A Generalized Grid-Based Fast Multipole Method for Integrating Helmholtz Kernels

Pauli Parkkinen, Sergio A. Losilla, Eelis Solala, Elias A. Toivanen, Wen-Hua Xu[a], Dage Sundholm[*]

Department of Chemistry, P.O. Box 55 (A. I. Virtanens plats 1), FIN-00014 University of Helsinki, Finland   [a] Present address: Department of Chemistry, Northwest University, Xian, China

E-mail: Dage.Sundholm@helsinki.fi

Abstract

A grid-based fast multipole method (GB-FMM) for optimizing three-dimensional (3D) numerical molecular orbitals in the bubbles and cube *double* basis has been developed and implemented. The present GB-FMM method is a generalization of our recently published GB-FMM approach for numerically calculating electrostatic potentials and two-electron interaction energies. The orbital optimization is performed by integrating the Helmholtz kernel in the *double* basis. The steep part of the functions in the vicinity of the nuclei is represented by one-center bubbles functions, whereas the remaining cube part is expanded on an equidistant 3D grid. The integration of the bubbles part is treated by using one-center expansions of the Helmholtz kernel in spherical harmonics multiplied with modified spherical Bessel functions of the first and second kind, analogously to the numerical inward and outward integration approach for

---

[*]To whom correspondence should be addressed

calculating two-electron interaction potentials in atomic structure calculations. The expressions and algorithms for massively parallel calculations on general-purpose graphics processing units (GPGPU) are described. The accuracy and the correctness of the implementation has been checked by performing Hartree-Fock self-consistent-field calculations (HF-SCF) on H_2, H_2O and CO. Our calculations show that an accuracy of 10^{-4} to 10^{-7} E_h can be reached in HF-SCF calculations on general molecules.

1 Introduction

Molecular electronic structure calculations are usually performed by using global basis sets of some kind. The most popular alternative is Gaussian type orbital (GTO) basis sets, whereas Slater type orbitals (STO) and plane-wave (PW) expansions are also employed.[1–3] The slow convergence to the basis-set limit is a common denominator for the global basis-set approaches. Basis-set limit calculations can be performed by utilizing local numerical basis sets.[4–30] However, the large computational costs of real-space electronic structure calculations *limit* the general use of such approaches. The advent of massively parallel computers with a huge number of central processing units (CPU) or general-purpose graphics processing units (GPGPU) renders local basis-set approaches feasible.[31–42] Efficient algorithms can be developed because the algorithms of fully numerical approaches are relatively simple involving huge amount of similar data. Thus, it is possible to design computational methods with the aim to run the program in parallel on thousands or millions of processing units with very little communication between the processors.

The aim of this research project is to develop fully numerical electronic structure methods for calculations on general molecules using massively parallel computers. To reach that goal we use a *double* basis where the steep parts of the functions around the nuclei are expanded in one-center functions and the remainder is expanded on an equidistant three-dimensional (3D) grid. The computational time for the one-center functions is negligible as compared to the computational time for the 3D grid functions. To speed up the computations one has to

preliminary focus on the parallelization of the projection of the one-center functions and the treatment of the functions on the grid. We have recently developed a computational approach for calculating two-electron interaction energies and potentials in a massively parallel fashion that runs very efficiently on GPGPUs.[43] We demonstrated in that work that an N^0 scaling of the wall time can be achieved when many processors are available. Thus, the wall time for calculations of the two-electron interactions were independent of the number of grid points. We reached the limit of Amdahl's law, where the sequential part of the calculation determines the total computational time. In the parallel version of our code, calculating an electrostatic potential lasted at that time about two minutes, whereas the computational time using the sequential code took about an hour for the largest grid when one CPU was utilized. The keys to the success were the grid-based fast multipole method (GB-FMM) and the use of GPGPUs. GB-FMM avoids explicit calculations of the long-ranged contributions to the two-electron interactions. Instead, these contributions can be rapidly obtained by calculating multipole moments. The linear transformations of the our integration approach are matrix-multiplication driven in the innermost loops implying that they run very efficient on GPGPUs. The grid-based identification of the the one-center functions also run efficiently on GPGPUs, because they comprise lots of similar computations.

In this work, we extend the GB-FMM methods to the integration of the Helmholtz kernel,[44–49] which can be used for optimizing numerical orbitals.[9,29,50] The structure of the expressions are similar to those for the Poisson kernel *i.e.*, the Coulomb integral expression. The optimization of the orbitals by using the Helmholtz kernel is an iterative process as it involves orbital energies and a given set of guess orbitals *e.g.*, the ones of the previous iteration. The one-center part of the functions can be integrated analytically in the angular dimensions, whereas in the radial direction they are integrated numerically using inward and outward integration in combination with Bessel functions.[51]

In Section 2, we describe the numerical integration of the Helmholtz kernel in the *bubbles* and *cube* basis. The theory and implementation of the grid-based fast multipole method

to integrate the Helmholtz kernel and how the approach is used for updating orbitals are discussed in Section 3 with further details given in the Appendix. The Hartree-Fock energies and some timings for calculations on H_2, H_2O and CO are reported in Section 4.

2 The Helmholtz Kernel Expression

2.1 Function Representation

We employ a *double* numerical basis, where all functions $(\nu(x, y, z))$ are divided into a three-dimensional (3D) Cartesian *cube* part $(\nu^\Delta(x, y, z))$ and a part consisting of one-center functions at each nucleus A, called *bubbles*.[18] Bubbles functions are written as numerical one-dimensional radial functions multiplied with spherical harmonics $(Y_l^m(\theta, \varphi))$. The expansion coefficients $\nu_{lm}^A(r)$ of the bubbles are values of the functions in discrete radial points with respect to atom A multiplied with spherical harmonics having angular momentum quantum numbers $l = [0, l_{\text{MAX}}]$ and $m = [-l, l]$. The entire function can be evaluated in Cartesian coordinates as

$$\nu(x, y, z) = \nu^\Delta(x, y, z) + \sum_A \sum_{l=0}^{l_{\text{MAX}}} \sum_{m=-l}^{l} \nu_{lm}^A(r) Y_l^m(\theta, \varphi), \tag{1}$$

where $Y_l^m(\theta, \varphi)$ are the real spherical harmonics in Racah's normalization. All contributions are formally considered as the cube part contains the remainder after the bubbles have been projected out.

2.2 Bound-State Helmholtz Equation

The Helmholtz equation for bound electronic states reads

$$(-\frac{1}{2}\nabla^2 + \kappa^2)f(\mathbf{r}) = g(\mathbf{r}), \tag{2}$$

4

where $-\frac{1}{2}\nabla^2$ is the kinetic energy operator and κ is a real number. The Helmholtz equation has also previously been used for updating the orbitals in numerical self-consistent field (SCF) electronic structure calculations[9,29,50] In the iterative SCF scheme, the updated orbitals $\psi_n^{(i+1)}(\mathbf{r})$ are obtained as

$$\psi_n^{(i+1)}(\mathbf{r}) = \left(-\nabla^2 - 2\epsilon_n^{(i)}\right)^{-1}\left[-2V_n^{(i)}(\mathbf{r})\psi_n^{(i)}(\mathbf{r})\right]. \tag{3}$$

where the source function $g(\mathbf{r})$ in Eq. (2) consists of twice the SCF interaction potential $V_n(\mathbf{r})$ between the electrons in orbital n with the rest of the electrons, multiplied with the orbital $\psi_n(\mathbf{r})$ of iteration (i). The parameter κ is obtained from the orbital energy in iteration (i) as $\kappa = \sqrt{-2\epsilon_n^{(i)}}$. The Laplacian of the denominator in Eq. (3) can be eliminated by using an integral transformation

$$\left(-\nabla^2 - 2\epsilon_n^{(i)}\right)^{-1}g(\mathbf{r}) = \left(-\nabla^2 + \kappa^2\right)^{-1}g(\mathbf{r}) = \int \frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{4\pi|\mathbf{r} - \mathbf{r}'|}g(\mathbf{r}')\mathrm{d}\mathbf{r}'. \tag{4}$$

By inserting Eq. (4) to Eq. (3) we arrive at[50]

$$\psi_n^{(i+1)}(\mathbf{r}) = -\int \frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{4\pi|\mathbf{r} - \mathbf{r}'|}[2V_n^{(i)}(\mathbf{r}')\psi_n^{(i)}(\mathbf{r}')]\mathrm{d}\mathbf{r}'. \tag{5}$$

In spherical coordinates, the integrand (kernel) in Eq. (4) can be expanded in a formally infinite series as[51]

$$\frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} = \frac{2\kappa}{\pi}\sum_{l=0}^{\infty}\hat{I}_{l+\frac{1}{2}}(\kappa r_<)\hat{K}_{l+\frac{1}{2}}(\kappa r_>)(2l+1)P_l(\cos(\varphi)), \tag{6}$$

where $\hat{I}_{l+\frac{1}{2}}$ and $\hat{K}_{l+\frac{1}{2}}$ are the first and second modified spherical Bessel functions of order l, $P_l(\cos(\varphi))$ is a Legendre polynomial of order l and φ is the angle between vectors $\mathbf{r}$ and $\mathbf{r}'$. The smaller and the larger of r and r' is denoted by $r_< = \min(r, r')$ and $r_> = \max(r, r')$. By using complex spherical harmonics $\mathcal{Y}_l^m(\theta, \varphi)$, the expression for the bound-state kernel becomes

$$\frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} = 8\kappa \sum_{l=0}^{\infty} \hat{I}_{l+\frac{1}{2}}(\kappa r_<) \hat{K}_{l+\frac{1}{2}}(\kappa r_>) \sum_{m=-l}^{l} \mathcal{Y}_l^m(\theta, \varphi) \mathcal{Y}_l^{-m}(\theta', \varphi'). \tag{7}$$

An similar expression is obtained when using real spherical harmonics $Y_l^m(\theta, \varphi)$.

3 The Grid-Based Fast Multipole Method

3.1 Domain Division

The procedure for the numerical integration of the Helmholtz kernel is similar to the one used for calculating two-electron interaction potentials by integrating the Poisson kernel.[43] The six-dimensional kernel integral expression in Eq. (3) and Eq. (4) can be divided into subdomains representing *near-field* and *far-field* contributions depending on the distance between $\mathbf{r}$ and $\mathbf{r}'$. In the grid-based fast multipole method (GB-FMM) for calculating of two-electron interaction,[43] we considered only source functions that can be accurately expanded on an equidistant grid as the cube functions in the *double* basis, whereas here we employ the *double* basis where the steep part of the functions are taken into account by using bubbles functions and the remainder is expanded on an equidistant 3D grid.

The *application of the* Helmholtz kernel ($G(\mathbf{r}, \mathbf{r}')$) in Eq. (8) is divided into the bubbles, *near-field* (NF) and *far-field* (FF) contributions, where the kernel of the bubbles part is expanded in local one-center functions and the kernel of the cube is treated numerically in 3D

$$\int G(\mathbf{r}, \mathbf{r}')g(\mathbf{r}')\mathrm{d}\mathbf{r}' = \sum_A \int G(\mathbf{r}, \mathbf{r}')g^A(\mathbf{r}')\mathrm{d}\mathbf{r}' + \int G(\mathbf{r}, \mathbf{r}')g_{\mathrm{NF}}^{\Delta}(\mathbf{r}')\mathrm{d}\mathbf{r}' + \int G(\mathbf{r}, \mathbf{r}')g_{\mathrm{FF}}^{\Delta}(\mathbf{r}')\mathrm{d}\mathbf{r}'$$
$$\tag{8}$$

To compute the cube part of the Helmholtz kernel for a given source function, the computational domain is further divided into a number of subdomains called *boxes.* The number

of boxes depends on the size and shape of the studied molecule. Here, we consider only the case with cubic shapes of the domain and boxes, because cubic shapes of the domains lead readily to a tree-data structure where each node has eight *children*. The obtained octree structure has 8^M equal-sized boxes, where M is the maximum level. *The boxes at level i are children of corresponding boxes at level $i-1$, where $i = 1, M$.* Level $i = 0$ means the entire computational domain.
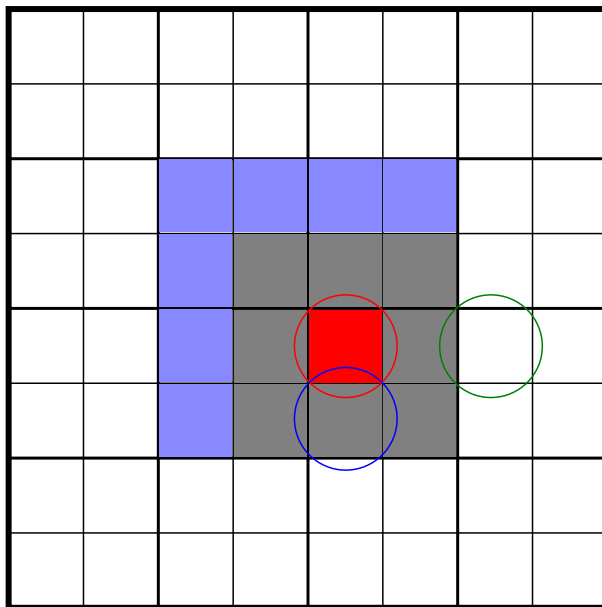

Figure 1: The grey squares represent the boxes belonging to the near-field of the red square, whereas the white ones belong to its far field. The circles illustrate overlapping and nonoverlapping domains representing the near-field and far-field contributions, respectively. The local far field is shown in blue.

To judge whether a given box b belongs to the near-field or far-field of box a, each box is enclosed by the smallest possible sphere. All boxes b that have an overlapping sphere with box a belong to the near-field of a, whereas the rest of the boxes belongs to its far-field. Because the bubbles functions and boxes overlap, they belong to the near-field of each box. The domain division is illustrated in Figure 1, where three boxes are enclosed by corresponding spheres. We can see that the box inside the green sphere does not belong to the near field of the red box a, as there is no overlap of the spheres. The blue sphere overlaps with the red one, indicating that the enclosed gray box belongs to the near field of the red

box. The far-field interaction is further split into *local* and *distant* contributions. *The local far-field of box a consists of all the sibling boxes of the nearest neighbors of a that do not belong to the near-field of a.* The local far-field of a are shown with blue squares in Figure 1.

To achieve an efficient parallelization, the boxes are distributed among the computational processes in such a way that the communication is minimized. In the best case, the shape of the *domain* of one process is cubic and the domain consists of one box at level i and all its children. For instance, if $M = 2$ and there are eight processes, the domain of each process consists of one box at level 1 and all its eight child boxes at level 2. Later in this article, we use the notation $Domain(i, L)$ to refer to boxes at a given level L, the calculations of which is preformed by prosess i.

3.2 Near-field Operator

The radial dependence of the bubbles functions $f_{lm}^A(r)$ representing a bubbles contribution to the updated orbital are obtained by integrating the radial part of the *bubbles contribution to the* Helmholtz kernel for each l and m value of the spherical harmonics

$$
\begin{aligned}
f_{lm}^A(r) = \;\; 8\kappa\big[\;\; & \hat{K}_{l+\frac{1}{2}}(\kappa r) \int_0^r g_{lm}^A(s)\hat{I}_{l+\frac{1}{2}}(\kappa s)s^2 \mathrm{d}s \\
+ \;\; & \hat{I}_{l+\frac{1}{2}}(\kappa r) \int_r^\infty g_{lm}^A(s)\hat{K}_{l+\frac{1}{2}}(\kappa s)s^2 \mathrm{d}s \big].
\end{aligned}
\tag{9}
$$

In Eq. (9), $g_{lm}^A(s)$ is the radial dependence of the bubbles source function at nucleus A, which is multiplied with the same spherical harmonics as $f_{lm}^A(r)$.

The near-field contribution to the updated orbital $f_a^\Delta(\mathbf{r})$ in box a is evaluated using the source function of the boxes $g_b^\Delta(\mathbf{r})$, where boxes b belong to the near field of box a. The near-field contributions to the updated orbital are calculated by employing the same numerical integration approach as used for the whole domain in the sequential implementation of our code.[18,50] The denominator of the Helmholtz kernel is eliminated by the integral

transformation

$$\frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} = \frac{2}{\sqrt{\pi}} \int_0^\infty \exp(-t^2(\mathbf{r} - \mathbf{r}')^2 - \kappa^2/4t^2)\mathrm{d}t. \tag{10}$$

By applying Eq. (10) to the cube part of the source function of boxes $g_b^\Delta(\mathbf{r})$ in Eq. (3) one obtains

$$f_a^\Delta(\mathbf{r}) = \frac{2}{\sqrt{\pi}} \int_0^\infty \exp(-\kappa^2/4t^2)Q_a(\mathbf{r}, t)\mathrm{d}t, \tag{11}$$

where

$$Q_a(\mathbf{r}, t) = \int \exp(-t^2(\mathbf{r} - \mathbf{r}')^2)g_b^\Delta(\mathbf{r}')\mathrm{d}^3\mathbf{r}'. \tag{12}$$

One integration is needed for each box, because all the nearest neighbor boxes are included in that calculation, *i.e.*, the evaluation of the nearfield contribution at the red area in Figure 1 can be evaluated by performing single numerical integration over the gray and red areas. The algorithm for the calculation of the near-field contribution to the updated orbital is shown in Figure 2.

---

evaluate bubbles potential using Eq. (9);
forall the *boxes $a \in Domain(i, M)$* **do**
  evaluate orbital contribution to box a, $f_a^\Delta(\mathbf{r})$, from the cube source function of the nearest neighbor boxes $g_b^\Delta(\mathbf{r})$ using Eq. (11) and Eq. (12).;
end

Figure 2: Evaluation of the near-field contributions to the orbital update.

3.3   Far-field Contributions

In the evaluation of far-field contributions to the updated orbital, the spherical harmonics expansion of the Helmholtz kernel is employed. The present method is similar to the GB-FMM approach for calculating Coulomb potentials. The GB-FMM approach for integrating the Helmholtz kernel is based on the two relations originating from the addition theorem in

Eq. (7). The first of them is the generalized multipole expansion

$$M_l^m(g_b^\Delta, \mathbf{r}_p) = \int g_b^\Delta(\mathbf{r}) R_l^m(\mathbf{r} - \mathbf{r}_p) \mathrm{d}\mathbf{r} = \int g_b^\Delta(\mathbf{r}) \hat{I}_{l+\frac{1}{2}}(\kappa r') Y_l^m(\theta', \varphi') \mathrm{d}\mathbf{r}, \qquad (13)$$

where $\mathbf{r}_p$ is the vector from origin to the center of the multipole expansion, and $\mathbf{r}' = \mathbf{r} - \mathbf{r}_p = (r', \theta', \varphi')$. The internal part, $R_l^m(\mathbf{r}')$, is regular for small input vector distances. The other relation is the local expansion

$$L_l^m(g_b^\Delta, \mathbf{r}_p) = \int g_b^\Delta(\mathbf{r}) S_l^m(\mathbf{r} - \mathbf{r}_p) \mathrm{d}\mathbf{r} = \int g_b^\Delta(\mathbf{r}) \hat{K}_{l+\frac{1}{2}}(\kappa r') Y_l^m(\theta', \varphi') \mathrm{d}\mathbf{r}, \qquad (14)$$

in which $S_l^m(\mathbf{r} - \mathbf{r}_p)$ is singular at $\mathbf{r} = \mathbf{r}_p$. By using these relations, the orbital update can be expressed using a generalized multipole expansion as

$$f_a(\mathbf{r}) = \frac{1}{4\pi} \int \frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} g_b^\Delta(\mathbf{r}') \mathrm{d}\mathbf{r}' = \frac{8\kappa}{4\pi} \sum_l^\infty \sum_{m=-l}^l \hat{K}_{l+\frac{1}{2}}(\kappa r) Y_l^m(\theta, \varphi) M_l^m(g_b^\Delta, \vec{0}) \qquad (15)$$

By using local expansion it can be written as

$$f_a(\mathbf{r}) = \frac{1}{4\pi} \int \frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{|\mathbf{r} - \mathbf{r}'|} g_b^\Delta(\mathbf{r}') \mathrm{d}\mathbf{r}' = \frac{8\kappa}{4\pi} \sum_l^\infty \sum_{m=-l}^l \hat{I}_{l+\frac{1}{2}}(\kappa r) Y_l^m(\theta, \varphi) L_l^m(g_b^\Delta, \vec{0}). \qquad (16)$$

However, the addition theorem in Eq. (7) limits the use of Eq. (15) to orbital contributions $f_a^\Delta(\mathbf{r})$ that are localized inside a sphere of radius r centered at the origin, whereas Eq. (16) can be used when the source function $g_b^\Delta(\mathbf{r})$ vanishes in the same sphere. This is analogous to the momentum expansion in the Coulomb case. Thus, it is the fundamental reason why the multipole method can be employed for calculating orbital-update contributions in a given subdomain from nonadjacent subdomains.

The use of Eq. (13) and Eq. (15) solves the scalability problem of the Helmholtz operator. However, the performance of the far-field evaluation based on these equations can still be

improved. The idea of the FMM approach is to explore translations of multipoles of local expansions and the conversion of multipole expansions to local expansions.[52] The expansions reduce the number of evaluations needed to calculate the far-field contribution to $f_a^\Delta(\mathbf{r})$ inside the sphere. We use the generalized GB-FMM approach in our final algorithm described in Figure 4. The implementation of the translation and conversion of the multipole expansions is discussed in the next subsection.

---

forall the *boxes $a \in Domain(i, M)$* **do**
    | calculate multipole moments of box a, centered at $\mathbf{r}_a$ using $M_l^m(g_a^\Delta, \mathbf{r}_a)$ in Eq. (13);
end
communicate multipole moments at level M;
for *level = M-1; level $\geq$ minlevel; level=level-1* **do**
    forall the *boxes $b \in Domain(i, level)$* **do**
        foreach *box $a \in Children(b)$* **do**
            Translate $M_l^m(g_a^\Delta, \mathbf{r}_a)$ to $M_l^m(g_a^\Delta, \mathbf{r}_b)$ (See Subsection 3.4);
            add the translated multipole moments to $M_l^m(g_b^\Delta, \mathbf{r}_b)$;
        end
    end
    communicate multipole moments at level *level*;
end

Figure 3: Multipole moment evaluation using computation process i

---

evaluate multipole moments at all levels as described in Figure 3 ;
for *level = minlevel; level $\leq$ M; level=level+1* **do**
    forall the *boxes $b \in Domain(i, level)$* **do**
        foreach *box $a \in LFF(b)$* **do**
            Convert $M_l^m(g_a^\Delta, \mathbf{r}_a)$ to $L_l^m(g_a^\Delta, \mathbf{r}_b)$ (See Subsection 3.4);
            add the translated multipole moments to $L_l^m(g_b^\Delta, \mathbf{r}_b)$;
        end
        use $L_l^m(g_b^\Delta, \mathbf{r}_b)$ to evaluate $f_{FF}^\Delta(\mathbf{r})$ in all grid points of box b using Eq. (16);
        add $f_{FF}^\Delta(\mathbf{r})$ to the final result $f^\Delta(\mathbf{r})$;
    end
end

Figure 4: Calculation of the far-field contribution to the orbital update using computational node i

3.4 Expansion Translation and Conversion

Useful expressions for calculating the multipole expansions in Eq. (13) and Eq. (14) are developed in this subsection. We concentrate on reformulating the integrands. Thus, for clarity we omit the integration symbol. The expansions of the multipole expressions are performed using complex spherical harmonics. Thus, the multipole expansions in real spherical harmonics $R_l^m(\mathbf{r})$ and $S_l^m(\mathbf{r})$ have to be converted to the corresponding representation in complex spherical harmonics before they are evaluated.

$$\mathcal{S}_l^m(\mathbf{r}) = \hat{K}_{l+\frac{1}{2}}(\kappa r)\mathcal{Y}_l^m(\theta, \varphi) \tag{17}$$

$$\mathcal{R}_l^m(\mathbf{r}) = \hat{I}_{l+\frac{1}{2}}(\kappa r)\mathcal{Y}_l^m(\theta, \varphi).$$

The relation between complex $(\mathcal{Y}_l^m(\theta, \varphi))$ and real spherical harmonics $(Y_l^m(\theta, \varphi))$ is

$$\mathcal{Y}_l^m(\theta, \varphi) = \begin{cases} \frac{1}{\sqrt{2}}\left[Y_l^{-m}(\theta, \varphi) - iY_l^m(\theta, \varphi)\right], & m < 0 \\ Y_l^m(\theta, \varphi), & m = 0 \\ \frac{(-1)^m}{\sqrt{2}}\left[Y_l^m(\theta, \varphi) + iY_l^{-m}(\theta, \varphi)\right], & m > 0 \end{cases} \tag{18}$$

and the conversion in the opposite direction is

$$Y_l^m(\theta, \varphi) = \begin{cases} \frac{-1}{\sqrt{2}}\left[\mathcal{Y}_l^m(\theta, \varphi) - (-1)^m\mathcal{Y}_l^{-m}(\theta, \varphi)\right], & m < 0 \\ \mathcal{Y}_l^m(\theta, \varphi), & m = 0 \\ \frac{1}{\sqrt{2}}\left[\mathcal{Y}_l^{-m}(\theta, \varphi) + (-1)^m\mathcal{Y}_l^m(\theta, \varphi)\right], & m > 0 \end{cases} \tag{19}$$

Several approaches to perform the generalized multipole expansions[53–55] and some bound state Helmholtz expansions[48] have been described in the literature. In this work, we use a similar nomenclature as used in Refs. 53–55.

Generally, a set of reexpansion coefficients, $(E|F)_{l,n}^{s,m}(\mathbf{r}'_{pq})$, are employed for transferring

the linear expansion coefficients $F_l^m(\mathbf{r}_q)$ at one expansion point $(\mathbf{r}_q)$ to another multipole expansion $E_n^m(\mathbf{r}_p)$ at position $\mathbf{r}_p$

$$E_n^m(\mathbf{r}_p) = \sum_{l=0}^{\infty} \sum_{s=-l}^{l} (E|F)_{l,n}^{s,m}(\mathbf{r}_{pq}')F_l^s(\mathbf{r}_q), \tag{20}$$

where $E_n^m(\mathbf{r}_p)$ and $F_l^s(\mathbf{r}_q)$ are regular $(\mathcal{R})$ or singular $(\mathcal{S})$ complex spherical harmonics expansions and $(E|F)_{l,n}^{s,m}$ can be reexpansion coefficients from singular to singular, $(\mathcal{S}|\mathcal{S})_{l,n}^{s,m}$, from regular to regular, $(\mathcal{R}|\mathcal{R})_{l,n}^{s,m}$, or from regular to singular, $(\mathcal{S}|\mathcal{R})$. The computational complexity of the approach can be reduced from $O(\mathcal{N}^4)$ to $O(\mathcal{N}^3)$ by first rotating the expansion to an orientation where the translation vector is along the z-axis. The complexity parameter $\mathcal{N}$ is proportional to the length of the multipole expansion. The expansion is then reexpanded in that direction and the resulting expansion is afterwards rotated back to the original direction. The complexity of the algorithm is then reduced because in the expansion to $E_n^m(\mathbf{r}_p)$ there is no angular dependency in the expansion coefficients and thus there are no contributions from $F_l^s(\mathbf{r}_q)$ coefficients where $s \neq m$. The expression in Eq. (20) then becomes

$$E_n^m(\mathbf{r}_p) = \sum_{l=|m|}^{\infty} (E|F)_{l,n}^m(\mathbf{r}_{pq}')F_l^m(\mathbf{r}_q), \tag{21}$$

where $(E|F)_{l,n}^m(\mathbf{r}_{pq}') = (E|F)_{l,n}^{m,m}(\mathbf{r}_{pq}')$ are the *coaxial reexpansion coefficients*.

We employ the rotation-translation-backrotation approach to transfer expansions described in Figure 5. The rotations are performed for real spherical harmonics expansions before and after they are converted and reexpanded, using the approach of Blanco *et al.*[56]. Generating coaxial reexpansion coefficients for the bound state case required some changes in existing algorithms developed for the non-bound case. These modifications are discussed in Subsection 3.4.1.

1. Rotate the real input expansion, $S_l^m(\mathbf{r}_q)$ or $R_l^m(\mathbf{r}_q)$ to a position where the z-axis is oriented along the translation vector $\mathbf{r}_{pq}$, using expressions in Ref. 56.

2. Convert the rotated real input expansion to complex input expansion $\mathcal{S}_l^m(\mathbf{r}_q)$ or $\mathcal{R}_l^m(\mathbf{r}_q)$ with Eq. (18).

3. Generate the coaxial translation coefficients $(E|F)_{l,n}^m(r'_{pq})$ as described in Subsection 3.4.1

4. Apply the coaxial translation coefficients to the complex input expansion using Eq. (21) yielding the complex output expansion $\mathcal{S}_l^m(\mathbf{r}_p)$ or $\mathcal{R}_l^m(\mathbf{r}_p)$.

5. Convert the complex output expansion back to real output expansion $S_l^m(\mathbf{r}_p)$ or $R_l^m(\mathbf{r}_p)$ using Eq. (19).

6. Rotate the real output expansion back to the original orientation.

Figure 5: Expansion algorithm for real spherical harmonics expansions.

3.4.1 Coaxial Translation Coefficient Generation

A large number of approaches have been developed for constructing coaxial reexpansion coefficients for Helmholtz expansions and for bound-state Helmholtz expansions.[45,48] We have here developed a bound-state version of the recursion approach by Gumerov et $al.$[53,54], where all coefficients $(E|F)_{l,n}^{m,m}(\mathbf{r}'_{pq})$ can be obtained from only a few initial values. We have reevaluated the equations in Ref. 53 for the bound state case. Proofs and detailed derivation are presented in $Appendix$ A. Here, we report the two recursion relations in Eq. (22) and Eq. (23), that are modified and more useful versions of Theorems 5 and 6.

$$(E|F)_{l,n+1}^m(\mathbf{r}'_{pq}) = \frac{a_l^s(E|F)_{l+1,n}^{s,m}(\mathbf{r}'_{pq}) + a_{l-1}^s(E|F)_{l-1,n}^{s,m}(\mathbf{r}'_{pq}) - a_{n-1}^m(E|F)_{l,n-1}^m(\mathbf{r}'_{pq})}{a_n^m} \qquad (22)$$

$$(E|F)_{l,n+1}^{m+1}(\mathbf{r}'_{pq}) = \frac{b_{l+1}^m(E|F)_{l+1,n}^m(\mathbf{r}'_{pq}) + b_l^{-m-1}(E|F)_{l-1,n}^m(\mathbf{r}'_{pq}) - b_n^m(E|F)_{l,n-1}^{m+1}(\mathbf{r}'_{pq})}{b_{n+1}^{-m+1}}. \qquad (23)$$

Eq. (23) is used when $n = m$ yielding

$$(E|F)^{m+1}_{l,m+1}(\mathbf{r}'_{pq}) = \frac{b^m_{l+1}(E|F)^m_{l+1,m}(\mathbf{r}'_{pq}) + b^{-m-1}_l(E|F)^m_{l-1,m}(\mathbf{r}'_{pq})}{b^{-m+1}_{m+1}}, \qquad (24)$$

where the last term of the numerator in Eq. (23) vanishes because $m + 1 > n = m - 1$. Eq. (22) and Eq. (24) have the following forms for the $(\mathcal{R}|\mathcal{R})$ translation

$$(\mathcal{R}|\mathcal{R})^m_{l,n+1}(\mathbf{r}'_{pq}) = \frac{a^s_l(\mathcal{R}|\mathcal{R})^{s,m}_{l+1,n}(\mathbf{r}'_{pq}) + a^s_{l-1}(\mathcal{R}|\mathcal{R})^{s,m}_{l-1,n}(\mathbf{r}'_{pq}) - a^m_{n-1}(\mathcal{R}|\mathcal{R})^m_{l,n-1}(\mathbf{r}'_{pq})}{a^m_n} \qquad (25)$$

$$(\mathcal{R}|\mathcal{R})^{m+1}_{l,m+1}(\mathbf{r}'_{pq}) = \frac{b^m_{l+1}(\mathcal{R}|\mathcal{R})^m_{l+1,m}(\mathbf{r}'_{pq}) + b^{-m-1}_l(\mathcal{R}|\mathcal{R})^m_{l-1,m}(\mathbf{r}'_{pq})}{b^{-m+1}_{m+1}}. \qquad (26)$$

For $(\mathcal{S}|\mathcal{S})$, they become

$$(\mathcal{S}|\mathcal{S})^m_{l,n+1}(\mathbf{r}'_{pq}) = \frac{-a^s_l(\mathcal{S}|\mathcal{S})^{s,m}_{l+1,n}(\mathbf{r}'_{pq}) - a^s_{l-1}(\mathcal{S}|\mathcal{S})^{s,m}_{l-1,n}(\mathbf{r}'_{pq}) - a^m_{n-1}(\mathcal{S}|\mathcal{S})^m_{l,n-1}(\mathbf{r}'_{pq})}{a^m_n} \qquad (27)$$

$$(\mathcal{S}|\mathcal{S})^{m+1}_{l,m+1}(\mathbf{r}'_{pq}) = \frac{-b^m_{l+1}(\mathcal{S}|\mathcal{S})^m_{l+1,m}(\mathbf{r}'_{pq}) - b^{-m-1}_l(\mathcal{S}|\mathcal{S})^m_{l-1,m}(\mathbf{r}'_{pq})}{b^{-m+1}_{m+1}} \qquad (28)$$

and finally for $(\mathcal{S}|\mathcal{R})$

$$(\mathcal{S}|\mathcal{R})^m_{l,n+1}(\mathbf{r}'_{pq}) = \frac{-a^s_l(\mathcal{S}|\mathcal{R})^{s,m}_{l+1,n}(\mathbf{r}'_{pq}) - a^s_{l-1}(\mathcal{S}|\mathcal{R})^{s,m}_{l-1,n}(\mathbf{r}'_{pq}) - a^m_{n-1}(\mathcal{S}|\mathcal{R})^m_{l,n-1}(\mathbf{r}'_{pq})}{a^m_n} \qquad (29)$$

$$(\mathcal{S}|\mathcal{R})^{m+1}_{l,m+1}(\mathbf{r}'_{pq}) = \frac{-b^m_{l+1}(\mathcal{S}|\mathcal{R})^m_{l+1,m}(\mathbf{r}'_{pq}) - b^{-m-1}_l(\mathcal{S}|\mathcal{R})^m_{l-1,m}(\mathbf{r}'_{pq})}{b^{-m+1}_{m+1}}. \qquad (30)$$

A fast and efficient algorithm to generate all needed coaxial reexpansion coefficients is summarized in Figure 6.

```
forall the 0 ≤ n ≤ n_max do
    evaluate (E|F)^0_{n,0}(r'_{pq}) and (E|F)^0_{0,n}(r'_{pq}) [See Eq. (55)), Eq. (56), and Eq. (58)];
end
communicate multipole moments at level M;
for m = 0; m ≤ n_max; m=m+1 do
    for n = m, n ≤ n_max; n=n+1 do
        for l = m, l ≤ n_max; l=l+1 do
            Evaluate (E|F)^m_{l,n+1}(r'_{pq}) [See Eq. (25), Eq. (27), and Eq. (29)];
        end
    end
    Evaluate (E|F)^{m+1}_{l,m+1}(r'_{pq}) [See Eq. (26), Eq. (28), and Eq. (30))];
end
```

Figure 6: The algorithm for generating the coaxial reexpansion coefficients

4   Results

The methods described in the previous sections have been implemented in a computer program using Fortran 2008 and Cuda C. The correctness and efficiency of the implementation has been demonstrated by performing Hartree-Fock calculations on H_2, H_2O, and CO. We compare the Hartree-Fock energies that are obtained using the parallel implementation employing the grid-based fast multipole method (GB-FMM) for the Coulomb and Helmholtz kernels with results obtained using the serial code employing our full numerical integration scheme.[50] The accuracy, timings and computational scaling of the parallel version of the GB-FMM approach are demonstrated by performing calculations using various inputs.

4.1   Benchmark Calculations on Small Molecules

The main factors affecting the accuracy in our approach are the grid spacings for the one dimensional grid of the bubbles and the 3D grid of the cube. Because of the steep shape of the bubbles functions, it is important to have a very dense bubbles grid. However, the computational costs as a function of the grid size of the bubbles grid are very small as compared to the increase in the computational costs when using denser grids for the cube. To avoid significant uncertainties due to the employed bubbles grid, we use in this work a

very large number of grid points to describe the radial dependence of the bubbles functions. The number of radial grid points is 36001 for hydrogen covering a radial range of $r = [0, 20]$ Å. For the other atoms, the number of grid points of the bubbles is scaled with the nuclear charge. Thus, the computational accuracy as a function of the number of grid points given in Table 1 depends only on the employed cube grid.

Other factors that affect the accuracy of the grid-based operators are the lengths of the multipole expansions. The input and output l_{MAX} values represent the maximum angular quantum number for the multipoles in Eq. (13) and the corresponding number of multipole moments used in the evaluation of the resulting local expansion in Eq. (16), respectively. Corresponding expansion parameters also exist in the GB-FMM expression of the Coulomb operator.[43] In the present calculations, we use $l_{MAX}^{IN} = l_{MAX}^{OUT} = 22$ for the Helmholtz operator, and $l_{MAX}^{IN} = l_{MAX}^{OUT} = 12$ for the Coulomb operator.

Table 1: The calculated Hartree-Fock energies for H_2, H_2O, and CO as a function of the grid step h. The energies are obtained using the GB-FMM and the full numerical integration scheme using the serial code, respectively.

Grid step	H_2	H_2O	CO
Serial			
$h = 0.20$	-1.130375595	-76.071028544	-112.789501793
$h = 0.15$	-1.134032068	-76.067244799	-112.792867814
$h = 0.10$	-1.133735078	-76.067470421	-112.792763416
$h = 0.05$	-1.133629272	-76.067474784	-112.792666765
GB-FMM			
$h = 0.20$	-1.130401575	-76.071171715	-112.790388750
$h = 0.15$	-1.133965953	-76.067135946	-112.793050452
$h = 0.10$	-1.133733946	-76.067473821	-112.792763956
$h = 0.05$	-1.133628982	-76.067471491	-112.792668171
Literature values	-1.133629571[a]	-76.067419[b]	-112.792463[c]

[a]Refs. [6,9] [b] Ref. [57] [c] Ref. [58]

The accuracy for the energies for H_2, H_2O, and CO in Table 1 calculated using the full numerical integration (serial) and the GB-FMM approaches is practically the same for a given set of parameters. The reference energy for H_2 have been calculated using numerical

methods. The reference energies for H_2O and CO have been calculated using very large Gaussian basis sets.[57,58] *Some values in Table 1 are lower than the energy in the basis-set limit, since numerical approaches are generally fully variational only when approaching the basis-set limit due to a variety of numerical approximations that have been introduced. The approximations are very accurate for dense grids, whereas for coarser grids, the numerical approximations may lead significant errors and energies that are slightly below the basis-set limit.*

Using a dense grid with $h = 0.05$, the H_2 calculation converges in eigth iterations when an energy threshold of 0.01 μE_h is used. In this example, the initial guess led to an error of 9 mE_h and each iteration reduced the energy difference to the correct result by more than 80%.

4.2   Performance

The main advantages with the GB-FMM approach is that the long-range contributions are accurately approximated by using multipole expansions leading to significant reduction in the computational efforts and an algorithm that formally scales linearly with the size of the grid. Furthermore, since the linear transformation of the near-field contributions, the bubbles projection, and the multipole method are well aimed for parallel computers a significant speed up is obtained when employing GPGPUs. The linear scaling of the algorithm is demonstrated by performing calculations on H_2 using three different computational resources consisting of one, two, and four identical nodes, respectively, each of them equipped with two NVIDIA Tesla K40 GPGPUs. The input parameters were chosen to be the same as used in the benchmarking calculations with a grid step h of 0.05.

The relative timings of different parts of the SCF calculation as a function of the number of computational nodes are plotted in Figure 7. With the employed computational resources, it takes 78.2 seconds wall time to perform one self-consistent-field (SCF) iteration on one node. With two and four nodes, one SCF iteration takes 42.6 s and 23.6 s, respectively.

The corresponding computational times for applying the Coulomb operator are 6.83 s, 3.47 s, and 1.81 s. Since the Helmholtz operator requires longer multipole expansions and the reexpansion coefficients have to be reevaluated at each application of the operator because of the energy dependence, it consumes more computational resources or 21.12 s, 10.80 s, and 5.51 s when using one, two, and four nodes, respectively.
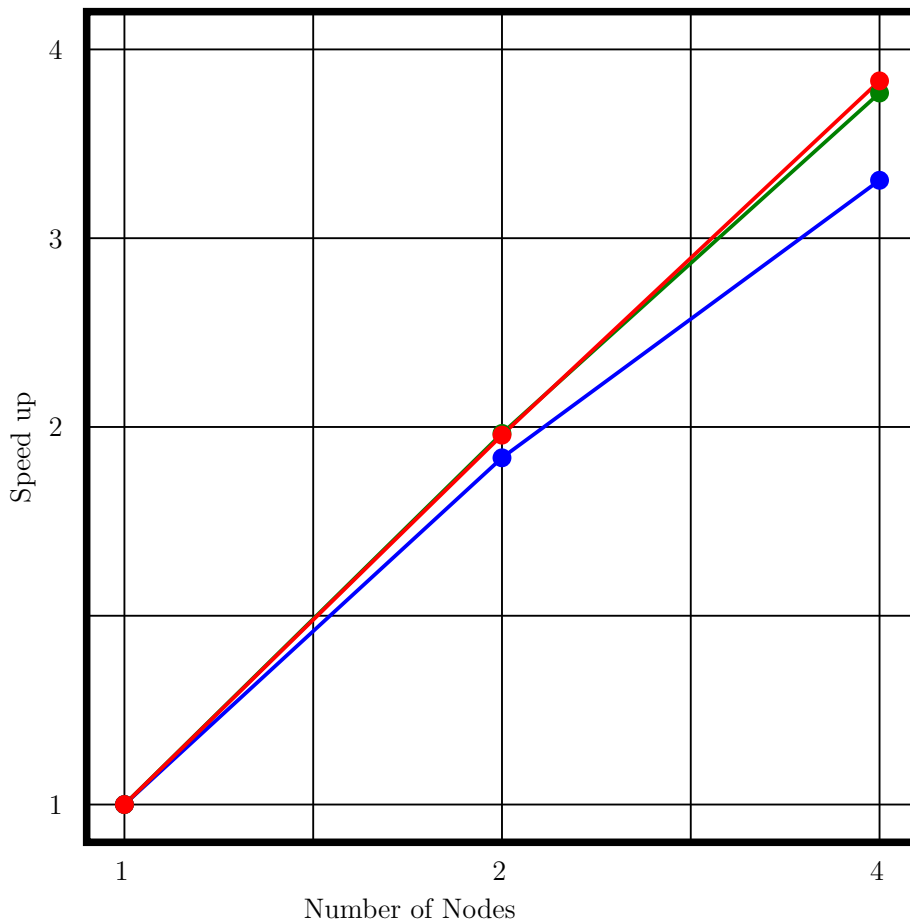

Figure 7: The relative speed up as a function of the number of computational nodes used for evaluating the Coulomb potential (red), the Helmholtz potential (green), and the total time for one SCF iteration (blue) for H_2 when employing the GB-FMM algorithms.

The timings show that the numerical methods scale practically linearly with the number of computational nodes. The linear scaling is almost perfect for the GB-FMM operators, whereas a small deviation from perfect linear scaling is obtained for the entire program. The main reasons for the deviation are the communication between the nodes, which is not present for the one-node case, and some segments of the software that are not parallelized.

The timings presented here are only for the current version of our software. The scaling of the whole program can be improved by eliminating some currently limiting bottlenecks, since there are not any fundamental reasons that prevent perfect parallelization. Calculations on H_2 is the worst case as far as the communication between computational nodes is concerned, because it that case most of the computational time is used in the potential evaluation that involves over 95% of the communication, whereas for larger systems more computational time goes to other functions in the code without significant communication.

Since the main part of the computations is performed on the GPGPUs, most efforts have been put on the optimization of the GPGPU part of the code, whereas the CPUs have not been maximally utilized. Thus, the CPUs are often idling when the calculations employ the GPGPUs. By utilizing the CPUs more efficiently one can expect significant increase in the computational speed without any needs for changes in the underlying algorithms. By starting two processes per each computational node, the computational time for an SCF loop on four nodes decreases by 13% from the 23.6 s to 20.6 s. The gain is partly due to a more effective usage of CPUs, but it also leads to a better use of the GPGPUs, as more CPUs submit computational tasks to the GPGPUs.

The calculations have in this work been performed on rather small molecules using modest computational resources. The algorithms allow other kind of parallelization schemes when a large number of nodes equipped with GPGPUs are employed. However, additional divide and conquer schemes are not only an option, such approaches will also be necessary because of the large memory requirement needed when calculating on large molecules.

5 Conclusions

We have developed a grid-based fast multipole method (GB-FMM) for optimizing numerical molecular orbitals and implemented it in our fully numerical Hartree-Fock code for electronic structure calculations on general molecules. The steep cusps in the vicinity of the nuclei are

considered by using a *double* basis, where the steep part of the functions are expanded in numerical one-center functions multiplied with spherical harmonics (*bubbles*), whereas the remainder of the functions (*cube*) is expanded on a three-dimensional (3D) equidistant grid. The Helmholtz kernel for the bubbles is integrated in spherical symmetry and a 3D numerical integration is performed for the cube part. We showed in this work that the cube integration can be made significantly faster by calculating the long-ranged contributions to the Helmholtz kernel by using a generalized grid-based fast multipole method. Thus, the time-consuming 3D integration has to be performed only for the near-field contributions. The use of an equidistant tensorial grid leads to homogeneous data structures and simple algorithms with little communication. The algorithms are well aimed for massively parallel computations using for example GPGPUs. The numerical Hartree-Fock calculations are shown to scale largely linearly with the number of computational nodes. By using a large number of GPGPUs, the computational speed for the 3D integration can be made faster than the rest of the calculations suggesting that the computational wall time can become independent of the grid size.

Acknowledgements

References

(1) Boys, S. F. Electronic wave functions I. A general method of calculation for the stationary states of any molecular system. *Proc. Roy. Soc. London* **1950**, *200*, 542–554.

(2) Slater, J. C. Atomic Shielding Constants. *Phys. Rev.* **1930**, *36*, 57–64.

(3) Phillips, J. C.; Kleinman, L. New Method for Calculating Wave Functions in Crystals and Molecules. *Phys. Rev.* **1959**, *116*, 287–294.

(4) Becke, A. D. Completely Numerical-Calculations on Diatomic-Molecules in the Local-Density Approximation. *Phys. Rev. A* **1986**, *33*, 2786–2788.

(5) McCullough Jr., E. A. Numerical Hartree-Fock Methods for Diatomic-Molecules - A Partial-Wave Expansion Approach. *Comput. Phys. Rep.* **1986**, *4*, 267–312.

(6) Laaksonen, L.; Pyykkö, P.; Sundholm, D. Fully Numerical Hartree-Fock Methods for Molecules. *Comput. Phys. Rep.* **1986**, *4*, 315–344.

(7) Kobus, J.; Laaksonen, L.; Sundholm, D. A numerical Hartree-Fock program for diatomic molecules. *Comput. Phys. Commun.* **1996**, *98*, 346–358.

(8) Beck, T. L. Real-space mesh techniques in density-functional theory. *Rev. Mod. Phys.* **2000**, *72*, 1041–1080.

(9) Harrison, R. J.; Fann, G. I.; Yanai, T.; Gan, Z.; Beylkin, G. Multiresolution quantum chemistry: basic theory and initial applications. *J. Chem. Phys.* **2004**, *121*, 11587–11598.

(10) Yanai, T.; Fann, G. I.; Gan, Z.; Harrison, R. J.; Beylkin, G. Multiresolution quantum chemistry in multiwavelet bases: Hartree–Fock exchange. *J. Chem. Phys.* **2004**, *121*, 6680–6688.

(11) Sundholm, D. Universal method for computation of electrostatic potentials. *J. Chem. Phys.* **2005**, *122*, 194107.

(12) Torsti, T. et al. Three real-space discretization techniques in electronic structure calculations. *Phys. Stat. Sol. B* **2006**, *243*, 1016–1053.

(13) Kronik, L.; Makmal, A.; Tiago, M. L.; Alemany, M. M. G.; Jain, M.; Huang, X.; Saad, Y.; Chelikowsky, J. R. PARSEC the pseudopotential algorithm for real-space

electronic structure calculations: recent advances and novel applications to nano-structures. *Phys. Status Solidi B* **2006**, *243*, 1063–1079.

(14) Natan, A.; Benjamini, A.; Naveh, D.; Kronik, L.; Tiago, M. L.; Beckman, S. P.; Chelikowsky, J. R. Real-space pseudopotential method for first principles calculations of general periodic and partially periodic systems. *Phys. Rev. B* **2008**, *78*, 75109.

(15) Genovese, L.; Neelov, A.; Goedecker, S.; Deutsch, T.; Ghasemi, S. A.; Willand, A.; Caliste, D.; Zilberberg, O.; Rayson, M.; Bergman, A.; Schneider, R. Daubechies wavelets as a basis set for density functional pseudopotential calculations. *J. Chem. Phys.* **2008**, *129*, 014109.

(16) Enkovaara, J. et al. Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method. *J. Phys.: Condens. Matter* **2010**, *22*, 253202.

(17) Andrade, X.; Alberdi-Rodriguez, J.; Strubbe, D. A.; Oliveira, M. J. T.; Nogueira, F.; Castro, A.; Muguerza, J.; Arruabarrena, A.; Louie, S. G.; Aspuru-Guzik, A.; Rubio, A.; Marques, M. A. L. Time-dependent density-functional theory in massively parallel computer architectures: the octopus project. *J. Phys. Cond. Mat.* **2012**, *24*, 233202.

(18) Losilla, S. A.; Sundholm, D. A divide and conquer real-space approach for all-electron molecular electrostatic potentials and interaction energies. *J. Chem. Phys.* **2012**, *136*, 214104.

(19) Sekino, H.; Matsumura, A.; Yokoi, Y.; Kato, T. Complete Space Quantum Chemistry By Multiresolution Multiwavelet Basis Set. *Int. J. Wavelets Multi.* **2013**, *11*, 1360008.

(20) Kobus, J. A finite difference Hartree-Fock program for atoms and diatomic molecules . *Comput. Phys. Commun.* **2013**, *184*, 799 – 811.

(21) Andrade, X.; Aspuru-Guzik, A. Real-Space Density Functional Theory on Graphical Processing Units: Computational Approach and Comparison to Gaussian Basis Set Methods. *J. Chem. Theory Comput.* **2013**, *9*, 4360–4373.

(22) Mohr, S.; Ratcliff, L. E.; Boulanger, P.; Genovese, L.; Caliste, D.; Deutsch, T.; Goedecker, S. Daubechies wavelets for linear scaling density functional theory. *J. Chem. Phys.* **2014**, *140*, 204110.

(23) Losilla, S. A.; Watson, M. A.; Aspuru-Guzik, A.; Sundholm, D. Construction of the Fock Matrix on a Grid-Based Molecular Orbital Basis Using GPGPUs. *J. Chem. Theory Comput.* **2015**, *11*, 2053–2062.

(24) Andrade, X. et al. Real-space grids and the Octopus code as tools for the development of new simulation approaches for electronic systems. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31371–31396.

(25) Khoromskaia, V.; Khoromskij, B. N. Tensor numerical methods in quantum chemistry: from Hartree-Fock to excitation energies. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31491–31509.

(26) Frediani, L.; Fossgaard, E.; Flå, T.; Ruud, K. Fully adaptive algorithms for multivariate integral equations using the non-standard form and multiwavelets with applications to the Poisson and bound-state Helmholtz kernels in three dimensions. *Mol. Phys.* **2013**, *111*, 1143–1160.

(27) Yanai, T.; Fann, G. I.; Beylkin, G.; Harrison, R. J. Multiresolution quantum chemistry in multiwavelet bases: excited states from time-dependent Hartree-Fock and density functional theory via linear response. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31405–31416.

(28) Mohr, S.; Ratcliff, L. E.; Genovese, L.; Caliste, D.; Boulanger, P.; Goedecker, S.; Deutsch, T. Accurate and efficient linear scaling DFT calculations with universal applicability. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31360–31370.

(29) Jensen, S. R.; Flå, T.; Jonsson, D.; Monstad, R. S.; Ruud, K.; Frediani, L. Magnetic properties with multiwavelets and DFT: the complete basis set limit achieved. *Phys. Chem. Chem. Phys.* **2016**, *18*, 21145–21161.

(30) Michaud-Rioux, V.; Zhang, L.; Guo, H. RESCU: A real space electronic structure method. *J. Comp. Phys.* **2016**, *307*, 593 – 613.

(31) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-Electron Integral Evaluation. *J. Chem. Theory Comput.* **2008**, *4*, 222–231.

(32) Ufimtsev, I.; Martinez, T. Graphical Processing Units for Quantum Chemistry. *Comput. Sci. Eng.* **2008**, *10*, 26–34.

(33) Olivares-Amaya, R.; Watson, M. A.; Edgar, R. G.; Vogt, L.; Shao, Y.; Aspuru-Guzik, A. Accelerating Correlated Quantum Chemistry Calculations Using Graphical Processing Units and a Mixed Precision Matrix Multiplication Library. *J. Chem. Theory Comput.* **2010**, *6*, 135–144.

(34) Asadchev, A.; Allada, V.; Felder, J.; Bode, B. M.; Gordon, M. S.; Windus, T. L. Uncontracted Rys Quadrature Implementation of up to G Functions on Graphical Processing Units. *J. Chem. Theory Comput.* **2010**, *6*, 696–704.

(35) DePrince, A. E.; Hammond, J. R. Coupled Cluster Theory on Graphics Processing Units I. The Coupled Cluster Doubles Method. *J. Chem. Theory Comput.* **2011**, *7*, 1287–1295.

(36) Isborn, C. M.; Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. Excited-State Electronic Structure with Configuration Interaction Singles and Tamm-Dancoff Time-Dependent Density Functional Theory on Graphical Processing Units. *J. Chem. Theory Comput.* **2011**, *7*, 1814–1823.

(37) Titov, A. V.; Ufimtsev, I. S.; Luehr, N.; Martinez, T. J. Generating Efficient Quantum Chemistry Codes for Novel Architectures. *J. Chem. Theory Comput.* **2013**, *9*, 213–221.

(38) Maurer, S. A.; Kussmann, J.; Ochsenfeld, C. Communication: A reduced scaling J-engine based reformulation of SOS-MP2 using graphics processing units. *J. Chem. Phys.* **2014**, *141*, 051106.

(39) Kussmann, J.; Ochsenfeld, C. Preselective Screening for Linear-Scaling Exact Exchange-Gradient Calculations for Graphics Processing Units and General Strong-Scaling Massively Parallel Calculations. *J. Chem. Theory Comput.* **2015**, *11*, 918–922.

(40) Fales, B. S.; Levine, B. G. Nanoscale Multireference Quantum Chemistry: Full Configuration Interaction on Graphical Processing Units. *J. Chem. Theory Comput.* **2015**, *11*, 4708–4716.

(41) Doran, A. E.; Hirata, S. Monte Carlo MP2 on Many Graphical Processing Units. *J. Chem. Theory Comput.* **2016**, *12*, 4821–4832.

(42) Choi, S.; Kwon, O.-K.; Kim, J.; Kim, W. Y. Performance of heterogeneous computing with graphics processing unit and many integrated core for hartree potential calculations on a numerical grid. *J. Comp. Chem.* **2016**, *37*, 2193–2201.

(43) Toivanen, E. A.; Losilla, S. A.; Sundholm, D. The grid-based fast multipole method - a massively parallel numerical scheme for calculating two-electron interaction energies. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31480–31490.

(44) Kalos, M. H. Monte Carlo Calculations of the Ground State of Three- and Four-Body Nuclei. *Phys. Rev.* **1962**, *128*, 1791–1795.

(45) Greengard, L. F.; Huang, J. A New Version of the Fast Multipole Method for Screened Coulomb Interactions in Three Dimensions . *J. Comp. Phys.* **2002**, *180*, 642–658.

(46) Zhang, B.; Huang, J.; Pitsianis, N. P.; Sun, X. Revision of FMM-Yukawa: An adaptive fast multipole method for screened Coulomb interactions. *Comput. Phys. Commun.* **2010**, *181*, 2206–2207.

(47) Cheng, H.; Crutchfield, W. Y.; Gimbutas, Z.; Greengard, L. F.; Ethridge, J. F.; Huang, J.; Rokhlin, V.; Yarvin, N.; Zhao, J. A wideband fast multipole method for the Helmholtz equation in three dimensions . *J. Comp. Phys.* **2006**, *216*, 300–325.

(48) Huang, J.; Jia, J.; Zhang, B. FMM-Yukawa: An adaptive fast multipole method for screened Coulomb interactions . *Comput. Phys. Commun.* **2009**, *180*, 2331–2338.

(49) Zimmermann, A.; Kuhn, S.; Richter, M. Poisson Green's function method for increased computational efficiency in numerical calculations of Coulomb coupling elements. *Phys. Rev. B* **2016**, *93*, 035308.

(50) Solala, E.; Losilla, S.; Sundholm, D.; Xu, W.; Parkkinen, P. Optimization of numerical orbitals using the Helmholtz kernel. *J. Chem. Phys.* **2016**, n/a–n/a, (manuscript).

(51) Abramowitz, M., Stegun, I. A., Eds. *Dover Books on Advanced Mathematics, New York: Corrected edition*; 1965.

(52) Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325–348.

(53) Gumerov, N. A.; Duraiswami, R. Fast, Exact, and Stable Computation of Multipole Translation and Rotation Coefficients for the 3-D Helmholtz Equation. *Institute for Advanced Computer Studies Technical Report UMIACS-TR 2001*, 44.

(54) Gumerov, N. A.; Duraiswami, R. Recursions for the Computation of Multipole Translation and Rotation Coefficients for the 3-D Helmholtz Equation. *SIAM J. Sci. Comput.* **2004**, *25*, 1344–1381.

(55) Gumerov, N. A., Duraiswami, R., Eds. *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*; Elsevier Series in Electromagnetism; Elsevier Science: Amsterdam, 2004.

(56) Blanco, M. A.; Flórez, M.; Bermejo, M. Evaluation of the rotation matrices in the basis of real spherical harmonics . *J. Mol. Struct. (Theochem)* **1997**, *419*, 19–27.

(57) Helgaker, T.; Klopper, W.; Koch, H.; Noga, J. Basis-set convergence of correlated calculations on water. *J. Chem. Phys.* **1997**, *106*, 9639–9646.

(58) Peterson, K. A.; Dunning, T. H. The CO molecule: the role of basis set and correlation treatment in the calculation of molecular properties. *J. Mol. Struct. (Theochem)* **1997**, *400*, 93–117.

Appendix A

In this Appendix, we rigorously prove that the recursion relations for multipole expansions lead to Eq. (22) and Eq. (23). The proof and the obtained recursion relations are very similar to the ones derived for the general Helmholtz equation by Gumerov *et al.*[53,54]

A.1. Derivatives for Multipole Expansions

We mainly follow the proof by Gumerov *et al.*, and merely discussing the differences caused by replacing spherical Bessel and spherical Hankel functions (j_n and h_n) with modified spherical Bessel functions of the first ($\hat{I}_{l+\frac{1}{2}}$) and second kind ($\hat{K}_{l+\frac{1}{2}}$), respectively. For details of the proof, we recommend the original papers by Gumerov *et al.*[53–55].

In the proof, the following differentiation operators in the spherical coordinates are used

$$\partial_z = \frac{\partial}{\partial z} = \mu \frac{\partial}{\partial r} + \frac{1-\mu^2}{r} \frac{\partial}{\partial \mu}, \mu = \cos(\theta). \tag{31}$$

$$\partial_{xy} = \frac{\partial}{\partial x} + i\frac{\partial}{\partial y} = \frac{e^{i\varphi}}{r\sqrt{1-\mu^2}}[(1-\mu^2)(r\frac{\partial}{\partial r} - \mu\frac{\partial}{\partial \mu}) + i\frac{\partial}{\partial \varphi})],$$

$$\overline{\partial_{xy}} = \frac{\partial}{\partial x} - i\frac{\partial}{\partial y} = \frac{e^{-i\varphi}}{r\sqrt{1-\mu^2}}[(1-\mu^2)(r\frac{\partial}{\partial r} - \mu\frac{\partial}{\partial \mu}) - i\frac{\partial}{\partial \varphi})],$$

$$\nabla = \mathbf{i}_x \frac{\partial}{\partial x} + \mathbf{i}_y \frac{\partial}{\partial y} + \mathbf{i}_z \frac{\partial}{\partial z} = \mathbf{i}_x \frac{1}{2}(\partial_{xy} + \overline{\partial_{xy}}) - \mathbf{i}_y \frac{i}{2}(\partial_{xy} - \overline{\partial_{xy}})$$

$$= \frac{1}{2}(\mathbf{i}_x - i\mathbf{i}_y)\partial_{xy} + \frac{1}{2}(\mathbf{i}_x + i\mathbf{i}_y)\overline{\partial_{xy}} + \mathbf{i}_z \partial_z$$

We define the functions $\mathcal{S}_l^m(\mathbf{r})$ and $\mathcal{R}_l^m(\mathbf{r})$ in Eq. (32) and Eq. (33), where the angular part is expanded in complex spherical harmonics and the radial part is Bessel functions of first and second kind, respectively. We note that the normalization constant $N_l^m = N_l^{-m}$ and that $P_l^m(\mu) = P_l^{-m}(\mu)$. We use the shorter N_l^m, $P_l^m(\mu)$ instead of $N_l^{|m|}$ and $P_l^{|m|}$ in the proof.

$$\mathcal{S}_l^m(\mathbf{r}) = N_l^m \hat{K}_{l+\frac{1}{2}}(\kappa r) P_l^m(\mu) e^{im\varphi} \tag{32}$$

$$\mathcal{R}_l^m(\mathbf{r}) = N_l^m \hat{I}_{l+\frac{1}{2}}(\kappa r) P_l^m(\mu) e^{im\varphi} \tag{33}$$

Theorem 1.

$$\frac{1}{\kappa}\partial_z F_l^m(\boldsymbol{r}) = a_{l-1}^m F_{l-1}^m(\boldsymbol{r}) + a_l^m F_{l+1}^m, \quad F = S, R, \tag{34}$$

where

$$a_l^m = 0 \; for \; l < |m|; \quad a_l^m = a_l^{|m|} = \sqrt{\frac{(l+1+|m|)(l+1-|m|)}{(2l+1)(2l+3)}} \; for \; l \geq |m| \tag{35}$$

Proof. We use following recursion relations for the Legendre polynomials

$$\mu P_l^m(\mu) = \frac{1}{(2l+1)}\left[(l+m)P_{l-1}^m(\mu) + (l-m+1)P_{l+1}^m(\mu)\right], \qquad (36)$$

$$(1-\mu^2)dP_l^m(\mu) = \frac{1}{(2l+1)}\left[(l+1)(l+m)P_{l-1}^m(\mu) - l(l-m+1)P_{l+1}^m\right] \qquad (37)$$

and following for the spherical modified Bessel functions

$$f_{l-1}(\kappa r) = \frac{l+1}{\kappa r}f_l(\kappa r) + f_l'(\kappa r) \qquad (38)$$

$$f_{l+1}(\kappa r) = -\frac{l}{\kappa r}f_l(\kappa r) + f_l'(\kappa r)$$

where f_l is $(-1)^l \hat{K}_{l+\frac{1}{2}}$ or $\hat{I}_{l+\frac{1}{2}}$ and the primes denote the first derivative. Using the first expression in Eq. (31) and the expressions in Eq. (32) and Eq. (33) we obtain

$$
\begin{aligned}
\frac{1}{\kappa}\partial_z F_l^m(\mathbf{r}) &= N_l^m e^{im\varphi}\left[\mu P_l^m(\mu)f_l'(\kappa r) + \frac{f_l(\kappa r)}{\kappa r}(1-\mu^2)\frac{dP^m(\mu)}{d\mu}\right] \\
&= \frac{N_l^m e^{im\varphi}}{(2l+1)}\Bigg\{(l+|m|)P_{l-1}^m(\mu)\left[f_l'(\kappa r) + \frac{(l+1)}{\kappa r}\right] \\
&\quad + (l-|m|+1)P_{l+1}^m(\mu)\left[-\frac{l}{\kappa r}f_l(\kappa r) + f_l'(\kappa r)\right]\Bigg\} \\
&= \frac{\kappa N_l^m e^{im\varphi}}{(2l+1)}\Bigg\{(l+|m|)f_{l-1}(\kappa r)P_{l-1}^m(\mu) + (l-|m|+1)P_{l+1}^m(\mu)f_{l+1}(\kappa r)\Bigg\}
\end{aligned}
$$

where the following recursion relations can be identified,

$$
\frac{1}{\kappa}\partial_z F_l^m(\mathbf{r}) =
\begin{cases}
-N_0^0 F_1^0(\mathbf{r}), & l = 0 \\
\frac{N_l^m}{2l+1}\left[\frac{(l+|m|)}{N_{l-1}^m}F_{l-1}^m(\mathbf{r}) + \frac{(l-|m|+1)}{N_{l+1}^m}F_{l+1}^m(\mathbf{r})\right], & l \geq 1
\end{cases}
$$

Thus for $l \geq 1$,

$$a_l^m = \frac{N_l^m}{N_{l+1}^m}\frac{(l-|m|+1)}{(2l+1)} = \sqrt{\frac{(l+|m|+1)(l-|m|+1)}{(2l+1)(2l+3)}}$$

30

□

The final expression and the coefficients are the same as obtained for the Helmholtz equation except for the sign between the first and the second term. This holds when the other differential operators in Eq. (31) are applied to $\mathcal{S}_l^m(\mathbf{r})$ and $\mathcal{R}_l^m(\mathbf{r})$ in Eq. (32) and Eq. (33). Therefore, proofs for Theorems 2, 3, and 4 are not given here. The reader who wants to become acquainted with the other proofs can read the original articles by Gumerov et al.[53–55]

Theorem 2. *For $\kappa \neq 0$ and integer l and m*

$$\frac{1}{\kappa}\partial_{xy}F_l^m(\boldsymbol{r}) = b_{l+1}^{-m-1}F_{l+1}^{m+1}(\boldsymbol{r}) + b_l^m F_{l-1}^{m+1}(\boldsymbol{r}), \tag{39}$$

where

$$b_l^m = \begin{cases} \sqrt{\frac{(l-m-1)(l-m)}{(2l-1)(2l+1)}}, & 0 \leq m \leq l, \\[2mm] -\sqrt{\frac{(l-m-1)(l-m)}{(2l-1)(2l+1)}}, & -l \leq m < l, \\[2mm] 0, & |m| > l, \end{cases} \tag{40}$$

Theorem 3. *For $\kappa \neq 0$ and integer l and m*

$$\frac{1}{\kappa}\overline{\partial_{xy}}F_l^m(\boldsymbol{r}) = b_{l+1}^{m-1}F_{l+1}^{m-1}(\boldsymbol{r}) + b_l^{-m}F_{l-1}^{m-1}(\boldsymbol{r}), \tag{41}$$

where b_l^m are defined in Eq. (40).

Theorem 4. *For $\kappa \neq 0$ and integer l and m*

$$\begin{aligned} \frac{1}{\kappa}\nabla F_l^m(\boldsymbol{r}) &= \frac{1}{2}(\boldsymbol{i}_x - i\,\boldsymbol{i}_y)\big[b_{l+1}^{-m-1}F_{l+1}^{m+1}(\boldsymbol{r}) + b_l^m F_{l-1}^{m+1}(\boldsymbol{r})\big] \\ &+ \frac{1}{2}(\boldsymbol{i}_x + i\,\boldsymbol{i}_y)\big[b_{l+1}^{m-1}F_{l+1}^{m-1}(\boldsymbol{r}) + b_l^{-m}F_{l-1}^{m-1}(\boldsymbol{r})\big] \\ &+ \frac{1}{2}\boldsymbol{i}_z\big[a_{l-1}^m F_{l-1}^m(\boldsymbol{r}) + a_l^m F_{l+1}^m(\boldsymbol{r})\big], \end{aligned} \tag{42}$$

where a_l^m and b_l^m are defined in Eq. (39) and Eq. (40), respectively.

A.2. Recurrence Relations for Translation Coefficients

In this subsection, we derive expressions for the coaxial reexpansion coefficients using the derivatives presented in the previous subsection. We present proofs for the two theorems that are needed for generating all coefficients $(E|F)^m_{ln}$. The recursion relation in Theorem 5 allows us to increase n or l, whereas Theorem 6 provides a tool for increasing m. Since the functions $\mathcal{R}^m_l$ and $\mathcal{S}^m_l$ in Eq. (32) and Eq. (33) follow the same recursion relation, the expansion coefficients $(E|F) = (\mathcal{S}|\mathcal{R})$, $(\mathcal{S}|\mathcal{S})$ or $(\mathcal{R}|\mathcal{R})$ can be obtained using the same expression (Eq. (43))

$$E^m_n(\mathbf{r}_p) = E^m_n(\mathbf{r}_q + \mathbf{r}_{pq}) = \sum_{l=0}^{\infty} \sum_{s=-l}^{l} (E|F)^{s,m}_{l,n}(\mathbf{r}_{pq}) F^s_l(\mathbf{r}_q), \tag{43}$$

where $E^m_n(\mathbf{r}_p)$, $F^s_l(\mathbf{r}_q)$ are expansions of type S or R centered at points denoted by $\mathbf{r}_p$ and $\mathbf{r}_q$, respectively. $\mathbf{r}_{pq}$ is the difference vector between the two centers. By denoting D_p as any of the operators $\partial/\partial z_p$, $\partial/\partial x_p \pm i\partial/\partial y_p$ with origin at $\mathbf{r}_p$ one obtains for a fixed $\mathbf{r}_{pq}$

$$D_p E^m_n(\mathbf{r}_p) = D_q E^m_n(\mathbf{r}_q + \mathbf{r}_{pq}) = \sum_{l=0}^{\infty} \sum_{s=-l}^{l} (E|F)^{s,m}_{l,n}(\mathbf{r}_{pq}) D_q F^s_l(\mathbf{r}_q), \tag{44}$$

Theorem 5. *For $\kappa \neq 0$ the following holds*

$$a^m_{n-1}(E|F)^{s,m}_{l,n-1}(\boldsymbol{r}_{pq}) + a^m_n(E|F)^{s,m}_{l,n+1}(\boldsymbol{r}_{pq}) = a^s_l(E|F)^{s,m}_{l+1,n}(\boldsymbol{r}_{pq}) + a^s_{l-1}(E|F)^{s,m}_{l-1,n}(\boldsymbol{r}_{pq}) \tag{45}$$

Proof. By setting $D_p = \kappa^{-1}\partial/\partial z_p = \kappa^{-1}\partial/\partial z_q = D_q$, and applying the operator to the reexpansion coefficient of Eq. (43) one obtains according to Theorem 1

$$\frac{1}{\kappa}\frac{\partial}{\partial x_p}E^m_n(\mathbf{r}_p) = \sum_{l=0}^{\infty} \sum_{s=-l}^{l} \left[a^m_{n-1}(E|F)^{s,m}_{l,n-1}(\mathbf{r}_{pq}) + a^m_n(E|F)^{s,m}_{l,n+1}(\mathbf{r}_{pq}) \right] F^s_l(\mathbf{r}_q).$$

By applying Theorem 1 to the same equation but choosing expansion $F^l_m(\mathbf{r}_q)$ instead of the

reexpansion coefficients one yields

$$
\begin{aligned}
\frac{1}{\kappa}\frac{\partial}{\partial x_p}E_n^m(\mathbf{r}_p) &= \sum_{l=0}^{\infty}\sum_{s=-l}^{l}(E|F)_{l,n}^{s,m}(\mathbf{r}_{pq})\frac{1}{\kappa}\frac{\partial}{\partial z_q}F_l^s(\mathbf{r}_q) \\
&= \sum_{l=0}^{\infty}\sum_{s=-l}^{l}(E|F)_{l,n}^{s,m}(\mathbf{r}_{pq})\left[a_{l-1}^s F_{l-1}^s(\mathbf{r}_q)+a_l^s F_{l+1}^s(\mathbf{r}_q)\right] \\
&= \sum_{l=-1}^{\infty}\sum_{s=-l-1}^{l+1} a_s^l (E|F)_{l+1,n}^{s,m}(\mathbf{r}_{pq})F_l^s(\mathbf{r}_q)+\sum_{l=1}^{\infty}\sum_{s=-l+1}^{l-1} a_{l-1,n}^s (E|F)_{l-1,n}^{s,m}(\mathbf{r}_{pq})F_l^s(\mathbf{r}_q) \\
&= \sum_{l=0}^{\infty}\sum_{s=-l}^{l}\left[a_l^s(E|F)_{l+1,n}^{s,m}(\mathbf{r}_{pq})+a_{l-1}^s(E|F)_{l-1,n}^{s,m}(\mathbf{r}_{pq})\right]F_l^s(\mathbf{r}_q),
\end{aligned}
$$

where the double sums can be combined because the coefficients a_n^m are zero when $n < 0$ or $m > |n|$. The last line of the two previous expressions show that the theorem holds. $\qquad\square$

Theorem 6. *For $\kappa \neq 0$ the following holds*

$$
b_n^m(E|F)_{l,n-1}^{s,m+1}(\boldsymbol{r}_{pq})+b_{n+1}^{-m+1}(E|F)_{l,n+1}^{s,m+1}(\boldsymbol{r}_{pq}) = b_{l+1}^{s-1}(E|F)_{l+1,n}^{s-1,m}(\boldsymbol{r}_{pq})+b_l^{-s}(E|F)_{l-1,n}^{s-1,m}(\boldsymbol{r}_{pq}) \quad (46)
$$

Proof. The proof is similar to the proof of Theorem 5. Now the operator ∂/∂_z is replaced by ∂/∂_x or ∂/∂_y. The proof of Theorem 6 in Ref. 53 is the same, except for the sign between the two terms on both sides of the expression, due to the sign differences in multipole reexpansion coefficient recursions. $\qquad\square$

Although Theorems 5 and 6 are valid for the three types of reexpansion coefficients $[(\mathcal{R}|\mathcal{R}), (\mathcal{S}|\mathcal{S})$ and $(\mathcal{S}|\mathcal{R})]$, it has to be emphasized that the signs are different in the respective recursion relations for $(\mathcal{S}|\mathcal{R})$ and $(\mathcal{S}|\mathcal{S})$ reexpansion coefficients. The explicit recursion relations for $(\mathcal{S}|\mathcal{R})$ in Eq. (45) and Eq. (46) read

$$
-a_{n-1}^m(\mathcal{S}|\mathcal{R})_{l,n-1}^{s,m}(\mathbf{r}_{pq}) - a_n^m(\mathcal{S}|\mathcal{R})_{l,n+1}^{s,m}(\mathbf{r}_{pq}) = a_l^s(\mathcal{S}|\mathcal{R})_{l+1,n}^{s,m}(\mathbf{r}_{pq}) + a_{l-1}^s(\mathcal{S}|\mathcal{R})_{l-1,n}^{s,m}(\mathbf{r}_{pq}) \quad (47)
$$

and

$$-b_n^m(\mathcal{S}|\mathcal{R})_{l,n-1}^{s,m+1}(\mathbf{r}_{pq}) - b_{n+1}^{-m+1}(\mathcal{S}|\mathcal{R})_{l,n+1}^{s,m+1}(\mathbf{r}_{pq}) = b_{l+1}^{s-1}(\mathcal{S}|\mathcal{R})_{l+1,n}^{s-1,m}(\mathbf{r}_{pq}) + b_l^{-s}(\mathcal{S}|\mathcal{R})_{l-1,n}^{s-1,m}(\mathbf{r}_{pq}).$$

(48)

For $(\mathcal{S}|\mathcal{S})$ the corresponding recursion relations are

$$a_{n-1}^m(\mathcal{S}|\mathcal{S})_{l,n-1}^{s,m}(\mathbf{r}_{pq}) + a_n^m(\mathcal{S}|\mathcal{S})_{l,n+1}^{s,m}(\mathbf{r}_{pq}) = -a_l^s(\mathcal{S}|\mathcal{S})_{l+1,n}^{s,m}(\mathbf{r}_{pq}) - a_{l-1}^s(\mathcal{S}|\mathcal{S})_{l-1,n}^{s,m}(\mathbf{r}_{pq}) \quad (49)$$

and

$$b_n^m(\mathcal{S}|\mathcal{S})_{l,n-1}^{s,m+1}(\mathbf{r}_{pq}) + b_{n+1}^{-m+1}(\mathcal{S}|\mathcal{S})_{l,n+1}^{s,m+1}(\mathbf{r}_{pq}) = -b_{l+1}^{s-1}(\mathcal{S}|\mathcal{S})_{l+1,n}^{s-1,m}(\mathbf{r}_{pq}) - b_l^{-s}(\mathcal{S}|\mathcal{S})_{l-1,n}^{s-1,m}(\mathbf{r}_{pq}). \quad (50)$$

The sign difference is caused by $(-1)^l$, which is included in the definition of the second modified spherical Bessel function $\hat{K}_{l+\frac{1}{2}}$ in Eq. (38). The sign factor affects the signs in the expressions for the reexpansion coefficients of $(\mathcal{S}|\mathcal{S})$ and $(\mathcal{S}|\mathcal{R})$.

A.3. Initial Values for Translation Coefficients

The initial values of the $(\mathcal{S}|\mathcal{R})_{l,0}^{s,0}$ coefficients can be easily obtained from recursion relations above. Rewriting the definition for the reexpansion coefficients in Eq. (43) as

$$\mathcal{S}_n^m(\mathbf{r}_p) = \mathcal{S}_n^m(\mathbf{r}_q + \mathbf{r}_{pq}) = \sum_{l=0}^{\infty} \sum_{s=-l}^{l} (\mathcal{S}|\mathcal{R})_{l,n}^{s,m}(\mathbf{r}_{pq}) \mathcal{R}_l^s(\mathbf{r}_q),$$

(51)

setting $m = n = 0$ and multiplying with $\frac{4\kappa}{\sqrt{\pi}}$ one obtains

$$\frac{e^{-\kappa \mathbf{r}_p}}{\mathbf{r}_p} = \frac{4\kappa}{\sqrt{\pi}} \mathcal{S}_0^0(\mathbf{r}_p) = \frac{4\kappa}{\sqrt{\pi}} \mathcal{S}_0^0(\mathbf{r}_q + \mathbf{r}_{pq}) = \frac{4\kappa}{\sqrt{\pi}} \sum_{l=0}^{\infty} \sum_{s=-l}^{l} (\mathcal{S}|\mathcal{R})_{l,0}^{s,0}(\mathbf{r}_{pq}) \mathcal{R}_l^s(\mathbf{r}_q).$$

(52)

Combining Eq. (7) and Eq. (17) one obtains

$$
\begin{aligned}
\frac{e^{-\kappa \mathbf{r}_p}}{\mathbf{r}_p} &= 8\kappa \sum_{l=0}^{\infty} \hat{I}_{l+\frac{1}{2}}(\kappa r_q)\hat{K}_{l+\frac{1}{2}}(\kappa r_{pq}) \sum_{m=-l}^{l} \mathcal{Y}_l^m(\theta_q,\varphi_q)(-1)^m \mathcal{Y}_l^{-m}(\theta_{pq},\varphi_{pq}) \qquad (53) \\
&= 8\kappa \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mathcal{S}_l^{-m}(-\mathbf{r}_{pq})\mathcal{R}_l^m(\mathbf{r}_q).
\end{aligned}
$$

which together with Eq. (52) yield an expression for the initial values of the $(\mathcal{S}|\mathcal{R})_{l,0}^{s,0}$ coefficients

$$
\begin{aligned}
(\mathcal{S}|\mathcal{R})_{l,0}^{s,0}(\mathbf{r}_{pq}) &= \sqrt{4\pi}\mathcal{S}_l^{-m}(-\mathbf{r}_{pq}) \qquad (54) \\
&= (-1)^m \sqrt{4\pi}\hat{K}_{l+\frac{1}{2}}(\kappa r_{pq})\mathcal{Y}_l^{-m}(\theta_{pq},\varphi_{pq}).
\end{aligned}
$$

For $s = 0$, the expression for translations along the z-axis ($\theta_{pq} = 0, \varphi_{pq} = 0$) can be written as

$$
\begin{aligned}
(\mathcal{S}|\mathcal{R})_{l,0}^{0,0}(r_{pq}) &= \sqrt{4\pi}\mathcal{S}_l^0(-\mathbf{r}_{pq}) \qquad (55) \\
&= \sqrt{4\pi}\hat{K}_{l+\frac{1}{2}}(\kappa r_{pq})\mathcal{Y}_l^0(0,0). \\
&= \sqrt{2l+1}\hat{K}_{l+\frac{1}{2}}(\kappa r_{pq})
\end{aligned}
$$

The $(\mathcal{R}|\mathcal{R})_{l,0}^{0,0}$ and $\mathcal{S}|\mathcal{S})_{l,0}^{0,0}$ values are

$$
(\mathcal{R}|\mathcal{R})_{l,0}^{0,0}(\mathbf{r}_{pq}) = \sqrt{2l+1}\hat{I}_{l+\frac{1}{2}}(\kappa r_{pq}) \qquad (56)
$$

and

$$
(\mathcal{S}|\mathcal{S})_{l,0}^{0,0}(\mathbf{r}_{pq}) = \sqrt{2l+1}\hat{I}_{l+\frac{1}{2}}(\kappa r_{pq}). \qquad (57)
$$

Additionally, for all of the reexpansion coefficients, the following holds

$$(E|F)^{0,0}_{l,0}(\mathbf{r}_{pq}) \;\; = \;\; (-1)^l (E|F)^{0,0}_{0,l}. \tag{58}$$