very slowly with increasing separation.

TABLE III. The convergence at Level 2 after the first Fock iteration for the alkene $C_{24}H_{26}$ using the cc-pVDZ basis set for four different choices of preconditioner: (a) Exact at Levels 2 and 3, (b) $R_{IJ}=10$ at Levels 2 and 3, (c) $R_{IJ}=16$ at Levels 2 and 3, and (d) exact at Level 2 and $R_{IJ}=10$ at Level 3. At each iteration we give the difference between the converged energy, $E_\epsilon^{\mathrm{conv}}$, and the sum of the orbital energies obtained at each iteration as well as the gradient norm. The converged energy is −656.811 588 894 5 a.u.

	Exact		$R_{IJ}=10$		$R_{IJ}=16$		Exact/$R_{IJ}=10$	
	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\|\mathbf{G}\|$	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\|\mathbf{G}\|$	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\|\mathbf{G}\|$	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\|\mathbf{G}\|$
1	0.021	1.80	0.191	5.69	0.030	2.05	0.021	1.80
2	0.000 006 2	0.034	0.056	2.57	0.000 39	0.17	0.000 006 1	0.033
3	0.000 000 000 3	0.000 23	0.010	1.29	0.000 000 73	0.0095	0.000 000 000 3	0.000 41
4	0.000 000 000 0	0.000 000 12	0.0018	0.63	0.000 000 004 0	0.000 68	0.000 000 000 0	0.000 011
5			0.000 37	0.29	0.000 000 000 0			
6			0.000 12	0.17				
7			0.000 038	0.11				
8			0.000 016	0.072				
10			0.000 007 0					

TABLE IV. The convergence at Level 2 after the third Fock iteration for the alkene $C_{24}H_{26}$ using the cc-pVDZ basis set for four different choices of preconditioner: (a) Exact at Levels 2 and 3, (b) $R_{IJ}=10$ at Levels 2 and 3, (c) $R_{IJ}=16$ at Level 2 and 3, and (d) exact at Level 2 and $R_{IJ}=10$ at Level 3. At each iteration we give the difference between the converged sum of orbital energies, $E_\epsilon^{\mathrm{conv}}$, and the sum of orbital energies obtained at each iteration as well as the gradient norm. The converged energy is $-621.218\,704\,107\,1$ a.u.

	Exact		$\lvert I-J\rvert=10$		$\lvert I-J\rvert=16$		Exact/$\lvert I-J\rvert=10$	
	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\lVert\mathbf{G}\rVert$	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\lVert\mathbf{G}\rVert$	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\lVert\mathbf{G}\rVert$	$E_\epsilon-E_\epsilon^{\mathrm{conv}}$	$\lVert\mathbf{G}\rVert$
1	0.000 000 001 6	0.000 41	0.000 0007 1	0.041	0.000 000 072	0.0028	0.000 000 001 6	0.000 43
2	0.000 000 000 0	0.000 001 1	0.000 000 65	0.012	0.000 000 000 0	0.000 075	0.000 000 000 0	0.000 006 5
3			0.000 000 10	0.0060				
4			0.000 000 031	0.0033				

For the alkenes in Fig. 4, an exponential but oscillating decay is observed. The decay is slower than for the alkanes but almost independent of the basis set. For the alkynes in Fig. 5, the STO-3G basis exhibits an oscillatory behavior similar to that of the alkenes, whereas the other basis sets display a slow, nonexponential decline. Again, the aug-cc-pVDZ basis behaves differently for the alkene and alkyne, the matrix elements decrease very slowly with increasing separation (and are larger than for the alkane).

In practice, what matters most is not so much the largest matrix element but rather the total number of large elements. In Table I we have listed the number of matrix elements with $|D_{\mu\nu}|>10^{-3}$ as a function of the atomic separation R_{IJ} for $C_{24}H_{50}$. The percentage of surviving elements is 24%, 20%, 13%, and 18% for the STO-3G, 3-21G, 6-31G*, and cc-pVDZ basis sets, respectively.

We note that the HOMO–LUMO gap differs greatly for the different basis sets and molecules. For the alkanes it is in the range 0.56 (cc-pVDZ)–0.89 (STO-3G), with the aug-cc-pVDZ basis having a gap of only 0.42. For the alkene and alkyne, the range for the HOMO–LUMO gap is 0.30–0.38 and 0.14–0.15, respectively.

We have seen that all matrices **D**, **F**, and **S** may be truncated according to a distance criteria—that is, for each of these matrices, there is a distance d such that the matrix elements M_{IJ} vanish if the distance between I and J exceeds d. Moreover, the product of such sparse matrices will also be a sparse matrix, whose nonzero distance is related to the nonzero distances of the matrices entering the product. For example, for two matrices **A** and **B** with nonzero distances d_A and d_B, respectively, the product matrix **AB** has nonzero matrix elements only for distances smaller than d_A+d_B. The Hessian times a sparse matrix may thus be evaluated as a sequence of sparse matrix multiplications to give a sparse matrix, in accordance with the linear scaling principle.

TABLE V. The norm of the residual at Level 3 using an exact preconditioner and approximate preconditioners.

	Level 1: iter. 1, Level 2: iter. 1			Level 1: iter. 4, Level 2: iter. 1		
	Exact	$R_{IJ}=10$	$R_{IJ}=16$	Exact	$R_{IJ}=10$	$R_{IJ}=16$
1	1835.6	1835.7	1835.6	1.40	1.41	1.40
2	0.020	220.9	16.5	0.000 093	0.19	0.20
3	0.0022	49.5	2.28		0.038	0.041
4	0.000 20	26.1	0.35		0.0060	0.0066
5	0.000 017	21.6	0.022		0.0013	0.0014
6	0.000 002 3	15.4	0.0087		0.000 25	0.000 29
7		10.5			0.000 049	0.000 064
8		7.06				
9		4.13				
10		2.83				
11		2.10				
12		1.68				
13		1.40				
14		0.91				
15		0.56				

X. OPTIMIZATION OF $C_{24}H_{26}$ IN THE CC-PVDZ BASIS

In the present section we examine the convergence of $C_{24}H_{26}$/cc-pVDZ calculations using the MPCG approach of Fig. 2, discussing each of the four levels in turn.

A. Level 1: Fock iterations

Table II describes the convergence at Level 1, giving in each iteration the energy difference between the converged Hartree–Fock (HF) energy E^{conv} and the energy obtained using the MPCG algorithm and exact preconditioning. For comparison the energy difference is also given for the RH DIIS algorithm. The behavior of the two methods is nearly identical, both converging to $10^{-3}E_h$ in 5 iterations and to $10^{-9}E_h$ in 11 iterations. The small differences that are observed between the two methods arise from the slightly different implementations of DIIS, which for MPCG involves not only the construction of an averaged density matrix but also its purification. Before we go on to study the higher levels, we recall that the Fock iterations at Level 1 are the

TABLE VI. The norm of the residual at Level 4 at the first iteration sequence at this level in the first Fock iteration and in a later iteration sequence.

	$\lVert\mathbf{r}\rVert$	$\lVert\mathbf{r}\rVert$
1	1.93×10^{5}	1.60
2	2.74×10^{4}	0.35
3	7.75×10^{3}	0.14
4	2.68×10^{3}	0.052
5	1.27×10^{3}	0.031
⋮	⋮	⋮
25	1.15	0.000 032
26	0.90	0.000 028
27	0.68	0.000 021
28	0.55	0.000 019
29	0.44	0.000 016
30	0.37	0.000 014

TABLE VII. Number of iterations at Levels 2 and 3 for alkanes of various lengths using the STO-3G, 3-21G, and 6-31G* basis sets. The cut-off criteria were 4, 4, and 5 for the STO-3G, 3-21G, and 6-31G* basis sets, respectively.

		STO-3G			3-21G			6-31G*
Preconditioner		$C_{18}H_{38}$	$C_{24}H_{50}$	$C_{30}H_{62}$	$C_{18}H_{38}$	$C_{24}H_{50}$	$C_{30}H_{62}$	$C_{24}H_{50}$
Exact	Level 2	1–2	1–2	1–2	1–2	1–2	1–2	1–3
	Level 3	1–3	1–3	1–3	1–3	1–3	1–3	1–4
Nondiagonal	Level 2	1–2	1–2	1–2	1–2	1–2	1–2	1–3
	Level 3	1–3	1–3	1–3	1–4	1–4	1–4	1–5
Diagonal	Level 2	1–10	1–10	1–10	1–30	1–30	1–30	1–35

only iterations that require the construction of the Fock matrix from two-electron integrals. Thus, at all higher levels only one-electron matrices are manipulated.

B. Level 2: Optimization of E_ϵ

In Tables III and IV the convergence at Level 2 is illustrated for the first and third Fock iterations, respectively, which are chosen as typical global and local regions. In both regions four preconditioning schemes are investigated:

(1) Exact Hessian for Levels 2 and 3;
(2) Hessian with cut-off at $R_{IJ}=10$ for Levels 2 and 3;
(3) Hessian with cut-off at $R_{IJ}=16$ for Levels 2 and 3;
(4) Exact Hessian at Level 2 and Hessian with cut-off at $R_{IJ}=10$ for Level 3.

In all cases, a diagonal unit preconditioner was used as preconditioner at Level 4. The energy differences between the converged sum of the orbital energies, E^{conv}, and the energy sum that is obtained at each iteration is given together with the norm of the gradient **G**.

The iterations with the exact Hessian preconditioner exhibit quadratic convergence, the number of correct digits about doubling in each iteration. This is expected as the PCG method with the exact Hessian preconditioner is essentially equivalent to Newtons method for optimizing E^{conv}. In the first Fock iteration (Table III), four PCG iterations are needed for convergence with the exact Hessian preconditioner whereas in the third Fock iteration (Table IV), only two iterations are needed. In the PCG calculation with the exact Hessian at Level 2 but an approximate Hessian at Level 3, the convergence is much the same, suggesting that the use of a hierarchy of preconditioners is a sound idea.

When an approximate preconditioner is used at Level 2, the convergence is slower than with the exact Hessian, in particular with cut-off at $R_{IJ}=10$. However, we note that tight convergence is seldom needed at Level 2, in particular in the global region of the Fock iterations.

C. Level 3: Solution of linear equations with a nondiagonal preconditioner

In Table V we have illustrated the convergence of the PCG method for solving the linear equations at Level 3, using as examples the first Level-2 iteration of the first and fourth Fock iterations. With an exact Hessian preconditioner convergence to a residual smaller than 10^{-5} is rapid with six Level-3 iterations needed for the first Fock iteration and only two for the fourth Fock iteration. With approximate preconditioners the convergence is noticeably slower in the first Fock iteration, in particular with the smallest cut-off at $R_{IJ}=10$. Recall that slow convergence with this cut-off was also observed at Level 2 in the same Fock iteration. In the fourth Fock iteration, the convergence is rapid in all cases.

D. Level 4: Solution of linear equations with a diagonal preconditioner

To illustrate the convergence at Level 4, we have in Table VI listed the residual of two typical Level-4 iteration sequences with unit preconditioner. The residual decreases rapidly in the first few iterations but then grinds almost to a halt. This behavior is fairly typical for the conjugate-gradient method and indicates that the optimization should be ''reset'' by recalculating the residual at Level 3 and restarting the optimization at Level 4. So far, we have not given much attention to the iterations at Level 4, but we shall consider the convergence at this level more carefully in the future to improve convergence.

TABLE VIII. Number of iterations at Levels 2 and 3 for alkanes of various lengths using the cc-pVDZ and aug-cc-pVDZ basis sets. For the cc-pVDZ basis set a cutoff of 6 was used for the non-diagonal preconditioner.

		cc-pVDZ			aug-cc-pVDZ
Preconditioner		$C_{18}H_{38}$	$C_{24}H_{50}$	$C_{30}H_{62}$	$C_{18}H_{38}$
Exact	Level 2	1–3	1–3	1–3	3–4
	Level 3	1–4	1–4	1–4	2–3
Nondiagonal	Level 2	1–3	1–3	1–3	⋯
	Level 3	1–6	1–7	1–7	⋯
Diagonal	Level 2	1–40	1–40	1–40	⋯

TABLE IX. Number of iterations at Levels 2 and 3 for alkenes using the STO-3G, 3-21G, 6-31G*, cc-pVDZ, and aug-cc-pVDZ basis sets. For the first four basis sets we studied $C_{24}H_{26}$, whereas for the aug-cc-pVDZ basis set $C_{18}H_{20}$ was studied. The cut-off criteria for the nondiagonal preconditioner were 6, 7, 10, and 16 for the STO-3G, 3-21G, 6-31G*, and cc-pVDZ, respectively.

Preconditioner		STO-3G	3-21G	6-31G*	cc-pVDZ	aug-cc-pVDZ
Exact	Level 2	1–2	1–2	1–2	1–3	1–4
	Level 3	1–2	1–3	1–3	1–3	4–9
Nondiagonal	Level 2	1–4	1–4	1–6	1–4	···
	Level 3	1–4	1–6	1–9	1–6	···
Diagonal	Level 2	1–10	1–40	1–55	1–90	···

XI. COMPARISONS OF DIFFERENT BASIS SETS AND MOLECULES

We now proceed to discuss and compare the overall convergence for alkanes, alkenes, and alkynes using different basis sets and preconditioners. In particular, we have used the basis sets STO-3G, 3-21G, 6-31G*, cc-pVDZ, and aug-cc-pVDZ with three preconditioning schemes: The exact Hessian at Levels 2 and 3, diagonal Hessian at Level 2 (no higher levels), and nondiagonal, approximate Hessians at Levels 2 and 3. The smallest and largest number of iterations needed for the different basis sets and preconditioners are listed in Tables VII and VIII for the alkanes, in Table IX for the alkenes, and in Table X for the alkynes.

With the exact Hessian preconditioner, convergence is straightforwardly obtained—even for the alkyne in the aug-cc-pVDZ basis set. Moreover, the number of iterations is almost independent of molecule and basis set.

With a diagonal preconditioner the convergence is in general slow and unsystematic. The number of iterations increases with the size of the basis set but is nearly independent of the size of the molecule. Because of the more delocalized electronic structure, the convergence is slower for the alkenes than for the alkanes. Using a diagonal preconditioner with the STO-3G basis set, Scuseria *et al.*[12] have reported convergence in fewer iterations. However, their convergence threshold is not nearly as tight as ours. In general, STO-3G calculations are quite easily converged with a diagonal preconditioner.

Finally, we consider the nondiagonal preconditioner. For the alkanes we used cut-off thresholds of 4, 4, 5, and 6 for the STO-3G, 3-21G, 6-31G*, and cc-pVDZ basis sets, respectively. For the alkenes the corresponding thresholds are 6, 7, 10, and 16. With these cut-offs convergence is much faster than with the diagonal preconditioner and, in fact, not significantly slower than with the exact preconditioner. Moreover, the number of iterations increases only slightly with the size of the basis set. One of the most difficult cases is the optimization of $C_{24}H_{26}$ in the cc-pVDZ basis, which was discussed in great detail in Sec. X.

In conclusion, the exact Hessian preconditioner works well for all molecules and basis sets we have considered. As expected, approximate preconditioners work better for the alkanes than for the alkenes and alkynes, which have sparser density matrices and smaller HOMO–LUMO gaps. We note, however, that the convergence of the alkenes and alkynes considered here is probably more difficult than that of most biological systems.

In our calculations the reference to the MO basis has not been completely eliminated as we have used a simple Hückel guess for the initial density matrix. For minimal basis sets such as STO-3G a simple guess is obtained by using the unit matrix and ensuring that the rank condition is fulfilled. For larger basis sets we may construct an initial density matrix using the STO-3G density matrix as a first guess. We may also use a less crude cut-off for the preconditioner in the first iterations and a cruder one as convergence is approached.

XII. CONCLUSIONS

We have presented an implementation of the direct optimization of the density matrix in the AO basis as an alternative to the diagonalization of the Fock matrix. Our optimization is based on an unconstrained but nonredundant exponential parametrization of the AO density matrix. The Fock iterations of the Roothaan–Hall scheme are retained, but the diagonalization of the Fock matrix is replaced by a direct minimization of the AO density matrix. For this minimization we propose the use of the conjugate-gradient algorithm with a nondiagonal multilevel preconditioner. With an exact Hessian preconditioner and use of projectors to control the redundancies, a robust second-order convergence is obtained for all systems we have considered: alkanes, alkenes, and alkynes with basis sets STO-3G, 3-21G, 6-31G*, cc-

TABLE X. Number of iterations at Levels 2 and 3 using the exact preconditioner for various alkynes using the STO-3G, 3-21G, 6-31G*, cc-pVDZ, and aug-cc-pVDZ basis sets.

		STO-3G		3-21G	6-31G*	cc-pVDZ		aug-cc-pVDZ
Preconditioner		$C_{18}H_2$	$C_{24}H_2$	$C_{24}H_2$	$C_{24}H_2$	$C_{18}H_2$	$C_{24}H_2$	$C_{18}H_2$
Exact	Level 2	1–2	1–2	1–3	1–3	1–3	1–3	1–6
	Level 3	1–2	1–2	1–2	1–4	1–4	1–4	3–11

pVDZ, and aug-cc-pVDZ. All calculations are in accordance with the linear-scaling principle, involving only simple sparse matrix additions and multiplications.

The cost of the calculations are reduced by using approximate Hessian preconditioners, where elements of the overlap, Fock, and density matrices are neglected based on a simple distance criterion, with the optimal cut-off threshold depending on the diffuseness of the basis set and the degree of electron delocalization. Additional reductions are possible by introducing a hierarchy of preconditioners. Tighter cut-off thresholds can be imposed on the higher preconditioners without affecting the overall convergence.

In short, we have demonstrated that linear scaling may be achieved for large molecules using basis sets that contain polarization and diffuse functions. Although our implementation is based on the exponential parametrization of the density matrix, the preconditioners suggested here may be employed also with conjugate-gradient implementations that are not based on this particular parametrization.

ACKNOWLEDGMENT

This work has been supported by the Danish Research Council (Grant No. 9901973).

[1] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).
[2] E. Schwegler and M. Challacombe, J. Chem. Phys. **105**, 2726 (1996).
[3] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **253**, 268 (1996).
[4] M. Challacombe and E. Schwegler, J. Chem. Phys. **106**, 5526 (1997).
[5] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).
[6] C. Ochsenfeld, C. A. White, and M. Head-Gordon, J. Chem. Phys. **109**, 1663 (1998).
[7] J. P. Stewart, P. Császár, and P. Pulay, J. Comput. Chem. **3**, 227 (1982).
[8] G. Galli and M. Parinello, Phys. Rev. Lett. **69**, 3547 (1992).
[9] J. P. Stewart, Int. J. Quantum Chem. **58**, 133 (1996).
[10] X.-P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).
[11] A. D. Daniels, J. M. Millam, and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).
[12] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).
[13] A. D. Daniels and G. E. Scuseria, J. Chem. Phys. **110**, 1321 (1999).
[14] M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).
[15] A. P. Rendell, Chem. Phys. Lett. **229**, 204 (1994).
[16] V. Termath, Int. J. Quantum Chem. **61**, 349 (1997).
[17] A. T. Wong and R. J. Harrison, J. Comput. Chem. **16**, 121 (1995).
[18] M. Challacombe, Comput. Phys. Commun. **128**, 93 (2000).
[19] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Wiley, Chichester, 2000).
[20] T. Helgaker, H. Larsen, J. Olsen, and P. Jørgensen, Chem. Phys. Lett. **327**, 397 (2000).
[21] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).
[22] P. Pulay, J. Comput. Chem. **3**, 556 (1982).
[23] M. Hestenes, *Conjugate Direction Methods in Optimization* (Springer-Verlag, New York, 1980).
[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Fortran Numerical Recipes*, 2nd ed. (Cambridge University Press, Cambridge, 1996).
[25] R. McWeeney, Rev. Mod. Phys. **32**, 335 (1960).
[26] An electronic structure program, Release 1.0 (1997) written by T. Helgaker, H. J. Aa. Jensen, P. Jørgensen *et al.*
[27] P. E. Maslen, C. Ochsenfeld, C. A. White, M. S. Lee, and M. Head-Gordon, J. Phys. Chem. **102**, 2215 (1998).
[28] C. Ochsenfeld and M. Head-Gordon, Chem. Phys. Lett. **270**, 399 (1997).
[29] W. Kohn, Int. J. Quantum Chem. **56**, 229 (1995).