

# An efficient orbital transformation method for electronic structure calculations

Cite as: J. Chem. Phys. **118**, 4365 (2003); <https://doi.org/10.1063/1.1543154>

Submitted: 31 October 2002 • Accepted: 11 December 2002 • Published Online: 20 February 2003

Joost VandeVondele and Jürg Hutter



View Online



Export Citation

## ARTICLES YOU MAY BE INTERESTED IN

[Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases](#)

The Journal of Chemical Physics **127**, 114105 (2007); <https://doi.org/10.1063/1.2770708>

[A consistent and accurate ab initio parametrization of density functional dispersion correction \(DFT-D\) for the 94 elements H-Pu](#)

The Journal of Chemical Physics **132**, 154104 (2010); <https://doi.org/10.1063/1.3382344>

[Direct energy functional minimization under orthogonality constraints](#)

The Journal of Chemical Physics **128**, 084113 (2008); <https://doi.org/10.1063/1.2841077>

Learn More

The Journal of Chemical Physics **Special Topics** Open for Submissions

# An efficient orbital transformation method for electronic structure calculations

Joost VandeVondele and Jürg Hutter

*Institute of Physical Chemistry, University of Zurich, Switzerland*

(Received 31 October 2002; accepted 11 December 2002)

An efficient method for optimizing single-determinant wave functions of medium and large systems is presented. It is based on a minimization of the energy functional using a new set of variables to perform orbital transformations. With this method convergence of the wave function is guaranteed. Preconditioners with different computational cost and efficiency have been constructed. Depending on the preconditioner, the method needs a number of iterations that is very similar to the established diagonalization–DIIS approach, in cases where the latter converges well. Diagonalization of the Kohn–Sham matrix can be avoided and the sparsity of the overlap and Kohn–Sham matrix can be exploited. If sparsity is taken into account, the method scales as  $O(MN^2)$ , where  $M$  is the total number of basis functions and  $N$  is the number of occupied orbitals. The relative performance of the method is optimal for large systems that are described with high quality basis sets, and for which the density matrices are not yet sparse. We present a benchmark calculation on a DNA crystal containing  $2 \times 12$  base pairs, solvent and counter ions (2388 atoms), using a TZV( $2d,2p$ ) basis (38 688 basis functions) and conclude that the electronic structure of systems of this size can now be studied routinely. © 2003 American Institute of Physics. [DOI: 10.1063/1.1543154]

## I. INTRODUCTION

The calculation of the electronic structure with density functional or Hartree–Fock theory amounts to the minimization of the electronic energy with respect to the orthonormal one-particle orbitals. These orbitals are used to construct the one-particle density matrix of the system. One of the most efficient methods that is commonly used to perform this optimization is diagonalization combined with a direct inversion in the iterative subspace (DIIS).<sup>1</sup> Diagonalization–DIIS requires often only 10 to 20 Kohn–Sham matrix constructions and hence has become the method of choice for many electronic structure calculations. However, in some cases diagonalization–DIIS might not converge to any solution. Methods with guaranteed convergence are therefore of significant interest, especially if they achieve convergence in approximately the same number of iterations as DIIS in the favorable cases.<sup>2,3</sup> Another deficiency of the diagonalization–DIIS approach is that the solution of the generalized eigenvalue problem needs  $O(M^3)$  time and becomes the computationally dominant part of the calculation for large systems.

The reason for this is that efficient and accurate methods exist to compute the energy and to construct the Kohn–Sham matrix in  $O(M^2)$  or even  $O(M)$  time for large systems.<sup>4</sup> These methods take advantage of the sparsity of the overlap matrix and the Kohn–Sham or Fock matrix, and employ fast methods to compute the Coulomb potential. Already rather small systems can be in the linear scaling regime since the sparsity of the Kohn–Sham matrix is related to the spatial extent of the atomic orbitals. In order to exploit the efficient Kohn–Sham matrix construction in an electronic structure program, it is essential to use fast methods to construct the density matrix of these systems. One possible approach is to

exploit that the density matrix of a very extended system can be sparse. For such systems, matrix diagonalization can be avoided and the construction of the density matrix can be performed in linear scaling time.<sup>4</sup> However, the sparsity of the density matrix is usually much less pronounced than the sparsity of the Kohn–Sham or overlap matrix,<sup>5</sup> especially if large basis sets are used. If accurate calculations have to be done, the density matrix cannot be treated as if it were a sparse matrix, and a molecular orbital calculation is appropriate. Thus, there exists a wide range of systems where diagonalization dominates the computational cost, but where a linear scaling construction of the density matrix is not yet possible.

Using high quality basis sets for large condensed systems is important to describe liquids or delicately hydrogen bonded systems such as proteins, DNA and RNA. The orbital transformation based minimizer presented in this paper is well suited for these systems. It needs relatively few iterations, is guaranteed to converge, and every iteration is significantly faster than a standard diagonalization–DIIS step. Furthermore, since the orbital transformation method makes computations on even larger condensed phase systems possible, it might provide results against which the efficiency and accuracy of linear scaling methods can be judged. For large systems with  $N$  occupied orbitals and  $M$  basis functions, the computational cost of the method scales as  $O(MN^2)$  with a small prefactor. The memory requirements are  $O(MN)$ . Since the resources needed scale linearly in the number of basis functions, substantial improvement over the standard diagonalization–DIIS method is to be expected whenever large basis sets are employed. In particular, the method could be well suited for plane wave bases.

The orbital transformation method has been imple-

mented in the program Quickstep. Quickstep is available as part of the CP2K suite of programs.<sup>6</sup> CP2K is freely available and comes with a GPL license.<sup>7</sup> It is based on a mixed Gaussian and plane wave scheme in which the wave functions are described using a Gaussian basis, but an intermediate plane wave basis is employed for the density.<sup>8</sup> Quickstep allows for a very fast, parallel and linear scaling construction of the Kohn–Sham matrix and is thus well suited for testing the orbital transformation method on large systems. Several possible preconditioners for the minimizer are tested on liquid water configurations, and the electronic structure of a DNA crystal has been computed to test the feasibility of accurate electronic structure calculations of condensed phase systems containing thousands of atoms.

## II. ORBITAL TRANSFORMATIONS

We express  $N$  occupied orthonormal wave functions  $\psi_i(r)$  as a linear combination of  $M$  basis functions  $\phi_\mu(r)$  using an  $M \times N$  matrix  $\mathbf{c}$  of orbital coefficients  $c_{\mu i}$ . The calculation of the electronic structure involves the minimization of the electronic energy  $E(\mathbf{c})$  with respect to  $\mathbf{c}$  subject to the constraint that  $\mathbf{c}^T \mathbf{S} \mathbf{c} = \mathbf{I}$ . The constraint enforces the orthonormality of  $\psi_i(r)$ .  $\mathbf{S}$  is the overlap matrix [ $S_{\alpha\beta} = \int \phi_\alpha(r) \phi_\beta(r) dr$ ] and  $\mathbf{I}$  the identity matrix. New variables to describe  $\mathbf{c}$ , inspired by the form of the exponential transformations<sup>9</sup> as given in Ref. 10, are introduced. Recently, we became aware of a minimization procedure also related to this exponential transformation that is similar to our minimization scheme.<sup>2</sup> However, in our scheme the new variables  $\mathbf{x}$  are linearly constrained according to

$$\mathbf{x}^T \mathbf{S} \mathbf{c}_0 = 0.$$

The constant initial vectors  $\mathbf{c}_0$  fulfill  $\mathbf{c}_0^T \mathbf{S} \mathbf{c}_0 = \mathbf{I}$ .  $\mathbf{c}$  is related to  $\mathbf{x}$  by the following transformation:

$$\mathbf{c}(\mathbf{x}) = \mathbf{c}_0 \cos(\mathbf{U}) + \mathbf{x} \mathbf{U}^{-1} \sin(\mathbf{U}),$$

where the matrix  $\mathbf{U}$  is defined as

$$\mathbf{U} = (\mathbf{x}^T \mathbf{S} \mathbf{x})^{1/2}.$$

The functions of the symmetric matrix  $\mathbf{x}^T \mathbf{S} \mathbf{x}$  are defined as functions of its eigenvalues. It can be verified that  $\mathbf{c}^T(\mathbf{x}) \mathbf{S} \mathbf{c}(\mathbf{x}) = \mathbf{I}$  for all choices of  $\mathbf{x}$ . The variables  $\mathbf{x}$  can be used to optimize the energy  $E(\mathbf{c}(\mathbf{x}))$  using any standard minimization algorithm, e.g., conjugate gradients. This is possible because  $\mathbf{x}$  is linearly constrained which implies that the space of allowed  $\mathbf{x}$  is a linear space. Therefore, a finite step along the gradient will produce a new point that still fulfills the constraint condition exactly. Hence, it is unnecessary to follow a curved geodesic during minimization, as is appropriate for variables that are constrained nonlinearly such as  $\mathbf{c}$ .<sup>2,11</sup> Using any globally convergent minimizer<sup>12</sup> convergence of the wave function is guaranteed. This is a desirable property that is not achieved with DIIS.<sup>2,3</sup>

The gradient of the energy with respect to the new variable  $\mathbf{x}$  is given by

$$\frac{\partial E(\mathbf{c}(\mathbf{x})) + \text{tr}(\mathbf{x}^T \mathbf{S} \mathbf{c}_0 \Lambda)}{\partial \mathbf{x}} = \frac{\partial E(\mathbf{c})}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial \mathbf{x}} + \mathbf{S} \mathbf{c}_0 \Lambda,$$

$$\Lambda = -[(\mathbf{S} \mathbf{c}_0)^T \mathbf{S} \mathbf{c}_0]^{-1} (\mathbf{S} \mathbf{c}_0)^T \frac{\partial E}{\partial \mathbf{x}},$$

where  $\Lambda$  is the matrix of Lagrangian multipliers that guarantees that the gradient fulfills the constraint condition. In order to compute the new term  $\partial \mathbf{c} / \partial \mathbf{x}$  it is advantageous to use a Cauchy representation of the matrix functions since it provides a simple way to compute the derivative

$$f(\mathbf{A}) = -\frac{1}{2\pi i} \oint f(z) \frac{1}{\mathbf{A} - z\mathbf{I}} dz,$$

$$\frac{\partial f(\mathbf{A})}{\partial \mathbf{x}} = \frac{1}{2\pi i} \oint f(z) \frac{1}{\mathbf{A} - z\mathbf{I}} \frac{\partial \mathbf{A}}{\partial \mathbf{x}} \frac{1}{\mathbf{A} - z\mathbf{I}} dz.$$

The gradient is given by

$$\frac{\partial E}{\partial \mathbf{x}} = (\mathbf{H} \mathbf{c}) \mathbf{U}^{-1} \sin(\mathbf{U}) + (\mathbf{S} \mathbf{x}) (\mathbf{R} (\mathbf{K}^T + \mathbf{K}) \mathbf{R}^T)$$

$$\mathbf{K} = (\mathbf{R}^T ((\mathbf{H} \mathbf{c})^T \mathbf{x}) \mathbf{R}) \otimes \mathbf{D}^1 + (\mathbf{R}^T ((\mathbf{H} \mathbf{c})^T \mathbf{c}_0) \mathbf{R}) \otimes \mathbf{D}^2.$$

The notation  $\otimes$  is used for the direct matrix product ( $\mathbf{A} \otimes \mathbf{B})_{ij} = A_{ij} B_{ij}$ ,  $\mathbf{H} \mathbf{c}$  represents  $\partial E / \partial \mathbf{c}$ ,  $\mathbf{R}$  is the matrix of eigenvectors of  $\mathbf{x}^T \mathbf{S} \mathbf{x}$ , and  $\mathbf{D}^1$  and  $\mathbf{D}^2$  are given by

$$D_{ij}^1 = \frac{\frac{\sin(\sqrt{\Lambda_i})}{\sqrt{\Lambda_i}} - \frac{\sin(\sqrt{\Lambda_j})}{\sqrt{\Lambda_j}}}{\Lambda_i - \Lambda_j},$$

$$D_{ij}^2 = \frac{\cos(\sqrt{\Lambda_i}) - \cos(\sqrt{\Lambda_j})}{\Lambda_i - \Lambda_j},$$

where  $\Lambda_i$  is the eigenvalue corresponding to the  $i$ th column of  $\mathbf{R}$ . The calculation of the gradient involves the computation of sparse matrix–full matrix products in order to compute  $\mathbf{H} \mathbf{c}$  and  $\mathbf{S} \mathbf{x}$ , and several full matrix–full matrix products such as  $(\mathbf{H} \mathbf{c})^T \mathbf{x}$  or  $\mathbf{x} f(\mathbf{U})$ . For large systems, they have a computational cost of  $O(MN)$  and  $O(MN^2)$ , respectively. Notice that the full matrix products run almost with peak performance on modern computers, and that they are well parallelized. Good performance of the method, both in serial and parallel can therefore be expected for large systems. It is more difficult to obtain peak performance and good parallel performance for the sparse matrix–full matrix multiply. In case the matrices  $\mathbf{H}$  and  $\mathbf{S}$  are not yet sparse the cost of computing  $\mathbf{H} \mathbf{c}$  and  $\mathbf{S} \mathbf{x}$  is  $O(M^2N)$ . For the special case of a plane wave basis  $\mathbf{S} = \mathbf{I}$  and  $\mathbf{H} \mathbf{c}$  can be computed in  $O(MN)$ , independent of the system size. The diagonalization of the  $N \times N$  matrix  $\mathbf{x}^T \mathbf{S} \mathbf{x}$ , which is needed to compute the matrix functions efficiently, costs only a small fraction of the total CPU time.

For a practical application it is important to precondition the minimization. Insight in the structure of the preconditioner is gained with a simple but important model system, i.e., the generalized eigenvalue problem. The function to be minimized is then

$$E(\mathbf{c}(\mathbf{x})) = \text{Tr}(\mathbf{c}(\mathbf{x})^T \mathbf{H} \mathbf{c}(\mathbf{x}) + \mathbf{x}^T \mathbf{S} \mathbf{c}_0 \Lambda),$$

where  $\mathbf{H}$  does not depend on  $\mathbf{c}$  and  $\mathbf{c}^T \mathbf{S} \mathbf{c} = \mathbf{I}$  is satisfied. To simplify the analysis, we assume that  $\mathbf{x}$  is small so that a second order expansion of the functional can be used, and we assume that  $(\mathbf{c}_0^T \mathbf{H} \mathbf{c}_0)_{ij} = \epsilon_i^0 \delta_{ij}$ . In this case the Hessian is given by

$$\frac{\partial^2 E}{\partial x_{\alpha i} \partial x_{\beta j}} = 2H_{\alpha\beta} \delta_{ij} - 2S_{\alpha\beta} \delta_{ij} \epsilon_i^0.$$

This Hessian is symmetric and positive definite [i.e.,  $x_{\alpha i} (\partial^2 E / \partial x_{\alpha i} \partial x_{\beta j}) x_{\beta j} > 0$  for all  $\mathbf{x}$  that satisfy  $(\mathbf{x}^T \mathbf{S} \mathbf{c}_0 = 0)$ ] if  $\mathbf{c}_0$  is close to the minimum. It can be observed that it would be optimal to use as a preconditioned gradient  $(\mathbf{H} - \mathbf{S} \epsilon_i^0)_{\alpha\beta}^{-1} (\partial E / \partial x_{\beta i})$ . Notice that every vector  $i$  of the gradient is multiplied with a different matrix. In practice, a single symmetric positive definite matrix  $\mathbf{P}$  is constructed so that  $\mathbf{P}(\mathbf{H} - \mathbf{S} \epsilon) \mathbf{x} - \mathbf{x} \approx 0$ . Since introducing a preconditioner is equivalent to a change in variables  $\mathbf{x} \rightarrow \sqrt{\mathbf{P}} \mathbf{x}$  the Lagrangian multipliers have to be adapted accordingly. The preconditioner is similar to the forms suggested in Refs. 13 and 14. Finally, a locking technique for orbital transformations based minimization, in which additional or only selected vectors are optimized, can be easily obtained by extending the constraint matrix ( $\mathbf{C}_0$ ) with a set of fixed vectors. This is useful to obtain LUMOs or could be employed during the SCF calculation to focus computational effort on a few slowly converging states.

### III. RESULTS

We have implemented orbital transformation based minimization in the program Quickstep and tested conjugate gradient and DIIS based minimizers. For the line search algorithm used in the first scheme quadratic interpolation (two points: one gradient, two energies) is sufficient in most cases, but a bracketing and golden section line search is available in order to guarantee convergence of the conjugate gradient minimizer in difficult cases. The DIIS scheme, using the preconditioned gradient as an error vector, needs generally about the same number of iterations as the conjugate gradient scheme needs line searches. Hence, DIIS is more efficient in terms of the number of function evaluations. However, convergence of DIIS to a minimum is not guaranteed and DIIS might fail for more difficult systems. Additionally, for large systems, the low memory requirements of the conjugate gradient algorithm make it the method of choice. All tests were initialized using the result from a minimal basis set calculation. The reference orbitals  $\mathbf{c}_0$  were constructed by projection of the minimal basis on the full basis set. Such a calculation has a small cost as compared to the calculation with the full basis set. In Fig. 1 results for a periodic system of 32 water molecules for two different basis sets and four different preconditioners are presented as well as the result without preconditioner (stars in Fig. 1). In all cases, a preconditioner has been constructed only once, even though the optimal preconditioner might change with the iterations. This is a good strategy for the large systems we are interested in, since the cost of constructing a preconditioner can be large compared to the cost of a few extra iterations. However, for small systems, where the cost of the construction of the Kohn–Sham

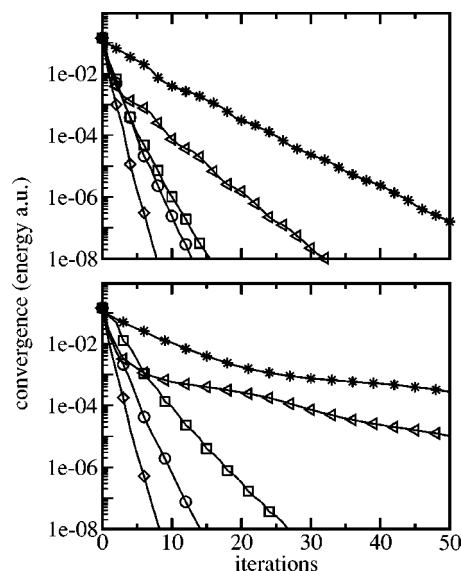


FIG. 1. Convergence of the total energy per molecule for a periodic system of 32 water molecules for different preconditioners using a conjugate gradient minimizer. Upper panel: DZV( $d,p$ ) basis 736 functions; lower panel: TZV( $2d,2p$ ) basis 1280 functions. The different symbols represent the different preconditioners: diamonds, diagonalization and inversion of  $\mathbf{H} - \epsilon^0 \mathbf{S}$ ; circles, overlap and kinetic energy matrix inversion; squares, overlap matrix inversion; triangles, diagonal atom block diagonalization; stars, none.

matrix is still dominant, it might be useful to update the preconditioner at every step in order to reduce the number of Kohn–Sham matrix evaluations. The most effective preconditioner (diamonds in Fig. 1) is based on a diagonalization and inversion of  $\mathbf{H} - \epsilon^0 \mathbf{S}$ .  $\epsilon^0$  is a constant that must be similar to the values of the eigenvalues of the occupied subspace. We use the highest eigenvalue of  $\mathbf{c}_0^T \mathbf{H} \mathbf{c}_0$  as an estimate. In order to guarantee a positive definite Hessian, all negative and small eigenvalues (corresponding to the occupied states, or due to nearby saddle points) are replaced with a positive constant that is similar to the expected gap of the system. As we show below, the cost of constructing this preconditioner (similar to the cost of a single DIIS step in the standard SCF scheme) can be prohibitive for large systems. The construction of this preconditioner cannot exploit the sparsity of the matrices  $\mathbf{H}$  and  $\mathbf{S}$ , and diagonalization based inversion is expensive. Computationally more efficient methods have therefore been developed. One simple preconditioner (triangles in Fig. 1) uses the same diagonalization based inversion as described before, but only for diagonal blocks of  $\mathbf{H} - \epsilon^0 \mathbf{S}$ . The diagonal blocks in our test are atom based. One advantage of this method is that the preconditioner is very sparse and can be computed easily. Applying this preconditioner is  $O(MN)$ , whereas the current implementation of the other preconditioners presented uses matrix products that do not exploit the sparsity of the preconditioner and hence are  $O(M^2N)$ . However, it can be observed that this strategy is only moderately successful for the smaller basis and performs poorly for the larger basis. The latter is most likely due to the significant overlap of basis functions on different atoms, so that the quality of the preconditioner would increase if larger blocks are considered. To really remedy the problem one should fully take the overlap matrix  $\mathbf{S}$  into account. One



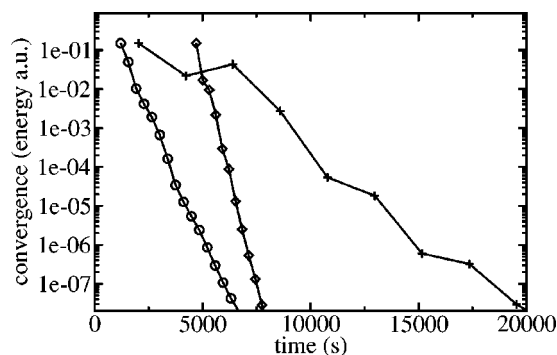


FIG. 2. Convergence of the total energy per molecule for a periodic system of 256 water molecules [TZV(2d,2p), 10 240 basis functions] vs elapsed time on four CPUs of a SUN ultra SPARC (750MHz). Circles, kinetic energy based preconditioner; diamonds, diagonalization based preconditioner; crosses, standard diagonalization and DIIS.

possible solution is to use  $\mathbf{S}^{-1}$  as a preconditioner (squares in Fig. 1). This preconditioner seems to work reasonably well. One significant advantage of this preconditioner is that it can be computed using a Cholesky factorization based inversion since the matrix is positive definite. Such an inversion is significantly faster than a diagonalization (by a factor of 5 to 10). A preconditioner that performs better than just  $\mathbf{S}^{-1}$  should take into account some of the information contained in  $\mathbf{H}$ . One simple way that maintains the sparse and positive definite character is to use  $(\mathbf{T} + \epsilon\mathbf{S})$ , where  $\mathbf{T}$  is the kinetic energy matrix (circles in Fig. 1). This preconditioner performs nearly as good as the preconditioner based on the diagonalization of the full Hamiltonian if the number of iterations is considered, but might outperform it on larger systems if the total time is considered. This is shown in Fig. 2 where the performance of these two preconditioners is shown together with the standard diagonalization–DIIS method. We notice that the orbital transformation based methods are significantly faster than the diagonalization based scheme and that this difference becomes larger for larger systems. Constructing the preconditioner for this system takes 550 and 4200 s, respectively. Tests with a 512 water molecule system (20 480 basis functions) indicate that the full matrix product needed to apply the preconditioner becomes the bottleneck whereas the Cholesky factorization based inversion is rather fast (42 and 11% of the total time). We plan to use either linear scaling techniques for computing the solution of sparse positive definite linear systems<sup>15</sup> or fast direct inversion methods for sparse positive definite systems.<sup>16</sup> Furthermore, a whole range of other approximate preconditioning methods such as, e.g., incomplete Cholesky factorization could be employed. These techniques would reduce the cost of preconditioning to  $O(MN)$ . Notice that for a plane wave basis set the efficient kinetic energy preconditioner can be computed in  $O(MN)$ , independent of the system size.

A last benchmark is the computation of the electronic structure of a DNA crystal. The system, which is based on a x-ray structure of synthetic DNA, has been studied before by Gervasio *et al.* (Ref. 17) and we refer to this paper for a detailed description of the system. However, the system has been doubled so that  $2 \times 12$  base-pairs, solvent and counter

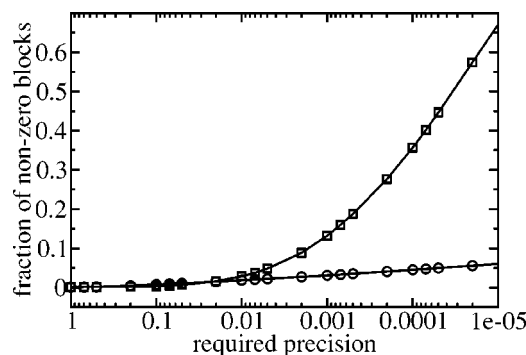


FIG. 3. The fraction of nonzero blocks of the density matrix (squares) and the overlap matrix (circles) is shown for the DNA system [DZV(*d,p*)] described in the text.

ions have been described, a total of 2388 atoms. A semi-core pseudopotential for sodium has been employed. A 350 Ry density cutoff has been employed in order to describe the semi-core states of sodium. The system has a total of 3960 occupied orbitals and DZV(*d,p*) and TZV(2d,2p) bases (22 596 and 38 688 basis functions) have been tested. A density mapping accuracy of  $10^{-12}$  has been chosen and elements of the overlap matrix have been neglected if they were estimated to be less than  $10^{-10}$ . The sparsity pattern of the overlap matrix has also been used for the Kohn–Sham matrix. The wave functions have been optimized using the orbital transformation method with a conjugate gradient minimizer and a kinetic energy based preconditioner. A minimal basis set has been used to initialize the wave function. The calculations have been performed on 32 CPUs of an SP4 (Power4 CPUs, 1.3 GHz), and needed about 2.5 and 5 h to converge. The time needed for a single two point line search was 675 and 1100 s for DZV(*d,p*) and TZV(2d,2p), respectively. We notice that these timings reflect accurately the effective linear scaling in the number of basis functions. Fourteen line searches were necessary to reach convergence (gradients  $2 \times 10^{-5}$ ). Interestingly, even for a system of this size, the cubically scaling part is not yet fully dominant. For the TZV(2d,2p) system, 45, 43, and 8% of the total time is spent in the cubic, the quadratic and the linear part, respectively. In part, this is an effect of the superior parallel efficiency of the cubic part. The timing for the cubic part includes the time needed for constructing and applying the preconditioner, 10 and 15% of the total time, respectively. Another observation of relevance is related to the sparsity of the density matrix. As shown in Fig. 3, over 60% of the atomic blocks of the density matrix are nonzero if a tolerance of only  $10^{-5}$  is used. Even if such a moderate accuracy is demanded, linear scaling methods that rely on the sparsity of the density matrix cannot be efficient.

#### IV. CONCLUSIONS

The orbital transformation method presented here is a fast method for accurate SCF calculations of medium to large systems. It is guaranteed to converge, and typically does so in less than 20 iterations. The computationally expensive part of these iterations are matrix–matrix products,

which reach near peak performance on many computers and scale well. We have shown that the electronic structure of systems containing a few thousand atoms can be calculated with an accuracy that is otherwise only common for smaller compounds. For large systems with a sparse Kohn–Sham and overlap matrix there is still potential to improve execution speed of the method. We expect that fast parallel sparse matrix–full matrix multiply and a preconditioning scheme that exploits the structure and sparsity of the Kohn–Sham matrix could increase the efficiency of the current implementation by factors 2 to 5.

## ACKNOWLEDGMENTS

One of the authors (J.V.) acknowledges access to the excellent computing environment of the Department of Mathematics. This work was supported by the Swiss National Science Foundation (No. 21-63305.00).

<sup>1</sup>P. Pulay, J. Comput. Chem. **3**, 556 (1982).

<sup>2</sup>T. Van Voorhis and M. Head-Gordon, Mol. Phys. **100**, 1713 (2002).

<sup>3</sup>K. N. Kudin, G. E. Scuseria, and E. Cancès, J. Chem. Phys. **116**, 8255 (2002).

<sup>4</sup>S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).

<sup>5</sup>P. E. Maslen, C. Ochsenfeld, C. A. White, M. S. Lee, and M. Head-Gordon, J. Phys. Chem. A **102**, 2215 (1998).

<sup>6</sup><http://cp2k.berlios.de/> can be publicly accessed and the latest version of the cp2k code can be downloaded anonymously.

<sup>7</sup>A description of the GPL license can be found at <http://www.gnu.org/licenses/gpl.html>.

<sup>8</sup>G. Lippert, J. Hutter, and M. Parrinello, Mol. Phys. **92**, 477 (1997).

<sup>9</sup>T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-structure Theory* (Wiley, Chichester, 2000), and references therein.

<sup>10</sup>J. Hutter, M. Parrinello, and S. Vogel, J. Chem. Phys. **101**, 3862 (1994).

<sup>11</sup>A. Edelman, T. A. Arias, and S. T. Smith, SIAM J. Matrix Anal. Appl. **20**, 303 (1998).

<sup>12</sup>J. Nocedal, in *The State of the Art in Numerical Analysis*, edited by A. Watson and I. Duff (Oxford University Press, Oxford, 1997).

<sup>13</sup>A. V. Knyazev, SIAM J. Sci. Comput. (USA) **23**, 517 (2001).

<sup>14</sup>C. K. Gan, P. D. Haynes, and M. C. Payne, Comput. Phys. Commun. **134**, 33 (2001).

<sup>15</sup>T. Ozaki, Phys. Rev. B **64**, 195110 (2001).

<sup>16</sup>PSPASES: Scalable parallel direct solver library for sparse symmetric positive definite linear systems, <http://www-users.cs.umn.edu/mjoshi/pspases/>

<sup>17</sup>G. L. Gervasio, P. Carloni, and M. Parrinello, Phys. Rev. Lett. **89**, 108102 (2002).