JCTC Journal of Chemical Theory and Computation

A Parallel Iterative Method for Computing Molecular Absorption Spectra

Peter Koval,*,[†] Dietrich Foerster,[‡] and Olivier Coulaud[§]

CNRS, HiePACS project, LaBRI, 351 Cours de la Liberation, 33405, Talence, France,
CPMOH, University of Bordeaux 1, 351 Cours de la Liberation, 33405, Talence, France,
and INRIA SUD OUEST, HiePACS project, 351 Cours de la Liberation,
33405, Talence, France

Abstract: We describe a fast parallel iterative method for computing molecular absorption spectra within TDDFT linear response and using the LCAO method. We use a local basis of "dominant products" to parametrize the space of orbital products that occur in the LCAO approach. In this basis, the dynamic polarizability is computed iteratively within an appropriate Krylov subspace. The iterative procedure uses a matrix-free GMRES method to determine the (interacting) density response. The resulting code is about 1 order of magnitude faster than our previous full-matrix method. This acceleration makes the speed of our TDDFT code comparable with codes based on Casida's equation. The implementation of our method uses hybrid MPI and OpenMP parallelization in which load balancing and memory access are optimized. To validate our approach and to establish benchmarks, we compute spectra of large molecules on various types of parallel machines. The methods developed here are fairly general, and we believe they will find useful applications in molecular physics/chemistry, even for problems that are beyond TDDFT, such as organic semiconductors, particularly in photovoltaics.

1. Introduction

The standard way to investigate the electronic structure of matter is by measuring its response to external electromagnetic fields. To describe the electronic response of molecules, one may use time-dependent density functional theory (TDDFT).[1] TDDFT has been particularly successful in the calculation of absorption spectra of finite systems such as atoms, molecules, and clusters,[1,2] and for such systems it remains the computationally cheapest *ab initio* approach without any empirical parameters.

In the framework of TDDFT,[3] time-dependent Kohn–Sham-like equations replace the Kohn–Sham equations of the static density-functional theory (DFT). Although these equations can be applied to very general situations,[4] we will restrict ourselves to the case where the external interaction

with light is small. This condition is satisfied in most practical applications of spectroscopy and can be treated by the linear response approximation.

TDDFT focuses on the time-dependent electron density $n(\mathbf{r}, t)$. One assumes the existence of an artificial time-dependent potential $V_{KS}(\mathbf{r}, t)$ in which (equally artificial) noninteracting reference electrons acquire exactly the same time-dependent density $n(\mathbf{r}, t)$ as the interacting electrons under study. The artificial time-dependent potential $V_{KS}(\mathbf{r}, t)$ is related to the true external potential $V_{ext}(\mathbf{r}, t)$ by the following equation:

$$V_{KS}(\mathbf{r}, t) = V_{ext}(\mathbf{r}, t) + V_{H}(\mathbf{r}, t) + V_{xc}(\mathbf{r}, t)$$

Here, $V_{H}(\mathbf{r}, t)$ is the Coulomb potential of the electronic density $n(\mathbf{r}, t)$, and $V_{xc}(\mathbf{r}, t)$ is the exchange-correlation potential. The exchange-correlation potential absorbs all the nontrivial dynamics, and in practice it is usually taken from numerical studies of the interacting homogeneous electron gas.

---

* To whom correspondence should be addressed E-mail: koval.peter@gmail.com.
[†] CNRS, HiePACS project, LaBRI.
[‡] CPMOH, University of Bordeaux 1.
[§] INRIA SUD OUEST, HiePACS project.

A Parallel Iterative Method

J. Chem. Theory Comput., Vol. 6, No. 9, 2010 **2655**

The above time-dependent extension of the Kohn–Sham equation was used by Gross et al.[5] to find the dynamic linear response function $\chi = \delta n(\mathbf{r}, t)/\delta V_{\text{ext}}(\mathbf{r}', t')$ of an interacting electron gas, a response function that expresses the change of the electron density $\delta n(\mathbf{r}, t)$ upon an external perturbation $\delta V_{\text{ext}}(\mathbf{r}', t')$. From the change $\delta n(\mathbf{r}, t)$ of the electron density, one may calculate the polarization $\delta \mathbf{P} = \int \delta n(\mathbf{r}, t)\mathbf{r} \, d^3r$ that is induced by an external field $\delta V_{\text{ext}}(\mathbf{r}, t)$. The imaginary part of its Fourier transform $\delta \mathbf{P}(\omega)$ provides us with the absorption coefficient, and the poles of its Fourier transform provide information on electronic transitions.

Most practical implementations of TDDFT in the regime of linear response are based on Casida's equations.[6,7] Casida has derived a Hamiltonian in the space of particle-hole excitations, the eigenstates of which correspond to the poles of the interacting response function $\chi(\mathbf{r}, \mathbf{r}', \omega)$. Although Casida's approach has an enormous impact in chemistry,[7] it is computationally demanding for large molecules. This is so because Casida's Hamiltonian acts in the space of particle-hole excitations, the number of which increases as the square of the number of atoms.

Alternatively, one may also solve the TDDFT linear response by iterative methods. A first example of an iterative method for computing molecular spectra is the direct calculation of the density change $\delta n(\mathbf{r}, \omega)$ in a modified Sternheimer approach.[1,7] In this scheme, one determines the variation of the Kohn–Sham orbitals $\delta \psi(\mathbf{r}, t)$ due to an external potential without using any response functions. By contrast, use of the response functions allowed van Gisbergen et al.[8,9] to develop a self-consistent iterative procedure for the variation of the density $\delta n(\mathbf{r}, t)$ without invoking the variation of the molecular orbitals. The absorption and dichroism properties have been extensively calculated with the response function[10] including iterative calculations. There exists also an iterative approach based on the density matrix.[11,12] This approach is quite different from ours, but its excellent results, obtained with a plane-wave basis, serve as a useful test of our LCAO-based method.

Over the past few years, the authors of the present paper developed and applied a new basis in the space of products that appear in the application of the LCAO method to excited states.[13–15] This methodological improvement allows for a simplified iterative approach for computing molecular spectra, and the present paper describes this approach. We believe that the methods developed here will be useful in molecular physics, not only in the context of TDDFT but also for systems such as large organic molecules that, because of their excitonic features, require methods beyond ordinary TDDFT.

This paper is organized as follows: In section 2, we briefly recall the TDDFT linear response equations. In section 3, we introduce a basis in the space of products of atomic orbitals. In section 4, we describe our iterative method of computing the polarizability, and in section 5, we explain how the interaction kernels are computed. Section 6 describes the parallelization of our iterative method in both multithread and multiprocess modes. In section 7, we present results and benchmarks of the parallel implementation of our code. We conclude in section 8.

2. Brief Review of Linear Response in TDDFT

To find the equations of the TDDFT linear response, one starts from the time-dependent extension of the Kohn–Sham equation that was already mentioned in the introduction and which we rewrite as follows:

$$V_{\text{KS}}([n], \mathbf{r}, t) = V_{\text{ext}}([n], \mathbf{r}, t) + V_{\text{H}}([n], \mathbf{r}, t) + V_{\text{xc}}([n], \mathbf{r}, t)$$

The notation $[n]$ indicates that the potentials V_{KS}, V_{H}, and V_{xc} in this equation depend on the distribution of electronic charge $n(\mathbf{r}, t)$ in all of space and at all times. To find out how the terms of this equation respond to a small variation, $\delta n(\mathbf{r}, t)$, of the electron density, we take their variational derivative with respect to the electron density:

$$\frac{\delta V_{\text{KS}}([n], \mathbf{r}, t)}{\delta n(\mathbf{r}', t')} = \frac{\delta V_{\text{ext}}([n], \mathbf{r}, t)}{\delta n(\mathbf{r}', t')} + \frac{\delta}{\delta n(\mathbf{r}', t')}[V_{\text{H}}([n], \mathbf{r}, t) + V_{\text{xc}}([n], \mathbf{r}, t)]$$

The important point to notice here is that $\delta V_{\text{KS}}/\delta n$ and $\delta V_{\text{ext}}/\delta n$ are inverses of $\chi_0 = \delta n/\delta V_{\text{KS}}$ and $\chi = \delta n/\delta V_{\text{ext}}$ that represent the response of the density of free and interacting electrons to, respectively, variations of the potentials V_{KS} and V_{ext}. As the response χ_0 of free electrons is known and since we wish to find the response χ of interacting electrons, we rewrite the previous equation compactly as follows:

$$\chi^{-1} = \chi_0^{-1} - f_{\text{Hxc}} \tag{1}$$

Here, an interaction kernel $f_{\text{Hxc}} = \delta/\delta n[V_{\text{H}} + V_{\text{xc}}]$ is introduced. Equation 1 has the form of a Dyson equation that is familiar from many-body perturbation theory.[16]

To make use of eq 1, it remains to specify χ_0 and f_{Hxc}. The Kohn–Sham response function χ_0 can be computed in terms of orbitals and eigenenergies,[1] as will be discussed later in this section (see eq 3 below). The exchange-correlation kernel $\delta V_{\text{xc}}/\delta n$ has to be approximated in practice. In this work, we use the simplest adiabatic local density approximation (ALDA) for the exchange-correlation potential $V_{\text{xc}}([n], \mathbf{r}, t) = V_{\text{xc}}(n(\mathbf{r}, t))$, and the kernel f_{Hxc} becomes:

$$f_{\text{Hxc}} = \frac{\delta(t - t')}{|\mathbf{r} - \mathbf{r}'|} + \delta(r - r') \, \delta(t - t')\frac{dV_{\text{xc}}}{dn}$$

Moreover, the exact-exchange functional can be easily formulated on the basis of dominant products, although this is beyond the scope of the present work.

In principle one could determine the interacting response function $\chi(\mathbf{r}, t, \mathbf{r}', t')$ from eq 1, determine the variation of density $\delta n(\mathbf{r}, t) = \int \chi(\mathbf{r}, t, \mathbf{r}', t') \, \delta V_{\text{ext}}(\mathbf{r}', t')d^3r'dt'$ due to a variation of the external potential δV_{ext} and find the observable polarization $\delta \mathbf{P} = \int \delta n(\mathbf{r}, t)\mathbf{r} \, d^3r$. To realize these operations in practice, we use a basis of localized functions to represent the space dependence of the functions $\chi_0(\mathbf{r}, \mathbf{r}', \omega)$ and $f_{\text{Hxc}}(\mathbf{r}, \mathbf{r}')$.

In order to introduce such a basis into the Petersilka–Gossmann–Gross equation, eq 1, we eliminate the inversions in this equation and transform to the frequency domain:

2656 *J. Chem. Theory Comput., Vol. 6, No. 9, 2010*

Koval et al.

$$\chi(\mathbf{r}, \mathbf{r}', \omega) = \chi_0(\mathbf{r}, \mathbf{r}', \omega) +$$
$$\int d^3 r_1 \, d^3 r_2 \, \chi_0(\mathbf{r}, \mathbf{r}_1, \omega) \, f_{\mathrm{Hxc}}(\mathbf{r}_1, \mathbf{r}_2) \, \chi(\mathbf{r}_2, \mathbf{r}', \omega) \quad (2)$$

The density response function of free electrons $\chi_0(\mathbf{r}, \mathbf{r}', \omega)$ can be expressed[1] in terms of molecular Kohn–Sham orbitals as follows:

$$\chi_0(\mathbf{r}, \mathbf{r}', \omega) = \sum_{E,F} (n_F - n_E) \frac{\psi_E(\mathbf{r}) \, \psi_F(\mathbf{r}) \, \psi_F(\mathbf{r}') \, \psi_E(\mathbf{r}')}{\omega - (E - F) + i\varepsilon} \quad (3)$$

Here, n_E and $\psi_E(\mathbf{r})$ are occupation factors and Kohn–Sham eigenstates of eigenenergy E, and the constant ε regularizes the expression, giving rise to a Lorentzian shape of the resonances. The eigenenergies E and F are shifted in such a way that occupied and virtual states have, respectively, negative and positive energies. Only pairs E,F of opposite signs, $E \cdot F < 0$, contribute in the summation, as is appropriate for transitions from occupied to empty states.

We express molecular orbitals $\psi_E(\mathbf{r})$ as linear combinations of atomic orbitals (LCAO):

$$\psi_E(\mathbf{r}) = X_a^E f^a(\mathbf{r}) \quad (4)$$

where $f^a(\mathbf{r})$ is an atomic orbital. The coefficients X_a^E are determined by diagonalizing the Kohn–Sham Hamiltonian that is the output of a prior DFT calculation, and we assume these quantities to be available. For the convenience of the reader, we use Einstein's convention of summing over repeated indices.

3. Treatment of Excited States within LCAO

The LCAO method was developed in the early days of quantum mechanics to express molecular orbitals as linear combinations of atomic orbitals. When inserting the LCAO ansatz (eq 4) into eq 3 to describe the density response, one encounters products of localized functions $f^a(\mathbf{r}) f^b(\mathbf{r})$—a set of quantities that are known to be linearly dependent.

There is extensive literature[17-20] on the linear dependence of products of atomic orbitals. Baerends et al. use an auxiliary basis of localized functions to represent the electronic density.[20,21] Their procedure of fitting densities by auxiliary functions is essential both for solving Casida's equations and in van Gisbergen's iterative approach.

In the alternative approach of Beebe and Linderberg,[17] one forms the overlaps of products $\langle ab|a'b'\rangle$ to disentangle the linear dependence of the products $f^a(r) f^b(r)$. The difficulty with this approach is its lack of locality and the $O(N^4)$ cost of the construction of the overlaps.[22]

Our approach is applicable to numerically given atomic orbitals of finite support, and in the special case of products on coincident atoms, it resembles an earlier construction by Aryasetiawan and Gunnarsson in the muffin tin context.[23] We ensure locality by focusing on the products of atomic orbitals for each overlapping pair of atoms at a time.[13,14] Because our construction is local in space, it requires only $O(N)$ operations. Our procedure removes a substantial part of the linear dependence from the set of products $\{f^a(\mathbf{r}) f^b(\mathbf{r})\}$. As a result, we find a vertex-like identity for the original products in terms of certain "dominant products" $F^\lambda(\mathbf{r})$:

$$f^a(\mathbf{r}) f^b(\mathbf{r}) \sim \sum_{\lambda > \lambda_{\min}} V_\lambda^{ab} F^\lambda(\mathbf{r}) \quad (5)$$

Here, the notation V_λ^{ab} alludes to the fact that the vertex, V_λ^{ab}, was obtained from an eigenvalue problem for the pair of atoms that the orbitals a and b belong to. The condition $\lambda > \lambda_{\min}$ says that the functions corresponding to the (very many) small eigenvalues were discarded. By construction, the vertex V_λ^{ab} is nonzero only for a few orbitals, a and b, that refer to the same pair of atoms that λ belongs to, and therefore, V_λ^{ab} is a sparse object. Empirically, the error in the representation (eq 5) vanishes exponentially fast[24] with the number of eigenvalues that are retained. The convergence is fully controlled by choosing the threshold for eigenvalues $\lambda_{\min}$, and from now on we assume equality in relation 5.

We introduce matrix representations of the response functions χ_0 and χ in the basis of the dominant products $\{F^\mu(\mathbf{r})\}$ as follows:

$$\chi_0(\mathbf{r}, \mathbf{r}', \omega) = \sum_{\mu\nu} F^\mu(\mathbf{r}) \chi_{\mu\nu}^0(\omega) \, F^\nu(\mathbf{r}')$$
$$\chi(\mathbf{r}, \mathbf{r}', \omega) = \sum_{\mu\nu} F^\mu(\mathbf{r}) \chi_{\mu\nu}(\omega) \, F^\nu(\mathbf{r}') \quad (6)$$

For the noninteracting response, $\chi_{\mu\nu}^0(\omega)$, one has an explicit expression

$$\chi_{\mu\nu}^0(\omega) = \sum_{abcd,E,F} (n_F - n_E) \frac{(X_a^E V_\mu^{ab} X_b^F)(X_c^F V_\nu^{cd} X_d^E)}{\omega - (E - F) + i\varepsilon} \quad (7)$$

which can be obtained by inserting the LCAO ansatz (eq 4) and the vertex ansatz (eq 5) into eq 3.

We insert the expansions (eq 6) into the Dyson equation, eq 2, and obtain the Petersilka–Gossmann–Gross equation in matrix form:

$$\chi_{\mu\nu}(\omega) = \chi_{\mu\nu}^0(\omega) + \chi_{\mu\mu'}^0(\omega) f_{\mathrm{Hxc}}^{\mu'\nu'}(\omega) \chi_{\nu'\nu}(\omega) \quad (8)$$

with the kernel $f_{\mathrm{Hxc}}^{\mu\nu}$ defined as

$$f_{\mathrm{Hxc}}^{\mu\nu} = \int d^3 r \, d^3 r' F^\mu(\mathbf{r}) f_{\mathrm{Hxc}}(\mathbf{r}, \mathbf{r}') F^\nu(\mathbf{r}') \quad (9)$$

The calculation of this matrix will be discussed in section 5.

Since molecules are small compared to the wavelength of light, one may use the dipole approximation and define the polarizability tensor

$$P_{ik}(\omega) = \int d^3 r \, d^3 r' \mathbf{r}_i \mathbf{r}'_k \chi(\mathbf{r}, \mathbf{r}', \omega)$$

Moreover, using eq 8, we find an explicit expression for the interacting polarizability tensor $P_{ik}(\omega)$ in terms of the known matrices χ^0 and f_{Hxc}:

$$P_{ik}(\omega) = d_i^\mu \left(\frac{1}{1 - \chi^0(\omega) f_{\mathrm{Hxc}}} \right)_{\mu\nu} \chi_{\nu\nu'}^0(\omega) \, d_k^{\nu'} \quad (10)$$

where the dipole moment has been introduced $d_i^\mu = \int F^\mu(\mathbf{r}) \mathbf{r}_i \, d^3 r$. Our iterative procedure for computing molecular absorption spectra is based on eq 10. Below, we will

A Parallel Iterative Method

J. Chem. Theory Comput., Vol. 6, No. 9, 2010 **2657**

omit Cartesian indices i and k because we compute tensor components $P_{ik}(\omega)$ independently of each other.

4. An Iterative Method for the Calculation of the Dynamical Polarizability

In eq 10, the polarizability $P(\omega)$ is expressed as a certain *average* of the inverse of the matrix $A_\mu^\nu \equiv \delta_\mu^\nu - \chi_{\mu\nu'}^0(\omega) f_{Hxc}^{\nu'\nu}$. A direct inversion of the matrix A_ν^μ is straightforward at this point, but it is computationally expensive and of unnecessary (machine) precision. Fortunately, iterative methods are available that provide the desired result at much lower computational cost and with a sufficient precision.

In fact, we already used an iterative biorthogonal Lanczos method to create an approximate representation of the matrix A_ν^μ, which can be easily inverted within the Krylov subspace under consideration.[14,15] In this work, however, we use a simpler approach of better efficiency to compute the polarizability (eq 10). First, we calculate an auxiliary vector $\mathbf{X} = \mathbf{A}^{-1}\chi^0 d$ by solving a linear system of equations $\mathbf{AX} = \chi^0 d$ with an iterative method of the Krylov type. Second, we compute the dynamic polarizability as a scalar product $P = d^\mu X_\mu$. In this way, we avoid the construction of the full and computationally expensive response matrix χ_0. Below, we give details on this iterative procedure.

4.1. The GMRES Method for the Iterative Solution of a Linear System of Equations. As we explained above, the polarizability $P(\omega)$ can be computed separately for each frequency, by solving the system of linear equations:

$$(1 - \chi^0(\omega)f_{Hxc})\mathbf{X}(\omega) = \chi^0(\omega)d \qquad (11)$$

and by computing the dynamic polarizability as a scalar product $P(\omega) = d^\mu X_\mu(\omega)$.

We apply a generalized minimal residual method (GMRES)[25,26] to solve the linear system of eq 11, which is of the form $\mathbf{AX} = b$. GMRES belongs to the Krylov-type methods[25,27] that represent a large matrix $\mathbf{A}$ in an iteratively built up Krylov-type basis $|0\rangle, |1\rangle...|i\rangle$. The first vector $|0\rangle$ in the basis is chosen equal to $|b\rangle$, while further vectors are computed recursively via $|i\rangle = \mathbf{A}|i - 1\rangle$. As the vectors $|i\rangle = A^i|0\rangle$ are not mutually orthogonal, one may enforce their orthogonality by using the Gram-Schmidt method:

$$|i\rangle = \mathbf{A}|i - 1\rangle - \sum_{j=0}^{i-1} |j\rangle\langle j|\mathbf{A}|i - 1\rangle \qquad (12)$$

The orthonormal basis built in this way is used in the GMRES method to approximately solve the linear system of equations $\mathbf{A}|\mathbf{X}\rangle = |b\rangle$ by minimizing the residual $|r\rangle = \mathbf{A}|\mathbf{X}\rangle - |b\rangle$ within the Krylov-type subspace (eq 12). The minimization of the residual occurs when the equation $\sum_j \langle i|\mathbf{A}|j\rangle\langle j|x\rangle = \langle i|b\rangle$ is satisfied and this set of equations is of much smaller size than the original problem. When the solution in the Krylov subspace $\langle i|x\rangle$ is found, then an approximate solution in the original space can be computed from $|\mathbf{X}\rangle = \sum_i |i\rangle\langle i|x\rangle$.

A suitable stopping criterion is essential for our method, and we tested several criteria in order to keep the number of iterations small and achieve a reliable result at the same time. The conventionally used criterion that $\varepsilon_r = |r|/|b|$ should be small is unreliable when the tolerance threshold is comparatively large ($\varepsilon_r \approx 1\%$). Therefore, we suggest an alternative combined criterion.

A natural stopping criterion for an iterative solver of the linear system of equations $\mathbf{A}|\mathbf{X}\rangle = |b\rangle$ is a condition on the relative error of the solution $\varepsilon_X = |\Delta\mathbf{X}|/|\mathbf{X}|$.

In our case, the quantity of interest is the dynamic polarizability $P = \langle d|\mathbf{X}\rangle$; therefore it is meaningful to impose a limit on the relative error of the polarizability $\varepsilon_P = |\Delta P|/|P|$. Estimations of the errors ε_X and ε_P can be easily obtained because $|\Delta\mathbf{X}\rangle = \mathbf{A}^{-1}|r\rangle$ and $\Delta P = \langle d|\mathbf{A}^{-1}|r\rangle$. We estimate $|\Delta\mathbf{X}\rangle$ and ΔP using a matrix norm,[28] $|\Delta\mathbf{X}\rangle \approx \|\mathbf{A}^{-1}\|\|r\rangle$ and $\Delta P \approx \|\mathbf{A}^{-1}\| \langle d|r\rangle$. We used the Frobenius norm of the Krylov representation of the matrix $\mathbf{A}^{-1}$, $\|\mathbf{A}^{-1}\| \approx (\sum_{ij}|\langle i|\mathbf{A}^{-1}|j\rangle|^2)^{1/2}$ because $\mathbf{A}$ is a non-Hermitian matrix.

Both errors ε_X and ε_P tend to zero in the limit $X \to X_{exact}$, but for a threshold in the range of a few percent, they behave differently. The condition upon ε_P is better tailored to the problem, and it saves unnecessary iterations in many cases. However, in a few cases, the condition upon ε_P fails, while the condition upon ε_X works. Therefore, we use a condition on ε_P in "quick and dirty" runs and a combined condition $\varepsilon_P < \varepsilon_{tolerance}$ and $\varepsilon_X < \varepsilon_{tolerance}$ for reliable results.

In general, iterative methods of the Krylov type involve only matrix−vector products $\mathbf{A}|z\rangle$. For an explicitly given matrix $\mathbf{A}$, the operation $|z\rangle \to \mathbf{A}|z\rangle$ requires $O(N^2)$ operations. Therefore, the whole iterative method will scale as $O(N^2 N_{iter})$, where N_{iter} is the number of iterations until convergence. This is better than direct methods when $N_{iter} \ll N$, because a matrix−matrix multiplication takes $O(N^3)$ operations.

To avoid matrix multiplications, the application of the matrix $\mathbf{A} = 1 - \chi^0(\omega)f_{Hxc}$ to a vector $|z\rangle$ is done sequentially by computing first $|z'\rangle = f_{Hxc}|z\rangle$ and then $\mathbf{A}|z\rangle = |z\rangle - \chi^0(\omega)|z'\rangle$. The kernel matrix f_{Hxc} is computed before the iterative procedure. Because it is frequency-independent, it can be easily stored and reused. By contrast, the response matrix $\chi^0(\omega)$ is frequency-dependent and computationally expensive, and its explicit construction should be avoided. Therefore, only matrix−vector products $\chi^0(\omega)|z\rangle$ will be computed as explained below without ever calculating the full response matrix $\chi^0(\omega)$.

4.2. Fast Application of the Kohn−Sham Response Matrix to Vectors. In previous papers,[14,15] we have described an $O(N^2)$ construction of the entire response function $\chi_{\mu\nu}^0(\omega)$, but the prefactor was large. Paradoxically, the Krylov method presented above allows for a much faster computation of the absorption spectrum, although the cost of matrix−vector products $\chi^0(\omega)|z\rangle$ scales asymptotically as $O(N^3)$ (see below).

The starting point of our construction of the matrix−vector product $\chi^0(\omega)|z\rangle$ is the expression (eq 7) for the Kohn−Sham response matrix in the basis of dominant products:

$$\chi_{\mu\nu}^0(\omega)z^\nu = \sum_{abcd,E,F} (n_F - n_E)\frac{(X_a^E V_\mu^{ab} X_b^F)(X_c^F V_\nu^{cd} X_d^E)}{\omega - (E - F) + i\varepsilon}z^\nu$$

To compute this matrix−vector product efficiently, we decompose its calculation into a sequence of multiplications

$$\chi^0_{\mu\nu}z^\nu \sim \boxed{\sum_E X^E_a V^{ab}_\mu \boxed{\sum_F X^F_b \boxed{X^F_c \boxed{\boxed{V^{cd}_\nu X^E_d} z^\nu}}}}$$
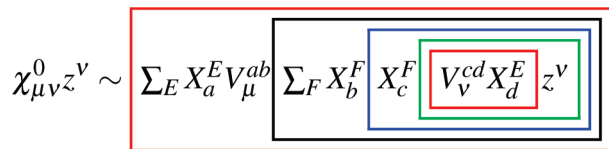
Figure 1. A sequence of operations to compute the matrix−vector product $\chi^0_{\mu\nu}z^\nu$.

Table 1. Complexity and Memory Requirements during the Computation of the Matrix−Vector Product $\chi^0(\omega)z^a$

step	expression	complexity	memory	details of the complexity
1	$\alpha^{cE}_\nu = V^{cd}_\nu X^E_d$	$O(N^2)$	$O(N^2)$	$N_{occ}n^2_{orb}N_{prod}$
2	$\beta^{cE} = \alpha^{cE}_\nu v^\nu$	$O(N^2)$	$O(N^2)$	$N_\omega N_{occ}n_{orb}N_{prod}$
3	$a^{FE} = X^F_c\beta^{cE}$	$O(N^3)$	$O(N^2)$	$N_\omega N_{occ}N_{virt}N_{orb}$
4	$\gamma^E_b = {}_FX^F_ba^{FE}$	$O(N^3)$	$O(N^2)$	$N_\omega N_{occ}N_{virt}N_{orb}$
5	$n_\mu = {}_E\alpha^{bE}_\mu\gamma^E_b$	$O(N^2)$	$O(N)$	$N_\omega N_{occ}n_{orb}N_{prod}$

^{a} There are steps in the sequence with $O(N^2)$ and $O(N^3)$ complexity, where N is the number of atoms. N_ω denotes the number of frequencies for which the polarizability is computed, and the remaining symbols are explained in the text.

that minimizes the number of arithmetical operations by exploiting the sparsity of the vertex V^{ab}_μ. The sequence we chose is graphically represented in Figure 1. For clarity, the frequency-dependent denominator is omitted. Boxes represent the products to be performed at different steps. The innermost box contains a frequency-independent quantity that is also used in the last step. An algebraic representation of the computational steps is given in Table 1.

In the first stage of the algorithm, we compute an auxiliary object $\alpha^{cE}_\nu \equiv V^{cd}_\nu X^E_d$. The vertex V^{cd}_ν is a sparse object by construction. Therefore, we will spend only $N_{occ}n^2_{orb}N_{prod}$ operations, where N_{occ} is the number of occupied orbitals, N_{prod} is the number of products, and n_{orb} is the number of orbitals that belong to a pair of atoms. The auxiliary object α^{cE}_ν is sparse and frequency-independent. Therefore, we store α^{cE}_ν and reuse it in the fifth step of the matrix−vector product. The product $\beta^{cE} \equiv \alpha^{cE}_\nu v^\nu$ will cost $N_{occ}n_{orb}N_{prod}$ operations because each product index ν "communicates" with the atomic orbital index c in one or two atoms. The matrix β^{cE} will be full; therefore the product $a^{FE} \equiv X^F_c\beta^{cE}$ will cost $N_{occ}N_{virt}N_{orb}$; i.e., this step has $O(N^3)$ complexity with N_{virt} being the number of unoccupied (virtual) states. The next step $\gamma^E_b \equiv \sum_F X^F_b a^{FE}$ also has $O(N^3)$ complexity with the same operation count. Finally, the sum $n_\mu \equiv \sum_E X^E_a V^{ab}_\mu \gamma^E_b$ takes only $O(N^2)$ operations because the vertex V^{ab}_μ is sparse. The sequence involves $O(N^3)$ operations, but due to prefactors, the run time is dominated by the $O(N^2)$ part of the sequence for molecules of up to 100 atoms.

4.3. Memory Requirements of the Algorithm. In our previous works,[14,15] we computed TDDFT absorption spectra as a byproduct of the fast construction of the Kohn−Sham response function in $O(N^2N_\omega)$ operations. The memory requirement of the previous algorithm was very high because the use of FFT-based convolutions prohibited a frequency-by-frequency construction. The present work avoids the computation of the full-matrix response, and therefore it requires much less memory. In fact, one needs at most $O(N^2)$ storage in the application of the response function when

following the sequence of matrix operations outlined in subsection 4.2.

The interaction kernels will be computed before the iterative procedure (see section 5), and they also require $O(N^2)$ storage. This is more memory than in van Gisbergen's iterative method.[8] However, as we keep only the dominant products in our approach, N_{prod} is small compared to the number of the original orbital products, and today's computers easily provide the memory needed for our algorithm. The storage of the interaction kernels in memory allows a full exploitation of the computational power of present day machines and implementation of an efficient iterative procedure. Its efficiency is essential because the dynamic polarizability is computed for many frequencies.

5. Calculation of the Interaction Kernels

In the adiabatic local-density approximation (ALDA), the interaction kernel

$$f_H + f_{xc} = \frac{\delta V_H}{\delta n} + \frac{\delta V_{xc}}{\delta n}$$

is a frequency-independent matrix. While the representation of the Hartree kernel is straightforward in our basis

$$f^{\mu\nu}_H = \iint d^3r d^3r' F^\mu(\mathbf{r})\frac{1}{|\mathbf{r} - \mathbf{r'}|}F^\nu(\mathbf{r'}) \qquad (13)$$

the exchange-correlation kernel is not known explicitly and must be approximated. The locality of the LDA potential $V_{xc}(\mathbf{r}) = V_{xc}(n(\mathbf{r}))$ leads to a simple expression for the ALDA exchange-correlation kernel:

$$f^{\mu\nu}_{xc} = \int d^3r\, F^\mu(\mathbf{r}) f_{xc}(\mathbf{r})\, F^\nu(\mathbf{r}) \qquad (14)$$

where $f_{xc}(\mathbf{r})$ is a (nonlinear) function of the density $n(\mathbf{r})$. In this section, we describe how the Hartree and ALDA exchange-correlation kernels are computed.

5.1. The Hartree Kernel. The basis functions $F^\mu(\mathbf{r})$ that appear in the interaction kernels 13 and 14 are built separately for each pair of atoms. They are either *local* or *bilocal* depending on whether the atoms of the pair coincide or not. While the local products are spherically symmetric functions, the bilocal products possess only axial symmetry. Because of their axial symmetry, the bilocal products have a particularly simple representation in a *rotated* coordinate system

$$F^\mu(\mathbf{r}) = \sum_{j=0}^{j_{cutoff}} F^\mu_j(r')\, Y_{jm}(\mathbf{R}\mathbf{r'}), \quad \mathbf{r'} = \mathbf{r} - \mathbf{c} \qquad (15)$$

where the rotation $\mathbf{R}$ and the center $\mathbf{c}$ depend on the atom pair. In the rotated frame, the Z axis coincides with the line connecting the atoms in the pair. We use radial products $F^\mu_j(r)$ that are given on a logarithmic grid. The local products have a simpler, LCAO-like representation

$$F^\mu(\mathbf{r}) = F^\mu(|\mathbf{r} - \mathbf{c}|)Y_{jm}(\mathbf{r} - \mathbf{c}) \qquad (16)$$

where the centers $\mathbf{c}$ coincide with the center of the atom.

A Parallel Iterative Method

J. Chem. Theory Comput., Vol. 6, No. 9, 2010 **2659**

Using the algebra of angular momentum, we can get rid of the rotations $\mathbf{R}$ in the bilocal basis functions (eq 15) and transform the kernel (eq 13) into a sum over two center Coulomb integrals $Cb(1, 2)$

$$Cb(1, 2) = \iint d^3r_1 d^3r_2 g_{l_1 m_1}(\mathbf{r}_1 - \mathbf{c}_1)|\mathbf{r}_1 - \mathbf{r}_2|^{-1} g_{l_2 m_2}(\mathbf{r}_2 - \mathbf{c}_2) \tag{17}$$

The elementary functions $g_{lm}(\mathbf{r}) = g_l(r)Y_{lm}(\mathbf{r})$ have a radial-angular decomposition similar to local products (eq 16). The Coulomb interaction (eq 17) becomes local in momentum space

$$Cb(1, 2) = \int d^3p\, g_{l_1 m_1}(\mathbf{p})p^{-2} \exp(i\mathbf{p}(\mathbf{c}_1 - \mathbf{c}_2)) g_{l_2 m_2}(\mathbf{p}) \tag{18}$$

where the Fourier image of an orbital $g_{lm}(\mathbf{p})$ conserves its spherical symmetry

$$g_{lm}(\mathbf{p}) = i g_l(p)\, Y_{lm}(\mathbf{p}) \tag{19}$$

The radial part $g_l(p)$ is given by the Hankel transform of the original radial orbital in coordinate space:

$$g_l(p) = \sqrt{\frac{2}{\pi}} \int_0^\infty g_l(r)\, j_l(pr) r^2 \, dr \tag{20}$$

Inserting expression 19 into eq 18, expanding the plane wave $\exp(i\mathbf{p}(\mathbf{c}_1 - \mathbf{c}_2))$ in spherical harmonics, and using the algebra of angular momentum, we can reduce the integration in momentum space to a one-dimensional integral:

$$I_{l_1, l_2, l}(R) = \int_0^\infty g_{l_1}(p)\, j_l(pR)\, g_{l_2}(p)\, dp \tag{21}$$

where $R \equiv |\mathbf{c}_1 - \mathbf{c}_2|$.

We distinguish two cases in treating this integral according to whether the orbitals $g_{lm}(\mathbf{r})$ are overlapping or not. For overlapping orbitals, we compute the integral (eq 21) numerically using Talman's fast Hankel transform.[29] For nonoverlapping orbitals, one can compute the integral (21) exactly using a multipole expansion. To derive this expansion, we replace the functions $g_l(p)$ in the eq 21 with their Hankel transforms (20):

$$I_{l_1, l_2, l}(R) =$$
$$\frac{2}{\pi} \int_0^\infty dr_1\, g_{l_1}(r_1) r_1^2 \int_0^\infty dr_2\, g_{l_2}(r_2) r_2^2 \int_0^\infty j_{l_1}(r_1)\, j_l(pR)\, j_{l_2}(r_2)\, dp \tag{22}$$

The integral over three spherical Bessel functions $I_{l_1, l_2, l}(r_1, r_2, R) \equiv \int_0^\infty j_{l_1}(r_1)\, j_l(pR)\, j_{l_2}(r_2)\, dp$ reduces to a simple separable expression[30] provided two conditions are satisfied, $R > r_1 + r_2$ (the basis functions do not overlap) and $0 \le l \le l_1 + l_2$ (the triangle relation for angular momentum). Under these conditions, we obtain

$$I_{l_1, l_2, l}(r_1, r_2, R) = \delta_{l, l_1 + l_2} \frac{\pi^{3/2}}{8} \frac{r_1^{l_1} r_2^{l_2}}{R^{l+1}} \frac{\Gamma(l + 1/2)}{\Gamma(l_1 + 3/2)\, \Gamma(l_2 + 3/2)} \tag{23}$$

Inserting this result into eq 22, we obtain an expression for the Coulomb interaction in terms of moments in closed form

$$\rho_l \equiv \int_0^\infty r^2 \, dr\, g_l(r) r^l \tag{24}$$

The moments (eq 24) can be computed and stored at the beginning of the calculation. Therefore, the calculation will consist of summing the angular-momentum coefficients, and this is clearly much faster than a direct numerical integration in eq 21.

The complexity of the near-field interactions will be proportional to the number of atoms N. The calculation of the multipoles (24) for the far-field interaction requires $O(N)$ mathematical operations. The remaining part of the far-field interactions (Wigner rotations) scales as N^2 with the number of atoms.

5.2. The ALDA Exchange-Correlation Kernel. Unlike the Hartree kernel, the exchange-correlation kernel (eq 14) is local, and we therefore compute it directly in coordinate space by numerical quadrature. The supports of the dominant products $F^u(\mathbf{r})$ and $F^v(\mathbf{r})$ in the integrand of eq 14 have, in general, the shape of a lens. Therefore, the support of the integrand will generally be an intersection of two lenses.

We found, however, that integration in spherical coordinates gives sufficiently accurate and quickly convergent results. This is because the important matrix elements involve neighboring dominant products and the support of the dominant products is large compared to the distance between their centers. Therefore, the integrands of the important matrix elements (eq 14) have nearly spherical support. This situation is illustrated by the cartoon in Figure 2.

For each pair of dominant products, we use spherical coordinates that are centered at the midpoint between them. We use Gauss−Legendre quadrature for integrating along the radial coordinate and Lebedev quadrature[31] for integrating over the solid angle

$$F_{xc} = \sum_{i=1}^{N_r} G_i \sum_{j=1}^{N_\Omega} L_j f_{xc}(r_i, \Omega_j) \tag{25}$$

Here, G_i and r_i are weights and knots of Gauss−Legendre quadrature, and L_j and Ω_j are weights and knots of Lebedev quadrature. The number of points $N_r \times N_\Omega$ can be kept reasonably small (24×170 by default).

The most time-consuming part of the exchange-correlation kernel is the electronic density. We found that calculating the density using the density matrix

$$n(\mathbf{r}) = \sum_{ab} f^a(\mathbf{r}) D_{ab} f^b(\mathbf{r}), \quad \text{where } D_{ab} = \sum_{E<0} X_a^E X_b^E \tag{26}$$

provides a linear scaling of the run time of f_{xc} with the number of atoms. However, a calculation of the density via molecular orbitals

$$n(\mathbf{r}) = \sum_{E<0} \psi_E(\mathbf{r})\, \psi_E(\mathbf{r}), \quad \text{where } \psi_E(\mathbf{r}) = \sum_a X_a^E f^a(\mathbf{r}) \tag{27}$$

is faster in many cases, although the run time scales quadratically with the number of atoms. In order to optimize the run time, rather than insist on linear scaling, we choose the calculational approach automatically depending on the
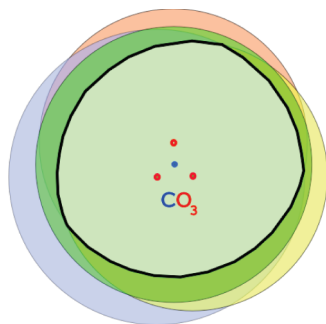
Figure 2. The spatial support of the integrand of the exchange correlation term (eq 14) depends on the support of its underlying atomic orbitals and on the geometry of the quadruplet of atoms under consideration. For neighboring atoms, the support of the orbitals is several times larger than their interatomic distances, and this results in a nearly spherical support of the integrand. The figure illustrates the case of (hypothetical) CO_3.

Construction of the dominant products
for *atom species* **do**
    Build local dominant products
!\$OMP PARALLEL DO
for *atom pairs* (a, b) **do**
    Build bilocal dominant products

Computation of the interaction kernels f_H **and** f_{xc}
!\$OMP PARALLEL DO
for *each couple of atom pairs* $(a, b; c, d)$ **do**
    Build block $(a, b; c, d)$ of kernels.

Computation of the dynamical polarizability
!\$OMP PARALLEL DO
for $\omega \in [\omega_{\text{begin}}, \omega_{\text{end}}]$ **do**
    Solve for $|X\rangle$: $\quad (1 - \chi^0(\omega) f_{\text{Hxc}}) |X\rangle = \chi^0(\omega)|d\rangle$
    Compute polarizability $P(\omega) = \langle d|X\rangle$

Figure 3. Skeleton of the algorithm. Its first (and computationally easiest) part is the construction of the dominant products. The second (and computationally most demanding) part is the construction of the interaction kernels. The third (and comparatively easy) part is the iterative calculation of the dynamic polarizabilities $P(\omega)$.

geometry of the molecule. For instance, in the case of a long polythiophene with 13 chains (see subsection 7.2), we use the $O(N)$ method, while in the other examples, it is better to use the $O(N^2)$ approach.

6. Parallelization of the Algorithm

The overall structure of our algorithm is given in Figure 3. First, the basis of dominant products is built. Then, the interaction kernels are computed, and finally both are used in the iterative procedure to compute the dynamic polarizability.

The individual components of the algorithm in Figure 3 suggest different parallelization strategies. The dominant products are built for each atom pair independently; therefore the corresponding code is parallelized over atom pairs. The

structure of the dominant products suggests a blockwise computation of the interaction kernels. These blocks are mutually independent; therefore we parallelize their construction. The dynamic polarizabilities are calculated independently for each frequency and are, therefore, parallelized over frequencies. Below, we go through the details of the algorithm and its hybrid OpenMP/MPI parallelization.

6.1. Multithread Parallelization. Modern computers are faster than previous generations of machines mainly due to their parallel design, as in the case of general-purpose multicore processors. For specially written programs, such a design allows several tasks or "threads" to run simultaneously. Fortunately, it is easy to write a multithreaded program using the current application programming interface OpenMP. Moreover, in OpenMP, data exchange between threads uses the common memory, and it is, therefore, faster than that on distributed-memory machines. For these reasons, our main emphasis here is on multithreaded (or shared-memory) parallelization. We use the OpenMP standard[32] that allows for an efficient parallelization of all three sections of the algorithm in Figure 3.

6.1.1. Building the Basis of Dominant Products. The construction of local dominant products involves only the atomic species that occur in a molecule. Therefore, it is computationally cheap, and any parallelization would only slow down their construction.

For bilocal dominant products, the situation is different. The construction of bilocal dominant products is done for all atom pairs, the orbitals of which overlap. Because these pairs are independent of each other, we parallelize the loop over pairs with OpenMP directives.

The dominant products $F^\mu(\mathbf{r})$ and vertices V_μ^{ab} are stored in suitably chosen data structures to allow for effective use of memory. Due to the locality of our construction, the amount of memory spent in the storage of dominant products and vertices grows linearly with the number of atoms.
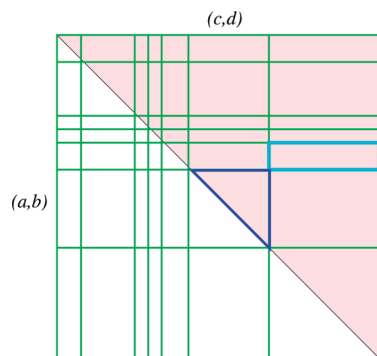
6.1.2. Construction of the Interaction Kernels. The interaction kernels 13 and 14 refer to a pair of dominant products $F^\mu(\mathbf{r})$ and $F^\nu(\mathbf{r})$. Because each of the dominant products in turn refers to a pair of atoms, the interaction kernel splits into blocks that are labeled by a quadruplet of atoms $(a, b; c, d)$. In practice, this is used to precompute and reuse some auxiliary quantities that belong to such a quadruplet. The block structure is schematically depicted in Figure 4. Generically, a block is rectangular, but when two pairs coincide, the block reduces to a triangle because of reflection symmetry.

The calculation of the interaction kernels is parallelized over blocks using dynamic scheduling. Because the computational load of each block is proportional to its area, we minimize the waiting time at the end of the loop by a descending sort of the blocks according to their area.

6.1.3. Computation of the Dynamic Polarizability. According to eq 10, the dynamic polarizability $P_{ik}(\omega)$ is computed independently for each frequency ω. Therefore, the loop over frequencies in the algorithm in Figure 3 is embarrassingly parallel.

In the parallelization over frequencies, we had to make thread-safe copies of module variables (working arrays in

Figure 4. The block structure of the interaction kernels. The blocks are defined by two atom pairs (a, b) and (c, d). The size of blocks is known from the construction of the dominant products. Because the entire interaction kernel $f_{Hxc}^{\mu\nu}$ is symmetric, only its upper triangular part is computed.

the GMRES solver) using the OpenMP directive threadprivate. This multiplies the memory requirement by the number of threads, but this poses no problem, because this part of the algorithm is not memory-intensive.

The number of iterations to reach convergence of the polarizability tensor (eq 10) varies with frequency and with the Cartesian components in an irregular way. To take this into account, we use a dynamic schedule with a single frequency and tensor component per thread. By treating tensor components on the same level as frequencies, we reduce the body of a loop by a factor of 3 provided only diagonal components are computed.

6.2. Hybrid MPI-Thread Parallelization. Most current supercomputers are organized as clusters of multicore nodes that are interconnected by a high-speed network. Although the number of cores has grown over the years, we still need several nodes for greater computational speed and to provide sufficient memory.

Our program has also been adapted to such distributed-memory parallel machines. We parallelize according to the Single Program Multiple Data (SPMD) paradigm with the message passing interface (MPI) to speed up only the computationally intensive parts of the algorithm—the construction of the bilocal products, the calculation of the kernels f_H and f_{xc}, and the iterative procedure. Moreover, each MPI process uses the multithread parallelism described above.

6.2.1. Parallelization of the Basis of Dominant Products. Only bilocal dominant products must be constructed in parallel. As described in subsection 6.1.1, the construction is naturally parallel in terms of atom pairs. Therefore, we distribute the atom pairs prior to computation and gather data after the computation in order to duplicate basis functions on each MPI process.

6.2.2. Parallelization of the Interaction Kernels. As explained in subsection 6.1.2, the interaction kernel depends on quadruplets of atoms, and it would appear natural to parallelize their construction over these quadruplets. We found it advantageous, however, to slice the interaction matrix into vertical bands that belong to one or more atom pairs and process them on different nodes. After the computation, the complete matrix of interactions is reconstituted on each node.

The optimal size of each vertical band is determined by an estimate of the work load prior to the computation. Since the Hartree and the exchange-correlation kernel differ in their properties (such as locality), their matrices are distributed differently. In both cases, however, the total work load is the sum of the work loads of its constituent quadruplets.

In the case of the Hartree kernel (eq 13), the work load of a block depends on its size and on whether its atom pairs are local or bilocal. While the local products are of the simple LCAO type (eq 16), the bilocal products (eq 15) contain additional summations. We found the following robust estimator of the workload of a block of the Hartree kernel:

$$\text{Workload(Hartree)} = \text{Size\_Of\_Block} \cdot$$
$$(j_{\text{cutoff}} \cdot \Theta(a \neq b) + 1) \cdot (j_{\text{cutoff}} \cdot \Theta(c \neq d) + 1) \quad (28)$$

where $\Theta(a \neq b)$ is equal to 1 for bilocal atom pairs and 0 otherwise; j_{cutoff} is the largest angular momentum in the expansion (eq 15). By default, its upper limit is set to 7.

In the exchange-correlation kernel (eq 14), the domain of integration is the intersection of two lens-like regions. Because there is no simple analytical expression for such a volume, we count the number of integration points within the support for each block and estimate the work load as proportional to the number of points. Rather surprisingly, the run time is independent of the dimension of the block (this is due to precomputing the values of the dominant products).

Because the kernel f_{Hxc} is frequency-independent, it is computed at the beginning of the iterative procedure and duplicated on all the MPI nodes. We found the time for gathering the matrix small compared to the time of its computation.

6.2.3. Parallelization of the Iterative Procedure. As mentioned previously in subsection 6.1.3, the iterative calculation of the polarizability tensor $P_{ik}(\omega)$ is naturally parallel both in its frequency and in its Cartesian tensor indices. Therefore, this part of the algorithm is parallelized over both frequency and tensor indices. However, the workload for each composite index is difficult to predict. To achieve a balanced workload on the average, we distribute this index cyclically over MPI nodes ("round-robin distribution").

7. Results

In this section, we present different absorption spectra of large molecules to validate the method and the parallelization approach. We start in subsection 7.1 by comparing our absorption spectra with previous calculations by other authors and also with experimental data. Then, in subsection 7.2, we present the scaling of the run time with the number of atoms. In subsection 7.3, we examine the efficiency of the hybrid parallelization for a variety of molecules run on different machines. Finally, in subsection 7.4, we present a comparison of absorption spectra of fullerene C_{60} with its derivative PCBM that is often used in organic solar cells.

7.1. Fullerene C_{60}. We already tested the basis of dominant products in our previous works,[13−15] where the absorption spectra of methane, benzene, indigo blue, and
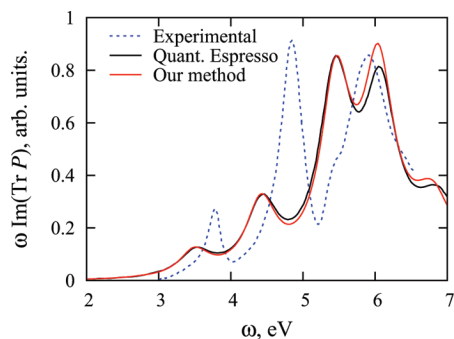
Figure 5. Comparison of the low-frequency absorption spectra of C_{60} fullerene. We see good agreement between the two theoretical predictions, while the experimental curve is shifted by 0.2−0.3 eV.

buckminster fullerene C_{60} were studied. The new element in the present paper is the use of an iterative method without constructing the full Kohn−Sham response function. We now test this method on buckminster fullerene.

Buckminster fullerene C_{60} is of considerable interest in materials science. Among other applications, it is used as an electron acceptor in organic solar cells.[33] Here, we compare our results with spectra from the Quantum Espresso package and with experimental results.

The absorption spectrum of fullerene C_{60} is shown in Figure 5 where we compare our results with the calculation by Rocca et al.[11] and with experimental results.[34] One can see a good agreement between the theoretical spectra, while both theoretical predictions deviate from the experimental data by 0.2−0.3 eV. The shift in the spectrum might be due to the solvent in the experimental setup or due to the inadequacy of the simplest LDA functional for this large molecule. We used DFT data from the SIESTA package,[35] where the pseudopotentials of the Troullier−Martin type and the LDA functional by Perdew and Zunger[36] are applied. A double-ζ polarized basis set has been used, and the broadening of levels has been set to 0.019 Ry. Our program spent a total of about 62.5 min on this calculation, of which 2098 s were spent on the Coulomb kernel, 1085 s on the exchange-

correlation kernel, and 500 s in the iterative procedure, i.e., approximately 2.27 s per frequency. The convergence parameter for the polarizability was set to 1%. With this convergence parameter, the dimension of the Krylov space was varying from 7 to 12, with an average of 8, while the dimension of the dominant product space was 8700. In this test, we used one thread on a machine with two Intel quad core Nehalem CPUs at 2.93 GHz, with an 8 MB cache, and 48 GB of DDR3 RAM, and consuming no more than 2.3% of RAM (1.2% during the iterative procedure).

7.2. Polythiophene Chains (Complexity of the Method). In sections 4 and 5, we discussed the complexity scaling of different parts of the algorithm theoretically. In this subsection, we measure the dependence of run time on the number of atoms N in polythiophene chains of different lengths. We shall see that the run time scaling follows the theoretical predictions for the complexity.

Sulfur-containing molecules are widely used in organic electronics.[33,37,38] In this work, we study pure polythiophene chains of 3−13 repeating units. The geometry of the longest polythiophene we considered is shown in Figure 6. Our calculations suggest (ignoring the excitonic character of these molecules) that the HOMO−LUMO energy difference decreases, while the absorption increases, with chain length. The calculated absorption spectra are collected in Figure 7.

We now use the calculations on polythiophene spectra in order to study the run time scaling with the number of atoms of different parts of our algorithm. Their scaling behavior will be described in terms of approximate scaling exponents. The run times for a few chains are collected in Table 2 for a machine of four AMD Dual-Core Opteron CPUs at 2.6 GHz with an 8 MB cache, and 32 GB of DDR2 RAM, running sequentially.

The application of the noninteracting response to a vector consists of N^2 and N^3 parts (see subsection 4.2). The total time for the product $\chi^0 z$ is collected in the third column of Table 2, while the run time of the N^3 part is collected in the fourth column. Using the run times, one can compute
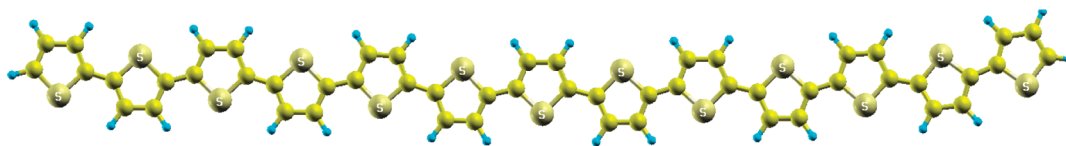
Figure 6. The geometry of the longest polythiophene chain we considered.
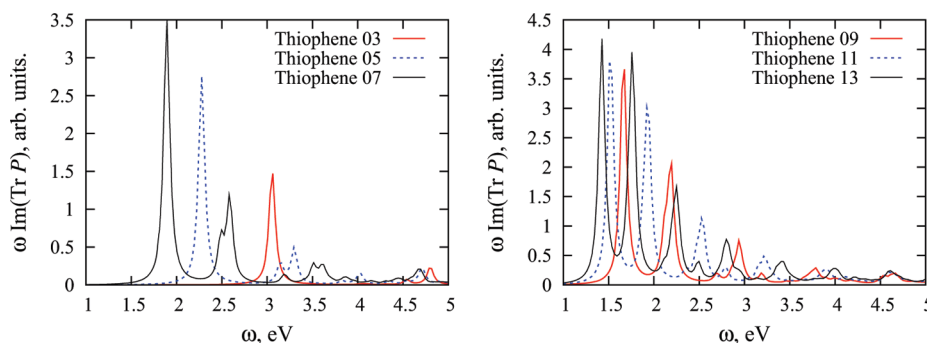
Figure 7. Comparison of low-frequency spectra for several polythiophene chains.

Table 2. Run Time in Different Parts of the Algorithm as a Function of the Number of Atoms N in the Polythiophene Chain[a]

| N	units	$\chi^0 z$, s	N^3 part in $\chi^0 z$, s	f_H, s	f_{xc}, s
23	3	1.51×10^{-2}	8.20×10^{-3}	157	173
37	5	4.42×10^{-2}	2.52×10^{-2}	417	300
51	7	9.29×10^{-2}	5.58×10^{-2}	807	428
65	9	0.166	0.104	1324	561
79	11	0.260	0.169	1977	694
93	13	0.382	0.255	2767	822

[a] The third column gives the total time for the matrix−vector product ($\chi^0 z$), while an N^3 part that arises in the construction of $\chi^0 z$ is given in the fourth column. The fifth and sixth columns display the run times of the interaction kernels f_H and f_{xc}, respectively.
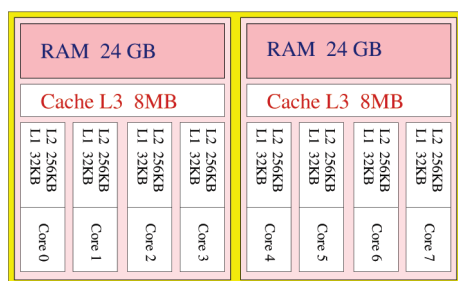
Figure 8. Memory/cache/core structure of Intel Xeon X5550 Nehalem machine. Two quad core nodes have fast access to one of the memory banks, while the internode communication is slower.

exponents x and x_3 for their corresponding scaling laws N^x and N^{x_3}. The exponents x and x_3 vary in the range of $x = 2.31−2.36$ and $x_3 = 2.49−2.53$, respectively. The run time of the Hartree kernel (fifth column) shows a scaling exponent in the range $x_H = 2.05−2.06$, while the run time in the exchange-correlation kernel (sixth column) scales almost linearly $x_{xc} = 1.04−1.12$. Therefore, the measured exponents are close to the predicted exponents $x = 3$, $x_3 = 3$, $x_H = 2$, and $x_{xc} = 1$.

The calculation of the Hartree kernel f_H via multipoles, as explained in subsection 5.1, improves the run time in the case of large molecules but could not improve the run time scaling of the Coulomb interaction. In fact, the Hartree kernel f_H is a nonlocal quantity, and the rotations involved in the bilocal dominant products contribute a substantial part to the run time.
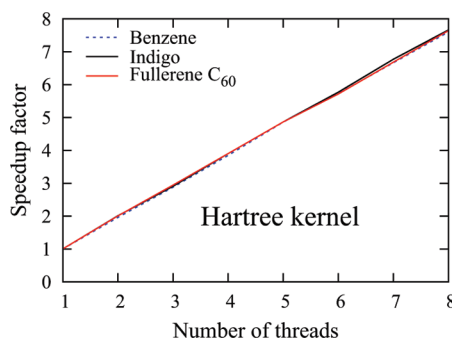
The scaling of the run time for the entire calculation of molecular spectra will also vary with the parameters of the calculation. Obviously, for a small number of frequencies, the scaling of total run time will be determined by the Hartree and exchange-correlation kernels. However, if the number of frequencies is large, then the application of the noninteracting response χ^0 on a vector will dominate the run time.

7.3. Quality of the Parallelization. Our parallel Fortran 90 code is adapted to current parallel architectures (see section 6). In this subsection, we evaluate the quality of the hybrid parallelization by testing our approach on three machines of different architecture. Two machines belong to shared-memory multicore architectures with nonuniform memory access, while the third machine is a cluster with 50 multicore nodes interconnected by an Infiniband network.

The program was compiled using Intel's Fortran compiler and linked against Intel's Math Kernel Library for BLAS, LAPACK, and Fast Fourier Transform libraries.

7.3.1. Multithread Parallelization. We consider parallelization on shared-memory machines as more important than on distributed-memory machines. Therefore, the speedup of our code is first tested on two shared-memory machines with Non-Uniform Memory Architecture (NUMA). The first machine has two quad core Nehalem 5500 CPUs (see Figure 8), while the second machine has 48 dual core Xeon CPUs (see Figure 10).

The speedup on the Nehalem machine is shown in Figure 9 for three molecules of different size: benzene, indigo, and fullerene. The speedup in both kernels and in the generation of bilocal products is good for all molecules; therefore we plot only the speedup for the Hartree kernel on the left panel. By contrast, the iterative procedure exhibits a lower speedup (as shown on the right panel in Figure 9). This is due to the high memory-bandwidth requirement of the iterative procedure. The high memory bandwidth is clearly revealed in two distinct ways of using the same number of cores in a test calculation, see subsection 7.3.2.

In spite of the loss of speedup for larger molecules on the Nehalem machine, we tested our implementation also on a parallel processor with a very large number of cores (see Figure 10). The speedup is shown in Figure 11.

Our results show a satisfactory speedup in both interaction kernels, while the iterative procedure again shows a poorer
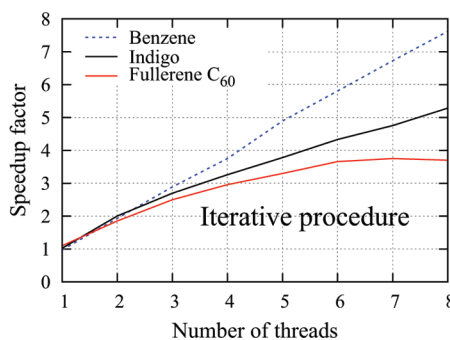
Figure 9. (Left panel) Speedup factor in the Hartree kernel for molecules of different size. The speedup in the exchange-correlation kernel and in the generation of dominant products is similar to the left panel. The right panel shows the speedup factor in the iterative procedure. We observe that the speedup decreases with the size of the molecule.

Figure 10. Memory/cache/core structure of Xeon-96 machine. A total of 48 dual core Xeon CPUs are connected to four memory banks. Although every core can address the whole memory space, the internode communication is slower.

performance due to its high memory-bandwidth requirements, which cannot be satisfied in this NUMA architecture.

7.3.2. Hybrid MPI-Thread Parallelization. Prior to large scale computations with many Nehalem (see Figure 8) nodes, we performed test runs on one node to find an optimal OpenMP/MPI splitting. Four calculations were done on the Nehalem machine using all eight cores of the machine, but differing in the OpenMP/MPI splitting. The results are collected in the Table 3. We observed a small workload unbalance of 8% in case of the Hartree kernel and an even smaller unbalance of 5% in the case of the exchange-correlation kernel. The iterative procedure with the "round robin distribution" of frequencies over the nodes shows an even lower workload unbalance of 2%. The best run time is achieved in a 2/4 hybrid parallelization, i.e., running two processes with four cores each. This optimal configuration reflects the structure of the machine, where each thread shares the same L3 cache. The node has two processors of four cores each and a relatively slower memory access between the processors. This weak internode communication results in an appreciable penalty in an OpenMP-only run (1/8 configuration), because the iterative procedure reads a rather large amount of data ($V_\mu^{ab} X_b^E$) twice during the application of the response function χ_0 (see subsection 4.2).

The high memory-bandwidth requirement is clearly revealed in two distinct ways of using the same number of cores (eight cores), either using all cores on one node or distributing them over two nodes. In the latter case, the memory-bandwidth is higher and the iterative procedure runs considerably faster (92 s versus 137 s in the case of fullerene C_{60}).

Table 3. Run Time (in seconds) in a Hybrid MPI/OpenMP Parallelization for Fullerene C_{60}[a]

proc/thr	domi. prod.	f_H	f_{xc}	iterative proc.	total
1/8	7.3	271	142	145	571
2/4	6.9 (6.9)	273 (267)	142 (141)	109 (108)	538
4/2	6.8 (6.8)	274 (264)	142 (141)	122 (112)	544
8/1	6.8 (6.8)	274 (257)	143 (140)	134 (120)	570

[a] In the parentheses, the smallest run time between the nodes is stated in order to estimate the MPI work load disbalance.

We used the above optimal 2/4 hybrid configuration in a massively parallel calculation on the chlorophyll-a molecule. The speedup due to hybrid OpenMP-MPI parallelization is shown in Figure 12. In this computation, we used up to 50 nodes of recent generation Nehalem machines. According to the previous experiment on fullerene C_{60}, we started two processes per node, each process running with four threads. The two processes were placed on sockets. One can see that the iterative procedure shows the best speedup among other parts of the code, while total run time is governed mainly by the calculation of the exchange-correlation kernel. The absolute run times (including communication time) in the first calculation with one node (two processes) were: total, 4003 s; for the exchange-correlation kernel, 1147 s; for the Hartree kernel, 1247 s; for the iterative procedure, 1574 s; and for the bilocal vertex, 11.8 s. The speedup in the bilocal vertex reaches a maximum at 10 nodes because of increasing communication time.

The starting geometry of the molecule was taken from Sundholm's supplementary data.[39] The geometry was further relaxed in the SIESTA package[35] using Broyden's algorithm until the remaining force was less than 0.04 Ry/Å. The relaxed geometry is shown in Figure 13. In order to achieve this (default) criterion, we had to use a finer internal mesh with a MeshCutoff of 185 Ry. The default DZP basis was used, but to achieve convergence, we used orbitals that are more extended in space than SIESTA's default orbitals. The spatial extension is governed by the parameter PAO. EnergyShift that was set to 0.002 Ry in the present calculation. The spectrum of the chlorophyll-a molecule is seen in Figure 14. Like in the case of fullerene C_{60}, there is excellent agreement between theoretical results that however differ from the experimental data.[40] The low frequency spectrum consists of two bands at 635 and 450 nm. Both bands are due to transitions in the porphyrin. According to our calculation, the first A band consists of two transitions

Figure 11. Speedup on a heavily parallel Xeon-96 machine. The results are satisfactory for the interaction kernels, while the memory-bandwidth requirements of the iterative procedure hamper the parallel performance in the case of the larger fullerene C_{60} molecule.

A Parallel Iterative Method

*J. Chem. Theory Comput., Vol. 6, No. 9, 2010* **2665**

Figure 12. Speedup due to hybrid OpenMP/MPI parallelization for chlorophyll-a. The job was run on up to 50 nodes with two processes per node. The code shows a linear speedup on up to 15 nodes (30 processes, 120 cores). Further increase of the number of nodes results in a steady acceleration of the whole program.


Figure 13. The relaxed geometry of the chlorophyll-a molecule obtained with the SIESTA package using a DZP basis set.
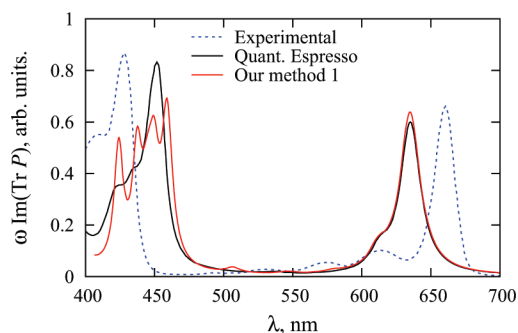
Figure 14. Low frequency absorption spectrum of chlorophyll-a.

between HOMO−LUMO and HOMO−1−LUMO, while the second (so-called Soret) band consist of six transitions to LUMO and LUMO+1 states.

7.4. Fullerene C_{60} versus PCBM. Fullerenes are often modified in order to tune their absorption spectra or their transport properties.[33,41,42] In this work, we compute the absorption spectra of [6,6]-phenyl C61 butyric acid methyl ester (PCBM) and compare it with the spectrum of pure fullerene C_{60}. We use the same parameters as in the case of C_{60} in subsection 7.1. A relaxed geometry of PCBM was obtained using the SIESTA package[35] and using its default convergence criterion (maximal force less than 0.04 eV/Å). Figure 15 shows the relaxed geometry. The absorption spectrum of PCBM is shown in Figures 16 and 17.
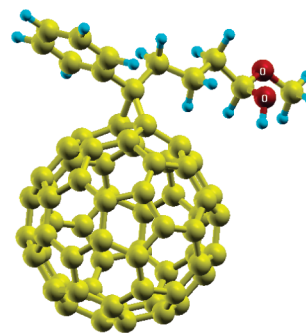
Figure 15. Geometry of PCBM. Relaxation is done in the SIESTA package with Broyden's algorithm.
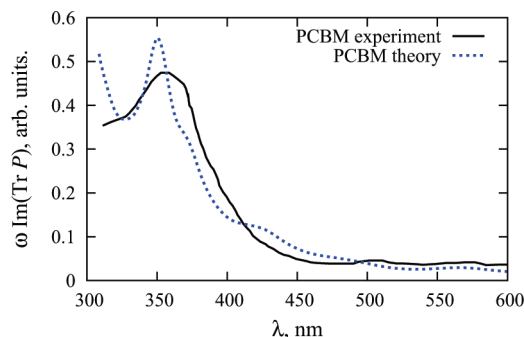
Figure 16. Comparison of the low-frequency spectra for PCBM with experimental data.

Figure 16 shows a comparison of our calculation with recent experimental results.[42] We can see that our results have similar features as the experimental data: the maxima at 350 nm agree well with a broad experimental resonance at 355 nm, and a substantial background at a longer wavelength is present both in the calculated and in the experimental spectrum. In this calculation, we set the damping constant $\varepsilon = 0.08$ Ry and compute the spectrum in the range where experimental data are available. However, in order to better understand the difference introduced by the functional group, we compute the spectra in a broader range of energies with a smaller value of the damping constant $\varepsilon = 0.003$ Ry. The result is shown in Figure 17.

One can see on the left panel of Figure 17 that PCBM absorbs much stronger in the visible range. This is a consequence of symmetry breaking and indicates a modified HOMO−LUMO gap. On the right panel of the figure, one can recognize the main difference between pure and modified fullerene. The high spatial symmetry of pure fullerene leads to a degeneracy of the electronic transitions, and several transitions contribute to the same resonance. The symmetry is broken in the case of PCBM. The degeneracy is lifted, and the spectral weight is spread out.

8. Conclusion and Outlook

In this paper, we have described a new iterative algorithm for computing molecular spectra. The method has two key ingredients. One is a previously constructed local basis in the space of products of atomic LCAO orbitals. The second is the computation of the density response not in the entire space of products, but in an appropriate Krylov subspace.
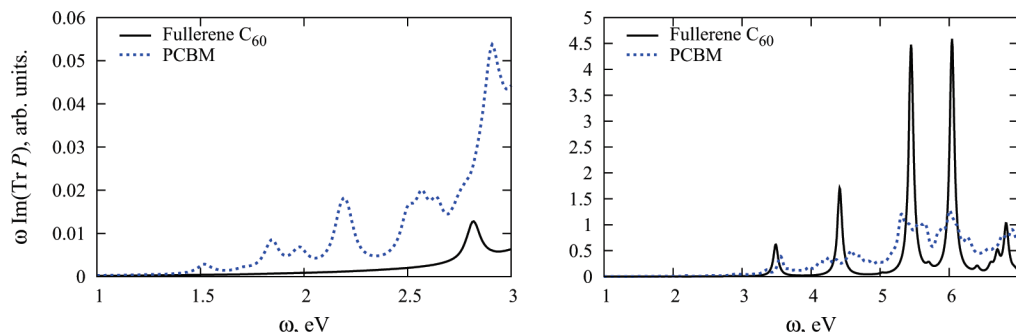
Figure 17. Comparison of the low-frequency spectra C_{60} versus PCBM.

The speed of our code is roughly comparable to TDDFT codes in commercially available software. The reader can judge the speed from the absolute timings we presented above. In general, we expect our method to be faster than the solution of Casida's equation for systems with dense spectra, when the target range contains many allowed transitions.

The algorithm was parallelized and was shown to be suitable for treating molecules of more than 100 atoms on large current heterogeneous architectures using the OpenMP/MPI framework.

Our approach leaves plenty of room for further improvements both in the method and in the algorithm. For example, we did not consider reducing the dimension of the space in which the response function acts, but such a reduction is feasible.

Also, we are working on an adaptive procedure to obtain good spectra with fewer frequency points and to compute the position of the poles and strength of their residues. A reduction in the number of frequencies will allow avoidance of the full calculation of the interaction kernels, replacing them by matrix−vector multiplications. There exist fast multipole methods[43] for computing fast matrix−vector products needed in the Hartree interaction.

Moreover, for large molecules, our embarrassingly parallel approach to compute spectra induces a memory-bandwidth bottleneck. To avoid it, one may parallelize the frequency loop using MPI and parallelize the matrix−vector operations using OpenMP.

More generally, because we do not use Casida's equations, the methods developed here should be useful beyond the TDDFT approach, for instance, in the context of Hedin's GW approximation[44] where Casida's approach is no longer available.

Our framework appears suitable for computing the Hartree−Fock exchange energy and for treating hybrid functionals and their response. In fact, the calculation of the standard Hartree−Fock integrals reduces to the Hartree kernel $f_H^{\mu\nu}$ presented in this paper:

$$\int\int d^3 r\, d^3 r' f^a(\mathbf{r}) f^b(\mathbf{r}) |\mathbf{r} - \mathbf{r}'|^{-1} f^c(\mathbf{r}') f^d(\mathbf{r}') = \sum_{\mu,\nu} V_\mu^{ab} f_H^{\mu\nu} V_\nu^{cd}$$

The calculation takes $O(N^2)$ operations because the vertices V_μ^{ab} are sparse by construction. Introduction of a screened Coulomb potential[45] would give linear scaling. The calcula-

tion of interaction kernels for hybrid functionals in our framework must be reserved to future work, however.

References

(1) Gross, E. K. U.; Burke, K. Basics. In *Time-Dependent Density Functional Theory*, 1st ed.; Marques, M. A. L., Ullrich, C. A., Nogueira, F., Rubio, A., Burke, K., Gross, E. K. U., Eds.; Springer: Berlin, Germany, 2008; pp 243−353 (part IV), p 231, p 9.

(2) Perdew, J. P.; Kurth, S. Density Functionals for Non-relativistic Coulomb Systems in the New Century. In *A Primer in Density Functional Theory*, 1st ed.; Fiolhais, C., Nogueira, F., Marques, M. A. L., Eds.; Springer: Berlin, Germany, 2003; pp 1−55.

(3) Runge, E.; Gross, E. K. U. Density-Functional Theory for Time-Dependent Systems. *Phys. Rev. Lett.* **1984**, *52*, 997.

(4) Marques, M. A. L.; Gross, E. K. U. Time-Dependent Density-Functional Theory. *Annu. Rev. Phys. Chem.* **2004**, *55*, 427.

(5) Petersilka, M.; Gossmann, U. J.; Gross, E. K. U. Excitation Energies from Time-Dependent Density-Functional Theory. *Phys. Rev. Lett.* **1996**, *76*, 1212.

A Parallel Iterative Method

J. Chem. Theory Comput., Vol. 6, No. 9, 2010 **2667**

(6) Casida, M. E. Time-Dependent Density-Functional Response Theory for Molecules. In *Recent Advances in Density Functional Theory*, 1st ed.; Chong, D. P., Ed.; World Scientific: Singapore, 1995; p 155.

(7) Casida, M. E. Time-dependent density-functional theory for molecules and molecular solids. *THEOCHEM* **2009**, *914*, 3.

(8) van Gisbergen, S. J. A.; Fonseca Guerra, C.; Baerends, E. J. Towards Excitation Energies and (Hyper)polarizability Calculations of Large Molecules. Application of Parallelization and Linear Scaling Techniques to Time-Dependent Density Functional Response Theory. *J. Comput. Chem.* **2000**, *21*, 1511.

(9) de Boeij, P. L. Solution of the Linear-Response Equations in a Basis Set In *Time-Dependent Density Functional Theory*, 1st ed.; Marques, M. A. L., Ullrich, C. A., Nogueira, F., Rubio, A., Burke, K., Gross, E. K. U., Eds.; Springer: Berlin, Germany, 2008; pp 211−215.

(10) Jensen, J.; Autschbach, J.; Schatz, G. C. Finite lifetime effects on the polarizability within time-dependent density-functional theory. *J. Chem. Phys.* **2005**, *122*, 224115. Seth, M.; Ziegler, T.; Banerjee, A.; Autschbach, J.; van Gisbergen, S. J. A.; Baerends, E. J. Calculation of the *A* term of magnetic circular dichroism based on time dependent-density functional theory I. Formulation and implementation. *J. Chem. Phys.* **2004**, *120*, 10942.

(11) Rocca, D.; Gebauer, R.; Saad, Y.; Baroni, S. Turbo charging time-dependent density-functional theory with Lanczos chains. Theoretical and experimental results for C_60 and chlorophyll-a. *J. Chem. Phys.* **2008**, *128*, 154105.

(12) Rocca, D. Time-dependent density functional perturbation theory. Ph.D. Thesis, Scuola Internazionale Superiore di Studi Advanzati, Trieste, Italy, 2004.

(13) Foerster, D. Elimination, in electronic structure calculations, of redundant orbital products. *J. Chem. Phys.* **2008**, *128*, 034108.

(14) Foerster, D.; Koval, P. On the Kohn-Sham density response in a localized basis set. *J. Chem. Phys.* **2009**, *131*, 044103.

(15) Koval, P.; Foerster, D.; Coulaud, O. Fast construction of the Kohn-Sham response function for molecules. *Phys. Status Solidi B* **2010**, *247*, 1841.

(16) Fetter, A. L.; Walecka, J. D. *Quantum Theory of Many-Particle Systems*, 1st ed.; McGraw-Hill: New York, 1971; p 214.

(17) Beebe, N. H. F.; Linderberg, J. Simplifications in the generation and transformation of two-electron integrals in molecular calculations. *Int. J. Quantum Chem.* **1977**, *7*, 683.

(18) Boys, S. F.; Shavitt, I. A. Fundamental Calculation of the Energy Surface for the System of Three Hydrogen Atoms. Technical Report WIS-AF-13; University of Wisconsin Naval Research Laboratory: Madison, WI, 1959.

(19) Skylaris, C.-K.; Gagliardi, L.; Handy, N. C.; Ioannou, A. G.; Spencer, S.; Willetts, A. On the resolution of identity Coulomb energy approximation in density functional theory. *THEOCHEM* **2000**, *501−502*, 229.

(20) Baerends, E. J.; Ellis, D. E.; Ros, P. Self-consistent molecular Hartree-Fock-Slater calculations I. The computational procedure. *Chem. Phys.* **1973**, *2*, 41.

(21) Te Velde, G.; Bickelhaupt, F. M.; Baerends, E. J.; Fonseca Guerra, C.; van Gisbergen, S. J. A.; Snijders, J. G.; Ziegler, T. Chemistry with ADF. *J. Comput. Chem.* **2001**, *22*, 931.

(22) Vysotskiy, V. P.; Cederbaum, L. S. On the Cholesky Decomposition for electron propagator methods: General aspects and application on C60. 2009, arXiv: physics/0912.1459. arXiv.org ePrint archive. http://arxiv.org/abs/0912.1459 (accessed Jun 29, 2010).

(23) Aryasetiawan, F.; Gunnarsson, O. Product-basis method for calculating dielectric matrices. *Phys. Rev. B* **1994**, *49*, 16214.

(24) Larrue, U. *Etude de la Densité spectrale dune métrique associée a l'equation de Schrödinger pour l'hydrogene.* Unpublished, Bordeaux, 2008. This study showed an asymptotically uniform density of eigenvalues, on a logarithmic scale, for the metric in the case of the hydrogen atom.

(25) Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed; SIAM: Philadelphia, PA: 2000; p 158.

(26) Fraysse, V.; Giraud, L.; Gratton, S.; Langou, J. Algorithm 842: A set of GMRES routines for real and complex arithmetics on high performance computers. *ACM Trans. Math. Softw.* **2005**, *31*, 228. http://doi.acm.org/10.1145/1067967.1067970 (accessed Jun 29, 2010).

(27) Ipsen, I. C. F.; Meyer, C. D. The Idea behind Krylov Methods. *Am. Math. Monthly* **1998**, *105*, 889.

(28) Barrett, R.; Berry, M. W.; Chan, T. F.; Demmel, J.; Donato, J.; Dongarra, J.; Eijkhout, V.; Pozo, R.; Romine, Ch.; van der Vorst, H. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed; SIAM: Philadelphia, PA, 1993; p 51.

(29) Talman, J. D. LSFBTR: a subroutine for calculating spherical Bessel transforms. *Comput. Phys. Commun.* **1983**, *30*, 93. Talman, J. D. NumSBT: A subroutine for calculating spherical Bessel transforms numerically. *Comput. Phys. Commun.* **2009**, *180*, 332.

(30) Gradsteyn, I. S.; Ryshik, I. M. *Tables of Integrals, Series and Products*, 5th ed; Academic Press: New York, 1980; formula 6.578/4.

(31) Lebedev, V. I. *Russ. Acad. Sci. Dokl. Math.* **1995**, *50*, 283. http://www.ccl.net/cca/software/SOURCES/FORTRAN/Lebedev-Laikov-Grids/ (accessed Jun 30, 2010).

(32) The OpenMP API specification for parallel programming. http://openmp.org/wp/ (accessed Jun 30, 2010).

(33) Brabec, Ch. J.; Sariciftci, N. S.; Hummelen, J. C. Plastic Solar Cells. *Adv. Funct. Mater.* **2001**, *11*, 15.

(34) Bauernschmit, R.; Ahlrichs, R.; Hennrich, F. H.; Kappes, M. M. Experiment versus Time Dependent Density Functional Theory Prediction of Fullerene Electronic Absorption. *J. Am. Chem. Soc.* **1998**, *120*, 5052.

(35) Ordejón, P.; Artacho, E.; Soler, J. M. Self-consistent order-*N* density-functional calculations for very large systems. *Phys. Rev. B.* **1996**, *53*, R10441. Soler, J. M.; Artacho, E.; Gale, J. D.; García, A.; Junquera, J.; Ordejón, P.; Sánchez-Portal, D. The SIESTA method for ab initio order-N materials simulation. *J. Phys.: Condens. Matter* **2002**, *14*, 2745. http://www.icmab.es/siesta/ (accessed Jun 30, 2010). We used different branches of SIESTA version 3 to perform calculations in this work.

(36) Perdew, J. P.; Zunger, A. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B* **1981**, *23*, 5048.

(37) Liang, F.; Lu, J.; Ding, J.; Movileanu, R.; Tao, Ye. Design and Synthesis of Alternating Regioregular Oligothiophenes/Benzothiadiazole Copolymers for Organic Solar Cells. *Macromolecules.* **2009**, *42*, 6107.

2668 *J. Chem. Theory Comput., Vol. 6, No. 9, 2010*

Koval et al.

(38) Gao, Y.; Martin, Th. P.; Thomas, A. K.; Grey, J. K. Resonance Raman Spectroscopic- and Photocurrent Imaging of Polythiophene/Fullerene Solar Cells. *J. Phys. Chem. Lett.* **2010**, *1*, 178.

(39) Sundholm, D. Density functional theory calculations of the visible spectrum of chlorophyll a. *Chem. Phys. Lett.* **1999**, *302*, 480. Optimized geometry has been taken from supplementary at http://www.chem.helsinki.fi/~sundholm/qc/chlorophylla/ (accessed Jun 30, 2010).

(40) Du, H.; Fuh, R. C. A.; Li, J.; Corkan, L. A.; Lindsey, J. S. PhotochemCAD: A Computer-Aided Design and Research Tool in Photochemistry. *Photochem. Photobiol.* **1998**, *68*, 141. Database is available online at http://omlc.ogi.edu/spectra/PhotochemCAD/html/index.html (accessed Jun 30, 2010).

(41) Mayer, A. C.; Scully, S. R.; Hardin, B. E.; Rowell, M. W.; McGehee, M. D. Polymer-based solar cells. *Mater. Today* **2007**, *10*, 28.

(42) Suresh, P.; Balraju, P.; Sharma, G. D.; Mikroyannidis, J. A.; Stylianakis, M. M. Effect of the Incorporation of a Low-Band-Gap Small Molecule in a Conjugated Vinylene Copolymer: PCBM Blend for Organic, Photovoltaic Devices. *ACS Appl. Mater. Interfaces* **2009**, *1*, 1370.

(43) Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325. Greengard, L. Fast Algorithms for Classical Physics. *Science.* **1994**, *265*, 909.

(44) Hedin, L.; Lundqvist, S. Effects of Electron-Electron and Electron-Phonon Interactions on the One-Electron States of Solids. In *Solid State Physics*, 1st ed; Academic Press: London, Great Britain, 1969; Vol. 23, pp 1−181.

(45) Heyd, J.; Scuseria, G. E.; Ernzerhof, M. Hybrid functionals based on a screened Coulomb potential. *J. Chem. Phys.* **2003**, *118*, 8207.