An Improvement of the Resolution of the Identity Approximation for the Formation of the Coulomb Matrix

FRANK NEESE
Max Planck Institut für Strahlenchemie, Stiftstr. 34-36,
D-45470, Mülheim an der Ruhr, Germany

Received 18 February 2003; Accepted 5 May 2003

Abstract: A straightforward modification of the resolution of the identity (RI) approximation to the Coulomb interaction is described. In the limit of basis sets that are dominated by high angular momentum functions the observed speedups in realistic test systems reach a factor of 2 compared to the standard RI algorithm, and a factor of up to 300 compared to the standard algorithm to form the Coulomb matrix. More moderate savings on the order of 0–20% are obtained for the more commonly used smaller basis sets. A series of test calculations is reported to illustrate the efficiency of the algorithm.

 J Comput Chem 24: 1740–1747, 2003

Key words: density functional theory; *ab initio* electronic structure theory; Coulomb interaction; RI approximation; Gaussian basis functions

Introduction

In recent years electronic structure methods are applied to ever larger molecules. In this context linear scaling methods based on density functional theory (DFT) are of particular interest, as they promise to give accurate results for large molecules with computation times that are directly proportional to the system size.[1] Through extensive efforts linear scaling has been achieved for essentially all major computational steps in a DFT calculation. The numerical exchange-correlation quadrature has long been recognized as being intrinsically linear scaling due to the fast decaying nature of the basis functions used,[2,3] and linear scaling quadrature grid construction is also feasible.[4] If hybrid functionals are used, the Hartree–Fock (orbital) exchange is also essentially linear scaling.[5–8] Of the remaining computationally significant steps the matrix diagonalization can be avoided by conjugate gradient[9–12] or quasi Newton approaches,[13–15] which could also be implemented in essentially linear scaling versions by using sparse matrix linear algebra. For the Coulomb interaction, linear scaling has been achieved by the use of the fast-multipole method.[16–18] In this method the computation of the Coulomb interaction is divided into a near-field and a far-field part, the latter of which is treated by accurate multipole approximations and the former by exact analytical integration. However, this analytical integration is still an expensive part of the calculation, and efficient algorithms are required to achieve high computational throughput. Almlöf et al. were the first to recognize that it is beneficial to split up the calculation of the two-electron interaction into the Coulomb and exchange part, and to develop the most efficient algorithm for each individual contribution.[19] Therefore, Ahmadi and Almlöf have developed an algorithm where the Coulomb matrix is formed in a Gaussian product basis with substantial benefits in the floating point operation (FLOP) cost.[20] A similar method has been used by Schwegler and Challombe in their linear scaling Fock matrix construction algorithms.[5,6,21–23] Head–Gordon et al. have developed a similar algorithm which they termed "J-matrix engine."[24] This algorithm has recently been improved[25,26] by using a scheme that was acknowledged to be similar to that of Ahmadi and Almlöf but with extensive use of recurrence relations to speed up the time-critical step of the computation.[25] The reported speedups relative to the standard algorithm to compute the Coulomb matrix were up to a factor of 10 if extensively polarized basis sets were used.[25]

If one gives up on the idea to compute the exact Coulomb interaction, the resolution of the identity (RI) approximation has been found to give an excellent approximation to the Coulomb interaction at greatly reduced computational cost albeit still with approximately quadratic scaling. The theory of this method has been thoroughly discussed by Whitten[27] and by Dunlap et al.,[28] and is implemented in a number of prominent density functional codes. An important contribution has been made by Vahtras et al.,[29] who have shown that the Coulomb metric gives an one order

Correspondence to: F. Neese; e-mail: neese@mpi-muelheim.mpg.de

Contract/grant sponsor: Max-Planck Gesellschaft and the Fonds der Chemischen Industrie

of magnitude more accurate approximation than other choices for the determination of the fit coefficients. The important subject of auxiliary basis sets has been systematically and efficiently treated by Ahlrichs et al.[30,31] Due to their work, accurate auxiliary basis sets are available that are consistent with polarized split-valence (SVP),[32] and also more accurate polarized triple-zeta valence (TZVP) orbital basis sets[33] for most of the periodic table. In addition, auxiliary basis sets have been developed for use in MP2 calculations[34–36] where the RI approximation also leads to large speedups.[34–38] The charge density fitting procedure can also be implemented in $O(N)$ versions by employing a divide-and-conquer approach.[39–42] Most recently, Ahlrichs et al. have combined the RI approximation with multipole expansions to reach a linear scaling algorithm.[43]

Other original and efficient methods for the calculation of the Coulomb interaction are available[44–53] but are less directly related to the work presented in this article.

The purpose of this article is to adapt the ideas of Ahmadi and Almlöf to the calculation of the Coulomb matrix in the RI approximation. Overall speedups of up to a factor of ~5 compared to the standard way of evaluating the Coulomb interaction in the exact treatment and up to ~300 with the use of the RI approximation were achieved. The corresponding algorithm was termed "Split-RI-J," and has been incorporated into our general purpose quantum chemical *ab initio* and DFT package ORCA.[54]

Theory

The standard of way of computing the Coulomb matrix in a standard linear combination of atomic orbitals (LCAO) approach is:

$$J_{\mu\nu} = \sum_{\kappa\tau} P_{\kappa\tau}\langle\mu\nu|\kappa\tau\rangle \tag{1}$$

Here, **P** is the one-particle density matrix and

$$\langle\mu\nu|\kappa\tau\rangle = \int\int \frac{\varphi_\mu(\mathbf{r}_1)\varphi_\nu(\mathbf{r}_1)\varphi_\kappa(\mathbf{r}_2)\varphi_\tau(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}\, d\mathbf{r}_1 d\mathbf{r}_2 \tag{2}$$

is an electron–electron repulsion integral (ERI) over real basis functions $\{\varphi\}$. Our integral evaluation algorithms are based on the McMurchie–Davidson (MD) method[55] in which the basis function products that appear in Eq. (2) are expanded in a linear combination of Hermite Gaussian functions. In an abbreviated notation this expansion can be written:[55,56]

$$\varphi_\mu^A(\mathbf{r})\varphi_\nu^B(\mathbf{r}) = \sum_p E_p^{\mu\nu}\Lambda_p^P(\mathbf{r}) \tag{3}$$

$$\Lambda_p^P(\mathbf{r}) = \left(\frac{\partial}{\partial P_x}\right)^{p_x}\left(\frac{\partial}{\partial P_y}\right)^{p_y}\left(\frac{\partial}{\partial P_z}\right)^{p_z}\exp(-(\alpha+\beta)(\mathbf{r}-\mathbf{P})^2) \tag{4}$$

here the Es are the (scaled) expansion coefficients, the Λs are the Hermite Gaussians, α and β are the exponents of the primitive Gaussians, and the subscript p has been used as a compound index. The basis functions are attached to centers A and B, respectively, while the Hermite Gaussians are centered at a point **P** along the line joining centers A and B:

$$\mathbf{P} = \frac{1}{\alpha+\beta}(\alpha\mathbf{R}_A + \beta\mathbf{R}_B) \tag{5}$$

Excellent publications that describe the details of this approach to ERI evaluation are available.[55,56] The ERI becomes:

$$\langle\mu\nu|\kappa\tau\rangle = \sum_p E_p^{\mu\nu}\sum_q (-1)^q E_q^{\kappa\tau}\mathbf{R}_{pq} \tag{6}$$

where R_{pq} is the ERI over Hermite Gaussian functions and the contraction of R_{pq} with the Es can be regarded as a Hermite to Cartesian Gaussian back transformation. Constant factors like products of contraction coefficients are assumed to be absorbed in the definition of the Es and the Rs. Both the Es and the Rs are easily generated from recursion relations.[55,56] The Coulomb matrix is therefore evaluated in this approach as (referred to as the "standard" approach; see Scheme 1):

$$J_{\mu\nu} = \sum_{\kappa\tau} P_{\kappa\tau}\sum_p E_p^{\mu\nu}\sum_q (-1)^q E_q^{\kappa\tau} R_{pq} \tag{7}$$

Our integral package does not use Eqs. (2)–(6) directly, but achieves substantial gains in efficiency by using hand-optimized routines for low angular momentum cases and the transfer equation for higher angular momentum integral batches. Similar design strategies have been used in other programs as well.[57,58] The same strategy has also been followed in coding the improved algorithms described below, but the general notation is preferred for simplicity. For the same reason, explicit reference to integral contraction and transformation to spherical harmonic Gaussian functions steps is also suppressed, although they are executed in the actual code.

Because the MD algorithm naturally decouples the coordinates of the two electrons and the density matrix does not connect the two electrons in the Coulomb matrix construction as well, it becomes evident that the standard approach is inefficient and

```
J(μ,ν)=0 (all μ,ν)
Loop μν
  Calculate E(μ,ν,p) (all p∈μν)
  Loop κτ≤μν
    Skip if max(|P(μ,ν)|,|P(κ,τ)|)*K(μ,ν)*K(κ,τ)<Thresh
    Calculate E(κ,τ,q) (all q∈κτ)
    Calculate R(p,q)   (all p∈μν and all q∈κτ)
    Calculate <μν|κτ>
    J(μ,ν)=J(μ,ν)+P(κ,τ)<μν|κτ>
    J(κ,τ)=J(κ,τ)+P(μ,ν)<μν|κτ>
  end
end
```

Scheme 1. Loop structure of the standard algorithm to form the Coulomb matrix. μ, ν, κ, τ are shell indices, and K is the prescreening matrix that contains the values $K(\mu, \nu) = \text{sqrt}(\langle\mu\nu|\mu\nu\rangle)$.The loops $\mu\nu$ and $\kappa\tau$ are executed over significant and unique shell pairs. Thresh is the integral neglect threshold.

```
J(μ,ν)=0 (all μ,ν)
Loop κτ
  Calculate E(κ,τ,q)   (all q∈κτ)
  Calculate X(q)=P(κ,τ)*E(κ,τ,q)*(-1)^q (all q∈κτ)
end
Loop μν
   Y(p)=0 (all p∈μν)
   Loop κτ
     Skip if max(|P(μ,ν)|,|P(κ,τ)|)*K(μ,ν)*K(κ,τ)<Thresh
     Calculate R(p,q)   (all p∈μν and all q∈κτ)
     Y(p)=Y(p)+R(p,q)*X(q)  (all q∈κτ) and all p∈μν)
   end
   Calculate E(μ,ν,p)       (all p∈μν)
   J(μ,ν)=J(μ,ν)+E(μ,ν,p)*Y(p)  (all p∈μν)
end
```

Scheme 2. Loop structure of the "Improved" algorithm to form the Coulomb matrix. X and Y are intermediate scratch arrays.

involves a large number of redundant operations in the innermost loops. Consequently, it is beneficial to first form the density in the Hermite Gaussian basis:[20,25]

$$X_q = \sum_{\kappa\tau} (-1)^q P_{\kappa\tau} E_q^{\kappa\tau} \tag{8}$$

Because the number of significant basis function products only grows as $O(N)$, the computation of the vector **X** and its storage requirements are negligible. In the second step the vector **X** is contracted with the Hermite basis ERIs to yield the vector **Y**:

$$Y_p = \sum_q R_{pq} X_q \tag{9}$$

Finally, in the third step, the Coulomb matrix is formed by contraction of the Hermite Coulomb "matrix" **Y** with the Hermite expansion coefficients of the bra as:

$$J_{\mu\nu} = \sum_p E_p^{\mu\nu} Y_p \tag{10}$$

This algorithm is termed "improved algorithm" (see Scheme 2), and it is essentially the same as the methods proposed by Ahmadi and Almlöf,[20] Schwegler and Challacombe,[21,22] and Shao and Head–Gordon.[25] The main differences probably concern the generation of the integrals R_{pq} and their contraction with the Hermite density X_q in the rate-limiting step of the algorithm. In our implementation, an efficient evaluation is obtained by using hand-optimized expressions for most shell pair combinations (all shell pairs with $l_\mu + l_\nu + l_\kappa + l_\tau \leq 6$), and direct formation of the contraction of R_{pq} with X_q without any further intermediate quantities. This approach is devoid of short loops or logic that would spoil computational efficiency, allows for extensive common subexpression elimination, and uses no additional central memory. For higher angular momenta the computational overhead due to nested loops is not a major factor, and the general recursion relations to form R_{pq} perform well. Compared to the standard algorithm, a substantial amount of overhead, namely the generation of the E coefficients and the contraction with the Hermite integrals, has been removed from the innermost loops. Although not explicitly shown, the transformation to spherical harmonic Gaussian functions is also done outside the innermost loops in contrast to the standard algorithm, which saves further time. Consequently, significant speedups are to be expected.

The RI Algorithm

The RI approximation to the Coulomb interaction[27–31] involves the introduction of a fitting basis $\{\eta\}$. The Coulomb matrix becomes (see Scheme 3):

$$J_{\mu\nu} = \sum_r d_r \langle \mu\nu | r \rangle \tag{11}$$

where the fit coefficients d are the solutions to the linear equation system:

$$\mathbf{Vd} = \mathbf{g} \tag{12}$$

The positive definite matrix **V** contains the ERIs over auxiliary basis functions:

$$V_{rs} = \langle r|s \rangle = \int\int \frac{\eta_r(\mathbf{r}_1)\eta_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 \tag{13}$$

and the vector **g** is the contraction of the density matrix with the three-center ERIs:

$$g_r = \sum_{\mu\nu} P_{\mu\nu} \langle \mu\nu | r \rangle \tag{14}$$

where:

```
J(μ,ν)=0 (all μ,ν)
Loop μν
  Calculate E(μ,ν,q)   (all q∈μν)
  Loop r
     Skip if |P(μ,ν)|*K(μ,ν)*sqrt(V(r,r))<Thresh
     Calculate E(r,p)   (all p∈r)
     Calcuate R(p,q)    (all p∈r and all q∈μν)
     Calculate <μν|r>
     g(r)= <μν|r>*P(μ,ν)
   end
end
Solve d=V^-1*g through the Cholesky decomposition of V
Loop μν
  Calculate E(μ,ν,q)   (all q∈ μν)
  Loop r
     Skip if |d(r)|*K(μ,ν)*sqrt(V(r,r))<Thresh
     Calculate E(r,p)   (all p∈r)
     Calcuate R(p,q)    (all p∈r and all q∈ μν)
     Calculate <μν|r>
     J(μ,ν)=J(μ,ν)+<μν|r>*d(r)
   end
end
```

Scheme 3. Loop structure of the standard RI algorithm for the formation of the Coulomb matrix. The index r loops over all auxiliary basis functions. V is the two index matrix of electron–electron repulsion integrals over auxiliary basis functions.

$$\langle \mu\nu|r\rangle = \int\int \frac{\varphi_\mu(\mathbf{r}_1)\varphi_\nu(\mathbf{r}_1)\eta_r(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}\, d\mathbf{r}_1 d\mathbf{r}_2 \tag{15}$$

Following the suggestion by Eichkorn et al.,[31] the linear equation set Eq. (12) is solved through the Choleksy decomposition of the matrix **V**, which is very efficient and numerically stable as long as **V** is properly positive definite. With this approximation and appropriate auxiliary basis sets[30,31] the Coulomb energy is typically reproduced to within a few millihartree, and the effect on optimized structure parameters is negligible. Furthermore, for a given density the RI-Coulomb energy is a lower bound to the exact Coulomb energy,[28] and therefore, the errors are highly systematic and tend to cancel to a large degree if energy differences are computed.

An Improved RI Algorithm

The dominating computational step in the RI approximation is the computation of the three center ERIs [Eq. (15)], which have to be computed twice during a direct SCF iteration (see Scheme 3). First, to calculate the vector **g** [Eq. (14)], and second, to complete the Coulomb matrix [Eq. (11)]. The three center ERIs can be written in the spirit of the MD algorithm as:

$$\langle r|\mu\nu\rangle = \sum_p E_p^r \sum_q (-1)^q E_p^{\mu\nu} R_{pq} \tag{16}$$

where the auxiliary basis function η_r has also been expanded in terms of Hermite Gaussians. Formally, Eq. (16) corresponds to a four center ERI in which the fourth (unnormalized) primitive Gaussian has an exponent of zero. Again, it is clear that the computation of the vector **g** and the matrix **J** with such an algorithm must be inefficient due to the unnecessary Hermite Gaussian to Cartesian Gaussian transformation in the innermost loops. The RI algorithm can, therefore, be rewritten as (see Scheme 4):

$$X_q = \sum_{\kappa\tau} P_{\kappa\tau} E_p^{\kappa\tau} (-1)^q \tag{17}$$

$$Y_p = \sum_q R_{pq} X_q \tag{18}$$

$$g_r = \sum_p E_p^r Y_p \tag{19}$$

After formation of the vector **g**, the linear equation is solved using the prestored Choleksy decomposition of **V**, and the Coulomb matrix is completed by the following three steps:

$$T_p = \sum_r E_p^r d_r \tag{20}$$

$$U_q = \sum_p R_{pq} T_p \tag{21}$$

```
J(μ,ν)=0 (all μ,ν)
Loop κτ
  Calculate E(κ,τ,q)   (all q∈κτ)
  Calculate X(q)= P(κ,τ)*E(κ,τ,q)(-1)^q (all q∈κτ)
end
Loop r
  Y(p)=0  (all p∈r)
  Loop κτ
    Skip if |P(κ,τ)|*K(κ,τ)*sqrt(V(r,r))<Thresh
    Calculate R(p,q)   (all p∈r and all q∈μν)
    Y(p)=Y(p)+R(p,q)*X(q)  (all p∈r and all q∈μν)
  end
  Calculate E(r,p)      (all p∈r)
  g(r) = E(r,p)*Y(p)  (all p∈r)
end
Solve d=V^-1*g through the Cholesky decomposition of V
Loop r
  Calculate E(r,p)              (all p∈r)
  Calculate T(p)= d(r)*E(r,p)   (all p∈r)
end
Loop μν
  U(q)=0  (all q∈μν)
  Loop r
    Skip if |d(r)|*K(μ,ν)*sqrt(V(r,r))<Thresh
    Calculate R(p,q)  (all p∈r and all q∈μν)
    U(q)=U(q)+R(p,q)*T(p)  (all p∈r and all q∈μν)
  end
  Calculate E(μ,ν,q)          (all q∈μν)
  J(μ,ν)=J(μ,ν)+E(μ,ν,q)*U(q)     (all q∈μν)
end
```

Scheme 4. Loop structure of the Split-RI-J algorithm for the formation of the Coulomb matrix.

$$J_{\mu\nu} = \sum_q (-1)^q E_q^{\mu\nu} U_q \tag{22}$$

All computational steps are straightforward, and short loops, which would significantly add to the overhead, can, again, be completely avoided. We refer to this algorithm as "Split-RI-J," because the computation of the Coulomb matrix has been split into several parts. The most expensive steps are the contractions of the Hermite ERIs with the vectors **X** and **T** in Eqs. (18) and (21), which will tend to $O(N^2)$ for large systems. However, the prefactor for the evaluation of the Split-RI-J approximation to the Coulomb matrix is much smaller than for the exact evaluation, as will become evident in the next section.

In conclusion, some straightforward manipulations allow moving the expensive steps of the ERI evaluation in the Coulomb matrix and RI-approximation outside the innermost loops. The intermediate quantities no longer contain the ERIs themselves but, instead, the density and electron repulsion operator in the intermediate Hermite basis. The back transformation to the orbital basis and the transformation of the density into the Hermite basis can then be executed outside the rate-determining loops of the program, and high efficiency can be achieved. In Schemes 1–4 conventional prescreening of negligible contributions has been used.[59] Even more efficient schemes based on the known intermediate quantities in the Hermite Gaussian basis are conceivable but have not been explored in this work.

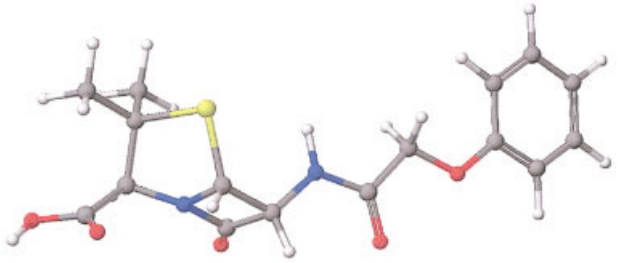

Figure 1. The structure of the Penicillin molecule used for benchmark calculations.

Calculations

Comparison of Algorithms

The "improved algorithm" has been thoroughly tested by Shao and Head–Gordon,[25] and we have observed very similar speedups in our implementation. It is therefore only necessary to asses the performance gain due to the improved RI algorithm. To this end calculations were carried out on the Penicillin molecule (Fig. 1) with basis sets of increasing size. Penicillin is an essentially arbitrarily chosen member of a class of medium sized organic molecules (42 atoms, 250–1000 basis functions) that are too small for linear scaling approaches to be effective and large enough for the calculations to become reasonably time consuming. We prefer this approach over purely theoretical FLOP counting because the required wall clock time of a given algorithm is the most interesting quantity to an user of a quantum chemical program package. Algorithms with a favorable FLOP cost do not necessarily lead to equally favorable wall clock times.

A comparison of the standard RI-J approximation with the Split-RI-J algorithm is shown in Figure 2. Except for the smallest basis set (SV), the Split-RI-J algorithm is consistently faster than the standard algorithm, although the speedups are not spectacular.

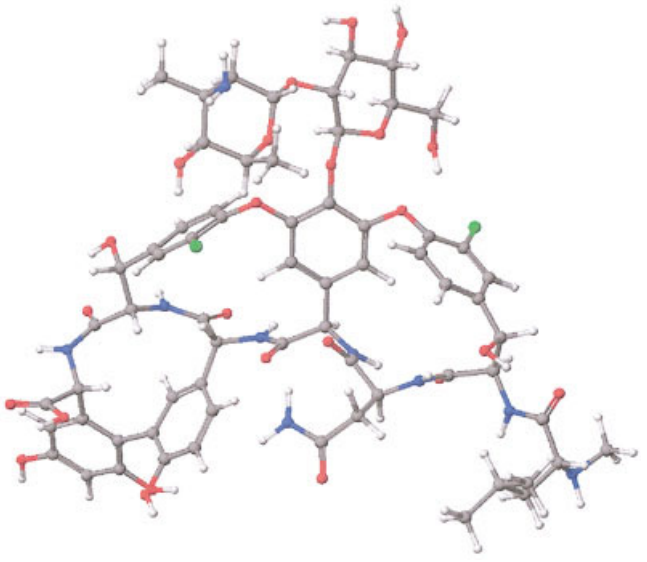

Figure 3. The structure of the Vancomycin molecule used in the benchmark calculations.

This is probably due to the fact that the computational effort for low angular momentum basis functions is dominated by the work that is common to both algorithms, that is, the evaluation of the incomplete gamma functions and the R_{pq}-coefficients. However, for large basis sets with significant numbers of f-functions a speedup of a factor of ~2 is observed compared to the standard RI algorithm. Compared to the standard algorithm the Split-RI-J method is a factor of ~300 faster for the largest basis set TZV(2df,2pd). For the less extensive TZV(d,p) basis set the standard RI and the Split-RI-J algorithms are almost identical and give a speedup of a factor of ~30 over the exact algorithm. Because the error introduced by the RI approximation is very small if the auxiliary basis sets developed by Eichkorn et al.[30,31] are used,

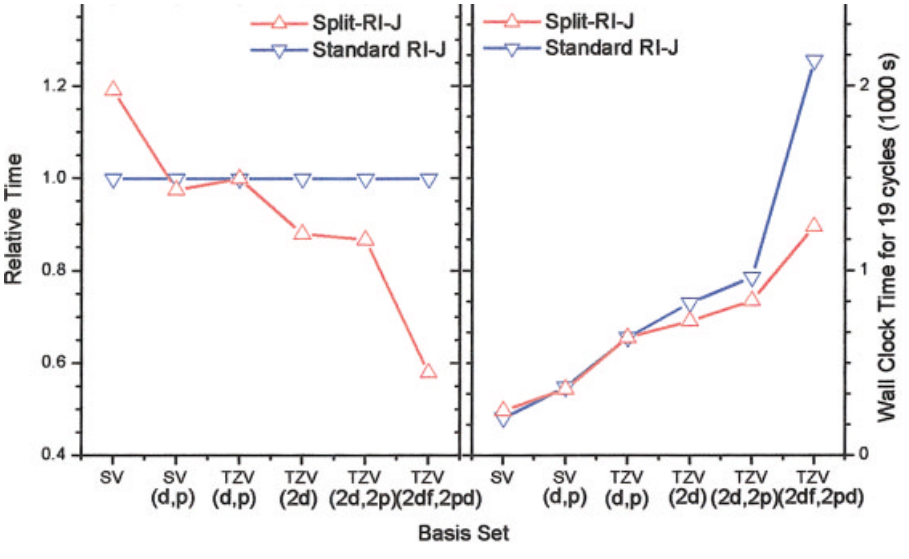

Figure 2. The relative and absolute performance of the two RI-based algorithms to build the Coulomb matrix for the penicillin molecule as a test example. The quoted times refer to the formation of the Coulomb contribution to the Kohn–Sham matrix only.

Table 1. Comparison of the Split-RI-J and Standard RI-J Approaches for Series of Medium-Sized Pharmacologically Relevant Molecules.

	N_{Atoms}	N_{BF}[TZV(2*df*, 2*pd*)]	$T_{\text{Split-RI-J}}/T_{\text{RI-J}}$	N_{BF}[SV(*d*)]	$T_{\text{Split-RI-J}}/T_{\text{RI-J}}$
Lisinopril	60	1333	0.64	468	0.96
Tetracycline	56	1328	0.70	496	0.87
Diazepam	33	805	0.61	310	0.86
Nitroglycerine	17	442	0.77	178	0.90
Bromocriptine	83	1910	0.72	700	0.95
Cocaine	43	976	0.65	350	0.86
Estradiol	44	956	0.69	328	0.87
Fluconazole	34	850	0.65	332	0.75

N_{BF}[basis]—Number of basis functions in the indicated basis set for the given molecule.
N_{Atoms}—Number of atoms.
$T_{\text{Split-RI-J}}/T_{\text{RI-J}}$—Ratio of wall clock times spend for the construction of the Coulomb matrix in all SCF cycles for the Split-RI-J and standard RI-J algorithms, respectively.

there appears to be little reason to avoid the RI approximation. The crossover of the RI approximation with linear scaling Coulomb algorithms has been estimated by Ahlrichs to occur in the range of ~2000 basis functions.[60]

To put these results on a somewhat more sound basis, additional calculations were carried out for a series of eight pharmacologically relevant molecules with the large TZV(2df,2pd) and small SV(d) basis sets. The results of these calculations are summarized in Table 1, and show essentially the same trends already observed for penicillin. Thus, for the calculations with the large basis set, the Split-RI-J algorithm is significantly more efficient than the standard RI-J algorithm. The achievable savings are system dependent and amount to factors of ~1.25–1.70. The efficiency gain for the small SV(d) basis set is less pronounced but still real and factors of 1.05–1.25 are observed.

A Large Calculation

A relatively large calculation was carried out to test the applicability of the Split-RI-J algorithm to large systems and also to document the overall performance of our program. The test molecule is the antibiotic Vancomycin (C_1 symmetry, Figure 3), which has 176 atoms and with the SV(d,p) basis is described by 1796 basis functions. The auxiliary basis set chosen was the smaller "DeMon Coulomb fitting set,"[61] which leads to 3676 auxiliary basis functions. The average amount of time taken in each SCF iteration for a local density calculation for each of the computationally significant steps is (Pentium P4 CPU, 1.7 GHz; peak memory allocation ~140 MB): Split-RI-J: 421 sec, exchange-correlation quadrature: 84 s, Fock matrix diagonalization: 206 s, linear algebra: 191 s, density matrix formation: 4 s. Thus, each SCF iteration takes ~15 min of computer time on one CPU of a low-cost personal computer, which means that a complete SCF for a molecule such as Vancomycin is obtained in a few hours. This result shows that at a system size of less than 2000 basis functions the matrix diagonalization and linear algebra (transformation of the Fock matrix to the MO basis and calculation of the commutator required for the DIIS procedure) operations already start to dominate due to their unfavorable $O(N^3)$ scaling. Thus, it appears advisable to optimize, avoid and/or parallelize these steps before a linearization of the Coulomb matrix formation appears to be necessary.

Computational Details

All calculations were performed with the electronic structure package ORCA, which is entirely coded in C++.[54] The basis sets used were based on the split-valence (SV) basis set and the triple-zeta valence (TZV) bases of Ahlrichs et al.[32,33] with increasing numbers of polarization functions, all taken from the TurboMole library.[62] The notation SV(*m,n*) implies the SV basis set, with *m* indicating the set of polarization functions added to all nonhydrogen atoms and *n* indicating the polarization functions added to the hydrogens. The number of basis functions used to describe the Penicillin molecule for each basis set is SV: 256, SV(d,p):430, TZV(d,p): 567, TZV(2d): 633, TZV(2d,2p): 741, TZV(2df,2pd): 999. The auxiliary basis was always chosen to be the one optimized by Eichkorn et al.[30,31]for the TZVP orbital basis (referred to as TZV/J). Even for the largest orbital basis the error introduced by the RI approximation does not exceed 3 millihartree, which is considered satisfactory. For convenience we have chosen the local density approximation and moderately sized integration Lebedev grids with a maximum of 194 angular points. The grid pruning is similar to that reported by Gill et al.,[63] and the radial integration is that proposed by Treutler and Ahlrichs.[64] These grids integrate the total electron density to an accuracy of $\sim 10^{-3}$ electrons. The integral neglect threshold has been set to 10^{-10} Hartree for all calculations, and they were converged to 10^{-5} in the total density, 10^{-6} Hartree in the total energy, and 10^{-6} Hartree in the DIIS error. The geometry of the Penicillin molecule came from a molecular mechanics calculation. The Vancomycin molecule came from the fragment library of the CAChe molecular modeling system.[65] All calculations were performed on a 1.3 GHz Athlon CPU running under Redhat Linux 7.1. All reported times are wall clock times, and are cumulative totals for all Coulomb builds. The number of cycles required to reach convergence was always 19 was the same for all algorithms and basis sets.

The calculations in Table 1 were performed on 2.4 GHz Xeon CPUs. The structures of the molecules came from the fragment library of the CAChe system.[65] The auxiliary basis sets were chosen to match the orbital basis in these calculations.

Conclusion

An improved RI algorithm to compute the near-field part of the Coulomb interaction in LCAO calculations with Gaussian basis functions was described. By design, it is most suitable for basis sets that have many high angular momentum functions (i.e., that are extensively polarized) and that are loosely contracted because no advantage is taken of contraction. If the expensive Hermite to Cartesian Gaussian transformation is moved outside the innermost loop, speedups of up to a factor of roughly 300 relative to the standard direct SCF algorithm were observed in demonstrative applications and a factor of up to ~2 compared to the standard RI-J approximation. However, this speedup is only obtained for basis sets that are extensively polarized. For basis sets that are dominated by s- and p-functions there is little to gain by the manipulations described in this article. Finally, we have proven that with these algorithms and our program ORCA large DFT calculations with more than 1700 basis functions and no symmetry on small limited memory personal computers can be carried out in a few hours of CPU time.

Acknowledgments

I am grateful to Prof. Ahlrichs and his group for stimulating discussions during a visit to Karlsruhe, and for allowing me to adapt their matrix diagonalization routine for my purposes. An anonymous referee is thanked for constructive criticism.

References

1. Scuseria, G. E. J Phys Chem A 1999, 103, 4782.
2. van Wüllen, C. Chem Phys Lett 1994, 219, 8.
3. Perez–Jorda, J. M.; Yang, W. Chem Phys Lett 1995, 241, 469.
4. Stratmann, R. E.; Scuseria, E.; Frisch, M. J. Chem Phys Lett 1996, 257, 213.
5. Schwegler, E.; Challacombe, M.; Head–Gordon, M. J Chem Phys 1997, 106, 9708.
6. Schwegler, E.; Challacombe, M. J Chem Phys 1999, 111, 6223.
7. Ochsenfeld, C. J Chem Phys 1998, 109, 1663.
8. Ochsenfeld, C. Chem Phys Lett 2000, 327, 216.
9. Termath, V. Int J Quantum Chem 1997, 61, 349.
10. Challacombe, M. J Chem Phys 1999, 110, 2332.
11. Daniels, A. D.; Scuseria, G. E. J Chem Phys 1999, 110, 1321.
12. Daniels, A. D.; Scuseria, G. E. Phys Chem Chem Phys 2000, 2, 2173.
13. Fischer, T. H.; Almlöf, J. J Phys Chem 1992, 95, 6607.
14. Chaban, G.; Schmidt, M. W.; Gordon, S. Theor Chem Acc 1997, 97, 88.
15. Neese, F. Chem Phys Lett 2000, 325, 93.
16. White, C. A.; Head–Gordon, M. J Chem Phys 1994, 101, 6593.
17. White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head–Gordon, M. Chem Phys Lett 1994, 230, 8.
18. White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head–Gordon, M. Chem Phys Lett 1996, 253, 268.
19. Almlöf, J. In Modern Electronic Structure Theory; Yarkony, D. R., Ed.; Worls Scientific: Singapore, 1995, p. 110, vol. 2.
20. Ahmadi, G. R.; Almlöf, J. Chem Phys Lett 1995, 246, 364.
21. Schwegler, E.; Challacombe, M. J Chem Phys 1997, 106, 5226.
22. Schwegler, E.; Challacombe, M. J Chem Phys 1998, 109, 8764.
23. Schwegler, E.; Challacombe, M.; Head–Gordon, M. J Chem Phys 1998, 109, 8764.
24. White, C. A.; Head–Gordon, M. J Chem Phys 1996, 104, 2620.
25. Shao, Y.; Head–Gordon, M. Chem Phys Lett 2000, 323, 415.
26. Shao, Y.; White, C. A.; Head–Gordon, M. J Chem Phys 2001, 114, 6572.
27. Whitten, J. L. J Chem Phys 1973, 58, 4496.
28. Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. J Chem Phys 1979, 71, 3396.
29. Vahtras, O.; Almlöf, J.; Feyereisen, M. W. Chem Phys Lett 1993, 213, 514.
30. Eichkorn, K.; Treutler, O.; Öm, H.; Häser, M.; Ahlrichs, R. Chem Phys Lett 1995, 240, 283.
31. Eichkorn, K.; Weigend, F.; Treutler, O.; Ahlrichs, R. Theor Chem Acc 1997, 97, 119.
32. Schäfer, A.; Horn, H.; Ahlrichs, R. J Chem Phys 1992, 97, 2571.
33. Schäfer, A.; Huber, C.; Ahlrichs, R. J Chem Phys 1994, 100, 5829.
34. Weigend, F.; Häser, M. Theor Chem Acc 1997, 97, 331.
35. Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. Chem Phys Lett 1998, 194, 143.
36. Weigend, F.; Köhn, A.; Hättig, C. J Chem Phys 2002, 116, 3175.
37. Feyereisen, M. W.; Fitzgerald, G.; Kormonicki, A. Chem Phys Lett 1993, 208, 359.
38. Bernhold, D. E.; Harrison, R. J. Chem Phys Lett 1996, 250, 477.
39. Goh, S. K.; St. Amant, A. Chem Phys Lett 1997, 264, 9.
40. Goh, S. K.; St. Amant, A. Chem Phys Lett 1997, 274, 429.
41. Goh, S. K.; Gallant, R. T.; St. Amant, A. Int J Quantum Chem 1998, 69, 405.
42. Gallant, R. T.; St. Amant, A. Chem Phys Lett 1996, 256, 569.
43. Sierka, M.; Hogekamp, A.; Ahlrichs, R. J Chem Phys 2003, 118, 9136.
44. Becke, A. D.; Dickson, R. M. J Chem Phys 1988, 89, 2993.
45. Delley, B.; Ellis, D. E. J Chem Phys 1982, 76, 1949.
46. Delley, B. J Chem Phys 1990, 92, 508.
47. Dombroski, J. P.; Taylor, S. W.; Gill, P. M. W. J Phys Chem 1996, 100, 7272.
48. Gill, P. M. W.; Adamson, R. D. Chem Phys Lett 1996, 261, 105.
49. Gill, P. M. W. Chem Phys Lett 1997, 170, 193.
50. Lee, A. M.; Taylor, S. W.; Dombroski, J. P.; Gill, P. M. W. Phys Rev A 1997, 55, 3233.
51. Manby, F. R.; Knowles, P. J.; Lloyd, A. W. J Chem Phys 2001, 115, 9144.
52. Thermath, V.; Handy, N. C. Chem Phys Lett 1994, 230, 17.
53. Friesner, R. A.; Murphy, R. B.; Rignalda, M. N. In Encyclopedia Computation Chemistry; Schleyer, P. v. R.; Allinger, N.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F., III; Schreiner, P. R., Eds.; Wiley: Chichester, 1998, p. 2290.
54. Neese, F. ORCA—An ab initio, Density Functional and Semiempirical Program Package, Version 2.2, revision 9, January 2002, Max Planck Institut für Strahlenchemie, Mülheim, Germany, 2002.
55. McMurchie, L. E.; Davidson, E. R. J Comp Phys 1978, 26, 218.
56. Helgaker, T.; Taylor, P. R. In Modern Electronic Structure Theory; Yarkony, D. R., Ed.; World Scientific: Singapore, 1995, p. 725.

57. Ahlrichs, R.; Bär, M.; Häser, M.; Horn, H.; Kölmel, C. Chem Phys Lett 1989, 162, 165.
58. Lindh, R. In Encyclopedia Computation Chemistry; Schleyer, P. v. R.; Allinger, N.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F., III; Schreiner, P. R., Eds.; Wiley: Chichester, 1998, p. 1337.
59. Häser, M.; Ahlrichs, R. J Comp Chem 1989, 10, 104.
60. Ahlrichs, R. In Encyclopedia Computation Chemistry; Schleyer, P. v. R.; Allinger, N.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F., III; Schreiner, P. R., Eds.; Wiley: Chichester, 1998, p. 3123.
61. Godbout, N.; Salahub, D. R.; Andzelm, J.; Wimmer, E. Can J Chem 1992, 70, 560.
62. Ahlrichs, R., et al. 2002.The TurboMole basis sets and auxiliary basis sets can be accessed via the World Wide Web http://www.chemie.uni-karlsruhe.de/PC/TheoChem/turbomole/index.html.
63. Gill, P. M. W.; Johnson, B. G.; Pople, J. A. Chem Phys Lett 1993, 220, 377.
64. Treutler, O.; Ahlrichs, R. J Chem Phys 1995, 102, 346.
65. CAChe—Computer Aided Chemistry—Version 3.2 ed. Oxford Molecular Ltd., 1999.