An efficient density-functional-theory force evaluation for large molecular systems

Simen Reine,[1,a)] Andreas Krapp,[1] Maria Francesca Iozzi,[1] Vebjørn Bakken,[1] Trygve Helgaker,[1] Filip Pawłowski,[2,b)] and Pawel Sałek[3,c)]
[1]*Department of Chemistry, Centre for Theoretical and Computational Chemistry, University of Oslo, P.O. Box 1033 Blindern, N-0315 Oslo, Norway*
[2]*Physics Institute, Kazimierz Wielki University, pl. Weyssenhoffa 11, 85-072 Bydgoszcz, Poland*
[3]*Laboratory of Theoretical Chemistry, The Royal Institute of Technology, Teknikringen 30, Stockholm SE-10044, Sweden*

(Received 14 April 2010; accepted 11 June 2010; published online 22 July 2010)

An efficient, linear-scaling implementation of Kohn–Sham density-functional theory for the calculation of molecular forces for systems containing hundreds of atoms is presented. The density-fitted Coulomb force contribution is calculated in linear time by combining atomic integral screening with the continuous fast multipole method. For higher efficiency and greater simplicity, the near-field Coulomb force contribution is calculated by expanding the solid-harmonic Gaussian basis functions in Hermite rather than Cartesian Gaussians. The efficiency and linear complexity of the molecular-force evaluation is demonstrated by sample calculations and applied to the geometry optimization of a few selected large systems. © *2010 American Institute of Physics*. [doi:10.1063/1.3459061]

I. INTRODUCTION

Over the past decade, density-functional theory (DFT) has been applied to larger and larger molecular systems, following the development of new computational techniques and the emergence of more powerful computers. Within the framework of Kohn–Sham theory, molecular energies can now be efficiently optimized for systems containing hundreds of atoms, at a cost that makes the quantum-mechanical study of such molecules meaningful. However, for such studies to be truly practical, methods must also be developed for the efficient evaluation of the forces acting on the atoms (the molecular gradient) of such molecules, needed to locate equilibrium and transition-state structures and for molecular dynamics.[1] In this paper, we present an efficient, linear-scaling evaluation of molecular gradients for large molecules, within the framework of pure (non-hybrid) Kohn–Sham DFT. Linear scaling is achieved by combining screening[2] with the continuous fast multipole method (CFMM),[3] while efficiency is ensured by combining density-fitting techniques[2,4] with the solid-harmonic Hermite scheme in Ref. 5 for differentiation.

A density-fitting molecular gradient code was first reported by Versluis and Ziegler[6] within the least-squares-fit Slater-type-orbital formulation. Later, a robust and variational density-fitting gradient code was presented by Fournier *et al.*[7] using the conventional density-fitting formulation in Ref. 8. Regular (non-density-fitting) molecular gradients within the Gaussian fast multipole method formulation were reported by Burant *et al.*[9] and later by Shao *et al.*,[10] who also reported a *J*-engine-based Coulomb gradient formulation. An efficient density-fitted excited-state gradient implementation was presented by Rappoport and Furche.[11] Density-fitted gradient calculations of zeolite models containing up to 1064 atoms have been reported by Domínguez-Soria *et al.*[12] and on fullerenes containing 540 carbon atoms by Calaminici *et al.*[13] To the best of our knowledge, however, linear-scaling density-fitting force evaluations such as those presented by us here have not previously been reported in the literature, although we note that TURBOMOLE allows an $N \log N$-scaling multipole-accelerated resolution-of-the-identity Coulomb gradient evaluation.

In this paper, the construction of the density-fitted Coulomb force contributions is accelerated using the McMurchie–Davidson *J*-engine integral scheme presented in Ref. 14, combined with integral screening[2] and multipole methods.[3,9,10] This approach allows a linear-scaling computation of the contributions to the gradient, while avoiding any explicit construction and storage of differentiated two- and three-center integrals. The evaluation of near-field contributions to the density-fitting Coulomb force is accelerated by using the solid-harmonic Hermite scheme in Ref. 5, thereby reducing the cost of the gradient evaluation and simplifying its implementation. The molecular gradient presented in this paper has been implemented within the DALTON program package for pure Kohn–Sham functionals; the exact-exchange contribution to the gradient was not discussed and has not been implemented.

This paper is organized as follows. First, in Sec. II, we discuss our molecular gradient implementation and give a brief report of the geometry optimization algorithm em-

[a)]Electronic mail: simen.reine@kjemi.uio.no. Present address: Department of Chemistry, Lundbeck Foundation Center for Theoretical Chemistry, University of Aarhus, DK-8000 Århus C, Denmark.
[b)]Also at Faculty of Chemistry, University of Warsaw, Pasteura 1, 02-093 Warsaw, Poland.
[c)]Present address: Zaporoska 8/4, 30-389 Krakow, Poland.

ployed. Next, in Sec. III, we present sample calculations and timings for molecules containing up to 642 atoms. Section IV contains some concluding remarks.

II. THEORY AND IMPLEMENTATION

The Kohn–Sham energy of a closed-shell molecule with electron density ρ is given by

$$E_{\mathrm{KS}} = T_{\mathrm{s}}[\rho] + h_{\mathrm{ne}}[\rho] + J[\rho] + E_{\mathrm{XC}}[\rho] + h_{\mathrm{nuc}}, \tag{1}$$

where $T_{\mathrm{s}}[\rho]$ is the non-interacting kinetic energy, $h_{\mathrm{ne}}[\rho]$ is the nuclear-attraction energy, $J[\rho]$ is the Coulomb self-repulsion energy, $E_{\mathrm{XC}}[\rho]$ is the exchange-correlation (XC) energy, and h_{nuc} is the nuclear repulsion energy. Let now R_e be one of the $3N$ nuclear coordinates where N is the number of atoms. Denoting partial differentiation with respect to the nuclear coordinate R_e by superscript e, we obtain the following expression for the molecular gradient:[1,15]

$$G_e = \frac{\mathrm{d}E_{\mathrm{KS}}}{\mathrm{d}R_e} = T_{\mathrm{s}}^e + h_{\mathrm{ne}}^e + J^e + E_{\mathrm{XC}}^e + h_{\mathrm{nuc}}^e - \mathrm{Tr}\mathbf{D}\mathbf{S}^e\mathbf{D}\mathbf{F}, \tag{2}$$

where **S**, **D**, and **F** are the overlap and converged density and Kohn–Sham matrices, respectively, all three in the atomic-orbital (AO) basis. Comparing Eqs. (1) and (2), we note that the molecular gradient contains one contribution from each term in the electronic energy, representing the partial derivative of that term with respect to the nuclear coordinate R_e. The additional term $-\mathrm{Tr}\mathbf{D}\mathbf{S}^e\mathbf{D}\mathbf{F}$ in the gradient arises from the orthonormality constraints on the molecular orbitals since the AO overlap matrix **S** changes with changing molecular geometry.

In the molecular gradient in Eq. (2), the terms T_{s}^e, h_{nuc}^e, and $-\mathrm{Tr}\mathbf{D}\mathbf{S}^e\mathbf{D}\mathbf{F}$ are inexpensive and easily calculated in linear time. By contrast, given the long-range nature of Coulomb interactions, the calculation of the nuclear-attraction and electron-repulsion terms h_{ne}^e and J^e is more difficult and will receive special attention in this paper. The evaluation of the exchange-correlation contribution E_{XC}^e is likewise time-consuming and will also be discussed in more detail.

A. The density-fitted Coulomb contributions to the energy and gradient

Consider the evaluation of the classical Coulomb self-repulsion energy

$$J[\rho] = \frac{1}{2}\int\int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}}\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 = \frac{1}{2}(\rho|\rho) \tag{3}$$

for a given electron density ρ expanded in AO products,

$$\rho(\mathbf{r}) = 2\sum_{\mu\nu}\chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r})D_{\mu\nu}, \tag{4}$$

where the $D_{\mu\nu}$ are the elements of the one-electron AO density matrix **D**. In the density-fitting approximation,[8,16–18] the electron density ρ is approximated by a fitted density $\tilde{\rho}$, expanded in a set of atom-centered auxiliary basis functions $\chi_\alpha(\mathbf{r})$ according to

$$\tilde{\rho}(\mathbf{r}) = \sum_\alpha c_\alpha\chi_\alpha(\mathbf{r}). \tag{5}$$

Following Dunlap,[19] the Coulomb contribution to the Kohn–Sham energy in Eq. (1) is approximated by

$$\tilde{J} = \tfrac{1}{2}(\rho|\tilde{\rho}) + \tfrac{1}{2}(\tilde{\rho}|\rho) - \tfrac{1}{2}(\tilde{\rho}|\tilde{\rho}) = J - \tfrac{1}{2}(\tilde{\rho}-\rho|\tilde{\rho}-\rho), \tag{6}$$

whose error is quadratic in $\tilde{\rho}-\rho$, yielding an approximate Kohn–Sham energy $\tilde{E}_{\mathrm{KS}}$ that is subsequently optimized variationally with respect to both the AO density-matrix elements $D_{\mu\nu}$ and the density-fitting coefficients c_α. The optimization with respect to $D_{\mu\nu}$ leads to the Coulomb contribution to the Kohn–Sham matrix,

$$\tilde{J}_{\mu\nu} = (\mu\nu|\tilde{\rho}), \tag{7}$$

while the optimization with respect to c_α gives a linear set of equations for the density-fitting coefficients,

$$\sum_\beta (\alpha|\beta)c_\beta = \sum_{\mu\nu}(\alpha|\mu\nu)D_{\mu\nu}, \tag{8}$$

whose solution scales cubically with respect to system size. Because of the low prefactor of standard library routines, cubical scaling becomes a problem only for very large systems (typically containing more than 10 000 auxiliary basis functions). For such systems, the fitting coefficients can be obtained in a linear-scaling fashion by using the techniques described in Refs. 20–24. However, in all calculations presented here (for molecules containing up to 642 atoms), we have used the LAPACK routines DPOTRF (for Cholesky decomposition) and DPOTRS (for forward and backward substitutions) to calculate the density-fittingcoefficients during the optimization of the Kohn–Sham energy.

Taking the partial derivative of the fitted electronic Coulomb repulsion energy of Eq. (6) with respect to the nuclear coordinate R_e, we obtain[25]

$$\begin{aligned}\tilde{J}^e &= (\rho^e|\tilde{\rho}) + (\tilde{\rho}^e|\rho) - (\tilde{\rho}^e|\tilde{\rho}) \\ &= J^e - (\tilde{\rho}^e - \rho^e|\tilde{\rho}-\rho) \\ &= \sum_{\mu\nu} D_{\mu\nu}\tilde{J}^e_{\mu\nu} + \sum_\alpha c_\alpha(J^e_\alpha - \tilde{J}^e_\alpha),\end{aligned} \tag{9}$$

where we have introduced the Coulomb potentials

$$\tilde{J}^e_{\mu\nu} = ((\mu\nu)^e|\tilde{\rho}), \quad J^e_\alpha = (\alpha^e|\rho), \quad \tilde{J}^e_\alpha = (\alpha^e|\tilde{\rho}). \tag{10}$$

Being partial derivatives with respect to the nuclear coordinates, this expression does not include contributions from variations in the density-matrix elements $D_{\mu\nu}$ of Eq. (4) and in the density-fitting coefficients c_α of Eq. (5). The contributions from the variations in the density matrix (induced by the orthonormality constraints) are included in $-\mathrm{Tr}\mathbf{D}\mathbf{S}^e\mathbf{D}\mathbf{F}$ of Eq. (2), whereas the contributions from the density-fitting coefficients vanish as seen by differentiation of Eq. (6) followed by the use of Eq. (8). We note from Eq. (9) that the error in the differentiated density-fitted Coulomb energy $\tilde{J}^e - J^e$ is bilinear in the errors in the undifferentiated fitted density $\tilde{\rho}-\rho$ and in the differentiated fitted density $\tilde{\rho}^e-\rho^e$.

Just like the undifferentiated Coulomb potential $\tilde{J}_{\mu\nu}$, the differentiated potentials in Eq. (10) are obtained in linear time by combining Cauchy–Schwarz screening[2,16]

$$|(f|g)| \leq \sqrt{(f|f)}\sqrt{(g|g)}, \tag{11}$$

(which for first-derivative density-fitted integrals includes second-derivative integrals) and by separating the Coulomb contributions into near-field (NF) and far-field (FF) contributions. The NF contributions are calculated using the solid-harmonic Hermite scheme in Ref. 5 as described in Sec. II B, whereas the FF contributions are calculated using the CFMM in Ref. 3 as discussed in Sec. II C.

B. The NF Coulomb contribution to the molecular gradient

The calculation of the three gradient contributions of Eq. (10) may be simplified and accelerated by noting that a solid-harmonic primitive Gaussian $S^{lm}_{a,\mathbf{A}}(\mathbf{r})$ of exponent a at $\mathbf{A}=(A_x,A_y,A_z)$ may be expanded with the same coefficients $S^{lm}_{\mathbf{i}}$ in Cartesian $G^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r})$ and Hermite $H^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r})$ Gaussian orbitals:[5]

$$S^{lm}_{a,\mathbf{A}}(\mathbf{r}) = \sum_{i_x+i_y+i_z=l} S^{lm}_{\mathbf{i}} H^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r}) = \sum_{i_x+i_y+i_z=l} S^{lm}_{\mathbf{i}} G^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r}), \tag{12}$$

$$H^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r}) = \frac{\partial^{i_x+i_y+i_z}\exp(-ar_A^2)}{(2a)^{i_x+i_y+i_z}\,\partial A_x^{i_x}\,\partial A_y^{i_y}\,\partial A_z^{i_z}}, \tag{13}$$

$$G^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r}) = x_A^{i_x} y_A^{i_y} z_A^{i_z} \exp(-ar_A^2), \tag{14}$$

where $\mathbf{i}=(i_x,i_y,i_z)$, $\mathbf{r}_A=\mathbf{r}-\mathbf{A}=(x_A,y_A,z_A)$, and $r_A=|\mathbf{r}_A|$. Traditionally, Cartesian Gaussians have been used for integral evaluations. However, as pointed out in Ref. 5, for derivative integrals, the Hermite expansion is preferable since differentiation with respect to nuclear coordinates merely increments one of the quantum numbers of $H^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r})$, whereas the differentiation of $G^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r})$ gives a linear combination of Cartesian Gaussians. In the gradient implementation reported here, we therefore employ Hermite Gaussians. Note here that the identity of Eq. (12) follows as a special case of the general theory for spherical tensor gradient operators, as reviewed by Weniger.[26]

Let us now consider the expansion of exact and fitted densities Eqs. (4) and (5) in Hermite Gaussians. Following the approach taken by McMurchie and Davidson[27] for products of primitive Cartesian Gaussians, the product of two Hermite Gaussians $H^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r})$ and $H^{\mathbf{j}}_{b,\mathbf{B}}(\mathbf{r})$ is expanded as

$$H^{\mathbf{i}}_{a,\mathbf{A}}(\mathbf{r})H^{\mathbf{j}}_{b,\mathbf{B}}(\mathbf{r}) = \sum_{|\mathbf{t}|=0}^{\mathbf{i}+\mathbf{j}} E^{\mathbf{ij}}_{\mathbf{t}}(a,b,\mathbf{R}_{AB})\Lambda_{\mathbf{t}}(p,\mathbf{r}_P),$$
$$\Lambda_{\mathbf{t}}(p,\mathbf{r}_P) = (2p)^t H^{\mathbf{t}}_{p,\mathbf{P}}(\mathbf{r}), \tag{15}$$

with $\mathbf{t}=(t_x,t_y,t_z)$, $t=|\mathbf{t}|$, $\mathbf{R}_{AB}=\mathbf{A}-\mathbf{B}$, $p=a+b$, and $\mathbf{P}=(a\mathbf{A}+b\mathbf{B})/p$. The recurrence relations for the expansion coefficients $E^{\mathbf{ij}}_{\mathbf{t}}(a,b,\mathbf{R}_{AB})$ in Eq. (15) are given in Ref. 5 and differ slightly from those for products of Cartesian Gaussians in Ref. 27. In the J-engine-based formalisms[14,28–30] significant speed-ups are obtained by contracting the density-matrix elements at an early stage of the integration, avoiding any explicit evaluation of the solid-harmonic electron-repulsion integrals. In the McMurchie–Davidson based formalism employed here[14] this is achieved by, prior to any contraction step, combining the $E^{\mathbf{ij}}_{\mathbf{t}}$ coefficients sharing identical Hermite Gaussians $\Lambda_{\mathbf{t}}(p,\mathbf{r}_P)$ with density-matrix elements, primitive contraction coefficients and solid-harmonic transformation coefficients—to form combined coefficients $F^{p,\mathbf{P}}_{\mathbf{t}}$. Denoting primitive Gaussians by the upright indices m and n, the electron density $\rho(\mathbf{r})$ of Eq. (4) may thus be expanded as

$$\rho(\mathbf{r}) = \sum_{p,\mathbf{P}} \sum_{|\mathbf{t}|=0}^{l_p} (-1)^{|\mathbf{t}|} F^{p,\mathbf{P}}_{\mathbf{t}} \Lambda_{\mathbf{t}}(p,\mathbf{r}_P), \tag{16}$$

$$F^{p,\mathbf{P}}_{\mathbf{t}} = (-1)^{|\mathbf{t}|} 2 \sum_{\mathrm{mn}\in p,\mathbf{P}} C_{\mathrm{m}} C_{\mathrm{n}} \sum_{\mathbf{ij}} S^{l_{\mathrm{m}}m_{\mathrm{m}}}_{\mathbf{i}} S^{l_{\mathrm{n}}m_{\mathrm{n}}}_{\mathbf{j}} E^{\mathbf{ij}}_{\mathbf{t},\mathrm{mn}} D_{\mathrm{mn}}, \tag{17}$$

where C_{m} and C_{n} are primitive contraction coefficients, D_{mn} is the corresponding density-matrix element, the notation $\mathrm{mn}\in p,\mathbf{P}$ indicates that the summation is over all primitive products $S^{l_{\mathrm{m}}m_{\mathrm{m}}}_{a_{\mathrm{m}},\mathbf{A}_{\mathrm{m}}}(\mathbf{r}) S^{l_{\mathrm{n}}m_{\mathrm{n}}}_{a_{\mathrm{n}},\mathbf{A}_{\mathrm{n}}}(\mathbf{r})$ of total exponent $p=a_{\mathrm{m}}+a_{\mathrm{n}}$ and center $\mathbf{P}=(a_{\mathrm{m}}\mathbf{A}_{\mathrm{m}}+a_{\mathrm{n}}\mathbf{A}_{\mathrm{n}})/(a_{\mathrm{m}}+a_{\mathrm{n}})$, and with maximal combined angular momentum $l_p=\max\{l_m+l_n\}$. The fitted density $\tilde{\rho}(\mathbf{r})$ of Eq. (5) may be expanded in the same manner,

$$\tilde{\rho}(\mathbf{r}) = \sum_{p,\mathbf{P}} \sum_{|\mathbf{t}|=l_p} (-1)^{|\mathbf{t}|} \tilde{F}^{p,\mathbf{P}}_{\mathbf{t}} \Lambda_{\mathbf{t}}(p,\mathbf{r}_P), \tag{18}$$

$$\tilde{F}^{p,\mathbf{P}}_{\mathbf{t}} = \sum_{\mathrm{m}\in p,\mathbf{P}} C_{\mathrm{m}} S^{l_{\mathrm{m}}m_{\mathrm{m}}}_{\mathbf{t}} (-2p)^{-l_{\mathrm{m}}} c_{\mathrm{m}}, \tag{19}$$

where the m summation is over all primitive auxiliary functions of exponent p and center $\mathbf{P}$.

Consider now the three differentiated contributions of Eq. (10). Denoting the nucleus of coordinate R_e by E and letting $\mathbf{e}$ be the vector (1,0,0), (0,1,0), or (0,0,1) depending on whether the differentiation is in the x, y, or z direction, respectively, we obtain

$$\tilde{J}^e_{\mu\nu} = \sum_{\mathbf{ij}} S^{l_\mu m_\mu}_{\mathbf{i}} S^{l_\nu m_\nu}_{\mathbf{j}} \sum_{\mathrm{mn}} C_{\mathrm{m}} C_{\mathrm{n}} \sum_{|\mathbf{t}|=0}^{l_\mu+l_\nu+1} (\delta_{EA} E^{\mathbf{i}+\mathbf{e},\mathbf{j}}_{\mathbf{t}} + \delta_{EB} E^{\mathbf{i},\mathbf{j}+\mathbf{e}}_{\mathbf{t}}) \times \sum_{q,\mathbf{Q}} \sum_{|\mathbf{u}|=l_q} \tilde{F}^{q,\mathbf{Q}}_{\mathbf{u}} R_{\mathbf{t}+\mathbf{u}}(\gamma,\mathbf{R}_{PQ}),$$

$$J^e_\alpha = \delta_{EP} \sum_{|\mathbf{t}|=l_\alpha} S^{l_\alpha m_\alpha}_{\mathbf{t}} \sum_{\mathrm{m}} C_{\mathrm{m}} (2p)^{1-l_\alpha} \sum_{q,\mathbf{Q}} \sum_{|\mathbf{u}|=0}^{l_q} F^{q,\mathbf{Q}}_{\mathbf{u}} R_{\mathbf{t}+\mathbf{e}+\mathbf{u}}(\gamma,\mathbf{R}_{PQ})$$

$$\tilde{J}^e_\alpha = \delta_{EP} \sum_{|\mathbf{t}|=l_\alpha} S^{l_\alpha m_\alpha}_{\mathbf{t}} \sum_{\mathrm{m}} C_{\mathrm{m}} (2p)^{1-l_\alpha} \sum_{q,\mathbf{Q}} \sum_{|\mathbf{u}|=l_q} \tilde{F}^{q,\mathbf{Q}}_{\mathbf{u}} R_{\mathbf{t}+\mathbf{e}+\mathbf{u}}(\gamma,\mathbf{R}_{PQ}), \tag{20}$$

with $\gamma=pq/(p+q)$ and $R_{PQ}=|\mathbf{P}-\mathbf{Q}|$. The Hermite repulsion integrals $R_{tuv}(\gamma,\mathbf{R}_{PQ})$ are found by recurrence relations.[27] As seen from Eq. (20), the expansion of solid harmonics in Hermite rather than Cartesian Gaussians has two advantages. First, for one-center auxiliary functions, the number of contractions is reduced, as seen, for instance, from the first term of $\tilde{J}^e_{\mu\nu}$ where the innermost summation contains terms for which $|\mathbf{u}|=l_q$ only, rather than all terms with $|\mathbf{u}|\leq l_q$. Second, no differentiated $E^{\mathbf{ij}}_{\mathbf{t}}$ coefficients arise; instead, these coeffi-

cients are incremented by one in the quantum numbers, thereby simplifying and accelerating the evaluation.

C. The FF Coulomb contribution to the molecular gradient

There are two possible ways to obtain the FF Coulomb gradient contribution, either by differentiating the CFMM multipole expansions of the undifferentiated classical interaction energy[10,31]

$$J^{\mathrm{cls}} = \frac{1}{2}\sum_{p}\sum_{q\in \mathrm{FF_P}} D_p \mathbf{q}_p(\mathbf{P})^{\mathrm{T}}\mathbf{W}(\mathbf{R}_{\bar{\mathbf{P}}\mathbf{P}})^{\mathrm{T}} \times \mathbf{T}(\mathbf{R}_{\bar{\mathbf{Q}}\bar{\mathbf{P}}})\mathbf{W}(\mathbf{R}_{\bar{\mathbf{Q}}\mathbf{Q}})\mathbf{q}_q(\mathbf{Q})D_q, \tag{21}$$

or by applying the CFMM approximation to the differentiated Coulomb repulsion energy.[9] The former approach may appear advantageous, providing the exact gradient for a given approximate CFMM treatment of the Coulomb repulsion energy. However, this statement holds only if we include in our calculations the dependence of the box centers $\bar{\mathbf{P}}$ on the nuclear coordinates. We also note that the CFMM energy is not always smooth, changing in a nondifferentiable manner as the centers $\mathbf{P}$ of the charge distributions $\Omega_p(\mathbf{r})$ move between the boxes when the molecule distorts, making its direct differentiation problematic.

For the above reasons, we have chosen the second approach to the calculation of the FF Coulomb gradient, applying the CFMM individually to the three contributions to the differentiated Coulomb energy in Eq. (9). This approach is simple to implement as it only includes the multipole moments $\mathbf{q}_p^e$ of the differentiated charge distributions $\Omega_p^e(\mathbf{r})$, keeping the FF potential fixed and thus leaving the translation matrices $\mathbf{W}(\mathbf{R})$ and the interaction matrices $\mathbf{T}(\mathbf{R})$ unchanged.

D. One-electron contributions to the molecular gradient

Linear scaling of the one-electron nuclear-attraction force h_{ne}^e is achieved in the same fashion as for the density-fitted two-electron repulsion Coulomb force in Sec. II A. In fact, the one-electron contributions can be obtained at essentially no additional cost by incorporating these into the two-electron contributions $\tilde{J}^e$ of Eq. (9). To see this, we rewrite the one-electron nuclear-attraction force as a two-particle term in the manner

$$h_{\mathrm{ne}}^e = -(\rho^e|\rho_{\mathrm{nuc}}) - (\rho_{\mathrm{nuc}}^e|\rho), \quad \rho_{\mathrm{nuc}}(\mathbf{r}) = \sum_A Z_A\delta(\mathbf{r}-\mathbf{A}), \tag{22}$$

where $\rho_{\mathrm{nuc}}(\mathbf{r})$ is the nuclear charge density and Z_A is the charge of the nucleus at position $\mathbf{A}$. The combined one- and two-electron Coulomb contributions to the molecular gradient may then be written as

$$\tilde{J}^e + h_{\mathrm{ne}}^e = (\rho^e|\tilde{\rho}-\rho_{\mathrm{nuc}}) + (\tilde{\rho}^e - \rho_{\mathrm{nuc}}^e|\rho) - (\tilde{\rho}^e|\tilde{\rho}). \tag{23}$$

This approach is particularly advantageous for the FF part of the molecular gradient. For the first term in Eq. (23), the Coulomb potential generated by all particles $\tilde{\rho}(\mathbf{r})-\rho_{\mathrm{nuc}}(\mathbf{r})$ is obtained at no additional cost compared with the potential from the electrons $\tilde{\rho}(\mathbf{r})$. Likewise, for the second term in Eq. (23), the potential from $\rho(\mathbf{r})$ can be interacted with $\tilde{\rho}^e(\mathbf{r})-\rho_{\mathrm{nuc}}^e(\mathbf{r})$ and $\tilde{\rho}^e(\mathbf{r})$ at the same cost. These simplifications occur since the auxiliary basis functions [in terms of which $\tilde{\rho}(\mathbf{r})$ is expanded] and the atomic charges are located at the same positions—namely, on the atomic centers. In this paper, we therefore calculate the Coulomb contributions to the molecular gradient from Eq. (23).

Linear scaling of the two remaining one-electron contributions to the molecular gradient in Eq. (2) (i.e., the kinetic-energy contribution T_{s}^e and the reorthonormalization contribution $-\mathrm{Tr}\mathbf{D}\mathbf{S}^e\mathbf{D}\mathbf{F}$) is easily obtained by neglecting terms for which the distance R_{AB} between the nuclear centers of two basis functions $\chi_a(\mathbf{r})$ and $\chi_b(\mathbf{r})$ is greater than the sum of their respective extents, r_a and r_b. The extent of a primitive Gaussian basis function $\chi_{a_m}(\mathbf{r})$ is determined iteratively from the nonlinear equation

$$r_{a_m} = \sqrt{\frac{l(\ln r_{a_m} + \ln N - \ln \epsilon)}{a_m}}, \tag{24}$$

where a_m is the primitive Gaussian exponent, l is the angular momentum, N is the normalized contraction coefficient, and ϵ is a threshold. The extent r_a of a contracted basis function $\chi_a(\mathbf{r})$ is taken to be the maxima of the extents of the primitive Gaussians forming the contracted Gaussian basis function. A threshold $\epsilon = 10^{-15}$ has been used in this work.

E. The exchange-correlation contribution to the molecular gradient

The exchange-correlation energy is obtained by integration over all space

$$E_{\mathrm{XC}}[\rho] = \int F_{\mathrm{XC}}(\mathbf{r})\mathrm{d}\mathbf{r}. \tag{25}$$

Given the complexity of $F_{\mathrm{XC}}(\mathbf{r})$, this integration is performed by numerical quadrature, employing Becke's atomic partitioning scheme[32]

$$E_{\mathrm{XC}}[\rho] = \sum_A \sum_i w_i w_A(\mathbf{r}_i) F_{\mathrm{XC}}(\mathbf{r}_i), \tag{26}$$

where the first summation is over all atoms A, the second summation is over the grid points i of atom A, w_i is the single-center spherical weight, and $w_A(\mathbf{r})$ is the atomic weight function. The Becke space partitioning with size corrections[32] is used to determine the atomic weights. The single-center integrations in Eq. (26) are separated into radial and angular integrations. For the angular integration, we use Lebedev grids.[33–37] For the radial integration, we employ a Gauss–Chebychev quadrature of second kind[38] and follow Treutler and Ahlrichs[39] in mapping the integration range from $[0,\infty)$ onto $[-1,1]$ by the transformation $r_i = (\xi/\ln 2)(1+x_i)^{0.6}\ln(2/(1-x_i))$ using atomic scaling parameters ξ. A detailed description of our exchange-correlation energy evaluation is found in Ref. 14.

Differentiation of the exchange-correlation energy Eq. (26) gives the following gradient expression:

$$E_{\mathrm{XC}}^{e} = \sum_{A} \sum_{i} w_i (w_A F_{\mathrm{XC}}^{e} + w_A^{e} F_{\mathrm{XC}}). \quad (27)$$

The first term in Eq. (27) requires the derivative of basis functions with respect to atomic displacements, whereas the second term involves the derivatives of the atomic grid-weight functions $w_A(\mathbf{r})$ as discussed by Johnson *et al.*[40] However, for the calculations presented in this manuscript, the grid-weight contributions are negligible and have not been calculated.

F. Geometry optimization

Traditional quantum-chemistry molecular-geometry optimizers[41–43] employ operations whose cost scales cubically with the number of atoms. However, even though the number of atoms is far smaller than the number of atomic basis functions, for sufficiently large molecules, it becomes necessary also to consider the computational complexity of the geometry optimizer. In recent years, significant progress has been made in reducing this complexity to quadratic[44–46] and linear[47–51] in the number of atoms. For an overview of optimization methods and computational challenges for large systems, see Ref. 52.

For the molecules considered by us here (containing less than a thousand atoms), a sufficiently efficient geometry optimizer has been obtained by making a few simple adjustments to the regular optimizer of DALTON, described in Ref. 43. To summarize, DALTON geometry optimizations are carried out in redundant internal coordinates[41,43] using the quasi-Newton method with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) updating formula[53] and the initial Hessian proposed by Lindh *et al.*[54] The steps are determined by the level-shifted trust-region method.[55] At each iteration, the root-mean-square of the gradient, its maximum absolute element, the root-mean-square of the step vector, and its maximum absolute element in internal coordinates are determined; for a chosen threshold parameter ϵ (typically 10^{-4}), convergence is declared if these four quantities are smaller than ϵ, 5ϵ, 3ϵ, and 15ϵ, respectively.

In adapting the DALTON geometry optimizer to large molecules, two performance bottlenecks were identified: calculation of the generalized inverse of the Wilson B-matrix and diagonalization of the updated Hessian in internal coordinates. In both cases, a previous explicit code was replaced by appropriate calls to LAPACK routines. For the largest geometry-optimized molecule considered here (containing 392 atoms), these simple modifications reduced the CPU time spent in the geometry optimizer to an acceptable one-fifth of the time spent on evaluating the molecular gradient, indicating that the present optimizer is sufficiently efficient for molecules containing less than 1000 atoms.

During the geometry optimization of a molecule, the changes in the density matrix are typically small from iteration to iteration—in particular, close to the stationary point. In the current implementation, we exploit this by taking as our starting guess the McWeeny-purified converged density-matrix at the previous geometry, thereby typically reducing the number of self-consistent field (SCF) iterations by a factor two compared with the use of the extended Hückel guess.

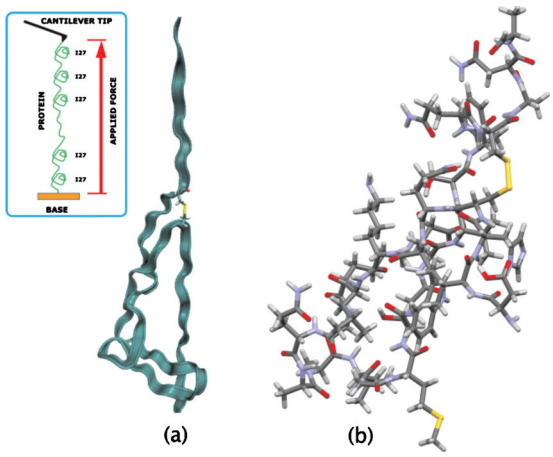

FIG. 1. (a) The titin-I27 domain highlighting the disulfide bridging bond and a schematic representation of the stretching of the polyprotein strain by the aid of the atomic force microscopy; (b) the titin-$I27_{SS}$ model, designed to model the redox-active site in the titin-I27 domain, created from one snapshot (Ref. 58) of titin-I27 during a molecular-dynamics simulation of the force induced unfolding.

However, when the geometry steps are large, McWeeny purification may change the number of electrons in the system; in such cases, we revert to the regular (extended Hückel) starting guess.

III. RESULTS AND DISCUSSION

To illustrate the performance of the new gradient code, implemented in a development version of DALTON,[56] we here report timings for the evaluation of the electronic energy, molecular gradient, and geometry step for some selected molecular systems. All calculations were carried out on a single Intel Xeon 2.66 GHz processor, using an executable compiled with the ifort Intel compiler version 10.1 and linked with the MKL library version 10.1. In the calculations, a threshold of 10^{-10} was used for integral screening. In the CFMM part, the charge distributions of contracted orbital products and density-fitting functions were expanded to order five, whereas the remaining expansions (of box charges and potentials) were carried out to order 12. The "grid 3" of Treutler and Ahlrichs[39] was used for the exchange-correlation quadrature. No point-group symmetry was used in the calculations.

In Sec. III A, we demonstrate linear scaling of the new integral code by calculating the electronic energy and molecular gradient of linear polyene chains. Next, in Sec. III B, we report timings for a wide variety of organic molecules, containing up to 642 atoms. Finally, in Sec. III C, we present timings for geometry optimizations of the valinomycin, taxol, and titin-$I27_{SS}$ (see Fig. 1) molecules, containing up to 392 atoms.

A. Linear polyene chains

In Fig. 2, the computational timings for the construction of the density-fitted Coulomb and exchange-correlation contributions to the (undifferentiated) Kohn–Sham matrix of linear polyenes are plotted. To understand the plots, we recall that the Coulomb evaluation consists of three steps. First, the

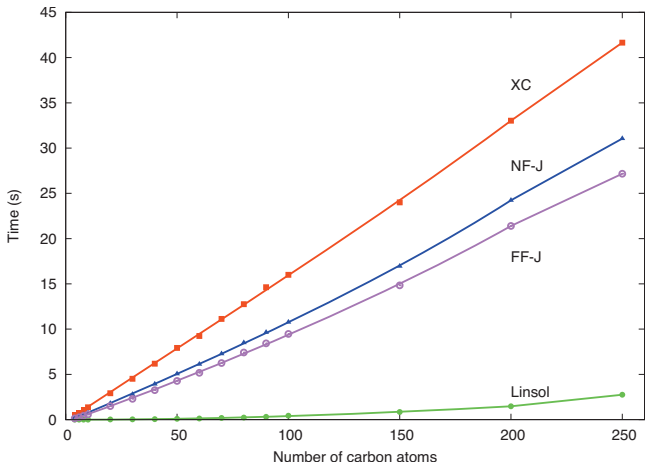

FIG. 2. Timings at the BP86/6-31G** level of theory for a single construction of the XC and the density-fitted-Coulomb contribution to the Kohn–Sham matrix as a function of the number of carbon atoms n for linear polyene chains $CH_2(CH)_{n-2}CH_2$. The Coulomb timings are separated into timings for the near-field contribution (NF-J), the far-field contributions (FF-J), and the linear equation solver (Linsol). The auxiliary basis set in Ref. 59 was used as the density-fitting basis.

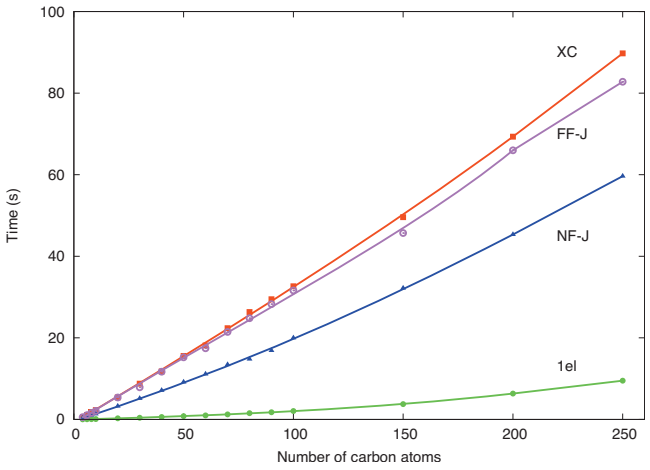

FIG. 3. Timings at the BP86/6-31G** level of theory for the calculation of the molecular gradient as a function of the number of carbon atoms n for linear polyene chains $CH_2(CH)_{n-2}CH_2$. Timings are for the non-Coulomb one-electron contribution (1el), the XC contribution, and the (one- and two-electron) near-field (NF-J) and far-field (FF-J) Coulomb contributions to the gradient. The auxiliary basis set in Ref. 59 was used as the density-fitting basis.

right-hand side of the density-fitting equations Eq. (8) is constructed; second, the fitted density is obtained by solving these equations with the LAPACK routine DPOTRS; third, the Coulomb contribution to the Kohn–Sham matrix Eq. (7) is calculated from the fitted density. In Fig. 2, we have plotted separate timings for the second step (the solution of the linear equations) and the combined NF and FF timings for the first and third steps (rather than separate timings for these two steps). The total Kohn–Sham Coulomb timing is the sum of these three timings.

As seen from Fig. 2, the costs of the NF and FF Coulomb contributions and the exchange-correlation contribution to the Kohn–Sham matrix scale linearly with system size, whereas the cost of the linear equations scales cubically, although with a low prefactor. Although it is possible to solve these equations in a linear-scaling manner, this is not necessary for the systems considered by us and has not been attempted.

It should be noted that Fig. 2 contains timings only for those operations that are repeated at each Kohn–Sham iteration—it does not contain timings for those operations that are carried out in the initialization step, here listed in order of decreasing cost: the generation of the exchange-correlation grid, the calculation and subsequent Cholesky decomposition of the two-electron metric matrix (through the LAPACK routine DPOTRF), the calculation of various one-electron matrices, and the preparation of a starting density-matrix. For large systems, this initialization typically consumes 15% of the total time spent on the energy optimization, as illustrated in Sec. III C.

Turning our attention to the molecular gradient, we have in Fig. 3 plotted the timings for the various contributions to the gradient of linear polyenes: the non-Coulomb one-electron (kinetic-energy and reorthonormalization) contributions, the (one- and two-electron) NF and FF Coulomb contributions, and the exchange-correlation contribution. Unlike for the construction of the Kohn–Sham matrix, no linear equations are solved and no such timings are therefore plotted.

All four contributions in Fig. 3 display near linear-scaling with system size. Comparing with Fig. 2, we note that the exchange-correlation and NF Coulomb contributions are about twice as expensive as the corresponding contributions to the Kohn–Sham matrix, while the FF Coulomb contribution is about three times more expensive. The comparatively slow FF gradient evaluation arises from a time-consuming evaluation of derivative multipole-moment integrals; for the energy, these integrals are evaluated during the initialization and not at each Kohn–Sham iteration. In general, our CFMM implementation for molecular gradients is not fully optimal and further optimization of this step will reduce gradient timings significantly.

B. Sample molecular gradient evaluations

In Table I, we report timings for the calculation of the Kohn–Sham matrix and the molecular gradient, for a total of 23 organic molecules at the BP86/6-31G level of theory on a single Xeon 2.66 GHz processor; for some molecules, the 6-31G** and def2-SV(P) basis sets were also used. For the largest 642-atom crambin molecule in this table, the construction of the Kohn–Sham matrix and the calculation of the molecular gradient take 17 and 26 min, respectively, in the 6-31G basis (with 8598 primitive and 3597 contracted AOs); in the larger def2-SV(P) basis (with 9180 primitive and 5232 contracted AOs), the corresponding timings are 48 and 69 min, respectively. For this molecule, therefore, the gradient evaluation takes about 50% more time than the evaluation of the Kohn–Sham matrix. For the smaller molecules in Table I, the molecular gradient is more expensive relative to the Kohn–Sham matrix—typically by a factor of 2 but never by more than a factor of 2.3.

Comparing the different gradient contributions in Table

TABLE I. Timings in seconds for a single construction of the Kohn–Sham matrix and for the calculation of the molecular gradient for a range of molecules, ordered by increasing number of atoms (Atoms), at the BP86 level of theory. The timings are split into contributions from the XC and the density-fitted NF and FF Coulomb contributions. For the force evaluation, we also give the one-electron kinetic-energy and reorthonormalization contribution (1el). Also listed are the number of primitive (Prim) and contracted (Cont) basis functions. The auxiliary basis set in Ref. 59 was used in all cases, except otherwise stated.

					Kohn–Sham matrix				Molecular gradient				
System	Basis set	Atoms	Prim	Cont	XC	NF	FF	Total[a]	1el	XC	NF	FF	Total
Histidine ($C_6H_9N_3O_2$)	6-31G	20	278	117	1.1	0.5	0.4	2.0	0.1	1.6	1.0	1.2	3.8
Ferrocene ($C_{10}H_{10}Fe$)	6-31G	21	350	137	1.6	0.9	0.5	3.0	0.1	2.5	1.8	1.9	6.4
AT-basepair ($C_{10}H_{11}N_7O_2$)	6-31G	30	462	193	2.0	1.0	0.9	3.9	0.1	3.2	2.0	2.7	8.1
Penicilin ($C_{16}H_{18}N_2O_5S$)	6-31G	42	624	256	3.2	1.8	1.6	6.6	0.2	5.6	3.5	4.4	13.9
Cu-complex $C_{16}H_4BCuF_{18}N_6O$[b]	6-31G	47	1030	413	6.2	4.8	4.2	15.2	0.6	12.6	9.3	12.1	34.5
Dibenzo-18-crown-6 ($C_{20}H_{24}O_6$)	6-31G	50	668	282	3.6	2.6	2.5	8.7	0.4	6.7	4.8	6.8	18.6
Tetracycline ($C_{22}H_{24}N_2O_8$)	6-31G	56	800	336	5.8	4.5	3.9	14.2	0.5	11.9	8.8	10.0	31.1
Beclomethasone ($C_{22}H_{29}ClO_5$)	6-31G	57	756	314	5.8	4.9	3.7	14.4	0.4	12.3	9.5	9.1	31.4
C_{60}-fullerene	6-31G	60	1320	540	18.3	15.3	10.3	43.9	1.2	37.6	30.5	28.1	78.0
Cholesterol ($C_{27}H_{46}O$)	6-31G	74	800	344	6.2	5.2	4.8	16.2	0.5	12.5	9.9	11.0	34.0
CO-heme ($C_{38}H_{36}FeN_6O_5$)	6-31G	86	1312	540	12.7	12.0	11.9	36.6	1.1	25.2	22.8	26.4	75.3
C_{90}-fullerene	6-31G	90	1980	810	31.7	30.4	23.5	85.6	2.3	64.3	58.6	55.4	154.9
Na(trideoxycytoside$-2H)^{-1}$ ($C_{27}H_{35}N_9NaO_{16}P_2$)[c]	6-31G	90	1422	577	10.2	8.1	10.1	28.4	1.0	20.1	15.4	21.9	58.6
C_{100}-fullerene	6-31G	100	2200	900	36.2	35.6	28.7	100.5	3.0	77.7	68.8	65.6	184.6
Taxol ($C_{47}H_{51}NO_{14}$)	6-31G	113	1568	660	17.1	17.1	20.9	55.1	1.5	34.2	30.8	39.0	102.8
	6-31G**		2031	1123	30.2	30.9	33.6	94.7	2.6	60.9	53.7	68.9	182.3
$(Guanine)_4NaCl(adenine)_4$ ($C_{48}H_{56}ClN_{40}NaO_4$)[d]	6-31G	150	2340	966	27.1	27.0	34.6	88.7	2.9	55.7	49.6	65.8	174.0
Valinomycin ($C_{54}H_{90}N_6O_{18}$)	6-31G	168	2076	882	25.3	29.7	46.7	101.7	2.7	49.9	50.2	77.0	173.4
	6-31G**		2736	1542	44.9	51.6	72.0	168.5	4.9	89.2	84.3	128.9	296.0
Vancomycin ($C_{66}H_{75}Cl_2N_9O_{24}$)	6-31G	176	2570	1067	25.2	25.2	38.8	89.2	2.9	53.6	45.7	69.6	172.0
C_{180}-fullerene	6-31G	180	3960	1620	63.0	66.9	76.6	206.5	7.5	130.3	124.4	147.9	410.7
C_{240}-fullerene	6-31G	240	5280	2160	84.8	88.3	107.4	280.5	11.7	173.8	165.2	203.0	553.8
Collagenlike peptide ($C_{56}H_{164}N_{21}O_{61}$)[e]	6-31G	302	4472	1904	76.9	113.8	221.9	412.6	11.0	155.2	190.6	322.4	679.3
Titin-I27$_{SS}$ ($C_{124}H_{192}N_{36}O_{37}S_3$)[f]	6-31G[g]	392	5265	2221	63.4	82.4	174.5	320.3	11.0	128.0	139.5	251.5	505.9
	def2-SV(P)[h]		5604	3196	109.8	307.6	462.9	880.3	18.0	219.0	475.5	616.0	1328.0
Crambin ($C_{202}H_{315}N_{55}O_{64}S_6$)[e]	6-31G	642	8598	3597	171.6	255.2	581.4	1008.2	35.0	327.5	422.5	803.5	1588.5
	def2-SV(P)[h]		9180	5232	283.6	960.7	1629.6	2873.9	75.0	535.5	1452.5	2068.0	4131.0

[a]SCF-initialization time (1el integrals, DFT-grid construction, etc.) not included.
[b]Reference 60.
[c]Reference 61.
[d]Reference 62.
[e]Reference 63.
[f]See caption of Fig. 1.
[g]6-31G* for three sulfur and two carbon atoms, 6-31G for all other atoms.
[h]Auxiliary basis set in Ref. 64.

I, we note that the Coulomb contribution is more expensive than the exchange-correlation contribution, typically by 30% and never by more than 41%. Except for four molecules, the FF Coulomb contribution is more expensive than the NF contribution (typically by 30%), especially for the large molecules in small basis sets—for crambin, for example, the FF contribution is 90% more expensive than the NF contribution in the 6-31G basis, but only 42% more expensive in the def2-SV(P) basis.

For all molecules in Table I, the CFMM is used for the FF contribution to the molecular gradient. However, for molecules containing less than about 60 atoms, it is best to treat all Coulomb interactions as NF interactions, avoiding the CFMM altogether. For larger molecules, the CFMM significantly improves overall efficiency—for example, by a factor two for the gradient of the 392-atom titin-I27$_{SS}$ molecule.

At the BP86/6-31G level of theory, the time spent on the molecular gradient per atom increases from 0.2 s for the 20-atom histidine molecule to 2.5 s for the 642-atom crambin molecule. Comparing molecules of different sizes, we note that the time spent on the gradient stabilizes at about 2 s per atom for molecules with more than 200 atoms, indicating that the linear-scaling regime has been reached. However, the exact time per atom varies considerably, depending on the nature of the system—for example, as much as 2.3 s is spent on each atom of the compact C_{240} molecule (which contains no hydrogen atoms), compared with only 1.3 s for the larger but less compact titin-I27$_{SS}$ molecule.

To set the presented gradient timings in a better perspective with respect to its performance, we compare with the TURBOMOLE gradient timings for the two largest molecules presented in this section. The presented titin-I27$_{SS}$ and crambin gradient timings are faster than TURBOMOLE both in the 6-31G (6-31G* for the three sulfur and two of the carbon atoms for titin-I27$_{SS}$) and in the def-SV(P) basis sets. In the 6-31G basis, our gradient timings of 506 and 1589 s are faster by little more than a factor two compared to the corresponding TURBOMOLE timings of 1049 and 3250 s for the

TABLE II. Average timings in seconds for the electronic energy, the molecular gradient, and the geometry step for the geometry optimizations of the taxol, valinomycin, and titin-I27$_{SS}$. All calculations were carried out at the BP86/6-31G level of theory with the auxiliary basis set in Ref. 59. Also reported are the average number of SCF iterations in each energy optimization, with the root-mean-square SCF gradient norm converged to 10^{-5}, and the number of geometry steps needed to converge the geometry, with threshold $\epsilon=10^{-4}$ (see text).

	Taxol	Valinomycin	Titin-I27$_{SS}$ [a]
Electronic energy	539	855	2996
Molecular gradient	103	173	506
Geometry step	3	9	106
Average number of SCF steps	10.7	9.1	9.9
Total number of geometry steps	36	27	21

[a]6-31G* for three sulfur and two carbon atoms, 6-31G for all other atoms.

two molecules, respectively. In the larger def-SV(P) basis, our 1328 and 4131 s gradient timings are about 13% faster than the TURBOMOLE timings of 1521 and 4131 s. The main reason our implementation is faster than TURBOMOLE by as much as a factor 2 for the 6-31G calculations is probably because our implementation calculates integrals involving *s*- and *p*-orbitals with shared exponents in one go, allowing significant speed-ups, whereas TURBOMOLE treats these integrals separately. We also note here that TURBOMOLE recalculates the exchange-correlation grid when evaluating the gradient, whereas in our implementation the grid constructed for the evaluation of the energy is reused for the gradient. The TURBOMOLE calculations were carried out with TURBOMOLE version 6.0.2 on the same platform as the DALTON calculations.

C. Geometry optimizations

To assess the overall efficiency of the new code for large molecules, we report in Table II timings for geometry optimizations of valinomycin, taxol, and titin-I27$_{SS}$, converging the SCF energy at each geometry step to a root-mean-square gradient norm of 10^{-5} and using the geometry convergence criteria of Sec. II F with $\epsilon=10^{-4}$. For these molecules, 83%–84% of the total CPU time at each geometry was spent on the electronic energy, 14%–17% was spent on the molecular gradient, and 1%–3% on the geometry step.

For these molecules, therefore, the optimization of the energy is the dominant computational step. For titin-I27$_{SS}$, the energy optimization takes about 50 min, of which 15% is spent on initialization, 13% on the NF Coulomb contribution, 45% on the FF Coulomb contributions, 17% on the exchange-correlation contribution, and 9% on the Roothaan–Hall diagonalization and density-matrix averaging; only 0.03% is spent on solving Eq. (8), to obtain the fitted density. While the Coulomb FF contribution is three times more expensive than the NF contribution in a full titin-I27$_{SS}$ energy optimization, it is only twice as expensive for the single construction of the Kohn–Sham matrix reported in Table I. This difference arises since our implementation of the incremental density scheme for the construction of the Kohn–Sham matrix[57] is more efficient for the NF contribution than for the FF contribution.

IV. CONCLUSIONS

We have demonstrated the efficient implementation of the Kohn–Sham molecular gradients for molecules containing up to 642 atoms. The gradients have been used for the geometry optimization of three selected molecules containing up to 392 atoms. We have presented a linear-scaling density-fitted-Coulomb force evaluation. We have further accelerated the near-field contribution to the Coulomb force evaluation using a novel scheme for the molecular integral evaluation, in which the solid-harmonic Gaussians are expanded in the Hermit rather the Cartesian Gaussians—reducing the cost of differentiated integrals and simplifying the implementation. The results in this paper clearly demonstrate the efficiency of the presented implementation. We note finally that to allow calculations on systems containing thousands of atoms a parallel implementation is needed. Such an implementation is under development in our laboratory.

ACKNOWLEDGMENTS

We would like to acknowledge the financial support from the Norwegian Research Council through a Strategic University Program in Quantum Chemistry (Grant No. 154011/420), through the Nanomat "Molecular Modeling in Nanotechnology" program (Grant No. 158538/431), and through the CoE Centre for Theoretical and Computational Chemistry (Grant No. 179568/V30). We would like to further acknowledge the NOTUR computing facilities which have been used to conduct the calculations presented in this paper. F.P. would like to acknowledge the financial support from the Foundation for Polish Science (FNP) via Homing program (Grant No. HOM/2008/10B) within EEA Financial Mechanism, which allowed for short visits to Oslo.

[1] P. Pulay, Mol. Phys. **17**, 197 (1969).
[2] M. Häser and R. Ahlrichs, J. Comput. Chem. **10**, 104 (1989).
[3] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).
[4] D. S. Lambrecht and C. Ochsenfeld, J. Chem. Phys. **123**, 184101 (2005).
[5] S. Reine, E. Tellgren, and T. Helgaker, Phys. Chem. Chem. Phys. **9**, 4771 (2007).
[6] L. Versluis and T. Ziegler, J. Chem. Phys. **88**, 322 (1988).
[7] R. Fournier, J. Andzelm, and D. R. Salahub, J. Chem. Phys. **90**, 6371 (1989).
[8] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, J. Chem. Phys. **71**, 3396 (1979).
[9] J. C. Burant, M. C. Strain, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **248**, 43 (1996).
[10] Y. Shao, C. A. White, and M. Head-Gordon, J. Chem. Phys. **114**, 6572 (2001).
[11] D. Rappoport and F. Furche, J. Chem. Phys. **122**, 064105 (2005).
[12] V. D. Domínguez-Soria, G. Geudtner, J. L. Morales, P. Calaminici, and A. M. Köster, J. Chem. Phys. **131**, 124102 (2009).
[13] P. Calaminici, G. Geudtner, and A. M. Köster, J. Chem. Theory Comput. **5**, 29 (2009).
[14] M. A. Watson, P. Sałek, P. Macak, and T. Helgaker, J. Chem. Phys. **121**, 2915 (2004).
[15] H. Larsen, T. Helgaker, J. Olsen, and P. Jørgensen, J. Chem. Phys. **115**, 10344 (2001).
[16] J. L. Whitten, J. Chem. Phys. **58**, 4496 (1973).
[17] E. J. Baerends, D. E. Ellis, and P. Ros, Chem. Phys. **2**, 41 (1973).
[18] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, J. Chem. Phys. **71**, 4993 (1979).
[19] B. I. Dunlap, J. Mol. Struct. **501–502**, 221 (2000).

[20] S. Reine, E. Tellgren, A. Krapp, T. Kjærgaard, T. Helgaker, B. Jansik, S. Høst, and P. Sałek, J. Chem. Phys. **129**, 104101 (2008).
[21] R. T. Gallant and A. St-Amant, Chem. Phys. Lett. **256**, 569 (1996).
[22] C. Fonseca Guerra, J. G. Snijders, G. te Velde, and E. J. Baerends, Theor. Chem. Acc. **99**, 391 (1998).
[23] A. Sodt, J. E. Subotnik, and M. Head-Gordon, J. Chem. Phys. **125**, 194109 (2006).
[24] P. Sałek, S. Høst, L. Thøgersen, P. Jørgensen, P. Manninen, J. Olsen, B. Jansik, S. Reine, F. Pawłowski, E. Tellgren, T. Helgaker, and S. Coriani, J. Chem. Phys. **126**, 114110 (2007).
[25] B. I. Dunlap, J. Andzelm, and J. W. Mintmire, Phys. Rev. A **42**, 6354 (1990).
[26] E. J. Weniger, Collect. Czech. Chem. Commun. **70**, 1225 (2005).
[27] L. E. McMurchie and E. R. Davidson, J. Comput. Phys. **26**, 218 (1978).
[28] G. Reza Ahmadi and J. Almlöf, Chem. Phys. Lett. **246**, 364 (1995).
[29] C. A. White and M. Head-Gordon, J. Chem. Phys. **104**, 2620 (1996).
[30] Y. Shao and M. Head-Gordon, Chem. Phys. Lett. **323**, 425 (2000).
[31] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic–Structure Theory* (Wiley, Chichester, 2000).
[32] A. D. Becke, J. Chem. Phys. **88**, 2547 (1988).
[33] V. I. Lebedev and D. N. Laikov, Dokl. Math. **59**, 477 (1999).
[34] V. I. Lebedev, Dokl. Math. **50**, 283 (1995).
[35] V. I. Lebedev and A. L. Skorokhodov, Dokl. Math. **45**, 587 (1992).
[36] V. I. Lebedev, Sib. Math. J. **18**, 99 (1977).
[37] V. I. Lebedev, Comput. Math. Math. Phys. **15**, 44 (1975).
[38] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1970).
[39] O. Treutler and R. Ahlrichs, J. Chem. Phys. **102**, 346 (1995).
[40] B. G. Johnson, P. M. W. Gill, and J. A. Pople, J. Chem. Phys. **98**, 5612 (1993).
[41] C. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch, J. Comput. Chem. **17**, 49 (1996).
[42] F. Eckert, P. Pulay, and H.-J. Werner, J. Comput. Chem. **18**, 1473 (1997).
[43] V. Bakken and T. Helgaker, J. Chem. Phys. **117**, 9160 (2002).
[44] Ö. Farkas and H. B. Schlegel, J. Chem. Phys. **109**, 7100 (1998).
[45] Ö. Farkas and H. B. Schlegel, J. Chem. Phys. **111**, 10806 (1999).
[46] J. Baker, D. Kinghorn, and P. Pulay, J. Chem. Phys. **110**, 4986 (1999).
[47] S. R. Billeter, A. J. Turner, and W. Thiel, Phys. Chem. Chem. Phys. **2**, 2177 (2000).
[48] K. Németh, O. Coulaud, G. Monard, and J. G. Ángyán, J. Chem. Phys. **113**, 5598 (2000).
[49] K. Németh, O. Coulaud, G. Monard, and J. G. Ángyán, J. Chem. Phys. **114**, 9747 (2001).
[50] O. Farkas and B. H. Schlegel, J. Mol. Struct.: THEOCHEM **666–667**, 31 (2003).
[51] C. L. Moss and X. Li, J. Chem. Phys. **129**, 114102 (2008).
[52] H. M. Senn and W. Thiel, Angew. Chem., Int. Ed. **48**, 1198 (2009).
[53] R. Fletcher, *Practical Methods of Optimization* Vol. 1, Unconstrained Optimization (Wiley, New York, 1981), Vol. 1.
[54] R. Lindh, A. Bernhardsson, G. Karlström, and P.-Å. Malmqvist, Chem. Phys. Lett. **241**, 423 (1995).
[55] H. J. A. Jensen, P. Jørgensen, and T. Helgaker, J. Chem. Phys. **85**, 3917 (1986).
[56] See: http://www.kjemi.uio.no/software/dalton/dalton.html for DALTON, an *ab initio* electronic structure program, Release 2.0, 2005.
[57] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).
[58] L. Dougan, G. Feng, H. Lu, and J. Fernandez, Proc. Natl. Acad. Sci. U.S.A. **105**, 3185 (2008).
[59] K. Eichkorn, F. Weigend, O. Treutler, and R. Ahlrichs, Theor. Chim. Acta **97**, 119 (1997).
[60] H. V. R. Dias and H. L. Lu, Inorg. Chem. **34**, 5380 (1995).
[61] J. Anichina, A. Krapp, E. Uggerud and D. K. Boehme (submitted).
[62] T. van der Wijst, C. F. Guerra, M. Swart, F. M. Bickelhaupt, and B. Lippert, Angew. Chem. **121**, 3335 (2009).
[63] J. J. P. Stewart, J. Mol. Model. **15**, 765 (2009).
[64] F. Weigend, Phys. Chem. Chem. Phys. **8**, 1057 (2006).