Robust and Efficient Density Fitting

Víctor D. Domínguez-Soria[a], Gerald Geudtner[a],

José Luis Morales[b], Patrizia Calaminici[a], Andreas M. Köster[a1]

[a]*Departamento de Química, CINVESTAV,*

Avenida Instituto Politécnico Nacional 2508

A.P. 14-740 México D.F. 07000, México

[b]*Departamento de Matemáticas, ITAM,*

Río Hondo 1, Col. Tizapán San Angel,

México D.F. 01000, México

Abstract

In this paper we propose an iterative method for solving the inhomogeneous systems of linear equations associated with density fitting. The proposed method is based on a version of the conjugate gradient method that makes use of automatically built quasi-Newton preconditioners. The paper gives a detailed description of a parallel implementation of the new method. The computational performance of the new algorithms is analyzed by benchmark calculations on systems with up to about 35,000 auxiliary functions. The numerical experiments indicate that the new method exhibits sub-quadratic complexity $\mathcal{O}(M^{1.7})$ with respect to M, the number of auxiliary functions. Comparisons with the standard, direct approach, show no significant differences in the computed solutions.

Keywords: Variational Coulomb Fitting, Auxiliary Density Functional Theory, Linear Scaling, Quasi-Newton Updating, Preconditioned Conjugate Gradient Methods

---

[1]Email: akoster@cinvestav.mx

1. Introduction

Over the last two decades density fitting methods have emerged from a technical particularity of some density functional theory (DFT) [1,2] programs to a very active research field in mainstream quantum chemistry. To the best of our knowledge these methods were first introduced independently by Whitten [3] and Baerends, Ellis, and Ross [4]. Whereas Whitten established bounds for the fitting of individual Coulomb integrals, Baerends, Ellis and Ross introduced a least-square orbital density fitting for the Xα method. Their approach was later on transfered to the linear combination of Gaussian-type orbital (LCGTO) implementation by Sambe and Felton [5]. Both methods, the fitting of individual molecular integrals as well as the least-square fit of the density (and potential), yield non-variational approximated energy expressions that might compromise the mathematical robustness of the underlying electronic structure method. In particular, the formulation of energy derivatives becomes cumbersome. In order to circumvent this problem Dunlap, Connolly, and Sabin [6] suggested a variational fitting procedure for the Coulomb potential which yields a variationally approximated energy. It was shown [7] that the accuracy of this particular fitting approach is within the intrinsic accuracy of LCGTO-DFT methods. Due to the variational nature, this approach can be used to formulate analytic gradients and higher energy derivatives, see for example [8–10]. The vast majority of modern density fitting methods [11–19] are based on the variational fitting of the Coulomb potential. In particular, the so-called resolution of the identity, nowadays employed in wave function methods, can be formulated as a variational fitting of orbital densities. Moreover, it has been shown that the variational nature of the approximated energy remains even if the approximated density from the variational fitting of the Coulomb potential is used for the calculation of the exchange-correlation energy and potential [16]. In this approach the Kohn-Sham potential is expressed over the auxiliary density and, therefore, it is named auxiliary density functional theory (ADFT). This theoretical framework enables first-principle calculations on systems with thousands of basis functions, see for example [20–22]. Recently, a modified form of McWeeny's self-consistent perturbation theory [23–25] has been formulated within the ADFT framework [26]. This – non iterative – approach is computationally more economical and departs from the rather expensive iterative solution of the coupled perturbed Kohn-Sham equations. Most recently, the variational nature of the corresponding perturbed potential fit was proven [27].

It is now well established that density fitting offers significant speed-ups for the calculation of Hartree-Fock or Kohn-Sham matrix elements at the cost of a negligible loss of accuracy. In particular, the Kohn-Sham matrix construction in ADFT methodology employing integral screening and expansion scales already linear for systems with less than 3000 basis functions [28]. As a consequence the linear algebra steps associated with ADFT calculations become a computational bottleneck already for medium size systems with only a few thousand basis functions. At this point it is important to note that density fitting introduces additional linear algebra steps, namely for the solution of the inhomogeneous fitting equation system, that contribute substantially to this bottleneck. So far this problem has been tackled by partitioning approaches [29–33] that alter the original formulation of the variational Coulomb fitting. In this paper we present an alternative approach. Instead of altering the variational fitting procedure we suggest an alternative technique for the solution of the inhomogeneous fitting equation system that removes the corresponding computational bottleneck. The method suggested in this paper builds upon a recently proposed algorithm for the automatic generation of preconditioners for conjugate gradient (CG) methods [34] by Morales and Nocedal [35]. The algorithm can be used for solving a sequence of linear equation systems,

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}_i \,,\ i = 1, \ldots, t \,, \tag{1}$$

in which the coefficient matrix is constant but the right-hand side varies. Briefly, at the time the i-th system is solved by the preconditioned conjugate gradient method, some information is carefully selected and stored. At completion of the i-th stage, that information is organized to build a preconditioner to be used in the $(i+1)$-th system. The preconditioner has the form of a BFGS matrix that approximates the inverse of $\mathbf{A}$. We will use this approach for the determination of the fitting coefficient from the variational Coulomb fit. In ADFT a second set of fitting coefficients occur [16]. Following Morales and Nocedal these coefficients can be determined by a preconditioned CG method based on the information generated during the calculation of the Coulomb fitting coefficients.

The proposed algorithm is well suited for very large problems. It can be efficiently parallelized as we will discuss later. In the following section the algorithmic details and the implementation of the iterative equation system solver will be described. The computational methodology for the validation and benchmark calculations is given in Section 3.

In Section 4 results of validation and benchmark calculations are discussed. Concluding remarks are drawn in the last section.

2.   Algorithm and Implementation

The variational fitting of the Coulomb potential is based on the minimization of the following error:

$$\mathcal{E}_2 = \frac{1}{2} \iint \frac{[\rho(\mathbf{r}_1) - \tilde{\rho}(\mathbf{r}_1)] \, [\rho(\mathbf{r}_2) - \tilde{\rho}(\mathbf{r}_2)]}{|\, \mathbf{r}_1 - \mathbf{r}_2 \,|} d\mathbf{r}_1 d\mathbf{r}_2 \, . \tag{2}$$

In the LCGTO approximation the (closed-shell) orbital density is expanded in terms of atomic orbitals:

$$\rho(\mathbf{r}) = \sum_{\mu,\nu} P_{\mu\nu} \mu(\mathbf{r}) \nu(\mathbf{r}) \, . \tag{3}$$

with

$$P_{\mu\nu} = 2 \sum_{i}^{occ} c_{\mu i} c_{\nu i} \, . \tag{4}$$

The $c_{\mu i}$ and $c_{\nu i}$ denote molecular orbital coefficients. For the expansion of the approximated density we employ atom centered primitive Hermite Gaussian auxiliary functions [36,37]:

$$\tilde{\rho}(\mathbf{r}) = \sum_{\bar{k}} x_{\bar{k}} \bar{k}(\mathbf{r}) \, , \tag{5}$$

where $\bar{k}(\mathbf{r})$ represents a Hermite Gaussian auxiliary function and $x_{\bar{k}}$ the corresponding density fitting coefficient. The variation of $\mathcal{E}_2$ with respect to these fitting coefficients yields:

$$\left(\frac{\partial \mathcal{E}_2}{\partial x_{\bar{k}}} \right)_{\mathbf{P}} = - \sum_{\mu,\nu} P_{\mu\nu} \langle \mu\nu \, \| \, \bar{k} \rangle + \sum_{\bar{l}} x_{\bar{l}} \langle \bar{l} \, \| \, \bar{k} \rangle \equiv 0 \; \forall \bar{k} \, . \tag{6}$$

In the above two- and three-center electron repulsion integral notation, the Coulomb operator $1/|\, \mathbf{r}_1 - \mathbf{r}_2 \,|$ is denoted by the symbol $\|$. The minimization of the Coulomb fitting error (2) is used for the determination of the $x_{\bar{k}}$ coefficients in each self-consistent field cycle. For a more extended discussion of the variational Coulomb fit in the context of the self-consistent field (SCF) procedure we refer to [38]. It is convenient to cast equation (6) in the following matrix form:

$$\mathbf{G}\,\mathbf{x} = \mathbf{J} \, . \tag{7}$$

Here $\mathbf{G}$ denotes the symmetric positive definite auxiliary function Coulomb matrix with elements $G_{\bar{k}\bar{l}} = \langle \bar{k} \, \| \, \bar{l} \rangle$; $\mathbf{J}$ is the corresponding Coulomb vector with elements

$$J_{\bar{k}} = \sum_{\mu,\nu} P_{\mu\nu} \langle \mu\nu \, \| \, \bar{k} \rangle \, ; \tag{8}$$

x is the vector of the Coulomb fitting coefficients. In ADFT a second set of fitting co-efficients occurs. These so-called exchange-correlation fitting coefficients are defined by [16]:

$$z_{\bar{k}} = \sum_{\bar{l}} \langle \bar{k} \parallel \bar{l} \rangle^{-1} \langle \bar{l} \mid v_{xc}[\tilde{\rho}] \rangle . \tag{9}$$

Again, it is convenient for our further discussion to cast equation (9) in matrix form:

$$\mathbf{G}\,\mathbf{z} = \mathbf{L} ; \tag{10}$$

where **z** collects the exchange-correlation fitting coefficients. The vector **L** on the right-hand side of equation (10) contains elements $L_{\bar{k}} = \langle \bar{k} \mid v_{xc}[\tilde{\rho}] \rangle$ that are obtained by numerical integration on a grid. Analog to the Coulomb vector in equation (7) we name it exchange-correlation vector. It is important to note that this vector is spin-polarized, as the corresponding exchange-correlation fitting coefficients. Thus, equation (10) must be solved for both spin manifolds separately. However, for the sake of clarity, we restrict ourself to the closed-shell case where these two solutions collapse. The comparison of equation (7) and (10) reveals that the two fitting coefficient sets appearing in ADFT can be obtained from two linear equation systems with the same coefficient matrix, namely the positive definite auxiliary function Coulomb matrix, and a varying right-hand side. Thus, the iterative solver proposed by Morales and Nocedal [35] for solving a sequence of linear equation systems can be applied.

2.1 Algorithmic Details

2.1.1 Coulomb Fitting Coefficients

The first task represents the determination of the Coulomb fitting coefficients by solving the inhomogeneous equation system (7). Instead of attempting a direct solution that involves factorizing the Coulomb matrix **G** we note that the solution of equation system (7) is equivalent to the minimization of the quadratic function $F(\mathbf{x})$ defined as:

$$F(\mathbf{x}) = \frac{1}{2} \sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}}\, x_{\bar{k}} x_{\bar{l}} - \sum_{\bar{k}} J_{\bar{k}}\, x_{\bar{k}} . \tag{11}$$

In order to find a minimum of $F(\mathbf{x})$ we perform an iterative quasi-Newton optimization [39, 40]. For this purpose we expand $F(\mathbf{x})$ as:

$$F(\mathbf{x} + \Delta\mathbf{x}) = \frac{1}{2} \sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}} \left(x_{\bar{k}} + \Delta x_{\bar{k}} \right) \left(x_{\bar{l}} + \Delta x_{\bar{l}} \right) - \sum_{\bar{k}} J_{\bar{k}} \left(x_{\bar{k}} + \Delta x_{\bar{k}} \right) . \tag{12}$$

The variation of $F(\mathbf{x} + \Delta\mathbf{x})$ with respect to the step $\Delta\mathbf{x}$ yields:

$$\frac{\partial F(\mathbf{x} + \Delta\mathbf{x})}{\partial \Delta x_{\bar{k}}} = \sum_{\bar{l}} G_{\bar{k}\bar{l}} (x_{\bar{l}} + \Delta x_{\bar{l}}) - J_{\bar{k}} \equiv 0 \ \forall \, \bar{k} \, . \tag{13}$$

The corresponding second derivative,

$$\frac{\partial^2 F(\mathbf{x} + \Delta\mathbf{x})}{\partial \Delta x_{\bar{k}} \, \partial \Delta x_{\bar{l}}} = G_{\bar{k}\bar{l}} \, , \tag{14}$$

ensures the minimization of $F(\mathbf{x} + \Delta\mathbf{x})$ by equation (13) due to the positive definite nature of $\mathbf{G}$. For the step direction then follows:

$$\sum_{\bar{l}} G_{\bar{k}\bar{l}} \Delta x_{\bar{l}} = -r_{\bar{k}} \ \Rightarrow \ \Delta x_{\bar{l}} = -\sum_{\bar{k}} G_{\bar{l}\bar{k}}^{-1} \, r_{\bar{k}} \, . \tag{15}$$

Here we have introduced the residual $\mathbf{r}$ with elements:

$$r_{\bar{k}} \equiv \sum_{\bar{l}} G_{\bar{k}\bar{l}} \, x_{\bar{l}} - J_{\bar{k}} \, . \tag{16}$$

The step formula (15) shows that the residual takes the role of the gradient and the Coulomb matrix represents the Hessian in the minimization of $F(\mathbf{x})$. In quasi-Newton methods the Hessian, here $\mathbf{G}$, is approximated by a symmetric positive definite matrix which is updated in each iterative step [41]. In order to avoid matrix inversion we employ the inverse BFGS update [42–45] that acts directly on the approximated inverse of $\mathbf{G}$, denoted by $\mathbf{B}$:

$$\mathbf{B} = \mathbf{B}^{old} + \left(1 + \frac{\boldsymbol{\gamma}^T \mathbf{B}^{old} \boldsymbol{\gamma}}{\boldsymbol{\delta}^T \boldsymbol{\gamma}}\right) \frac{\boldsymbol{\delta} \, \boldsymbol{\delta}^T}{\boldsymbol{\delta}^T \boldsymbol{\delta}} - \frac{\boldsymbol{\delta} \, \boldsymbol{\gamma}^T \mathbf{B}^{old} + \mathbf{B}^{old} \boldsymbol{\gamma} \, \boldsymbol{\delta}^T}{\boldsymbol{\delta}^T \boldsymbol{\gamma}} \, , \tag{17}$$

with:

$$\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}^{old} \ \text{and} \ \boldsymbol{\gamma} = \mathbf{r} - \mathbf{r}^{old} \, . \tag{18}$$

The superscript *old* denotes quantities from the previous iteration cycle. The approximated inverse Hessian matrix is initialized by the inverse of the Hessian computed in the previous SCF cycle. As initial approximation for $\mathbf{B}$ in the very first SCF cycle the identity matrix can be used. After the step direction is determined by equation (15) the step length must be calculated. Because $F(\mathbf{x})$ is a convex quadratic function the line-search parameter α can be computed analytically as follows [46]:

$$x_{\bar{k}}^{new} = x_{\bar{k}} + \alpha \, \Delta x_{\bar{k}} \ \text{with} \ \min_{\alpha > 0} F(\mathbf{x} + \alpha \, \Delta\mathbf{x}) \, . \tag{19}$$

The superscript *new* denotes quantities for the following iteration cycle. Inserting (19) into equation (11) and deriving with respect to the line-search parameter α yields:

$$\frac{dF(\mathbf{x} + \alpha\,\Delta\mathbf{x})}{d\alpha} = \sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}}\,\Delta x_{\bar{k}}\,x_{\bar{l}} + \alpha \sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}}\,\Delta x_{\bar{k}}\,\Delta x_{\bar{l}} - \sum_{\bar{k}} J_{\bar{k}}\,\Delta x_{\bar{k}} \equiv 0\,. \tag{20}$$

Thus, it follows,

$$\alpha \sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}}\,\Delta x_{\bar{k}}\,\Delta x_{\bar{l}} = \sum_{\bar{k}}\left(J_{\bar{k}} - \sum_{\bar{l}} G_{\bar{k}\bar{l}}\,x_{\bar{l}}\right)\Delta x_{\bar{k}} = -\sum_{\bar{k}} r_{\bar{k}}\,\Delta x_{\bar{k}}\,, \tag{21}$$

and further:

$$\alpha = -\frac{\displaystyle\sum_{\bar{k}} r_{\bar{k}}\,\Delta x_{\bar{k}}}{\displaystyle\sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}}\,\Delta x_{\bar{k}}\,\Delta x_{\bar{l}}}\,. \tag{22}$$

With the determination of α a new set of Coulomb fitting coefficients are generated according to equation (19). With these coefficients a new residual is then calculated. Based on the norm of this residual the iteration is either stopped by the convergence criterion or continued with a new iteration step.

2.1.2 Exchange-Correlation Fitting Coefficients

As already mentioned, in the case of ADFT calculations a second set of fitting coefficients is needed. These so-called exchange-correlation fitting coefficients are obtained by equation (10). Because the Coulomb equation system (7) is solved first, a preconditioning matrix $\mathbf{B}$ for equation (10) is available. Thus, we can precondition equation (10) by:

$$\mathbf{B\,G\,z} = \mathbf{B\,L} \text{ with } \mathbf{B\,G} \approx \mathbf{E}\,. \tag{23}$$

The preconditioning of equation (10) results in a transformation of the inhomogeneous equation system as described in the appendix. The new exchange-correlation coefficients can be again calculated by a one-dimensional analytic line-search as:

$$z_{\bar{k}}^{new} = z_{\bar{k}} + \alpha\,\Delta z_{\bar{k}}\,. \tag{24}$$

The line-search parameter α is given by:

$$\alpha = -\frac{\displaystyle\sum_{\bar{k}} r_{\bar{k}}\,\Delta z_{\bar{k}}}{\displaystyle\sum_{\bar{k},\bar{l}} G_{\bar{k}\bar{l}}\,\Delta z_{\bar{k}}\,\Delta z_{\bar{l}}}\,. \tag{25}$$

Different to the calculation of the Coulomb fitting coefficients, $\Delta \mathbf{z}$ in equation (24) is obtained from a CG rather than quasi-Newton step:

$$\Delta z_{\bar{k}} = -s_{\bar{k}} + \beta \, \Delta z_{\bar{k}}^{old} \, . \tag{26}$$

Here $s_{\bar{k}}$ denotes a component of the preconditioned residual given by,

$$s_{\bar{k}} = \sum_{\bar{l}} B_{\bar{k}\bar{l}} \, r_{\bar{l}} \, , \tag{27}$$

where $r_{\bar{l}}$ is a component of the residual of equation (10):

$$r_{\bar{l}} = \sum_{\bar{k}} G_{\bar{l}\bar{k}} \, z_{\bar{k}} - L_{\bar{l}} \, . \tag{28}$$

The CG parameter β is given as:

$$\beta = -\frac{\displaystyle\sum_{\bar{k}} r_{\bar{k}} \, s_{\bar{k}}}{\displaystyle\sum_{\bar{k}} r_{\bar{k}}^{old} \, s_{\bar{k}}^{old}} \, . \tag{29}$$

It is set to zero in the first iteration step. Thus, despite the fact that the transformed preconditioned equation system is solved for the exchange-correlation fitting coefficients, the new exchange-correlation fitting coefficients can be calculated solely in terms of the original residual of equation (10) and the preconditioning matrix $\mathbf{B}$. No explicit transformation is necessary.

2.2 Implementation Details

In order to provide more insight into the implementation of the just described iterative fitting algorithms we now discuss their implementation in the LCGTO-DFT program de-Mon2k [47] employing the Message Passing Interface (MPI) and FORTRAN90. In Figure 1 the pseudo code for the calculation of the Coulomb fitting coefficients according to the quasi-Newton algorithm presented in (2.1.1) is depicted. This code fragment is executed in each SCF step after the diagonalization of the Kohn-Sham matrix. Thus, a set of new molecular orbital coefficients and, therefore, a new density matrix is available. On input, the Coulomb vector $\mathbf{J}$, calculated from the new density matrix, is provided to the iterative solver along with initial values for the Coulomb fitting coefficients $\mathbf{x}^{ini}$ and an initial guess for the approximated inverse auxiliary function Coulomb matrix $\mathbf{B}^{ini}$. As initial fitting coefficients the converged Coulomb fitting coefficients from the previous SCF cycle are used.

In case of the very first SCF cycle they are initialized to zero. Similar, as initial guess for $\mathbf{B}$ the converged solution of the previous SCF cycle is used. In case of a geometry optimization the initial $\mathbf{B}$ matrix for the first SCF of a new geometry is taken from the last SCF step of the previous structure. For the very first SCF the $\mathbf{B}$ matrix can be initialized as the unit matrix. In parallel runs each thread holds a copy of the fitting coefficients and the Coulomb vector. However, the $\mathbf{B}$ matrix exist only in distributed form. Because only a triangular part is used the rows of $\mathbf{B}$ have different physical lengths. Nevertheless, the matrix is distributed such that the number of elements on each thread is by and large the same. As a result, one thread may hold a few long rows of the triangular $\mathbf{B}$ matrix, whereas another may hold many short rows. A similar distribution scheme is applied to the Coulomb matrix $\mathbf{G}$ which also exist only in distributed form.

As Figure 1 shows each quasi-Newton iteration for the calculation of the Coulomb fitting coefficients consist of three principal parts. In the first one the residual is calculate with the current fitting coefficients. The necessary matrix-vector multiplication is performed as a series of vector-vector operations because the matrix $\mathbf{G}$ is loaded row-wise from disk in this step. In parallel runs each thread calculates a portion of the result vector. Afterwards, each thread sends its result to all others. Thus, at the end a complete copy of the result vector and, therefore, of the residual is available on each parallel thread. In order to check the convergence of the residual two tests are performed. The first one is based on the normalized length of the residual,

$$ r = \frac{|\,\mathbf{r}\,|}{M} \,, \tag{30} $$

where M denotes the number of auxiliary functions, *i.e.* the number of Coulomb fitting coefficients. In the first two SCF cycles the corresponding convergence threshold ε_r is set to 10^{-5}. In the following SCF cycles this threshold is calculated according to the formula:

$$ \varepsilon_r = \frac{\varepsilon_0}{10 - \min\left(\log(\varepsilon_{SCF}/\varepsilon_{CON}), 9\right)} \tag{31} $$

Here ε_{SCF} and ε_{CON} refer to the current SCF energy error and the requested SCF energy convergence (default is 10^{-5} a.u.), respectively. For the calculation of the Coulomb fitting coefficients ε_0 is set to $\varepsilon_{CON}/100$. According to (31) this represents also the upper bound for ε_r. The use of (31) guarantees that the convergence of the iterative solution for the Coulomb fitting coefficients adapts automatically to the convergence of the SCF solution. The second convergence test is based on the spread of the current residual around the

residuals of the last six iterations,

$$\sigma = \sum_{i=1}^{6} (r - r_i)^2 \, , \tag{32}$$

where r_i denote residuals from previous iterations. The corresponding convergence threshold ε_σ is set to 10^{-8} in the first two SCF cycles, to 10^{-9} in the following two and finally to 10^{-10} in all other SCF cycles.

The second part in the quasi-Newton iteration of the Coulomb fitting coefficients represents the update of the approximated inverse Coulomb matrix $\mathbf{B}$. The calculation of $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ as well as the inverse BFGS update is implemented according to the formulas in (2.1.1). In a parallel run $\mathbf{B}$ is distributed over all threads and, thus, each thread only updates its portion of $\mathbf{B}$. Because $\mathbf{r}$ and $\mathbf{x}$ is available to all threads no communication arises.

The last part of the quasi-Newton iteration represents the calculation of a new set of Coulomb fitting coefficients. In parallel runs the step direction vector $\Delta\mathbf{x}$ is calculated in portions on the different threads according to the distribution of the $\mathbf{B}$ matrix. They are then combined and a complete step direction vector is copied to each thread. A similar strategy is applied for the calculation of the denominator of the line-search parameter α in equation (22). A new set of Coulomb fitting coefficients is then calculated on each thread. With this calculation the iterative loop closes as shown in Figure 1. After convergence is reached the updated approximated inverse of $\mathbf{G}$, $\mathbf{B}^{upd}$, is written to disk. As the above discussion shows only auxiliary function vectors are copied between threads in parallel runs despite the fact that the auxiliary function matrices $\mathbf{B}$ and $\mathbf{G}$ are distributed over all threads. As a result, the parallel implementation of the iterative Coulomb fitting algorithm is not communication intensive.

In Figure 2 the pseudo code for the calculation of the exchange-correlation fitting coefficients according to the preconditioned CG algorithm described in (2.1.2) is depicted. This code fragment is intrinsic to ADFT calculations and is executed directly after the above described iterative Coulomb fitting. On input, the exchange-correlation vector $\mathbf{L}$ and the updated matrix $\mathbf{B}^{upd}$ from the Coulomb fitting, which now acts as preconditioner, are provided along with an initial set of exchange-correlation fitting coefficients $\mathbf{z}^{ini}$. As for the Coulomb fitting coefficient vector $\mathbf{z}^{ini}$ is taken from the last SCF or, in case of the very first SCF cycle, initialized to zero. In parallel runs each thread holds an own copy of the exchange-correlation fitting coefficients. The $\mathbf{B}$ and $\mathbf{G}$ matrices are distributed as described for the

calculation of the Coulomb fitting coefficients.

As Figure 2 shows the calculation of the exchange-correlation fitting coefficients with the preconditioned CG method consists of two principal parts, namely the calculation of the residual and the corresponding convergence test, and the calculation of a new set of fitting coefficients. The residual calculation is alike to the one for the Coulomb fitting coefficients. The convergence check for the exchange-correlation fitting coefficients is based on the same two convergence thresholds, ε_r and ε_σ, that were introduced for the Coulomb fitting coefficients. The only difference is the value of ε_0 in formula (31) that is set to $\varepsilon_{CON}/10$ for the exchange-correlation fitting coefficient convergence.

The second part in the preconditioned CG iteration represents the calculation of a new set of exchange-correlation fitting coefficients. It starts with the calculation of the preconditioned residual $\mathbf{s}$. This step is computational identical to the step direction calculation in the quasi-Newton algorithm (see Figure 1). Therefore, the same implementation is used, *i.e.* in parallel runs $\mathbf{s}$ is calculated in portions on the different threads according to the distribution of the $\mathbf{B}$ matrix and then combined and copied to all threads. Because the $\mathbf{r}$ and $\mathbf{s}$ vectors are available to all threads the calculation of β according to formula (29) and, thus, the calculation of the CG step direction $\Delta\mathbf{z}$ is straightforward. The calculation of the line-search parameter α for the preconditioned CG algorithm is computationally identical to the α calculation in the quasi-Newton algorithm. Thus, the same implementation is used. Altogether, the calculation of the new exchange-correlation fitting coefficients follows a very similar strategy as for the Coulomb fitting coefficients. In particular, only auxiliary function vectors are copied between threads in parallel runs, too.

3. Computational Details

All validation and benchmark calculations were performed with the DFT program deMon2k [47]. For the validation the alkane molecules, $C_{16}H_{34}$, $C_{50}H_{102}$, $C_{100}H_{202}$ and $C_{150}H_{302}$, were used. For the benchmark calculations C_{60} and three three-dimensional ZSM5 zeolite structures with one, two and three unit cells were employed. The stoichiometry of the zeolite systems is $Si_{96}O_{200}H_{80}$, $Si_{192}O_{400}H_{128}$ and $Si_{288}O_{600}H_{176}$, respectively. The Coulomb energy was calculated by the variational fitting procedure as described above. The obtained auxiliary density was used for the calculation of the exchange-correlation energy and po-

tential, too. These energies and potentials were numerically integrated on an adaptive grid [48]. The grid accuracy was set to 10^{-5} a.u. in all calculations. Thus, all calculations were performed in the framework of ADFT [16]. The structure optimizations were performed with the local density approximation (LDA) employing Dirac exchange [49] in combination with the correlation functional of Vosko, Wilk and Nusair (VWN) [50]. DFT optimized all-electron double zeta valence plus polarization (DZVP) basis sets [51] were employed. For the fitting of the density the auxiliary function set A2 was used. A restricted step quasi-Newton method in internal redundant coordinates with analytic energy gradients was used for the structure optimization [52,53]. The convergence was based on the Cartesian gradient and displacement vectors with a threshold of 10^{-4} and 10^{-3} a.u., respectively. The test calculations were performed in serial as well as in parallel [20,54, 55] with deMon2k.

4.   Results and Discussion

In Table I results from alkane structure optimizations employing direct and iterative fitting procedures are compared. The direct fit refers to the calculation of $\mathbf{G}^{-1}$ according to [38] whereas the iterative fit refers to the iterative solvers described in (2.2). The calculation are performed in serial using a 2.4 GHz single core AMD Opteron$^{\text{TM}}$ processor. The data in Table I refer to the second step of the structure optimization and are representative for the other steps, too. The number of auxiliary functions ranges from 680 to 6308 in this test set. The reported CPU times refer to one optimization step, *i.e.* they correspond to a single point energy and gradient calculation. As the comparison between the direct and iterative fitting CPU times show the savings due to the use of the iterative solvers are little for these systems. For small systems with only a few thousands auxiliary functions such a result is expected because the direct approach relies heavily on well optimized linear algebra routines. In fact it is entirely possible that for very small systems the direct fitting method is computationally superior to the iterative approach, particularly if many SCF cycles are needed. However, the main purpose of Table I is a validation rather than performance analysis of the iterative solver for the fitting coefficients. The comparison of the direct and iterative single point energies demonstrate the accuracy and robustness of the iterative fit. Energy differences are in the range of 10^{-5} a.u. or below. This is in accordance with the

SCF energy convergence criterion of 10^{-5} a.u. and holds for all optimization steps. As a result the optimized structure parameters are virtually indistinguishable. A more detailed analysis reveals that the energies obtained from iterative fits are always below the once from direct fits. This is not by chance. In fact, the iterative solution of the Coulomb fitting equation system introduces an error in the approximated Coulomb energy which effects the lower bound of the MinMax SCF [38]. As a result lower energies are found. The difference between the iterative and direct energies are always in the range of the SCF energy convergence criterion due to (31). Thus, they can be made arbitrarily small.

As already discussed the alkane systems in Table I are too small to appreciate the computational benefits of the iterative solvers for the fitting coefficients. For this reason we have performed benchmark calculations on zeolite models with up to about 35,000 auxiliary functions. In order to provide a common reference point for the direct and iterative fitting coefficient calculations we included C_{60} in the benchmark set, too. The structures of the benchmark systems along with their stoichiometric formulas are depicted in Figure 3. The larger ones reach well into the nanometric length scale. In Figure 4 the CPU times for the direct and iterative calculation of the fitting coefficients are plotted against the number of auxiliary functions on a double logarithmic scale. These benchmark calculations were performed in parallel employing 10 2.4 GHz single core AMD Opteron$^{\text{TM}}$ processors. This figure shows that the direct calculation of the fitting coefficients scales cubic with respect to the number of auxiliary functions. Due to the linear algebra steps involved and the fact that the Coulomb matrix inversion is performed only one time per energy calculation this behavior is expected. In the case of the iterative calculation of the fitting coefficients the number of SCF cycles influence the timings because the iterative solvers are called in each cycle. As a result a system independent scaling cannot be expected. However, independently from this behavior Figure 4 clearly demonstrates the computational benefits associated with the iterative calculations of the fitting coefficients. The computational savings increase dramatically with system size. Moreover, for the largest system, $Si_{288}O_{600}H_{176}$, the inversion of the Coulomb matrix consumes about 40 % of the total CPU time for a single point energy calculation. Thus, the use of the iterative solvers decrease considerably the overall CPU time in this case, too. The benchmark calculations in Figure 4 needed on average around 20 SCF cycles for convergence. Thus, for systems with such a SCF convergence behavior a sub-quadratic scaling of around $\mathcal{O}^{1.7}$ for the iterative calculation of the fitting coefficients

with respect to the number of auxiliary functions is observed.

5. Conclusions

In this paper an alternative approach for the solution of the inhomogeneous equation system(s) associated to density fitting is presented. It is based on the iterative quasi-Newton and preconditioned conjugate gradient algorithms. A detailed description of the parallel implementations of these algorithms is given. It is shown that the resulting parallel code is not communication intensive despite the fact that the auxiliary function matrices are distributed over all threads. The validation calculations on alkanes demonstrate the accuracy and robustness of the new iterative solvers for the calculation of the fitting coefficients. It is shown that the error introduced by the iterative solution of the fitting equation systems is directly coupled to the SCF energy convergence. As a result optimized structures and energies from the direct and iterative fitting approaches are virtually indistinguishable. The parallel benchmark calculations show that significant computational savings are obtained by the here proposed iterative density fitting for systems with more than 10,000 auxiliary functions. Example calculations with up to about 35,000 auxiliary functions indicate a sub-quadratic scaling for the iterative density fitting of around $\mathcal{O}^{1.7}$ with respect to the number of auxiliary functions. We estimate that systems with more than 100,000 auxiliary functions can be tackled with our current implementation on appropriate computational architectures. For such large systems a limited memory quasi-Newton updating would be desirable. Work in this direction is underway in our laboratory.

Appendix: Preconditioned Conjugate Gradient Transformation

The preconditioning of equation (10) with the approximate inverse, $\mathbf{B}$, of the Coulomb matrix $\mathbf{G}$ yields the transformed equation system (23):

$$\mathbf{B}\,\mathbf{G}\,\mathbf{z} = \mathbf{B}\,\mathbf{L}\,.$$

Because $\mathbf{B}$ is symmetric and positive definite by construction the Cholesky factorization $\mathbf{B} = \mathbf{C}\,\mathbf{C}^T$ is always possible. Using this factorization we can transform the above equation to:

$$\mathbf{C}\,\mathbf{C}^T\,\mathbf{G}\,\mathbf{z} = \mathbf{C}\,\mathbf{C}^T\,\mathbf{L}\,. \tag{33}$$

Multiplication of $\mathbf{C}^{-1}$ from the left-hand side and introduction of the identity $\mathbf{C}\,\mathbf{C}^{-1}$ between $\mathbf{G}$ and $\mathbf{z}$ then yields:

$$\mathbf{C}^T\,\mathbf{G}\,\mathbf{C}\,\mathbf{C}^{-1}\,\mathbf{z} = \mathbf{C}^T\,\mathbf{L} \;\Rightarrow\; \breve{\mathbf{G}}\,\mathbf{y} = \breve{\mathbf{L}}\,. \tag{34}$$

From now on we will use a breve to indicate transformed quantities. The new fitting coefficient set $\mathbf{y}$ is given by $\mathbf{C}^{-1}\,\mathbf{z}$. For the solution of the preconditioned transformed equation system (34) the CG method [34] is employed. The residual of this equation system is given by:

$$\breve{r}_{\bar{k}} = \sum_{\bar{l}} \breve{G}_{\bar{k}\bar{l}}\,y_{\bar{l}} - \breve{L}_{\bar{k}}\,. \tag{35}$$

From this residual the CG step direction is calculated as:

$$\Delta y_{\bar{k}} = -\breve{r}_{\bar{k}} + \beta\,\Delta y_{\bar{k}}^{old}\,. \tag{36}$$

The parameter β is determined from the conjugate direction criterion:

$$\sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}}\,\Delta y_{\bar{k}}^{old}\,\Delta y_{\bar{l}} = 0\,. \tag{37}$$

Inserting (36) into (37) then yields:

$$\sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}}\,\Delta y_{\bar{k}}^{old}\,\breve{r}_{\bar{l}} = \beta\,\sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}}\,\Delta y_{\bar{k}}^{old}\,\Delta y_{\bar{l}}^{old}\,. \tag{38}$$

Thus, we obtain for β:

$$\beta = \frac{\displaystyle\sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}}\,\Delta y_{\bar{k}}^{old}\,\breve{r}_{\bar{l}}}{\displaystyle\sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}}\,\Delta y_{\bar{k}}^{old}\,\Delta y_{\bar{l}}^{old}}\,. \tag{39}$$

As already discussed for the Coulomb fitting coefficients the solution of (34) is equivalent to the minimization of a quadratic function. Therefore, an exact line-search can be be applied for the step length calculation. The new coefficients are then given by:

$$y_{\bar{k}}^{new} = y_{\bar{k}} + \alpha \, \Delta y_{\bar{k}} \, . \tag{40}$$

The corresponding new residual takes the form:

$$\breve{r}_{\bar{k}}^{new} = \sum_{\bar{l}} \breve{G}_{\bar{k}\bar{l}} \, y_{\bar{l}}^{new} - \breve{L}_{\bar{k}} = \breve{r}_{\bar{k}} + \alpha \, \sum_{\bar{l}} \breve{G}_{\bar{k}\bar{l}} \, \Delta y_{\bar{l}} \, . \tag{41}$$

From the zero slope condition of the exact line-search,

$$\sum_{\bar{k}} \breve{r}_{\bar{k}}^{new} \, \Delta y_{\bar{k}} = 0 \, , \tag{42}$$

follows:

$$\sum_{\bar{k}} \breve{r}_{\bar{k}} \, \Delta y_{\bar{k}} + \alpha \, \sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}} \, \Delta y_{\bar{k}} \, \Delta y_{\bar{l}} = 0 \, . \tag{43}$$

Thus, we obtain for α:

$$\alpha = - \frac{\displaystyle\sum_{\bar{k}} \breve{r}_{\bar{k}} \, \Delta y_{\bar{k}}}{\displaystyle\sum_{\bar{k},\bar{l}} \breve{G}_{\bar{k}\bar{l}} \, \Delta y_{\bar{k}} \, \Delta y_{\bar{l}}} \, . \tag{44}$$

As for the calculation of the Coulomb fitting coefficients the iterative cycle of the pre-conditioned CG method for the calculation of the **y** fitting coefficients finishes with the determination of α and the calculation of the new fitting coefficients by equation (40).

A drawback of the above formulation is the need of the transformation matrix **C** for the transformed quantities and the back-transformation of the **y** fitting coefficients. In order to avoid the explicit use of the transformation matrix **C** the expression for β must be further simplified. To do so we now employ the zero slope condition (42) once again, however now formulated for the current residual. Thus, we obtain:

$$\sum_{\bar{l}} \breve{G}_{\bar{k}\bar{l}} \, \Delta y_{\bar{l}}^{old} = \frac{\breve{r}_{\bar{k}} - \breve{r}_{\bar{k}}^{old}}{\alpha} \, . \tag{45}$$

Inserting this expression into the formula for the β parameter (39) yields:

$$\beta = \frac{\displaystyle\sum_{\bar{k}} \left(\breve{r}_{\bar{k}} - \breve{r}_{\bar{k}}^{old} \right) \breve{r}_{\bar{k}}}{\displaystyle\sum_{\bar{k}} \left(\breve{r}_{\bar{k}} - \breve{r}_{\bar{k}}^{old} \right) \Delta y_{\bar{k}}^{old}} \, . \tag{46}$$

Substituting,

$$\Delta y_{\bar{k}}^{old} = -\check{r}_{\bar{k}}^{old} + \beta \, \Delta y_{\bar{k}}^{older} \,, \tag{47}$$

with $\Delta y_{\bar{k}}^{older}$ referring to the change in the **y** fitting coefficients in the second to last iteration step, into (46) and employing once again the zero slope condition (42) results in the following form of β:

$$\beta = \frac{\displaystyle\sum_{\bar{k}} \left(\check{r}_{\bar{k}} - \check{r}_{\bar{k}}^{old} \right) \check{r}_{\bar{k}}}{\displaystyle\sum_{\bar{k}} \check{r}_{\bar{k}}^{old} \, \check{r}_{\bar{k}}^{old}} \,. \tag{48}$$

In case the preconditioned CG iteration is started with a steepest descent step, *i.e.* β is set to zero in the first iteration, and the function is quadratic, as in our case, it can be shown that the residuals are orthogonal. Under these conditions, the form of β simplifies further to:

$$\beta = \frac{\displaystyle\sum_{\bar{k}} \check{r}_{\bar{k}} \, \check{r}_{\bar{k}}}{\displaystyle\sum_{\bar{k}} \check{r}_{\bar{k}}^{old} \, \check{r}_{\bar{k}}^{old}} \,. \tag{49}$$

With this form of β and the following relationships,

$$\check{\mathbf{r}} = \mathbf{C}^T \mathbf{r} \quad \text{with} \quad \mathbf{r} = \mathbf{G}\,\mathbf{z} - \mathbf{L} \tag{50}$$

$$\mathbf{s} = \mathbf{C}\,\check{\mathbf{r}} \quad \Longrightarrow \quad \mathbf{s} = \mathbf{B}\,\mathbf{r} \tag{51}$$

$$\Delta\mathbf{z} = \mathbf{C}\,\Delta\mathbf{y} \quad \Longrightarrow \quad \Delta\mathbf{z} = -\mathbf{s} + \beta\,\Delta\mathbf{z}^{old} \,, \tag{52}$$

the explicit transformation of $\mathbf{G}$, $\mathbf{L}$ and $\mathbf{z}$ can be avoided. The parameters α and β can then be expressed in terms of the original quantities, too, as given by the equations (25) and (29). Thus, the transformation matrix $\mathbf{C}$ is never invoked.

References

[1] P. Hohenberg, W. Kohn, Phys. Rev. B864 (1964).

[2] W. Kohn, L.J. Sham, Phys. Rev. **140**, A1133 (1965).

[3] J.L. Whitten, J. Chem. Phys. **58**, 4496 (1973).

[4] E.J. Baerends, D.E. Ellis, P. Ros, Chem. Phys. **2**, 41 (1973).

[5] H. Sambe, R.H. Felton, J. Chem. Phys. **62**, 1122 (1975).

[6] B.I. Dunlap, J.W.D. Connolly, J.R. Sabin, J. Chem. Phys. **71**, 4993 (1979).

[7] B.I. Dunlap, J. Mol. Struct. (THEOCHEM) **529**, 37 (2000).

[8] R. Fournier, J. Andzelm, D.R. Salahub, J. Chem. Phys. **90**, 6371 (1989).

[9] R. Fournier, J. Chem. Phys. **92**, 5422 (1990).

[10] A. Komornicki, G. Fitzgerald, J. Chem. Phys. **118**, 1398 (1993).

[11] J.W. Mintmire, B.I. Dunlap, Phys. Rev. A **25**, 88 (1982).

[12] J.W. Mintmire, J.R. Sabin, S.B. Trickey, Phys. Rev. B **26**, 1743 (1982).

[13] O. Vahtras, J. Almlöf, M.W. Feyereisen, Chem. Phys. Lett. **213**, 514 (1993).

[14] D.N. Laikov, Chem. Phys. Lett. **281**, 151 (1997).

[15] A.M. Köster, *Entwicklung einer LCGTO-Dichtefunktionalmethode mit Hilfsfunktionen*, Habilitation thesis, Universität Hannover (1998).

[16] A.M. Köster, J.U. Reveles, J.M. del Campo, J. Chem. Phys. **121**, 3417 (2004).

[17] U. Birkenheuer, A.B. Gordienko, V.A. Nasluzov, M.K. Fuchs-Rohr, N. Rösch, Int. J. Quantum Chem. **102**, 743 (2005).

[18] L. Belpassi, F. Tarantelli, A. Sgamellotti, H.M. Quiney, J. Chem. Phys. **124**, 124104 (2006).

[19] R.R. Zope, B.I. Dunlap, J. Chem. Phys. **124**, 044107 (2006).

[20] G. Geudtner, F. Janetzko, A.M. Köster, A. Vela, P. Calaminici, J. Comp. Chem. **27**, 483 (2006).

[21] V. Dominguez-Soria, P. Calaminici, A. Goursot, J. Chem. Phys. **127**, 154710 (2007).

[22] P. Calaminici, G. Geudtner, A.M. Köster, J. Comput. Theor. Chem. **5**, 29 (2009)

[23] R. McWeeny, Phys. Rev. **126**, 1028 (1962).

[24] J.L. Dodds, R. McWeeny, W.T. Raynes, J.P. Riley, Mol. Phys. **33**, 611 (1977).

[25] R. McWeeny, *Methods of Molecular Quantum Mechanics*, Second reprinting (Academic Press, London, 2001).

[26] R. Flores-Moreno, A.M. Köster, J. Chem. Phys. **128**, 134105 (2008).

[27] B.I. Dunlap, J. Chem. Phys. **129**, 244109 (2008).

[28] A.M. Köster, A. Goursot, D.R. Salahub, in *Comprehensive Coordination Chemistry–II: From Biology to Nanotechnology*, Vol. 1, Chapter 2.57, pp. 681, Editors: J. McCleverty, T.J. Meyer, B. Lever, (Elsevier, Amsterdam, 2003).

[29] R.T. Gallant, A. St.-Amant, Chem. Phys. Lett. **256**, 569 (1996).

[30] C. Fonseca Guerra, J.G. Snijders, G. te Velde, E.J. Baerends, Theor. Chem. Acc. **99**, 391 (1998).

[31] A. Sodt, J.E. Subotnik, M. Head-Gordon, J. Chem. Phys. **125**, 194109 (2006).

[32] P. Sałek, S. Høst, L. Thøgersen, P. Jørgensen, P. Manninen, J. Olsen, B. Jansik, S. Reine, F. Powlowski, E. Tellgren, T. Helgaker, S. Coriani, J. Chem. Phys. **126**, 114110 (2007).

[33] S. Reine, E. Tellgren, A. Krapp, T. Kjærgaard, T. Helgaker, B. Jansik, S. Høst, P. Sałek, J. Chem. Phys. **129**, 104101 (2008).

[34] M.R. Hestenes, E. Stiefel, J. Res. N.B.S. **49**, 409 (1952).

[35] J.L. Morales, J. Nocedal, SIAM J. Optim. **10**, 1079 (2000).

[36] J.C. Boettger, S.B. Trickey, Phys. Rev. B **53**, 3007 (1996).

[37] A.M. Köster, J. Chem. Phys. **118**, 9943 (2003).

[38] A.M. Köster, J.M. del Campo, F. Janetzko, B. Zuniga-Gutierrez, J. Chem. Phys. **130**, 114106 (2009).

[39] W.C. Davidon, *Variable metric method for minimization*, Technical Report ANL-5990 (revised) (Argonne National Laboratory, Argonne, Il, 1959).

[40] W.C. Davidon, SIAM J. OPTIM. **1**, 1 (1991).

[41] R. Fletcher, *Practical Methods of Optimization*, Vol. 1, pp. 38 (J. Wiley & Sons, New York, 1980).

[42] C.G. Broyden, J. Inst. Math. Appl. **6**, 76 (1970).

[43] R. Fletcher, Comp. J. **13**, 317 (1970).

[44] D. Goldfarb, Math. Comput. **24**, 23 (1970).

[45] D.F. Shanno, Math. Comput. **24**, 647 (1970).

[46] J. Nocedal, S.J. Wright, *Numerical Optimization*, pp. 47 (Springer-Verlag, New York, 1999).

[47] A.M. Köster, P. Calaminici, R. Flores-Moreno, G. Geudtner, A. Goursot, T. Heine, F. Janetzko, S. Patchkovskii, J.U. Reveles, A. Vela and D.R. Salahub, deMon2k, The deMon developers, Cinvestav, Mexico-City, 2004.

[48] A.M. Köster, R. Flores-Moreno, J.U. Reveles, J. Chem. Phys. **121**, 681 (2004).

[49] P.A.M. Dirac, Proc. Camb. Phil. Soc. **26**, 376 (1930).

[50] S.H. Vosko, L. Wilk, M. Nusair, Can. J. Phys. **58**, 1200 (1980).

[51] N. Godbout, D.R. Salahub, J. Andzelm, E. Wimmer, Can. J. Phys. **70**, 560 (1992).

[52] J.U. Reveles, A.M. Köster, J. Comput. Chem. *25*, 1109 (2004).

[53] J.M. del Campo, A.M. Köster, J. Chem. Phys. **129**, 024107 (2008).

[54] P. Calaminici, V.D. Domínguez-Soria, G. Geudtner, E. Hernández-Marín, A.M. Köster, Revista Superficies y Vacío **18**, 1 (2005).

[55] P. Calaminici, V.D. Domínguez-Soria, G. Geudtner, E. Hernández-Marín, A.M. Köster, Theor. Chem. Acc. **115**, 221 (2006)

Table I: Number of auxiliary functions, direct and iterative energy [a.u.], energy difference [a.u.] and direct and iterative CPU times [seconds] for an optimization step of $C_{16}H_{34}$, $C_{50}H_{102}$, $C_{100}H_{202}$ and $C_{150}H_{302}$. All calculations were performed in serial on a 2.4 GHz AMD Opteron^{TM} processor.

System	Auxiliary Functions	Direct Energy	Iterative Energy	ΔE	Direct Time	Iterative Time
$C_{16}H_{34}$	680	-624.312335	-624.312346	1×10^{-5}	62	60
$C_{50}H_{102}$	2108	-1948.362468	-1948.362472	4×10^{-6}	1038	1015
$C_{100}H_{202}$	4208	-3895.425677	-3895.425681	4×10^{-6}	4793	4377
$C_{150}H_{302}$	6308	-5842.439588	-5842.439593	5×10^{-6}	12988	11123

Figure Captions

Fig. 1   Pseudo code of the quasi-Newton algorithm for the calculation of the Coulomb fitting coefficients.

Fig. 2   Pseudo code of the preconditioned CG algorithm for the calculation of the exchange-correlation fitting coefficients.

Fig. 3   Structures and stoichiometric formulas of the benchmark systems.

Fig. 4   Double logarithmic plot of fitting CPU time vs. number of auxiliary functions. The calculations are performed in parallel on 10 2.4 GHz single core AMD Opteron$^{\text{TM}}$ processors.

START PSEUDO CODE

input: $\mathbf{J}$, $\mathbf{x}^{ini}$, $\mathbf{B}^{ini}$

start loop:

    calculate residual $\mathbf{r} = \mathbf{G}\,\mathbf{x} - \mathbf{J}$

    test convergence of $\mathbf{r}$

    exit loop if convergence test was successful

    calculate $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$

    update $\mathbf{B}$

    calculate step direction $\Delta\mathbf{x} = -\mathbf{B}\,\mathbf{r}$

    calculate line search parameter α

    calculate new Coulomb fitting coefficients $\mathbf{x}^{new} = \mathbf{x} + \alpha\,\Delta\mathbf{x}$

next loop cycle

output: $\mathbf{B}^{upd}$, $\mathbf{x}^{new}$

END PSEUDO CODE

Figure 1: V.D. Domínguez-Soria, G. Geudtner, J.L. Morales, P. Calaminici, A.M. Köster

START PSEUDO CODE

input: $\mathbf{L}$, $\mathbf{z}^{ini}$, $\mathbf{B}^{upd}$

start loop:

    calculate residual $\mathbf{r} = \mathbf{G}\,\mathbf{z} - \mathbf{L}$

    test convergence of $\mathbf{r}$

    exit loop if convergence test was successful

    calculate preconditioned residual $\mathbf{s} = \mathbf{B}\,\mathbf{r}$

    calculate CG parameter β

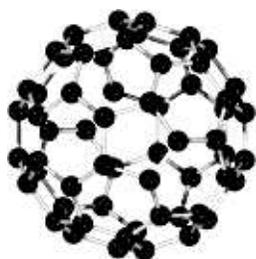   calculate CG step direction $\Delta\mathbf{z} = -\mathbf{s} + \beta\,\Delta\mathbf{z}^{old}$

    calculate line-search parameter α

    calculate new exchange-correlation fitting coefficients $\mathbf{z}^{new} = \mathbf{z} + \alpha\,\Delta\mathbf{z}$
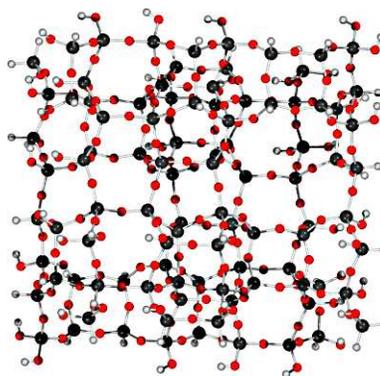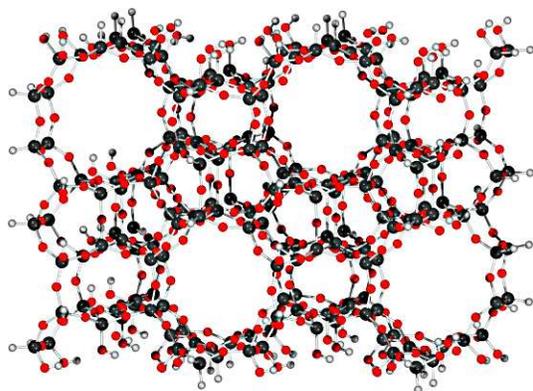
next loop cycle

output: $\mathbf{z}^{new}$

END PSEUDO CODE

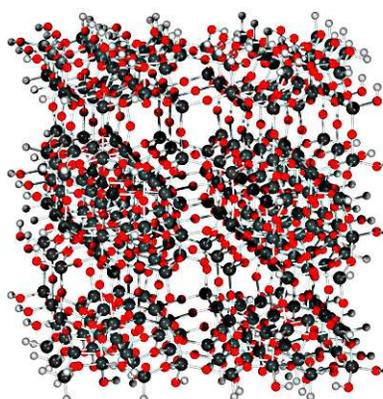Figure 2: V.D. Domínguez-Soria, G. Geudtner, J.L. Morales, P. Calaminici, A.M. Köster

C_{60}

$Si_{96}O_{200}H_{80}$

$Si_{192}O_{400}H_{128}$

$Si_{288}O_{600}H_{176}$

Figure 3: V.D. Domínguez-Soria, G. Geudtner, J.L. Morales, P. Calaminici, A.M. Köster
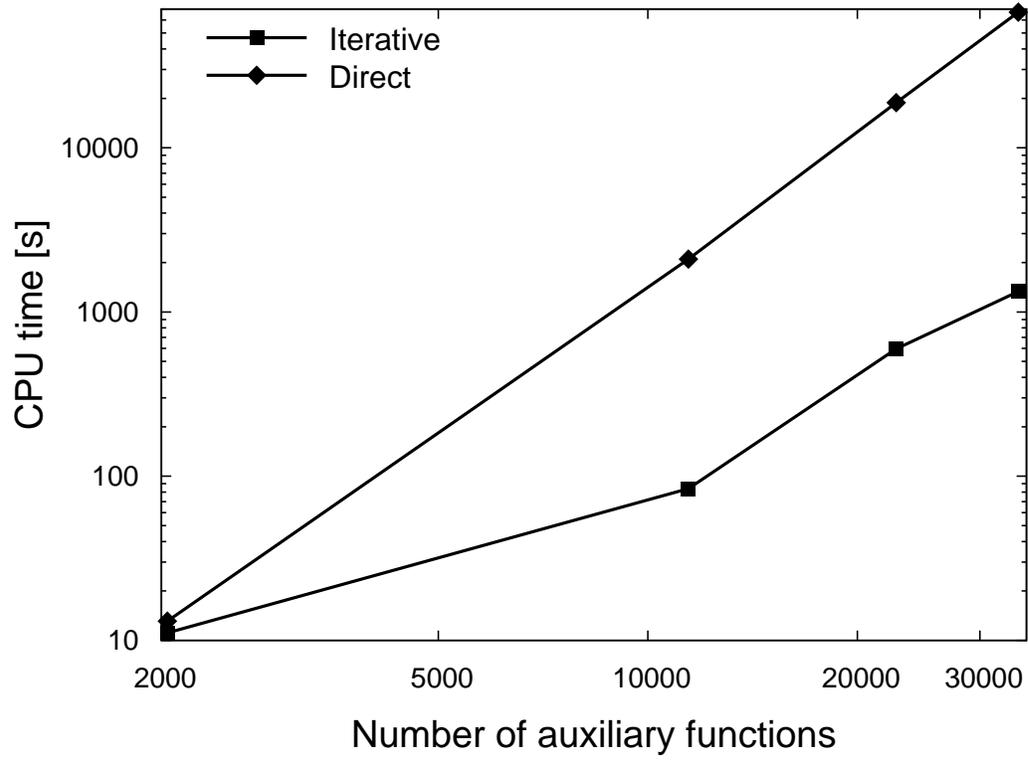
Figure 4: V.D. Domínguez-Soria, G. Geudtner, J.L. Morales, P. Calaminici, A.M. Köster