Dissertation zur Erlangung des Doktorgrades
der Fakultät für Chemie und Pharmazie
der Ludwig-Maximilians-Universität München

Development of Highly Efficient and Accurate Real-Space Integration Methods for Hartree-Fock and Hybrid Density Functional Calculations

Henryk Sebastian Laqua

aus

München

2022

Erklärung

Diese Dissertation wurde im Sinne von §7 der Promotionsordnung vom 28. November 2011 von Herrn Prof. Dr. Christian Ochsenfeld betreut.

Eidesstattliche Versicherung

Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, 12.10.2022

Henryk Sebastian Laqua

Dissertation eingereicht am:	07.06.2022
1. Gutachterin/Gutachter:	Prof. Dr. Christian Ochsenfeld
2. Gutachterin/Gutachter:	Prof. Dr. Regina de Vivie-Riedle
Mündliche Prüfung am:	27.07.2022

Danksagung

An erster Stelle gilt mein Dank Prof. Dr. Christian Ochsenfeld für die Möglichkeit, diese Dissertation in seinem Arbeitskreis, unter seine Anleitung, und mit seiner jahrelangen Unterstützung anzufertigen.
Des Weiteren bedanke ich mich bei Prof. Dr. Regina de Vivie-Riedle für das Anfertigen des Zweitgutachtens.
Darüberhinaus möchte ich mich ganz herzlich beim gesamten Arbeitskreis Ochsenfeld für das wunderbar positive Arbeitsklima bedanken. Mein besondere Dank gilt dabei Dr. Jörg Kussmann, Dr. Daniel Graf, Dr. Travis Thompson, Dr. Andrea Kreppel, Dr. Johannes Dietschreit, Lars Urban, Viktoria Drontschenko, und Fillipo Sacchetta für die ergiebige und oftmals auch sehr unterhaltsame Zusammenarbeit.

Zuletzt möchte ich mich auch bei meinen Freunden sowie meiner Familie für die Unterstützung während meinem Studium und meiner Promotion bedanken.

Summary

The central focus of molecular electronic structure theory is to find approximate solutions to the electronic Schrödinger equation for molecules, and as such represents an essential part of any theoretical (*in silico*) study of chemical processes. However, a steep increase of the computational cost with increasing system size often prevents the application of accurate approximations to the molecules of interest.

The main focus of the present work is the efficient evaluation of Fock-exchange contributions, which typically represents the computational bottleneck in Hartree-Fock (HF) and hybrid density functional theory (DFT) calculations. This bottleneck is addressed by means of seminumerical integration, i.e., one electronic coordinate within the 4-center-2-electron integral tensor is represented analytically and one numerically. In this way, an asymptotically linear scaling method for computing the exchange matrix (denoted as sn-LinK) is developed, enabling fast and accurate *ab-initio* calculations on large molecules, comprising hundreds or even thousands of atoms, even in combination with large atomic orbital basis sets.

The novel sn-LinK method comprises improvements to the numerical integration grids, a rigorous, batch-wise integral screening scheme, the optimal utilization of modern, highly parallel compute architectures (e.g., graphics processing units; GPUs), and an efficient combination of single- and double-precision arithmetic. In total, these optimizations enable over two orders of magnitude faster evaluation of Fock-exchange contributions. Consequently, this greatly improved performance allows to perform previously unfeasible computations, which is also demonstrated at the example of an *ab initio* molecular dynamics simulation (AIMD) study on the hydrogen bond strengths within double-stranded DNA.

In addition to Fock-exchange, the other two computational bottlenecks in hybrid-DFT applications – the evaluation of the Coulomb potential and the numerical integration of the semilocal exchange-correlation functional – are also addressed. Finally, more efficient methods to evaluate more accurate post-HF/DFT methods, namely the random-phase approximation (RPA) and the second-order approximate coupled cluster (CC2) method, are also put forward.

In this way, the highly efficient methods introduced in this thesis cover some of the most substantial computational bottlenecks in electronic-structure theory – the evaluation of the Coulomb- and the exchange-interactions, the integration of the semilocal exchange-correlation functional, and the computation of post-Hartree-Fock correlation energies. Consequently, computational chemistry studies on large molecules (>100 atoms) are accelerated by multiple orders of magnitude, allowing for much more accurate and thorough in-silico studies than ever before.

List of Publications

The present work is a cumulative dissertation comprising 11 articles (labeled I-XI) published in peer-reviewed Journals. All articles and the author's contribution to each of them are stated below:

I **H. Laqua**, J. Kussmann, C. Ochsenfeld
"Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals"
J. Chem. Theory Comput. **14**, 3451 (2018).
Contribution by the author: *All of the theory, all of the implementation, all of the test calculations and the writing.*

II **H. Laqua**, T. H. Thompson, J. Kussmann, C. Ochsenfeld
"Highly Efficient, Linear-Scaling Seminumerical Exact-Exchange Method for Graphic Processing Units"
J. Chem. Theory Comput. **16**, 1456 (2020).
Contribution by the author: *Most of the theory, most of the implementation, all of the test calculations and the writing.*

III **H. Laqua**, J. Kussmann, C. Ochsenfeld
"Accelerating seminumerical Fock-exchange calculations using mixed single- and double-precision arithmetic"
J. Chem. Phys. **154**, 214116 (2021).
Contribution by the author: *All of the theory, all of the implementation, all of the test calculations and the writing.*

IV **H. Laqua**, J. C. B. Dietschreit, J. Kussmann, C. Ochsenfeld
"Accelerating Hybrid Density Functional Theory Molecular Dynamic Simulations by Seminumerical Integration, Resolution-of-the-Identity Approximation, and Graphics Processing Units"
J. Chem. Theory Comput. **18**, 6010 (2022).
Contribution by the author: *Most of the theory, most of the implementation, all of the test calculations including the illustrative application and the writing.*

V L. Urban, **H. Laqua**, C. Ochsenfeld
"Highly Efficient and Accurate Computation of Multiple Orbital Spaces Spanning Fock Matrix Elements on Central and Graphics Processing Units for Application in F12 Theory"
J. Chem. Theory Comput. **18**, 4218 (2022).

Contribution by the author: *Development of theory and implementation in cooperation with L. Urban. Contributions to the manuscript.*

VI **H. Laqua**, J. Kussmann, C. Ochsenfeld
"An improved molecular partitioning scheme for numerical quadratures in density functional theory"
J. Chem. Phys. **149**, 204111 (2018).
Contribution by the author: *All of the theory, all of the implementation, all of the test calculations and the writing.*

VII J. Kussmann, **H. Laqua**, C. Ochsenfeld
"Highly Efficient Resolution-of-Identity Density Functional Theory Calculations on Central and Graphics Processing Units"
J. Chem. Theory Comput. **17**, 1512 (2021).
Contribution by the author: *Shared first authorship. Theory and Implementation of semi-local exchange-correlation potentials and gradients. All of the test calculations. Contributions to the manuscript.*

VIII **H. Laqua**, J. Kussmann, C. Ochsenfeld
"Communication: Density functional theory model for multi-reference systems based on the exact-exchange hole normalization"
J. Chem. Phys. **148**, 121101 (2018).
Contribution by the author: *All of the theory, all of the implementation, all of the test calculations and the writing.*

IX A. Kreppel, D. Graf, **H. Laqua**, C. Ochsenfeld
"Range-Separated Density-Functional Theory in Combination with the Random Phase Approximation: An Accuracy Benchmark"
J. Chem. Theory Comput. **16**, 2985 (2020).
Contribution by the author: *Development of theory and implementation of the range-separated PBE correlation functional in cooperation with A. Kreppel. Contributions to the manuscript.*

X V. Drontschenko, D. Graf, **H. Laqua**, C. Ochsenfeld
"Lagrangian-Based Minimal-Overhead Batching Scheme for the Efficient Integral-Direct Evaluation of the RPA Correlation Energy"
J. Chem. Theory Comput. **17**, 5623 (2021).
Contribution by the author: *Development of theory and implementation in cooperation with V. Drontschenko and D. Graf. Contributions to the manuscript.*

XI F. Sacchetta, D. Graf, **H. Laqua**, M. A. Ambroise, J. Kussmann, A. Dreuw, C. Ochsenfeld.
"An effective sub-quadratic scaling atomic-orbital reformulation of the scaled opposite-spin RI-CC2 ground-state model using Cholesky-decomposed densities and an attenuated Coulomb-metric"

J. Chem. Phys. **157**, 104104 (2022).
Contribution by the author: *Implementation of the block-sparse matrix algebra. Contributions to the manuscript.*

Contents

1 Introduction

The exact solution to the Schrödinger equation[1] provides, in principle, access to all non-relativistic properties of any physical system. However, the complexity of its solutions – the many-particle wave-functions – scales exponentially with the number of particles N, since the correlated movement of all N particles needs to be represented within this $3N$-dimensional function. Therefore, approximations to this exact treatment are essential to describe all but the smallest systems.

Kohn-Sham density functional theory (KS-DFT)[2] approximations, especially hybrid-DFT methods,[3–7] are arguably the most successful approximations to date in this regard, due to their exceptional price-performance ratio. These hybrid-DFT methods incorporate a fraction of the exact (Fock-) exchange energy, whose evaluation poses a significant computational bottleneck, which this thesis addresses by means of seminumerical integration.[8–29] That is, one electronic coordinate within the 4-center-2-electron (4c2e) electron repulsion integral (ERI) tensor is represented analytically employing Gaussian-type atomic orbitals (AOs) and one coordinate is represented numerically on molecular integration grids.

The resulting seminumerical exchange method reduces the computational scaling with respect to the size of the AO basis from $\mathcal{O}(N_{\text{bas}}^4)$ to $\mathcal{O}(N_{\text{bas}}^2)$, albeit at the cost of a larger pre-factor proportional to the number of grid points. Consequently, seminumerical integration is particularly advantageous for large molecules in combination with large basis sets. Moreover, this *formal* scaling can be further reduced to *asymptotically* linear-scaling with respect to the molecular size by exploiting the intrinsic locality of the exchange interaction for non-metallic systems (i.e., significant HOMO-LUMO gap). While such linear-scaling algorithms are now standard for the conventional (4c2e integral based) evaluation of the Fock-Matrix,[30–38] the seminumerical 3-center-1-electron (3c1e) integral based evaluation largely remains quadratic scaling, despite some efforts to reduce it via the chain-of-spheres exchange (COSX) algorithm.[16,19] Therefore, **Publication I** presents a LinK[34,35] and pre-LinK[36,37] inspired method to reduce the asymptotic scaling of the seminumerical exchange evaluation to linear by combining an initial pre-screening with a tight, density-dependent batch-wise selection scheme for the 3c1e integrals.

This method is further refined in **Publication II** employing very recently developed, rigorous and position-independent integral estimates.[39] In this way, the implementation of the 3c1e integral screening is significantly simplified, which greatly eases the transfer to graphics processing units (GPUs) providing up to 10× improved performance compared to central processing units (CPU). Furthermore, both the batch-wise nature of the integral screening and the lower local storage requirements for the computation of the 3c1e integrals compared to the 4c2e integrals make this linear-scaling seminumerical exchange method, denoted as sn-LinK [**Publication II**], particularly well suited for execution on GPUs, which provide very little low-level storage (e.g., L1, L2 cache) per thread and require identical branching within batches of typically 32 threads.

The sn-LinK method is subsequently improved even further in **Publication III** by exploring the possibility of executing most of the computation with 32-bit single precision (fp32) arithmetic instead of the standard 64-bit double precision (fp64) arithmetic,[40] providing up to 2× speedups on most CPUs and up to 64× speedups on some GPUs, especially on much more affordable consumer hardware. It is shown, however, that pure fp32 execution leads to unacceptably large numerical errors. Instead, the most significant contributions, i.e., the most significant 3c1e integrals, have to be computed with double-precision, whereas the vast majority ($\sim$99 %) of less significant contributions can be computed with reduced (fp32) precision, accelerating the integral evaluation by nearly 2× with virtually no impact ($<1\,\mu E_\mathrm{h}$) on the accuracy of the final result. In practice, this separation between fp64- and fp32-executed integrals is a straightforward extension of the sn-LinK integral-screening, substantiating the future value of the sn-LinK method.

Moreover, mixed-precision execution can be combined with incremental Fock-builds,[41,42] where, based on the linearity of the Fock-matrix with respect to the density matrix, the Fock matrix is not fully recomputed within each self-consistent-field (SCF) step and is instead only incremented from the previous step. Since these increments are many orders of magnitude smaller than the full Fock-matrix, they can, as shown in **Publication III**, indeed be computed with pure single precision without numerical artifacts, as long as one full Fock matrix as well as the final energy (and forces) are computed with higher numerical precision.

Next, the application of seminumerical integration to compute nuclear exchange-forces, i.e., the derivative of the exchange energy with respect to the nuclear positions, is studied in **Publication IV**. There, the value of seminumerical integration is especially notable, since the exchange-forces can be obtained at virtually no overhead from a converged SCF calculation if energy and forces are computed in one combined step. In particular, the evaluation of 3c1e integral derivatives can be completely avoided – a substantial advantage compared to fully analytical integration, where the necessary evaluation of the 4c2e integral derivatives is about 3-5 times more expensive than the 4c2e integrals themselves. The availability of such computationally affordable nuclear forces is especially relevant for *ab initio* molecular dynamics (AIMD) simulations, where millions of nuclear gradient calculations are required for a single trajectory. Therefore, the applicability of the sn-LinK method to AIMD simulations is also investigated in **Publication IV** studying the hydrogen bond strengths within double stranded DNA as an illustrative application.

Subsequently, in **Publication V** sn-LinK is applied to the complementary auxiliary basis set (CABS) singles corrections method,[43] which aims to reduce the one-particle basis set error and is typically combined with F12-theory in order to obtain highly accurate post-Hartree-Fock correlation energies by, e.g., second-order Møller Plesset perturbation theory (MP2-F12)[44–46] or the coupled cluster with singles, doubles, and perturbative triples approximation (CCSD(T)-F12).[47] For the CABS singles corrections, a full Fock-matrix has to be constructed in a large auxiliary basis, meaning that the reduced (quadratic instead of quartic) basis set scaling of seminumerical integration is even more impactful. Therefore, very impressive speedups of over 1000× can be obtained with the sn-LinK method in this case.

Next, since seminumerical integration requires numerical integration grids, optimized versions of the standard Becke-type[48] grids are developed in **Publication VI**. In particular, it is shown that Becke's molecular partitioning scheme – which is ubiquitously aplied in DFT calculations worldwide – leads to problematic numerical artifacts for weak, non-covalent interactions and a surprisingly simple adjustment is put forward, solving this problem entirely.

Furthermore, due to the profound acceleration of the Fock-exchange evaluation with sn-LinK, other steps in hybrid DFT calculations, namely the computation of the Coulomb interaction and the semilocal exchange-correlation (XC) functional can now also represent possible computational bottlenecks. Therefore, **Publication VII** describes the highly efficient execution of these two other steps: The Coulomb interaction is computed using the resolution-of-the-identity (RI) approximation[49] in combination with a variant of the J-engine algorithm[50,51] and the semilocal XC functional is numerically integrated employing the improved integration grids from **Publication VI**. Additionally exploiting the locality of the AO basis functions as well as GPU acceleration allows for very fast Kohn-Sham calculations, even for large molecules. E.g., the runtime for one Kohn-Sham build for the $(AT)_{16}$ DNA fragment with the def2-TZVP basis set[52] (1052 atoms, 22742 basis functions) is reduced from multiple hours to only 24 s.

Another noteworthy advantage of seminumerical integration is its natural connection to local-hybrid functionals, where – in contrast to global hybrid functionals – the functional incorporates a locally varying fraction of exact exchange.[53–59] Due to this higher flexibility, local hybrid functionals often provide a better description of difficult electronic structures, especially with regard to strong static correlation.[57–60] Inspired by the local-hybrid functional of Johnson,[57] **Publication VIII** presents a functional that accurately describes the unusual strong static correlation during covalent bond dissociation employing fractional orbital occupation numbers to renormalize the exchange-correlation hole for these strong-correlation structures.

In addition to these hybrid-Kohn-Sham methods, **Publications IX** and **X** study the random-phase approximation (RPA),[61–63] a potentially more accurate post-Kohn-Sham approximation. In **Publication IX**, RPA is combined with the semilocal Perdew-Burke-Ernzerhof (PBE) correlation functional[64] by range separation of the electron-electron interaction, i.e., short range correlation is described with PBE[65] while long-range correlation is described with RPA.[66,67] In this way, the overall accuracy of RPA is substantially improved, especially for smaller basis sets,[68] as shown by the extensive benchmark (>10000 calculations) in **Publication IX**.

In **Publication X**, the high computational cost of RPA and in particular its high memory demand is tackled by combining on-the-fly evaluation of the necessary 3-center-2-electron (3c2e) integrals with a Lagrangian-optimized batching-scheme that determines the best batch dimensions for any given molecule and memory configuration. In this way, an optimal trade-off between memory utilization and program runtime is achieved, enabling RPA calculations on large molecules (>1000 atoms) without limitations by the available system memory.

Finally, **Publication XI** applies a variant of this Lagrangian-optimized batching-scheme combined with the Cholesky decomposition of the ground state density[69,70] to the scaled opposite-spin second-order approximate coupled cluster singles and doubles (SOS-CC2) method[71,72] to reduce its computational scaling scaling from $\mathcal{O}(M^4)$ to asymptotically $\mathcal{O}(M)$. Thanks to the reduced computational scaling and the reduced memory demand, accurate CC2 calculations are now possible even for large molecules.

Below, the theoretical foundations for this thesis are outlined in chapter 2, followed by the complete collection of **Publications I-XI** – the main part of this dissertation – provided in chapter 3, finalized by some concluding remarks in chapter 4.

2 Theoretical Background

2.1 The Schrödinger Equation

Molecular electronic structure theory describes the movement of electrons and nuclei within molecules by solving the time-dependent Schrödinger equation (TDSE)[1]

$$i\frac{\partial}{\partial t}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_n,\mathbf{R}_1,\ldots,\mathbf{R}_N,t) = \hat{H}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_n,\mathbf{R}_1,\ldots,\mathbf{R}_N,t) \tag{2.1}$$

where the wave function Ψ depends on the coordinates (including spin) of all electrons $\mathbf{r}_i$ $(1 \leq i \leq n)$, the coordinates of all nuclei $\mathbf{R}_A$ $(1 \leq A \leq N)$, and the time t. The Hamilton operator $\hat{H}$ describes the movements and interactions of all particles and may be summarized as:

$$\hat{H} = \hat{T}_\mathrm{n} + \hat{T}_\mathrm{e} + \hat{V}_\mathrm{nn} + \hat{V}_\mathrm{en} + \hat{V}_\mathrm{ee}, \tag{2.2}$$

i.e., it comprises the kinetic energy of the nuclei $\hat{T}_\mathrm{n}$ and the electrons $\hat{T}_\mathrm{e}$, the Coulomb repulsion between nuclei $\hat{V}_\mathrm{nn}$ and between electrons $\hat{V}_\mathrm{ee}$, as well as the Coulomb attraction between electrons and nuclei $\hat{V}_\mathrm{en}$.

The stationary solutions of eq. (2.1) may be separated into a time-independent spatial wave function and a time-dependent phase-factor of the form

$$\Psi(\{\mathbf{r}_i\},\{\mathbf{R}_A\},t) = \Psi(\{\mathbf{r}_i\},\{\mathbf{R}_A\})e^{-iEt} \tag{2.3}$$

where $\Psi(\{\mathbf{r}_i\},\{\mathbf{R}_A\})$ solves the time-independent Schrödinger equation

$$\hat{H}\Psi = E\Psi. \tag{2.4}$$

Since the nuclei are at least 1800 times heavier than the electrons and therefore move significantly slower, the movement of the electrons can typically be separated from the movement of the nuclei (Born-Oppenheimer approximation)[73] which allows for the separation of the wave function into an electronic part Ψ_e and a nuclear part Ψ_n. Consequently, the electronic wave function Ψ_e depends only parametrically on the position of the nuclei and solves the electronic Schrödinger equation

$$(\hat{T}_\mathrm{e} + \hat{V}_\mathrm{ee} + V_\mathrm{en}(\mathbf{r}))\Psi_\mathrm{e}(\{\mathbf{r}_i\}) = E_\mathrm{e}\Psi_\mathrm{e}(\{\mathbf{r}_i\}), \tag{2.5}$$

where the electron-nuclear attraction is now simplified into a stationary 3-dimensional potential acting on each electron identically:

$$V_\mathrm{en}(\mathbf{r}) \equiv \sum_i v_\mathrm{en}(\mathbf{r}_i) \tag{2.6}$$

2.2 Hartree-Fock Theory

Since the electronic wave function of eq. (2.1) is still a $4N_\mathrm{e}$-dimensional function (three spatial and one spin-coordinate for each electron), it quickly becomes unfeasible to handle computationally, due to the exponential increase of the wave function domain with respect to the electron count. Therefore, efficient approximations for $\Psi_\mathrm{e}(\{\mathbf{r}_i\})$ have to be found.

2.2.1 Slater Determinants

In the simplest case, the electron-electron repulsion $\hat{V}_\mathrm{ee}$ is completely neglected, leading to a non-interaction (NI) differential equation of the form

$$(\hat{T}_\mathrm{e} + V_\mathrm{en}(\mathbf{r}))\Psi_\mathrm{e}^\mathrm{NI} = E_\mathrm{e}\Psi_\mathrm{e}^\mathrm{NI}, \tag{2.7}$$

which is solved exactly by any product (Hartree product)[74] of one-particle wave functions (orbitals) $\varphi_i(\mathbf{r}_j)$, e.g.,

$$\Psi_\mathrm{e}^\mathrm{NI}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n) = \varphi_1(\mathbf{r}_1)\varphi_2(\mathbf{r}_2)\ldots\varphi_n(\mathbf{r}_n) = \prod_i^n \varphi_i(\mathbf{r}_i), \tag{2.8}$$

and any permutation thereof, e.g.,

$$\varphi_1(\mathbf{r}_2)\varphi_2(\mathbf{r}_1)\ldots\varphi_n(\mathbf{r}_n). \tag{2.9}$$

Moreover, it is also solved exactly by any linear combination of such permuted Hartree products. However, only one of such linear combinations is antisymmetric with respect to the interchange of two electronic coordinates (Pauli principle)

$$\Psi_\mathrm{e}(\mathbf{r}_1, \ldots, \mathbf{r}_i, \ldots, \mathbf{r}_j, \ldots, \mathbf{r}_n) = -\Psi_\mathrm{e}(\mathbf{r}_1, \ldots, \mathbf{r}_j, \ldots, \mathbf{r}_i, \ldots, \mathbf{r}_n), \tag{2.10}$$

which a Fermionic wave function has to satisfy.[75] This specific linear combination of Hartree products, which can be expressed as a matrix determinant (Slater determinant)

$$\Psi^\mathrm{HF} \equiv \Phi = \frac{1}{\sqrt{n!}} \det \begin{pmatrix} \varphi_1(\mathbf{r}_1) & \varphi_2(\mathbf{r}_1) & \ldots & \varphi_n(\mathbf{r}_1) \\ \varphi_1(\mathbf{r}_2) & \varphi_2(\mathbf{r}_2) & \ldots & \varphi_n(\mathbf{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{r}_n) & \varphi_2(\mathbf{r}_n) & \ldots & \varphi_n(\mathbf{r}_n) \end{pmatrix} \tag{2.11}$$

is therefore the ansatz for the wave function in Hartree-Fock theory.[76,77]

2.2.2 Hartree-Fock Energy

Inserting this ansatz into the expression for the expectation value of the energy

$$E = \langle\Psi_\mathrm{e}|\hat{H}_\mathrm{e}|\Psi_\mathrm{e}\rangle \equiv \int \mathrm{d}\mathbf{r}_1 \int \mathrm{d}\mathbf{r}_2 \cdots \int \mathrm{d}\mathbf{r}_n \Psi_\mathrm{e}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n)\hat{H}_\mathrm{e}\Psi_\mathrm{e}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n) \tag{2.12}$$

results in the Hartree-Fock energy expression

$$E_{\mathrm{HF}} = \langle\Phi|\hat{H}_{\mathrm{e}}|\Phi\rangle = \sum_i \langle\varphi_i| -\frac{1}{2}\nabla_i^2 + v_{\mathrm{en}}(\mathbf{r}_i)|\varphi_i\rangle + \frac{1}{2}\sum_{ij}[(ii|jj) - (ij|ji)], \tag{2.13}$$

introducing the Mulliken integral notation

$$(ij|kl) \equiv \iint \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \varphi_i^*(\mathbf{r}_1)\varphi_j(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\varphi_k^*(\mathbf{r}_2)\varphi_l(\mathbf{r}_2). \tag{2.14}$$

The last term in eq. (2.13), denoted as the exchange energy $E_{\mathrm{X}} = \frac{1}{2}\sum_{ij}(ij|ji)$, is a direct consequence of the antisymmetry property of the Slater determinant and corresponds to a reduction of the electron-electron repulsion due to an inherent "correlation" of same-spin electrons (Fermi-correlation). The efficient and accurate computation of this exact (Fock-)exchange interaction and extensions thereof represents the main focus of this thesis, especially of **Publications I-V**.

2.2.3 Hartree-Fock Equations

So far, only the construction of an approximate many-body wave function (i.e., the Slater determinant) from orthonormal one-body functions (molecular orbitals) has been discussed. However, no equation to compute the precise form of these molecular orbitals (MOs) has been given.

Such equations are obtained by minimizing the Hartree-Fock energy expression of eq. (2.13) with respect to the MOs according to the variational principle and ensuring the MO orthonormality

$$\frac{\delta E_{\mathrm{HF}}}{\delta\varphi_i} \overset{!}{=} 0 \quad \text{with} \quad \langle\varphi_i|\varphi_j\rangle = \delta_{ij}, \tag{2.15}$$

by employing Lagrange's method of constrained optimization. This leads to the general Hartree-Fock equations

$$\hat{F}\varphi_i(\mathbf{r}) = \sum_j \varepsilon_{ij}\varphi_j(\mathbf{r}), \tag{2.16}$$

which, due to the invariance of the Fock-Operator $\hat{F}$ with respect to unitary orbital transformations, may be simplified into the canonical Hartree-Fock equations

$$\hat{F}\varphi_i(\mathbf{r}) = \varepsilon_i\varphi_i(\mathbf{r}), \tag{2.17}$$

where the matrix of the Lagrange multipliers ε_{ij} is diagonal.

Since the Fock operator

$$\hat{F} = -\frac{1}{2}\nabla^2 + V_{\mathrm{en}}(\mathbf{r}) + J(\mathbf{r}) + \hat{K} \tag{2.18}$$

itself also depends on the molecular orbitals through the Coulomb potential

$$J(\mathbf{r}) = \int \mathrm{d}\mathbf{r}' \frac{\sum_i |\varphi_i(\mathbf{r}')|^2}{|\mathbf{r} - \mathbf{r}'|} \tag{2.19}$$

and the exchange operator

$$\hat{K}\varphi_i(\mathbf{r}) = \sum_j \int \mathrm{d}\mathbf{r}' \frac{\varphi_i(\mathbf{r}')\varphi_j^*(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\varphi_j(\mathbf{r}), \tag{2.20}$$

the Hartree-Fock equations can only be solved iteratively employing the self-consistent-field (SCF) method.

2.2.4 Linear Combination of Atomic Orbitals

In order to evaluate eq. (2.17) numerically, the molecular orbitals are expanded in a fixed, finite set of atom-centered basis functions χ_ν (atomic orbitals, AOs) in the linear combination of atomic orbital (LCAO) ansatz

$$\varphi_i(\mathbf{r}) = \sum_\nu C_{\nu i}\chi_\nu(\mathbf{r}), \tag{2.21}$$

employing the linear expansion coefficients $C_{\nu i}$. Inserting this ansatz into eq. (2.17) and projecting onto one trial orbital χ_μ, leads to the Roothaan-Hall equations[78]

$$\sum_\nu \langle\chi_\mu|\hat{F}|\chi_\nu\rangle C_{\nu i} = \sum_\nu \langle\chi_\mu|\chi_\nu\rangle C_{\nu i}\varepsilon_i, \tag{2.22}$$

which represents a non-orthogonal matrix eigenvalue problem:

$$\mathbf{FC} = \mathbf{SC}\varepsilon. \tag{2.23}$$

This problem can be solved numerically using existing linear-algebra routines (matrix diagonalization) requiring $\mathcal{O}(N_{\mathrm{bas}}^3)$ operations, which only becomes a computational bottleneck for systems comprising multiple thousand atoms. In addition, a variety of diagonalization alternatives exist, achieving asymptotic linear-scaling for very large systems.[79–82]

2.2.5 The Fock Matrix

In contrast, for most systems of practical interest typically comprising a few hundred atoms, the formally $\mathcal{O}(N_{\mathrm{bas}}^4)$ scaling formation of the Fock matrix

$$F_{\mu\nu} = \langle\chi_\mu|\hat{F}|\chi_\nu\rangle \tag{2.24}$$

from the one-particle density matrix

$$P_{\mu\nu} = \sum_i C_{\mu i}C_{\nu i}, \tag{2.25}$$

requires the majority of the computation time.

In this step, the computation of the Coulomb matrix

$$J_{\mu\nu} = \langle\chi_\mu|J(\mathbf{r})|\chi_\nu\rangle = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu\nu|\lambda\sigma) \tag{2.26}$$

and the exchange matrix

$$K_{\mu\nu} = \langle \chi_\mu | \hat{K} | \chi_\nu \rangle = \sum_{\lambda\sigma} P_{\lambda\sigma} (\mu\sigma|\nu\lambda), \tag{2.27}$$

are particularly expensive. Therefore, alternative quadratures for $\mathbf{J}[\mathbf{P}]$ and especially $\mathbf{K}[\mathbf{P}]$, that improve on the formal $\mathcal{O}(N_{\mathrm{bas}}^4)$ time complexity, are studied within **Publications I-VI** and are discussed in more detail in section 2.4.

2.2.6 The Full Configuration Interaction Wave Function

Since the non-interacting wave function ansatz of eq. (2.11) cannot account for the correlated movement of electrons, the electronic structure of molecules and consequently their properties cannot be described exactly. Nevertheless, this ansatz provides a basis for a principally exact representation of the electronic wave function in the form of a linear combination of all possible excited Slater determinants:

$$\Psi_{\mathrm{FCI}} = c_0 \Phi_0 + \sum_{ia} c_i^a \Phi_i^a + \sum_{ijab} c_{ij}^{ab} \Phi_{ij}^{ab} + \sum_{ijkabc} c_{ijk}^{abc} \Phi_{ijk}^{abc} + \dots . \tag{2.28}$$

Since the set of all possible single (Φ_i^a), double (Φ_{ij}^{ab}), triple (Φ_{ijk}^{abc}), etc. excited determinants spans a complete basis for the interacting wave function, this full configuration-interaction (FCI) ansatz can, in principle, represent the interacting wave function exactly.

However, the exponential increase of the computational cost – the number of possible determinants scales as $\binom{N_{\mathrm{e}}}{N_{\mathrm{bas}}}$ – limits the practical application of eq. (2.28) to very small molecules. Therefore, efficient alternatives to this exact treatment need to be found. As such, the most commonly employed formulations of Kohn-Sham density functional theory (KS-DFT) provide alternatives at a similar or, if exchange interactions are not treated exactly, even lower cost as Hartree-Fock theory, while often, albeit not always, providing substantially more accurate results.

2.3 Kohn-Sham Density Functional Theory

The fundamental idea behind Kohn-Sham density functional theory[2] is to retain the non-interacting wave equation of eq. (2.7) which is consequently solved exactly by a single Slater determinant, but construct an additional potential $V_{\mathrm{XC}}(\mathbf{r})$ in such a way, that the resulting Slater determinant provides the same electron density

$$\rho(\mathbf{r}) = \langle \Psi_{\mathrm{e}} | \sum_i \delta(\mathbf{r} - \mathbf{r}_i) | \Psi_{\mathrm{e}} \rangle = N \int \mathrm{d}\mathbf{r}_2 \cdots \int \mathrm{d}\mathbf{r}_n |\Psi_{\mathrm{e}}(\mathbf{r}, \mathbf{r}_2, \dots, \mathbf{r}_n)|^2 \tag{2.29}$$

as the exact wave function. For a single Slater determinant, the electron density is directly available from the occupied MOs as

$$\rho(\mathbf{r}) = \sum_i |\varphi_i(\mathbf{r})|^2, \tag{2.30}$$

or the corresponding density matrix

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}\chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r}). \tag{2.31}$$

Similar to the effective potential, which ensures the exact electron density, the exchange-correlation energy functional $E_{\mathrm{XC}}[\rho]$ ensures the exact ground state energy (cf. eq. (2.13)), which is defined in the Kohn-Sham formalism as

$$E = E_{\mathrm{V}}[\rho] + E_{\mathrm{J}}[\rho] + E_{\mathrm{T}}[\{\varphi_i\}] + E_{\mathrm{XC}}[\rho]. \tag{2.32}$$

Here, the first two terms, namely the electron-nucleus attraction energy

$$E_{\mathrm{V}}[\rho] = \int \mathrm{d}\mathbf{r}\rho(\mathbf{r})V_{\mathrm{en}}(\mathbf{r}) \tag{2.33}$$

and the electron-electron Coulomb repulsion energy

$$E_{\mathrm{J}}[\rho] = \frac{1}{2}\iint \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \tag{2.34}$$

are direct analytical functionals of the electron density, whereas the non-interacting kinetic energy

$$E_{\mathrm{T}}[\{\varphi_i\}] = -\frac{1}{2}\sum_i \int \mathrm{d}\mathbf{r}\varphi_i^*(\mathbf{r})\nabla^2\varphi_i(\mathbf{r}) \tag{2.35}$$

is only known analytically as a functional of the occupied Kohn-Sham orbitals $\varphi_i(\mathbf{r})$, and the exact exchange-correlation energy functional E_{XC} is generally unknown.

This energy expression of eq. (2.32) is minimized by the Kohn-Sham orbitals, i.e., the solutions to the Kohn-Sham equation[2]

$$\big(-\frac{1}{2}\nabla^2 + V_{\mathrm{en}}(\mathbf{r}) + V_{\mathrm{J}}(\mathbf{r}) + V_{\mathrm{XC}}(\mathbf{r})\big)\varphi_i(\mathbf{r}) = \varepsilon_i\varphi_i(\mathbf{r}) \tag{2.36}$$

where the Coulomb potential

$$V_{\mathrm{J}}(\mathbf{r}) = \frac{\delta E_{\mathrm{J}}}{\delta\rho(\mathbf{r})} = \int \mathrm{d}\mathbf{r}'\frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \tag{2.37}$$

and the exchange-correlation potential

$$V_{\mathrm{XC}}(\mathbf{r}) = \frac{\delta E_{\mathrm{XC}}}{\delta\rho(\mathbf{r})} \tag{2.38}$$

can be obtained as the density variations of their respective energy expressions.

2.3.1 Jacob's Ladder of Density Functional Theory

In practice, analytical (semi-)local expressions of the form

$$E_{\text{XC}} = \int \mathrm{d}\mathbf{r} \varepsilon(\rho(\mathbf{r}), \nabla\rho(\mathbf{r}), \dots) \tag{2.39}$$

that employ only (semi-)local information of the electronic structure around each reference point, are used to approximate E_{XC}. The specific choice of DFT "ingredients" that enter eq. (2.39) determines the possible accuracy of an approximate functional and thus defines a hierarchy of practical density functional approximations (DFAs) – the "Jacob's ladder" of DFT.[83]

In the lowest rank, only the local electron density $\rho(\mathbf{r})$ enters, forming the local-density approximation (LDA).[84–86] In the second rank, also the gradient of the electron density $\nabla\rho(\mathbf{r})$ is utilized, defining the generalized-gradient approximation (GGA).[64,87–89] Adding the Laplacian of the electron density $\Delta\rho(\mathbf{r})$ and/or the noninteracting kinetic energy density

$$\tau(\mathbf{r}) = -\frac{1}{2}\sum_i \varphi_i^*(\mathbf{r})\nabla^2\varphi_i(\mathbf{r}) \tag{2.40}$$

leads to the third rank of the "Jacob's ladder", denoted as meta-GGA.[90–92]

The fourth rank adds a non-local dependence with respect to the occupied Kohn-Sham orbitals in the form of the exact exchange energy E_{X}, denoted as global hybrid functionals,[3–7] or the exact exchange energy *density*

$$\varepsilon_{\text{X}}^{\text{ex}}(\mathbf{r}) = \frac{1}{2}\sum_{ij}\int \mathrm{d}\mathbf{r}' \frac{\varphi_i^*(\mathbf{r})\varphi_j(\mathbf{r})\varphi_i(\mathbf{r}')\varphi_j^*(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \tag{2.41}$$

denoted as local hybrid functionals.[53–59] The latter class of functionals represents a promising approach to model static correlation due to the greater flexibility by incorporating exact exchange locally and is therefore studied in **Publications I** and **VIII**.

Finally, the fifth rank is defined by including information about the unoccupied (virtual) Kohn-Sham orbitals, typically by incorporating a fraction of the second-order Møller Plesset perturbation theory (MP2)[93] energy or the random phase approximation (RPA)[61–63] energy.[94–97] One such RPA based double-hybrid functional, which employs range-separation to mix a semilocal correlation functional with the RPA correlation functional is studied in **Publication IX** and the (memory-)efficient evaluation of the necessary RPA correlation energy is presented in **Publication X**.

2.4 Numerical Quadratures

2.4.1 Molecular Integration Grids

The integration of the exchange-correlation functional (eq. (2.39)) can, in general, not be performed analytically and therefore numerical integration over a finite three-dimensional integration grid is required. That is, the integral of eq. (2.39) is transformed into a finite sum over grid points $\mathbf{r}_g$ with corresponding weights w_g:

$$E_{\mathrm{XC}} \approx \sum_g w_g \varepsilon(\mathbf{r}_g). \tag{2.42}$$

In order to respect the spherical symmetry of the electronic structure around each nucleus, these molecular integration grids are typically constructed as a linear combination of spherical atomic grids, weighted according to Becke's molecular partitioning scheme[48] or variations thereof,[98] in order to account for the overlap of the individual grids and thus to avoid double-counting in the overlapping regions. A revised version of this molecular partitioning scheme which greatly improves the description of weakly bound complexes is developed in **Publication VI**.

The necessary DFT "ingredients" are then obtainable at each grid-point from the AO density matrix. To exemplify, the electron density $\rho(\mathbf{r}_g)$ can be computed with a formal $\mathcal{O}(N_{\mathrm{bas}}^2 N_{\mathrm{grid}} \sim M^3)$ time complexity as:

$$\rho(\mathbf{r}_g) = \sum_{\mu\nu} P_{\mu\nu} \chi_\mu(\mathbf{r}_g) \chi_\nu(\mathbf{r}_g), \tag{2.43}$$

which can be improved to asymptotically linear time complexity by exploiting the locality of the AO basis functions, as presented in **Publication VII**.

2.4.2 Seminumerical Integration

Numerical integration can, however, not only be used to evaluate semilocal DFT expressions, but also to accelerate the formally $\mathcal{O}(N_{\mathrm{bas}}^4)$-scaling evaluation of the exact exchange matrix (eq. (2.27)) by expressing one function pair of the 4-center-2-electron (4c2e) integral tensor (eq. (2.14)) numerically, i.e.,

$$(\mu\sigma|\nu\lambda) \approx \frac{1}{2}\left[([\mu\sigma]^{\mathrm{num}}|[\nu\lambda]^{\mathrm{ana}}) + ([\mu\sigma]^{\mathrm{ana}}|[\nu\lambda]^{\mathrm{num}})\right], \tag{2.44}$$

where

$$([\mu\sigma]^{\mathrm{num}}|[\nu\lambda]^{\mathrm{ana}}) \equiv \sum_g w_g \chi_\mu(\mathbf{r}_g) \chi_\sigma(\mathbf{r}_g) (g|\nu\lambda) \tag{2.45}$$

which involves only 3-center-1-electron (3c1e) integrals of the form

$$(g|\nu\lambda) \equiv \int \mathrm{d}\mathbf{r} \frac{\chi_\nu(\mathbf{r}) \chi_\lambda(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|}. \tag{2.46}$$

Inserting this tensor decomposition into eq. (2.27) leads to the seminumerical expression for the exchange matrix:

$$K_{\mu\nu} = \frac{1}{2} \sum_{\lambda\sigma g} w_g \chi_\mu(\mathbf{r}_g) (g|\nu\lambda) P_{\lambda\sigma} \chi_\sigma(\mathbf{r}_g) + \text{transpose}, \tag{2.47}$$

where the transpose is due to the symmetrized expression of eq. (2.44) and ensures the symmetry of the resulting approximate exchange matrix.

Equation (2.47) is best evaluated in three consecutive steps

$$\text{step 1:} \qquad F_{\lambda g} = \sum_\sigma P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) \tag{2.48}$$

$$\text{step 2:} \qquad G_{\nu g} = w_g \sum_\lambda (g|\nu\lambda) F_{\lambda g} \tag{2.49}$$

$$\text{step 3:} \qquad K_{\mu\nu} = \sum_g \chi_\mu(\mathbf{r}_g) G_{\nu g} \tag{2.50}$$

finalized by a symmetrization of $\mathbf{K}$ to account for the transpose in eq. (2.47). In this way, the time complexity of evaluating the exchange matrix is reduced from $\mathcal{O}(N_{\text{bas}}^4)$ to $\mathcal{O}(N_{\text{bas}}^2 N_{\text{grid}} \sim M^3)$, which is particularly beneficial for large AO basis sets. Exploiting the locality of the AO basis functions in combination with the locality of the exchange interaction for non-metallic systems (i.e., significant HOMO-LUMO gap), and employing rigorous, density-dependent integral screening techniques [**Publications I-III**] for the 3c1e integrals reduces the computational scaling to asymptotically linear. The development of such linear-scaling seminumerical methods is the main focus of this thesis and forms the basis of **Publications I-V**.

Another advantage of seminumerical integration is its simple and efficient extension to exchange-forces, i.e., the derivative of the exchange energy with respect to the nuclear positions.[16,22] Since the contribution from the perturbed density matrix can be substituted with the perturbed overlap matrix and the energy-weighted density matrix (Pulay-term),[99] only the contribution from the integral derivatives remains. This term of the form

$$E_{\text{X}}^x \equiv \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}P_{\lambda\sigma}(\mu\sigma|\nu\lambda)^x = 4\sum_{\mu\nu\lambda\sigma} P_{\mu\nu}P_{\lambda\sigma}(\mu^x\sigma|\nu\lambda), \tag{2.51}$$

where the superscript x denotes the derivative with respect to one nuclear coordinate, can be reformulated with a non-symmetric variant of the seminumerical tensor-decomposition of eq. (2.44) to

$$E_{\text{X}}^x \approx 4\sum_{\mu\nu\lambda\sigma} P_{\mu\nu}P_{\lambda\sigma}([\mu^x\sigma]^{\text{num}}|[\nu\lambda]^{\text{ana}}) \tag{2.52}$$

$$\equiv 4\sum_{\mu\nu\lambda\sigma g} w_g P_{\mu\nu}P_{\lambda\sigma}\chi_\mu^x(\mathbf{r}_g)\chi_\sigma(\mathbf{r}_g)(g|\nu\lambda) \tag{2.53}$$

where only the perturbed function-pair is expressed numerically in order to avoid explicit derivatives of the 3c1e integrals. In practice, eq. (2.53) is evaluated together with the final Fock-build to obtain the nuclear forces with only marginal computational overhead from the intermediate quantity $G_{\nu g}$ of eq. (2.49) as:

$$Z_{\mu g} = \sum_\nu P_{\mu\nu}G_{\nu g} \tag{2.54}$$

$$E_{\text{X}}^x = 4\sum_{\mu g} \chi_\mu^x(\mathbf{r}_g) Z_{\mu g}, \tag{2.55}$$

where the perturbed basis functions are available from the gradients of the basis functions as:

$$\chi_\mu^x(\mathbf{r}_g) = \begin{cases} -\frac{\partial}{\partial x}\chi_\mu(\mathbf{r}) & \chi_\mu \text{ centered at perturbed nucleus} \\ 0 & \text{otherwise} \end{cases} \tag{2.56}$$

The availability of nuclear gradients without any substantial computational overhead is particularly useful for ab-initio molecular dynamics (AIMD) simulations, where millions of gradient calculations are necessary for the computation of a single trajectory. Therefore, **Publication IV** studies the applicability of seminumerical exchange gradients within AIMD simulations.

2.4.3 Resolution-of-the-Identity Approximation

In contrast to the exchange matrix, the Coulomb matrix can be evaluated more efficiently by a different form of tensor decomposition – the resolution-of-the-identity (RI) method[49] – employing Coulomb-fitting for the 4c2e tensor

$$(\mu\nu|\lambda\sigma) \approx \sum_{PQ} (\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma), \tag{2.57}$$

where P,Q denote auxiliary AO basis functions[100] and $(P|Q)^{-1}$ denotes the *matrix* inverse of the 2-center-2-electron (2c2e) integrals

$$(P|Q) = \iint \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \frac{\chi_P(\mathbf{r})\chi_Q(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}. \tag{2.58}$$

Thus, the Coulomb matrix can be obtained as

$$J_{\mu\nu} = \sum_{\lambda\sigma PQ} P_{\lambda\sigma}(\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma) \tag{2.59}$$

which can be evaluated in $\mathcal{O}(N_{\text{bas}}^2 N_{\text{aux}} \sim M^3)$ time complexity, a significant improvement from the $\mathcal{O}(N_{\text{bas}}^4)$ complexity of evaluating eq. (2.26) directly.

The computational bottleneck is typically the on-the-fly evaluation of the 3-center-2-electron (3c2e) integrals

$$(\mu\nu|P) = \iint \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \frac{\chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r})\chi_P(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \tag{2.60}$$

which is substantially accelerated by employing a modified J-engine algorithm[50,51] in combination with GPU acceleration, as presented in **Publication VII**.

3 Publications

3.1 Publication I: Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals

H. Laqua, J. Kussmann, C. Ochsenfeld

J. Chem. Theory Comput. **14**, 3451 (2018).

Abstract

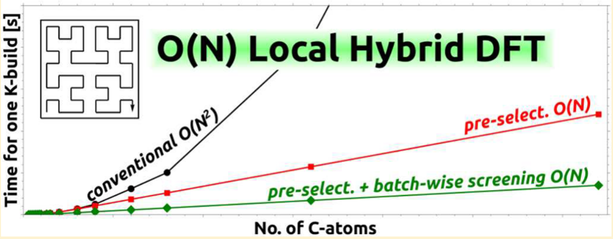
Local hybrid functionals, that is, functionals with local dependence on the exact exchange energy density, generalize the popular class of global hybrid functionals and extend the applicability of density functional theory to electronic structures that require an accurate description of static correlation. However, the higher computational cost compared to conventional Kohn-Sham density functional theory restrained their widespread application. Here, we present a low-prefactor, linear-scaling method to evaluate the local hybrid exchange-correlation potential as well as the corresponding nuclear forces by employing a seminumerical integration scheme. In the seminumerical scheme, one integration in the electron repulsion integrals is carried out analytically and the other one is carried out numerically, employing an integration grid. A high computational efficiency is achieved by combining the preLinK method [J. Kussmann and C. Ochsenfeld, *J. Chem. Phys.* **2013** *138*, 134114] with explicit screening of integrals for batches of grid points to minimize the screening overhead. This new method, denoted as preLinX, provides an 8-fold performance increase for a DNA fragment containing four base pairs as compared to existing S- and P-junction-based methods. In this way, our method allows for the evaluation of local hybrid functionals at a cost similar to that of global hybrid functionals. The linear-scaling behavior, efficiency, accuracy, and multinode parallelization of the presented method is demonstrated for large systems containing more than 1000 atoms.

JCTC Journal of Chemical Theory and Computation
Article
Cite This: J. Chem. Theory Comput. 2018, 14, 3451−3458
pubs.acs.org/JCTC

Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals

Henryk Laqua, Jörg Kussmann, and Christian Ochsenfeld*

Department of Chemistry and Center for Integrated Protein Science (CIPSM), University of Munich (LMU), D-81377 München, Germany



ABSTRACT: Local hybrid functionals, that is, functionals with local dependence on the exact exchange energy density, generalize the popular class of global hybrid functionals and extend the applicability of density functional theory to electronic structures that require an accurate description of static correlation. However, the higher computational cost compared to conventional Kohn−Sham density functional theory restrained their widespread application. Here, we present a low-prefactor, linear-scaling method to evaluate the local hybrid exchange−correlation potential as well as the corresponding nuclear forces by employing a seminumerical integration scheme. In the seminumerical scheme, one integration in the electron repulsion integrals is carried out analytically and the other one is carried out numerically, employing an integration grid. A high computational efficiency is achieved by combining the preLinK method [J. Kussmann and C. Ochsenfeld, *J. Chem. Phys.* **2013** *138*, 134114] with explicit screening of integrals for batches of grid points to minimize the screening overhead. This new method, denoted as preLinX, provides an 8-fold performance increase for a DNA fragment containing four base pairs as compared to existing S- and P-junction-based methods. In this way, our method allows for the evaluation of local hybrid functionals at a cost similar to that of global hybrid functionals. The linear-scaling behavior, efficiency, accuracy, and multinode parallelization of the presented method is demonstrated for large systems containing more than 1000 atoms.

1. INTRODUCTION

In the last decades, Kohn−Sham density functional theory (KS-DFT)[1] has become very popular in computational chemistry. The introduction of exact (Hartree−Fock like) exchange into DFT by Becke[2] created the popular class of (global-)hybrid functionals with numerous representatives.[3−6]

A more flexible approach, which incorporates exact exchange locally instead of globally, was then suggested by Jaramillo et al., introducing the class of local hybrid functionals.[7] Since then a variety of functionals of this form have been developed,[7−19] offering the chance of transferring the accuracy of hybrid DFT to a wider variety of problems, especially to ones requiring an accurate description of static correlation.

However, despite their advantages over conventional functionals, local hybrids have not made their way into mainstream applications yet because of their higher computational effort. The initially proposed resolution of the identity (RI) approaches[20−24] require large uncontracted basis sets and scale unfavorably as $O(N^3)$ with the system size and are therefore limited to small molecules. Recently, seminumerical implementations (SENEX),[25−28] where one integration in the electron repulsion integrals (ERIs) is carried out numerically on the DFT grid and the other one is carried out analytically, employing three-center-one-electron integrals, became more popular.

The seminumerical evaluation of exact exchange has been developed by Friesner and co-workers in the pseudospectral scheme since the late 1970s and is available in the Jaguar program package.[29−34] The approach became more popular since the development of the chains-of-spheres-exchange (COSX) algorithm by Neese et al.[35] and the SENEX implementation of the Weigend group.[36,37] However, both the pseudospectral and the COSX approach are designed for Hartree−Fock and global hybrid DFT computations, because some two-electron integrals (e.g., all integrals where all four basis-functions are centered at the same atom) are evaluated analytically to allow for the use of quite coarse integration grids while maintaining a reasonable accuracy. This is not possible for local hybrid functionals, so that the use of tight integration grids, comparable to typical DFT-integration grids, is inevitable.

In this work, we present an efficient linear-scaling method for local hybrid functionals employing the seminumerical integration scheme. Our method is based on rigorous screening techniques that exploit the locality of the exchange interaction for systems with a nonvanishing highest occupied molecular orbital−lowest unoccupied molecular orbital (HOMO−LUMO) gap.

During the last decades, several screening algorithms for analytical exact-exchange calculations have been developed.[38−43] A transfer of these integral screening techniques to seminumerical integration schemes is desirable to allow for the application of local hybrid functionals to larger systems with

Received: January 22, 2018
Published: June 15, 2018

hundreds or even thousands of atoms. The approach of the present paper utilizes the preLinK scheme of ref 42 to first preselect significant contributions to the final exchange matrix in order to reduce the asymptotic scaling to linear. In a subsequent step, significant three-center-one-electron-integrals are then selected for whole batches of grid points at once to further reduce the prefactor.

The underlying theory of the seminumerical evaluation of local hybrid exchange-correlation (XC) potentials and nuclear forces[44] follows the approach of Kaupp et al.[25,27] and is briefly summarized in section 2.1. Subsequently, we present our new (pre)screening method, denoted as preLinX, in section 3. Finally, the performance, accuracy, and parallel efficiency of our approach is assessed in section 4, employing the local hybrid functional of Perdew, Staroverov, Tao, and Scuseria (PSTS).[45]

2. SEMINUMERICAL EVALUATION OF LOCAL HYBRID FUNCTIONALS

In this section, the seminumerical scheme to evaluate the nonlocal (exact-exchange-dependent) part of local hybrid functionals is briefly summarized. A more detailed derivation of the underlying equations may be found in, for example, refs 25 and 27. Moreover, our summary is restricted to the exact-exchange-dependent part, because the evaluation of the semilocal part is equivalent to conventional meta-generalized gradient approximation (meta-GGA) functionals.

2.1. Local Hybrid Functional Form.

Local hybrid functionals incorporate, in addition to the usual meta-GGA ingredients (density ρ, square of the gradient of the density $|\nabla\rho|^2$, Laplacian of the density $\Delta\rho(\mathbf{r})$, and kinetic energy density $\tau = \sum_i \frac{1}{2}|\nabla\varphi_i(\mathbf{r})|^2$), the exact exchange energy density $\varepsilon_{\mathrm{X}}^{\mathrm{ex}}$, yielding the hyper-GGA functional form

$$E_{\mathrm{XC}}^{lh} = \int \varepsilon_{\mathrm{XC}}^{lh}\Big(\rho(\mathbf{r}),\, |\nabla\rho(\mathbf{r})|^2,\, \Delta\rho(\mathbf{r}),\, \tau(\mathbf{r}),\, \varepsilon_{\mathrm{X}}^{\mathrm{ex}}(\mathbf{r})\Big)\, \mathrm{d}\mathbf{r} \tag{1}$$

In the seminumerical scheme, the exact exchange energy density is computed for every grid point $\mathbf{r}_g$ as

$$\varepsilon_{\mathrm{X}}^{\mathrm{ex}}(\mathbf{r}_g) = -\frac{1}{2}\sum_{\mu\nu\lambda\sigma} \chi_\mu(\mathbf{r}_g) P_{\mu\nu} \int \frac{\chi_\nu(\mathbf{r}')\chi_\lambda(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}_g|}\mathrm{d}\mathbf{r}' P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) \tag{2}$$

where μ, ν, λ, and σ denote indices of basis functions χ and $\mathbf{P}$ denotes the one-particle density matrix representation within the atomic orbital (AO) basis.

2.2. Evaluation of the Local Hybrid Exchange−Correlation Potential.

The expression for the exchange−correlation potential of local hybrid functionals in the generalized Kohn−Sham (GKS) scheme is obtained by differentiation of the energy expression (eq 1) with respect to a density matrix element $P_{\mu\nu}$. For local hybrid functionals, this leads to an exact-exchange-dependent term of the form

$$\left[\frac{\partial\varepsilon_{\mathrm{XC}}^{lh}}{\partial\varepsilon_{\mathrm{X}}^{\mathrm{ex}}}(\mathbf{r}_g)\right] \times \left[\frac{\partial\varepsilon_{\mathrm{X}}^{\mathrm{ex}}}{\partial P_{\mu\nu}}(\mathbf{r}_g)\right] \tag{3}$$

containing the derivative (i.e., the potential matrix) of $\varepsilon_{\mathrm{X}}^{\mathrm{ex}}$, which is readily obtained by differentiation of eq 2 to yield

$$\frac{\partial\varepsilon_{\mathrm{X}}^{\mathrm{ex}}}{\partial P_{\mu\nu}}(\mathbf{r}_g) = -\frac{1}{2}\sum_{\lambda\sigma}\left[\chi_\mu(\mathbf{r}_g)\int\frac{\chi_\nu(\mathbf{r}')\chi_\lambda(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}_g|}\mathrm{d}\mathbf{r}' P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) + \chi_\nu(\mathbf{r}_g)\int\frac{\chi_\mu(\mathbf{r}')\chi_\lambda(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}_g|}\mathrm{d}\mathbf{r}' P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g)\right] \tag{4}$$

The potential matrix $\mathbf{K}$, which corresponds to the nonlocal part of the exchange−correlation potential, may thus be computed as

$$K_{\mu\nu} = -\frac{1}{2}\left[\sum_g \chi_\mu(\mathbf{r}_g) w_g \frac{\partial\varepsilon_{\mathrm{XC}}^{lh}}{\partial\varepsilon_{\mathrm{X}}^{\mathrm{ex}}}(\mathbf{r}_g)\sum_\lambda A_{\nu\lambda g}\sum_\sigma P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) + \text{transpose}\right] \tag{5}$$

where $\mathbf{r}_g$ denotes grid points, w_g denotes grid weights, and defining the three-center-one-electron integrals

$$A_{\nu\lambda g} = \int\frac{\chi_\nu(\mathbf{r}')\chi_\lambda(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}_g|}\mathrm{d}\mathbf{r}' \tag{6}$$

Similar to the approach of ref 35, eq 5 is evaluated in three consecutive steps on a numerical integration grid as

$$F_{\lambda g} = \sum_\sigma P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) \tag{7}$$

$$G_{\nu g} = -\sum_\lambda w_g\left[\frac{\partial\varepsilon_{\mathrm{XC}}^{lh}}{\partial\varepsilon_{\mathrm{X}}^{\mathrm{ex}}}(\mathbf{r}_g)\right]A_{\nu\lambda g}F_{\lambda g} \tag{8}$$

$$K_{\mu\nu} = \sum_g \chi_\mu(\mathbf{r}_g)G_{\nu g} \tag{9}$$

The integrals $A_{\nu\lambda g}$ are evaluated using the Obara−Saika recursion scheme[46,47] with automatically generated integral kernels for the different l-quantum-number combinations.

The so obtained nonlocal GKS-matrix $\mathbf{K}$ is finally symmetrized (to account for the transpose in eq 5) as

$$K_{\mu\nu}^{\mathrm{symm}} = \frac{1}{2}(K_{\mu\nu} + K_{\nu\mu}) \tag{10}$$

2.3. Nuclear Forces.

Expressions for nuclear forces, i.e., derivatives of the energy with respect to the positions of the nuclei, are obtained in a similar fashion as the expressions for the exchange−correlation potential and following the approach of ref 27.

Differentiation of eq 1 with respect to the nuclear positions (denoted as ∇_A) leads to an exact-exchange-dependent term of the form

$$\left[\frac{\partial\varepsilon_{\mathrm{XC}}^{lh}}{\partial\varepsilon_{\mathrm{X}}^{\mathrm{ex}}}(\mathbf{r}_g)\right] \times [\nabla_A\varepsilon_{\mathrm{X}}^{\mathrm{ex}}(\mathbf{r}_g)] \tag{11}$$

incorporating the derivatives of the exact exchange energy density $\nabla_A\varepsilon_{\mathrm{X}}^{\mathrm{ex}}$. An expression for the latter is obtained by differentiation of eq 2 to give

DOI: 10.1021/acs.jctc.8b00062
J. Chem. Theory Comput. 2018, 14, 3451−3458

$$\nabla_A e_X^{ex}(\mathbf{r}_g) = -\sum_{\mu\nu\lambda\sigma} \chi_\mu(\mathbf{r}_g)(\nabla_A P_{\mu\nu})A_{\nu\lambda g}P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) \tag{12}$$

$$-\sum_{\mu\nu\lambda\sigma} [\nabla_A \chi_\mu(\mathbf{r}_g)]P_{\mu\nu}A_{\nu\lambda g}P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) \tag{13}$$

$$-\frac{1}{2}\sum_{\mu\nu\lambda\sigma} \chi_\mu(\mathbf{r}_g)P_{\mu\nu}(\nabla_A A_{\nu\lambda g})P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) \tag{14}$$

Similar to conventional forces calculations, the term of eq 12 contains the derivatives of the density matrix $\nabla_A P_{\mu\nu}$, which can, however, be simplified by employing derivatives of the overlap matrix. This term is already included in conventional forces implementations.[44]

The other two terms are then evaluated seminumerically employing an integration grid: The term of eq 13, denoted as G_1, is calculated from the quantity $G_{\nu g}$ defined in eq 8 and the gradient of the basis functions $\nabla\chi_\mu$, employing two additional steps:

$$Z_{\mu g} = \sum_{\nu} P_{\mu\nu}G_{\nu g} \tag{15}$$

$$G_1 = -\sum_{\mu g} [\nabla\chi_\mu(\mathbf{r}_g)]Z_{\mu g} \tag{16}$$

Furthermore, the term of eq 14, denoted as G_2, can be evaluated directly from the integral derivatives as

$$G_2 = -\sum_{\nu\lambda g} w_g \left[\frac{\partial \varepsilon_{XC}^{lh}}{\partial \varepsilon_X^{ex}}(\mathbf{r}_g)\right] F_{\nu g}(\nabla_A A_{\nu\lambda g})F_{\lambda g} \tag{17}$$

where the intermediate quantity $F_{\nu g}$, defined in eq 7, is employed. The three-center-one-electron integral derivatives $\nabla_A A_{\nu\lambda g}$ are obtained analogously to the integrals in eq 6 from the Obara–Saika recursion scheme[47] using our code generator for the different l-quantum-number combinations.

3. LINEAR-SCALING CALCULATION OF THE EXACT EXCHANGE ENERGY DENSITY: THE PRELINX METHOD

To enable the application of local hybrid functionals to large molecular systems, an efficient and linear-scaling screening algorithm that removes as many insignificant integrals as possible is necessary. Therefore, we propose the following two-step-scheme: First, significant basis function shells are determined for each batch of grid points using a combination of the preLinK method[42] with the S- and P-junction-based method of Neese et al.[35] Subsequently, significant integrals are selected by direct investigation of the contribution to the total energy.

3.1. Preselection of Significant Contributions.

The prescreening is performed using the preLinK scheme,[42] which provides a rigorous upper bound to the elements of the **K** matrix of the form

$$|K_{\mu\nu}| \le \tilde{K}_{\mu\nu} = \sum_{\lambda\sigma} \sqrt{(\mu\lambda|\mu\lambda)}\,|P_{\lambda\sigma}|\sqrt{(\nu\sigma|\nu\sigma)} \tag{18}$$

Equation 18 may be evaluated by two matrix multiplications

$$\tilde{\mathbf{K}} = \mathbf{Q} \times |\mathbf{P}| \times \mathbf{Q} \tag{19}$$

where the Schwarz matrix **Q** is defined as

$$Q_{\mu\nu} = \sqrt{(\mu\nu|\mu\nu)} \tag{20}$$

Note that for systems with a significant HOMO–LUMO gap, the two matrix multiplications of eq 19 may be evaluated in a linear-scaling fashion using sparse algebra. However, because the step has a very small prefactor, we evaluate eq 19 using dense matrix algebra in our current preLinX implementation. In analogy to the S- and P-junction notation of ref 35, we denote the sparsity pattern of $\tilde{\mathbf{K}}$ as K-junction.

The significant basis function shells for the evaluation of eqs 7–9 are then obtained for batches of grid points according to the following algorithm, which is similar to the COSX method of Neese et al. except for the use of K-junctions instead of P-junctions:

```
for all batches of grid points do
    Determine shells $\mu$ with significant basis function value within current batch
    for all $\mu$ do
        Determine all shells $\nu$ with $\tilde{K}_{\mu\nu} \ge \vartheta_{\mathrm{pre}} \rightarrow \{\nu\}$
    end for
    Determine all shell-pairs SHP$_{\nu\lambda}$ with $\lambda \in \{\nu\}$
end for
```

For large systems with a nonzero HOMO–LUMO gap, the amount of shells in the secondary set $\{\nu\}$ is asymptotically constant because of the exponential decay of the exchange interaction. Because the amount of grid batches scales linearly and there is an asymptotically constant workload for each batch, the algorithm scales asymptotically linearly with the system size.

3.2. Batchwise Integral Screening.

So far the contributions from S-junctions (overlap of basis functions) and K-junctions (sparsity of the exchange matrix) have only been used *independently* from each other, leading to the incorporation of a vast amount of insignificant integrals $A_{\nu\lambda g}$. We therefore propose a secondary screening step that directly selects significant shell pairs for a whole batch of grid points, employing an estimate of the maximal contribution to the exchange energy of the form

$$\varepsilon^{\mathrm{cont}} = w_g F_{\nu g}A_{\nu\lambda g}F_{\lambda g} \le F_{\nu b}^{\max}A_{\nu\lambda b}^{\max}F_{\lambda b}^{\max} \tag{21}$$

with

$$F_{\nu b}^{\max} = \max_{g\in b}(w_g^{1/2}F_{\nu g}) \tag{22}$$

$$A_{\nu\lambda b}^{\max} = \max_{g\in b}(A_{\nu\lambda g}) \tag{23}$$

and b denoting the index of a batch of grid points. Instead of using the maximum integral for a whole batch $A_{\nu\lambda b}^{\max}$, we instead use the integral at the center of the batch $A_{\nu\lambda c}$, sacrificing the upper bound property of eq 21 but obtaining a simple and effective screening scheme, which can be used for both SCF and forces calculations. This simplification is possible because the integrals $A_{\nu\lambda g}$ do not vary significantly within one batch of spatially adjacent points, because the $\frac{1}{\mathbf{r}}$ distance dependence is insignificant for small batches. Moreover, for spatially large batches the exponential decay of the exchange interaction, which is included in $F_{\nu g}$, is much stronger (over orders of magnitude) than the $\frac{1}{\mathbf{r}}$-dependence of the integrals $A_{\nu\lambda g}$. Therefore, the batchwise screening does not significantly underestimate any contribution in practice.

Because the fineness of the integration grid is not supposed to influence the tightness of the integral screening, the significant contributions are determined by the condition

$$\vartheta_X \le \frac{e^{\text{cont}}}{w_{\text{ave}}} \tag{24}$$

where w_{ave} is the average grid weight of all grid points and ϑ_X denotes a given threshold. This batchwise integral screening is essential for an efficient seminumerical local hybrid implementation, reducing the number of evaluated integrals and thus the computation time for, for example, $(\text{DNA})_4$, by about 1 order of magnitude (see section 4).

3.3. Generation of Grid Batches. To allow for an efficient grid-based algorithm, the grid has to be divided into batches of spatially adjacent points. The standard meta-GGA-part, as well as the matrix multiplications of eqs 7, 9 (SCF), and 15 (forces) are evaluated using our standard DFT-grid batches of typically 5 000−20 000 points, obtained from an octree algorithm.[48] However, the integral evaluations of eq 8 (SCF) and eq 17 (forces) are performed on smaller sub-batches of 100 points, to allow for a tighter batchwise integral screening. Our preselection scheme is performed only for the larger batches, while the batchwise integral screening of eq 24 is performed in a hierarchic approach, first on the level of the large DFT batches and subsequently on the level of the sub-batches.

Because our octree algorithm leads to batches with highly varying amounts of grid points, we propose a different batching algorithm to generate the sub-batches. For that purpose we utilize a 3-dimensional Hilbert curve (see Figure 1 for a schematic representation of 2D Hilbert curves).[49−51] This curve allows for the projection of any 3D point onto a 1D line, while ensuring that points that are close to each other on the projected 1D line are also close in 3D space, which is a desirable property for our batchwise screening algorithm. In our proposed sub-batching algorithm, the grid points are thus processed into sub-batches according to their position on the Hilbert curve, i.e., their Hilbert indices.

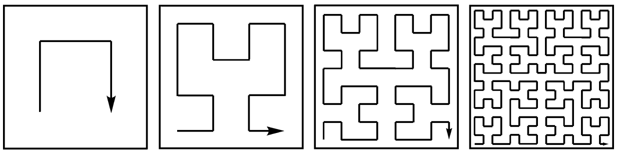


Figure 1. 2D Hilbert-curve of level $b = 1$ to $b = 4$ from left to right in ascending order.

The Hilbert indices are obtained from the algorithm of ref 50, where the level of the Hilbert curve is chosen to be $b = 21$, because the resulting index between 0 and $2^{63} - 1$ still fits into a 64-bit unsigned integer.

4. ILLUSTRATIVE CALCULATIONS

The present local hybrid scheme was implemented in our FermiONs++ program[42,43] and is tested in terms of numerical accuracy and performance. If not stated otherwise, all calculations are run on an openMP[52] parallelized multicore setup employing 12 cores (2×Intel-E5645@2.40 GHz). The code was compiled with the GNU compiler collection (GCC), version 5.2.1[53] using compiler optimizations (-O3). The *xyz*-structures of the systems employed in this work are available online at http://www.cup.lmu.de/pc/ochsenfeld/download/.[54] Throughout this work, the def2-basis sets of ref 55 have been employed.

First, the effect on the performance of both the preselection and the explicit integral screening is illustrated employing the example of linear alkanes. Figure 2 clearly demonstrates the O(N) scaling of our preLinX method with a 25× speedup compared to the conventional $O(\text{N}^2)$ implementation for the largest system ($C_{320}H_{642}$). Additionally, the prefactor is further reduced by approximately 3.5 because of the batchwise integral screening. Moreover, because our preLinX method has virtually no overhead, it is also efficient for small molecules.

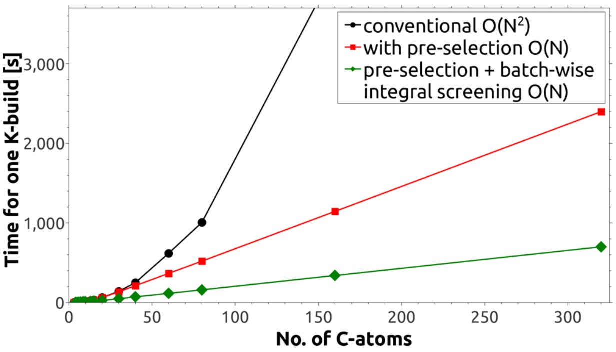


Figure 2. Wall times in seconds for one exchange−correlation−potential calculation for a series of linear alkanes on 12 CPU-cores (2×Intel-E5645@2.40 GHz) employing different screening techniques and averaged over all full SCF-cycles. The values are given for PSTS/def2-SVP, a [50/194]-grid and the screening thresholds $\vartheta_{\text{pre}} = 10^{-4}$ and $\vartheta_X = 10^{-11}$. The largest system is $C_{320}H_{642}$ containing 962 atoms.

Next, we analyze the impact of the prescreening threshold ϑ_{pre} on the accuracy and performance of our preLinX method in Table 1. For the following calculations, we choose the very tight value of $\vartheta_{\text{pre}} = 10^{-4}$ to ensure high accuracy, which is also in accordance with the recommended value of ref 42, noting that looser thresholds up to 10^{-2} may still yield very accurate results. Note that, in analogy to ref 42, ϑ_{pre} may rather be regarded as a matrix-sparsity threshold than an integral threshold, explaining why such large threshold values still yield accurate results.

After a reasonable prescreening threshold is chosen, we present the influence of the batchwise integral screening threshold ϑ_X on the accuracy and performance of SCF calculations in Table 2. The results illustrate the importance of the batchwise integral screening, with a 2−4-fold performance increase even for the tightest threshold, while introducing an error of <1 nE_h. Note that for $(\text{DNA})_4$, the column $\vartheta_X = 0$ in Table 2 represents the performance of the conventional chain-of-spheres (COSX) method. Here, neither P- nor K-junction-based preselection methods have any significant impact (see also refs 35 and 42).

Moreover, the error introduced by the integral screening can be rigorously controlled by the choice of the threshold. We further notice that the coarseness of the integration grid influences the error introduced by the screening, leading to smaller errors for coarse grids. This behavior is best explained by the amount of overestimation in eq 21: Employing coarse grids leads to spatially larger grid batches (and sub-batches), resulting in a higher overestimation of contributions in eq 21.

Table 1. Effect of the Screening Threshold ϑ_{pre} on the Computation Time and the Final SCF-Energy for Different Molecules, Basis Sets, and Integration Grids Given as Molecule/Basis-Set/Grid[a]

		ϑ_{pre}				
system/basis/grid		10^{-1}	10^{-2}	10^{-3}	10^{-4}	0
$(DNA)_4$/SVP/[50/194]	ΔE [μE_h]	−0.007	0.000	0.000	0.000	0.000
	time [s]	907	942	948	949	949
$(DNA)_4$/SVPD/[50/194]	ΔE [μE_h]	0.001	0.000	0.000	0.000	0.000
	time [s]	5754	5698	5686	5701	5826
$(DNA)_4$/TZVP/[50/194]	ΔE [μE_h]	2.524	0.000	0.000	0.000	0.000
	time [s]	3637	3616	3617	3630	3626
$(H_2O)_{142}$/SVP/[50/194]	ΔE [μE_h]	−0.019	0.000	0.000	0.000	0.000
	time [s]	770	873	910	912	916
$(DNA)_{16}$/SVP/[50/194]	ΔE [μE_h]	−0.116	0.003	0.002	−0.002	0.000
	time [s]	5035	5531	5917	6011	7098

[a] The deviations from the $\vartheta_{\mathrm{pre}} = 0$ value (ΔE) are given in μE_h. The wall times for one XC-potential calculation are given in seconds as the average over all full SCF-cycles employing 12 CPU-cores (2×Intel-E5645@2.40 GHz). For all calculations, a very tight threshold for the batchwise integral screening $\vartheta_X = 10^{-14}$ has been employed.

In essence, the batchwise screening is less tight for coarse grids. The introduced errors, however, vary over only about 1 order of magnitude, which may easily be overcome by choice of a tighter threshold. Overall, a threshold of $\vartheta_X = 10^{-11}$ represents a good compromise between accuracy (approximately 1 μE_h to 3 μE_h error) and performance (up to 10-fold performance increase) for a variety of systems, basis sets, and grids and is therefore employed as the standard for the remaining calculations.

Next, the effect of the integral threshold ϑ_X on the performance and accuracy of nuclear forces calculations is investigated (Table 3). Note that the same threshold has been employed both for the preceding SCF and the forces calculation. First, we observe that the speedup due to our batchwise screening is significantly larger (up to 19-fold for $(H_2O)_{142}$@$\vartheta_X = 10^{-11}$) as compared to XC-potential calculations. Furthermore, the standard threshold of $\vartheta_X = 10^{-11}$ yields a reasonable accuracy of approximately 1 $\mu E_h\, a_0^{-1}$ for the tested systems, which is of the same order of magnitude as the errors introduced by our CFMM method for the Coulomb matrix.[56]

In summary, our local hybrid implementation (preLinX) yields excellent performance and reliable accuracy for single-point energies and nuclear forces, when employing the thresholds $\vartheta_{\mathrm{pre}} = 10^{-4}$ and $\vartheta_X = 10^{-11}$. Overall, the gradient computation takes approximately 1.5−5 times longer than a single SCF cycle, which is still considerably less than the preceding SCF calculation, which typically requires around 10−20 SCF steps for tight convergence thresholds.

Employing the above proposed thresholds of $\vartheta_{\mathrm{pre}} = 10^{-4}$ and $\vartheta_X = 10^{-11}$, we now investigate the scaling behavior with respect to the molecular system size for a variety of molecules in Table 4. Overall, the scaling behavior is better than $O(N^2)$ for all tested systems (except for $(DNA)_1$ to $(DNA)_2$ which are too small for insignificant contributions), reaching asymptotic linear-scaling behavior for large systems like $(amy)_{64}$. Moreover, the scaling behavior of the forces computation is typically better compared to the SCF, because the integrals and integral

Table 2. Effect of the Screening Threshold ϑ_X on the Final SCF-Energy for Different Molecules, Basis Sets, and Integration Grids (Molecule/Basis-Set/Grid)[a]

		ϑ_X							
system/basis/grid		10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	0
$(DNA)_4$/SVP/[50/194]	ΔE [μE_h]	494.6	68.05	8.495	0.961	0.102	0.011	0.001	0.000
	time [s]	270	314	377	471	595	756	946	3260
$(DNA)_4$/SVP/[99/590]	ΔE [μE_h]	841.1	113.3	14.77	1.732	0.187	0.020	0.002	0.000
	time [s]	1146	1343	1656	2125	2768	3604	4630	18267
$(DNA)_4$/SVP/[10/50]	ΔE [μE_h]	143.7	18.27	2.067	0.215	0.023	0.003	0.000	0.000
	time [s]	41	45	51	59	72	85	99	233
$(DNA)_4$/SVPD/[50/194]	ΔE [μE_h]	−641.3	62.72	2.607	1.025	0.164	0.016	0.000	–
	time [s]	1434	1702	2146	2786	3610	4518	5522	–
$(DNA)_4$/TZVP/[50/194]	ΔE [μE_h]	487.0	71.36	9.406	1.013	0.116	0.012	0.000	–
	time [s]	929	1055	1261	1571	2109	2612	3339	–
$(H_2O)_{142}$/SVP/[50/194]	ΔE [μE_h]	643.3	96.21	12.28	1.437	0.161	0.017	0.002	0.000
	time [s]	348	365	405	470	572	717	912	4991

[a] The deviations (ΔE) to the tightest threshold $\vartheta_X = 10^{-14}$ or $\vartheta_X = 0$, respectively, are given in μE_h. $\vartheta_X = 0$ corresponds to no batchwise screening. Note that for the very demanding calculations with def2-SVPD and def2-TZVP no computations without batchwise screening were performed because of large computational cost. The wall times on 12 CPU-cores (2×Intel-E5645@2.40 GHz) for one XC-potential calculation is given in seconds as the average over all full SCF-cycles. The SCF energy convergence threshold dE (i.e., the change in the total energy) was set to 10^{-9} for $(DNA)_4$ and to 10^{-10} for $(H_2O)_{142}$ for the tightest screening-thresholds $\vartheta_X \leq 10^{-12}$. When less tight screening thresholds ϑ_X were employed, the SCF could not be converged so tightly. In these cases, the convergence criterion was set to be less strict, but still sufficiently tight to ensure that the convergence error is negligible compared to the error introduced by the screening.

Table 3. Effect of the Screening Threshold ϑ_X on the Computation Time and Accuracy of Nuclear Forces Calculations for Different Molecules, Basis Sets, and Integration Grids (Given as Molecule/Basis-Set/Grid)[a]

		ϑ_{Int}							
system/basis/grid		10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	0
$(DNA)_4$/SVP	max Error	5.871	1.808	0.146	0.109	0.002	0.000	0.000	0.000
/[50/194]	time [s]	385	523	738	1048	1462	1990	2624	10314
$(DNA)_4$/SVP	max Error	58.45	27.29	1.111	1.255	0.017	0.003	0.000	–
/[99/590]	time [s]	1672	2325	3382	4943	7083	9866	13258	–
$(DNA)_4$/SVP	max Error	61.175	24.987	1.297	1.521	0.059	0.085	0.087	0.000
/[10/50]	time [s]	48	63	85	114	151	196	249	695
$(DNA)_4$/SVPD	max Error	3074	274.8	47.44	6.748	0.430	0.047	0.000	–
/[50/194]	time [s]	2946	3896	5471	7744	10641	13948	17418	–
$(DNA)_4$/TZVP	max Error	451	93.2	7.56	1.720	0.124	0.018	0.000	–
/[50/194]	time [s]	1953	2818	4201	6161	8888	12219	16107	–
$(H_2O)_{142}$/SVP	max Error	7.593	1.702	0.425	0.051	0.065	0.065	0.065	0.000
/[50/194]	time [s]	392	458	581	788	1105	1559	2172	15067

[a] The maximum deviations in the molecular forces (maxError) referenced to the tightest threshold $\vartheta_X = 10^{-14}$ or $\vartheta_X = 0$, respectively, are given in $\mu E_h\, a_0^{-1}$. $\vartheta_X = 0$ corresponds to no batchwise screening. The preceding SCF calculation has been converged to the same convergence thresholds as in Table 2. The wall times for the computation of the exchange−correlation contributions to the nuclear forces employing 12 CPU-cores (2×Intel-E5645@2.40 GHz) are given in seconds.

Table 4. Scaling Behavior of the Computation Time for on XC-Potential Build (K-time) and the Computation Time for One XC-Forces Build (forces-time)[a]

fragment	atoms	K-time [s]	K-time scaling	forces-time [s]	forces scaling
		DNA Fragments			
$(DNA)_1$	62	21	–	55	–
$(DNA)_2$	128	88	2.05	218	1.91
$(DNA)_4$	260	276	1.54	619	1.40
$(DNA)_8$	524	789	1.42	1545	1.24
$(DNA)_{16}$	1052	1983	1.25	3592	1.16
		Spherical Water Clusters			
$(H_2O)_{68}$	204	80	–	155	–
$(H_2O)_{142}$	426	281	1.68	478	1.47
$(H_2O)_{285}$	855	949	1.68	1411	1.47
$(H_2O)_{569}$	1707	3040	1.60	3968	1.41
		Amylose Chains			
$(amy)_2$	45	17	–	48	–
$(amy)_4$	87	42	1.26	108	1.17
$(amy)_8$	171	100	1.21	242	1.14
$(amy)_{16}$	339	223	1.12	515	1.07
$(amy)_{32}$	675	472	1.06	1071	1.04
$(amy)_{48}$	1011	755	1.07	1645	1.03
$(amy)_{64}$	1347	1046	1.04	2278	1.04
		Fullerenes			
C_{60}	60	85	–	243	–
C_{100}	100	209	1.48	607	1.50
C_{180}	180	529	1.41	1407	1.29
C_{240}	240	821	1.16	2098	1.12

[a] In all computations, the def2-SVP basis set, the SG1 numerical integration grid,[57] and the thresholds $\vartheta_{pre} = 1 \times 10^{-4}$ and $\vartheta_X = 10^{-11}$ have been employed. The computation times employing 12 CPU-cores (2×Intel-E5645@2.40 GHz) are given in seconds, and the scaling behavior (per atom) is given compared to the respective predecessor.

derivatives, which are screened very effectively by ϑ_X, comprise a larger fraction of the total computation time.

To give some further insights into the efficiency of our seminumerical implementation, we compare its performance to analytical global hybrid calculations (employing the LinK method[40]) in Table 5. Overall, our preLinX method allows for an evaluation of local hybrid functionals at comparable cost to global hybrid functionals. Additionally, seminumerical methods scale more favorable as $O(N^2)$ with the size of the basis set compared to analytical methods, which scale as $O(N^4)$. Therefore, our seminumerical integration method is especially efficient for large basis sets, e.g., SVPD and TZVP. Moreover, the use of a coarse integration grid, i.e., the SG1-grid,[57] can considerably speed up seminumerical local hybrid calculations.

Table 5. Comparison of the Wall Times for One XC-Potential Build for Global Hybrid Functionals (Analytically) and Local Hybrid Functionals (Seminumerically) for a $(DNA)_4$ Fragment[a]

		basis set				
functional	grid	STO-3G	SV	SVP	SVPD	TZVP
PSTS (local hybrid)	[75/302]	364	642	978	5783	3291
PSTS (local hybrid)	SG1	84	177	268	1529	899
TPSSh (global hybrid)	[75/302]	56	224	511	7236	9382
TPSSh (global hybrid)	SG1	36	176	449	6814	9051

[a] The four-center-two-electron integrals for the exact-exchange part of the global hybrid functional TPSSh are evaluated analytically using our LinK method. The PSTS functional is evaluated seminumerically using the preLinX method of the present work. The fine [75/302] (about 20 000 points per atom) and the coarse SG1-grid (a pruned version of the [50/194]-grid with about 4000 points per atom) have been employed. The wall times on 12 CPU-cores (2×Intel-E5645@2.40 GHz) for one XC-potential build (including the DFT exchange−correlation part) are given in seconds as the average over all full SCF-cycles.

Finally, the parallel scaling of our local hybrid implementation is investigated in Table 6, employing the message-passing interface (MPI)[58] in combination with a fast interconnected architecture.[59] Due to the high parallel workload of the grid-

Table 6. Parallel Multinode Efficiency on Multiple Fast-Interconnected 8-Core Nodes (2×Intel-E5620@2.40 GHz)[a]

	no. of nodes	1	2	4	8	12
$(DNA)_4$/SV	time [s]	992	493	246	135	93
	par. eff. [%]	100.0	100.6	100.9	91.6	88.7
	speedup	1.00	2.01	4.03	7.33	10.64
$(DNA)_4$/SVP	time [s]	1467	743	372	210	139
	par. eff. [%]	100.0	98.7	98.4	87.3	87.8
	speedup	1.00	1.97	3.94	6.99	10.53
$(DNA)_4$/SVPD	time [s]	8810	4551	2270	1236	853
	par. eff. [%]	100.0	96.8	97.0	89.1	86.1
	speedup	1.00	1.94	3.88	7.13	10.33
$(DNA)_{16}$/SVP	time [s]	–	–	–	–	1034

[a]The wall times for one XC-potential calculation on multiple nodes for different basis sets and employing the [75/302] integration grid are given in seconds as the average over all full SCF cycles.

based implementation, even on 12 computing nodes with a total of 96 CPU cores, a high parallel efficiency of >86% is observed. Computing one SCF step for a $(DNA)_4$ fragment (260 atoms) with a fine integration grid (20 000 points per atom) and the def2-SVP basis in 2 min is very promising with regard to the application of local hybrid functionals for production calculations. Finally, we notice that our implementation allows for the calculation of large systems like $(DNA)_{16}$, comprising 1052 atoms, 11 230 basis functions, and 20.8 million grid points, requiring less than 20 min for a single SCF cycle.

5. CONCLUSION AND OUTLOOK

We presented an efficient and linear-scaling seminumerical algorithm for local hybrid functionals, denoted as preLinX. A high computational efficiency was conceived using preselection and batchwise integral screening. The computational cost of our seminumerical algorithm is comparable to that of analytical global hybrid DFT calculations. Moreover, because of the asymptotic linear-scaling behavior, our method allows for the evaluation of local hybrid functionals for large systems, comprising more than 1000 atoms. Additionally, we achieve high parallel efficiency on multiple computing nodes within a massively parallel setup.

In future work we plan to transfer the preLinX algorithm to graphic processing units to further accelerate local hybrid DFT calculations. We further note that the herein presented preLinX method may also be used to speed up global-hybrid functional calculations. Furthermore, an efficient seminumerical algorithm for response properties (e.g., vibrational spectra or excited states within TD-DFT) similar to the seminumerical TD-DFT method of Kaupp et al.[26] is crucial for mainstream applications and will also be investigated in future work. Finally, we hope that the availability of an efficient linear-scaling method to evaluate local hybrid functionals will encourage further developments in this field.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: christian.ochsenfeld@uni-muenchen.de.

ORCID

Christian Ochsenfeld: 0000-0002-4189-6558

Funding

The authors acknowledge the financial support by the *Deutsche Forschungsgemeinschaft* (DFG) in the *SFB* 749 "Dynamik und Intermediate molekularer Transformationen" and the DFG *cluster of excellence* (EXC114) "Center for Integrative Protein Science Munich" (CIPSM). C.O. acknowledges in addition financial support as a Max-Planck-Fellow at the MPI-FKF (Stuttgart).

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133−A1138.

(2) Becke, A. D. A new mixing of Hartree-Fock and local density-functional theories. *J. Chem. Phys.* **1993**, *98*, 1372−1377.

(3) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98*, 11623−7.

(4) Adamo, C.; Barone, V. Toward reliable density functional methods without adjustable parameters: the PBE0 model. *J. Chem. Phys.* **1999**, *110*, 6158−6170.

(5) Perdew, J. P.; Kurth, S.; Zupan, A.; Blaha, P. Accurate Density Functional with Correct Formal Properties: A Step Beyond the Generalized Gradient Approximation. *Phys. Rev. Lett.* **1999**, *82*, 2544−2547.

(6) Staroverov, V. N.; Scuseria, G. E.; Tao, J.; Perdew, J. P. Comparative assessment of a new nonempirical density functional: Molecules and hydrogen-bonded complexes. *J. Chem. Phys.* **2003**, *119*, 12129−12137.

(7) Jaramillo, J.; Scuseria, G. E.; Ernzerhof, M. Local hybrid functionals. *J. Chem. Phys.* **2003**, *118*, 1068−1073.

(8) Karasiev, V. V. Local "hybrid" functionals based on exact-expression approximate exchange. *J. Chem. Phys.* **2003**, *118*, 8576−8583.

(9) Becke, A. D. Real-space post-Hartree-Fock correlation models. *J. Chem. Phys.* **2005**, *122*, 064101.

(10) Becke, A. D.; Johnson, E. R. A unified density-functional treatment of dynamical, nondynamical, and dispersion correlations. *J. Chem. Phys.* **2007**, *127*, 124108.

(11) Kaupp, M.; Bahmann, H.; Arbuznikov, A. V. Local hybrid functionals: An assessment for thermochemical kinetics. *J. Chem. Phys.* **2007**, *127*, 194102.

(12) Janesko, B. G.; Scuseria, G. E. Local hybrid functionals based on density matrix products. *J. Chem. Phys.* **2007**, *127*, 164117.

(13) Perdew, J. P.; Ruzsinszky, A.; Csonka, G. I.; Vydrov, O. A.; Scuseria, G. E.; Staroverov, V. N.; Tao, J. Exchange and correlation in open systems of fluctuating electron number. *Phys. Rev. A: At., Mol., Opt. Phys.* **2007**, *76*, 040501.

(14) Bahmann, H.; Rodenberg, A.; Arbuznikov, A. V.; Kaupp, M. A thermochemically competitive local hybrid functional without gradient corrections. *J. Chem. Phys.* **2007**, *126*, 011103.

(15) Janesko, B. G.; Scuseria, G. E. Parameterized local hybrid functionals from density-matrix similarity metrics. *J. Chem. Phys.* **2008**, *128*, 084111.

(16) Becke, A. D. Density functionals for static, dynamical, and strong correlation. *J. Chem. Phys.* **2013**, *138*, 074109.

(17) Johnson, E. R. A density functional for strong correlation in atoms. *J. Chem. Phys.* **2013**, *139*, 074110.

(18) Johnson, E. R. Local-hybrid functional based on the correlation length. *J. Chem. Phys.* **2014**, *141*, 124120.

(19) Kong, J.; Proynov, E. Density Functional Model for Non-dynamic and Strong Correlation. *J. Chem. Theory Comput.* **2016**, *12*, 133−143.

(20) Arbuznikov, A. V.; Kaupp, M.; Bahmann, H. From local hybrid functionals to ”localized local hybrid” potentials: Formalism and thermochemical tests. *J. Chem. Phys.* **2006**, *124*, 204102.

(21) Janesko, B. G.; Krukau, A. V.; Scuseria, G. E. Self-consistent generalized Kohn-Sham local hybrid functionals of screened exchange: Combining local and range-separated hybridization. *J. Chem. Phys.* **2008**, *129*, 124110.

(22) Proynov, E.; Shao, Y.; Kong, J. Efficient self-consistent DFT calculation of nondynamic correlation based on the B05 method. *Chem. Phys. Lett.* **2010**, *493*, 381−385.

(23) Proynov, E.; Liu, F.; Shao, Y.; Kong, J. Improved self-consistent and resolution-of-identity approximated Becke'05 density functional model of nondynamic electron correlation. *J. Chem. Phys.* **2012**, *136*, 034102.

(24) Liu, F.; Proynov, E.; Yu, J.-G.; Furlani, T. R.; Kong, J. Comparison of the performance of exact-exchange-based density functional methods. *J. Chem. Phys.* **2012**, *137*, 114104.

(25) Bahmann, H.; Kaupp, M. Efficient Self-Consistent Implementation of Local Hybrid Functionals. *J. Chem. Theory Comput.* **2015**, *11*, 1540−1548.

(26) Maier, T. M.; Bahmann, H.; Kaupp, M. Efficient Semi-numerical Implementation of Global and Local Hybrid Functionals for Time-Dependent Density Functional Theory. *J. Chem. Theory Comput.* **2015**, *11*, 4226−4237.

(27) Klawohn, S.; Bahmann, H.; Kaupp, M. Implementation of Molecular Gradients for Local Hybrid Density Functionals Using Seminumerical Integration Techniques. *J. Chem. Theory Comput.* **2016**, *12*, 4254−4262.

(28) Liu, F.; Kong, J. Efficient Computation of Exchange Energy Density with Gaussian Basis Functions. *J. Chem. Theory Comput.* **2017**, *13*, 2571−2580.

(29) Friesner, R. A. Solution of self-consistent field electronic structure equations by a pseudospectral method. *Chem. Phys. Lett.* **1985**, *116*, 39−43.

(30) Friesner, R. A. Solution of the Hartree-Fock equations by a pseudospectral method: application to diatomic molecules. *J. Chem. Phys.* **1986**, *85*, 1462−1468.

(31) Friesner, R. A. Solution of the Hartree-Fock equations for polyatomic molecules by a pseudospectral method. *J. Chem. Phys.* **1987**, *86*, 3522−3531.

(32) Ringnalda, M. N.; Belhadj, M.; Friesner, R. A. Pseudospectral Hartree-Fock theory: applications and algorithmic improvements. *J. Chem. Phys.* **1990**, *93*, 3397−3407.

(33) Murphy, R. B.; Cao, Y.; Beachy, M. D.; Ringnalda, M. N.; Friesner, R. A. Efficient pseudospectral methods for density functional calculations. *J. Chem. Phys.* **2000**, *112*, 10131−10141.

(34) Bochevarov, A. D.; Harder, E.; Hughes, T. F.; Greenwood, J. R.; Braden, D. A.; Philipp, D. M.; Rinaldo, D.; Halls, M. D.; Zhang, J.; Friesner, R. A. Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences. *Int. J. Quantum Chem.* **2013**, *113*, 2110−2142.

(35) Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. Efficient, approximate and parallel Hartree-Fock and hybrid DFT calculations. A 'chain-of-spheres' algorithm for the Hartree-Fock exchange. *Chem. Phys.* **2009**, *356*, 98−109.

(36) Plessow, P.; Weigend, F. Seminumerical calculation of the Hartree-Fock exchange matrix: Application to two-component procedures and efficient evaluation of local hybrid density functionals. *J. Comput. Chem.* **2012**, *33*, 810−816.

(37) Baldes, A.; Weigend, F. Efficient two-component self-consistent field procedures and gradients: implementation in TURBOMOLE and application to Au_{20}^{-}. *Mol. Phys.* **2013**, *111*, 2617−2624.

(38) Schwegler, E.; Challacombe, M. Linear scaling computation of the Hartree-Fock exchange matrix. *J. Chem. Phys.* **1996**, *105*, 2726−2734.

(39) Schwegler, E.; Challacombe, M.; Head-Gordon, M. Linear scaling computation of the Fock matrix. II. Rigorous bounds on exchange integrals and incremental Fock build. *J. Chem. Phys.* **1997**, *106*, 9708−9717.

(40) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. Linear and sublinear scaling formation of Hartree-Fock-type exchange matrices. *J. Chem. Phys.* **1998**, *109*, 1663−1669.

(41) Ochsenfeld, C. Linear scaling exchange gradients for Hartree-Fock and hybrid density functional theory. *Chem. Phys. Lett.* **2000**, *327*, 216−223.

(42) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(43) 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*, 918−22.

(44) Pulay, P. Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules. I. Theory. *Mol. Phys.* **1969**, *17*, 197−204.

(45) Perdew, J. P.; Staroverov, V. N.; Tao, J.; Scuseria, G. E. Density functional with full exact exchange, balanced nonlocality of correlation, and constraint satisfaction. *Phys. Rev. A: At., Mol., Opt. Phys.* **2008**, *78*, 052513.

(46) Obara, S.; Saika, A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1986**, *84*, 3963−74.

(47) Obara, S.; Saika, A. General recurrence formulas for molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1988**, *89*, 1540−1559.

(48) Burow, A. M.; Sierka, M. Linear Scaling Hierarchical Integration Scheme for the Exchange-Correlation Term in Molecular and Periodic Systems. *J. Chem. Theory Comput.* **2011**, *7*, 3097−3104.

(49) Hilbert, D. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.* **1891**, *38*, 459−460.

(50) Skilling, J. Programming the Hilbert curve. *AIP Conf. Proc.* **2004**, *707*, 381−387.

(51) Skilling, J. Using the Hilbert curve. *AIP Conf. Proc.* **2004**, *707*, 388−405.

(52) OpenMP library, version 4.0; http://www.openmp.org.

(53) GNU compiler collection, version 5.2.1; http://gcc.gnu.org.

(54) Maurer, S. A.; Lambrecht, D. S.; Flaig, D.; Ochsenfeld, C. Distance-dependent Schwarz-based integral estimates for two-electron integrals: Reliable tightness vs. rigorous upper bounds. *J. Chem. Phys.* **2012**, *136*, 144107.

(55) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.

(56) White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. The continuous fast multipole method. *Chem. Phys. Lett.* **1994**, *230*, 8−16.

(57) Gill, P. M. W.; Johnson, B. G.; Pople, J. A. A standard grid for density functional calculations. *Chem. Phys. Lett.* **1993**, *209*, 506−512.

(58) Message Passing Interface (MPI), version 1.8.1; http://www.mcs.anl.gov/research/projects/mpi.

(59) Infiniband; www.openfabrics.org. MVAPICH2 library, version 2.2; http://mvapich.cse.ohio-state.edu.

3.2 Publication II: Highly Efficient, Linear-Scaling Seminumerical Exact-Eachange Method for Graphic Processing Units

H. Laqua, T. H. Thompson, J. Kussmann, C. Ochsenfeld

J. Chem. Theory Comput. **14**, 3451 (2018).

Abstract

We present a highly efficient and asymptotically linear-scaling graphic processing unit accelerated seminumerical exact-exchange method (sn-LinK). We go beyond our previous central processing unit-based method (Laqua, H.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Theory Comput.* **2018**, *14*, 3451–3458) by employing our recently developed integral bounds (Thompson, T. H.; Ochsenfeld, C. J. Chem. Phys. **2019**, *150*, 044101) and high-accuracy numerical integration grid (Laqua, H.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2018**, *149*, 204111). The accuracy is assessed for several established test sets, providing errors significantly below 1 mE_h for the smallest grid. Moreover, a comprehensive performance analysis for large molecules between 62 and 1347 atoms is provided, revealing the outstanding performance of our method, in particular, for large basis sets such as the polarized quadruple-zeta level with diffuse functions.

Reprinted with permission from:

H. Laqua, T. H. Thompson, J. Kussmann, C. Ochsenfeld
“Highly Efficient, Linear-Scaling Seminumerical Exact-Eachange Method for Graphic Processing Units”
J. Chem. Theory Comput. **16**, 1456 (2020).

https://pubs.acs.org/doi/10.1021/acs.jctc.9b00860

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

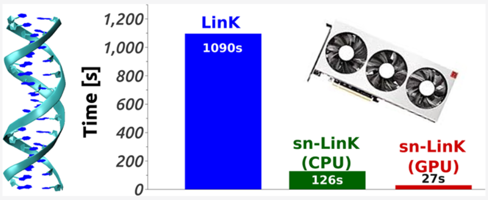
Highly Efficient, Linear-Scaling Seminumerical Exact-Exchange Method for Graphic Processing Units

Henryk Laqua, Travis H. Thompson, Jörg Kussmann, and Christian Ochsenfeld*

Cite This: J. Chem. Theory Comput. 2020, 16, 1456−1468 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We present a highly efficient and asymptotically linear-scaling graphic processing unit accelerated seminumerical exact-exchange method (sn-LinK). We go beyond our previous central processing unit-based method (Laqua, H.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Theory Comput.* **2018**, *14,* 3451−3458) by employing our recently developed integral bounds (Thompson, T. H.; Ochsenfeld, C. *J. Chem. Phys.* **2019**, *150,* 044101) and high-accuracy numerical integration grid (Laqua, H.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2018**, *149,* 204111). The accuracy is assessed for several established test sets, providing errors significantly below 1mE_h for the smallest grid. Moreover, a comprehensive performance analysis for large molecules between 62 and 1347 atoms is provided, revealing the outstanding performance of our method, in particular, for large basis sets such as the polarized quadruple-zeta level with diffuse functions.



1. INTRODUCTION

During the last 15 years, graphic processing units (GPUs) have gained increasing interest within the quantum chemistry community, focusing, in particular, on the evaluation of 4-center-2-electron (4c-2e) integrals, which represent the major bottleneck in most Hartree−Fock and Kohn−Sham calculations.[1−15] Since, for Hartree−Fock (HF) and hybrid density functional theory (DFT) calculations, the mandatory computation of exact (Fock-like) exchange matrices is particularly expensive, efficient and linear-scaling implementations have been developed since the late 1990s[16−21] including recent developments.[7,9] However, for larger molecules and particularly in combination with larger basis sets (i.e., triple-ζ or larger), resolution-of-the-identity (RI)[22,23] or seminumerical methods, that is, grid-based methods employing 3-center-1-electron (3c-1e) integrals,[24−46] are possibly more efficient due to their superior $O(N_{\text{bas}}^2)$ formal scaling compared to the formal $O(N_{\text{bas}}^4)$ scaling of the conventional 4c-2e integral-based methods.

As we demonstrated recently,[44] seminumerical exchange methods can, in contrast to the asymptotically $O(M^3)$ scaling RI-K method,[22,23] be implemented in an asymptotically linear-scaling fashion. This is an increasingly important property since modern computer hardware now allows for the routine calculation of multiple thousand atoms on conventional server nodes or workstations. In addition, seminumerical methods may directly be employed to compute the exact-exchange part of local hybrid functionals, which represent a very promising new class of functionals due to their higher variability and therefore more general applicability.[47−58] Indeed, the prospects of these new functionals have been the major motivation for many recently developed seminumerical methods.[38−40,42−46]

In this publication, we present a reformulation of our previous method[44] that allows for an efficient and highly performant GPU implementation. These changes include the use of our recently developed generalized integral bounds[59] and our improved molecular grids.[60−62] Not only were these new techniques necessary for a performant GPU implementation but they are also applied to our existing central processing unit (CPU) implementation, in this way, further improving its performance as well, especially if run on modern CPUs, which provide an ever-increasing support for single-instruction-multiple-data (SIMD) vector instructions. Particularly, the batch-wise integral selection, which we pioneered in our previous work[44] and refined in this work, is essential for a highly efficient and performant implementation on SIMD computer architectures, such as GPUs and modern CPUs. That is, our new method exploits the superior computing performance of SIMD computer architectures while maintaining the asymptotic linear-scaling behavior of our previous work.

The paper is organized as follows: we begin with a brief review of the theory underlying the seminumerical method in Section 2.1, followed by the description of our revised integral screening in Sections 2.2 and 2.3, and our newly developed prescreening method in Section 2.4. Subsequently, we provide an outline of our GPU implementation for the Compute Unified Device Architecture (CUDA)[63] and the Open Computing Language (OpenCL)[64,65] frameworks in Section

Received: August 28, 2019
Published: February 13, 2020

https://dx.doi.org/10.1021/acs.jctc.9b00860

3, focusing on the particular techniques employed to maximize performance. Finally, we assess our new accelerated seminumerical exchange method, denoted as sn-LinK, in terms of the accuracy of the numerical integration in Section 5.1 and in terms of performance in Sections 5.2.1 to 5.2.5. For simplicity, we restrict the discussion within this paper to the computation of Fock exchange within the Hartree–Fock theory, noting that the application to global and local hybrid DFT is straightforward.

2. THEORY

2.1. Seminumerical Exchange Matrix.

The exchange matrix is given in the atomic orbital (AO) basis as

$$K_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu\sigma|\nu\lambda) \tag{1}$$

where μ, ν, λ, and σ represent AO basis function indices, and the 4-center-2-electron integral $(\mu\sigma \mid \nu\lambda)$ is defined as

$$(\mu\sigma|\nu\lambda) = \int\int \chi_\mu(\mathbf{r}_1)\chi_\sigma(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\chi_\nu(\mathbf{r}_2)\chi_\lambda(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{2}$$

Within the seminumerical ansatz, the integration over one of the coordinates ($\mathbf{r}_1$ or $\mathbf{r}_2$) is performed analytically, and the other one is performed numerically by employing discrete grid points with coordinates $\mathbf{r}_g$ and weights w_g. To preserve all the symmetries within the integral tensor, this decomposition is performed symmetrically over both coordinates $\mathbf{r}_1$ and $\mathbf{r}_2$, leading to

$$\begin{aligned}(\mu\sigma|\nu\lambda) \approx \frac{1}{2}\Bigg[&\sum_g w_g \chi_\mu(\mathbf{r}_g)\chi_\sigma(\mathbf{r}_g)\left(\int \frac{\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}d\mathbf{r}\right)\\ &+ \sum_g w_g\left(\int \frac{\chi_\mu(\mathbf{r})\chi_\sigma(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}d\mathbf{r}\right)\chi_\nu(\mathbf{r}_g)\chi_\lambda(\mathbf{r}_g)\Bigg]\end{aligned} \tag{3}$$

Inserting eq 3 into the definition of the exchange matrix (eq 1) yields

$$K_{\mu\nu} \approx \frac{1}{2}\left[\sum_g w_g \sum_{\lambda\sigma} \chi_\mu(\mathbf{r}_g)\left(\int \frac{\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|}d\mathbf{r}\right)P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) + \text{transpose}\right] \tag{4}$$

where the transpose is the result of the symmetric integral decomposition in eq 3 in combination with the symmetry of the ground-state density matrix $P_{\mu\nu} = P_{\nu\mu}$.

The exchange matrix may thus be computed in three consecutive steps

$$F_{\lambda g} = \sum_\sigma \chi_\sigma(\mathbf{r}_g)P_{\lambda\sigma} \tag{5}$$

$$G_{\nu g} = \sum_\lambda w_g A_{\nu\lambda g}F_{\lambda g} \tag{6}$$

$$K_{\mu\nu} = \sum_g \chi_\mu(\mathbf{r}_g)G_{\nu g} \tag{7}$$

where $A_{\nu\lambda g}$ denotes 3-center-1-electron integrals of the form

$$A_{\nu\lambda g} = \int \frac{\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}d\mathbf{r} \tag{8}$$

The integrals $A_{\nu\lambda g}$ are evaluated on-the-fly using optimized automatically generated Obara–Saika[66,67] recursions for the different l-quantum number combinations. The so-obtained exchange matrix K is subsequently symmetrized as

$$K_{\mu\nu}^{\text{symm}} = \frac{1}{2}(K_{\mu\nu} + K_{\nu\mu}) \tag{9}$$

to account for the transpose in eq 4.

2.2. Integral Screening.

The integral screening of our previous method[44] targeted only the exchange energy, that is, significant integrals were selected solely based on their contribution to the exchange energy

$$\begin{aligned}\varepsilon_{\nu\lambda g}^{E} &= \left| w_g \sum_{\mu\sigma} \chi_\mu(\mathbf{r}_g)P_{\mu\nu}A_{\nu\lambda g}P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g)\right|\\ &= |w_g^{1/2}F_{\nu g}A_{\nu\lambda g}w_g^{1/2}F_{\lambda g}|\end{aligned} \tag{10}$$

This is a simple and symmetrical expression, which provides the tightest screening possible if one is only interested in energies.

However, during the self-consistent field (SCF) iterations, a high accuracy in the exchange potential matrix is also desirable, in particular if larger basis sets are employed. In analogy to the conventional (4c-2e integral-based) LinK method,[20] we revised our scheme to also include contributions to the exchange matrix K into the screening

$$\begin{aligned}\varepsilon_{\nu\lambda g}^{K} &= |w_g| \max\left(\sum_{\mu\sigma} |\chi_\mu(\mathbf{r}_g)||P_{\mu\nu}||A_{\nu\lambda g}||\chi_\sigma(\mathbf{r}_g)|,\right.\\ &\qquad\left.\sum_{\mu\sigma} |\chi_\mu(\mathbf{r}_g)||A_{\nu\lambda g}||P_{\lambda\sigma}||\chi_\sigma(\mathbf{r}_g)|\right)\\ &\leq |w_g| \max(|F_{\nu g}|,|F_{\lambda g}|)|A_{\nu\lambda g}|\sum_\mu |\chi_\mu(\mathbf{r}_g)|\end{aligned} \tag{11}$$

In our new implementation, a 3c-1e integral $A_{\nu\lambda g}$ is labeled as significant if it is significant by either eq 10 or 11, that is

$$\varepsilon_{\nu\lambda g}^{E} \geq \vartheta_E \vee \varepsilon_{\nu\lambda g}^{K} \geq \vartheta_K \tag{12}$$

employing two different thresholds ϑ_E and ϑ_K for each criterion. Since, during the SCF, both the exchange matrix and the exchange energy are of interest, we employ both eqs 10 and 11 for the screening during the SCF, whereas for the final energy calculation (which is typically performed on a larger grid), we screen only for the energy (eq 10). A similar optimization was also described in ref 36.

In order for the screening to be efficient, not every integral is inspected individually; instead, a whole batch of spatially adjacent grid points is considered at once, which reduces the screening overhead to an insignificant amount (<5% of the total cost of the integral evaluation). For this purpose, the maximum contribution within a whole batch b of points has to be estimated, that is

$$\begin{aligned}\varepsilon_{\nu\lambda b}^{E} &= \max_{g\in b}(\varepsilon_{\nu\lambda g}^{E}) \\ &= \max_{g\in b}(|w_g^{1/2}F_{\nu g}|)\max_{g\in b}(|w_g^{1/2}F_{\lambda g}|)\max_{g\in b}(|A_{\nu\lambda g}|)\end{aligned} \tag{13}$$

and

$$\begin{aligned}\varepsilon_{\nu\lambda b}^{K} &= \max_{g\in b}(\varepsilon_{\nu\lambda g}^{K}) \\ &= \max(\max_{g\in b}(|w_g^{1/2}F_{\nu g}|),\ \max_{g\in b}(|w_g^{1/2}F_{\lambda g}|)) \\ &\quad \max_{g\in b}\left(|w_g^{1/2}|\sum_{\mu}|\chi_{\mu}(\mathbf{r}_g)|\right)\max_{g\in b}(|A_{\nu\lambda g}|)\end{aligned} \tag{14}$$

The necessary quantities $\max_{g\in b}(|w_g^{1/2}F_{\nu g}|)$ and $\max_{g\in b}\left(|w_g^{1/2}|\sum_{\mu}|\chi_{\mu}(\mathbf{r}_g)|\right)$ are trivially precomputed from F and $\chi_{\mu}(\mathbf{r}_g)$, and a rigorous upper bound for the integral $\max_{g\in b}(|A_{\nu\lambda g}|)$ is obtained as a special case of our recently developed partition bounds for many classes of electronic integrals.[59] This bound is briefly described below.

2.3. Integral Bounds for 3c-1e Integrals.

In ref 59, two different rigorous bounds result for $|A_{\nu\lambda g}|$. A distance-dependent bound that captures the Coulomb decay of far away grid points can be formulated by calculating rigorous centers and extents for each shell pair. The resulting bounds are very tight but also necessarily batch-dependent.

A simpler bound that is independent of $\mathbf{r}_g$, and therefore also batch-independent, is given by

$$\max_{g\in b}(|A_{\nu\lambda g}|) \leq \max_{\mathbf{r}_g\in\mathbb{R}^3}\left(\int\frac{|\chi_{\nu}(\mathbf{r})\chi_{\lambda}(\mathbf{r})|}{|\mathbf{r}_g - \mathbf{r}|}d\mathbf{r}\right) = \mathcal{V}_{\nu\lambda} \tag{15}$$

The batch independency of 15 allows to decrease the complexity of the screening algorithm because it only relies on precomputable shell-pair quantities. This is particularly useful for a high-performance GPU implementation, where algorithm complexity should be kept to a minimum. The equation necessary to compute $\mathcal{V}_{\nu\lambda}$ is given in Appendix A of ref 59. In short, we bound $\mathcal{V}_{\nu\lambda}$ by simpler integrals over spherically symmetric functions and use the fact that for any function S that is spherically symmetric with respect to a point $\mathbf{p}$, one can show that

$$\max_{\mathbf{r}_g\in\mathbb{R}^3}\left(\int\frac{|S(\mathbf{r})|}{|\mathbf{r}_g - \mathbf{r}|}d\mathbf{r}\right) = \int\frac{|S(\mathbf{r})|}{|\mathbf{p} - \mathbf{r}|}d\mathbf{r} \tag{16}$$

i.e., the maximum is always achieved at the spherical center.

In contrast to our previous screening scheme,[44] our new screening employing (eq 15) is completely rigorous and requires only a few multiplications for each grid batch and shell pair instead of a full integral evaluation, which better suits the parallel architecture of GPUs. In our sn-LinK method, this screening is performed in a hierarchical way, that is, a set of significant shell pairs is first selected for a large batch (typically around 10,000 to 20,000 grid points) on the CPU and subsequently a tighter selection for sub-batches of 64 grid points is made on-the-fly on the GPU.

This batch-wise integral selection is essential for both an efficient CPU and, to even greater extent, an efficient GPU implementation since it does not interfere with single-instruction-multiple-data (SIMD) vector instructions because identical branching within one sub-batch is guaranteed.

2.4. Prescreening.

Having determined a tight screening for 3c-1e integrals (i.e., the shell pairs $\nu\lambda$ involved in eq 6), we now consider the screening of the set of indices μ/σ and ν/λ to also guarantee the asymptotic linear-scaling evaluation of eqs 5 and 7.

For each given batch of grid points, the set containing all the significant basis function indices μ is identical to the set containing the indices σ. These sets, which we refer to as $\{\mu\}$, are determined solely by the extent of the AO basis functions, that is, only functions χ_{μ} with a significant basis function value within a given batch are labeled significant. Due to the exponential decay of Gaussian-type AO basis functions, the size of the set $\{\mu\}$ is asymptotically constant for any given grid batch.

Analogously, the sets of significant basis function indices ν are also identical to the sets containing the indices λ. However, these sets (denoted as $\{\nu\}$) cannot be determined by the basis function extents since they couple indirectly via $\{\mu\}$ and the density matrix element $P_{\mu\nu}$ or, equivalently, by the extent of the exchange hole. Therefore, depending on the electronic structure of the system, a variable amount of basis function shells need to be considered in $\{\nu\}$.

For this preselection, we simply select all the significant shell pairs $\nu\lambda$ by eq 13 or 14 and subsequently select the set of all shells that contribute to at least one of these shell pairs since all other shells do not contribute to the exchange matrix at all because all of their contributions would be screened out by the integral screening later anyhow. This method thus introduces no additional error and is sufficient to ensure asymptotic linear scaling (and constant memory scaling) since only a constant amount of shell pairs $\nu\lambda$ are significant for each batch. That is, all batch-wise quantities are of asymptotically constant size, resulting in a constant workload per batch and therefore in an overall linear scaling since the amount of grid batches scales linearly with the system size.

One complication arises from the fact that the intermediate quantity $\boldsymbol{F}$ is required for the preselection, while the set $\{\nu\}$ needs to be known prior to the computation of $\boldsymbol{F}$, since $\{\nu\}$ is just the set of significant entries in $\boldsymbol{F}$. Therefore, an upper bound for the absolute value of $\boldsymbol{F}$ that can be computed at low cost prior to the computation of $\boldsymbol{F}$ is required. For this quantity, we choose the batch-wise maximum of $\boldsymbol{F}$, that is

$$\begin{aligned}\max_{g\in b}(|w_g^{1/2}\boldsymbol{F}_{\lambda g}|) &= \max_{g\in b}\left(\left|\sum_{\sigma}P_{\lambda\sigma}w_g^{1/2}\chi_{\sigma}(\mathbf{r}_g)\right|\right) \\ &\leq \sum_{\sigma}|P_{\lambda\sigma}|\max_{g\in b}(|w_g^{1/2}\chi_{\sigma}(\mathbf{r}_g)|)\end{aligned} \tag{17}$$

Therefore, an upper bound for $\boldsymbol{F}$ can be obtained by only one matrix-vector multiplication of $|\boldsymbol{P}|$ with $\max_{g\in b}(|w_g^{1/2}\chi_{\sigma}(\mathbf{r}_g)|)$ for each batch.

3. IMPLEMENTATION

We implemented the above described sn-LinK method within our C++-based FermiONs++ program,[7,9] revising our CPU-based local hybrid implementation described in ref 44. Our implementation for the AMD GPUs is based on OpenCL,[64] whereas our NVIDIA GPU implementation employs CUDA[63] since we found OpenCL to be less performant on NVIDIA GPUs. All steps are performed exclusively with double precision (fp64) to obtain reliable results and to allow for tight convergence even with large basis sets.

In contrast to the analytical integral-direct method, which consists of only one compute-intensive step, the seminumerical implementation contains multiple bottlenecks, that is, the three steps of eqs 5−7 and the evaluation of the basis function values $\chi_\mu(\mathbf{r}_g)$. Therefore, all these steps have to be performed on the GPU to minimize bottlenecks from the CPU and the CPU-GPU data transfer. Additionally, we decided to use multiple concurrent streams of instructions on each GPU, which allows for the data transfer of one stream to be performed concurrently with GPU kernel execution of another stream, maximizing the utilization of the available hardware in this way.

3.1. GPU Implementation.

The sn-LinK algorithm operates on grid batches of typically 10 000 to 20 000 grid points on GPUs and typically 512 points per batch on CPUs. We found 256 points per AMD compute unit (CU) or NVIDIA streaming multiprocessor (SM) to be optimal, totaling 15,360 points for the AMD Radeon VII GPU and 20,480 points for the NVIDIA GV100. In contrast to our previous implementation,[44] we adapted our Hilbert curve-based sub-batching scheme (see Section 3.3 of ref 44) to also generate the large grid batches. The main advantage of the new approach is the fixed size of grid points per batch, that is, every grid batch except the very last one contains exactly the same amount of points, which ensures optimal utilization of the parallel compute capabilities of the GPUs.

Algorithm 1 sn-LinK GPU implementation

```
for all batches b do                                              ▷ openMP parallel
    perform pre-screening (Section 2.4)                                       ▷ CPU
end for
Allocate GPU memory
for all batches b do                       ▷ openMP/multi-GPU/multi-stream parallel
    copy grid data and significant shell-data to GPU
    compute basis function values $\chi_\mu(\mathbf{r}_g)$                    ▷ GPU
    get batch-local density matrix $\mathbf{P}$                               ▷ CPU
    copy batch-local $\mathbf{P}$ to GPU
    evaluate eq. (5) (BLAS-3)                                                 ▷ GPU
    collect data for all shellpairs significant by eqs. (13) and (14)         ▷ CPU
    copy shell-pair-data to GPU
    evaluate eq. (6) (integral evaluation)                                    ▷ GPU
    evaluate eq. (7) (BLAS-3)                                                 ▷ GPU
    copy batch-local exchange matrix $\mathbf{K}$ to CPU
    add batch-local $\mathbf{K}$ to global $\mathbf{K}$                       ▷ CPU
end for
```

In our GPU implementation (see Algorithm 1), we primarily parallelize over these large grid batches, employing multiple parallel host threads, each of which maps to one device stream, which we implemented using CUDA streams and OpenCL command queues. For maximum performance, we found two or three parallel streams per device to be optimal (see also Section 5.2.3), allowing for concurrent CPU execution, GPU execution, and CPU-GPU data transfer, maximizing hardware utilization in this way. This strategy requires the pre-allocation of GPU memory since allocation of device memory forces stream synchronization.

For a small system (up to ∼200 atoms), the evaluation of the 3c-1e integrals in eq 6 is by far the slowest step, amounting to over 90% of the computation time, whereas for larger systems, the matrix multiplications of eqs 5 and 7 become comparatively more expensive, for example, for the system over 1000 atoms, the integral evaluation amounts to less than 50% of the total computation time. In contrast to the Intel Xeon Phi implementation presented in ref 43, we therefore decided to implement all four compute-intensive steps, that is, the computation of the basis functions $\chi_\mu(\mathbf{r})$ and the evaluation of eqs 5−7 on the GPU, thereby also reducing the amount of CPU-GPU memory transfer.

To achieve asymptotic linear scaling of the implementation while still utilizing the high performance of dense matrix algebra routines provided by basic linear algebra subroutines (BLAS-3) libraries (i.e., Intel MKL for CPUs, cuBLAS for NVIDIA GPUs, and clBLAS for AMD GPUs), we employ dense batch-local submatrices of asymptotically constant size for $\boldsymbol{P}$ and $\boldsymbol{K}$, containing only entries for the significant basis functions within the current batch, determined by the preselection algorithm outlined in Section 2.4, thereby also guaranteeing asymptotically constant GPU memory requirements.

3.2. Implementation of the 3c-1e Integrals.

The prescreening of Section 2.4 also provides an asymptotically constant-sized set of shell pairs for each batch, which is further refined on the CPU using the integral selection methods described in Sections 2.2 and 2.3. The shell-pair data is then copied to the GPU, where all the significant 3c-1e integrals $A_{\nu\lambda g}$ for the respective batch are subsequently computed and directly multiplied with $F_{\lambda g}$ to form $G_{\nu g}$ according to eq 6 (see Algorithm 2), performing on-the-fly integral screening on the sub-batch level.

The performance of GPUs relies heavily on single-instruction-mutliple-data (SIMD) vector operations, that is, 32 or 64 parallel threads are collected within one "warp" (NVIDIA) or "wavefront" (AMD), respectively. Since branching within a warp necessitates the evaluation of both branches, such warp-level branching has to be avoided for a highly performant code, which is particularly problematic if combined with integral screening. However, our sub-batch implementation of ref 44 provides spatially local sub-batches with exactly the same number of grid points. Therefore, we choose sub-batches of exactly 64 points, which perfectly maps to the warp/wavefront size of current GPUs. We thus perform the tightest level of integral screening (employing eqs 13 and 14) for 64 points at once, thereby minimizing the screening overhead and ensuring identical branching within each warp/wavefront.

Algorithm 2 Evaluation of eq. (6) (integral evaluation)

```
for all sign. shell-pairs $\nu\lambda$ do                                     ▷ sequential
    for all sub-batches $\tilde{b}$ of 64 points do                    ▷ multi warp parallel
        if shell-pair $\nu\lambda$ is significant by eqs. (13) and (14) for $\tilde{b}$ then
            for all grid points $g \in \tilde{b}$ do               ▷ SIMD-parallel (within warp)
                for all primitive shell-pairs do                             ▷ sequential
                    compute primitive integrals (Boys integrals)
                    perform optimized Obara-Saika recursions → Cartesian integrals
                end for
                multiply Cartesian integrals with $F_{\lambda g}$ and add onto $G_{\nu g}$
            end for
        end if
    end for
end for
```

For our GPU implementation, we employ the same computer-optimized Obara−Saika[66,67] recursions as for the CPU code, that is, our CUDA, OpenCL, and CPU implementations share the same input file for the integral kernels. This reuse of the 3c-1e integral code simplifies the GPU implementation significantly, an important advantage compared to the analytical 4c-2e integral-based methods (see, e.g., refs7,11), where considerable modifications have to be made to obtain an efficient and performant code. Since the integral kernels are parallelized solely over the grid point within each batch, there is no need for communication between different threads. This also considerably simplifies the GPU implementation because neither shared memory nor explicit synchronization have to be utilized.

The optimized recurrence relations were generated by application of common sub-expression elimination (CSE) (implemented in the SymPy Python package[68]) to the unrolled Obara–Saika recursions for each specific *l*-quantum number combination. Moreover, we found that for most practical applications, Head-Gordon–Pople-like (HGP)[69] shifts on the contracted level do not provide speedups for 3c-1e integrals in practice since the recursions are only relevant for larger *l*-quantum number combinations, but basis functions with *l*-quantum numbers larger than 1 (*d* functions or higher) rarely contain more than one primitive Gaussian. Therefore, we decided not to perform any HGP-like contracted recursion steps.

To avoid the transformation between pure and Cartesian integrals, we perform the whole sn-LinK algorithm with Cartesian basis functions, that is, we initially transform the density matrix into the Cartesian basis, then perform the whole sn-LinK algorithm in the Cartesian basis, and finally transform the exchange matrix back to the pure basis. Analogously, we also multiply the non-axial normalization factors, which are needed to ensure normalization of the non-axial basis functions (e.g., d_{xy}), onto the initial density matrix and onto the final exchange matrix, thereby avoiding the necessity to multiply these factors within the integral code.

4. COMPUTATIONAL DETAILS

To provide a fair comparison between GPU and CPU codes, all possible optimization options were enabled for both the CPU and GPU integral codes. The CPU kernels are compiled with the Intel C++ compiler (ICPC) version 19.0.1[70] with the "-Ofast", "-march = native", options, to enable autovectorization of our integral code using the AVX2 instruction set extensions. We have also tested GCC[71] and Clang[72] but found that the Intel C++ compiler provides significantly better performance (up to a factor of two) due to better optimization heuristics, more aggressive autovectorization, and more advanced instruction reordering. The 3c-1e CPU integral kernels benefit particularly from these optimization because parallelization over the grid index is well suited for SIMD vectorization, whereas for the 4c-2e integral kernels, vectorization is hindered by the heterogeneity of the shell pairs (i.e., different amounts of primitive Gaussians), the branching associated with LinK,[20] and the need for more local storage.

The CUDA kernels were compiled with NVCC-10.0 (CUDA-10.0)[63] with "-O3" and "-use_fast_math" using GCC-7.1 as the host compiler. The OpenCL kernels were precompiled with amdgpu-pro-19.20[65] employing the "-O3", "-cl-mad-enable", "-cl-finite-math-only", and "-cl-no-signed-zeros" options. The CPU timings are performed on one server node with 2 Intel Xeon Silver 4216 CPUs comprising 32 cores at 2.1 GHz providing a performance of 1.075 × 10^{12} floating-point operations (FLOPs) per second (1.075 TFLOPs/s). The GPU timings are performed on the NVIDIA-GV100 GPU (8.33 TFLOPs/s) and the Radeon VII (3.36 TFLOPs/s). The geometries of the molecules[73] employed in this work are available online at http://www.cup.lmu.de/pc/ochsenfeld/download/.

Throughout this work, we employ our recently developed grids defined in the appendix of ref 62 and briefly summarized in Table 1. All presented timings are given for one full exchange matrix build employing a converged density matrix and the smaller (SCF) grid of the multigrids defined in Table 1, in this way, representing a typical SCF step without incremental Fock builds. Note that molecular grids typically contain about 10 to 30% less grid points than the atomic grids defined in Table 1 due to the erasure of grid points with zero weights, a consequence of our modification to Becke's molecular partitioning scheme[60] (see also discussion in ref 62).

Table 1. Specification of the Grids Employed in the Present Work Given as "$n_{\text{rad}}/n_{\text{ang}}$ (Number of Points per C Atom)"[a]

grid	SCF grid	final grid
"gm3"	35/110 (2586)	50/302 (9564)
"gm4"	40/194 (5056)	55/434 (15526)
"gm5"	50/302 (9564)	60/590 (21330)

[a] Within the SCF, a coarser grid (denoted as SCF grid) was employed and a finer grid was used for the final energy calculation (denoted as final grid). Grids have been pruned, that is, less angular points are employed for the inner radial shells of each atom.

The timings of the conventional (4c-2e integral-based) code exclude the preLinK[7] preselection, since in the current version of our FermiONs++ program,[7,9] the two matrix multiplications within the preLinK algorithm are performed on the CPU using dense matrix algebra, adding a significant overhead for large systems. However, the sn-LinK timings comprise every step needed for exchange matrix formation, including the preselection.

For all sn-LinK calculations, we choose the screening thresholds $\vartheta_K = 1.0 \times 10^{-7}$ and $\vartheta_E = 1.0 \times 10^{-10}$ during the SCF and $\vartheta_E = 1.0 \times 10^{-11}$ for final energy calculation. These thresholds provide screening errors smaller than 1nE_h per basis function for all tested systems, which is consistent with our default threshold for the analytical 4c-2e integrals (10^{-10}). Although significantly looser thresholds could probably be used for most applications, we wanted to provide a very safe default in terms of numerical stability and encourage the user to fine tune these parameters for the specific system of interest to obtain even better performance than presented here.

5. RESULTS AND DISCUSSION

5.1. Accuracy of the Numerical Integration Grids.

We begin the analysis of the sn-LinK method by investigating the errors caused by the numerical integration. In Table 2, we investigate the grid-induced errors in the Hartree–Fock energy and the indirectly induced errors in the MP2 energy, caused by the errors in the converged density matrix and serving as a measure for the accuracy of the density matrix. We employ the G2 test set[74] (atomization energies of small molecules), the S22x5 test set[75] (noncovalently bound small dimers), and the L7 test set[76] (7 noncovalently bound dimers with up to 101 atoms) in combination with the def2-TZVP basis set.[77]

Even for our smallest grid "gm3", all errors are significantly below 1mE_h and are therefore considered insignificant compared to typical errors from methods and basis sets. Moreover, these errors rapidly decrease with larger grids, and the "gm5" grid provides numerical accuracy up to a few μE_h. Interestingly, the Hartree–Fock errors agree well with the observation we made in ref 62 about the grid errors of the Perdew–Burke–Ernzerhof (PBE)[78] functional despite the use of a very different energy functional.

If only single-point energies are of interest, "gm3" should be the best choice for maximum efficiency, whereas if energy

Table 2. Grid-Induced Errors in the Absolute Hartree−Fock (HF) Energy and the Absolute MP2 Correlation Energy (G2 Test Set) or the Respective Interaction Energies (S22x5 and L7 Test Sets) Referenced to the Analytical (4c-2e Integral-Based) Method Employing the def2-TZVP Basis Set[a]

		HF			MP2		
test set	deviation	gm3	gm4	gm5	gm3	gm4	gm5
G2	MaxD	20.0	7.0	2.0	69.1	12.2	2.1
	MAD	2.3	0.7	0.2	6.0	1.2	0.1
S22 (0.9x)	MaxD	84.5	20.2	5.9	178.2	39.6	5.2
	MAD	18.8	4.5	1.3	31.4	9.6	1.9
S22 (1.0x)	MaxD	47.8	17.0	7.1	176.8	43.2	3.7
	MAD	15.1	3.7	1.3	31.9	9.9	1.5
S22 (1.2x)	MaxD	57.4	13.8	4.9	66.2	46.7	3.4
	MAD	16.6	4.6	1.2	32.5	9.7	1.1
S22 (1.5x)	MaxD	63.0	18.9	4.1	154.6	49.3	4.5
	MAD	12.7	3.8	0.9	34.8	10.5	1.2
S22 (2.0x)	MaxD	80.3	16.9	3.0	117.6	51.0	5.2
	MAD	13.6	3.7	0.8	30.8	10.9	1.2
L7	MaxD	165.3	42.4	21.3	489.9	119.1	25.2
	MAD	20.2	24.3	5.0	147.7	28.6	7.4

[a] The errors in the MP2 energy are only due to the errors in the converged density matrix. The seminumerical integration was only used for the exchange matrix formation within the SCF but not within the MP2 calculation.

derivatives (forces and vibrational frequencies) are investigated, we recommend to default to the finer "gm5" grid for the higher numerical stability. Note that much smaller grids have been recommended in the works of Neese et al.[36] and Friesner and co-workers.[27,28,33,79] However, we advise caution for the use of small grid especially when computing molecular properties and recommend to carefully test the influence of the grid on the specific quantity of interest prior to any application.

In contrast to our method, analytical corrections to the seminumerical exchange matrix are added within the approach of ref 36 and to even greater extent within the approach of refs,[27,28,33,79] that is, some selected 4c-2e integrals are computed analytically to reduce the grid error. In particular, Neese et al.[36] proposed to only employ one-center corrections, that is, all integrals where all four basis functions reside at the same atom are computed analytically. We tested this approach but found no improvement at all since our grids integrate every atom-centered function pair virtually exactly. The grid errors thus arise solely from non atom-centered function pairs, where the two different functions reside on different atoms and are therefore not considered within the one-center corrections.

5.2. Performance Analysis. In the following, the performance of sn-LinK is assessed in terms of asymptotic scaling behavior (Section 5.2.1), floating-point performance (Section 5.2.2), multistream GPU performance (Section 5.2.3), and multi-GPU performance (Section 5.2.4). Finally, we give a comparison with the 4c-2e-based preLinK method[7,20] on both CPUs and GPUs in Section 5.2.5.

5.2.1. Scaling with Respect to the System Size. Although the evaluation of eqs 5 to 7 formally scales as $O(M^3)$ with respect to the system size (more specifically $O(N_{\text{grid}}N_{\text{bas}}^2)$), exploitation of the locality of the Gaussian basis functions and of the locality of the exchange interaction for systems with nonzero HOMO−LUMO gaps should result in an asymptotic $O(M)$ scaling, if the screening techniques described in Sections 2.2 to 2.4 are employed. That is, sn-LinK is asymptotically linear scaling by construction. For most practical systems, however, the observed scaling lies somewhere between the formal $O(M^3)$ and the asymptotic $O(M)$ scaling.

In Figure 1, we investigate the scaling behavior for linear alkanes, separated into the 3c-1e integral part required for the evaluation of eq 6 and the matrix multiplication (BLAS-3) steps of eqs 5 and 7.

In all cases, almost linear scaling is reached for the largest fragments. Unsurprisingly, with larger and more diffuse basis sets, linear scaling is reached later (i.e., for larger fragments) since the selection schemes of Sections 2.2 and 2.4 exploit the locality of the Gaussian basis functions. Interestingly, the 3c-1e integral part reaches linear scaling faster than the BLAS-3 steps. This is a consequence of different screening techniques employed for these two steps, that is, the preselection scheme of Section 2.4 compared to the integral selection scheme of Section 2.4, where the latter is tighter (individual contributions are overestimated to a lesser extent).

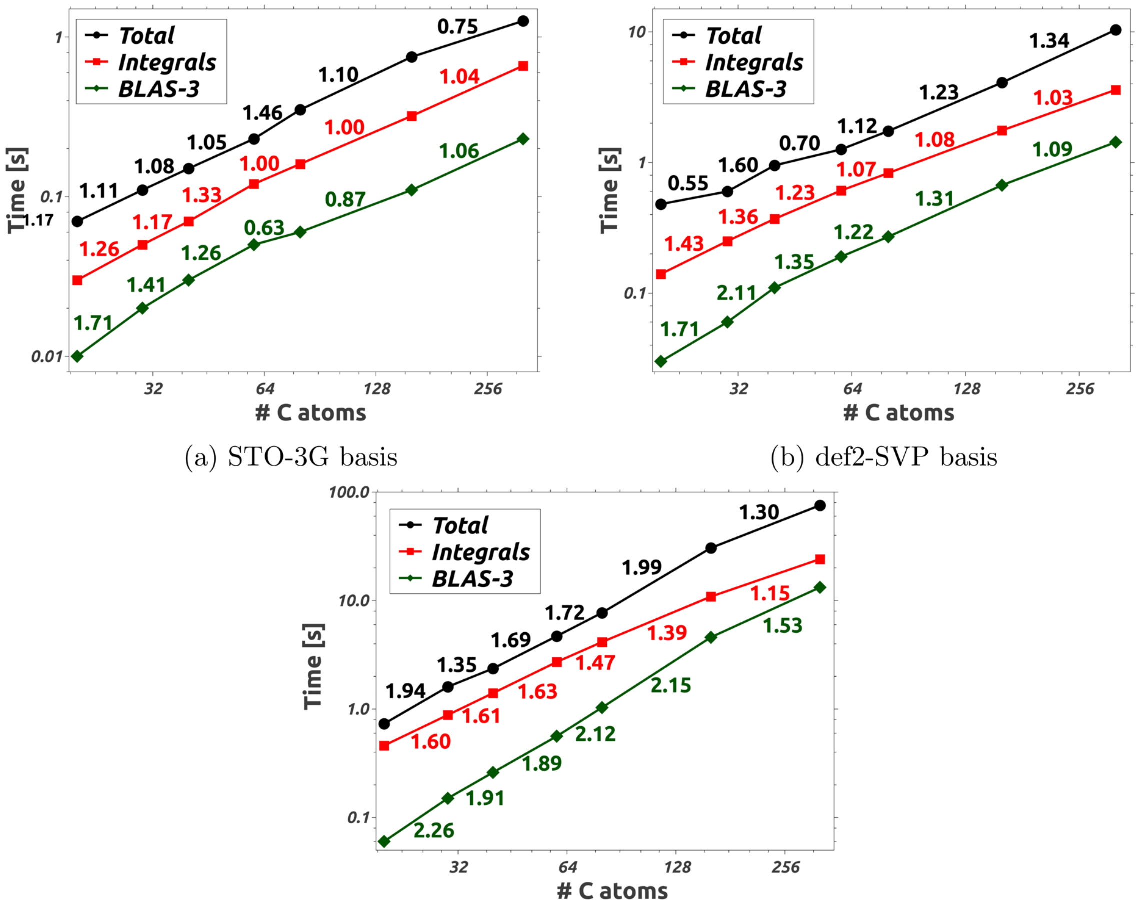
Although linear alkane chains are a valuable model system to analyze the asymptotic scaling behavior, more globular systems are of interest for many practical applications. Therefore, a more detailed efficiency analysis of our sn-LinK method is given for adenine-thymine DNA fragments in Figure 2 and for spherical water clusters in Figure 3.

Here, all the observations discussed above for linear alkanes are still valid. That is, the integrals reach linear scaling faster than the BLAS-3 steps, and the asymptotically linear scaling is reached later for larger basis sets. Indeed, the linear-scaling onset for def2-TZVP is so late that even the largest fragment of $(\text{DNA})_{16}$ still scales quadratically. Such a late onset of linear scaling has also been observed by ref 36. In contrast to our previous work,[44] sn-LinK (present work) selects significant shells and shell pairs according to their contributions to the exchange potential matrix instead of the exchange energy. This results in a later onset of linear scaling but provides better SCF convergence, particularly for larger basis sets.

Moreover, due to the heterogeneity of GPU computing, the total execution time within sn-LinK also contains a considerable amount of noncompute steps, for example, CPU-GPU data transfer and memory management, as illustrated for $(\text{DNA})_{16}$/TZVP in Figure 4. The performance impact of these other steps can, however, be significantly reduced by employing multiple streams per GPU since the different steps do not compete for the same computational resources (see also Section 5.2.3). Moreover, the high cost of these other steps necessitates the use of rather large grid batches and prohibits the use of block-sparse matrix multiplications to accelerate the BLAS-3 steps since the management steps would dominate the computation time otherwise. The larger grid batches also contribute to a later onset of linear scaling within the BLAS-3 steps.

In summary, although sn-LinK scales linearly by construction, perfect $O(M)$ scaling is only archived for the largest systems and smaller basis sets. This is the expected behavior since the selection schemes within sn-LinK exploit the locality of the basis functions and of the electronic structure.

5.2.2. FLOP Utilization of the 3c-1e Integral Kernels. Since the 3c-1e integrals still represent the most time-consuming step in the seminumerical exchange build, we put significant effort into its optimization. In particular, the batch-wise integral screening described in Section 2.2 allows for SIMD parallelization resulting in comparatively high utilization of the



(a) STO-3G basis

(b) def2-SVP basis

(c) def2-TZVP basis

Figure 1. (a−c) Total program execution time and individual execution times for 3c-1e integrals (eq 6) and for the two BLAS-3 steps (eqs 5 and 7) within one exchange build for linear alkanes on one NVIDIA GV100 GPU with one CUDA stream given as a double logarithmic plot. The colored numbers correspond to the scaling with respect to the preceding fragment.

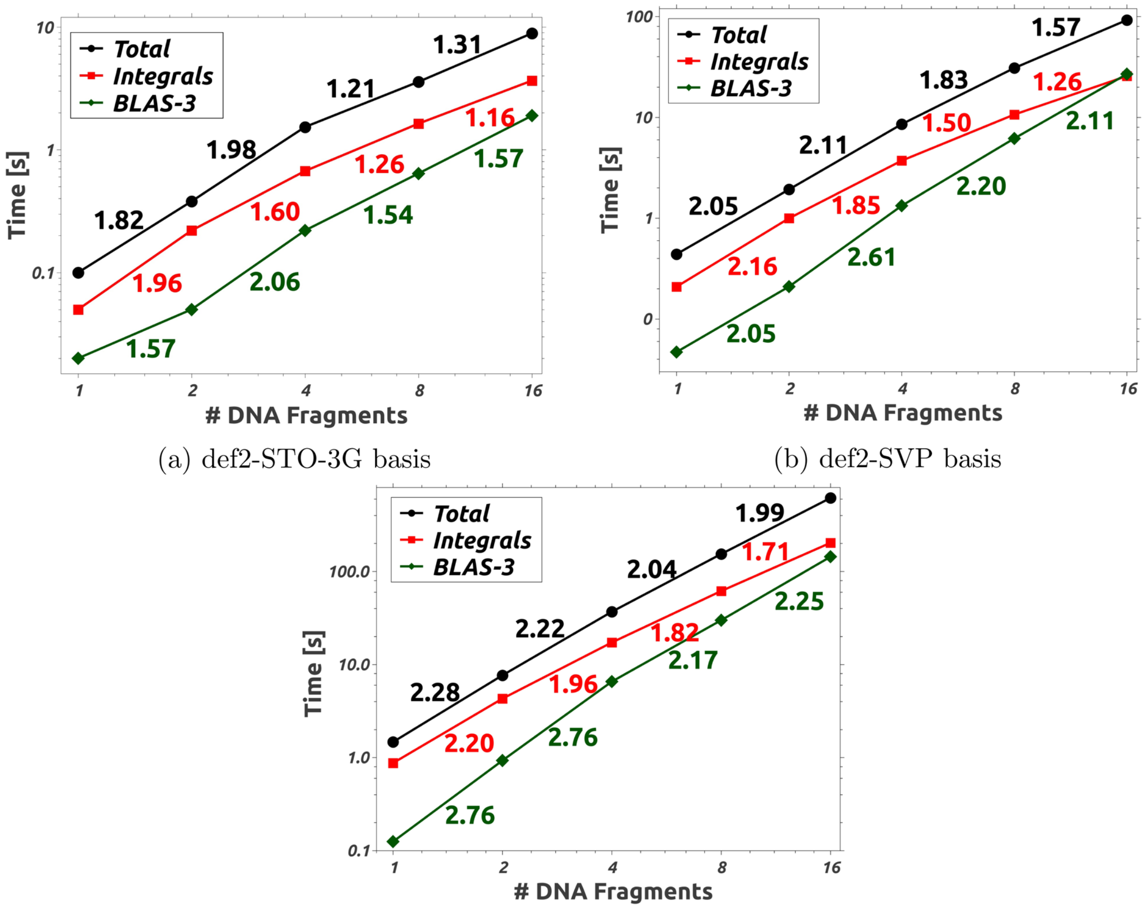
theoretical floating-point performance, as presented in Table 3. An outline how the FLOP counts were obtained is contained within the Supporting Information.

To provide some context, the best dense linear algebra libraries achieve about 70 to 80% FLOP utilization, the GAMESS program package[80,81] was reported[13] to provide 0.51 to 1.4 GFLOPs/s (6 to 17.5% utilization) on CPUs, and GPU implementations for 4c-2e integrals were reported to provide 10 to 30 GFLOPs/s (13 to 39% utilization)[13] in double precision and up to 80 GFLOPs/s (6% utilization)[6] in single precision. Although the theoretical FLOP performance of processors grows exponentially due to ongoing developments in microarchitectures, it becomes increasingly difficult to utilize their full potential since other bottlenecks (cache, memory latency, and bandwidth) dominate in many cases. In this context, the FLOP performance of our 3c-1e integral kernels (230 to 330 GFLOPs/s; 22 to 32% utilization on CPUs and up to 1040 GFLOPs/s; 11 to 16% utilization on GPUs) is very promising.

5.2.3. Multiple Streams on One GPU. In all of the above performance analysis, only one stream per GPU was utilized to time the different steps separately. However, employing more than one stream per GPU should provide some additional speedup since CPU workloads, GPU workloads, and CPU-GPU data transfer allocate different resources and can therefore be performed concurrently. That is, one stream can, for example, transfer data to the GPU, while another stream performs GPU calculations at the same time, thus optimizing the total device utilization (see also discussion in Section 3). The performance gains of this optimization are presented in Table 4.

Compared to the single-streamed evaluation, speedups of up to 50% can be achieved with multiple streams, where the majority of this speedup is already achieved with two streams per GPU. However, the memory use of each GPU scales proportionally with the amount of employed streams, and we therefore decided to employ three streams per GPU as a sensible compromise between performance and GPU memory usage.

5.2.4. Multi-GPU Scaling. Since many high-performance-computing (HPC) servers or workstation are available with up to 16 GPUs per node, the parallel scaling with an increasing amount of GPUs is also of high interest, particularly if employing comparatively inexpensive GPUs like the AMD Radeon VII. We therefore present the multi-GPU scaling of our sn-LinK code in Table 5, activating one, two, or four AMD Radeon VII GPUs.



(a) def2-STO-3G basis

(b) def2-SVP basis

(c) def2-TZVP basis

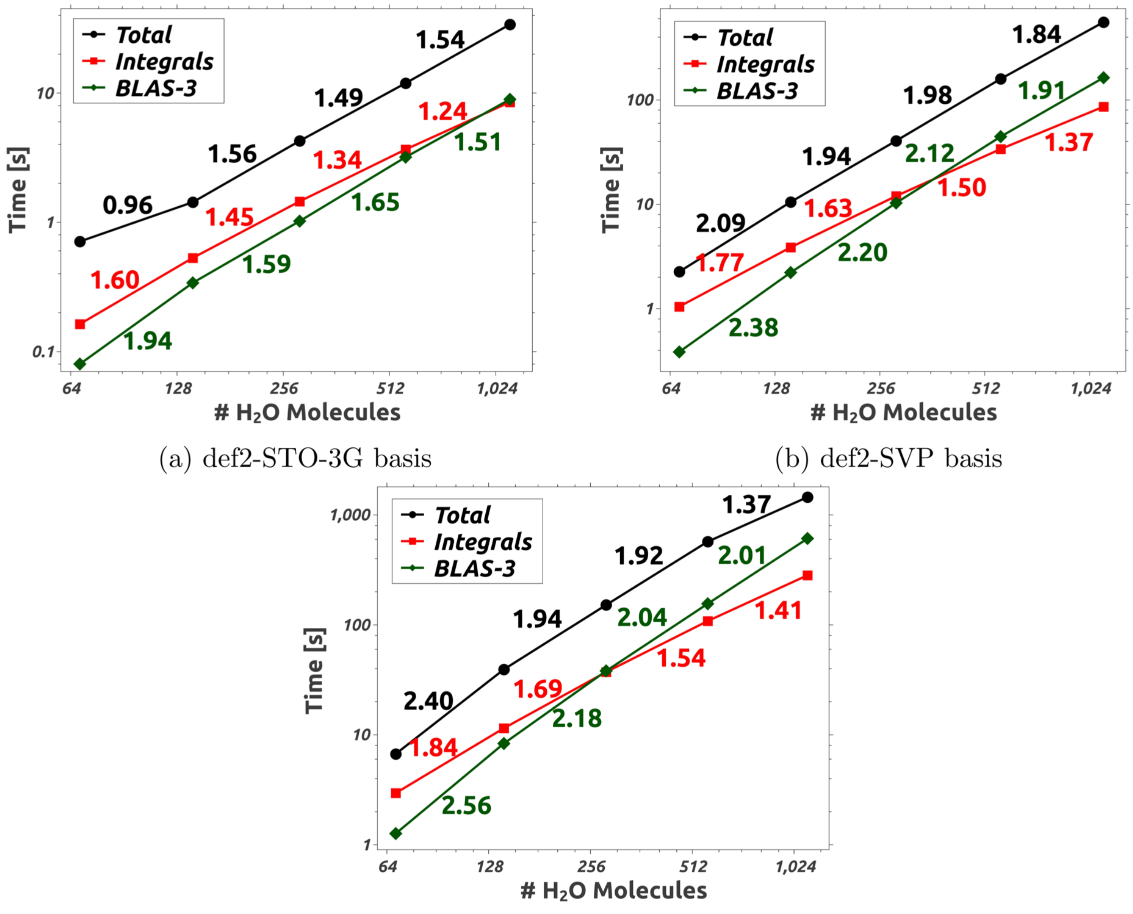
Figure 2. (a−c) Total program execution time and individual execution times for 3c-1e integrals (eq 6) and for the two BLAS-3 steps (eqs. 5 and 7) within one exchange build for adenine-thymine DNA fragments on one NVIDIA GV100 GPU with one CUDA stream given as a double logarithmic plot. The colored numbers correspond to the scaling with respect to the preceding fragment.

We indeed observe very favorable parallel scaling (3.5× to 4.4× speedups for 4 GPUs), particularly considering that the available CPU resources and memory bandwidth need to be shared between four devices. The over 100% parallel efficiency for $(DNA)_4$/TZVP/"gm5" is a consequence of a particular fortunate workload distribution, that is, all 12 streams finished at very similar times. Such timing fluctuations are typical within such a highly parallel setup because the number of work batches per stream is very small, for example, for $(DNA)_4$/"gm3", there are only 40 grid batches in total, which needs to be split between a total of 12 streams if all four GPUs are utilized.

5.2.5. Comparison with PreLinK. In Table 6, we compare the CPU and GPU performance of the 3c-1e integral-based sn-LinK method of the present work with the analytical (4c-2e integral-based) preLinK method[7] employing 32 CPU cores, 4 Radeon VII GPUs, or 1 NVIDIA GV100 GPU. In this comparison, preLinK typifies all other 4c-2e integral-based methods for Fock exchange, as implemented in most quantum chemistry programs, and allows for a consistent comparison within the same program on both CPUs and GPUs.

The sn-LinK method outperforms the analytical method in most tested applications on CPUs and, to even greater extent, on GPUs. The performance gains from sn-LinK compared to the analytical method are most significant for larger systems and larger basis sets (e.g., factor 17 (14.7 s vs 252 s) for $(DNA)_4$/def2-TZVP/"gm3") due to the superior basis set scaling $(O(N_{\text{bas}}^2))$ of sn-LinK compared to preLinK $(O(N_{\text{bas}}^4))$. Moreover, the seminumerical code provides better CPU → GPU speedups (up to a factor of 9.5 on four AMD Radeon VII GPUs and a factor of 5.5 on one NVIDIA GV100 GPU) than the analytical code (up to 4.6 on four Radeon VII and 3.8 on one GV100). The better speedups are a direct consequence of the reduced local storage requirements of the 3c-1e integral code compared to the 4c-2e code, resulting in a significantly better utilization of the GPU's floating-point compute units.

In summary, the sn-LinK methods transfer particularly well to GPUs and therefore enable the routine computation of large molecules containing hundreds of atoms and large basis sets. This represents a substantial improvement over existing seminumerical methods, for example, for the fullerene C_{240}/cc-PVTZ, our sn-LinK method is close to 100 times faster than the seminumerical Intel Xeon Phi-based implementation of ref 43 (30.5 s vs 2970 s). In addition, our sn-LinK method allows for routine calculation for hundreds of atoms and augmented quadruple-ζ basis sets, (e.g., one exchange build for $(DNA)_4$/def2-QZVPPD/"gm3" takes only 257 s), which is of particular interest in combination with post-Hartree−Fock correlation



(a) def2-STO-3G basis

(b) def2-SVP basis

(c) def2-TZVP basis

Figure 3. (a−c) Total program execution time and individual execution times for 3c-1e integrals (eq 6) and for the two BLAS-3 steps (eqs. 5 and 7) within one exchange build for spherical water clusters on one NVIDIA GV100 GPU with one CUDA stream given as a double logarithmic plot. The colored numbers correspond to the scaling with respect to the preceding fragment.

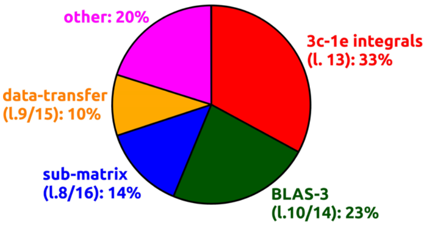


Figure 4. Breakdown of the total execution time of one exchange build for $(DNA)_{16}$/TZVP into different lines of Algorithm 1.

Table 3. Number of Floating-Point Operations Necessary for the Evaluation of All 3c-1e Integrals (with Batch-Wise Integral Selection Activated) for One Exchange Build for $(DNA)_4$ and Floating-Point Performance of the Integral Code Given as GFLOPs/s (Utilization of the Theoretical FLOP Performance in Parentheses)

basis	#GFLOPs	CPU	GV100	R. VII
STO-3G	700	330 (30.7%)	1040 (15.5%)	426 (12.8%)
def2-SVP	2780	239 (22.2%)	750 (11.2%)	429 (12.9%)
def2-TZVP	16,100	234 (21.8%)	932 (13.9%)	470 (14.1%)
def2-QZVPPD	199,000	257 (23.9%)	797 (11.9%)	

methods, which typically require large basis set for accurate results.

Furthermore, calculations employing basis functions with very high angular momentum (e.g., *g*-functions) are very challenging for 4c-2e-based GPU implementations since the high complexity of the high-*l*-quantum number kernels (e.g., our (*gg*|*gg*) kernel contains over 100,000 lines of code) can lead to numerical instabilities of our present GPU code. The extent of this problem depends on the specific GPU in use and has also been reported by other groups.[6] Our sn-LinK method, however, does not suffer from these issues because the 3c-1e integrals are much simpler to evaluate.

6. CONCLUSIONS AND OUTLOOK

Within the present work, we described a new, highly efficient seminumerical exchange method, denoted as sn-LinK, and outlined its implementation for graphic processing units. After validating the accuracy of the numerical integration, we compared the performance of this new method with our conventional (4c-2e integral-based) preLinK method[7] and found outstanding performance improvements, especially for larger basis sets. Moreover, we showed that the sn-LinK algorithm benefits particularly well from GPU acceleration due to the lower local storage requirements of the 3c-1e integral

Table 4. Time in Seconds for One Exchange Build Using One to Four Streams on One GPU Employing the "gm3" Grid

		GV100				R. VII			
system	basis streams	1	2	3	4	1	2	3	4
$(DNA)_1$	def2-SVP	0.36	0.29	0.26	0.27	0.85	0.68	0.79	0.98
$(DNA)_1$	def2-TZVP	1.36	1.11	0.98	1.10	3.01	2.73	2.38	2.85
$(DNA)_4$	def2-SVP	7.4	5.5	5.3	5.0	15.1	11.3	10.9	10.5
$(DNA)_4$	def2-TZVP	33.8	27.4	25.4	25.2	64.3	52.9	50.7	49.8

Table 5. Multi-GPU Scaling Employing One, Two, and Four Radeon VII GPUs within One Node[a]

system	basis	grid	1 GPU	2 GPUs	4 GPUs
$(DNA)_4$	def2-SVP	"gm3"	10.9	5.4 (2.0)	3.0 (3.6)
$(DNA)_4$	def2-TZVP	"gm3"	50.7	26.0 (1.9)	14.7 (3.5)
$(DNA)_4$	def2-SVP	"gm5"	27.3	13.9 (1.9)	6.3 (4.4)
$(DNA)_4$	def2-TZVP	"gm5"	143.3	73.2 (2.0)	37.9 (3.8)

[a] Timings are given in seconds for one exchange matrix build, employing a converged density matrix using the smaller (SCF) grid from Table 1. Speedup compared to 1 GPU in parentheses.

kernels compared to the 4c-2e kernels that are required within conventional implementations. Furthermore, we could verify the asymptotic linear-scaling behavior of our implementation for linear alkanes, DNA fragments, and spherical water clusters for small basis sets. For the larger def2-TZVP basis sets, the onset for linear scaling is so late that it was only observed for large linear alkanes and water clusters.

Although the focus of the present work was solely on single-point calculations, seminumerical methods are particularly efficient for computing molecular forces since no integral derivatives need to be evaluated.[29,36] Moreover, the extension of the sn-LinK algorithm to local hybrid functionals is straightforward in principle, however, requiring quite some additional implementation effort to merge the CPU-based DFT code with the GPU-based sn-LinK code. Thus, our, herein, presented sn-LinK algorithm also facilitates future developments of local hybrid functionals, which used to be restrained by their high computational cost. These two extensions are currently under development and will be discussed in future work.

Finally, we want to emphasize the applicability of the seminumerical/pseudospectral method to other molecular properties[31,32,82,83] and post-Hartree−Fock correlation methods[84] as well as the conceptional similarities to the tensor hypercontraction (THC) framework.[85]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.9b00860.

Evaluation of 3c-1e integral FLOP counts (PDF)

Table 6. Timings in Seconds for CPU (32 Cores/64 Threads@2.10 GHz) and GPU Codes Run on either Four AMD Radeon VII (4× R. VII) or One NVIDIA GV100 (GV100) Using sn-LinK (Denoted as "sn-LinK@gm3"/"sn-LinK@gm5") as Compared to the preLinK Method of Ref 7 (Denoted as "preLinK")[a]

system	basis	method	#BFs	CPU	4× R. VII	GV100
$(DNA)_1$	def2-SVP	"sn-LinK@gm3"	660	0.8	0.3 (2.4)	0.4 (2.0)
$(DNA)_1$	def2-SVP	"sn-LinK@gm5"	660	2.5	0.6 (4.2)	0.8 (3.2)
$(DNA)_1$	def2-SVP	"preLinK"	660	1.7	1.0 (1.8)	1.2 (1.5)
$(DNA)_1$	def2-TZVP	"sn-LinK@gm3"	1422	3.6	1.0 (3.7)	1.2 (3.0)
$(DNA)_1$	def2-TZVP	"sn-LinK@gm5"	1422	11.0	2.4 (4.7)	3.1 (3.6)
$(DNA)_1$	def2-TZVP	"preLinK"	1422	29.6	10.4 (2.8)	15.6 (1.9)
$(DNA)_1$	def2-QZVPPD	"sn-LinK@gm3"	3815	30.0	8.1 (3.7)	10.5 (2.9)
$(DNA)_1$	def2-QZVPPD	"sn-LinK@gm5"	3815	99.6	21.5 (4.6)	32.5 (3.1)
$(DNA)_1$	def2-QZVPPD	"preLinK"	3815	2035	–	–
$(DNA)_4$	def2-SVP	"sn-LinK@gm3"	2904	17.6	3.0 (5.9)	5.8 (3.1)
$(DNA)_4$	def2-SVP	"sn-LinK@gm5"	2904	59.3	6.3 (9.5)	14.4 (4.1)
$(DNA)_4$	def2-SVP	"preLinK"	2904	54.2	18.5 (2.9)	23.2 (2.3)
$(DNA)_4$	def2-TZVP	"sn-LinK@gm3"	6336	139.5	14.7 (9.5)	27.2 (5.1)
$(DNA)_4$	def2-TZVP	"sn-LinK@gm5"	6336	316.3	37.9 (8.3)	75.3 (4.2)
$(DNA)_4$	def2-TZVP	"preLinK"	6336	1038.8	252.2 (4.1)	419.9 (2.5)
$(DNA)_4$	def2-QZVPPD	"sn-LinK@gm3"	16,574	1334	257.4 (5.2)	307.8 (4.3)
$(DNA)_4$	def2-QZVPPD	"sn-LinK@gm5"	16,574	5119	814.6 (6.3)	924.2 (5.5)
$(DNA)_4$	def2-QZVPPD	"preLinK"	16,574	101,250	–	–
C_{240}	cc-pVDZ	"sn-LinK@gm3"	3600	49.9	6.6 (7.6)	12.2 (4.1)
C_{240}	cc-pVDZ	"sn-LinK@gm5"	3600	168.8	19.4 (8.7)	38.6 (4.4)
C_{240}	cc-pVDZ	"preLinK"	3600	411.4	162.1 (2.5)	215.6 (1.9)
C_{240}	cc-pVTZ	"sn-LinK@gm3"	8400	207.0	30.5 (6.8)	55.0 (3.8)
C_{240}	cc-pVTZ	"sn-LinK@gm5"	8400	678.6	86.8 (7.8)	170.4 (4.0)
C_{240}	cc-pVTZ	"preLinK"	8400	6294	1365 (4.6)	1660 (3.8)

[a] Timings are given for one exchange matrix build, employing a converged density matrix using the smaller (SCF) grid from Table 1. For context, the number of Cartesian basis functions (#BFs) is given for each system, and the CPU → GPU speedups are given in parentheses.

AUTHOR INFORMATION

Corresponding Author

Christian Ochsenfeld − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU) D-81377 München, Germany;* orcid.org/0000-0002-4189-6558; Email: christian.ochsenfeld@uni-muenchen.de

Authors

Henryk Laqua − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU) D-81377 München, Germany*

Travis H. Thompson − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU) D-81377 München, Germany*

Jörg Kussmann − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU) D-81377 München, Germany;* orcid.org/0000-0002-4724-8551

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.9b00860

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) (SFB 1309-325871075). In addition, the authors acknowledge the financial support by the DFG cluster of excellence (EXC2111) "Munich Center for Quantum Science and Technology", MCQST. C.O. further acknowledges financial support as Max-Planck-Fellow at the MPI-FKF Stuttgart. We dedicate this work to Dr. Klaus Römer on the occasion of his 80th birthday in February 2020. We are grateful to him for supporting our Faculty at LMU Munich for many years.

REFERENCES

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

(2) Yasuda, K. Accelerating Density Functional Calculations with Graphics Processing Unit. *J. Chem. Theory Comput.* **2008**, *4*, 1230−1236.

(3) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics. *J. Chem. Theory Comput.* **2009**, *5*, 2619−2628.

(4) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field Implementation. *J. Chem. Theory Comput.* **2009**, *5*, 1004−1015.

(5) Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. Dynamic Precision for Electron Repulsion Integral Evaluation on Graphical Processing Units (GPUs). *J. Chem. Theory Comput.* **2011**, *7*, 949−954.

(6) 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.

(7) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(8) 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*, No. 051106.

(9) 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.

(10) Kussmann, J.; Ochsenfeld, C. Employing OpenCL to Accelerate Ab Initio Calculations on Graphics Processing Units. *J. Chem. Theory Comput.* **2017**, *13*, 2712−2716.

(11) Kussmann, J.; Ochsenfeld, C. Hybrid CPU/GPU Integral Engine for Strong-Scaling Ab Initio Methods. *J. Chem. Theory Comput.* **2017**, *13*, 3153−3159.

(12) Beuerle, M.; Kussmann, J.; Ochsenfeld, C. Screening methods for linear-scaling short-range hybrid calculations on CPU and GPU architectures. *J. Chem. Phys.* **2017**, *146*, 144108.

(13) 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.

(14) Asadchev, A.; Gordon, M. S. New Multithreaded Hybrid CPU/GPU Approach to Hartree-Fock. *J. Chem. Theory Comput.* **2012**, *8*, 4166−4176.

(15) Miao, Y.; Merz, K. M., Jr. Acceleration of Electron Repulsion Integral Evaluation on Graphics Processing Units via Use of Recurrence Relations. *J. Chem. Theory Comput.* **2013**, *9*, 965−976.

(16) Schwegler, E.; Challacombe, M. Linear scaling computation of the Hartree-Fock exchange matrix. *J. Chem. Phys.* **1996**, *105*, 2726−2734.

(17) Burant, J. C.; Scuseria, G. E.; Frisch, M. J. A linear scaling method for Hartree-Fock exchange calculations of large molecules. *J. Chem. Phys.* **1996**, *105*, 8969−8972.

(18) Challacombe, M.; Schwegler, E. Linear scaling computation of the Fock matrix. *J. Chem. Phys.* **1997**, *106*, 5526−5536.

(19) Schwegler, E.; Challacombe, M.; Head-Gordon, M. Linear scaling computation of the Fock matrix. II. Rigorous bounds on exchange integrals and incremental Fock build. *J. Chem. Phys.* **1997**, *106*, 9708−9717.

(20) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. Linear and sublinear scaling formation of Hartree-Fock-type exchange matrices. *J. Chem. Phys.* **1998**, *109*, 1663−1669.

(21) Ochsenfeld, C. Linear scaling exchange gradients for Hartree-Fock and hybrid density functional theory. *Chem. Phys. Lett.* **2000**, *327*, 216−223.

(22) Früchtl, H. A.; Kendall, R. A.; Harrison, R. J.; Dyall, K. G. An implementation of RI-SCF on parallel computers. *Int. J. Quantum Chem.* **1997**, *64*, 63−69.

(23) Weigend, F. A fully direct RI-HF algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency. *Phys. Chem. Chem. Phys.* **2002**, *4*, 4285−4291.

(24) Friesner, R. A. Solution of self-consistent field electronic structure equations by a pseudospectral method. *Chem. Phys. Lett.* **1985**, *116*, 39−43.

(25) Friesner, R. A. Solution of the Hartree-Fock equations by a pseudospectral method: application to diatomic molecules. *J. Chem. Phys.* **1986**, *85*, 1462−1468.

(26) Friesner, R. A. Solution of the Hartree-Fock equations for polyatomic molecules by a pseudospectral method. *J. Chem. Phys.* **1987**, *86*, 3522−3531.

(27) Friesner, R. A. An automatic grid generation scheme for pseudospectral self-consistent field calculations on polyatomic molecules. *J. Phys. Chem.* **1988**, *92*, 3091−3096.

(28) Ringnalda, M. N.; Belhadj, M.; Friesner, R. A. Pseudospectral Hartree-Fock theory: applications and algorithmic improvements. *J. Chem. Phys.* **1990**, *93*, 3397−3407.

(29) Won, Y.; Lee, J. G.; Ringnalda, M. N.; Friesner, R. A. Pseudospectral Hartree-Fock gradient calculations. *J. Chem. Phys.* **1991**, *94*, 8152−8157.

(30) Greeley, B. H.; Russo, T. V.; Mainz, D. T.; Friesner, R. A.; Langlois, J.-M.; Goddard, W. A., III; Donnelly, R. E., Jr.; Ringnalda, M. N. New pseudospectral algorithms for electronic structure calculations: length scale separation and analytical two-electron integral corrections. *J. Chem. Phys.* **1994**, *101*, 4028−4041.

(31) Cao, Y.; Friesner, R. A. Molecular (hyper)polarizabilities computed by pseudospectral methods. *J. Chem. Phys.* **2005**, *122*, 104102.

(32) Ko, C.; Malick, D. K.; Braden, D. A.; Friesner, R. A.; Martínez, T. J. Pseudospectral time-dependent density functional theory. *J. Chem. Phys.* **2008**, *128*, 104103.

(33) Bochevarov, A. D.; Harder, E.; Hughes, T. F.; Greenwood, J. R.; Braden, D. A.; Philipp, D. M.; Rinaldo, D.; Halls, M. D.; Zhang, J.; Friesner, R. A. Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences. *Int. J. Quantum Chem.* **2013**, *113*, 2110−2142.

(34) Zhang, J.; Weisman, A. L.; Saitta, P.; Friesner, R. A. Efficient simulation of large materials clusters using the jaguar quantum chemistry program: Parallelization and wavefunction initialization. *Int. J. Quantum Chem.* **2016**, *116*, 357−368.

(35) Cao, Y.; Hughes, T.; Giesen, D.; Halls, M. D.; Goldberg, A.; Vadicherla, T. R.; Sastry, M.; Patel, B.; Sherman, W.; Weisman, A. L.; Friesner, R. A. Highly efficient implementation of pseudospectral time-dependent density-functional theory for the calculation of excitation energies of large molecules. *J. Comput. Chem.* **2016**, *37*, 1425−1441.

(36) Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. Efficient, approximate and parallel Hartree-Fock and hybrid DFT calculations. A 'chain-of-spheres' algorithm for the Hartree-Fock exchange. *Chem. Phys.* **2009**, *356*, 98−109.

(37) Plessow, P.; Weigend, F. Seminumerical calculation of the Hartree-Fock exchange matrix: Application to two-component procedures and efficient evaluation of local hybrid density functionals. *J. Comput. Chem.* **2012**, *33*, 810−816.

(38) Bahmann, H.; Kaupp, M. Efficient Self-Consistent Implementation of Local Hybrid Functionals. *J. Chem. Theory Comput.* **2015**, *11*, 1540−1548.

(39) Maier, T. M.; Bahmann, H.; Kaupp, M. Efficient Seminumerical Implementation of Global and Local Hybrid Functionals for Time-Dependent Density Functional Theory. *J. Chem. Theory Comput.* **2015**, *11*, 4226−4237.

(40) Klawohn, S.; Bahmann, H.; Kaupp, M. Implementation of Molecular Gradients for Local Hybrid Density Functionals Using Seminumerical Integration Techniques. *J. Chem. Theory Comput.* **2016**, *12*, 4254−4262.

(41) Liu, F.; Furlani, T.; Kong, J. Optimal Path Search for Recurrence Relation in Cartesian Gaussian Integrals. *J. Phys. Chem. A* **2016**, *120*, 10264−10272.

(42) Liu, F.; Kong, J. Efficient Computation of Exchange Energy Density with Gaussian Basis Functions. *J. Chem. Theory Comput.* **2017**, *13*, 2571−2580.

(43) Liu, F.; Kong, J. An efficient implementation of semi-numerical computation of the Hartree-Fock exchange on the Intel Phi processor. *Chem. Phys. Lett.* **2018**, *703*, 106−111.

(44) Laqua, H.; Kussmann, J.; Ochsenfeld, C. Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals. *J. Chem. Theory Comput.* **2018**, *14*, 3451−3458.

(45) Maier, T. M.; Ikabata, Y.; Nakai, H. Efficient Semi-Numerical Implementation of Relativistic Exact Exchange within the Infinite-Order Two-Component Method Using a Modified Chain-of-Spheres Method. *J. Chem. Theory Comput.* **2019**, *15*, 4745−4763.

(46) Grotjahn, R.; Furche, F.; Kaupp, M. Development and Implementation of Excited-State Gradients for Local Hybrid Functionals. *J. Chem. Theory Comput.* **2019**, 5508.

(47) Becke, A. D. A real-space model of nondynamical correlation. *J. Chem. Phys.* **2003**, *119*, 2972−2977.

(48) Becke, A. D. Real-space post-Hartree-Fock correlation models. *J. Chem. Phys.* **2005**, *122*, 064101.

(49) Becke, A. D. Density functionals for static, dynamical, and strong correlation. *J. Chem. Phys.* **2013**, *138*, 074109.

(50) Becke, A. D. Communication: Calibration of a strong-correlation density functional on transition-metal atoms. *J. Chem. Phys.* **2013**, *138*, 161101.

(51) Johnson, E. R. A density functional for strong correlation in atoms. *J. Chem. Phys.* **2013**, *139*, 074110.

(52) Kong, J.; Proynov, E. Density Functional Model for Nondynamic and Strong Correlation. *J. Chem. Theory Comput.* **2016**, *12*, 133−143.

(53) Perdew, J. P.; Staroverov, V. N.; Tao, J.; Scuseria, G. E. Density functional with full exact exchange, balanced nonlocality of correlation, and constraint satisfaction. *Phys. Rev. A* **2008**, *78*, No. 052513.

(54) Jiménez-Hoyos, C. A.; Janesko, B. G.; Scuseria, G. E.; Staroverov, V. N.; Perdew, J. P. Assessment of a density functional with full exact exchange and balanced non-locality of correlation. *Mol. Phys.* **2009**, *107*, 1077−1088.

(55) Bahmann, H.; Rodenberg, A.; Arbuznikov, A. V.; Kaupp, M. A thermochemically competitive local hybrid functional without gradient corrections. *J. Chem. Phys.* **2007**, *126*, 011103−011103/4.

(56) Arbuznikov, A. V.; Bahmann, H.; Kaupp, M. Local Hybrid Functionals with an Explicit Dependence on Spin Polarization. *J. Phys. Chem. A* **2009**, *113*, 11898−11906.

(57) Arbuznikov, A. V.; Kaupp, M. Advances in local hybrid exchange-correlation functionals: from thermochemistry to magnetic-resonance parameters and hyperpolarizabilities. *Int. J. Quantum Chem.* **2011**, *111*, 2625−2638.

(58) Theilacker, K.; Arbuznikov, A. V.; Kaupp, M. Gauge effects in local hybrid functionals evaluated for weak interactions and the GMTKN30 test set. *Mol. Phys.* **2016**, *114*, 1118−1127.

(59) Thompson, T. H.; Ochsenfeld, C. Integral partition bounds for fast and effective screening of general one-, two-, and many-electron integrals. *J. Chem. Phys.* **2019**, *150*, No. 044101.

(60) Becke, A. D. A multicenter numerical integration scheme for polyatomic molecules. *J. Chem. Phys.* **1988**, *88*, 2547−2553.

(61) Treutler, O.; Ahlrichs, R. Efficient molecular numerical integration schemes. *J. Chem. Phys.* **1995**, *102*, 346−354.

(62) Laqua, H.; Kussmann, J.; Ochsenfeld, C. An improved molecular partitioning scheme for numerical quadratures in density functional theory. *J. Chem. Phys.* **2018**, *149*, 204111.

(63) *CUDA Toolkit 10.0*, see https://developer.nvidia.com/cuda-10.0-download-archive.

(64) *OpenCL-2.1*, see: https://www.khronos.org/opencl.

(65) *AMDGPU-Pro Driver 19.20*, see: https://www.amd.com.

(66) Obara, S.; Saika, A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1986**, *84*, 3963−3974.

(67) Obara, S.; Saika, A. General recurrence formulas for molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1988**, *89*, 1540−1559.

(68) *SymPy version 1.1.1*, see https://www.sympy.org.

(69) Head-Gordon, M.; Pople, J. A. A method for two-electron Gaussian integral and integral derivative evaluation using recurrence relations. *J. Chem. Phys.* **1988**, *89*, 5777−5786.

(70) *Intel C++ Compiler version 19.0.2.187*, see https://software.intel.com/c-compilers.

(71) *GNU compiler collection*, see http:/gcc.gnu.org.

(72) *Clang C++ Compiler version 4.0.0*, see https://clang.llvm.org.

(73) Maurer, S. A.; Lambrecht, D. S.; Flaig, D.; Ochsenfeld, C. Distance-dependent Schwarz-based integral estimates for two-electron integrals: Reliable tightness vs. rigorous upper bounds. *J. Chem. Phys.* **2012**, *136*, 144107.

(74) Curtiss, L. A.; Raghavachari, K.; Redfern, P. C.; Pople, J. A. Assessment of Gaussian-2 and density functional theories for the computation of enthalpies of formation. *J. Chem. Phys.* **1997**, *106*, 1063−1079.

(75) Gráfová, L.; Pitoňák, M.; Řezáč, J.; Hobza, P. Comparative Study of Selected Wave Function and Density Functional Methods for Noncovalent Interaction Energy Calculations Using the Extended S22 Data Set. *J. Chem. Theory Comput.* **2010**, *6*, 2365−2376.

(76) Sedlak, R.; Janowski, T.; Pitoňák, M.; Řezáč, J.; Pulay, P.; Hobza, P. Accuracy of Quantum Chemical Methods for Large Noncovalent Complexes. *J. Chem. Theory Comput.* **2013**, *9*, 3364−3374.

(77) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.

(78) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(79) Murphy, R. B.; Cao, Y.; Beachy, M. D.; Ringnalda, M. N.; Friesner, R. A. Efficient pseudospectral methods for density functional calculations. *J. Chem. Phys.* **2000**, *112*, 10131−10141.

(80) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; et al. General atomic and molecular electronic structure system. *J. Comput. Chem.* **1993**, *14*, 1347−1363.

(81) Gordon, M. S.; Schmidt, M. W. Advances in electronic structure theory: GAMESS a decade later. *Theory Appl. Comput. Chem.: First Forty Years* **2005**, 1167−1189.

(82) Friesner, R. A.; Bentley, J. A.; Menou, M.; Leforestier, C. Adiabatic-pseudospectral methods for multidimensional vibrational potentials. *J. Chem. Phys.* **1993**, *99*, 324−335.

(83) Cao, Y.; Beachy, M. D.; Braden, D. A.; Morrill, L.; Ringnalda, M. N.; Friesner, R. A. Nuclear-magnetic-resonance shielding constants calculated by pseudospectral methods. *J. Chem. Phys.* **2005**, *122*, 224116.

(84) Friesner, R. A.; Murphy, R. B.; Beachy, M. D.; Ringnalda, M. N.; Pollard, W. T.; Dunietz, B. D.; Cao, Y. Correlated ab Initio Electronic Structure Calculations for Large Molecules. *J. Phys. Chem. A* **1999**, *103*, 1913−1928.

(85) Parrish, R. M.; Hohenstein, E. G.; Schunck, N. F.; Sherrill, C. D.; Martínez, T. J. Exact tensor hypercontraction: a universal technique for the resolution of matrix elements of local finite-range N-body potentials in many-body quantum problems. *Phys. Rev. Lett.* **2013**, *111*, 132505.

Supporting information to the paper: Highly efficient, linear scaling seminumerical exact-exchange method for graphic processing units

Henryk Laqua, Travis H. Thompson, Jörg Kussmann, and Christian Ochsenfeld*

Department of Chemistry and Center for Integrated Protein Science (CIPSM), University of Munich (LMU), D-81377 München, Germany

E-mail: christian.ochsenfeld@uni-muenchen.de

1 Evaluation of floating point operation (FLOP) counts

For a dense matrix-matrix multiplications of the form

$$C_{ij} = \sum_{k} A_{ik} B_{kj} \tag{1}$$

the FLOP count is simply given as

$$N_{\text{FLOPs}}^{\text{BLAS-3}} = 2N_i N_j N_k, \tag{2}$$

conventionally counting fused-multiply-add operations as two FLOPs.

For completeness, we repeat eq. (6) of the main manuscript, involving the integral evaluation, here:

$$G_{\nu g} = \sum_{\lambda} w_g A_{\nu\lambda g} F_{\lambda g}. \tag{3}$$

The FLOP counts for the evaluation of the set of 3-center-1-electron integrals required for eq. (3) are computed for one shell-pair as

$$N^{\text{3c-1e}}_{\text{FLOPs}} = N_{\text{FLOPs for primitive integrals}} + N_{\text{FLOPs for primitive contractions}} + N_{\text{FLOPs for forming } G_{\nu g}} , \tag{4}$$

where

$$N_{\text{FLOPs for primitive integrals}} = N_{\text{primitive shellpairs}} N_{\text{FLOPs per primitive shellpair}} \tag{5}$$

denotes the amount of floating-point operations needed to compute the primitive integrals,

$$N_{\text{FLOPs for primitive contractions}} = N_{\text{primitive shellpairs}} N_{\text{Cartesian integrals}} \tag{6}$$

denotes the floating-point operation needed to sum all primitive integrals to form the contracted integrals,

$$N_{\text{Cartesian integrals}} = \frac{(l_1+1)(l_1+2)}{2} \frac{(l_2+1)(l_2+2)}{2} \tag{7}$$

denotes the amount of contracted Cartesian integrals, and

$$N_{\text{FLOPs for forming } G_{\nu g}} = 4 N_{\text{Cartesian Integrals}} \tag{8}$$

denotes the amount of floating point operations needed to multiply $F_{\lambda g}$ with $G_{\nu g}$. The pre-factor of 4 in eq. (8) is due to the exploitation of the integral-symmetry $A_{\nu\lambda g} = A_{\lambda\nu g}$, requiring two multiplications and two additions for each contracted integral.

The amount of FLOPs needed to form the primitive integrals was obtained by counting all additions, multiplications, subtractions and square-roots (the latter counted as four FLOPs) within the primitive loop. The result of that counting is given in Table S1.

Table S1: Number of FLOPs required to form the full set of primitive Cartesian integrals for a given l-quantum number combination (l1, l2).

l1/l2	0 (s)	1 (p)	2 (d)	3(f)	4 (g)
0 (s)	22	44	82	171	330
1 (p)	44	91	237	589	1271
2 (d)	82	237	547	1420	3147
3 (f)	171	589	1420	2714	6107
4 (g)	330	1271	3147	6107	10280

3.3 Publication III: Accelerating seminumerical Fock-exchange calculations using mixed single- and double-precision arithmetic

H. Laqua, J. Kussmann, C. Ochsenfeld

J. Chem. Phys. **154**, 214116 (2021).

Abstract

We investigate the applicability of single-precision (fp32) floating point operations within our linear-scaling, seminumerical exchange method sn-LinK [Laqua *et al.*, J. Chem. Theory Comput. **16**, 1456 (2020)] and find that the vast majority of the three-center-one-electron (3c1e) integrals can be computed with reduced numerical precision with virtually no loss in overall accuracy. This leads to a near doubling in performance on central processing units (CPUs) compared to pure fp64 evaluation. Since the cost of evaluating the 3c1e integrals is less significant on graphic processing units (GPUs) compared to CPU, the performance gains from accelerating 3c1e integrals alone is less impressive on GPUs. Therefore, we also investigate the possibility of employing only fp32 operations to evaluate the exchange matrix within the self-consistent-field (SCF) followed by an accurate one-shot evaluation of the exchange energy using mixed fp32/fp64 precision. This still provides very accurate ($1.8\,\mu E_\mathrm{h}$ maximal error) results while providing a sevenfold speedup on a typical “gaming” GPU (GTX 1080Ti). We also propose the use of incremental exchange-builds to further reduce these errors. The proposed SCF scheme (i-sn-LinK) requires only one mixed-precision exchange matrix calculation, while all other exchange-matrix builds are performed with only fp32 operations. Compared to pure fp64 evaluation, this leads to 4-7× speedups for the whole SCF procedure without any significant deterioration of the results or the convergence behavior.

Reproduced from:

H. Laqua, J. Kussmann, C. Ochsenfeld
“Accelerating seminumerical Fock-exchange calculations using mixed single- and double-precision arithmetic”
J. Chem. Phys. **154**, 214116 (2021).

with the permission of AIP Publishing.
`https://aip.scitation.org/doi/10.1063/5.0045084`

Accelerating seminumerical Fock-exchange calculations using mixed single- and double-precision arithmethic

Cite as: J. Chem. Phys. 154, 214116 (2021); doi: 10.1063/5.0045084
Submitted: 22 January 2021 • Accepted: 29 April 2021 •
Published Online: 7 June 2021

Henryk Laqua,[1,2] Jörg Kussmann,[1,2] and Christian Ochsenfeld[1,2,a)]

AFFILIATIONS

[1] Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU), D-81377 München, Germany
[2] Max Planck Institute for Solid State Research (MPI-FKF), 70569 Stuttgart, Germany

[a)] Author to whom correspondence should be addressed: christian.ochsenfeld@uni-muenchen.de

ABSTRACT

We investigate the applicability of single-precision (fp32) floating point operations within our linear-scaling, seminumerical exchange method sn-LinK [Laqua *et al.*, J. Chem. Theory Comput. **16**, 1456 (2020)] and find that the vast majority of the three-center-one-electron (3c1e) integrals can be computed with reduced numerical precision with virtually no loss in overall accuracy. This leads to a near doubling in performance on central processing units (CPUs) compared to pure fp64 evaluation. Since the cost of evaluating the 3c1e integrals is less significant on graphic processing units (GPUs) compared to CPU, the performance gains from accelerating 3c1e integrals alone is less impressive on GPUs. Therefore, we also investigate the possibility of employing only fp32 operations to evaluate the exchange matrix within the self-consistent-field (SCF) followed by an accurate one-shot evaluation of the exchange energy using mixed fp32/fp64 precision. This still provides very accurate (1.8 **μ***E*h maximal error) results while providing a sevenfold speedup on a typical "gaming" GPU (GTX 1080Ti). We also propose the use of incremental exchange-builds to further reduce these errors. The proposed SCF scheme (i-sn-LinK) requires only one mixed-precision exchange matrix calculation, while all other exchange-matrix builds are performed with only fp32 operations. Compared to pure fp64 evaluation, this leads to 4–7× speedups for the whole SCF procedure without any significant deterioration of the results or the convergence behavior.

I. INTRODUCTION

The evaluation of the exact (Fock)-exchange matrix usually represents the major computational bottleneck (typically >80% of the computation time) within Hartree–Fock or hybrid-density functional theory (DFT) calculations. Traditionally, this requires the computation of the four-center-two-electron (4c2e) integral-tensor leading to a formal $\mathcal{O}(\mathrm{N}^4)$ scaling, which is particularly problematic when combined with larger atomic-orbital (AO) basis sets, even if efficient screening techniques[1–10] are employed.

Therefore, seminumerical integration techniques, e.g., the pseudospectral method of Friesner *et al.*,[11–16] the chain-of-spheres-exchange (COS-X) method of Neese *et al.*,[17] the seminumerical methods of Plessow and Weigend,[18] Liu *et al.*,[19–21] and Kaupp and co-workers (the latter focusing more on local-hybrid functionals),[22–26] the semi-JK algorithm of Holzer,[27] and our sn-LinK method[28,29] are more efficient for large basis sets due to their superior $\mathcal{O}(M^3)$ formal scaling and their $\mathcal{O}(N_{\mathrm{bas}}^2)$ scaling with the basis set size. Moreover, as shown in Refs. 27 and 29, seminumerical integration is perfectly suited for modern, highly parallel hardware, where performance relies heavily on the use of single-instruction-multiple-data (SIMD) instructions, which applies to both modern central processing units (CPUs) and even to a greater extent to graphic processing units (GPUs).

By default, most quantum chemistry programs execute the necessary floating point operations with double numeric precision (fp64) due to its reliable accuracy (about 10^{-16} relative precision).[30] However, since most "gaming grade" GPUs typically provide significantly less computational performance for fp64 operations compared to single-precision floating point (fp32) operations (e.g., the fp32:fp64 ratio for the GTX 1080Ti is 32:1), the possibility of executing as many of the necessary computation using fp32 operations needs to be explored, despite its lower numerical precision of about 10^{-7}. In addition, the possibility of twofold speedups on CPUs also justifies the need for such a study even for pure CPU code. Investigations of pure fp32 or mixed fp32/fp64 execution have indeed already been carried out for the traditional 4c2e integral based Fock-exchange evaluation[31–36] and post-HF correlation methods,[32,37–39] but, to our knowledge, the applicability of reduced numerical precision within seminumerical integration has not yet been studied.

Therefore, we provide such a study in this work, which is organized as follows: First, we briefly review our seminumerical integration method sn-LinK from Ref. 29 in Sec. II. After reporting on the computational details in Sec. III, we explore the applicability of fp32 operations regarding performance and numerical stability in Sec. IV. This exploration is partitioned into three parts: In part 1 (Sec. IV A), we explore mixed fp32/fp64 evaluation of the three-center-one-electron (3c1e) integral tensor. In part 2 (Sec. IV B), we then explore the possibility of pure fp32 evaluation in all steps instead of only the 3c1e integral evaluation. Finally, in part 3 (Sec. IV C), we propose a specific self-consistent-field (SCF) method, denoted as i-sn-LinK, that employs incremental exchange-builds to reduce the numerical error from pure fp32 execution. Afterward, we illustrate the performance and accuracy of the so developed mixed-precision methods for a wide variety of molecules and basis sets in Sec. V and finally summarize our results in Sec. VI.

II. THEORY: SN-LINK REVIEWED

A. The seminumerical exchange method

Seminumerical integration decomposes the 4c2e integral tensor

$$(\mu\sigma|\nu\lambda) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \chi_\mu(\mathbf{r}_1)\chi_\sigma(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\chi_\nu(\mathbf{r}_2)\chi_\lambda(\mathbf{r}_2) \quad (1)$$

symmetrically as

$$(\mu\sigma|\nu\lambda) \approx \frac{1}{2}\left[\sum_g w_g \chi_\mu(\mathbf{r}_g)\chi_\sigma(\mathbf{r}_g)\int d\mathbf{r}\frac{\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|} + \sum_g w_g \int d\mathbf{r}\frac{\chi_\mu(\mathbf{r})\chi_\sigma(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}\chi_\nu(\mathbf{r}_g)\chi_\lambda(\mathbf{r}_g)\right], \quad (2)$$

employing numeric integration grids with grid points $\mathbf{r}_g$ and corresponding weights w_g.

Inserting this decomposition into the atomic orbital (AO) representation of the exact-exchange matrix leads to

$$K_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu\sigma|\nu\lambda) \quad (3)$$

$$\approx \frac{1}{2}\left[\sum_g w_g \sum_{\lambda\sigma}\chi_\mu(\mathbf{r}_g)\int \frac{\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|}d\mathbf{r} P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g) + \text{transpose}\right], \quad (4)$$

which is evaluated in three consecutive steps:

$$F_{\lambda g} = \sum_\sigma \chi_\sigma(\mathbf{r}_g)P_{\lambda\sigma}, \quad (5)$$

$$G_{\nu g} = \sum_\lambda w_g A_{\nu\lambda g}F_{\lambda g}, \quad (6)$$

$$K_{\mu\nu} = \sum_g \chi_\mu(\mathbf{r}_g)G_{\nu g}. \quad (7)$$

The so-obtained exchange-matrix is finally symmetrized to account for the transpose in Eq. (3).

If only the exchange energy

$$E_X = \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}P_{\lambda\sigma}(\mu\sigma|\nu\lambda) = tr(\mathbf{PK}) \quad (8)$$

is of interest (e.g., to compute the final energy after converging the SCF), the evaluation of Eq. (7) can be avoided and E_X can instead be obtained as

$$E_X = \sum_{\nu g} G_{\nu g}F_{\nu g}. \quad (9)$$

Equations (5) and (7) are evaluated with dense matrix–matrix multiplications employing batch-local matrices of asymptotically constant size (see Sec. 2.4 of Ref. 29 for details) utilizing highly optimized BLAS-3 libraries.

In contrast, evaluation of Eq. (6) requires the computation of the 3c-1e integral tensor

$$A_{\nu\lambda g} = \int \frac{\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}d\mathbf{r}, \quad (10)$$

which usually represents the most expensive step. Effective integral screening techniques are therefore essential for an efficient implementation.

B. Screening for 3c1 e-integrals reviewed

In order to assess the significance of a 3c1e-integral $A_{\nu\lambda g}$, we consider its contribution to the total exchange energy

$$\begin{aligned}\varepsilon^E_{\nu\lambda g} &= \left|w_g \sum_{\mu\sigma}\chi_\mu(\mathbf{r}_g)P_{\mu\nu}A_{\nu\lambda g}P_{\lambda\sigma}\chi_\sigma(\mathbf{r}_g)\right| \\ &= \left|w_g^{\frac{1}{2}}F_{\nu g}A_{\nu\lambda g}w_g^{\frac{1}{2}}F_{\lambda g}\right|\end{aligned} \quad (11)$$

and to the final exchange matrix

$$\begin{aligned}\varepsilon^K_{\nu\lambda g} = |w_g|\max\Bigg(&\sum_{\mu\sigma}|\chi_\mu(\mathbf{r}_g)\|P_{\mu\nu}\|A_{\nu\lambda g}\|\chi_\sigma(\mathbf{r}_g)|, \\ &\times \sum_{\mu\sigma}|\chi_\mu(\mathbf{r}_g)\|A_{\nu\lambda g}\|P_{\lambda\sigma}\|\chi_\sigma(\mathbf{r}_g)|\Bigg) \\ \le |w_g|&\max(|F_{\nu g}|,|F_{\lambda g}|)|A_{\nu\lambda g}|\sum_\mu|\chi_\mu(\mathbf{r}_g)|.\end{aligned} \quad (12)$$

An integral is then labeled as significant if either of the contributions is larger than a given threshold, i.e.,

$$\varepsilon^{E}_{\nu\lambda g} \geq \vartheta_E \quad \text{or} \quad \varepsilon^{K}_{\nu\lambda g} \geq \vartheta_K. \tag{13}$$

For optimal performance, this selection is performed for whole batches of grid points at once, employing distance-independent (and therefore grid-point independent) integral bounds for $A_{\nu\lambda g}$,[10] as detailed in Secs. 2.2 and 3 of Ref. 29.

III. COMPUTATIONAL DETAILS

Unless stated otherwise, all calculations are performed with our FermiONs++ quantum chemistry program,[7,8] employing the "Karlsruhe" ("def2-") basis sets[40] (the prefix "def2-" is omitted for simplicity) and the "gm4" multi-grid[41] (~5000 points per atom within the SCF and ~15 000 points per atom for the final energy calculation). For yttrium, the respective effective core potential (ECP)[40] was employed to replace core electrons. For the evaluation of the Coulomb-interaction, the resolution-of-the-identity (RI-J) approximation[42–44] in combination with the RI-J-optimized basis set of Ref. 45 was used throughout this work. These settings are chosen to represent typical applications.

Since the computation of the exact-exchange contributions is usually the most expensive step within hybrid-DFT calculations, the discussion for Hartree–Fock calculations within this work is equally meaningful for hybrid-DFT calculations. Indeed, since only a fraction of exact Fock-exchange is employed in hybrid-DFT methods, the numerical errors from single-precision execution are proportionally lower in this case. We provide analogous results to Tables IX and XI using the PBE0 hybrid functional[46] instead of the Hartree–Fock method within the supplementary material.

For optimal performance, the CPU code was compiled with the Intel C++ compiler (ICPC) version 19.1.0[47] with all compiler-optimization enabled ("-Ofast," "-march = native"), which is necessary to fully utilize SIMD-vector-instructions within the 3c1e integral kernels. The GPU code was compiled with NVCC-10.1 (CUDA-10.1),[48] also employing all possible compiler-optimizations ("-O3" and "-use fast math"). To provide sufficient parallel workload for each device, grid-batches of 512 grid points on CPUs, 20 480 points on the NVIDIA GV100 GPU, and 10 240 points on the 1080Ti GPU are employed.

SCF convergence is always measured by the root mean square of the DIIS error matrices ($\mathbf{SPF}$–$\mathbf{FPS}$).[49,50] Unless stated otherwise, all timings are given for one full exchange-matrix build averaged over all but the very first SCF cycles, since the very first Fock-build is considerably faster due to the sparsity of the superposition-of-atomic-densities (SAD) guess density matrix. For rigorous comparisons, errors in the converged energy (denoted ΔE), the root mean square deviation (RMSD) of the converged density matrix (denoted as $\Delta\mathbf{P}$), and the root mean square deviation of the converged nuclear forces (denoted as ΔForces) are always referenced to pure fp64-execution with all other settings being identical.

The test geometries[51,52] in this work are chosen to represent typical applications. In particular, A–T DNA-fragments [$(\mathrm{DNA})_x$] exemplify biochemistry applications, spherical water balls [$(H_2O)_{68}$], explicit solvent environments, LiF cutouts [$(\mathrm{LiF})_{36}$], materials science applications, and $Y_2C_5H_{20}N_{20}O_{13}$ inorganic complex chemistry. All geometries employed in this work are provided in the supplementary material.

IV. THE MIXED FP32/FP64 PRECISION SN-LINK METHOD

In this section, the possibility of employing single-precision algebra is systematically studied, focusing on the speedups and accuracy. The total execution times corresponding to the here presented speedups are given in Sec. 2 of the supplementary material.

A. Part 1: Mixed precision evaluation of the 3c1e integral tensor

The evaluation and contraction of the 3c1e-integral tensor [Eq. (6)] usually represents the most time-consuming step within sn-LinK, particularly on CPUs and for smaller systems. Therefore, we first investigate the applicability of fp32 operations for this step.

We can assign each 3c1e-integral (or batch of integrals) an upper bound to its contribution to the final exchange-energy [Eq. (11)] and exchange-potential [Eq. (12)]. These upper bounds not only enable the linear-scaling evaluation of the 3c1e integrals via computing only the significant subset of the tensor but also enable a finer-grained partitioning of the 3c1e tensor into three instead of only two categories:

- Category one contains the most significant 3c1e integrals ($\varepsilon^{E/K}_{\nu\lambda g} \geq \vartheta^{\text{fp64}}_{E/K}$). These are computed with fp64 operations.
- Category two contains all 3c1e integrals that are too significant to be completely neglected, but not so significant that they require fp64 evaluation ($\vartheta^{\text{fp64}}_{E/K} > \varepsilon^{E/K}_{\nu\lambda g} \geq \vartheta^{\text{fp32}}_{E/K}$). These are computed and accumulated with fp32 operations.
- Category three contains all the insignificant integrals ($\varepsilon^{E/K}_{\nu\lambda g} < \vartheta^{\text{fp32}}_{E/K}$). These are not computed at all, even with pure fp64-execution.

Compared to the original sn-LinK from Ref. 29, we just split the set of significant integrals into two subsets (category one and two), resulting in only a slight adjustment of the pseudocode for the 3c1e-integral evaluation, which is given in Fig. 1.

The complete discarding of all category three integrals (line 3 of Fig. 1) is just standard integral screening, so we can employ the same "parent" thresholds as in the original sn-LinK, i.e., $\vartheta^{\text{fp32}}_{K} = \vartheta_K = 10^{-8}$, $\vartheta^{\text{fp32}}_{E} = \vartheta_E = 10^{-11}$ within the SCF, and $\vartheta^{\text{fp32}}_{E} = \vartheta_E = 10^{-12}$ in the final energy calculation. The screening-errors from these thresholds are usually well below 1 $\boldsymbol{\mu}E_\mathrm{h}$. This integral screening is not altered within this work, and instead, the speedups presented here are solely caused by a more efficient handling of the category two integrals by using a lower precision arithmetic.

We decided to choose our second set of thresholds, which distinguish between fp32- and fp64-execution (l.5 and 8 of Fig. 1) $\vartheta^{\text{fp64}}_{E/K}$ dynamically, based on the original thresholds $\vartheta_{E/K}$. Indeed, since the relative numerical precision of fp32 numbers is about 10^{-7}, one would naturally choose $\vartheta^{\text{fp64}}_{E/K} = 10^{7}\vartheta_{E/K}$. In Table I, we test a variety of threshold-multipliers and find that we can employ even

```
Copy fp64-$F_{\lambda g}$ to fp32-$F_{\lambda g}$
for all sign. shell-pairs $\nu\lambda$ do
    if shell-pair $\nu\lambda$ is not in category three then
        if shell-pair $\nu\lambda$ is in category one then                      ▷ fp64 execution
            for all grid points in (sub-)batch do
                Compute integrals fp64-$A_{\nu\lambda g}$, multiply with fp64-$F_{\lambda g}$, and add onto fp64-$G_{\nu g}$
            end for
        else if shell-pair $\nu\lambda$ is in category two then                 ▷ fp32 execution
            for all grid points in (sub-)batch do
                Compute integrals fp32-$A_{\nu\lambda g}$, multiply with fp32-$F_{\lambda g}$, and add onto fp32-$G_{\nu g}$
            end for
        end if
    end if
end for
Add fp32-$G_{\nu g}$ to fp64-$G_{\nu g}$
```

FIG. 1. Evaluation of Eq. (6) (3c1e integral evaluation).

tighter fp64-thresholds, since still only a very small fraction of 3c1e-integrals need to be computed with fp64 operations. Therefore, we decided to use a fp64 threshold multiplier of 10^5 for all further calculations, providing virtually perfect accuracy regarding both the converged energy and the converged density matrix, while over 97% of all integrals are computed using fp32 instructions. As expected, this results in nearly 2× speedup for the evaluation of the 3c1e integrals.

We further evaluate the robustness and possible speedups of mixed fp32/fp64 3c1e-integral evaluation for different molecular systems and basis sets in Table II. Mixed precision evaluation provides essentially error-free results for both the converged energy and the density matrix and converges within exactly the same amount of SCF cycles. That is, mixed-precision evaluation of the 3c1e integrals is essentially error-free while providing close to 2× speedups (1.73–1.97×) for the 3c1e integral evaluation, in line with the expected 2× performance gains for CPUs.

As presented in Sec. 1 of the supplementary material, this speedup is only achieved if vector-instructions (AVX2 in this work) are utilized. This matches our expectations, since one 256-bit vector instruction processes four fp64 but eight fp32 values, whereas one scalar instruction always processes 1 number, regardless of the precision.

TABLE I. Impact of the fp64 multiplier on the accuracy and performance for $(DNA)_4$/HF/SVP. Speedups are given for two Intel Xeon E5-2630 CPUs (20 cores@2.20 GHz) using AVX-2 instructions. The corresponding thresholds for switching between fp32- and fp64 evaluation $\vartheta_{E/K}^{\mathrm{fp64}}$ derived from the "parent" screening-thresholds $\vartheta_E = 10^{-11}$ and $\vartheta_K = 10^{-8}$ are given for context.

fp64-multiplier	10^{12}	10^{10}	10^{8}	10^{7}	10^{6}	10^{5}	10^{4}
$\vartheta_K^{\mathrm{fp64}}$	10^{4}	10^{2}	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$\vartheta_E^{\mathrm{fp64}}$	10^{1}	10^{-1}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Energy error (μEh)	34.19	0.55	0.08	0.01	<0.01	<0.01	<0.01
$\Delta\mathbf{P}$ (10^{-6})	0.20	0.20	0.15	0.04	0.01	<0.01	<0.01
% of integrals fp64	0.000	0.001	0.083	0.37	1.2	3.0	7.9
Speedup (only integrals)	1.99×	1.98×	1.96×	1.95×	1.94×	1.89×	1.79×

TABLE II. Comparison of mixed fp32/fp64 3c1e integral evaluation and pure fp64 evaluation. Speedups for only the 3c1e integrals and one whole exchange build are given for two Intel E5-2630 CPUs (20 cores @2.20 GHz) using AVX-2 instructions. The number of SCF cycles is given for a convergence threshold of $\vartheta_{\text{conv}} = 10^{-8}$ for the DIIS-error.

System/Basis	Speedup 3c1e	Speedup (K)	Error (μEh)	$\Delta\mathbf{P}$ (10^{-6})	n_{iter} (fp64)	n_{iter} (mp)
$(DNA)_1$/SVP	1.73×	1.55×	<0.01	<0.01	15	15
$(DNA)_4$/SVP	1.86×	1.62×	<0.01	<0.01	14	14
$(H_2O)_{68}$/SVP	1.92×	1.61×	<0.01	<0.01	11	11
$(LiF)_{36}$/SVP	1.90×	1.73×	<0.01	0.19	11	11
$Y_2C_5H_{20}N_{20}O_{13}$/SVP	1.79×	1.62×	<0.01	<0.01	15	15
$(DNA)_1$/TZVP	1.87×	1.69×	<0.01	0.03	14	14
$(DNA)_4$/TZVP	1.87×	1.68×	<0.01	0.04	14	14
$(H_2O)_{68}$/TZVP	1.88×	1.58×	<0.01	<0.01	11	11
$(LiF)_{36}$/TZVP	1.97×	1.74×	<0.01	0.18	10	10
$Y_2C_5H_{20}N_{20}O_{13}$/TZVP	1.91×	1.72×	<0.01	0.01	14	14
$(DNA)_1$/QZVP	1.80×	1.65×	<0.01	0.07	14	14
$(H_2O)_{68}$/QZVP	1.78×	1.53×	<0.01	<0.01	10	10

However, the 2×-speedup for the 3c1e integral evaluation does not perfectly translate to the full K-build, since the other two steps [Eqs. (5) and (7)] have not been accelerated. As depicted in Table III, this effect is even more pronounced with GPU evaluation because the 3c1e integral evaluation contributes less to the total runtime on GPUs (compare also with the discussion in Sec. 5.2.1 of Ref. 29). For this reason, only minor speedups from employing the mixed precision evaluation of the 3c1e integrals are obtained for the GV100 GPU (0.96–1.79×) and the speedups on the GTX 1080Ti (1.6× to 2.5×) are far away from the theoretical value of 32×.

To summarize part 1, the 3c1e integral tensor is partitioned so that only the most significant contributions are evaluated with fp64 operations. The introduced errors are negligible while providing nearly 2× speedups in the integral-evaluation part. However, when employing GPUs with low fp64-performance, the BLAS-3 steps [Eqs. (5) and (7)] also need to be performed with fp32 operations to achieve optimal performance.

TABLE III. Speedups for one full K-build from employing mixed-precision kernels for 3c1e-integral evaluation on two Intel E5-2630 CPUs (20 cores@2.20 GHz) using AVX-2 instructions on one NVIDIA GV100 GPU (fp32:fp64 ratio 2:1) and one GTX 1080Ti GPU (fp32:fp64 ratio 32:1).

System/Basis	CPU	GV100	1080Ti
$(DNA)_4$/SVP	1.55×	1.01×	1.61×
$(DNA)_1$/SVP	1.62×	1.21×	1.95×
$(H_2O)_{68}$/SVP	1.61×	0.96×	1.83×
$(LiF)_{36}$/SVP	1.73×	1.03×	2.51×
$Y_2C_5H_{20}N_{20}O_{13}$/SVP	1.62×	0.97×	1.94×
$(DNA)_1$/TZVP	1.69×	1.22×	2.19×
$(DNA)_4$/TZVP	1.68×	1.42×	2.05×
$(H_2O)_{68}$/TZVP	1.58×	1.22×	1.70×
$(LiF)_{36}$/TZVP	1.74×	1.21×	2.04×
$Y_2C_5H_{20}N_{20}O_{13}$/TZVP	1.72×	1.34×	2.00×
$(DNA)_1$/QZVP	1.65×	1.79×	2.14×
$(H_2O)_{68}$/QZVP	1.53×	1.44×	1.73×

B. Part 2: Pure fp32 evaluation

In contrast to the integral evaluation, computing the BLAS-3 steps [Eqs. (5) and (7)] with mixed precision instruction is not straightforward because they are best evaluated with a single call of a highly optimized dense linear algebra routine for optimal performance. Although algorithms for mixed precision linear-algebra have been proposed in the literature,[53] we prefer to, if possible, avoid them for the sake of simplicity.

Therefore, the possibility of pure fp32 evaluation, i.e., evaluation of Eqs. (5)–(7) with only fp32 operations and accumulation of the batch-local fp32-**K**-matrices into a global fp64-**K**-matrix, needs to be explored. Note that in the final energy calculation, Eqs. (5) and (6) are evaluated with single-precision, but the accumulation in Eq. (9) is performed with double-precision.[54]

As depicted in Table IV, pure fp32 evaluation results in up to 7.4× speedups compared to pure fp64 execution. This speedup is still significantly less than the theoretically possible 32× speedups because the code is not purely limited by the floating point throughput alone and the demand for memory bandwidth and local storage (cache) is only reduced by a factor of two.

More important, however, is the fact that pure fp32 execution leads to a significant deterioration of the accuracy, with errors up to 90 μEh. In contrast, the exchange *matrix* from pure fp32 evaluation is much more accurate than the errors in the *energy* indicate. That is, even though converging the SCF using fp32 only can lead to quite significant errors in the converged density matrix of up to 4×10^{-5} a.u., computing the final energy from this inaccurate matrix employing mixed precision integral evaluation instead of pure fp32-execution (denoted as "fp32*") leads to significantly smaller errors (1.8 μEh at most) while providing the performance of full fp32 execution (i.e., up to 7× speedups) within the SCF.

The situation is analogous to adaptive integration grids, i.e., employing three times coarser grids within the SCF than for the final energy calculation provides virtually the same result as performing the whole calculation with the large grid.[41] That is, the exchange *energy* is generally more sensitive to numeric errors than the exchange *matrix*.

TABLE IV. Speedups (averaged over all but the very first K-builds) of mixed precision integral evaluation (mp) and full fp32 evaluation (fp32) on GTX 1080Ti compared to pure fp64-execution. Errors of the converged energy from pure fp32-evaluation (ΔE_{fp32}) and from fp32-evaluation in the SCF followed by a post-SCF mixed precision energy evaluation ($\Delta E_{\text{fp32*}}$) and the root-mean-square error of the converged density matrix ($\Delta\mathbf{P}$) are given referenced to pure fp64 execution.

System/Basis	Speedup (mp)	Speedup (fp32)	ΔE_{fp32} (μEh)	$\Delta E_{\text{fp32*}}$ (μEh)	$\Delta\mathbf{P}$ (10^{-6})
$(\text{DNA})_1$/SVP	1.61×	5.14×	14.2	<0.01	0.40
$(\text{DNA})_4$/SVP	1.95×	6.38×	68.3	0.08	2.56
$(\text{H}_2\text{O})_{68}$/SVP	1.83×	5.78×	20.1	<0.01	0.05
$(\text{LiF})_{36}$/SVP	2.51×	5.45×	89.3	<0.01	6.26
$\text{Y}_2\text{C}_5\text{H}_{20}\text{N}_{20}\text{O}_{13}$/SVP	1.94×	6.39×	15.8	<0.01	0.18
$(\text{DNA})_1$/TZVP	2.19×	7.12×	12.2	0.17	39.44
$(\text{DNA})_4$/TZVP	2.05×	7.42×	85.4	1.82	26.79
$(\text{H}_2\text{O})_{68}$/TZVP	1.70×	6.49×	27.9	0.01	0.24
$(\text{LiF})_{36}$/TZVP	2.04×	5.65×	19.8	0.09	15.94
$\text{Y}_2\text{C}_5\text{H}_{20}\text{N}_{20}\text{O}_{13}$/TZVP	2.00×	6.94×	19.9	0.08	5.52
$(\text{DNA})_1$/QZVP	2.14×	6.27×	5.3	1.32	31.83
$(\text{H}_2\text{O})_{68}$/QZVP	1.73×	7.32×	14.5	0.33	1.89

However, as presented in Table V, executing all **K**-builds within the SCF with pure fp32 arithmetic deteriorates the SCF convergence behavior measurably, meaning that very tight convergence, e.g., to 10^{-8}, is impossible due to the increase in numerical fluctuations. Although the level of precision is sufficient for many applications (e.g., *ab initio* molecular dynamics), a truly error-free method similar to part 1 (i.e., no measurable change in the SCF convergence behavior and as little change in the converged energy and density matrix as possible) is our goal in the following.

TABLE V. Number of SCF steps to achieve a given SCF convergence ϑ_{conv} quantified by the DIIS error. "n.c." denotes non-converged SCF.

System/Basis	ϑ_{conv}	fp64	mp	fp32
$(\text{DNA})_1$/SVP	10^{-6}	9	9	9
	10^{-8}	15	15	n.c.
$(\text{DNA})_4$/SVP	10^{-6}	9	9	9
	10^{-8}	14	14	n.c.
$(\text{H}_2\text{O})_{68}$/SVP	10^{-6}	7	7	7
	10^{-8}	11	11	11
$(\text{LiF})_{36}$/SVP	10^{-6}	7	7	7
	10^{-8}	10	11	n.c.
$\text{Y}_2\text{C}_5\text{H}_{20}\text{N}_{20}\text{O}_{13}$/SVP	10^{-6}	9	9	9
	10^{-8}	15	15	n.c.
$(\text{DNA})_1$/TZVP	10^{-6}	9	9	9
	10^{-8}	15	15	n.c.
$(\text{DNA})_4$/TZVP	10^{-6}	9	9	9
	10^{-8}	14	14	n.c.
$(\text{H}_2\text{O})_{68}$/TZVP	10^{-6}	7	7	7
	10^{-8}	12	12	12
$(\text{LiF})_{36}$/TZVP	10^{-6}	7	7	7
	10^{-8}	10	10	n.c.
$\text{Y}_2\text{C}_5\text{H}_{20}\text{N}_{20}\text{O}_{13}$/TZVP	10^{-6}	9	9	9
	10^{-8}	14	14	n.c.
$(\text{DNA})_1$/QZVP	10^{-6}	8	8	8
	10^{-8}	13	13	n.c.
$(\text{H}_2\text{O})_{68}$/QZVP	10^{-6}	7	7	7
	10^{-8}	10	10	n.c.

C. Part 3: Incremental K-builds (i-sn-LinK)

The idea of incremental Fock/exchange-builds, i.e., computing $\mathbf{K}[\Delta\mathbf{P}]$ and then incrementally updating **K** instead of always recomputing the full exchange matrix, was initially proposed by Almloef *et al.*[55] and later improved by Haeser and Ahlrichs[56] to allow for tighter density-matrix-based screening of the 4c2e-integrals within the SCF. However, because only small increments to **K** are computed, incremental Fock-builds have also been shown to reduce the numerical error introduced by fp32-evaluation, since the absolute error is proportional to the magnitude of the contribution.[31] In part 2 (Sec. IV B), we showed that pure fp32 exchange-builds are already quite accurate and allow for unproblematic convergence to about 10^{-6}. Therefore, the possibility to perform the last exchange-builds incrementally should be explored.

We thus propose i-sn-LinK, a special SCF scheme given in Fig. 2. First, the SCF is converged to a relative loose threshold with non-incremental fp32 K-builds. Since the difference density is comparatively large for these early SCF steps, there are no significant performance gains from incremental updates in these steps. In this work, we choose a convergence criterion of 10^{-5} to ensure that convergence to this point is always reached.

Then, one full K-build is performed with mixed fp32/fp64 precision (i.e., employing part 1) and the SCF is subsequently converged further with incremental fp32 K-builds. These incremental K-builds are always built from the difference density matrix $\Delta\mathbf{P}$ referenced to

```
while RMSD$\big(\mathbf{FPS}-\mathbf{SPF}\big) > 10^{-5}$ do          ▷ only loose convergence
    Full K-builds: $\mathbf{K}=\mathbf{K}[\mathbf{P}]$          ▷ fp32 3c1e-integrals, fp32-BLAS-3
end while
One full K-build: $\mathbf{K}_{\text{ref}}=\mathbf{K}[\mathbf{P}_{\text{ref}}]$          ▷ mixed-precision 3c1e-integrals, fp64-BLAS-3
while RMSD$\big(\mathbf{FPS}-\mathbf{SPF}\big) > \vartheta_{\text{conv}}$ (e.g., $10^{-8}$) do          ▷ tight convergence
    Incremental K-builds: $\mathbf{K}=\mathbf{K}_{\text{ref}}+\mathbf{K}[\mathbf{P}-\mathbf{P}_{\text{ref}}]$          ▷ fp32 3c1e-integrals, fp32-BLAS-3
end while
Compute energy $E_X[\mathbf{P}]$ (eq. (9))          ▷ mixed-precision 3c1e-integrals, fp64-BLAS-3
```

FIG. 2. Self-consistent field algorithm in i-sn-LinK.

the last full K-build, not to the previous density matrix. This removes the possibility of incremental error accumulation, further improving the numerical stability in this way.[57] In order to reduce errors from the density-matrix including integral screening when operating on incremental matrices, we employ two orders of magnitude tighter screening thresholds $\vartheta_{E/K}$ for these incremental builds, which, due to the elements of $\Delta\mathbf{P}$ being much smaller than $\mathbf{P}$, leads to a similar performance as full **K**-builds with looser thresholds. The final energy (also optionally nuclear forces) is then computed using mixed fp32/fp64 precision, analogous to part 2.

In contrast to the works of Almloef *et al.*[55] and Haeser and Ahlrichs,[56] the aim of our incremental scheme i-sn-LinK is only to improve the convergence behavior while using as many fp32 K-builds as possible, not to improve the tightness of the integral-selection.

The performance, accuracy, and SCF convergence of i-sn-LinK are presented in Table VI. Overall, we obtain virtually the same accuracy and numerical stability as for pure fp64 execution while obtaining up to 5.2× speedups (averaged over the whole calculation). As expected, the speedups are higher if more SCF cycles need to be performed because the cost of the two remaining mixed-precision K-builds, which comprise 36%–47% of the computation time, is comparatively less impactful in this situation.

However, the most significant result of Table VI is the fact that i-sn-LinK nearly always converges within the same amount of SCF cycles as pure fp64 execution, proving the numerical stability of the method.

D. Parts 1 , 2, and 3 in comparison

In order to compare the different approaches discussed in Secs. IV A–IV C, a brief summary of them is given in Table VII. In this context, the mixed-precision 3c1e integral method ("mp") of part 1 represents the simplest introduction of reduced numerical precision; that is, only the 3c1e integral evaluation part [Eq. (6)] is

TABLE VI. Comparison of i-sn-LinK compared to non-incremental pure fp64 evaluation. Speedups on the GTX 1080Ti are given for the sum of all exchange-builds including the final exchange energy calculation. The percentage of the two mixed-precision K-builds to the total time for all exchange-builds (%fp64), the error of the converged energy of i-sn-LinK ΔE, and the RMSD of the converged density matrix $\Delta\mathbf{P}$ are referenced to full fp64 evaluation, and the number of SCF cycles for $\vartheta_{\text{conv}} = 10^{-8}$ is given for reference.

System/Basis	Speedup	%fp64 (%)	ΔE (μEh)	$\Delta\mathbf{P}$ (10^{-6})	n_{iter} (fp64)	n_{iter} (i-sn-LinK)
$(\text{DNA})_1$/SVP	4.81×	42	<0.01	0.01	15	15
$(\text{DNA})_4$/SVP	5.03×	40	<0.01	0.01	14	14
$(\text{H}_2\text{O})_{68}$/SVP	4.59×	46	<0.01	0.00	11	11
$(\text{LiF})_{36}$/SVP	4.27×	45	<0.01	0.65	10	10
$\text{Y}_2\text{C}_5\text{H}_{20}\text{N}_{20}\text{O}_{13}$/SVP	4.93×	45	<0.01	0.00	15	15
$(\text{DNA})_1$/TZVP	4.97×	42	<0.01	0.15	15	15
$(\text{DNA})_4$/TZVP	5.19×	39	0.03	0.29	14	14
$(\text{H}_2\text{O})_{68}$/TZVP	4.67×	47	<0.01	0.00	11	11
$(\text{LiF})_{36}$/TZVP	4.40×	42	0.01	0.37	10	10
$\text{Y}_2\text{C}_5\text{H}_{20}\text{N}_{20}\text{O}_{13}$/TZVP	4.99×	45	<0.01	0.02	14	14
$(\text{DNA})_1$/QZVP	4.42×	36	0.01	0.29	13	14
$(\text{H}_2\text{O})_{68}$/QZVP	4.62×	47	<0.01	0.01	10	10

TABLE VII. Comparison of the methods in parts 1, 2, and 3 within the SCF and for final energy-build (final). "3c1e" denotes numerical precision in the evaluation of Eq. (6), "BLAS-3" denotes precision in the evaluation of Eqs. (5) and (7).

Method	3c1e (SCF)	3c1e (final)	BLAS-3 (SCF)	BLAS-3 (final)
fp64	fp64	fp64	fp64	fp64
mp (part1)	fp32 + fp64	fp32 + fp64	fp64	fp64
fp32 (part2)	fp32	fp32	fp32	fp32
fp32* (part2)	fp32	fp32 + fp64	fp32	fp64
i-sn-LinK (part3)	fp32 (fp32 + fp64)[a]	fp32 + fp64	fp32 (fp64)[a]	fp64

[a] Higher precision in one non-incremental Fock-build.

adjusted. This approach has virtually no impact on the SCF convergence behavior or the accuracy of the final result. Since the 3c1e integral evaluation is by far the most significant bottleneck for CPU execution, accelerating the other steps cannot provide significant speedups. Hence, we recommend to only employ the "mp" approach on CPUs.

On GPUs, however, the BLAS-3 steps with fp64 execution can become significant. Therefore, we investigated the possibility of pure fp32-execution in Sec. IV B (part 2) and found that the so-converged SCF can yield surprisingly accurate energies, if the final energy was evaluated with higher precision. However, the convergence behavior and the quality of the converged density matrix were measurably deteriorated.

These shortcomings were then addressed in Sec. IV C (i-sn-LinK; part 3) by introducing incremental **K**-builds, which update a once-computed high-precision **K**-matrix. This recovers the accuracy and stable convergence behavior of the "mp" approach, but only one non-fp32 **K**-build has to be performed within the SCF. The final energy is, of course, never computed with pure single-precision, since this leads to unacceptably large errors (cf. "fp32" in Table IV). Because the i-sn-LinK scheme represents the best trade-off between accuracy and performance on GPUs, i.e., close to pure-fp32 performance and essentially pure-fp64 accuracy, we recommend the i-sn-LinK method for GPUs.

V. ILLUSTRATIVE CALCULATIONS

A. SCF convergence for difficult electronic structures

Since deterioration in SCF convergence stability was the main reason to develop the i-sn-LinK method of part 3 (Sec. IV C) and since it was also shown to be problematic in other works on single-precision execution,[31–36] we provide a more detailed investigation on difficult molecules regarding SCF stability in Table VIII. In order to exemplify the worst-case scenario, we selected the ten molecules with the worst convergence behavior of the ASCDB benchmark set[58] and employ the def-TZVPPD basis set, which, due to the additional diffuse basis-functions, requires even higher numerical precision.

The results verify that both the mixed precision 3c1e evaluation and i-sn-LinK have no impact on the stability of the SCF convergence, even when considering very difficult electronic structures. Furthermore, the largest error of 0.83 μE_h occurs for the most difficult structure (*cis*HO_3) and is still smaller than the SCF convergence criterion of 10^{-6}.

B. Performance for large systems and large basis sets

To illustrate the practical applicability of the mixed-precision methods developed in this work, a range of systems are tested with up to quadruple-ζ basis sets in Table IX.

TABLE VIII. Number of SCF steps (N_{iter}) and error in the converged SCF energy of full-fp64 execution, mixed-precision integral evaluation (mp), and i-sn-LinK for the ten most difficult molecules of the ASCDB database.

		fp64	mp		i-sn-LinK	
Molecule	ϑ_{conv}	N_{iter}	N_{iter}	ΔE (μEh)	N_{iter}	ΔE(μEh)
cisHO_3	10^{-6}	32	32	0.06	32[a]	0.83
$CrCl_2$	10^{-6}	11	11	0.02	11[a]	0.10
$H_6LiB_3OMg_2AlSi_2$	10^{-7}	28	28	<0.01	28[a]	0.04
VO	10^{-7}	25	25	<0.01	25[b]	<0.01
$C_2H_5N_4O_2MgS_2$	10^{-7}	23	23	<0.01	23[a]	<0.01
transHO_3	10^{-7}	23	23	<0.01	23[a]	<0.01
ClOO	10^{-7}	23	23	<0.01	23[a]	<0.01
$C_3H_6B_3NAlSi_2$	10^{-7}	22	22	0.01	22[a]	0.08
FOO	10^{-7}	22	22	<0.01	22[a]	<0.01
OHCl	10^{-7}	20	20	<0.01	20[a]	<0.01

[a] Incremental K-builds starting at 10^{-4}.
[b] Incremental K-builds starting at 10^{-3}.

TABLE IX. Comparison of mixed-precision 3c1e integral evaluation (mp) and i-sn-LinK in terms of the speed of convergence [number of SCF steps to reach RMSD(**FPS** − **SPF**) > 10^{-7}] and in terms of the accuracy of the converged energy (ΔE), the root mean square deviation of the converged density matrix ($\Delta\mathbf{P}$), and the root mean square deviation of the nuclear forces (ΔForces).

		$N_{\text{iter}}(10^{-7})$			ΔE (μEh)		$\Delta\mathbf{P}$ (10^{-6})		ΔForces (μEh $a_0{}^{-1}$)	
Molecule	Basis	fp64	mp	i-sn-LinK	mp	i-sn-LinK	mp	i-sn-LinK	mp	i-sn-LinK
Taxol	6-31G*	12	12	12	<0.01	<0.01	<0.01	<0.01	<0.01	0.03
Valinomycin	6-31G*	12	12	12	<0.01	<0.01	<0.01	<0.01	<0.01	0.06
Olestra	6-31G*	11	11	11	<0.01	<0.01	<0.01	<0.01	<0.01	0.05
Fullerene C_{60}	TZVP	10	10	11	<0.01	<0.01	0.48	10.36	0.03	0.20
Fullerene C_{60}	QZVP	10	10	10	−0.01	0.01	0.34	8.00	0.04	0.67
$(S_8)_{20}$	TZVP	10	10	10	<0.01	<0.01	<0.01	0.01	<0.01	0.20
Crambin	TZVP	10	10	10	0.01	0.05	0.09	0.31	0.01	0.20
$(DNA)_1$	SVP	12	12	12	<0.01	<0.01	<0.01	0.01	<0.01	0.10
$(DNA)_4$	SVP	12	12	12	<0.01	<0.01	<0.01	0.03	<0.01	0.13
$(DNA)_{16}$	SVP	11	11	11	<0.01	−0.02	<0.01	0.03	0.05	0.14
$(DNA)_1$	TZVP	12	12	12	<0.01	<0.01	0.01	0.25	<0.01	0.20
$(DNA)_4$	TZVP	11	11	11	<0.01	0.05	0.02	0.39	<0.01	0.15
$(DNA)_1$	QZVP	11	11	11	<0.01	<0.01	0.03	0.32	<0.01	0.12
$(DNA)_4$	QZVP	10	10	10	<0.01	−0.03	0.05	0.60	0.01	0.40
$Y_2C_5H_{20}N_{20}O_{13}$	TZVP	11	11	12	<0.01	<0.01	<0.01	0.04	<0.01	0.15
$Y_4C_{10}H_{42}N_{40}O_{27}$	TZVP	12	12	12	<0.01	<0.01	<0.01	0.02	<0.01	0.26

First, the reliability and accuracy of the two methods is proven again, i.e., nearly all molecules converge within the same amount of SCF steps (10–12 steps) regardless of the method used for the Fock-exchange. Furthermore, the converged energies match the fp64 results to <0.1 μE_{h} even in the case of heavy elements (e.g., yttrium-complexes in Table XI), the nuclear forces are accurate within 0.7 μE_{h} $a_0{}^{-1}$, and the converged density matrices are accurate within 10^{-6} a.u. except for C_{60}.

For the C_{60}-Fullerene, larger deviations (~ 10^{-5} a.u.) are, however, not directly due to fp32-execution but are instead caused by the slightly different integral screening in i-sn-link, where $\Delta\mathbf{P}$ is employed instead of $\mathbf{P}$ within the incremental updates. This is supported by Table X, where tighter thresholds are employed for both the fp64-reference and the i-sn-LinK calculations in order to remove the effect of the integral screening. The remaining deviations are then only caused by the reduced numerical precision in i-sn-LinK. These remaining errors are significantly smaller for the most challenging molecules such as C_{60} or $(DNA)_4$/QZVP, proving that those larger deviations in Table IX were indeed caused by the altered integral screening and not by the reduced numerical precision.

Summarizing the accuracy comparisons, the two tested mixed-precision methods "mp" and i-sn-LinK lead to essentially negligible errors ($\Delta E <$ 0.1 μEh, $\Delta\mathbf{P} \leq 10^{-6}$, ΔForces <1 μEh $a_0{}^{-1}$), which are multiple orders of magnitude smaller than "chemical accuracy"(~1000 μEh).

Meanwhile, the methods lead to considerable speedups compared to pure fp64-execution, as presented in Table XI: 1.4–1.8× speedups are obtained from the mixed-precision method on CPUs, around 2× speedups are obtained from the mixed-precision method on GPUs, and up to 6.9× speedups are obtained with the incremental i-sn-LinK method on GPUs. The speedups of i-sn-LinK are typically larger for more expensive computations (larger molecules and larger basis sets) because the BLAS-3 steps [Eqs. (5) and (7)] are comparatively more expensive in these situations.

Comparing the CPU and GPU performance, we note that our incremental i-sn-LinK method is essential for an effective GPU acceleration with gaming GPUs. That is, pure fp64 execution is about as fast on the GPU as on CPUs,[59] but i-sn-LinK is up to 4.4× faster on the GPU [333 vs 1475 s for $(S_8)_{20}$] than on CPUs.

TABLE X. Comparison of the root mean square deviation of the converged density matrix ($\Delta\mathbf{P}$) from i-sn-LinK with the original thresholds $\vartheta_K = 10^{-8}/\vartheta_E = 10^{-11}$ (same values as in Table IX) and with very tight thresholds $\vartheta_K = 10^{-10}/\vartheta_E = 10^{-13}$.

Molecule	Basis	$\Delta\mathbf{P}$ (default thresh) (10^{-6})	$\Delta\mathbf{P}$ (tight thresh) (10^{-6})
Fullerene C_{60}	TZVP	10.36	0.63
Fullerene C_{60}	QZVP	8.00	1.00
Crambin	TZVP	0.31	0.12
$(DNA)_1$	TZVP	0.25	0.16
$(DNA)_4$	TZVP	0.39	0.26
$(DNA)_1$	QZVP	0.32	0.17
$(DNA)_4$	QZVP	0.60	0.21

TABLE XI. Performance-comparison of pure fp64 execution (fp64), mixed-precision 3c1e integral evaluation (mp), and i-sn-LinK on GTX 1080Ti GPU and two Xeon E5-2630 CPUs (CPU). Timings are given in seconds as the cumulative time for all exchange-builds, including the final exchange-energy calculation. The speedups compared to the respective pure fp64 implementation are given in parenthesis.

			K-time (s) GPU			K-time (s) CPU	
Molecule	Basis	N_{bfc}	fp64	mp	i-sn-LinK	fp64	mp
Taxol	6-31G*	1 013	194	92 (2.1×)	40 (4.8×)	187	107 (1.7×)
Valinomycin	6-31G*	1 350	385	180 (2.1×)	77 (5.0×)	353	203 (1.7×)
Olestra	6-31G*	3 181	698	421 (1.7×)	147 (4.7×)	567	376 (1.5×)
Fullerene C_{60}	TZVP	2 160	718	343 (2.1×)	167 (4.3×)	771	436 (1.8×)
Fullerene C_{60}	QZVP	4 320	2 690	1 220 (2.2×)	622 (4.3×)	2 780	1 590 (1.7×)
$(S_8)_{20}$	TZVP	6 720	2 290	1 220 (1.9×)	333 (6.9×)	2 470	1 480 (1.7×)
Crambin[a]	TZVP	13 698	22 400	11 200 (2.0×)	5260 (4.3×)	28 800	16 800 (1.7×)
$(DNA)_1$	SVP	660	48	27 (1.8×)	12 (4.0×)	52	32 (1.6×)
$(DNA)_4$	SVP	2 904	1 240	627 (2.0×)	256 (4.8×)	1 120	678 (1.6×)
$(DNA)_{16}$	SVP	11 880	9 600	6 200 (1.5×)	1890 (5.1×)	8 800	6 390 (1.4×)
$(DNA)_1$	TZVP	1 422	205	105 (2.0×)	45 (4.6×)	176	102 (1.7×)
$(DNA)_4$	TZVP	6 336	5 070	2 480 (2.0×)	1000 (5.0×)	5 200	3 060 (1.7×)
$(DNA)_1$	QZVP	3 465	975	450 (2.2×)	212 (4.6×)	920	550 (1.7×)
$(DNA)_4$[a]	QZVP	15 030	19 700	9 490 (2.1×)	3700 (5.3×)	19 952	12 349 (1.6×)
$Y_2C_5H_{20}N_{20}O_{13}$	TZVP	1 580	237	120 (2.0×)	53 (4.5×)	248	143 (1.7×)
$Y_4C_{10}H_{42}N_{40}O_{27}$	TZVP	3 208	1 060	514 (1.7×)	202 (5.2×)	1 060	614 (1.7×)

[a] Reduced grid-batch size due to limited GPU memory.

VI. CONCLUSION AND OUTLOOK

This work presents a systematic study of the applicability of single-precision instructions to evaluate the Fock-exchange matrix within seminumerical integration schemes. First, we demonstrated that only a very small fraction of the 3c1e integrals needs to be evaluated with fp64 instructions to still provide virtually the same result while providing nearly 2× speedups for this step on CPUs. We then demonstrated that pure fp32 execution still leads to surprisingly accurate results, as long as the final energy is computed at higher accuracy, although the SCF convergence behavior is somewhat deteriorated. Finally, we proposed the i-sn-LinK method, where incremental exchange-builds are employed for the last SCF steps, which removes all instabilities and inaccuracies from pure fp32 execution while requiring only a single high precision K-matrix build. This method provides up to 7× speedups for the whole SCF on typical "gaming" GPUs (1080Ti) without any significant impact on the numerical stability or accuracy and is therefore essential for an effective GPU acceleration using more affordable "gaming" GPUs.

Finally, we note that the present study on single-precision execution within seminumerical integration is also relevant for the evaluation of local-hybrid functionals, where the most time consuming step, i.e., the evaluation of the local exchange contributions, is identical to the seminumerical expression for the exchange matrix. Thus, we expect the performance benefits presented in this work to also translate to this novel and exciting class of functionals.

SUPPLEMENTARY MATERIAL

See the supplementary material for the effect of vector-instructions on the timings in Table II, absolute timings associated with the speedups of Tables II–IV and VI, analogous results to Tables IX and XI for the PBE0-functional, and all xyz-structures employed in this work.

ACKNOWLEDGMENTS

The authors acknowledge financial support by the "Deutsche Forschungsgemeinschaft" (DFG) via the Grant No. SFB 1309-32587107 and the Cluster of Excellence "Munich Center for Quantum Science and Technology" (MCQST, Grant No. EXC2111-390814868). C.O. further acknowledges financial support as a Max-Planck-Fellow at the MPI-FKF, Stuttgart.

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material.

REFERENCES

[1] J. C. Burant, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **105**, 8969 (1996).
[2] E. Schwegler and M. Challacombe, J. Chem. Phys. **105**, 2726 (1996).
[3] M. Challacombe and E. Schwegler, J. Chem. Phys. **106**, 5526 (1997).
[4] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).
[5] C. Ochsenfeld, C. A. White, and M. Head-Gordon, J. Chem. Phys. **109**, 1663 (1998).
[6] C. Ochsenfeld, Chem. Phys. Lett. **327**, 216 (2000).
[7] J. Kussmann and C. Ochsenfeld, J. Chem. Phys. **138**, 134114-1 (2013).
[8] J. Kussmann and C. Ochsenfeld, J. Chem. Theory Comput. **11**, 918 (2015).
[9] J. Kussmann and C. Ochsenfeld, J. Chem. Theory Comput. **13**, 3153 (2017).
[10] T. H. Thompson and C. Ochsenfeld, J. Chem. Phys. **150**, 044101-1 (2019).

[11] R. A. Friesner, Chem. Phys. Lett. **116**, 39 (1985).
[12] R. A. Friesner, J. Chem. Phys. **85**, 1462 (1986).
[13] R. A. Friesner, J. Chem. Phys. **86**, 3522 (1987).
[14] R. A. Friesner, J. Phys. Chem. **92**, 3091 (1988).
[15] M. N. Ringnalda, M. Belhadj, and R. A. Friesner, J. Chem. Phys. **93**, 3397 (1990).
[16] B. H. Greeley, T. V. Russo, D. T. Mainz, R. A. Friesner, J. M. Langlois, W. A. Goddard III, R. E. Donnelly, Jr., and M. N. Ringnalda, J. Chem. Phys. **101**, 4028 (1994).
[17] F. Neese, F. Wennmohs, A. Hansen, and U. Becker, Chem. Phys. **356**, 98 (2009).
[18] P. Plessow and F. Weigend, J. Comput. Chem. **33**, 810 (2012).
[19] F. Liu, T. Furlani, and J. Kong, J. Phys. Chem. A **120**, 10264 (2016).
[20] F. Liu and J. Kong, J. Chem. Theory Comput. **13**, 2571 (2017).
[21] F. Liu and J. Kong, Chem. Phys. Lett. **703**, 106 (2018).
[22] H. Bahmann and M. Kaupp, J. Chem. Theory Comput. **11**, 1540 (2015).
[23] T. M. Maier, H. Bahmann, and M. Kaupp, J. Chem. Theory Comput. **11**, 4226 (2015).
[24] S. Klawohn, H. Bahmann, and M. Kaupp, J. Chem. Theory Comput. **12**, 4254 (2016).
[25] R. Grotjahn, F. Furche, and M. Kaupp, J. Chem. Theory Comput. **15**, 5508 (2019).
[26] T. M. Maier, Y. Ikabata, and H. Nakai, J. Chem. Theory Comput. **15**, 4745 (2019).
[27] C. Holzer, J. Chem. Phys. **153**, 184115 (2020).
[28] H. Laqua, J. Kussmann, and C. Ochsenfeld, J. Chem. Theory Comput. **14**, 3451 (2018).
[29] H. Laqua, T. H. Thompson, J. Kussmann, and C. Ochsenfeld, J. Chem. Theory Comput. **16**, 1456 (2020).
[30] ANSI/IEEE Std 754-1985, 1, 1985.
[31] N. Luehr, I. S. Ufimtsev, and T. J. Martínez, J. Chem. Theory Comput. **7**, 949 (2011).
[32] G. Knizia, W. Li, S. Simon, and H.-J. Werner, J. Chem. Theory Comput. **7**, 2387 (2011).
[33] A. Asadchev and M. S. Gordon, Comput. Phys. Commun. **183**, 1563 (2012).
[34] Á. Rák and G. Cserey, Chem. Phys. Lett. **622**, 92 (2015).
[35] R. M. Parrish, F. Liu, and T. J. Martínez, J. Chem. Phys. **144**, 131101-1 (2016).
[36] J. J. Eriksen, Mol. Phys. **115**, 2086 (2017).
[37] L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, and A. Aspuru-Guzik, J. Phys. Chem. A **112**, 2049 (2008).
[38] V. P. Vysotskiy and L. S. Cederbaum, J. Chem. Theory Comput. **7**, 320 (2011).
[39] P. Pokhilko, E. Epifanovsky, and A. I. Krylov, J. Chem. Theory Comput. **14**, 4088 (2018).
[40] F. Weigend and R. Ahlrichs, Phys. Chem. Chem. Phys. **7**, 3297 (2005).
[41] H. Laqua, J. Kussmann, and C. Ochsenfeld, J. Chem. Phys. **149**, 204111-1 (2018).
[42] E. J. Baerends, D. E. Ellis, and P. Ros, Chem. Phys. **2**, 41 (1973).
[43] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, J. Chem. Phys. **71**, 3396 (1979).
[44] B. I. Dunlap, Phys. Chem. Chem. Phys. **2**, 2113 (2000).
[45] F. Weigend, Phys. Chem. Chem. Phys. **8**, 1057 (2006).
[46] C. Adamo and V. Barone, J. Chem. Phys. **110**, 6158 (1999).
[47] See https://software.intel.com/c-compilers for Intel C++ Compiler version 19.1.0.166.
[48] See https://developer.nvidia.com/cuda-10.1-download-archive-base for CUDA Toolkit 10.1.
[49] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).
[50] P. Pulay, J. Comput. Chem. **3**, 556 (1982).
[51] S. A. Maurer, D. S. Lambrecht, D. Flaig, and C. Ochsenfeld, J. Chem. Phys. **136**, 144107-1 (2012).
[52] D. Mueller, C. Knoll, A. Herrmann, G. Savasci, J. M. Welch, W. Artner, J. Ofner, B. Lendl, G. Giester, P. Weinberger, and G. Steinhauser, Eur. J. Inorg. Chem. **2018**, 1969.
[53] R. Olivares-Amaya, M. A. Watson, R. G. Edgar, L. Vogt, Y. Shao, and A. Aspuru-Guzik, J. Chem. Theory Comput. **6**, 135 (2010).
[54] We also tested evaluating the exchange energy as $tr(\mathbf{PK})$ instead, which results in up to 10 times larger errors.
[55] J. Almloef, K. Faegri, Jr., and K. Korsell, J. Comput. Chem. **3**, 385 (1982).
[56] M. Haeser and R. Ahlrichs, J. Comput. Chem. **10**, 104 (1989).
[57] We also tested conventional incremental updates, which lead to the need of slightly less integrals in the last SCF steps (<10% performance gains for the whole calculation) at the cost of slightly less stable convergence.
[58] P. Morgante and R. Peverati, Phys. Chem. Chem. Phys. **21**, 19092 (2019).
[59] Note that we compare a 700$ GPU with 1400$ (2 × 700$) CPUs.

Supporting Information to the Paper: Accelerating Seminumerical Fock-Exchange Calculations Using Mixed Single- and Double-Precision Arithmethic

Henryk Laqua,[1,2] Jörg Kussmann,[1,2] and Christian Ochsenfeld[1,2,a]

[1] *Department of Chemistry, Chair of Theoretical Chemistry,*
University of Munich (LMU), D-81377 München, Germany

[2] *Max Planck Institute for Solid State Research (MPI-FKF), 70569 Stuttgart,*
Germany.

(Dated: 26 April 2021)

[a] Electronic mail: christian.ochsenfeld@uni-muenchen.de

I. IMPACT OF VECTOR INSTRUCTIONS ON FP32 ACCELERATION

To show the importance of vector-instruction for improved floating point throughput, the code was recompiled with an additional "-no-vec" specifier (all other options were identical) to suppress vectorization by the compiler and present analogous timings to table 2 of the main document in table S1. As expected, execution without the use of SIMD instructions is significantly slower, i.e., up to 2.2x times slower with pure fp64 execution and up to 3.8x times slower with mixed-precision execution. Furthermore, the speedups from employing mixed precision in an otherwise completely identical setting are only 1.1 - 1.2x.

TABLE S1. The speedups within the evaluation of the 3c1e integrals from employing SIMD instructions in the pure fp64 code (vec/novec (fp64)) and in the mixed-precision code (vec/novec (mp)) and speedup from mixed precision evaluation compared to fp64 evaluation employing no vector-instructions (mp/fp64 (novec)) are given for 2 Intel Xeon E5-2630 CPUs (20 cores@2.20 GHz).

System/Basis	vec/novec (fp64)	vec/novec (mp)	mp/fp64 (novec)
$(DNA)_1$/SVP	2.06x	3.22x	1.11x
$(DNA)_4$/SVP	1.84x	2.95x	1.16x
$(H_2O)_{68}$/SVP	1.68x	2.73x	1.18x
$(LiF)_{36}$/SVP	1.87x	3.14x	1.13x
$Y_2C_5H_{20}N_{20}O_{13}$/SVP	2.22x	3.60x	1.10x
$(DNA)_1$/TZVP	2.22x	3.68x	1.13x
$(DNA)_4$/TZVP	1.85x	3.00x	1.15x
$(H_2O)_{68}$/TZVP	1.75x	2.81x	1.17x
$(LiF)_{36}$/TZVP	1.97x	3.32x	1.17x
$Y_2C_5H_{20}N_{20}O_{13}$/TZVP	2.26x	3.80x	1.14x
$(DNA)_1$/QZVP	2.19x	3.49x	1.13x
$(H_2O)_{68}$/QZVP	1.78x	2.84x	1.12x

II. ABSOLUTE EXECUTION TIMES CORRESPONDING TO SECTION IV

In table S2 we provide the CPU-timings corresponding to the speedups in table II of the main document and table S1. Moreover, we provide the GPU timings corresponding to table III and table IV of the main document in table S3. Finally we provide the cumulative GPU timings for the i-sn-LinK method corresponding to table 6 of the main document in table S4.

TABLE S2. Time [s] for one exchange-build (K) and for the 3c1e integral evaluation (3c1e) averaged over all but the very first SCF cycle for pure fp64 execution (fp64) and mixed-precision execution (mp) employing vector instructions (first 4 columns) or no vector instructions (last 4 columns), respectively. Calculations were performed on 2 Intel Xeon E5-2630 CPUs (20 cores@2.20 GHz).

System/Basis	fp64 [s]		mp [s]		fp64 novec [s]		mp novec [s]	
	3c1e	K	3c1e	K	3c1e	K	3c1e	K
$(DNA)_1$/SVP	3.06	3.51	1.77	2.26	6.29	6.76	5.68	6.19
$(DNA)_4$/SVP	69.62	81.94	37.44	50.45	128.17	140.05	110.33	122.59
$(H_2O)_{68}$/SVP	17.27	21.05	8.97	13.07	28.95	32.66	24.52	28.44
$(LiF)_{36}$/SVP	8.99	9.98	4.72	5.78	16.78	17.75	14.81	15.83
$Y_2C_5H_{20}N_{20}O_{13}$/SVP	4.92	5.54	2.76	3.43	10.91	11.52	9.94	10.62
$(DNA)_1$/TZVP	13.07	14.79	6.97	8.75	29.00	30.78	25.69	27.51
$(H_2O)_{68}$/TZVP	52.08	63.30	27.74	39.98	91.08	101.92	77.89	89.26
$(DNA)_4$/TZVP	363.41	413.57	194.77	246.73	672.18	719.98	584.46	633.13
$(LiF)_{36}$/TZVP	17.47	20.17	8.85	11.58	34.48	37.30	29.40	32.27
$Y_2C_5H_{20}N_{20}O_{13}$/TZVP	16.28	18.36	8.53	10.69	36.78	38.87	32.39	34.52
$(DNA)_1$/QZVP	63.75	71.08	35.43	43.00	139.54	147.04	123.72	131.30
$(H_2O)_{68}$/QZVP	360.32	426.71	202.95	278.37	643.00	704.93	576.58	639.76

TABLE S3. Time [s] for one exchange-build averaged over all but the very first SCF cycle for pure fp64 execution (fp64), mixed-precision 3c1e integral evaluation execution (mp), and pure fp32-execution (fp32; only 1080Ti) on the GV100 and the GTX1080Ti.

System/Basis	1080Ti [s]			GV100 [s]	
	fp64	mp	fp32	fp64	mp
$(DNA)_1$/SVP	3.2	2.0	0.6	1.0	1.0
$(DNA)_4$/SVP	86.1	44.1	13.5	11.7	9.6
$(H_2O)_{68}$/SVP	24.6	13.4	4.2	3.9	4.0
$(LiF)_{36}$/SVP	10.2	4.1	1.9	1.7	1.6
$Y_2C_5H_{20}N_{20}O_{13}$/SVP	5.6	2.9	0.9	1.2	1.2
$(DNA)_1$/TZVP	15.0	6.9	2.1	2.4	2.0
$(H_2O)_{68}$/TZVP	380.5	185.2	51.3	51.7	36.5
$(DNA)_4$/TZVP	68.7	40.4	10.6	10.1	8.2
$(LiF)_{36}$/TZVP	20.4	10.0	3.6	3.2	2.7
$Y_2C_5H_{20}N_{20}O_{13}$/TZVP	17.6	8.8	2.5	2.9	2.2
$(DNA)_1$/QZVP	71.0	33.1	11.3	15.5	8.7
$(H_2O)_{68}$/QZVP	457.4	265.0	62.5	74.1	51.3

III. RESULTS FOR THE PBE0 FUNCTIONAL

Analogous results to table IX of Section IV-B assessing the errors from reduced precision execution, but employing the PBE0 hybrid density functional instead of Hartree-Fock, are given in table S5.

Since only 25 % of exact-exchange is employed in this hybrid-functional, the errors are expected to be proportionally lower than the corresponding Hartree-Fock results. This fact is indeed supported by the results of table S5, where the errors for $\Delta\mathbf{P}$ are always smaller than the respective Hartree-Fock errors of the main manuscript. Therefore, the presented mixed-precision methods can just as well be employed for hybrid-DFT as for Hartree-Fock calculations without any additional considerations regarding the accuracy.

Moreover, the speedups for exact-exchange evaluation within the PBE0-functional given

TABLE S4. Cumulative time [s] for all exchange-builds within the SCF and the final exchange-energy build for pure fp64 execution (fp64) and i-sn-LinK on the GTX1080Ti. In addition, the individual timings within i-sn-LinK are given: time for *one* mixed-precision K-build (mp-K) and for the final energy calculation (final K).

System/Basis	fp64 [s]	i-sn-LinK [s]	mp-K [s]	final K [s]
$(DNA)_1$/SVP	63.9	13.3	1.8	3.8
$(DNA)_4$/SVP	1358.4	270.1	43.7	64.7
$(H_2O)_{68}$/SVP	313.5	68.3	13.3	18.1
$(LiF)_{36}$/SVP	127.4	29.8	4.0	9.4
$Y_2C_5H_{20}N_{20}O_{13}$/SVP	101.4	20.6	2.7	6.5
$(DNA)_1$/TZVP	257.4	51.8	7.6	14.4
$(H_2O)_{68}$/TZVP	5987.1	1153.4	184.6	263.0
$(DNA)_4$/TZVP	889.8	190.5	40.0	49.8
$(LiF)_{36}$/TZVP	259.6	59.0	9.4	15.7
$Y_2C_5H_{20}N_{20}O_{13}$/TZVP	296.4	59.4	8.9	17.7
$(DNA)_1$/QZVP	1082.3	244.7	32.0	56.5
$(H_2O)_{68}$/QZVP	5348.6	1158.0	263.3	281.9

in table S6 are very similar to the Hartree-Fock results presented in the main manuscript.

TABLE S5. Analogous results to table IX of main manuscript for the PBEH-functional.

Molecule	Basis	$N_{\mathrm{iter}}(10^{-7})$			ΔE [µE_{h}]		$\Delta\mathbf{P}$ [10^{-6}]		ΔForces [µ$E_{\mathrm{h}}\,a_0{}^{-1}$]	
		fp64	mp	i-sn-LinK	mp	i-sn-LinK	mp	i-sn-LinK	mp	i-sn-LinK
Taxol	6-31G*	12	12	12	<0.01	<0.01	<0.01	<0.01	<0.01	0.02
Valinomycin	6-31G*	12	12	12	<0.01	<0.01	<0.01	<0.01	<0.01	0.03
Olestra	6-31G*	12	12	12	<0.01	<0.01	<0.01	<0.01	<0.01	0.04
Fullerene C_{60}	TZVP	11	11	12	<0.01	<0.01	0.04	1.34	0.01	0.10
Fullerene C_{60}	QZVP	10	10	12	-0.01	<0.01	0.09	0.65	0.02	0.04
$(S_8)_{20}$	TZVP	11	11	11	<0.01	<0.01	<0.01	<0.01	<0.01	0.17
Crambin	TZVP	16	16	16	<0.01	0.02	0.01	0.27	<0.01	0.10
$(\mathrm{DNA})_1$	SVP	11	11	11	<0.01	<0.01	<0.01	<0.01	<0.01	0.03
$(\mathrm{DNA})_4$	SVP	12	12	12	<0.01	<0.01	<0.01	0.01	<0.01	0.07
$(\mathrm{DNA})_{16}$	SVP	14	14	14	<0.01	<0.01	<0.01	0.02	<0.01	0.28
$(\mathrm{DNA})_1$	TZVP	11	11	12	<0.01	<0.01	<0.01	0.06	<0.01	0.04
$(\mathrm{DNA})_4$	TZVP	11	11	12	<0.01	<0.01	<0.01	0.05	<0.01	0.11
$(\mathrm{DNA})_1$	QZVP	11	11	11	<0.01	<0.01	0.01	0.06	<0.01	0.07
$(\mathrm{DNA})_4$	QZVP	11	11	11	0.02	0.02	0.01	0.08	0.01	0.24
$Y_2C_5H_{20}N_{20}O_{13}$	TZVP	12	12	12	<0.01	<0.01	<0.01	0.01	<0.01	0.02
$Y_4C_{10}H_{42}N_{40}O_{27}$	TZVP	11	11	12	<0.01	<0.01	<0.01	0.06	<0.01	0.07

TABLE S6. Analogous results to table XI of main manuscript for the PBEH-functional.

Molecule	Basis	N_{bfc}	K-Time [s] GPU			K-Time [s] CPU	
			fp64	mp	i-sn-LinK	fp64	mp
Taxol	6-31G*	1013	198	93 (2.1x)	42 (4.8x)	193	112 (1.7x)
Valinomycin	6-31G*	1350	388	183 (2.1x)	78 (4.9x)	371	211 (1.8x)
Olestra	6-31G*	3181	759	451 (1.7x)	157 (4.8x)	653	431 (1.5x)
Fullerene C_{60}	TZVP	2160	765	366 (2.1x)	169 (4.5x)	831	474 (1.8x)
Fullerene C_{60}	QZVP	4320	2659	1238 (2.1x)	686 (3.9x)	2808	1634 (1.7x)
$(S_8)_{20}$	TZVP	6720	2966	1523 (1.9x)	529 (5.6x)	2792	1657 (1.7x)
Crambin[a]	TZVP	13698	33085	16326 (2.0x)	6940 (4.8x)	37066	24293 (1.5x)
$(\mathrm{DNA})_1$	SVP	660	49	27 (1.8x)	13 (3.9x)	49	31 (1.6x)
$(\mathrm{DNA})_4$	SVP	2904	1185	603 (2.0x)	254 (4.7x)	1187	727 (1.6x)
$(\mathrm{DNA})_{16}$	SVP	11880	12822	8262 (1.6x)	2690 (4.8x)	11293	7938 (1.4x)
$(\mathrm{DNA})_1$	TZVP	1422	192	97 (2.0x)	30 (6.4x)	199	117 (1.7x)
$(\mathrm{DNA})_4$	TZVP	6336	4841	2412 (2.0x)	1030 (4.7x)	5373	3147 (1.7x)
$(\mathrm{DNA})_1$	QZVP	3465	957	459 (2.1x)	213 (4.5x)	941	560 (1.7x)
$(\mathrm{DNA})_4$[a]	QZVP	15030	21305	9673 (2.2x)	5277 (4.0x)	22589	13931 (1.6x)
$Y_2C_5H_{20}N_{20}O_{13}$	TZVP	1580	278	138 (2.0x)	57 (4.9x)	272	156 (1.7x)
$Y_4C_{10}H_{42}N_{40}O_{27}$	TZVP	3208	1014	491 (2.1x)	207 (4.9x)	1014	587 (1.7x)

[a] Reduced grid-batch size due to limited GPU memory.

3.4 Publication IV: Accelerating Hybrid Density Functional Theory Molecular Dynamic Simulations by Seminumerical Integration, Resolution-of-the-Identity Approximation, and Graphics Processing Units

H. Laqua, J. C. B. Dietschreit, J. Kussmann, C. Ochsenfeld

J. Chem. Theory Comput. **18**, 6010 (2022).

Abstract

The computationally very demanding evaluation of the 4-center-2-electron (4c2e) integrals and their respective integral derivatives typically represents the major bottleneck within hybrid Kohn-Sham density functional theory molecular dynamics simulations. Building upon our previous works on seminumerical exact-exchange (sn-LinK) [Kussmann J., Laqua H., Ochsenfeld C., *J. Chem. Theory Comput.* **2021**, *17,* 1512], and resolution-of-the-identity Coulomb (RI-J) [Kussmann J., Laqua H., Ochsenfeld C., *J. Chem. Theory Comput.* **2021**, *17,* 1512], the expensive 4c2e integral evaluation can be avoided entirely, resulting in a highly efficient electronic structure theory method, allowing for fast ab initio molecular dynamics (AIMD) simulations even with large basis sets. Moreover, we propose to combine the final self-consistent field (SCF) step with the subsequent nuclear forces evaluation, providing the forces at virtually no additional cost after a converged SCF calculation, reducing the total runtime of an AIMD simulation by about another 25 %. In addition, multiple independent MD trajectories can be computed concurrently on a single node, leading to a greatly increased utilization of the available hardware – especially when combined with graphics processing unit acceleration – improving the overall throughput by up to another 5 times in this way. With all of those optimizations combined, our proposed method provides nearly 3 orders of magnitude faster execution times than traditional 4c2e integral-based methods. To demonstrate the practical utility of the approach, quantum-mechanical/molecular-mechanical dynamics simulations on double-stranded DNA were performed, investigating the relative hydrogen bond strength between adenine-thymine and guanine-cytosine base pairs. In addition, this illustrative application also contains a general accuracy assessment of the introduced approximations (integration grids, resolution-of-the-identity) within AIMD simulations, serving as a protocol on how to apply these new methods to practical problems.

Reprinted with permission from:

H. Laqua, J. C. B. Dietschreit, J. Kussmann, C. Ochsenfeld
"Accelerating Hybrid Density Functional Theory Molecular Dynamic Simulations by Seminumerical Integration, Resolution-of-the-Identity Approximation, and Graphics Processing Units"
J. Chem. Theory Comput. **18**, 6010 (2022).

https://pubs.acs.org/doi/10.1021/acs.jctc.2c00509

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

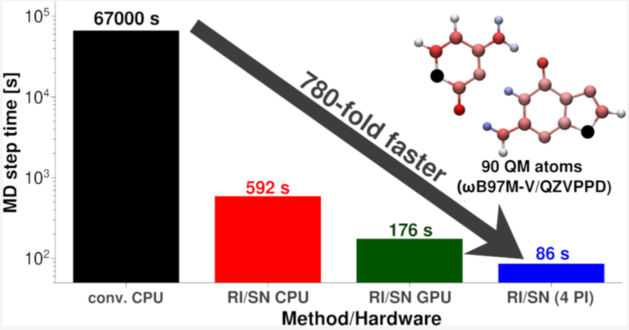
Accelerating Hybrid Density Functional Theory Molecular Dynamics Simulations by Seminumerical Integration, Resolution-of-the-Identity Approximation, and Graphics Processing Units

Henryk Laqua, Johannes C. B. Dietschreit, Jörg Kussmann, and Christian Ochsenfeld*

Cite This: *J. Chem. Theory Comput.* 2022, 18, 6010−6020

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: The computationally very demanding evaluation of the 4-center-2-electron (4c2e) integrals and their respective integral derivatives typically represents the major bottleneck within hybrid Kohn−Sham density functional theory molecular dynamics simulations. Building upon our previous works on seminumerical exact-exchange (sn-LinK) [Laqua, H., Thompsons, T. H., Kussmann, J., Ochsenfeld, C., *J. Chem. Theory Comput.* **2020**, *16,* 1465] and resolution-of-the-identity Coulomb (RI-J) [Kussmann, J., Laqua, H., Ochsenfeld, C., *J. Chem. Theory Comput.* **2021**, *17,* 1512], the expensive 4c2e integral evaluation can be avoided entirely, resulting in a highly efficient electronic structure theory method, allowing for fast ab initio molecular dynamics (AIMD) simulations even with large basis sets. Moreover, we propose to combine the final self-consistent field (SCF) step with the subsequent nuclear forces evaluation, providing the forces at virtually no additional cost after a converged SCF calculation, reducing the total runtime of an AIMD simulation by about another 25%. In addition, multiple independent MD trajectories can be computed concurrently on a single node, leading to a greatly increased utilization of the available hardware—especially when combined with graphics processing unit acceleration—improving the overall throughput by up to another 5 times in this way. With all of those optimizations combined, our proposed method provides nearly 3 orders of magnitude faster execution times than traditional 4c2e integral-based methods. To demonstrate the practical utility of the approach, quantum-mechanical/molecular-mechanical dynamics simulations on double-stranded DNA were performed, investigating the relative hydrogen bond strength between adenine−thymine and guanine−cytosine base pairs. In addition, this illustrative application also contains a general accuracy assessment of the introduced approximations (integration grids, resolution-of-the-identity) within AIMD simulations, serving as a protocol on how to apply these new methods to practical problems.

1. INTRODUCTION

Hybrid density functional theory (hybrid-DFT) has become the de facto standard for many quantum chemistry applications due to its excellent cost-performance ratio. However, the evaluation of the 4-center-2-electron (4c2e) integrals, which are necessary for the Coulomb and exact-exchange interactions, as well as the numerical integration to evaluate the semi-local exchange-correlation (XC) functional, represents significant computational bottlenecks.

For the evaluation of the Coulomb-interaction, the explicit computation of the 4c2e integrals can be avoided using the resolution-of-the-identity approximation (RI-J),[1] which can be further accelerated by a modified[2] J-engine algorithm[2−5] and the use of graphic processing units (GPUs). Similarly, for the exact-exchange interaction, the computation of the 4c2e integrals can be avoided using seminumerical integration,[6−14] for which we recently developed the linear-scaling, GPU-accelerated sn-LinK method.[15,16] Finally, the slowest steps of the numerical integration of the semi-local XC functional can be formulated as matrix−matrix multiplications, for which highly optimized linear algebra libraries can be employed. In combination with batch-wise screening and GPU acceleration, this integration can therefore also be performed very efficiently.[2]

In addition, within seminumerical integration the nuclear forces, that is, the derivative of the energy with respect to the nuclear coordinates, are obtainable without computing the corresponding integral derivatives.[8,9] Moreover, when combined with the final self-consistent-field (SCF) step, the exact-exchange gradient can be obtained at only marginal overhead after a converged SCF calculation, leading to another

Received: May 14, 2022
Published: September 22, 2022

ACS Publications

https://doi.org/10.1021/acs.jctc.2c00509

substantial efficiency gain compared to analytical integration. To our knowledge, such a combination of seminumerical exchange energy and forces has so far not been exploited and is the prime motivation for our present work.

Because similar combinations of energy and forces computations can also be applied to the resolution-of-the-identity-Coulomb (RI-J) approximation and the numerical integration of the semi-local XC functional (cf. Sections 2.2 and 2.3 of ref 2), nuclear gradients are obtainable at virtually no overhead from a converged hybrid-DFT calculation. These savings are especially relevant within ab initio molecular dynamics (AIMD) simulations, where, due to the use of the extended-Lagrangian (xl) extrapolation of the density matrix,[17−21] only a few SCF-cycles have to be performed in each MD step, so that the reduced computation time for the nuclear forces is even more impactful.

Therefore, we present the applicability of these low-overhead nuclear forces to AIMD simulations and discuss the practical impact of the introduced approximations (RI, numerical integration, finite time steps) on the quality of the MD trajectories and the associated performance gains. As an example, we investigate the different hydrogen-bond strengths of the two Watson−Crick pairs in double-stranded (DS) DNA. We especially focus on the corresponding markers in vibrational spectra, that is, the red or blue shift of the respective modes. In this way, we aim to provide a protocol regarding, for example, grids, thresholds etc., outlining how to apply our sn-LinK method to practical quantum-chemical simulations.

The paper is structured as follows: first, we briefly summarize the theory of the RI-J contributions (matrix, energy, and forces) and of the seminumerical exchange method sn-LinK and subsequently derive the corresponding integral-derivative-free exchange forces in Section 2. We report on the computational setup in Section 3 and then assess the accuracy of the introduced approximations (RI, finite integration grids, and finite time steps) by looking at the vibrational density of state (VDoS) spectra of DNA bases obtained from Fourier analysis[22] of AIMD trajectories in Section 4. Subsequently, we compare the performance to conventional 4c2e integral-based methods (LinK[23,24] for exchange, J-engine[3−5] for Coulomb contributions) for different basis sets in Section 5.

Finally, we utilize the significantly improved efficiency to perform AIMD simulations of DS-DNA in order to quantify the hydrogen-bond strength between the base pairs by analysis of the vibrational free energies[22,25,26] of the corresponding hydrogen atoms in Section 6. This application illustrates the practical importance of our method for low-cost evaluation of nuclear gradients because the required AIMD trajectories, which comprise over 2 million nuclear gradient calculations in total, would otherwise come at a prohibitively high computational cost.

2. THEORY

2.1. Resolution-of-the-identity Approximation for Coulomb Interactions (RI-J). The resolution-of-the-identity (RI) approximation employing Coulomb-fitting decomposes the 4-center-2-electron (4c2e) integral tensor as

$$(\mu\nu|\lambda\sigma) \equiv \int d\mathbf{r}_1 \int d\mathbf{r}_2 \chi_\mu(\mathbf{r}_1)\chi_\nu(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\chi_\lambda(\mathbf{r}_2)\chi_\sigma(\mathbf{r}_2) \approx \sum_{PQ} (\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma) \tag{1}$$

where μ, ν, λ, σ denote atomic orbital (AO) basis functions, P, Q, R, S denote auxiliary AO-type basis functions, and $(P|Q)^{-1}$ denotes the *matrix* inverse of the 2-center-2-electron (2c2e) integrals $(P|Q)$.

This decomposition is particularly useful for Coulomb-like interactions, for example, the Coulomb matrix can be obtained as

$$J_{\mu\nu} = \sum_{\lambda\sigma} (\mu\nu|\lambda\sigma)P_{\lambda\sigma} \approx \sum_{\lambda\sigma PQ} (\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma)P_{\lambda\sigma} \tag{2}$$

where $P_{\lambda\sigma}$ denotes elements of the density matrix. Equation 2 is evaluated in three consecutive steps

$$\text{step 1: } B_Q = \sum_{\lambda\sigma} (Q|\lambda\sigma)P_{\lambda\sigma} \tag{3}$$

$$\text{step 2: } B'_P = \sum_{Q} (P|Q)^{-1}B_Q \tag{4}$$

$$\text{step 3: } J_{\mu\nu} = \sum_{P} (\mu\nu|P)B'_P \tag{5}$$

where steps 1 and 3 require on-the-fly computation of the 3-center-2-electron (3c2e) integrals $(Q|\lambda\sigma)$ and thus represent the computational bottleneck. In order to overcome this bottleneck, a modified J-engine algorithm is employed to evaluate both steps very efficiently.[1−5]

Note that, if only the Coulomb energy

$$E_J = \frac{1}{2}\sum_{\mu\nu} P_{\mu\nu}J_{\mu\nu} \approx \frac{1}{2}\sum_{\mu\nu\lambda\sigma PQ} P_{\mu\nu}(\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma)P_{\lambda\sigma} = \frac{1}{2}\sum_{P} B'_P B_P \tag{6}$$

needs to be calculated (e.g., after the final SCF step), the evaluation of step 3 (eq 5) can be omitted.

By differentiation of eq 6 with respect to the nuclear coordinates (cf. Section 2.2 of ref 2), the expression for the nuclear forces within the RI-J approximation is obtained as

$$\begin{aligned} E_J^x &\approx \frac{1}{2}\sum_{\mu\nu\lambda\sigma PQ} P_{\mu\nu}P_{\lambda\sigma}[(\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma)]^x \\ &= \frac{1}{2}\sum_{\mu\nu\lambda\sigma PQ} P_{\mu\nu}P_{\lambda\sigma}[2([\mu\nu]^x|P)(P|Q)^{-1}(Q|\lambda\sigma) \\ &\qquad + 2(\mu\nu|P^x)(P|Q)^{-1}(Q|\lambda\sigma) \\ &\quad - \sum_{RS} (\mu\nu|P)(P|R)^{-1}[(R|S)]^x(S|Q)^{-1}(Q|\lambda\sigma)] \end{aligned} \tag{7}$$

where the superscript x denotes the derivative of the respective quantity with respect to one nuclear coordinate. All three terms can be evaluated from the intermediate quantity B'_P of step 2 (eq 4) as

$$E_{J,1}^x = \sum_{\mu\nu P} ([\mu\nu]^x|P)B'_P) \tag{8}$$

$$E^{x}_{J,2} = \sum_{\mu\nu P} (\mu\nu|P^{x})B'_{P} \tag{9}$$

$$E^{x}_{J,3} = -\frac{1}{2}\sum_{PQ} B'_{P}[(P|Q)]^{x}B'_{Q} \tag{10}$$

Consequently, combining the final (post-SCF) energy and forces builds avoids the repeated evaluation of steps 1 and 2, which are necessary in both cases. The slowest steps within the application of the RI-J approximation are the ones involving 3c2e integrals or their respective derivatives, eqs 3, 5, 8, and 9, all of which scale formally as $O(N_{\mathrm{AO}}^{2}N_{\mathrm{aux}} \sim M^{3})$ (N_{AO} = number of AO basis functions, N_{aux} = number of auxiliary basis functions, M = number of atoms) which reduces to asymptotically $O(M^{2})$ due to the fast exponential distance decay of the AO overlap distributions $[\mu\nu]$.

2.2. Seminumerical Integration. In contrast to the RI approximation, seminumerical integration decomposes the 4c2e integral tensor as

$$(\mu\sigma|\nu\lambda) \approx ([\mu\sigma]^{\mathrm{num}}|[\nu\lambda]^{\mathrm{ana}}) \equiv \sum_{g} w_{g}\chi_{\mu}(\mathbf{r}_{g})\chi_{\sigma}(\mathbf{r}_{g}) \int \mathrm{d}\mathbf{r}\frac{\chi_{\nu}(\mathbf{r})\chi_{\lambda}(\mathbf{r})}{|\mathbf{r}_{g} - \mathbf{r}|} \tag{11}$$

where $\mathbf{r}_{g}$ denotes grid-points with corresponding weights w_{g}. Depending on the specific use case, eq 11 may also be symmetrized as

$$(\mu\sigma|\nu\lambda) \approx \frac{1}{2}[([\mu\sigma]^{\mathrm{num}}|[\nu\lambda]^{\mathrm{ana}}) + ([\mu\sigma]^{\mathrm{ana}}|[\nu\lambda]^{\mathrm{num}})] \tag{12}$$

Inserting the symmetric decomposition into the AO representation of the exchange matrix leads to

$$K_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu\sigma|\nu\lambda) \approx \frac{1}{2}\sum_{\lambda\sigma} P_{\lambda\sigma}[([\mu\sigma]^{\mathrm{num}}|[\nu\lambda]^{\mathrm{ana}}) + ([\mu\sigma]^{\mathrm{ana}}|[\nu\lambda]^{\mathrm{num}})] \tag{13}$$

$$=\frac{1}{2}\left[\sum_{g} w_{g}\sum_{\lambda\sigma}\chi_{\mu}(\mathbf{r}_{g}) \int \mathrm{d}\mathbf{r}\frac{\chi_{\nu}(\mathbf{r})\chi_{\lambda}(\mathbf{r})}{|\mathbf{r}_{g} - \mathbf{r}|}P_{\lambda\sigma}\chi_{\sigma}(\mathbf{r}_{g}) + \text{transpose}\right] \tag{14}$$

where the transpose is due to the symmetrization in eq 12. Equation 13 is best evaluated in three steps

$$\text{step 1:}\quad F_{\lambda g} = \sum_{\sigma}\chi_{\sigma}(\mathbf{r}_{g})P_{\lambda\sigma} \tag{15}$$

$$\text{step 2:}\quad G_{\nu g} = \sum_{\lambda} w_{g}A_{\nu\lambda g}F_{\lambda g} \tag{16}$$

$$\text{step 3:}\quad K_{\mu\nu} = \sum_{g}\chi_{\mu}(\mathbf{r}_{g})G_{\nu g} \tag{17}$$

and finally symmetrized to account for the transpose. Here, the evaluation of step 2 (eq 16) typically represents the slowest step due to the 3-center-1-electron (3c1e) integrals

$$A_{\nu\lambda g} = \int \mathrm{d}\mathbf{r}\frac{\chi_{\nu}(\mathbf{r})\chi_{\lambda}(\mathbf{r})}{|\mathbf{r}_{g} - \mathbf{r}|}, \tag{18}$$

which are on-the-fly evaluated using machine-optimized [common sub-expression elimination (CSE) using SymPy[27]] Obara–Saika[28,29] recurrence relations.

2.3. Seminumerical Exact-Exchange Gradients. Inserting the asymmetric tensor decomposition (eq 11) into the AO expression for the exchange-gradients

$$E^{x}_{K} \equiv \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}P_{\lambda\sigma}(\mu\sigma|\nu\lambda)^{x} = 4\sum_{\mu\nu\lambda\sigma} P_{\mu\nu}P_{\lambda\sigma}(\mu^{x}\sigma|\nu\lambda) \tag{19}$$

leads to

$$E^{x}_{K} \approx 4\sum_{g} w_{g}\sum_{\mu\nu\lambda\sigma}\chi^{x}_{\mu}(\mathbf{r}_{g})P_{\mu\nu}\int \mathrm{d}\mathbf{r}\frac{\chi_{\nu}(\mathbf{r})\chi_{\lambda}(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_{g}|}P_{\lambda\sigma}\chi_{\sigma}(\mathbf{r}_{g}) \tag{20}$$

In this way, the computation of any integral-derivatives is avoided completely—a significant advantage compared to the analytical exchange gradients of eq 19. However, the so obtained gradients are, contrary to the RI-J gradients, *not* the exact derivative of the semi-numerical exchange energy when using incomplete grids. Instead, they utilize a slightly different approximation to the exact (analytical) gradient than the seminumerical approximation of Section 2.2. The practical impact of this finite-grid effect, which vanishes rapidly for sufficiently large integration grids, is further investigated in Sections 4 and in S2 of the Supporting Information.

Equation 20 can be evaluated with little overhead from the intermediate quantity $G_{\nu g}$ of eq 16, employing two additional steps

$$Z_{\mu g} = \sum_{\nu} P_{\mu\nu}G_{\nu g} \tag{21}$$

$$E^{x}_{K} = 4\sum_{\mu g}\chi^{x}_{\mu}(\mathbf{r}_{g})Z_{\mu g} \tag{22}$$

The perturbed basis-function values are readily available from the gradient of the basis functions

$$\chi^{x}_{\mu}(\mathbf{r}_{g}) = \begin{cases} -\dfrac{\partial}{\partial x}\chi_{\mu}(\mathbf{r}) & \chi_{\mu} \text{ centered at perturbed nucleus} \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

Because the intermediate quantity $G_{\nu g}$ is required for the exchange matrix, exchange energy, and the exchange forces, all three quantities are best evaluated in one combined step. That is, the exchange-forces are best evaluated together with the final Fock-build, which adds only a very small overhead (dominated by the evaluation of eq 21) because the evaluation of the 3c1e integrals necessary for the evaluation of $G_{\nu g}$ (eq 16) is usually the computational bottleneck due to its formal $O(N_{\mathrm{AO}}^{2}N_{\mathrm{g}} \sim M^{3})$ (N_{g} = number of grid points) time complexity. This scaling can be reduced to asymptotically $O(M)$, exploiting both the locality of the AO basis functions and the sparsity of the density matrix employing tight batch-wise integral screening, representing the fundamental principle of our sn-LinK method.[15,16]

For details regarding the implementation of sn-LinK, we refer the reader to ref 15 and within this work focus instead more on the practical implications regarding the application of seminumerical exchange methods within AIMD simulations, for example, which grid to choose, what errors and artifacts to expect, and what level of performance can be achieved.

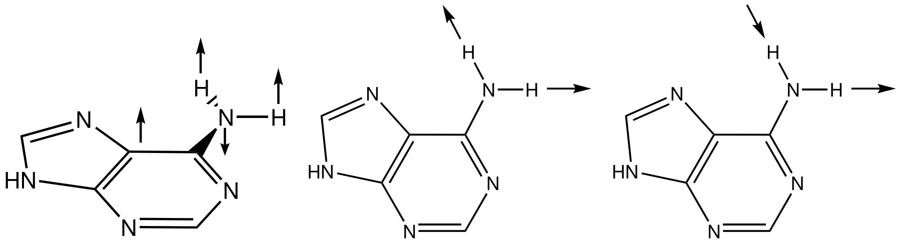


Figure 1. Vibrational modes of the NH_2-group investigated in this section. Left: out-of-plane wagging. Middle: symmetric bond-stretching. Right: asymmetric bond stretching.

3. COMPUTATIONAL DETAILS

All calculations were performed with our Fermions++ program.[30,31] The exchange gradients were implemented on top of our existing GPU-accelerated seminumerical exact-exchange method sn-LinK,[15,16,32] employing mixed-precision floating point execution in combination with incremental Fock-builds (cf. ref 16). That is, all incremental exchange-builds were evaluated with pure single-precision (fp32), whereas the very first (nonincremental) exchange-matrix build as well as the final energy/forces build were evaluated with mixed single/double precision (fp32/fp64).

For both, the seminumerical integration as well as the numerical integration of the semi-local exchange−correlation functional, the integration grids defined in ref 33 have been employed. Throughout this work, we use the multi-grids "gm[2−5]", that is, the SCF was converged with smaller grids (e.g., "g1" for "gm3") than the final energy and forces computation, providing a significant efficiency improvement with virtually no deterioration of the overall accuracy (cf. discussion in ref 33). Furthermore, because only a fraction (e.g., 42% for PBEh-3c[34]) of exact-exchange is employed in hybrid-DFT and the grid-errors of the exact-exchange part are typically smaller than for the semi-local exchange−correlation part, we employ smaller grids for the exact-exchange contributions (e.g., "gm2" for "gm3" parent grid). For the RI-J approximation, the universal-J-fit basis of Weigend[35] has been used throughout this work.

In order to accelerate the SCF convergence in each MD simulation step, accurate guess densities were obtained from the previous nine density matrices according to the extended-Lagrangian extrapolation method[17−21] so that SCF convergence to within 10^{-6} for the DIIS-error RMS($\mathbf{FPS} - \mathbf{SPF}$) was achieved within 4−5 SCF cycles. Moreover, the dissipative force term in the extended-Lagrangian extrapolation of the density matrix prevents a cumulative build-up of numerical errors from incomplete SCF convergence and ensures time-reversibility.

The VDoS were obtained from the Fourier transformation of the velocity autocorrelation function $\langle \vec{v}_A(t+\tau)\vec{v}_A(\tau)\rangle_\tau$ for each nucleus A (cf. Section 2.2 of ref 22) as

$$\begin{aligned} S(\nu) &= \sum_A S_A(\nu) \\ &= \sum_A 4m_A \int_0^\infty dt \langle \vec{v}_A(t+\tau)\vec{v}_A(\tau)\rangle_\tau \cos(2\pi\nu t) \end{aligned} \tag{24}$$

All VDoS spectra presented in this work are averaged over 10 MD trajectories of 20 ps length each, where (unless stated otherwise) the temperature was kept at 300 K employing the Bussi−Donadio−Parinello thermostat.[36] The quantum-mechanical (QM) vibrational free energy was obtained from the VDoS spectra according to ref 37 as

$$A_{\text{vib}}^{\text{QM}} = \int d\nu S(\nu) W_A^{\text{QM}}(\nu) \tag{25}$$

$$W_A^{\text{QM}}(\nu) = \ln\left[\frac{1 - \exp(-2\pi\nu)}{\exp(-\pi\nu)}\right] \tag{26}$$

All further details regarding the setup of the AIMD simulations can be found in Section S1 of the Supporting Information.

4. ACCURACY OF THE NUMERICAL QUADRATURES

The numerical integration on a finite set of grid points leads to small numerical errors in the potential energy surface (PES), which quickly vanish with an increase in grid resolution. In practice, this manifests as tiny waves in the PES, that are on the order of a few μE_h, which is usually insignificant for chemical energy differences, but can become relevant for very low-frequency harmonic vibrational modes.

In contrast, high-frequency vibrations are instead mostly affected by the finite-time step errors in an MD simulation because a sufficient amount of sampling points per oscillation is required to properly resolve the vibration. To illustrate the effects of these two numerical artifacts, we investigate the influence of the numerical integration grid and the MD time step size on the very high-frequency N−H bond stretch modes and the very low-frequency CNH_2 out-of-plane wagging modes of adenine, as shown in Figure 1.

First, the effect of numerical integration and the RI approximation on single-point harmonic frequencies is investigated in Table 1, confirming that the low-frequency wagging mode is indeed substantially more sensitive to the RI and the numerical integration exhibiting errors in excess of 5 cm^{-1}, compared to the high-frequency modes with errors below 1 cm^{-1}. In any case, the results can always be converged to the analytical results with tighter grids and larger RI basis sets.

Table 1. Vibrational Frequencies [cm^{-1}] of Selected Harmonic Modes of Adenine for Different Numerical Quadratures (Grid, RI-J) Employing PBEh-3c

method	NH_2-wagging	sym. NH_2 stretch	asym. NH_2 stretch
analytical	110.1	3706.2	3836.7
gm5 no RI-J	110.2	3706.2	3836.6
gm3 no RI-J	116.9	3706.0	3836.2
gm5 with RI-J	103.2	3706.1	3835.9

After the investigation of single-point harmonic frequencies, we examine the respective peaks in the VDoS spectra (cf. eq

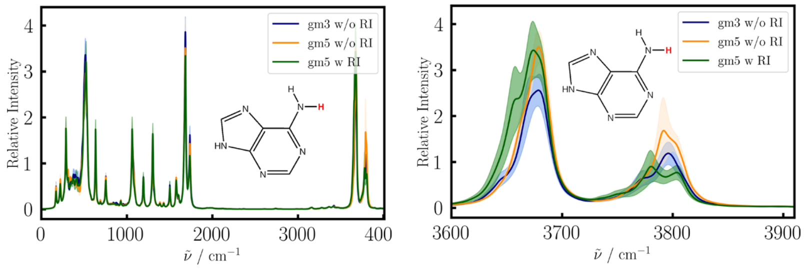


Figure 2. VDoS spectrum of the adenine amino-H in vacuo (colored red; H61 see Figure 5) employing different numerical quadratures. The solid line represents the mean over all 10 trajectories and the lightly shaded regions correspond to the standard error of the mean (SEM). Left: whole spectrum. Right: zoom-in at 3600−3900 cm^{-1}.

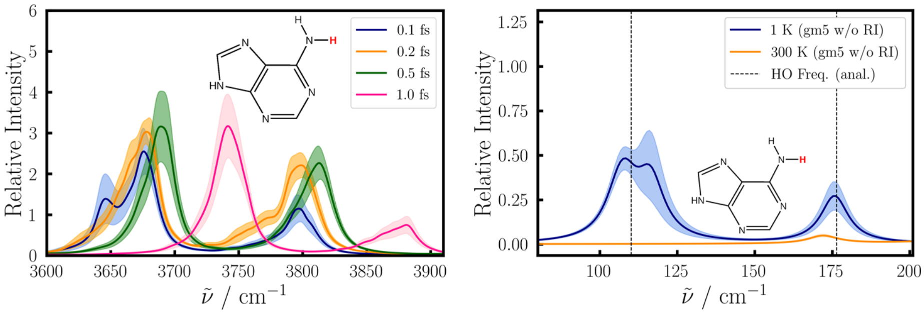


Figure 3. VDoS spectrum of the adenine amino-H in vacuo (colored red; H61 see Figure 5). Solid lines represent the mean over 10 trajectories and shaded regions the SEM. Left plot: zoom-in on N−H stretch mode at 300 K and gm5 w/o RI-J for simulations with different time steps. Right plot: zoom-in on NH_2-wagging mode with a time step of 0.2 fs.

24) of adenine in Figures 2 and 3. In contrast to the harmonic frequencies, these spectra are not substantially affected by either RI or numerical integration even with the smallest (gm3) grid, neither at low nor at high frequencies. The RI-J approximation has a minor impact on the shape of the high-frequency peaks given on the right side of Figure 2, whereas the effect of the integration grid is completely insignificant (smaller than the MD sampling error).

In contrast, the high-frequency modes are instead much more substantially affected by the size of the simulation time-step (left side of Figure 3), requiring about 0.2 fs per step to be sufficiently converged. Furthermore, the results for the low-frequency wagging mode (right side of Figure 3) are particularly surprising because this mode is completely absent from the VDoS spectrum at 300 K and only appears at extremely low temperatures (1 K; right side of Figure 3).

This temperature effect is best explained by a scan of the wagging mode, as presented in Figure 4: the mode is substantially anharmonic within the thermally accessible region of the energy surface. Thus, the harmonic approximation provides a qualitatively wrong result at 300 K and the actual vibration, which is only properly captured within the MD simulation, appears at much higher frequencies and overlaps with other molecular vibrations, explaining its apparent absence.

This result that the harmonic approximation is inapplicable for ultralow-frequency modes at finite temperatures is completely general. To illustrate, because the thermally accessible region within a low-frequency mode is always large, truncation of the Taylor series expansion around the minimum after the quadratic term generally incurs a large error. Consequently, proper sampling of the PES (e.g., via MD) in a large region around the minimum is necessary instead. Therefore, we argue that numerical errors of low-frequency harmonic vibrations (as presented in Table 1) are only a symptom of the underlying problem within the harmonic approximation, which disappears when employing proper MD-based sampling as shown on the right side of Figure 3.

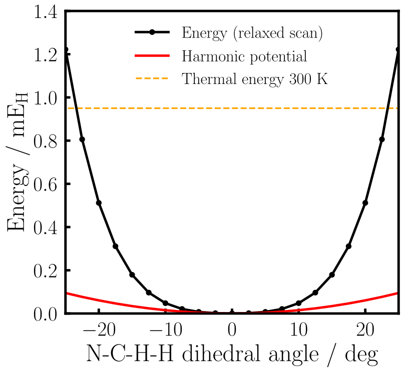


Figure 4. Relaxed scan (PBEh-3c) of the N−C−H−H dihedral angle (NH_2-wagging mode).

5. PERFORMANCE

For a practical illustration, we compare the performance for three increasingly demanding exact-exchange including electronic structure methods, that is, PBEh-3c[34] (reparametrized PBE0-functional[38−41] with modified def2-SVP[42] basis set), ωB97M-V/def2-TZVP, and ωB97M-V/def2-QZVPPD (the def2-prefix is omitted in the following for brevity).[43,44] We decided to test the performance on the main system of interest of this work, namely, an adenine−thymine DNA DS employing quantum-mechanical/molecular-mechanical (QM/MM) electrostatic embedding[45] with a total of 90 atoms (three DNA-

Table 2. Cumulative Time in Seconds for All Exchange (K), Coulomb (J), and Exchange−Correlation (XC) Contributions for One QM/MM AIMD Step of One AT DNA Double Strand in Aqueous Solution (90 QM, 9239 MM Atoms) Averaged over 10 MD Steps, Decomposed into the Respective Potential Matrix Builds within the SCF (Pot.) and the Final Energy and Gradient Builds (E + G)[a]

		K [s]		J [s]		XC [s]			
method	hardware	Pot.	E + G	Pot.	E + G	Pot.	E + G	other	total
PBEh-3c/mSVP									
conv. J/K (sep. grad.)	CPU	14.2	13.5	2.0	2.1	0.8	1.8	4.8	39.3
RI-J/sn-LinK (sep. grad.)	CPU	2.5	5.5	0.4	0.4	0.7	1.6	3.1	16.5
RI-J/sn-LinK (comb. grad.)	CPU	2.5	2.9	0.4	0.2	0.7	1.3	4.8	12.8
RI-J/sn-LinK (comb. grad.)	GPU	1.7	1.7	0.4	0.2	0.2	0.2	5.8	10.4
RI-J/sn-LinK 16 instances[b]	GPU								2.1
ωB97M-V/TZVP									
conv. J/K (sep. grad.)	CPU	300.8	410.7	37.6	38.8	6.3	9.0	16.7	820.0
RI-J/sn-LinK (sep. grad.)	CPU	18.9	41.0	1.3	1.0	6.2	6.9	17.1	92.4
RI-J/sn-LinK (comb. grad.)	CPU	19.4	22.1	1.3	0.8	6.0	4.3	15.6	69.5
RI-J/sn-LinK (comb. grad.)	GPU	5.2	3.9	1.2	0.8	1.6	0.9	15.9	29.5
RI-J/sn-LinK 8 instances[b]	GPU								10.7
ωB97M-V/QZVPPD									
conv. J/K (sep. grad.)	CPU	34261.0	29470.7	1624.2	1289.0	31.7	39.8	60.6	66777.0
RI-J/sn-LinK (sep. grad.)	CPU	257.3	373.4	7.6	6.1	35.2	30.6	83.2	793.4
RI-J/sn-LinK (comb. grad.)	CPU	256.6	190.7	7.7	4.9	34.9	17.2	80.0	591.9
RI-J/sn-LinK (comb. grad.)	GPU	43.7	22.9	9.0	4.7	7.2	2.2	86.2	176.0
RI-J/sn-LinK 4 instances[b]	GPU								90.2

[a] Timings are given for pure CPU execution (2 Intel Xeon Silver 4216 CPUs; 32 cores, 64 threads@2.1 GHz) and with GPU acceleration (4 Radeon VII GPUs) employing conventional (4c2e integral based) Coulomb/exchange builds (conv. J/K) or RI-J and sn-LinK. The latter is given for separate (sep. grad.) and combined energy and gradient builds (comb. grad). [b] Effective step time as measured by total MD throughput by running multiple program instances concurrently on one compute node (see also text for explanation).

base pairs) treated with QM and ~9000 atoms treated with MM. Although the overall computational efficiency is mostly determined by the QM part, we provide the details of the MM treatment and the QM/MM embedding in Section S1 of the Supporting Information.

All timings are given for one full MD step averaged over 10 steps, where each MD step consists of a fully converged SCF calculation and a subsequent energy and forces calculation, either separately or one combined computation as described in Section 2. The SCF calculation typically requires ~3 SCF steps to achieve sufficient convergence [DIIS error RMS(**FPS** − **SPF**) < 10^{-6}]. This fast SCF convergence is due to the extended-Lagrangian extrapolation method[17−21] which provides a very accurate initial guess by linear-combination of the previous nine density matrices.

As apparent from the timings in Table 2, RI-J and sn-LinK always outperform conventional, 4c2e integral-based methods for Coulomb and exchange builds, especially with regards to larger basis sets, for example, they yield speedups of 5.7× (2.5 s instead of 14.2 s) for the exchange matrix builds (**K**-Pot.) using PBEh-3c/mSVP or 130× (257.3 s instead of 34,261.0 s) for ωB97M-V/QZVPPD. The better speedups for larger basis sets are a direct consequence of the lower $O(N_{\text{bas}}^2)$ scaling with respect to the AO basis set of RI-J and sn-LinK compared to the formal $O(N_{\text{bas}}^4)$ scaling of conventional methods.

Moreover, the overall time spent on exact-exchange contributions is generally much larger than on Coulomb contributions so that the absolute saving from RI-J compared to the conventional J-engine[3] method are less significant. Nevertheless, the use of the RI-J approximation is still essential for large basis sets: to illustrate, the 1624.2 s required for the Coulomb-potential builds for ωB97M-V/QZVPPD—albeit small compared to 34,261.0 s for the conventional computation of the exchange matrix—is still very substantial when contrasted with the total step-time of RI-J/sn-LinK of only 793.4 s. Using the RI-J approximation removes this bottleneck entirely, that is, reducing the computation time over 200-fold from 1624.2 to 7.6 s.

In addition, the advantage of combining energy and force builds becomes apparent: the cost of exact-exchange energy and forces is reduced by about 50%, for example, 190.7 s instead of 373.4 s for ωB97M-V/QZVPPD, reflecting the fact that the combination of energy and forces is obtained at essentially the same cost as each separate energy and forces computation individually. The so-obtained time savings are quite significant, roughly equaling the time of all SCF steps *combined* (e.g., 22.1 s instead of 41.0 s equaling 18.9 s reduction for ωB97M-V/TZVP), ultimately leading to a 25% reduction of the total MD step time (e.g., 69.5 s instead of 92.4 s for ωB97M-V/TZVP).

Similarly, the semilocal XC energy and forces can be obtained up to 1.8× faster (17.2 s instead of 30.6 s for ωB97M-V/QZVPPD). The speedups from combining RI-J energy and forces, however, are considerably smaller, for example, only 1.3× speedups (4.9 s instead of 6.1 s) for ωB97M-V/QZVPPD, because the evaluation of the RI-J forces requires additional compute-expensive steps, particularly the steps involving the perturbed 3c−2e integrals (eqs 8 and 9).

This already quite impressive performance is even further improved with GPU acceleration, especially for the larger basis-sets TZVP and QZVPPD, which provide more parallel workload, for example, another 5.9× acceleration (43.7 s instead of 256.6 s) is achieved for the **K**-potential builds for ωB97M-V/QZVPPD. Similar levels of GPU acceleration are also observed for the semilocal XC contribution, for example, 4.8× (7.2 s instead of 34.9 s) and 7.8× (2.2 s instead of 17.2 s) for the ωB97M-V/QZVPPD XC potential and forces, respectively. Note that the RI-J part was exclusively computed with CPUs because the simulated system size of 90 QM atoms is not large enough to provide sufficient parallel workload for efficient GPU acceleration (cf. discussion in Section 4.1 of ref 2). However, given the comparatively low cost of the RI-J steps, this does not result in a significant performance loss in practice.

To summarize, compared to conventional 4c2e integral-based evaluation, the application of sn-LinK in combination with GPUs accelerates the computation of the exact-exchange potential by up to 780× (43.7 s instead of 34,261.0 s for ωB97M-V/QZVPPD) and the exact-exchange energy and forces by up to 1300× (22.9 s instead of 29,470.7 s for ωB97M-V/QZVPPD). Due to this immense acceleration of the typical computational bottlenecks, that is, the computation of the Coulomb-, the exact-exchange, and the semilocal exchange−correlation contributions, these steps only comprise about 50% of the total computation time. The other 50% of the computation time (denoted by the "other" column in Table 2) is split between dozens of other necessary steps, for example, the preparation of the shell-pair data, the generation of the integration grids, the diagonalization of the Kohn−Sham matrix, the evaluation of one-electron terms, QM/MM interactions, the MM-forces, and many more.

Because many of these other steps run exclusively on CPUs, we expect substantial speedups (increased total throughput) by running multiple independent program instances concurrently on a single node, as indicated by results presented in Table 3 and the last rows of Table 2. Addtional data to multi-instance performance is also provided in Section S3 of the Supporting Information.

Table 3. Performance Comparison for Multiple Concurrent Program Instances on a Single Compute-Node (2 Intel Xeon Silver 4216 CPUs; 32 Cores, 64 threads@2.1 GHz), Each Instance Using All Four Available Radeon VII GPUs, but Only a Proportional Share of the Total 64 CPU Threads (See Also Text for Technical Details)[a]

hardware	#instances	individual step time [s]	effective step time [s]
32 CPU cores	1	12.8	12.8
	2	20.0	10.0
	4	36.0	9.0
	8	68.7	8.6
	16	160.1	10.0
32 CPU cores + 4 GPUs	1	10.4	10.4
	2	10.5	5.2
	4	12.6	3.1
	8	20.3	2.5
	16	32.9	2.1

[a] All calculations employed PBEh-3c/mSVP with sn-LinK and RI-J.

The program instances run completely independently without any synchronization and each process has access to all four GPUs but only their proportional share of the total 64 CPU threads (e.g., 4 threads per instance at 16 program instances). In this way, the available hardware is better utilized because some instances can perform GPU-accelerated workloads (e.g., exact-exchange calculations with sn-LinK), while another instance computes CPU intensive workloads (e.g., generation of shell-pair data). Consequently, substantial speedups from this approach are only obtained in combination with GPU execution, for example, 5.0× speedup (2.1 s instead of 10.4 s) for PBEh-3c/mSVP and 16 instances *with* GPU acceleration, but only 1.5× (8.6 s instead of 12.8 s) for PBEh-3c/mSVP and 8 instances *without* GPUs.

In practice, each individual MD trajectory runs somewhat slower, but the overall throughput is greatly improved because multiple trajectories can be computed concurrently on the same node, thus improving the utilization of limited hardware resources. Because each process operates on private memory, this multi-instance approach results in proportionally increased overall memory demand, restricting the amount of possible instances, particularly for the more demanding calculation employing larger basis sets. Nevertheless, we consider this multi-instance approach very worthwhile considering possible speedups of up to 5-fold on top of the already impressive speedups from sn-LinK, RI-J, and GPU acceleration.

Overall, the performance that can be achieved with RI-J and sn-LinK together with GPU acceleration and multi-instancing is very impressive, for example, 2.1 s for PBEh-3c/mSVP (921 basis functions), 10.7 s for ωB97M-V/TZVP (1965 basis functions), and 90.2 s for ωB97M-V/QZVPPD (4698 basis functions), the latter improving the runtime per MD step 740-fold from 66,777.0 to 90.2 s.

6. ILLUSTRATIVE APPLICATION: HYDROGEN-BOND STRENGTH IN DS DNA

In the following, we illustrate a practical application of the fast AIMD method presented above (RI-J + sn-LinK), investigating the hydrogen bond strength in DS DNA as indicated by red and blue shifts of the corresponding covalent NH-bond vibrations. In the following, the atoms of the nucleobases are referenced by their IUPAC-numbering, represented in Figure 5.

In order to incorporate the influence of the surrounding DNA-bases and the aqueous solvent environment, the two model systems contain nine DNA base pairs including the sugar-phosphate backbone embedded in a water cube, where the central three pairs (excluding the sugar phosphate backbone) are treated quantum mechanically. The DS-DNA consists only of alternating AT- or CG-pairs. Details regarding the setup are given in Section S1 of the Supporting Information.

As an example for a strong hydrogen-bond, we present the VDoS spectrum of thymine-H_3 in Figure 6. The spectra of the other bridging H-atoms are given in Section S4 of the Supporting Information.

The near perfect similarity (i.e., within the statistically significance of the sampling error) between the two outer (at the edge of the QM region) base pairs and the inner DNA base pair proves the validity of the chosen QM/MM approach. Moreover, a very strong red-shift of over 500 cm^{-1} for the thymine−N_3−H_3 stretch vibration between the isolated monomer in vacuum (around 3600 cm^{-1}) and the double-helix indicates a very strong H-bond between thymine−H_3 and adenine−N_1. This red shift is a consequence of the weakened covalent thymine−N_3−H_3-bond, which is generally associated with hydrogen-bonding. On the other hand, the two thymine−N_3−H_3 deformation modes at ∼800 and ∼1500 cm^{-1}, respectively, are significantly blue-shifted compared to the monomer. This blue shift is due to the thymine−H_3−

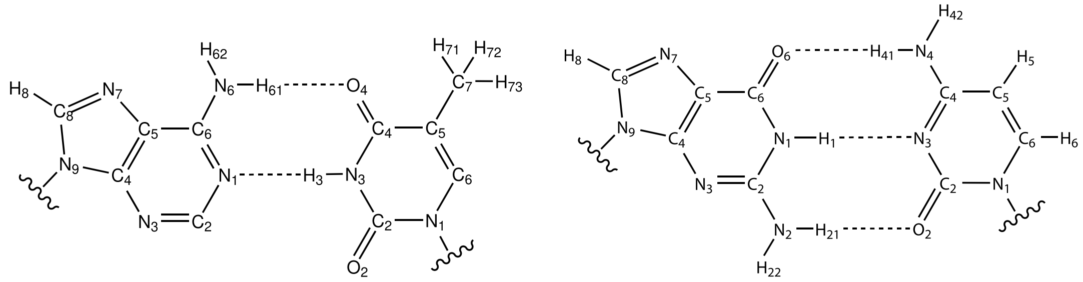


Figure 5. IUPAC-numbering of nucleobases in DNA base pairs. Left is an AT-pair and on the right is a GC-pair.

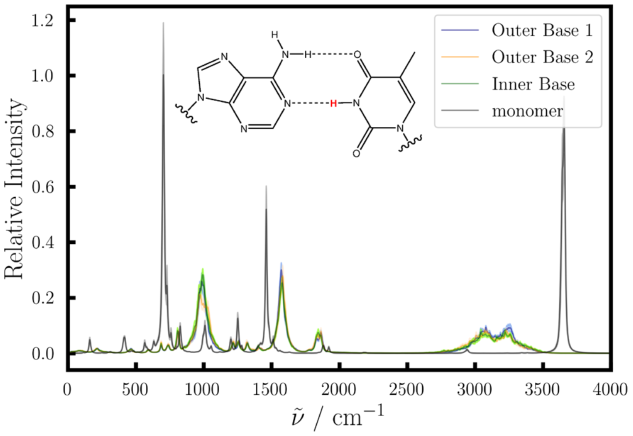


Figure 6. VDoS spectrum of thymine-H_3 in solvated DS DNA compared to the isolated monomer in vacuum. "Inner base" denotes the base-pair in the middle of the QM region, whereas "outer base 1" and "outer base 2" denote the two base pairs above and below at the edge of the QM region. The solid line represents the mean over 10 trajectories and the shaded regions the SEM.

adenine$-N_1$ H-bond getting partially broken by this vibrational mode, which leads to a steeper PES for this particular movement.

Thus, a H-bond is generally characterized by a red shift of the covalent X$-$H bond stretch vibration in the high-frequency region (>2500 cm^{-1}) and a blue shift in the respective bond deformation mode in the low-frequency region (<2500 cm^{-1}). Therefore, we decided to analyze these two regions separately to avoid these two effects canceling each other out. Because there are no peaks located between 2000 and 2800 cm^{-1}, the precise choice of this threshold is irrelevant for this analysis.

In order to measure these red/blue shifts with a single number, we compute the difference of the QM vibrational free energy (eq 25) between isolated monomer and DS complex, analogously to ref 26, but separated into a high- and low-frequency region at 2500 cm^{-1}. In contrast to the whole VDoS spectrum, which is always normalized to three (given the three degrees of freedom per atom), each individual part of the spectrum is not necessarily normalized. However, a consistent normalization is essential for the vibrational free energy analysis. Therefore, we decided to normalize each sub-spectrum to the mean norm of the respective monomeric sub-spectrum. Note, however, that the specific value of the normalization constant has no impact on the resulting ΔA

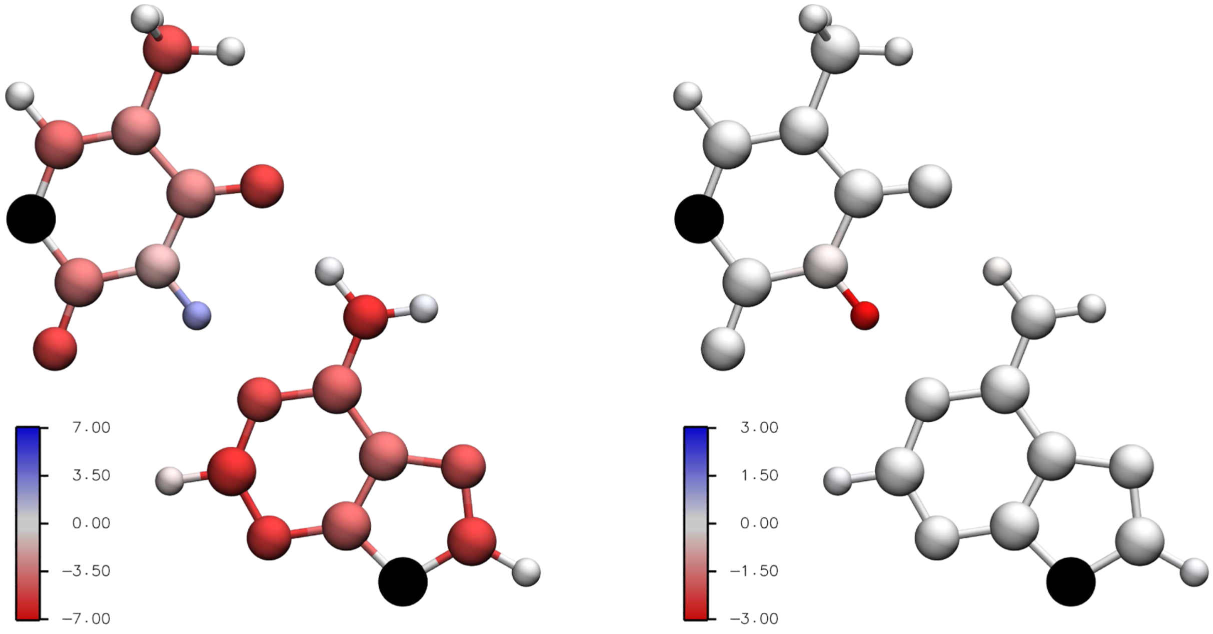


Figure 7. $\Delta A_{vib}{}^{QM}$ [kJ/mol] for the adenine−thymine dimer in DNA. The QM/MM-link-position (black) is removed from the analysis. Left: analysis for $\tilde{\nu}$ < 2500 cm^{-1}. Right: analysis for $\tilde{\nu}$ > 2500 cm^{-1}.

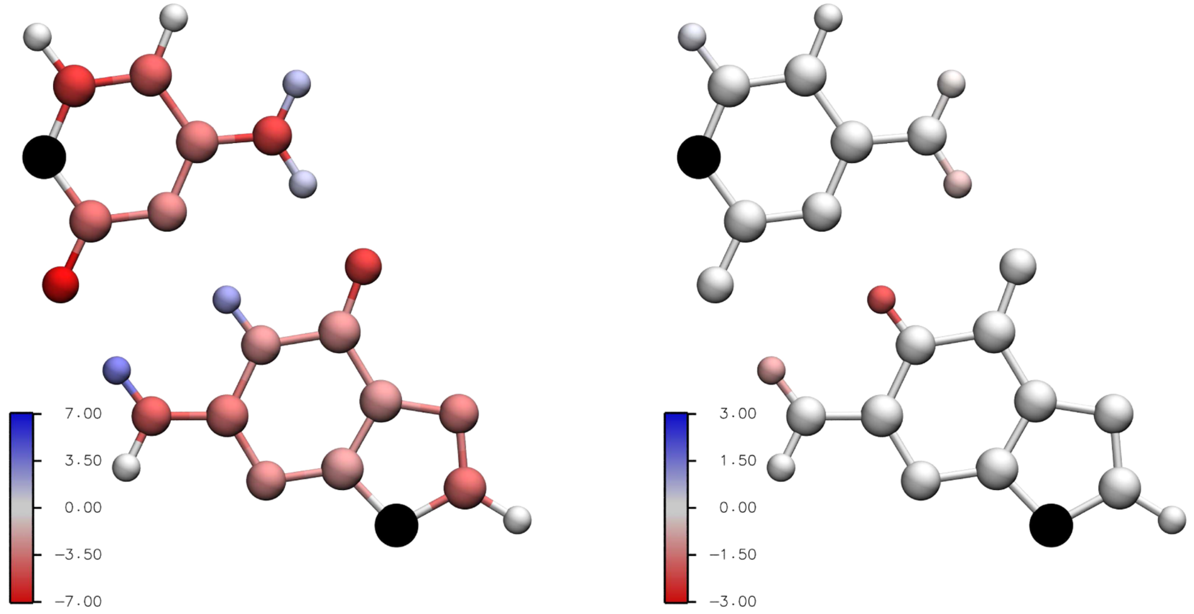


Figure 8. $\Delta A_{\text{vib}}^{\text{QM}}$ [kJ/mol] for the guanine−cytosine dimer in DNA. The QM/MM-link-position (black) is removed from the analysis. Left: analysis for $\tilde{\nu}$ < 2500 cm^{-1}. Right: analysis for $\tilde{\nu}$ > 2500 cm^{-1}.

values other than a constant proportionality factor, as long as it is employed consistently.

The results $\Delta A_{\text{vib}}^{\text{QM}}$ of this vibrational free-energy analysis are depicted in Figures 7 and 8 for adenine−thymine and guanine−cytosine, respectively.

In general, most atoms are red-shifted compared to the monomer in the low-frequency region due to the addition of extra low-frequency lattice-vibrations that are absent in the isolated monomer, cf. ref 26. Moreover, this analysis clearly identifies the strong hydrogen-bonds in DS DNA: in particular, the two non-amino N−H−N-bonds (T−H_3−N_1−A and G−H_1−N_3−C) are exceptionally strong, as characterized by a very strong red shift in the high-frequency part and a significant blue-shift in the low-frequency part. The three amino-hydrogen-bonds (A−H_{61}−O_4−T, G−H_{21}−O_2−C, and C−H_{41}−O_6−G) show less significant red/blue shifts, indicating that these contribute comparatively less to the stability of the DNA DS.

Due to the ability to clearly identify important hydrogen bonds in complex aggregates, we anticipate many other useful applications from the above hydrogen bond analysis, for example, solvent effects, supramolecular host−guest complexes, enzyme catalysis, or protein−drug binding. Hence, our illustrative example employing accelerated AIMD to study the H-bonds in DS DNA is also supposed to represent a protocol for other AIMD applications.

7. CONCLUSIONS AND OUTLOOK

We presented a highly efficient method to calculate the Coulomb energy and the exact-exchange energy together with the corresponding nuclear forces in one combined computation employing the RI approximation and seminumerical integration, respectively (RI-J + sn-LinK), and demonstrate its accuracy and performance in the context of QM/MM AIMD simulations. We found that—in stark contrast to single-point harmonic frequency analysis—accurate vibrational spectra obtained from such AIMD simulations can be obtained with comparatively coarse real-space integration grids. The impact of the simulation time step was found (as expected) to be more impactful, especially for the fastest vibrational modes.

The presented RI-J + sn-LinK combination allows for a significant improvement in the computational performance per MD time-step as compared to the conventional approach, particularly when using large basis sets because the expensive evaluation of the 4c2e integrals and their derivatives is completely avoided. In addition, computing energy and forces in one combined step instead of separately results in another 25% faster MD step time because many computationally expensive intermediates are required for both parts and thus do not have to be computed twice in this way. Further acceleration of AIMD simulations was achieved by running multiple program instances concurrently on a single node resulting in better utilization of the computational resources (especially GPUs) leading to an up to fivefold additional increase in overall MD throughput. With all of those optimization combined, AIMD simulations were accelerated between 19 and 740 times as compared to conventional methods.

Only with these significant performance gains at hand and having thoroughly accessed the accuracy of our methodology, the illustrative application, namely, quantifying the hydrogen bond strength within DS DNA, could be tackled. This study identified the two non-amino H-atoms thymine−H_3 and guanine−H_1 as the dominant hydrogen bonds in DS DNA.

To conclude, fast ab initio MD methods are a necessary requirement for such studies of dynamic behavior in a complex environment. Consequently, the development of highly efficient AIMD propagation methods, such as the seminumerical exchange gradients presented in this work, is essential to the advancement of the field.

■ DATA AVAILABILITY

The data that support the findings of this study are available upon request. In addition, we plan to release our FermiONs++ program in the future.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.2c00509.

> Details regarding the QM/MM embedding and the generation of start geometries for the AIMD simulations, impact of finite integration grids on the conservation of total energy within microcanonical MD simulations, detailed listings of the performance gains from running multiple concurrent program instances, and VDoS spectra of all five hydrogen atoms participating in hydrogen bonding between DNA strands (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Christian Ochsenfeld − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU), D-81377 München, Germany; Max Planck Institute for Solid State Research, 70569 Stuttgart, Germany;* orcid.org/0000-0002-4189-6558; Email: christian.ochsenfeld@uni-muenchen.de

Authors

Henryk Laqua − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU), D-81377 München, Germany*

Johannes C. B. Dietschreit − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU), D-81377 München, Germany;* Present Address: Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; orcid.org/0000-0002-5840-0002

Jörg Kussmann − *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU), D-81377 München, Germany;* orcid.org/0000-0002-4724-8551

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.2c00509

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors acknowledge financial support by the “Deutsche Forschungsgemeinschaft” (DFG) via the SFB 1309-32587107 and the cluster of excellence (EXC2111-390814868) “Munich Center for Quantum Science and Technology” (MCQST). C.O. further acknowledges financial support as Max-Planck-Fellow at the MPI-FKF Stuttgart.

■ REFERENCES

(1) Mintmire, J. W.; Dunlap, B. I. Fitting the Coulomb potential variationally in linear-combination-of-atomic-orbitals density-functional calculations. *Phys. Rev. A* **1982**, *25*, 88−95.

(2) Kussmann, J.; Laqua, H.; Ochsenfeld, C. Highly Efficient Resolution-of-Identity Density Functional Theory Calculations on Central and Graphics Processing Units. *J. Chem. Theory Comput.* **2021**, *17*, 1512−1521.

(3) White, C. A.; Head-Gordon, M. A J matrix engine for density functional theory calculations. *J. Chem. Phys.* **1996**, *104*, 2620−2629.

(4) Shao, Y.; Head-Gordon, M. An improved J matrix engine for density functional theory calculations. *Chem. Phys. Lett.* **2000**, *323*, 425−433.

(5) Neese, F. An improvement of the resolution of the identity approximation for the formation of the Coulomb matrix. *J. Comput. Chem.* **2003**, *24*, 1740−1747.

(6) Friesner, R. A. Solution of self-consistent field electronic structure equations by a pseudospectral method. *Chem. Phys. Lett.* **1985**, *116*, 39−43.

(7) Friesner, R. A. Solution of the Hartree-Fock equations for polyatomic molecules by a pseudospectral method. *J. Chem. Phys.* **1987**, *86*, 3522−3531.

(8) Won, Y.; Lee, J. G.; Ringnalda, M. N.; Friesner, R. A. Pseudospectral Hartree-Fock gradient calculations. *J. Chem. Phys.* **1991**, *94*, 8152−8157.

(9) Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. Efficient, approximate and parallel Hartree-Fock and hybrid DFT calculations. A ’chain-of-spheres’ algorithm for the Hartree-Fock exchange. *Chem. Phys.* **2009**, *356*, 98−109.

(10) Plessow, P.; Weigend, F. Seminumerical calculation of the Hartree-Fock exchange matrix: Application to two-component procedures and efficient evaluation of local hybrid density functionals. *J. Comput. Chem.* **2012**, *33*, 810−816.

(11) Bahmann, H.; Kaupp, M. Efficient Self-Consistent Implementation of Local Hybrid Functionals. *J. Chem. Theory Comput.* **2015**, *11*, 1540−1548.

(12) Klawohn, S.; Bahmann, H.; Kaupp, M. Implementation of Molecular Gradients for Local Hybrid Density Functionals Using Seminumerical Integration Techniques. *J. Chem. Theory Comput.* **2016**, *12*, 4254−4262.

(13) Liu, F.; Kong, J. Efficient Computation of Exchange Energy Density with Gaussian Basis Functions. *J. Chem. Theory Comput.* **2017**, *13*, 2571−2580.

(14) Holzer, C. An improved seminumerical Coulomb and exchange algorithm for properties and excited states in modern density functional theory. *J. Chem. Phys.* **2020**, *153*, 184115.

(15) Laqua, H.; Thompson, T. H.; Kussmann, J.; Ochsenfeld, C. Highly Efficient, Linear-Scaling Seminumerical Exact-Exchange Method for Graphic Processing Units. *J. Chem. Theory Comput.* **2020**, *16*, 1456−1468.

(16) Laqua, H.; Kussmann, J.; Ochsenfeld, C. Accelerating seminumerical Fock-exchange calculations using mixed single- and double-precision arithmethic. *J. Chem. Phys.* **2021**, *154*, 214116.

(17) Niklasson, A. M. N.; Tymczak, C. J.; Challacombe, M. Time-Reversible Born-Oppenheimer Molecular Dynamics. *Phys. Rev. Lett.* **2006**, *97*, 123001.

(18) Niklasson, A. M. N. Extended Born-Oppenheimer Molecular Dynamics. *Phys. Rev. Lett.* **2008**, *100*, 123004.

(19) Niklasson, A. M. N.; Steneteg, P.; Odell, A.; Bock, N.; Challacombe, M.; Tymczak, C. J.; Holmström, E.; Zheng, G.; Weber, V. Extended Lagrangian Born-Oppenheimer molecular dynamics with dissipation. *J. Chem. Phys.* **2009**, *130*, 214109.

(20) Cawkwell, M. J.; Niklasson, A. M. N. Energy conserving, linear scaling Born-Oppenheimer molecular dynamics. *J. Chem. Phys.* **2012**, *137*, 134105.

(21) Peters, L. D. M.; Kussmann, J.; Ochsenfeld, C. Efficient and Accurate Born-Oppenheimer Molecular Dynamics for Large Molecular Systems. *J. Chem. Theory Comput.* **2017**, *13*, 5479−5485.

(22) Thomas, M.; Brehm, M.; Fligg, R.; Vöhringer, P.; Kirchner, B. Computing vibrational spectra from ab initio molecular dynamics. *Phys. Chem. Chem. Phys.* **2013**, *15*, 6608−6622.

(23) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. Linear and sublinear scaling formation of Hartree-Fock-type exchange matrices. *J. Chem. Phys.* **1998**, *109*, 1663−1669.

(24) Ochsenfeld, C. Linear scaling exchange gradients for Hartree-Fock and hybrid density functional theory. *Chem. Phys. Lett.* **2000**, *327*, 216−223.

(25) Peters, L. D. M.; Dietschreit, J. C. B.; Kussmann, J.; Ochsenfeld, C. Calculating free energies from the vibrational density of states function: Validation and critical assessment. *J. Chem. Phys.* **2019**, *150*, 194111.

(26) Dietschreit, J. C. B.; Peters, L. D. M.; Kussmann, J.; Ochsenfeld, C. Identifying Free Energy Hot-Spots in Molecular Transformations. *J. Phys. Chem. A* **2019**, *123*, 2163−2170.

(27) *SymPy Version 1.1.1*, see https://www.sympy.org.

(28) Obara, S.; Saika, A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1986**, *84*, 3963−3974.

(29) Obara, S.; Saika, A. General recurrence formulas for molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1988**, *89*, 1540−1559.

(30) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(31) 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.

(32) Laqua, H.; Kussmann, J.; Ochsenfeld, C. Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals. *J. Chem. Theory Comput.* **2018**, *14*, 3451−3458.

(33) Laqua, H.; Kussmann, J.; Ochsenfeld, C. An improved molecular partitioning scheme for numerical quadratures in density functional theory. *J. Chem. Phys.* **2018**, *149*, 204111.

(34) Grimme, S.; Brandenburg, J. G.; Bannwarth, C.; Hansen, A. Consistent structures and interactions by density functional theory with small atomic orbital basis sets. *J. Chem. Phys.* **2015**, *143*, 054107.

(35) Weigend, F. Accurate Coulomb-fitting basis sets for H to Rn. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1057−1065.

(36) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, *126*, 014101.

(37) Berens, P. H.; Mackay, D. H. J.; White, G. M.; Wilson, K. R. Thermodynamics and quantum corrections from molecular dynamics for liquid water. *J. Chem. Phys.* **1983**, *79*, 2375−2389.

(38) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(39) Burke, K.; Ernzerhof, M.; Perdew, J. P. The adiabatic connection method: A non-empirical hybrid. *Chem. Phys. Lett.* **1997**, *265*, 115−120.

(40) Adamo, C.; Barone, V. Toward reliable density functional methods without adjustable parameters: the PBE0 model. *J. Chem. Phys.* **1999**, *110*, 6158−6170.

(41) Ernzerhof, M.; Scuseria, G. E. Assessment of the Perdew-Burke-Ernzerhof exchange-correlation functional. *J. Chem. Phys.* **1999**, *110*, 5029−5036.

(42) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.

(43) Vydrov, O. A.; Van Voorhis, T. Nonlocal van der Waals density functional: the simpler the better. *J. Chem. Phys.* **2010**, *133*, 244103.

(44) Mardirossian, N.; Head-Gordon, M. ωB97M-V: A combinatorially optimized, range-separated hybrid, meta-GGA density functional with VV10 nonlocal correlation. *J. Chem. Phys.* **2016**, *144*, 214110.

(45) Field, M. J.; Bash, P. A.; Karplus, M. A combined quantum mechanical and molecular mechanical potential for molecular dynamics simulations. *J. Comput. Chem.* **1990**, *11*, 700−733.

Supporting Information to the Paper: Accelerating Hybrid Density Functional Theory Molecular Dynamic Simulations by Seminumerical Integration, Resolution-of-the-Identity Approximation, and Graphics Processing Units

Henryk Laqua,[†] Johannes C. B. Dietschreit,[†,¶] Jörg Kussmann,[†] and Christian Ochsenfeld[*,†,‡]

† *Department of Chemistry, Chair of Theoretical Chemistry, University of Munich (LMU), D-81377 München, Germany*

‡ *Max Planck Institute for Solid State Research, 70569 Stuttgart, Germany.*

¶ *Present address: Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

E-mail: christian.ochsenfeld@uni-muenchen.de

S1 Details on Molecular Dynamics

This section reports on the specific MD settings and the generation of start structures. Regarding the AIMD simulations, the temperature was maintained at 300 K (unless stated otherwise) with the Bussi-Donadio-Parinello thermostat.[1] For every simulation setting ten simulations of 20 ps length were carried out to allow for error statistics. Monomeric simulations in vacuum were initialized from the global minimum structure using different random number seeds which affects the initial vectors drawn from the Maxwell-Boltzmann distribution as well as the thermostat. The QM/MM simulations of the double stranded DNA were (other than the monomers) initialized from different starting geometries and velocities.

The QM/MM simulations for the AT and CG pairs contained a single double strand with 9 base pairs with either sequence ATATATATA or CGCGCGCGC, respectively. The double-strands were generated in an ideal Z-DNA configuration. Each DS was then solvated in a cube water with 40 Å side length and neutralized with sodium ions using tleap from the Ambertools.[2] The water molecules were described with the TIP3P[3] force field; the DNA with the OL15 force-field.[4] The resulting model systems contain 9293 and 9074 atoms for adenine-thymine and guanine-cytosine systems, respectively.

To generate initial configurations for the QM/MM simulations, the simulation engine NAMD 2.10[5] was used: First, the solvated DS-DNA systems were minimized, for the first 10,000 steps only the water molecules, followed by the full system for another 10,000 steps. The two obtained structures (AT and GC) were gradually heated to 300 K over a period of 30 ps and subsequently equilibrated for 200 ps. Then, a single force-field MD run of 100 ns was carried out for both systems (AT and GC) under periodic boundary conditions, employing Langevin dynamics for temperature control with a friction constant of 1 ps^{-1}, and the Langevin piston Nosé-Hoover method for pressure control. For this simulation we chose a time step of 2.0 fs, while constraining covalent bond-length using RATTLE.[6] Periodic electrostatic interactions were computed with the particle mesh Ewald summation method, with a 6th order interpolation, a cut-off radius of 12 Å and a smooth switching function damping long-range interaction between 10 and 12 Å. Furthermore, a Verlet nearest neighbour list with a radius of 13.5 Å was used. The QM/MM AIMD start structures were then taken out of this force-field MD run, where each start frame was spaced exactly 10 ns apart.

In the QM/MM AIMD simulations, the central three base pairs were treated quantum-mechanically, where only the base of each nucleotide (i.e., not the sugar-phosphate backbone) was included in the QM region. The glycosidic bond between sugar and base contained the link atom, resulting in a total of 90 and 87 QM atoms for AT and GC, respectively. The interactions between QM and MM were treated with electrostatic embedding.[7]

S2 Impact of Integration Grids on Conservation of Energy

Since the semi-numerical exact-exchange forces are only the exact derivatives of the energy with respect to the nuclear position in the limit of infinite integration grids, energy conservation in a microcanonical MD simulation is slightly violated, which, of course, depends strongly on the grid size. As demonstrated in Figure S1, the total energy drifts randomly by up to 100 μE_h for the smallest (gm3) grid.

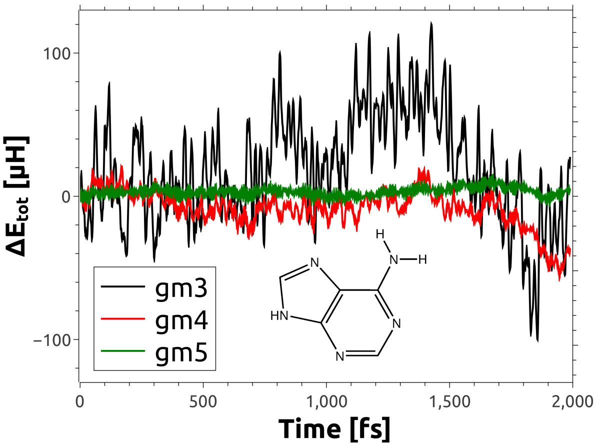


Figure S1: Change of total energy during a 2 ps MD simulation of adenine *in vacuo* employing no thermostat (microcanonical ensemble) for different numerical integration grids (gm3, gm4, gm5).

With increased grid size, however, this numerical artifact quickly vanishes, e.g., for the gm5 grid, this energy fluctuation is decreased to only a few micro-Hartrees. Moreover, this effect is only problematic for microcanonical MD simulations, since within canonical simulations the total energy within the system is, of course, never conserved in the first place due to the influence of the thermostat.

S3 Performance Gains from Multiple Program Instances

In Tables S1 to S3 we provide more details on the AIMD throughput improvements from running multiple program processes concurrently on the same compute node, c.f. Section 5 of the main manuscript.

Table S1: Performance comparison for multiple concurrent program instances on a single compute-node (2 Intel Xeon Silver 4216 CPUs; 32 cores, 64 threads@2.1 GHz), each instance using all four available Radeon VII GPUs, but only a proportional share of the total 64 CPU threads The results are given for PBEh-3c/def2-mSVP employing RI-J/sn-LinK (same as Table 3 of the main manuscript).

Hardware	# instances	individual step time [s]	effective step time [s]
32 CPU cores	1	12.8	12.8
	2	20.0	10.0
	4	36.0	9.0
	8	68.7	8.6
	16	160.1	10.0
32 CPU cores + 4 GPUs	1	10.4	10.4
	2	10.5	5.2
	4	12.6	3.1
	8	20.3	2.5
	16	32.9	2.1

In particular, we want to emphasize that GPU accelerated computations benefit most from this form of concurrent execution, since the available hardware is better utilized if, e.g., one instance performs a GPU intensive task (e.g., Fock-build), whereas another instance performs a CPU intensive workload (e.g., diagonalization of the Fock-matrix).

Table S2: Same as Table S1 but for ωB97M-V/def2-TZVP.

Hardware	# instances	individual step time [s]	effective step time [s]
32 CPU cores	1	69.5	69.5
	2	118	58.8
	4	221	55.3
	8	461	57.6
32 CPU cores + 4 GPUs	1	29.4	29.4
	2	37.9	19.0
	4	51.1	12.8
	8	86.0	10.7

Table S3: Same as Table S1 but for ωB97M-V/def2-QZVPPD.

Hardware	# instances	individual step time [s]	effective step time [s]
32 CPU cores	1	592	592
	2	1060	532
	4	2040	509
32 CPU cores + 4 GPUs	1	176	176
	2	232	116
	4	361	90.2

S4 Velocity Density of State Spectra for all bridging H-Atoms in DNA

In Figures S2 to S6 we present the vibrational density of state (VDoS) spectra of the bridging hydrogen atoms in DNA analogously to Figure 7 of the main manuscript. The high-frequency red-shifts and the low-frequency blue-shifts are – in accordance with the discussion of Section 6 of the main article – particularly strong for the two non-amino H-atoms thymine-H_3 and guanine-H_1, while the effect is substantially weaker for the three amino-H atoms A-H_{61}, C-H_{41}, and G-H_{21}, indicating that these contribute less to the stability of the DNA double-strand.

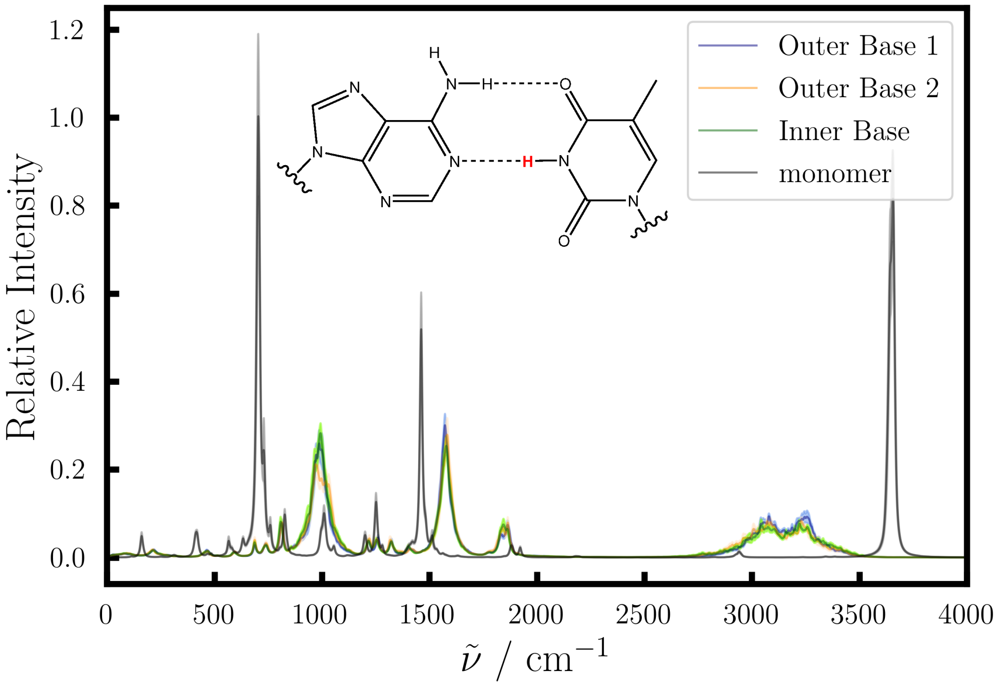


Figure S2: VDoS spectrum of thymine-H_3 in solvated double-stranded DNA compared to the isolated monomer in vacuum (same as Figure 7 of the main manuscript).

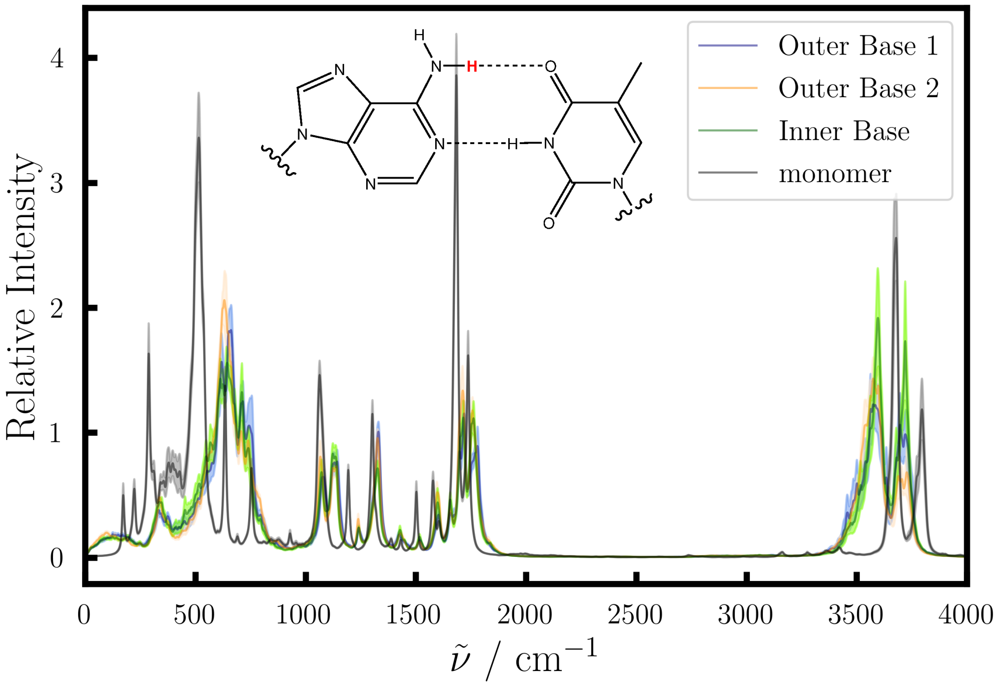


Figure S3: Same as Figure S2 but for adenine-H_{61}.

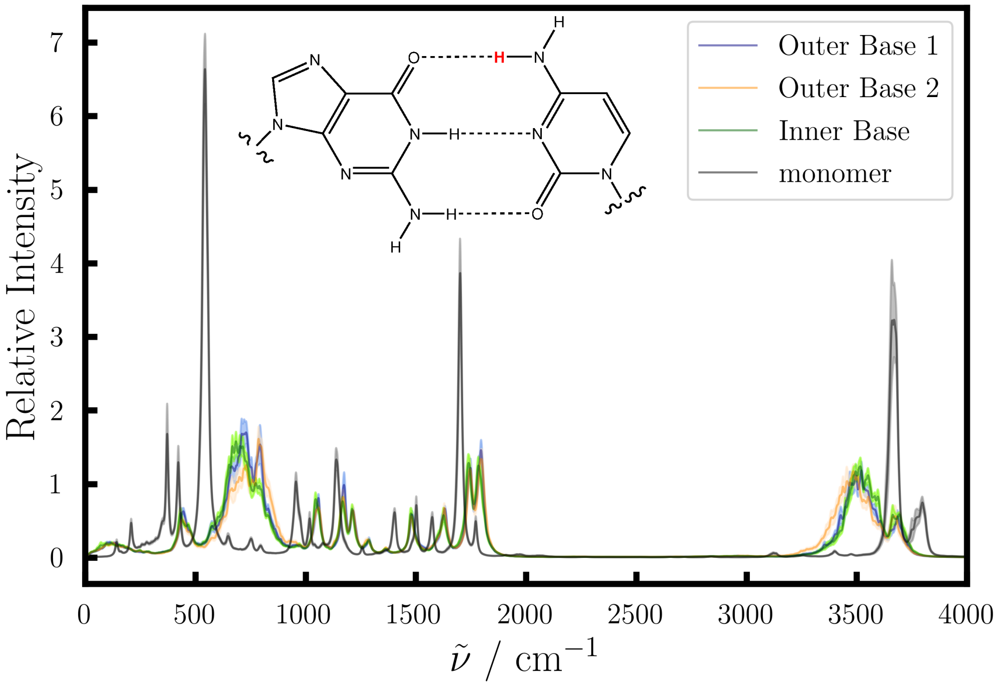


Figure S4: Same as Figure S2 but for cytosine-H_{41} within the cytosine-guanine dimer.

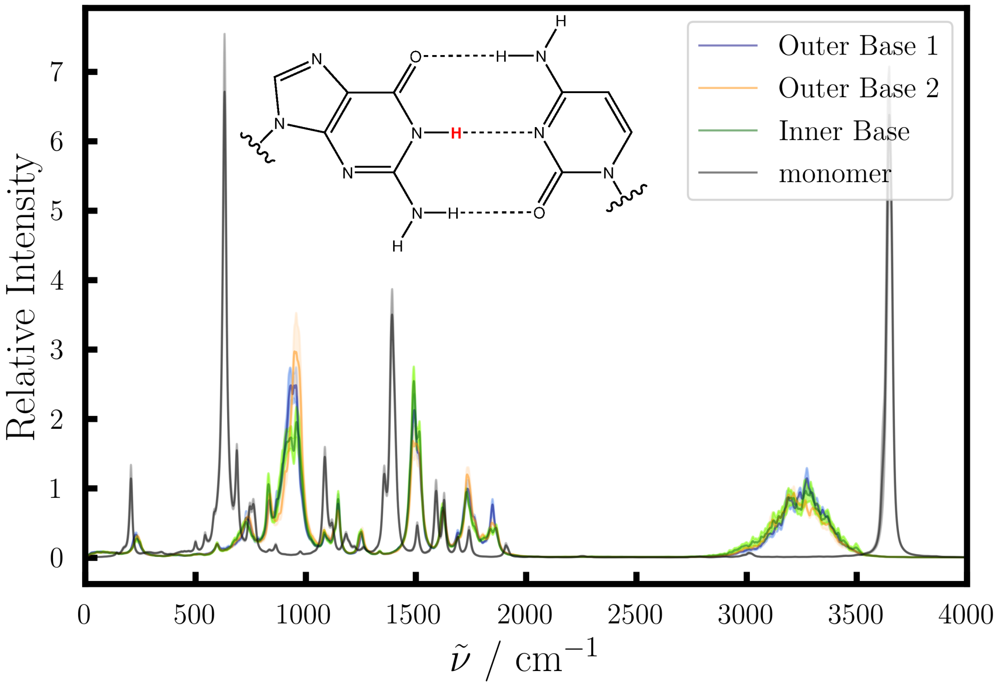


Figure S5: Same as Figure S4 but for guanine-H_1.

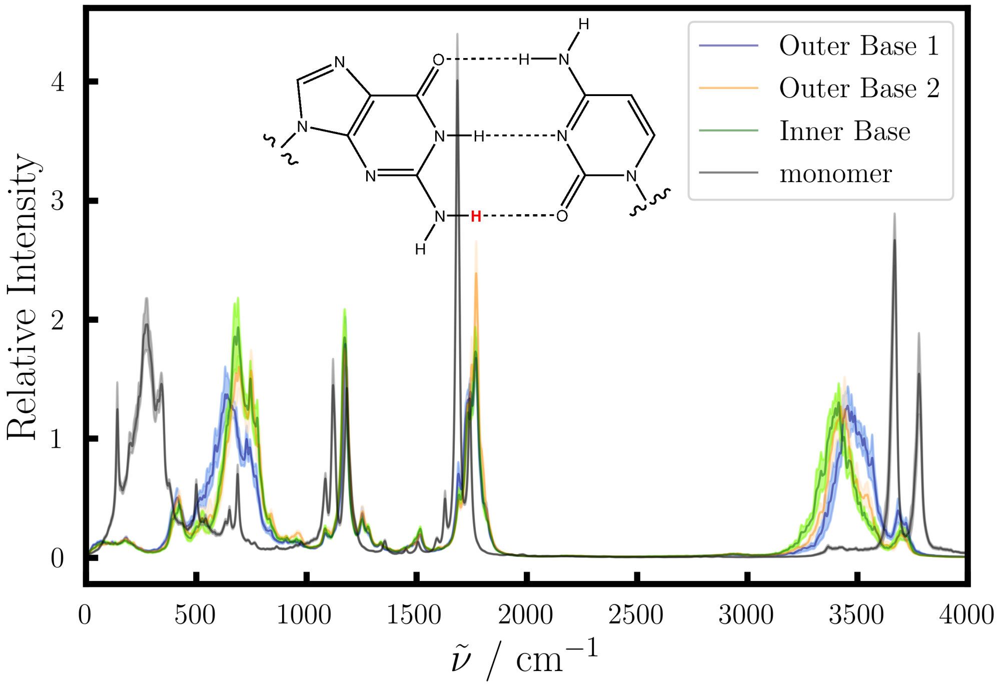


Figure S6: Same as Figure S4 but for guanine-H_{21}.

References

(1) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, *126*, 014101.

(2) Case, D.; Cerutti, D.; Cheatham, T.; Darden, T.; Duke, R.; Giese, T.; Gohlke, H.; Goetz, A.; Greene, D.; Homeyer, N.; Izadi, S.; Kovalenko, A.; Lee, T.; LeGrand, S.; Li, P.; Lin, C.; Liu, J.; Luchko, T.; Luo, R.; Mermelstein, D.; Merz, K.; Monard, G.; Nguyen, H.; Omelyan, I.; Onufriev, A.; Pan, F.; Qi, R.; Roe, D.; Roitberg, A.; Sagui, C.; Simmerling, C.; Botello-Smith, W.; Swails, J.; Walker, R.; Wang, J.; Wolf, R.; Wu, X.; Xiao, L.; York, D.; Kollman, P. AMBER 2016. 2016.

(3) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics* **1983**, *79*, 926–935.

(4) Galindo-Murillo, R.; Robertson, J. C.; Zgarbovic, M.; Sponer, J.; Otyepka, M.; Jureska, P.; Cheatham, T. E. Assessing the Current State of Amber Force Field Modifications for DNA. *J. Chem. Theory Comput.* **2016**, *12*, 4114–4127.

(5) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

(6) Andersen, H. C. Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics* **1983**, *52*, 24–34.

(7) Field, M. J.; Bash, P. A.; Karplus, M. A combined quantum mechanical and molecular mechanical potential for molecular dynamics simulations. *J. Comput. Chem.* **1990**, *11*, 700–733.

3.5 Publication V: Highly Efficient and Accurate Computation of Multiple Orbital Spaces Spanning Fock Matrix Elements on Central and Graphics Processing Units for Application in F12 Theory

L. Urban, H. Laqua, C. Ochsenfeld

J. Chem. Theory Comput. **18**, 4218 (2022).

Abstract

We employ our recently published highly efficient seminumerical exchange (sn-LinK) [Laqua, H.; Thompson, T. H.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Theory Comput.* **2020**, *16*, 1456–1468] and integral-direct resolution of the identity Coulomb (RI-J) [Kussmann, J.; Laqua, H.; Ochsenfeld, C. *J. Chem. Theory Comput.* **2021**, *17*, 1512–1521] methods to significantly accelerate the computation of the demanding multiple orbital spaces spanning Fock matrix elements present in R12/F12 theory on central and graphics processing units. The errors introduced by RI-J and sn-LinK into the RI-MP2-F12 energy are thoroughly assessed for a variety of basis sets and integration grids. We find that these numerical errors are always below "chemical accuracy" ($\sim$1 mH) even for the coarsest settings and can easily be reduced below 1 μH by employing only moderately large integration grids and RI-J basis sets. Since the number of basis functions of the multiple orbital spaces is notably larger compared with conventional Hartree-Fock theory, the efficiency gains from the superior basis scaling of RI-J and sn-LinK ($\mathcal{O}(N_{\text{bas}}^2)$ instead of $\mathcal{O}(N_{\text{bas}}^4)$ for both) are even more significant, with maximum speedup factors of 37000 for RI-J and 4500 for sn-LinK. In total, the multiple orbital spaces spanning Fock matrix evaluation of the largest tested structure using a triple-ζ F12 basis set (5058 AO basis functions, 9267 CABS basis functions) is accelerated over 1575×× using CPUs and over 4155× employing GPUs.

Reprinted with permission from:

L. Urban, H. Laqua, C. Ochsenfeld
" Highly Efficient and Accurate Computation of Multiple Orbital Spaces Spanning Fock Matrix Elements on Central and Graphics Processing Units for Application in F12 Theory"
J. Chem. Theory Comput. **18**, 4218 (2022).

https://pubs.acs.org/doi/10.1021/acs.jctc.2c00215

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

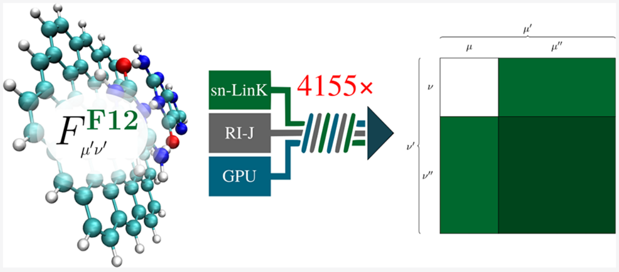
Highly Efficient and Accurate Computation of Multiple Orbital Spaces Spanning Fock Matrix Elements on Central and Graphics Processing Units for Application in F12 Theory

Lars Urban, Henryk Laqua, and Christian Ochsenfeld*

Cite This: https://doi.org/10.1021/acs.jctc.2c00215 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We employ our recently published highly efficient seminumerical exchange (sn-LinK) [Laqua, H.; Thompson, T. H.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Theory Comput.* **2020**, *16*, 1456−1468] and integral-direct resolution of the identity Coulomb (RI-J) [Kussmann, J.; Laqua, H.; Ochsenfeld, C. *J. Chem. Theory Comput.* **2021**, *17*, 1512−1521] methods to significantly accelerate the computation of the demanding multiple orbital spaces spanning Fock matrix elements present in R12/F12 theory on central and graphics processing units. The errors introduced by RI-J and sn-LinK into the RI-MP2-F12 energy are thoroughly assessed for a variety of basis sets and integration grids. We find that these numerical errors are always below "chemical accuracy" (∼1 mH) even for the coarsest settings and can easily be reduced below 1 μH by employing only moderately large integration grids and RI-J basis sets. Since the number of basis functions of the multiple orbital spaces is notably larger compared with conventional Hartree−Fock theory, the efficiency gains from the superior basis scaling of RI-J and sn-LinK ($O(N_{\mathrm{bas}}{}^2)$ instead of $O(N_{\mathrm{bas}}{}^4)$ for both) are even more significant, with maximum speedup factors of 37 000 for RI-J and 4500 for sn-LinK. In total, the multiple orbital spaces spanning Fock matrix evaluation of the largest tested structure using a triple-ζ F12 basis set (5058 AO basis functions, 9267 CABS basis functions) is accelerated over 1575× using CPUs and over 4155× employing GPUs.



1. INTRODUCTION

Over the last decades, considerable efforts[1] have been devoted to accelerate the computation and contraction of direct- and exchange-type four-center−two-electron (4c-2e) integrals present in Hartree−Fock[2−5] (HF) and Kohn−Sham[6] (KS) theory, which in their conventional formulation show an unfavorable scaling of $O(N_{\mathrm{bas}}{}^4)$ with the number atomic orbital (AO) basis functions N_{bas}. Among these aspirations are methods focusing on the efficient contraction of 4c-2e integrals[7−9] and attempts like the resolution of the identity (RI) approximation,[10−12] a frequently used approach to reduce the computational prefactor substantially. Especially the application of the RI technique for Coulomb contributions (RI-J)[13,14] yields significant speedups, which are additionally improved when RI-J is combined with the J-engine algorithm.[15−17] Here, our recently published[18] integral-direct RI-J version additionally accelerates the evaluation of the occurring three-center−two-electron (3c-2e) integrals, reducing the number of floating-point operations (FLOPs) and required local memory, which makes the method more efficient for both central processing units (CPUs) as well as graphics processing units (GPUs).

However, the relative RI efficiency for exchange contributions (RI-K)[12,19−21] succumbs to advanced RI-J algorithms due to the comparable formal $O(N_{\mathrm{occ}}N_{\mathrm{aux}}N_{\mathrm{bas}}{}^2)$ scaling as the conventional evaluation when not combined with local approximations.[22,23] Here, seminumerical integration,[24−35] where one electronic coordinate of the 4c-2e integrals is represented numerically on real-space integration grids while the other one remains in its analytical representation, provides a promising alternative. For maximal performance, the number of grid points N_{grid} has to be as small as possible, and the computationally demanding evaluation of the necessary three-center−one-electron (3c-1e) integrals needs to be as efficient as possible. Recent efforts by our group systematically refined both of these aspects by introducing revised molecular integration grids[36−38] and a more effective batchwise integral screening method (sn-LinK).[39−41] Moreover, the possibility to parallelize over the grid index within the numerical integration combined with the reduced demand for local storage (e.g., L1 cache) to evaluate the 3c-1e integrals compared with the 4c-2e

Received: March 2, 2022

integrals makes this method for computing exchange contributions particularly well suited for GPU acceleration.

Besides HF and KS theory, the computation of Fock matrix elements is essential for R12/F12 methods,[42−47] which are powerful tools to overcome the basis set incompleteness error (BSIE). These elements, denoted in the following as F12-type Fock matrix elements, need to be evaluated for multiple orbital spaces introduced by the strong orthogonality operator $\hat{Q}_{12}$, which notably increases the number of required basis functions. In general, the formally quartic-scaling evaluation of the direct and exchange contributions to F12-type Fock matrices is thus substantially more demanding than the evaluation of normal Fock matrices and frequently represents an extremely expensive step in applications of F12 theory.[44,48,49] Efficient evaluation is thus highly beneficial since a series of theories require these elements, among them the complementary auxiliary basis set (CABS) singles correction,[48,50] explicitly correlated second-order Møller−Plesset perturbation theory (MP2-R12/F12),[43,44,49,51−57] coupled cluster-F12 (CC-F12),[48,50,58−64] multireference-F12 (MR-F12),[65−72] and other explicitly correlated approaches.[73−75] Previous works applied RI[49,51,53,54,72] and seminumerical integral[76] approaches but focused primarily on other aspects of F12 theory, not the investigation of their influence on accuracy and efficiency. Motivated by these circumstances, we transferred our recently introduced RI-J and sn-LinK methods to F12 theory. Both of these are well-suited because of the improved formal $\mathcal{O}(N_{\text{bas}}{}^2)$ scaling with respect to the AO basis set size compared with the conventional $\mathcal{O}(N_{\text{bas}}{}^4)$ scaling, which is particularly relevant in view of the substantially larger size of the combined CABS (N_{CABS}) and AO basis.

The paper is structured as follows: We present the necessary underlying formulas for extending RI-J and sn-LinK to F12 theory in section 2, followed by the most important findings regarding accuracy in section 4.1 and efficiency in section 4.2 (additional data are provided in the Supporting Information).

2. THEORY

2.1. Approximation-Free Evaluation of F12-Type Fock Matrix Elements. In contrast to HF theory, Fock matrix elements in explicitly correlated F12 theory span the additional orbital spaces given in Table 1. The AO representation of the F12-type Fock matrix, visualized in Figure 1a, differs from the standard HF/KS approach by evaluation of $\{\mu'\}$, the union of the atomic orbitals $\{\mu\}$ and the CABS atomic orbitals $\{\mu''\}$, as given by

$$F_{\mu'\nu'} = H^{\text{core}}_{\mu'\nu'} + J_{\mu'\nu'} - \frac{1}{2}K_{\mu'\nu'} \tag{1}$$

where $H^{\text{core}}_{\mu'\nu'}$, $J_{\mu'\nu'}$, and $K_{\mu'\nu'}$ are elements of the core-Hamiltonian matrix, Coulomb matrix, and exchange matrix of the combined orbital space, respectively. While $H^{\text{core}}_{\mu'\nu'}$ contributions are trivial to evaluate with insignificant computational cost, the $J_{\mu'\nu'}$ and $K_{\mu'\nu'}$ elements are calculated via

$$J_{\mu'\nu'} = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu'\nu'|\lambda\sigma) \tag{2}$$

$$K_{\mu'\nu'} = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu'\sigma|\lambda\nu') \tag{3}$$

where $P_{\lambda\sigma}$ are the elements of the density matrix of the final SCF iteration in the AO space, and the computationally demanding 4c-2e integrals are given by

$$(\mu'\nu'|\lambda\sigma) = \iint \mathrm{d}\mathbf{r}_1\, \mathrm{d}\mathbf{r}_2\, \chi_{\mu'}(\mathbf{r}_1)\chi_{\nu'}(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\chi_\lambda(\mathbf{r}_2)\chi_\sigma(\mathbf{r}_2) \tag{4}$$

Table 1. Summary of Orbital Spaces and Indexing Conventions

orbital space	indices
AO space	$\mu, \nu, \lambda, \sigma$
AO complementary auxiliary space	$\mu'', \nu'', \lambda'', \sigma''$
combined AO space ($\{\mu\} \cup \{\mu''\}$)	$\mu', \nu', \lambda', \sigma'$
MO occupied space	i, j, k, l
MO virtual space	a, b, c, d
MO occupied + virtual space ($\{i\} \cup \{a\}$)	p, q, r, s
MO complementary auxiliary space	p'', q'', r'', s''
combined MO CABS/HF space ($\{p\} \cup \{p''\}$)	p', q', r', s'
RI-J auxiliary space	P, Q

After construction of the AO F12-type Fock matrix, the transformation into the molecular orbital (MO) space is achieved by contraction with the MO coefficients from the final SCF iteration and precomputed CABS coefficients.[46] Figure 1b visualizes the different MO spaces present in F12 theory and the resulting F12-type Fock matrix elements. The MO transformation and the determination of the required CABS coefficients require only minor computational effort. In contrast, the evaluations of the **J** and **K** matrices are computationally intensive procedures. The usually large CABS space further increases the cost of the direct- and exchange-type matrix element evaluation, leading to about an order of magnitude longer runtimes for computations using the F12-type Fock matrix compared with the standard SCF Fock matrix. Increasing the angular momentum quantum number (l) amplifies this effect, making more efficient methods having reduced time complexity with respect to the basis set size highly desirable.

2.2. An Integral-Direct J-Engine-Based Resolution of the Identity Coulomb Method. In this section, we briefly summarize the necessary theory for the integral-direct RI-J method that we employ for the F12-type Fock matrix element evaluation. For more insights and illustrative calculations, we refer the reader to the original literature.[17,18] Applying the RI approximation to the Coulomb potential leads to

$$\begin{aligned} J_{\mu'\nu'} &= \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu'\nu'|\lambda\sigma) \\ &\approx \sum_{\lambda\sigma}\sum_{PQ} P_{\lambda\sigma}(\mu'\nu'|P)(P|Q)^{-1}(Q|\lambda\sigma) \end{aligned} \tag{5}$$

where for an integral-direct algorithm three consecutive steps are executed:

$$\text{step 1:}\quad J_P = \sum_{\lambda\sigma}(P|\lambda\sigma)P_{\lambda\sigma} \tag{6}$$

$$\text{step 2:}\quad J'_Q = \sum_{P}(Q|P)^{-1}J_P \tag{7}$$

$$\text{step 3:}\quad J_{\mu'\nu'} = \sum_{Q}(\mu'\nu'|Q)J'_Q \tag{8}$$

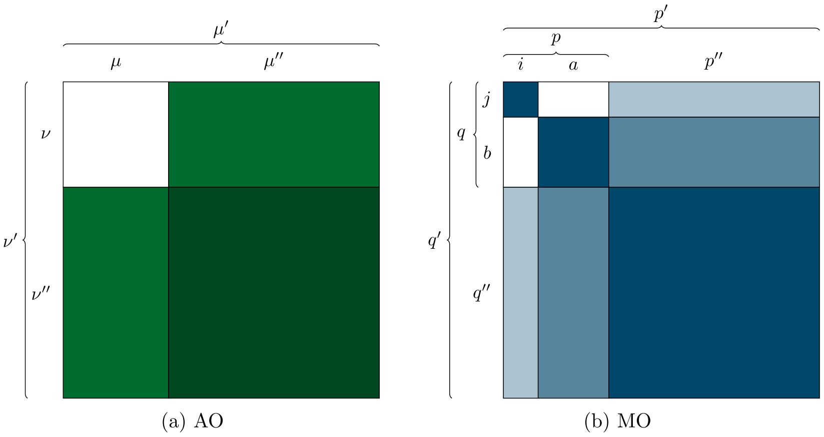


Figure 1. Representations of (a) the AO and (b) the MO spaces present in F12 theory covered by F12-type Fock matrix elements.

Several techniques allow the efficient evaluation of these formulas. For step 2, the well-known Coulomb fitting method of Mintmire and Dunlap[77] avoids the matrix inversion step for the two-center integrals by a direct evaluation of the Coulomb metric $(Q|P)x = J_P$, leading to $x = (Q|P)^{-1}J_P = J'_Q$, for which we employ a Cholesky decomposition of $(Q|P)$. The typically rate-determining steps 1 and 3 requiring 3c-2e integrals are accelerated by a J-engine[15,16] algorithm: First, the AO (step 1) and auxiliary (step 3) density(-like) matrices are transformed in a preprocessing computation into the Hermite basis. Subsequently, the resulting intermediates are contracted with the 3c-2e integrals to form the Coulomb potential in the Hermite basis, representing by far the most expensive step within the J-engine algorithm due to its asymptotic quadratic scaling. In the final postprocessing, the Coulomb potential is back-transformed into the AO (step 3) or auxiliary (step 1) basis, respectively.

Since some intermediate Hermite factors (i.e., all odd l quantum numbers) are not necessary for the representation of the auxiliary basis functions, these contributions can be omitted, leading to further efficiency gains (cf. section 2.1 of ref 18). Combining all of these aspects results in a highly efficient evaluation of the **J** matrix, especially when the massively parallel behavior of GPUs is utilized to compute the expensive Coulomb potential in the Hermite basis.

2.3. Seminumerical Exchange: sn-LinK.

The general integration scheme for seminumerical integral evaluation[39,41,78] results in the symmetrical decomposition of the 4c-2e integrals as

$$(\mu\sigma|\lambda\nu) \approx \frac{1}{2}[((\mu\sigma)^{\text{num}}|(\nu\lambda)^{\text{ana}}) + ((\mu\sigma)^{\text{ana}}|(\nu\lambda)^{\text{num}})] \equiv \frac{1}{2}\left[\sum_g w_g\chi_\mu(\mathbf{r}_g)\chi_\sigma(\mathbf{r}_g)\int \mathrm{d}\mathbf{r}\,\frac{\chi_\lambda(\mathbf{r})\chi_\nu(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|} + \sum_g w_g\int \mathrm{d}\mathbf{r}\,\frac{\chi_\mu(\mathbf{r})\chi_\sigma(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}\chi_\lambda(\mathbf{r}_g)\chi_\nu(\mathbf{r}_g)\right] \tag{9}$$

where Becke-type molecular integration grids[36−38] with grid points $\mathbf{r}_g$ and associated weights w_g are employed. Application of this ansatz to the AO representation of the approximation-free F12-type exchange matrix results in

$$K_{\mu'\nu'} = \sum_{\lambda\sigma} P_{\lambda\sigma}(\mu'\sigma|\lambda\nu') \approx \frac{1}{2}\left[\sum_g w_g \sum_{\lambda\sigma}\chi_{\mu'}(\mathbf{r}_g)\int \mathrm{d}\mathbf{r}\,\frac{\chi_\lambda(\mathbf{r})\chi_{\nu'}(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|}\chi_\sigma(\mathbf{r}_g)P_{\lambda\sigma} + \text{transpose}\right] \tag{10}$$

which is computed in three consecutive steps:

$$\text{step 1:}\quad F_{\lambda g} = \sum_\sigma \chi_\sigma(\mathbf{r}_g)P_{\lambda\sigma} \tag{11}$$

$$\text{step 2:}\quad G_{\nu' g} = \sum_\lambda w_g A_{\lambda\nu' g}F_{\lambda g} \tag{12}$$

$$\text{step 3:}\quad \bar{K}_{\mu'\nu'} = \sum_g \chi_{\mu'}(\mathbf{r}_g)G_{\nu' g} \tag{13}$$

where $A_{\lambda\nu' g}$ are mixed-basis 3c-1e integrals, given by

$$A_{\lambda\nu' g} = \int \frac{\chi_\lambda(\mathbf{r})\chi_{\nu'}(\mathbf{r})}{|\mathbf{r}_g - \mathbf{r}|}\,\mathrm{d}\mathbf{r} \tag{14}$$

Finally, the F12-type exchange matrix is obtained via the symmetrization

$$K_{\mu'\nu'} = \frac{1}{2}(\bar{K}_{\mu'\nu'} + \bar{K}_{\nu'\mu'}) \tag{15}$$

to take care of the transpose in eq 10. Steps 1 (eq 11) and 3 (eq 13) are implemented as dense matrix−matrix multiplications, whereas step 2 (eq 12) requires on-the-fly evaluation of the 3c-1e integrals (eq 14). The matrix−matrix multiplications in steps 1 and 3 utilize batch-local matrices with asymptotically constant size computed via BLAS-3 libraries to achieve the best performance, as described in detail in ref 41. However, the evaluation of the 3c-1e integrals in step 2 is generally the most computationally demanding part in the seminumerical integral evaluation, requiring an efficient integral screening procedure for optimal performance. Practical approaches to determine the significance of a 3c-1e integral $A_{\lambda\nu' g}$ are screening for the F12-type exchange energy $\epsilon^{E}_{\lambda\nu' g}$ and the final F12-type exchange matrix contributions $\epsilon^{K}_{\lambda\nu' g}$, given by

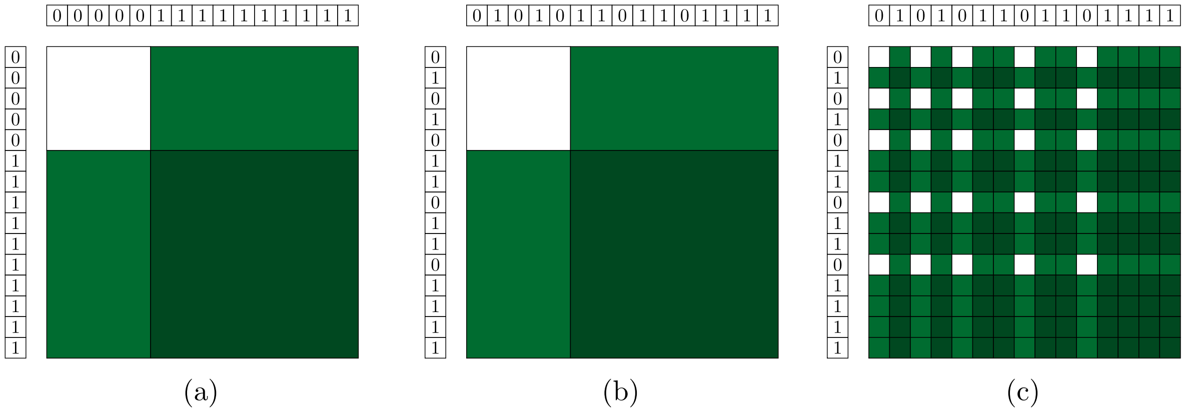


Figure 2. (a) Standard representation of the orbital spaces leading to separate blocks. (b) Construction of new shell-pair data. (c) Mixed pattern of the resulting orbital spaces.

$$\epsilon^{E}_{\lambda\nu' g} = \left| w_g \sum_{\mu\sigma} \chi_\mu(\mathbf{r}_g) P_{\mu\nu} A_{\lambda\nu' g} P_{\lambda\sigma} \chi_\sigma(\mathbf{r}_g) \right| = \left| w_g^{1/2} F_{\nu g} A_{\lambda\nu' g} w_g^{1/2} F_{\lambda g} \right| \tag{16}$$

$$\begin{aligned}\epsilon^{K}_{\lambda\nu' g} &= |w_g| \max\left(\sum_{\mu\sigma'} |\chi_\mu(\mathbf{r}_g)||P_{\mu\nu}||A_{\lambda\nu' g}||\chi_{\sigma'}(\mathbf{r}_g)|, \right.\\ &\qquad \left. \sum_{\mu'\sigma} |\chi_{\mu'}(\mathbf{r}_g)||A_{\lambda\nu' g}||P_{\lambda\sigma}||\chi_\sigma(\mathbf{r}_g)| \right)\\ &\le |w_g| \max(|F_{\nu g}|, |F_{\lambda g}|)|A_{\lambda\nu' g}| \sum_{\mu'} |\chi_{\mu'}(\mathbf{r}_g)|\end{aligned} \tag{17}$$

In accordance with ref 41, integrals are considered to be significant if either one or both of these values are above a given threshold, i.e.,

$$\epsilon^{E}_{\lambda\nu' g} \ge \vartheta_E \quad \text{or/and} \quad \epsilon^{K}_{\lambda\nu' g} \ge \vartheta_K \tag{18}$$

Details regarding the evaluation of the required integral bounds

$$\mathcal{V}_{\nu'\lambda} \le \max_{\mathbf{r}_g \in \mathbb{R}^3} \left(\int \frac{|\chi'_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})|}{|\mathbf{r}_g - \mathbf{r}|} \mathrm{d}\mathbf{r} \right) \tag{19}$$

are provided in ref 40.

2.4. Adoption of RI-J/sn-LinK for F12-Type Fock Matrices.

Our previous approximation-free reference implementation for the F12-type Fock matrix is based on the blockwise computation represented in Figure 2a, where elements of the different AO spaces are calculated via separate integral routines. The evaluation of the $F_{\mu''\nu}$ and $F_{\mu''\nu''}$ blocks requires additional implementational work, e.g., support for mixed shell-pairs.

In this present study, however, we avoid this extra effort by merging the AO basis and the CABS basis into one combined basis according to atom and angular momentum quantum numbers, represented by zeros and ones in Figure 2b. Capitalizing on this computationally inexpensive transformation from two separate basis sets to one combined basis set allows the direct use of standard integral routines as well as RI-J and sn-LinK. Back-transformation of the resulting F12-type Fock matrix (Figure 2c) to a blockwise representation is easily possible.

Since only the AO elements of the density matrix $P_{\lambda\sigma}$ are nonzero, a large number of irrelevant contributions are included within the combined basis set, which could in principle lead to substantial inefficiencies. However, all of those zero elements in $P_{\lambda'\sigma'}$ are excluded early on by the density-including integral screening techniques within both RI-J and sn-LinK, resulting in virtually no overhead in practice. In this way, our advanced RI-J[18] and sn-LinK[41] methods with all of their benefits (e.g., GPU acceleration) are directly applicable to the evaluation of F12-type Fock matrix elements.

3. COMPUTATIONAL DETAILS

All of the reported calculations were performed with our FermiONs++ program package,[79−82] where the grids[38] summarized in Table 2 and the cc-pVYZ-JKfit[12] (Y = D, T, Q, 5) RI-J basis sets were employed for sn-LinK and RI-J, respectively. As proposed in ref 38, SCF- and F12-type exchange matrices were computed using the smaller gX grid, whereas the final energy evaluation utilized a larger gX+2 grid (compressed in a shorthand multigrid gm[X+2/X] notation). As a reference we set our approximation-free code using the Obara−Saika recursion scheme[83] for the 4c-2e integrals. Throughout the following, the F12 correction refers to the explicitly correlated F12 energy correction to second-order Møller-Plesset perturbation theory in the 3*C variant,[51] where we employed Ten-no's fixed-amplitude ansatz[84] in combination with the extended Brillouin condition (EBC).[44] A fixed Slater-type geminal (STG) correlation factor[45,65] of the form $\hat{F}_{12} = \frac{1}{\gamma}[1 - \exp(-\gamma r_{12})]$ with $\gamma = 1.3$ was used together with

Table 2. Definition of the Employed Grids with Separation into Inner, Medium, and Outer Regions for the Example of the C Atom

grid	n_{rad}	n_{ang}(inner/medium/outer)	$n_{\text{tot,C}}$
g0	30	14/38/74	1654
g1	35	14/50/110	2586
g2	40	26/74/194	5056
g3	50	38/110/302	9564
g4	55	50/194/434	15526
g5	60	50/194/590	21330
g6	70	86/302/974	40838
g7	80	110/434/1454	68770

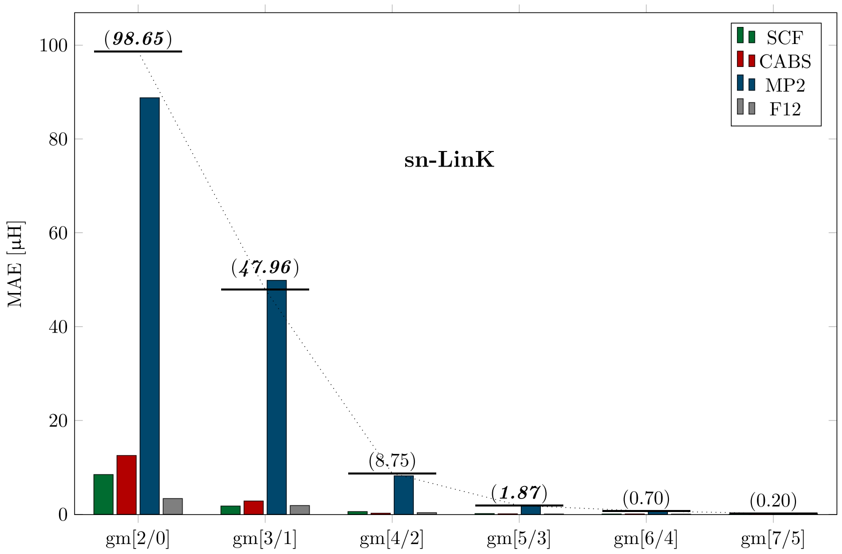


Figure 3. Visualization of HF, CABS singles, MP2, F12 correction, and total sn-LinK NCI MAEs (numbers in parentheses) in dependence on the grid size for the L7 test set, employing a cc-pVDZ-F12 AO basis and a cc-pVDZ-F12/OptRI+ CABS basis.

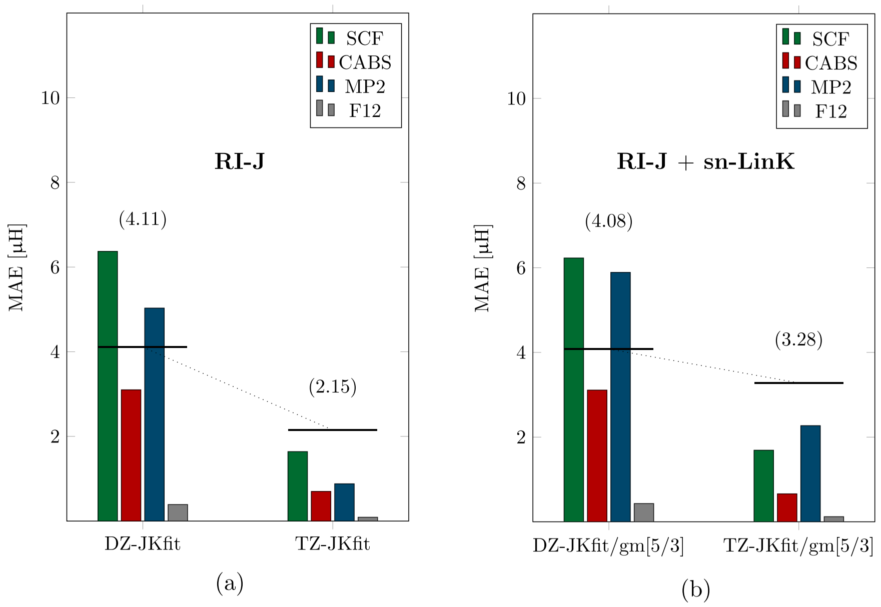


Figure 4. Visualization of L7 HF, CABS singles, MP2, F12 correction, and total NCI MAEs (numbers in parentheses) for (a) RI-J and (b) RI-J/sn-LinK, employing a cc-pVDZ-F12 AO basis set, a cc-pVDZ-F12/OptRI+ CABS basis set, and two different cc-pVYZ-JKfit (Y = D, T) RI-J basis sets.

the cc-pVXZ-F12[85−87] basis set family and corresponding CABS cc-pVXZ-F12/OptRI+[88] and density fitting (DF) cc-pVXZ-F12/MP2fit[89] basis sets (X = D, T, Q).

For optimal performance, the integral kernels were compiled with the Intel Compiler 19.1.0[90] (flags: -Ofast -march=cascadelake) and the OpenCL GPU kernels with ROCm-3.8.0[91] (flags: -O3 -cl-mad-enable -cl-finite-math-only -cl-no-signed-zeros). The performance was assessed on two Intel Xeon Silver 4216 processors (32 cores/64 threads; 2.1 GHz) and 4 AMD Radeon VII cards to ensure a fair comparison between CPU and GPU (roughly equal acquisition cost of hardware). For the conventionel 4c-2e and RI-J 3c-2e integrals, we set a threshold of 10^{-13}, and for sn-LinK we chose $\vartheta_K = 10^{-10}$ and $\vartheta_E = 10^{-13}$, employing mixed single- and double-precision 3c-1e integral evaluation[78] on CPUs. The SCF was converged to within 10^{-7} of the DIIS commutator norm ($\|\mathbf{FPS} - \mathbf{SPF}\|$),[92,93] and interaction energies were counterpoise-corrected[94] to take the basis set superposition error (BSSE) into account.

4. RESULTS

In the following, we summarize the most essential and representative findings of a benchmark study on the accuracy and efficiency of the F12-type Fock matrix element evaluation using varying sn-LinK and RI-J settings for prominent test sets.[95−98] Since the results are qualitatively identical among the test sets, here we focus on the L7 non-covalent interaction (NCI) energies[98] and refer the reader to the Supporting Information (SI) for more insights and detailed data on the remaining systems (the S22, S66, and ISO34 test sets), including deviations in isomerization and absolute energies as well as conventional RI-JK results.

Table 3. RI-J/sn-LinK (cc-pVYZ-JKt/gm[2/0]; Y = X) Speedups on CPUs (S_{CPU}) and GPUs (S_{GPU}) for the Full F12-Type Fock Build for Each Member of the L7 Test Set (cc-pVXZ-F12; X = D, T, Q); Additional RI-JK (cc-pVDZ-JKfit) Results ($S_{\mathrm{CPU}}^{\mathrm{RI\text{-}JK}}$) Are Given for a cc-pVDZ-F12 Basis

	cc-pVDZ-F12					cc-pVTZ-F12				cc-pVQZ-F12		
L7 structure[a]	N_{bas}	N_{CABS}	$S_{\mathrm{CPU}}^{\mathrm{RI\text{-}JK}}$	S_{CPU}	S_{GPU}	N_{bas}	N_{CABS}	S_{CPU}	S_{GPU}	N_{bas}	N_{CABS}	S_{CPU}[b]
L1	1836	5288	19	209	528	3676	7804	639	1627	6996	9700	1551
L2	1191	3426	29	251	469	2331	4311	733	1550	4281	5229	1488
L3	2255	6486	41	517	1340	4405	8044	1423	3403	8065	9733	2761
L4	2588	7444	47	584	1635	5058	9267	1575	4155	9268	11220	2710
L5	1818	5232	31	316	742	3588	6999	976	2017	6678	8574	2024
L6	1752	5044	48	446	1072	3432	6384	1308	2884	6312	7752	2548
L7	1396	4016	33	331	682	2736	5106	939	1880	5036	6204	1918

[a] L7 test set structures: L1, octadecane dimer; L2, guanine trimer; L3, circumcoronene−adenine dimer; L4, circumcoronene−guanine−cytosine trimer; L5, phenylalanine residues trimer; L6, coronene dimer; L7, guanine−cytosine−guanine−cytosine tetramer. [b] Reference extrapolated from double- and triple-ζ F12 timings.

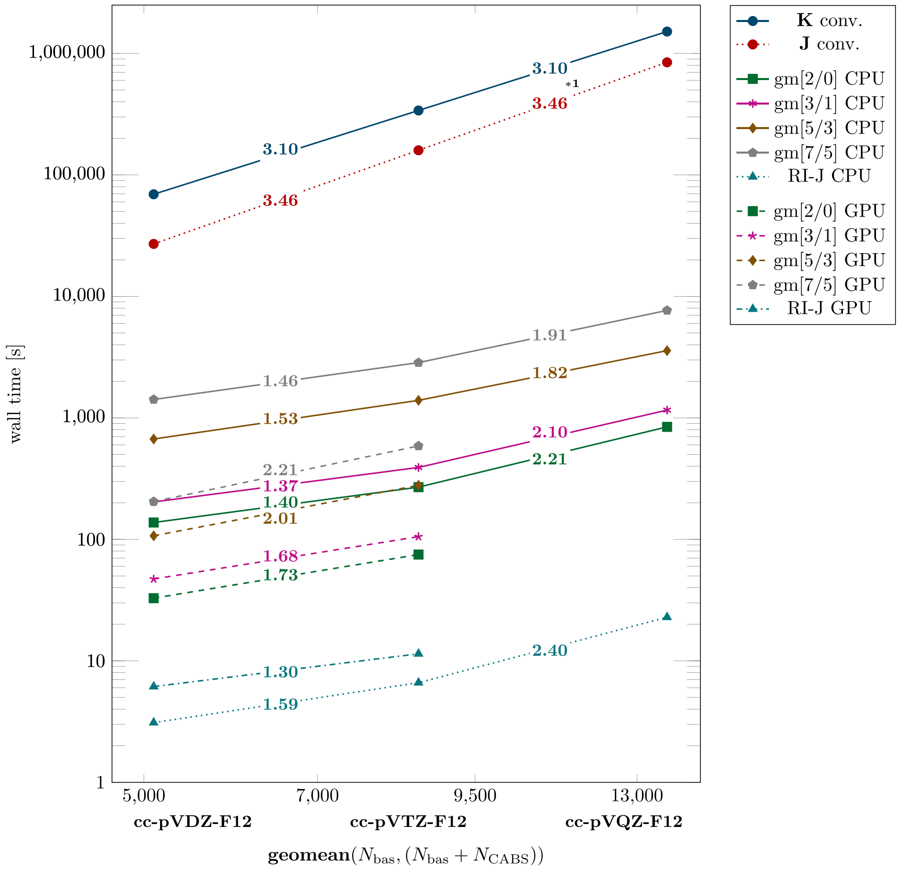


Figure 5. Log−log plot of the wall times for the L7 circumcoronene−guanine−cytosine trimer $J_{\mu'\nu'}$ and $K_{\mu'\nu'}$ matrix element evaluation employing an approximation-free reference code, RI-J, and sn-LinK for a cc-pVXZ-F12 AO basis set and a cc-pVXZ-F12/OptRI+ CABS basis set (X = D, T, Q) on CPUs and GPUs. sn-LinK and RI-J timings are reported for various gm[X+2/X] multigrid sizes and a cc-pVYZ-JKfit RI-J basis set (Y = X). Scaling coefficients are given within the lines. $*^1$ extrapolated from double- and triple-ζ timings.

4.1. Accuracy.

Starting with the main bottleneck of computing the F12-type Fock matrix elements, the evaluation of the exchange-type matrix $K_{\mu'\nu'}$, we explore the behavior of sn-LinK for different gm[X+2/X] multigrids for the cc-pVDZ-F12 AO basis set, with qualitatively identical results for triple- and quadruple-ζ F12 calculations given in the SI. We investigate the accuracy of HF, CABS singles, MP2, F12 correction, and total energies for counterpoise-corrected NCI of the L7 test set compared with an approximation-free reference code visualized in Figure 3. As expected, higher accuracy is achieved with increasing grid size, with total mean absolute errors (MAEs) ranging from 100 to 0.2 μH from the smallest to the largest multigrid, which is always substantially below the MP2 method error (>10 mH). With regard to the individual error contributions, the MP2 deviations dominate the total MAE, matching observations in ref 41. This error is only caused by a slight imperfection in the self-consistently converged Fock and density matrices due to the finite grid error in the SCF.

Table 4. Absolute Timings for the F12-Type Fock Build Using Our Original Implementation, RI-JK (cc-pVDZ-JKfit), RI-J/sn-LinK (cc-pVDZ-JKt/gm[2/0]), and the Total MP2-F12 Correlation Calculation Excluding the F12-Type Fock Build (t_{corr}) for Each Member of the L7 Test Set (cc-pVDZ-F12) with the Corresponding Ratios of Fock Build Time to Remaining Correlation Time

L7 structure[a]	$t^{\text{ref}}_{\text{F12-Fock}}$ [s]	$t^{\text{RI-JK}}_{\text{F12-Fock}}$ [s]	$t^{\text{RI-J/sn-LinK}}_{\text{F12-Fock}}$ [s]	t_{corr} [s]	$R^{\text{ref}}_{\text{corr}}$ [%]	$R^{\text{RI-JK}}_{\text{corr}}$ [%]	$R^{\text{RI-J/sn-LinK}}_{\text{corr}}$ [%]
L1	13894	715	66	12854	108.1	5.6	0.5
L2	6584	225	26	2606	252.6	8.6	1.0
L3	61013	1459	118	29729	205.2	4.9	0.4
L4	96379	2044	165	57095	168.8	3.5	0.3
L5	22381	701	71	13568	164.9	5.2	0.5
L6	34536	725	77	9930	347.8	7.3	0.8
L7	12412	372	40	4925	252.0	7.6	0.8

[a] L7 test set structures: L1, octadecane dimer; L2, guanine trimer; L3, circumcoronene−adenine dimer; L4, circumcoronene−guanine−cytosine trimer; L5, phenylalanine residues trimer; L6, coronene dimer; L7, guanine−cytosine−guanine−cytosine tetramer.

To investigate the accuracy of RI-J for the L7 system, we utilized the cc-pVYZ-JKfit (Y = D, T) RI-J basis set family together with an AO cc-pVDZ-F12 basis set for both the SCF cycle and the F12-type Fock matrix evaluation. Following the same pattern as for sn-LinK, the MAEs show excellent accuracy with negligible deviations of roughly 4 μH even for a double-ζ RI-J basis set (Figure 4a). With a triple-ζ auxiliary basis set, even more precise values are possible, reducing the errors by a factor of approximately 2 compared with the double-ζ evaluation. For the combination of sn-LinK and RI-J, we employed the gm[5/3] multigrid because of the comparable errors of the two methods (Figure 4b). The actual RI-J/sn-LinK combination experiences some form of error cancellation with a MAE close to the RI-J errors and less than the sum of the individual-method MAEs. In practice, the errors of only 3−4 μH for RI-J/sn-LinK relative to the exact analytical treatment are virtually irrelevant.

4.2. Peformance Comparison. To demonstrate the full potential of our sn-LinK and RI-J methods, we focus on timings for the largest and consequently computationally most expensive member of the L7 test set (L4 in Table 3): a circumcoronene−guanine−cytosine trimer with 102 atoms. Figure 5 compares the performance of sn-LinK and RI-J for increasing AO and CABS basis sets employing the corresponding (DZ/TZ/QZ) RI-J basis sets and a variety of multigrids with the exact analytical treatment for increasing AO and CABS basis sets on CPUs and GPUs.

Because of the decreasing N_{bas} to N_{CABS} ratio for larger cardinal numbers X, we decided to plot the geometric mean $[N_{\text{bas}}(N_{\text{bas}} + N_{\text{CABS}})]^{1/2}$ against the required time to ensure a fair comparison between the different AO basis set sizes and the formal $N_{\text{bas+CABS}}{}^2 N_{\text{bas}}{}^2$ scaling of the reference. Quadruple-zeta F12 reference values were extrapolated from double- and triple-ζ results. GPU calculations for quadruple-ζ F12 calculations are currently not feasible due to numerical problems regarding the GPU execution of some integrals for h (l = 5) functions.

The observed time complexity with respect to the basis set size roughly matches the theoretical $O(N_{\text{bas}}{}^4)$ or $O(N_{\text{bas}}{}^2)$ scaling for the analytical or sn-LinK/RI-J treatment, respectively. The observed variations around these theoretical values, i.e, $O(N_{\text{bas}}{}^{3.10})$ to $O(N_{\text{bas}}{}^{3.46})$ instead of $O(N_{\text{bas}}{}^4)$ for the exact treatment and $O(N_{\text{bas}}{}^{1.30})$ to $O(N_{\text{bas}}{}^{2.40})$ instead of $O(N_{\text{bas}}{}^2)$ for RI-J/sn-LinK, are expected given that the average cost of evaluating one integral and the effectiveness of integral screening methods vary substantially among the basis sets because of, e.g., different l quantum numbers or the addition of diffuse functions. Moreover, the amount of parallel workload, which significantly affects GPU performance, also increases with larger basis sets. The choice of the multigrid contributes approximately as a constant prefactor (largely independent of the basis set). As a result of this reduced basis set scaling in combination with the large size of the CABS basis sets, tremendous speedups are achieved. For example, using only CPUs, sn-LinK with a gm[2/0] multigrid provides speedups ranging from 500× to 1800× (DZ-F12 to QZ-F12, respectively), and RI-J gives speedups ranging from 8700× to 37000×. Employing GPUs improves the performance of sn-LinK even further, with accelerations ranging from 3200× to 4500× (DZ-F12 to TZ-F12, respectively). In contrast, RI-J does not benefit from GPU acceleration due to a lack of parallel workload in this case (cf. discussion in ref 18). However, this is not a concern in practice because of the comparatively low cost of the RI-J part regardless.

To further illustrate the profound efficiency improvement of sn-LinK and RI-J, in Table 3 we summarize the total speedups for the full F12-type Fock build (**J**, **K**, core Hamiltonian, and ordering algorithm) for each member of the L7 benchmark set for a gm[2/0] integration grid (results for larger multigrids are given in the SI) alongside RI-JK double-ζ results (see details of our RI-K implementation in the SI).

While RI-JK yields good accelerations with, on average, 35× faster computations compared with our conventional implementation, double- and triple-ζ results were not feasible because of the vast memory demand of the required three-center integrals in RI-K. Generally, the steep $O(N_{\text{occ}}N_{\text{aux}}N_{\text{bas}}{}^2)$ scaling of RI-K makes it unfavorable for medium- to large-sized structures compared with the alternative of sn-LinK.

Our RI-J/sn-LinK methods proposed in this work result in excellent speedups, surpassing RI-JK with on average 379× faster computation for a double-ζ basis, with the performance gains over all basis sets (DZ−QZ) ranging between 209× and 4155×. For example, the total runtime for one triple-ζ F12-type Fock build for the circumcoronene−guanine−cytosine complex (L4) is reduced by a factor of 4155 from ~6 days (analytically) to only ~2 min (RI-J/sn-LinK).

To illustrate the importance of a fast F12-type Fock matrix evaluation, in Table 4 we compare timings for our conventional reference, RI-JK, and RI-J/sn-LinK for the L7 test set

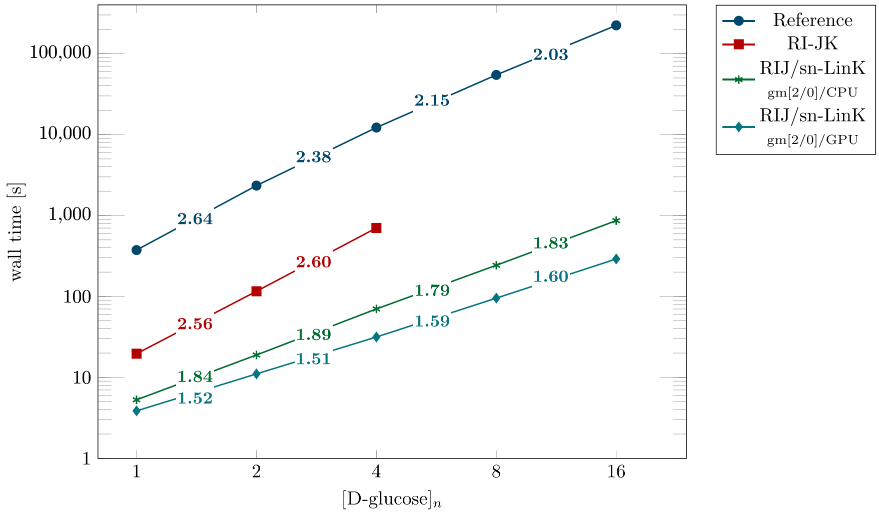


Figure 6. Log−log plot of the wall times against the length of increasing amylose chains for the evaluation of the corresponding F12-type Fock matrix elements, employing cc-pVDZ-F12 AO and cc-pVDZ-F12/OptRI+ CABS basis sets together with our reference, RI-JK (cc-pVDZ-JKfit), and RI-J/sn-LinK (cc-pVDZ-JKt/gm[2/0]) on CPUs and GPUs. Scaling coefficients are given within the lines.

(cc-pVDZ-F12) with our RI-MP2-F12 implementation excluding the F12-type Fock build. Our reference represents by far the most expensive step in the total correlation calculations for all L7 members, requiring on average 214.2% more time than the remaining terms, including the standard MP2 correction. For RI-JK, this ratio continues to be significant at 6.1%, whereas RI-J/sn-LinK lowers this ratio to a desirable 0.6%.

Finally, to demonstrate the behavior for a fixed basis set and increasing molecule sizes, we examined systematically increasing amylose chain lengths[99] using our reference code, RI-JK, RI-J/sn-LinK (CPU), and RI-J/sn-LinK (GPU) in combination with a double-ζ F12 basis. The results are shown in Figure 6. RI-JK decreases the required time by a factor of roughly 20, where memory limitations restrict the computation to a chain length of four D-glucose subunits, e.g., the 3c-2e RI-K integrals for eight subunits require 538 GB of storage, which increases to 4.21 TB for 16 subunits, matching its $\mathcal{O}(M^{3.0})$ memory scaling with the molecule size (M). RI-J/sn-LinK leads once again to excellent speedups with scaling coefficients between $\mathcal{O}({N_{\text{glucose}}}^{2})$ and $\mathcal{O}({N_{\text{glucose}}}^{1.5})$, matching the asymptotic $\mathcal{O}(M^2)$ and $\mathcal{O}(M)$ scalings of RI-J and sn-LinK with the molecule size for systems with local electronic structure. For the largest chain length (16 D-glucose subunits), we observed 768× faster evaluation using our GPUs, reducing the required time from ~2.6 days to less than 5 min.

5. CONCLUSION

We employed our recently published highly efficient RI-J[18] and sn-LinK[41] methods to overcome the two major bottlenecks of the **J** and **K** matrix computation in the evaluation of F12-type Fock matrices. We tested the accuracy of the methods for multiple benchmark sets covering non-covalent interactions and isomerization energies (also see the Supporting Information). Even for the smallest grids and RI-J basis sets, the mean absolute errors are always below 0.43 mH and are easily reducible to below 5 μH for slightly larger integration grids.

Moreover, since both methods lower the formal scaling with respect to the basis set size from $\mathcal{O}({N_{\text{bas}}}^{4})$ to $\mathcal{O}({N_{\text{bas}}}^{2})$, impressive performance improvements of up to 37000× for the direct (Coulomb) contribution and 1800× for the exchange contribution were achieved, and the latter could be improved even further to over 4500× faster execution when GPU acceleration was employed. In total, RI-J/sn-LinK combines remarkable efficiency with high accuracy for evaluation of the F12-type Fock matrix, leading to tremendous time savings with over 3 orders of magnitude faster computations. We therefore expect wide applicability in F12 theories.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.2c00215.

Detailed individual and total NCI, isomerization, and total energy data for all test sets (L7, S22, S66, ISO34) and methods employing a cc-pVXZ-F12 (X = D, T, Q) AO basis and additional timings for the L7 structures, and an outline of our RI-K implementation (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Christian Ochsenfeld − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany; Max Planck Institute for Solid State Research, D-70569 Stuttgart, Germany;* orcid.org/0000-0002-4189-6558; Email: christian.ochsenfeld@uni-muenchen.de

Authors

Lars Urban − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany; Max Planck Institute for Solid State Research, D-70569 Stuttgart, Germany;* orcid.org/0000-0001-9891-3577

Henryk Laqua − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.2c00215

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank F. H. Bangerter (LMU Munich) and V. Drontschenko (LMU Munich) for helpful discussions and J. Kussmann (LMU Munich) for providing a development version of the FermiONs++ program package.[79−82] Financial support was provided by the Excellence Cluster EXC2111-390814868 at the Munich Center for Quantum Science and Technology (MCQST) by the Deutsche Forschungsgemeinschaft (DFG). C.O acknowledges, in addition, financial support as Max Planck Fellow of MPI-FKF Stuttgart.

REFERENCES

(1) Kussmann, J.; Beer, M.; Ochsenfeld, C. Linear-scaling self-consistent field methods for large molecules. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2013**, *3*, 614−636.

(2) Hartree, D. R. The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part I Theory and Methods. *Math. Proc. Camb. Philos. Soc.* **1928**, *24*, 89−110.

(3) Fock, V. Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems. *Z. Phys.* **1930**, *61*, 126−148.

(4) Roothaan, C. C. J. New developments in molecular orbital theory. *Rev. Mod. Phys.* **1951**, *23*, 69−89.

(5) Hall, G. G. The molecular orbital theory of chemical valency VIII. A method of calculating ionization potentials. *Proc. R. Soc. London, Ser. A* **1951**, *205*, 541−552.

(6) Kohn, W.; Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **1965**, *140*, A1133−A1138.

(7) White, C. A.; Johnson, B. G.; Gill, P. M.; Head-Gordon, M. The continuous fast multipole method. *Chem. Phys. Lett.* **1994**, *230*, 8−16.

(8) Strain, M. C.; Scuseria, G. E.; Frisch, M. J. Achieving linear scaling for the electronic quantum coulomb problem. *Science* **1996**, *271*, 51−53.

(9) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. Linear and sublinear scaling formation of Hartree-Fock-type exchange matrices. *J. Chem. Phys.* **1998**, *109*, 1663−1669.

(10) Whitten, J. L. Coulombic potential energy integrals and approximations. *J. Chem. Phys.* **1973**, *58*, 4496−4501.

(11) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. On some approximations in applications of Xα theory. *J. Chem. Phys.* **1979**, *71*, 3396−3402.

(12) Weigend, F. A fully direct RI-HF algorithm: Implementation, optimized auxiliary basis sets, demonstration of accuracy and efficiency. *Phys. Chem. Chem. Phys.* **2002**, *4*, 4285−4291.

(13) Eichkorn, K.; Treutler, O.; Öhm, H.; Häser, M.; Ahlrichs, R. Auxiliary basis sets to approximate Coulomb potentials. *Chem. Phys. Lett.* **1995**, *240*, 283−290.

(14) Eichkorn, K.; Weigend, F.; Treutler, O.; Ahlrichs, R. Auxiliary basis sets for main row atoms and transition metals and their use to approximate Coulomb potentials. *Theor. Chem. Acc.* **1997**, *97*, 119−124.

(15) Reza Ahmadi, G.; Almlöf, J. The Coulomb operator in a Gaussian product basis. *Chem. Phys. Lett.* **1995**, *246*, 364−370.

(16) White, C. A.; Head-Gordon, M. A J matrix engine for density functional theory calculations. *J. Chem. Phys.* **1996**, *104*, 2620−2629.

(17) Neese, F. An improvement of the resolution of the identity approximation for the formation of the Coulomb matrix. *J. Comput. Chem.* **2003**, *24*, 1740−1747.

(18) Kussmann, J.; Laqua, H.; Ochsenfeld, C. Highly Efficient Resolution-of-Identity Density Functional Theory Calculations on Central and Graphics Processing Units. *J. Chem. Theory Comput.* **2021**, *17*, 1512−1521.

(19) Früchtl, H. A.; Kendall, R. A.; Harrison, R. J.; Dyall, K. G. An Implementation of RI-SCF on Parallel Computers. *Int. J. Quantum Chem.* **1997**, *64*, 63−69.

(20) Hamel, S.; Casida, M. E.; Salahub, D. R. Assessment of the quality of orbital energies in resolution-of-the-identity Hartree-Fock calculations using deMon auxiliary basis sets. *J. Chem. Phys.* **2001**, *114*, 7342−7350.

(21) Weigend, F. Hartree−Fock exchange fitting basis sets for H to Rn. *J. Comput. Chem.* **2008**, *29*, 167−175.

(22) Polly, R.; Werner, H.-J.; Manby, F. R.; Knowles, P. J. Fast Hartree-Fock theory using local density fitting approximations. *Mol. Phys.* **2004**, *102*, 2311−2321.

(23) Köppl, C.; Werner, H.-J. Parallel and low-order scaling implementation of Hartree-Fock exchange using local density fitting. *J. Chem. Theory Comput.* **2016**, *12*, 3122−3134.

(24) Friesner, R. A. Solution of self-consistent field electronic structure equations by a pseudospectral method. *Chem. Phys. Lett.* **1985**, *116*, 39−43.

(25) Friesner, R. A. Solution of the Hartree-Fock equations by a pseudospectral method: Application to diatomic molecules. *J. Chem. Phys.* **1986**, *85*, 1462−1468.

(26) Friesner, R. A. Solution of the Hartree-Fock equations for polyatomic molecules by a pseudospectral method. *J. Chem. Phys.* **1987**, *86*, 3522−3531.

(27) Ringnalda, M. N.; Belhadj, M.; Friesner, R. A. Pseudospectral Hartree-Fock theory: Applications and algorithmic improvements. *J. Chem. Phys.* **1990**, *93*, 3397−3407.

(28) Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. Efficient, approximate and parallel Hartree-Fock and hybrid DFT calculations. A 'chain-of-spheres' algorithm for the Hartree-Fock exchange. *Chem. Phys.* **2009**, *356*, 98−109.

(29) Plessow, P.; Weigend, F. Seminumerical calculation of the Hartree-Fock exchange matrix: Application to two-component procedures and efficient evaluation of local hybrid density functionals. *J. Comput. Chem.* **2012**, *33*, 810−816.

(30) Bahmann, H.; Kaupp, M. Efficient self-consistent implementation of local hybrid functionals. *J. Chem. Theory Comput* **2015**, *11*, 1540−1548.

(31) Maier, T. M.; Bahmann, H.; Kaupp, M. Efficient Seminumerical Implementation of Global and Local Hybrid Functionals for Time-Dependent Density Functional Theory. *J. Chem. Theory Comput.* **2015**, *11*, 4226−4237.

(32) Klawohn, S.; Bahmann, H.; Kaupp, M. Implementation of Molecular Gradients for Local Hybrid Density Functionals Using Seminumerical Integration Techniques. *J. Chem. Theory Comput.* **2016**, *12*, 4254−4262.

(33) Liu, F.; Kong, J. Efficient Computation of Exchange Energy Density with Gaussian Basis Functions. *J. Chem. Theory Comput.* **2017**, *13*, 2571−2580.

(34) Holzer, C. An improved seminumerical Coulomb and exchange algorithm for properties and excited states in modern density functional theory. *J. Chem. Phys.* **2020**, *153*, 184115.

(35) Helmich-Paris, B.; de Souza, B.; Neese, F.; Izsák, R. An improved chain of spheres for exchange algorithm. *J. Chem. Phys.* **2021**, *155*, 104109.

(36) Becke, A. D. A multicenter numerical integration scheme for polyatomic molecules. *J. Chem. Phys.* **1988**, *88*, 2547−2553.

(37) Treutler, O.; Ahlrichs, R. Efficient molecular numerical integration schemes. *J. Chem. Phys.* **1995**, *102*, 346−354.

(38) Laqua, H.; Kussmann, J.; Ochsenfeld, C. An improved molecular partitioning scheme for numerical quadratures in density functional theory. *J. Chem. Phys.* **2018**, *149*, 204111.

(39) Laqua, H.; Kussmann, J.; Ochsenfeld, C. Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals. *J. Chem. Theory Comput.* **2018**, *14*, 3451−3458.

(40) Thompson, T. H.; Ochsenfeld, C. Integral partition bounds for fast and effective screening of general one-, two-, and many-electron integrals. *J. Chem. Phys.* **2019**, *150*, 044101.

(41) Laqua, H.; Thompson, T. H.; Kussmann, J.; Ochsenfeld, C. Highly Efficient, Linear-Scaling Seminumerical Exact-Exchange Method for Graphic Processing Units. *J. Chem. Theory Comput.* **2020**, *16*, 1456−1468.

(42) Kutzelnigg, W. r_{12}-Dependent terms in the wave function as closed sums of partial wave amplitudes for large *l*. *Theor. Chim. Acta.* **1985**, *68*, 445−469.

(43) Klopper, W.; Kutzelnigg, W. Møller-plesset calculations taking care of the correlation CUSP. *Chem. Phys. Lett.* **1987**, *134*, 17−22.

(44) Kutzelnigg, W.; Klopper, W. Wave functions with terms linear in the interelectronic coordinates to take care of the correlation cusp. I. General theory. *J. Chem. Phys.* **1991**, *94*, 1985−2001.

(45) Ten-no, S. Initiation of explicitly correlated Slater-type geminal theory. *Chem. Phys. Lett.* **2004**, *398*, 56−61.

(46) Valeev, E. F. Improving on the resolution of the identity in linear R12 ab initio theories. *Chem. Phys. Lett.* **2004**, *395*, 190−195.

(47) Kato, T. On the eigenfunctions of many-particle systems in quantum mechanics. *Commun. Pure Appl. Math.* **1957**, *10*, 151−177.

(48) Noga, J.; Šimunek, J. On the one-particle basis set relaxation in R12 based theories. *Chem. Phys.* **2009**, *356*, 1−6.

(49) Bachorz, R. A.; Bischoff, F. A.; Glöß, A.; Hättig, C.; Höfener, S.; Klopper, W.; Tew, D. P. The MP2-F12 Method in the Turbomole Program Package. *J. Comput. Chem.* **2011**, *32*, 2492−2513.

(50) Adler, T. B.; Knizia, G.; Werner, H.-J. A simple and efficient CCSD(T)-F12 approximation. *J. Chem. Phys.* **2007**, *127*, 221106.

(51) Werner, H.-J.; Adler, T. B.; Manby, F. R. General orbital invariant MP2-F12 theory. *J. Chem. Phys.* **2007**, *126*, 164102.

(52) Werner, H.-J. Eliminating the domain error in local explicitly correlated second-order Møller−Plesset perturbation theory. *J. Chem. Phys.* **2008**, *129*, 101103.

(53) Höfener, S.; Klopper, W. Analytical nuclear gradients of the explicitly correlated Møller−Plesset second-order energy. *Mol. Phys.* **2010**, *108*, 1783−1796.

(54) Ma, Q.; Werner, H.-J. Scalable Electron Correlation Methods. 2. Parallel PNO-LMP2-F12 with Near Linear Scaling in the Molecular Size. *J. Chem. Theory Comput.* **2015**, *11*, 5291−5304.

(55) Wang, Y. M.; Hättig, C.; Reine, S.; Valeev, E.; Kjærgaard, T.; Kristensen, K. Explicitly correlated second-order Møller-Plesset perturbation theory in a Divide-Expand-Consolidate (DEC) context. *J. Chem. Phys.* **2016**, *144*, 204112.

(56) Győrffy, W.; Knizia, G.; Werner, H.-J. Analytical energy gradients for explicitly correlated wave functions. I. Explicitly correlated second-order Møller-Plesset perturbation theory. *J. Chem. Phys.* **2017**, *147*, 214101.

(57) Urban, L.; Thompson, T. H.; Ochsenfeld, C. A scaled explicitly correlated F12 correction to second-order Møller-Plesset perturbation theory. *J. Chem. Phys.* **2021**, *154*, 044101.

(58) Valeev, E. F. Coupled-cluster methods with perturbative inclusion of explicitly correlated terms: a preliminary investigation. *Phys. Chem. Chem. Phys.* **2008**, *10*, 106−113.

(59) Noga, J.; Kedžuch, S.; Šimunek, J.; Ten-no, S. Explicitly correlated coupled cluster F12 theory with single and double excitations. *J. Chem. Phys.* **2008**, *128*, 174103.

(60) Torheyden, M.; Valeev, E. F. Variational formulation of perturbative explicitly-correlated coupled-cluster methods. *Phys. Chem. Chem. Phys.* **2008**, *10*, 3410−3420.

(61) Valeev, E. F.; Crawford, T. D. Simple coupled-cluster singles and doubles method with perturbative inclusion of triples and explicitly correlated geminals: The $\text{CCSD(T)}_{\overline{\text{R12}}}$ Model. *J. Chem. Phys.* **2008**, *128*, 244113.

(62) Hättig, C.; Tew, D. P.; Köhn, A. Communications: Accurate and efficient approximations to explicitly correlated coupled-cluster singles and doubles, CCSD-F12. *J. Chem. Phys.* **2010**, *132*, 231102.

(63) Schmitz, G.; Hättig, C.; Tew, D. P. Explicitly correlated PNO-MP2 and PNO-CCSD and their application to the S66 set and large molecular systems. *Phys. Chem. Chem. Phys.* **2014**, *16*, 22167−22178.

(64) Győrffy, W.; Werner, H.-J. Analytical energy gradients for explicitly correlated wave functions. II. Explicitly correlated coupled cluster singles and doubles with perturbative triples corrections: CCSD(T)-F12. *J. Chem. Phys.* **2018**, *148*, 114104.

(65) Ten-no, S. A simple F12 geminal correction in multi-reference perturbation theory. *Chem. Phys. Lett.* **2007**, *447*, 175−179.

(66) Shiozaki, T.; Werner, H.-J. Communication: Second-order multireference perturbation theory with explicit correlation: CASPT2-F12. *J. Chem. Phys.* **2010**, *133*, 141103.

(67) Shiozaki, T.; Knizia, G.; Werner, H.-J. Explicitly correlated multireference configuration interaction: MRCI-F12. *J. Chem. Phys.* **2011**, *134*, 034113.

(68) Shiozaki, T.; Werner, H.-J. Multireference explicitly correlated F12 theories. *Mol. Phys.* **2013**, *111*, 607−630.

(69) Haunschild, R.; Mao, S.; Mukherjee, D.; Klopper, W. A universal explicit electron correlation correction applied to Mukherjee's multi-reference perturbation theory. *Chem. Phys. Lett.* **2012**, *531*, 247−251.

(70) Liu, W.; Hanauer, M.; Köhn, A. Explicitly correlated internally contracted multireference coupled-cluster singles and doubles theory: ic-MRCCSD(F12*). *Chem. Phys. Lett.* **2013**, *565*, 122−127.

(71) Roskop, L. B.; Kong, L.; Valeev, E. F.; Gordon, M. S.; Windus, T. L. Assessment of Perturbative Explicitly Correlated Methods for Prototypes of Multiconfiguration Electronic Structure. *J. Chem. Theory Comput.* **2014**, *10*, 90−101.

(72) Guo, Y.; Sivalingam, K.; Valeev, E. F.; Neese, F. Explicitly correlated N-electron valence state perturbation theory (NEVPT2-F12). *J. Chem. Phys.* **2017**, *147*, 064110.

(73) Hehn, A.-S.; Klopper, W. Communication: Explicitly-correlated second-order correction to the correlation energy in the random-phase approximation. *J. Chem. Phys.* **2013**, *138*, 181104.

(74) Sylvetsky, N.; Peterson, K. A.; Karton, A.; Martin, J. M. L. Toward a W4-F12 approach: Can explicitly correlated and orbital-based ab initio CCSD(T) limits be reconciled? *J. Chem. Phys.* **2016**, *144*, 214101.

(75) Kodrycka, M.; Holzer, C.; Klopper, W.; Patkowski, K. Explicitly Correlated Dispersion and Exchange Dispersion Energies in Symmetry-Adapted Perturbation Theory. *J. Chem. Theory Comput.* **2019**, *15*, 5965−5986.

(76) Liakos, D. G.; Izsák, R.; Valeev, E. F.; Neese, F. What is the most efficient way to reach the canonical MP2 basis set limit? *Mol. Phys.* **2013**, *111*, 2653−2662.

(77) Mintmire, J. W.; Dunlap, B. I. Fitting the Coulomb potential variationally in linear-combination-of- atomic-orbitals density-functional calculations. *Phys. Rev. A* **1982**, *25*, 88−95.

(78) Laqua, H.; Kussmann, J.; Ochsenfeld, C. Accelerating seminumerical Fock-exchange calculations using mixed single- and double-precision arithmethic. *J. Chem. Phys.* **2021**, *154*, 214116.

(79) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(80) 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.

(81) Kussmann, J.; Ochsenfeld, C. Employing OpenCL to Accelerate Ab Initio Calculations on Graphics Processing Units. *J. Chem. Theory Comput.* **2017**, *13*, 2712−2716.

(82) Kussmann, J.; Ochsenfeld, C. Hybrid CPU/GPU Integral Engine for Strong-Scaling Ab Initio Methods. *J. Chem. Theory Comput.* **2017**, *13*, 3153−3159.

(83) Obara, S.; Saika, A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1986**, *84*, 3963−3974.

(84) Ten-no, S. Explicitly correlated second order perturbation theory: Introduction of a rational generator and numerical quadratures. *J. Chem. Phys.* **2004**, *121*, 117−129.

(85) Peterson, K. A.; Adler, T. B.; Werner, H.-J. Systematically convergent basis sets for explicitly correlated wavefunctions: The atoms H, He, B-Ne, and Al-Ar. *J. Chem. Phys.* **2008**, *128*, 084102.

(86) Hill, J. G.; Peterson, K. A. Correlation consistent basis sets for explicitly correlated wavefunctions: Valence and core-valence basis sets for Li, Be, Na, and Mg. *Phys. Chem. Chem. Phys.* **2010**, *12*, 10460−10468.

https://doi.org/10.1021/acs.jctc.2c00215

(87) Hill, J. G.; Peterson, K. A. Correlation consistent basis sets for explicitly correlated wavefunctions: Pseudopotential-based basis sets for the post-*d* main group elements Ga-Rn. *J. Chem. Phys.* **2014**, *141*, 094106.

(88) Shaw, R. A.; Hill, J. G. Approaching the Hartree-Fock Limit through the Complementary Auxiliary Basis Set Singles Correction and Auxiliary Basis Sets. *J. Chem. Theory Comput.* **2017**, *13*, 1691−1698.

(89) Kritikou, S.; Hill, J. G. Auxiliary Basis Sets for Density Fitting in Explicitly Correlated Calculations: The Atoms H-Ar. *J. Chem. Theory Comput.* **2015**, *11*, 5269−5276.

(90) *Intel C++ Compiler*, ver. 19.1.0.166, 2019. https://software.intel.com/c-compilers.

(91) ROCm 3.8.0, 2021. https://www.amd.com/en/graphics/servers-solutions-rocm.

(92) Pulay, P. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem. Phys. Lett.* **1980**, *73*, 393−398.

(93) Pulay, P. Improved SCF Convergence Acceleration. *J. Comput. Chem.* **1982**, *3*, 556−560.

(94) Boys, S. F.; Bernardi, F. The calculation of small molecular interactions by the differences of separate total energies. Some procedures with reduced errors. *Mol. Phys.* **1970**, *19*, 553−566.

(95) Jurečka, P.; Šponer, J.; Černy, J.; Hobza, P. Benchmark database of accurate (MP2 and CCSD(T) complete basis set limit) interaction energies of small model complexes, DNA base pairs, and amino acid pairs. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1985−1993.

(96) Řezáč, J.; Riley, K. E.; Hobza, P. S66: A well-balanced database of benchmark interaction energies relevant to biomolecular structures. *J. Chem. Theory Comput.* **2011**, *7*, 2427−2438.

(97) Grimme, S.; Steinmetz, M.; Korth, M. How to compute isomerization energies of organic molecules with quantum chemical methods. *J. Org. Chem.* **2007**, *72*, 2118−2126.

(98) Sedlak, R.; Janowski, T.; Pitoňák, M.; Řezáč, J.; Pulay, P.; Hobza, P. Accuracy of quantum chemical methods for large noncovalent complexes. *J. Chem. Theory Comput.* **2013**, *9*, 3364−3374.

(99) Kussmann, J.; Ochsenfeld, C. Linear-scaling method for calculating nuclear magnetic resonance chemical shifts using gauge-including atomic orbitals within Hartree-Fock and density-functional theory. *J. Chem. Phys.* **2007**, *127*, 054103.

Highly Efficient and Accurate Computation of Multiple Orbital Spaces Spanning Fock Matrix Elements on Central and Graphics Processing Units for Application in F12 Theory:

Supporting Information

Lars Urban,[†,‡] Henryk Laqua,[†] and Christian Ochsenfeld[*,†,‡]

†*Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany*

‡*Max Planck Institute for Solid State Research, D-70569 Stuttgart, Germany*

E-mail: christian.ochsenfeld@uni-muenchen.de

1 Non-Covalent Interaction Energy MAEs/ Isomerization Energy MAEs

Table 1: Detailed MAEs of the HF, CABS singles, MP2, F12, and total non-covalent interaction energies for the S22+S66 test set using our sn-LinK, RI-J, and RI-K methods with different grid combinations and RI basis sets. $*^1$RI-J values. $*^2$RI-JK values.

Test Set(s)	Basis Set(s)	Grid	SCF [µH]	CABS [µH]	MP2 [µH]	F12 [µH]	tot. MAE [µH]	tot. MAE [kJ·mol^{-1}]
S22 + S66 (NCI)	cc-pVDZ-F12	gm[2/0]	1.06	4.47	9.09	0.76	9.61	0.0252
		gm[3/1]	0.19	0.65	2.30	0.22	2.56	0.0067
		gm[4/2]	0.08	0.14	0.45	0.04	0.55	0.0014
		gm[5/3]	0.03	0.03	0.14	0.01	0.16	0.0004
		gm[6/4]	0.01	0.01	0.05	0.01	0.05	0.0001
		gm[7/5]	0.00	0.00	0.03	0.00	0.03	0.0001
	+ cc-pVDZ-JKfit^{*1}	-	0.88	0.57	0.75	0.06	1.09	0.0029
	+ cc-pVTZ-JKfit^{*1}	-	0.45	0.36	0.37	0.04	0.74	0.0019
	+ cc-pVDZ-JKfit^{*2}	-	11.42	2.40	2.47	1.30	13.85	0.0364
	+ cc-pVTZ-JKfit^{*2}	-	9.89	0.73	1.82	0.45	12.04	0.0316
	+ cc-pVDZ-JKfit	gm[5/3]	0.89	0.57	0.76	0.06	1.12	0.0029
	+ cc-pVTZ-JKfit	gm[5/3]	0.44	0.36	0.43	0.04	0.79	0.0021
	cc-pVDTZ-F12	gm[2/0]	1.44	3.87	10.87	0.40	12.09	0.0317
		gm[3/1]	0.21	0.34	2.59	0.12	2.66	0.0070
		gm[4/2]	0.08	0.07	0.52	0.02	0.58	0.0015
		gm[5/3]	0.03	0.01	0.16	0.01	0.17	0.0004
		gm[6/4]	0.01	0.00	0.05	0.00	0.06	0.0001
		gm[7/5]	0.00	0.00	0.03	0.00	0.03	0.0001
	+ cc-pVTZ-JKfit^{*1}	-	0.72	0.11	0.45	0.03	0.79	0.0021
	+ cc-pVQZ-JKfit^{*1}	-	0.21	0.04	0.23	0.02	0.37	0.0009
	+ cc-pVTZ-JKfit^{*2}	-	9.97	0.33	2.27	0.34	12.64	0.0332
	+ cc-pVQZ-JKfit^{*2}	-	2.97	0.14	0.50	0.11	3.56	0.0093
	+ cc-pVTZ-JKfit	gm[5/3]	0.73	0.11	0.50	0.03	0.84	0.0022
	+ cc-pVQZ-JKfit	gm[5/3]	0.22	0.05	0.31	0.02	0.44	0.0011
	cc-pVQZ-F12	gm[2/0]	2.63	7.12	11.13	0.17	12.56	0.0330
		gm[3/1]	0.22	0.40	2.72	0.05	2.69	0.0071
		gm[4/2]	0.09	0.07	0.56	0.01	0.57	0.0015
		gm[5/3]	0.03	0.01	0.17	0.00	0.18	0.0004
		gm[6/4]	0.00	0.01	0.06	0.00	0.06	0.0002
		gm[7/5]	0.00	0.00	0.03	0.00	0.03	0.0001
	+ cc-pVQZ-JKfit^{*1}	-	0.23	0.43	0.26	0.02	0.81	0.0021
	+ cc-pV5Z-JKfit^{*1}	-	0.15	0.37	0.14	0.02	0.60	0.0015
	+ cc-pVQZ-JKfit^{*2}	-	2.91	0.14	0.57	0.10	3.56	0.0093
	+ cc-pV5Z-JKfit^{*2}	-	6.66	0.01	0.07	0.07	6.66	0.0175
	+ cc-pVQZ-JKfit	gm[5/3]	0.21	0.29	0.33	0.02	0.68	0.0018
	+ cc-pV5Z-JKfit	gm[5/3]	0.12	0.23	0.24	0.02	0.47	0.0012

Table 2: Detailed MAEs of the HF, CABS singles, MP2, F12, and total non-covalent interaction energies for the L7 test set using our sn-LinK, RI-J, and RI-K methods with different grid combinations. ^{*1}RI-J values. ^{*2}RI-JK values.

Test Set(s)	Basis Set(s)	Grid	SCF [μH]	CABS [μH]	MP2 [μH]	F12 [μH]	tot. MAE [μH]	tot. MAE [kJ·mol^{-1}]
L7 (NCI)	cc-pVDZ-F12	gm[2/0]	8.47	12.53	88.78	3.36	98.65	0.2590
		gm[3/1]	1.76	2.83	49.87	1.87	47.96	0.1259
		gm[4/2]	0.58	0.23	8.21	0.35	8.75	0.0230
		gm[5/3]	0.15	0.10	1.98	0.05	1.87	0.0049
		gm[6/4]	0.05	0.06	0.70	0.03	0.70	0.0018
		gm[7/5]	0.01	0.01	0.13	0.06	0.20	0.0005
	+ cc-pVDZ-JKfit^{*1}	-	6.37	3.10	5.03	0.39	4.11	0.0108
	+ cc-pVTZ-JKfit^{*1}	-	1.64	0.70	0.88	0.09	2.15	0.0056
	+ cc-pVDZ-JKfit^{*2}	-	129.27	6.61	10.88	4.78	136.03	0.3571
	+ cc-pVTZ-JKfit^{*2}	-	187.68	3.41	8.41	1.73	196.05	0.5147
	+ cc-pVDZ-JKfit	gm[5/3]	6.23	3.11	5.89	0.44	4.08	0.0100
	+ cc-pVTZ-JKfit	gm[5/3]	1.69	0.66	2.27	0.12	3.28	0.0086

Table 3: Detailed MAEs of the HF, CABS singles, MP2, F12, and total isomerization energies for the ISO34 test set using our sn-LinK, RI-J, and RI-K methods with different grid combinations. ^{*1}RI-J values. ^{*2}RI-JK values.

Test Set(s)	Basis Set(s)	Grid	SCF [μH]	CABS [μH]	MP2 [μH]	F12 [μH]	tot. MAE [μH]	tot. MAE [kJ·mol^{-1}]
ISO34 (Isomerization)	cc-pVDZ-F12	gm[2/0]	52.93	357.44	157.42	13.64	424.36	1.1138
		gm[3/1]	14.45	33.36	31.50	2.45	57.11	0.1499
		gm[4/2]	5.08	8.94	5.58	0.65	12.31	0.0323
		gm[5/3]	1.27	1.59	1.53	0.18	2.91	0.0076
		gm[6/4]	0.23	0.32	0.33	0.04	0.70	0.0018
		gm[7/5]	0.03	0.08	0.11	0.01	0.18	0.0004
	+ cc-pVDZ-JKfit^{*1}	-	38.88	14.78	9.03	0.60	44.82	0.1177
	+ cc-pVTZ-JKfit^{*1}	-	5.08	2.84	1.45	0.16	6.48	0.0170
	+ cc-pVDZ-JKfit^{*2}	-	28.53	21.08	11.03	2.68	45.97	0.1207
	+ cc-pVTZ-JKfit^{*2}	-	8.43	4.56	1.27	0.77	12.98	0.0341
	+ cc-pVDZ-JKfit	gm[5/3]	38.63	14.98	9.65	0.65	45.62	0.1198
	+ cc-pVTZ-JKfit	gm[5/3]	5.47	3.27	2.51	0.28	7.53	0.0198
	cc-pVDTZ-F12	gm[2/0]	71.97	38.24	148.44	8.10	156.11	0.4099
		gm[3/1]	15.61	1.20	32.72	1.39	36.43	0.0956
		gm[4/2]	5.21	0.25	5.57	0.33	7.05	0.0185
		gm[5/3]	1.31	0.09	1.52	0.09	2.11	0.0055
		gm[6/4]	0.24	0.10	0.32	0.03	0.57	0.0015
		gm[7/5]	0.03	0.11	0.11	0.01	0.20	0.0005
	+ cc-pVTZ-JKfit^{*1}	-	7.23	1.03	1.75	0.10	6.82	0.0179
	+ cc-pVQZ-JKfit^{*1}	-	0.69	0.22	0.25	0.02	0.75	0.0020
	+ cc-pVTZ-JKfit^{*2}	-	11.77	1.58	1.90	0.55	13.95	0.0366
	+ cc-pVQZ-JKfit^{*2}	-	1.62	0.26	0.39	0.15	1.91	0.0050
	+ cc-pVTZ-JKfit	gm[5/3]	7.51	1.23	2.71	0.16	7.31	0.0192
	+ cc-pVQZ-JKfit	gm[5/3]	1.76	0.55	1.59	0.10	2.44	0.0064
	cc-pVQZ-F12	gm[2/0]	105.84	207.81	153.53	4.60	277.10	0.7275
		gm[3/1]	15.31	28.48	31.10	0.73	44.38	0.1165
		gm[4/2]	5.15	3.79	5.26	0.16	9.17	0.0241
		gm[5/3]	1.29	0.96	1.46	0.04	2.37	0.0062
		gm[6/4]	0.24	0.53	0.31	0.01	0.80	0.0021
		gm[7/5]	0.00	0.45	0.11	0.00	0.47	0.0012
	+ cc-pVQZ-JKfit^{*1}	-	0.69	0.55	0.27	0.01	0.79	0.0021
	+ cc-pV5Z-JKfit^{*1}	-	0.21	0.25	0.12	0.01	0.31	0.0008
	+ cc-pVQZ-JKfit^{*2}	-	1.58	0.09	0.49	0.17	1.73	0.0046
	+ cc-pV5Z-JKfit^{*2}	-	2.36	0.05	0.32	0.08	2.47	0.0065
	+ cc-pVQZ-JKfit	gm[5/3]	1.80	1.04	1.53	0.05	2.78	0.0072
	+ cc-pV5Z-JKfit	gm[5/3]	1.45	0.93	1.48	0.05	2.41	0.0063

2 Absolute Energy MAEs

Table 4: Detailed MAEs of the HF, CABS singles, MP2, F12, and total absolute energies for the S22+S66 test set using our sn-LinK, RI-J, and RI-K methods with different grid combinations and RI basis sets. ^{*1}RI-J values. ^{*2}RI-JK values.

Test Set(s)	Basis Set(s)	Grid	SCF [μH]	CABS [μH]	MP2 [μH]	F12 [μH]	tot. MAE [μH]	tot. MAE [kJ·mol^{-1}]
S22 + S66 (Absolute)	cc-pVDZ-F12	gm[2/0]	89.51	735.12	147.07	17.83	735.35	1.9307
		gm[3/1]	10.17	22.05	22.08	3.06	39.86	0.1047
		gm[4/2]	3.15	6.37	5.81	0.53	11.04	0.0290
		gm[5/3]	1.01	1.08	0.74	0.14	1.83	0.0048
		gm[6/4]	0.12	0.25	0.26	0.04	0.44	0.0011
		gm[7/5]	0.03	0.09	0.07	0.02	0.13	0.0004
	+ cc-pVDZ-JKfit^{*1}	-	218.57	62.03	34.70	2.24	247.67	0.6503
	+ cc-pVTZ-JKfit^{*1}	-	108.12	29.57	15.42	0.73	122.56	0.3218
	+ cc-pVDZ-JKfit^{*2}	-	561.69	364.40	188.87	398.08	1511.93	3.9696
	+ cc-pVTZ-JKfit^{*2}	-	138.20	72.02	88.84	177.73	476.20	1.2503
	+ cc-pVDZ-JKfit	gm[5/3]	218.71	61.79	34.96	2.24	247.29	0.6493
	+ cc-pVTZ-JKfit	gm[5/3]	108.26	29.37	15.67	0.73	122.18	0.3208
	cc-pVDTZ-F12	gm[2/0]	210.78	1327.61	147.08	8.91	1185.88	3.1135
		gm[3/1]	11.42	28.67	22.90	1.54	34.21	0.0898
		gm[4/2]	3.29	2.76	5.88	0.27	6.13	0.0161
		gm[5/3]	1.06	0.51	0.79	0.07	1.53	0.0040
		gm[6/4]	0.12	0.10	0.26	0.02	0.27	0.0007
		gm[7/5]	0.03	0.03	0.08	0.01	0.09	0.0002
	+ cc-pVTZ-JKfit^{*1}	-	156.26	3.44	23.75	0.78	136.60	0.3586
	+ cc-pVQZ-JKfit^{*1}	-	32.44	4.03	1.12	0.06	27.75	0.0729
	+ cc-pVTZ-JKfit^{*2}	-	174.18	33.96	245.93	110.17	563.92	1.4806
	+ cc-pVQZ-JKfit^{*2}	-	26.04	8.55	28.36	32.25	95.20	0.2499
	+ cc-pVTZ-JKfit	gm[5/3]	156.34	3.61	24.02	0.78	136.56	0.3585
	+ cc-pVQZ-JKfit	gm[5/3]	32.52	3.88	1.54	0.10	27.72	0.0728
	cc-pVQZ-F12	gm[2/0]	523.49	2631.00	141.18	5.16	2157.07	5.6634
		gm[3/1]	14.93	59.18	22.55	0.78	52.78	0.1386
		gm[4/2]	3.25	3.58	5.67	0.14	5.82	0.0153
		gm[5/3]	1.07	0.58	0.80	0.04	1.59	0.0042
		gm[6/4]	0.13	0.23	0.26	0.01	0.32	0.0008
		gm[7/5]	0.00	0.07	0.08	0.00	0.11	0.0003
	+ cc-pVQZ-JKfit^{*1}	-	29.94	2.88	1.51	0.06	31.70	0.0832
	+ cc-pV5Z-JKfit^{*1}	-	21.23	1.79	0.63	0.03	23.53	0.0618
	+ cc-pVQZ-JKfit^{*2}	-	32.95	1.22	49.60	16.88	100.66	0.2643
	+ cc-pV5Z-JKfit^{*2}	-	11.71	0.97	13.14	8.96	33.47	0.0879
	+ cc-pVQZ-JKfit	gm[5/3]	30.04	1.73	1.88	0.06	30.29	0.0795
	+ cc-pV5Z-JKfit	gm[5/3]	21.33	0.82	1.07	0.05	22.13	0.0581

Table 5: Detailed MAEs of the HF, CABS singles, MP2, F12, and total absolute energies for the L7 test set using our sn-LinK, RI-J, and RI-K methods with different grid combinations. ^{*1}RI-J values. ^{*2}RI-JK values.

Test Set(s)	Basis Set(s)	Grid	SCF [μH]	CABS [μH]	MP2 [μH]	F12 [μH]	tot. MAE [μH]	tot. MAE [kJ·mol^{-1}]
L7 (Absolute)	cc-pVDZ-F12	gm[2/0]	706.17	4519.68	1440.70	187.91	5558.74	14.5945
		gm[3/1]	99.35	94.74	137.98	40.27	297.76	0.7818
		gm[4/2]	32.90	52.02	66.39	5.81	126.71	0.3327
		gm[5/3]	6.53	11.55	10.07	1.91	18.13	0.0476
		gm[6/4]	1.55	2.10	2.65	0.48	5.29	0.0139
		gm[7/5]	0.30	0.82	0.85	0.13	1.83	0.0048
	+ cc-pVDZ-JKfit^{*1}	-	962.58	315.08	183.61	7.80	1101.85	2.8929
	+ cc-pVTZ-JKfit^{*1}	-	539.44	146.05	90.33	3.19	598.34	1.5710
	+ cc-pVDZ-JKfit^{*2}	-	3414.55	2193.01	1076.44	1901.82	8585.82	22.5421
	+ cc-pVTZ-JKfit^{*2}	-	1070.82	434.30	491.17	842.63	2838.92	7.4536
	+ cc-pVDZ-JKfit	gm[5/3]	968.18	322.63	178.57	7.99	1119.24	2.9386
	+ cc-pVTZ-JKfit	gm[5/3]	545.03	153.59	85.29	3.35	615.72	1.6166

Table 6: Detailed MAEs of the HF, CABS singles, MP2, F12, and total absolute energies for the ISO34 test set using our sn-LinK, RI-J, and RI-K methods with different grid combinations. ^{*1}RI-J values. ^{*2}RI-JK values.

Test Set(s)	Basis Set(s)	Grid	SCF [μH]	CABS [μH]	MP2 [μH]	F12 [μH]	tot. MAE [μH]	tot. MAE [kJ·mol^{-1}]
ISO34 (Absolute)	cc-pVDZ-F12	gm[2/0]	68.75	483.70	146.12	13.75	481.03	1.2629
		gm[3/1]	9.48	25.01	21.57	2.36	39.66	0.1041
		gm[4/2]	2.90	6.83	4.63	0.43	10.45	0.0274
		gm[5/3]	0.73	0.98	0.87	0.11	1.69	0.0044
		gm[6/4]	0.12	0.22	0.22	0.03	0.44	0.0012
		gm[7/5]	0.02	0.06	0.07	0.01	0.11	0.0003
	+ cc-pVDZ-JKfit^{*1}	-	175.58	55.91	30.90	2.26	202.85	0.5326
	+ cc-pVTZ-JKfit^{*1}	-	66.61	21.87	10.24	0.68	78.92	0.2072
	+ cc-pVDZ-JKfit^{*2}	-	226.04	147.22	115.97	199.94	689.17	1.8094
	+ cc-pVTZ-JKfit^{*2}	-	61.81	27.01	49.09	88.92	226.76	0.5954
	+ cc-pVDZ-JKfit	gm[5/3]	175.81	56.09	30.74	2.21	203.37	0.5339
	+ cc-pVTZ-JKfit	gm[5/3]	66.84	22.06	10.13	0.65	79.44	0.2086
	cc-pVDTZ-F12	gm[2/0]	139.56	717.20	141.14	8.02	653.14	1.7148
		gm[3/1]	9.73	20.49	23.14	1.29	28.36	0.0745
		gm[4/2]	2.90	2.50	4.61	0.22	6.22	0.0163
		gm[5/3]	0.73	0.40	0.85	0.06	1.27	0.0033
		gm[6/4]	0.13	0.11	0.22	0.02	0.34	0.0009
		gm[7/5]	0.02	0.02	0.07	0.00	0.07	0.0002
	+ cc-pVTZ-JKfit^{*1}	-	98.99	3.18	15.10	0.58	87.58	0.2299
	+ cc-pVQZ-JKfit^{*1}	-	19.56	2.18	0.64	0.04	16.91	0.0444
	+ cc-pVTZ-JKfit^{*2}	-	69.17	15.86	129.67	55.81	270.52	0.7102
	+ cc-pVQZ-JKfit^{*2}	-	12.14	4.63	14.85	16.27	47.89	0.1257
	+ cc-pVTZ-JKfit	gm[5/3]	99.15	3.33	14.95	0.56	88.01	0.2311
	+ cc-pVQZ-JKfit	gm[5/3]	19.72	2.04	1.07	0.07	17.33	0.0455
	cc-pVQZ-F12	gm[2/0]	325.47	1386.96	140.29	4.87	1128.47	2.9628
		gm[3/1]	10.47	37.83	22.00	0.67	34.73	0.0912
		gm[4/2]	2.86	2.50	4.33	0.11	6.39	0.0168
		gm[5/3]	0.73	0.42	0.82	0.03	1.37	0.0036
		gm[6/4]	0.13	0.11	0.21	0.01	0.36	0.0010
		gm[7/5]	0.00	0.04	0.06	0.00	0.08	0.0002
	+ cc-pVQZ-JKfit^{*1}	-	18.06	1.47	0.89	0.04	18.71	0.0491
	+ cc-pV5Z-JKfit^{*1}	-	13.18	1.04	0.22	0.01	14.33	0.0376
	+ cc-pVQZ-JKfit^{*2}	-	16.10	0.56	25.61	8.88	51.16	0.1343
	+ cc-pV5Z-JKfit^{*2}	-	5.43	0.47	6.73	4.73	17.28	0.0454
	+ cc-pVQZ-JKfit	gm[5/3]	18.24	0.87	1.20	0.05	18.32	0.0481
	+ cc-pV5Z-JKfit	gm[5/3]	13.36	0.51	0.89	0.03	13.94	0.0366

3 RI-J/sn-LinK Speedups

Table 7: RI-J/sn-LinK (cc-pVYZ-JKfit/gm[X+2/X]; Y = D, T Q) speedups on CPUs (S_{CPU}) and GPUs (S_{GPU}) for the full F12-type Fock build for each member of the L7 test (cc-pVXZ-F12; X = D, T, Q).*1 Reference extrapolated from double- and triple-zeta F12 timings.

				gm[2/0]		gm[3/1]		gm[4/2]		gm[5/3]		gm[6/4]		gm[7/5]	
Basis Set	L7 Structure	N_{bas}	N_{CABS}	S_{CPU}	S_{GPU}	S_{CPU}	S_{GPU}	S_{CPU}	S_{GPU}	S_{CPU}	S_{GPU}	S_{CPU}	S_{GPU}	S_{CPU}	S_{GPU}
cc-pVDZ-F12 + cc-pVDZ-JKfit	L1	1836	5288	209	528	153	442	89	339	52	246	34	184	25	148
	L2	1191	3426	251	469	196	454	123	357	72	284	48	226	35	182
	L3	2255	6486	517	1340	367	1071	214	877	123	627	82	458	60	358
	L4	2588	7444	584	1635	419	1311	240	1060	138	715	91	535	67	422
	L5	1818	5232	316	742	232	723	134	530	79	382	52	294	38	233
	L6	1752	5044	446	1072	327	1037	190	778	109	561	72	409	53	323
	L7	1396	4016	311	682	228	630	137	495	82	371	53	281	39	224
cc-pVTZ-F12 + cc-pVTZ-JKfit	L1	3676	7804	693	1627	496	1353	289	963	169	623	111	475	82	363
	L2	2331	4311	733	1550	554	1295	351	955	208	668	138	502	102	382
	L3	4405	8044	1423	3403	1007	2718	589	2050	342	1345	223	949	154	706
	L4	5058	9267	1575	4155	1137	3318	645	2276	346	1551	212	1058	172	795
	L5	3588	6999	976	2017	695	1845	410	1334	240	928	157	646	116	507
	L6	3432	6384	1308	2884	971	2715	554	1811	322	1192	211	864	154	650
	L7	2736	5106	939	1880	686	1688	412	1237	239	847	156	637	117	491
cc-pVQZ-F12*1 + cc-pVQZ-JKfit	L1	6996	9700	1551	-	1113	-	637	-	325	-	229	-	163	-
	L2	4281	5229	1488	-	1140	-	668	-	358	-	247	-	185	-
	L3	8065	9733	2761	-	1978	-	1045	-	613	-	413	-	297	-
	L4	9268	11220	2710	-	2053	-	1230	-	704	-	449	-	333	-
	L5	6678	8574	2024	-	1497	-	787	-	453	-	300	-	221	-
	L6	6312	7752	2548	-	1897	-	986	-	578	-	374	-	268	-
	L7	5036	6204	1918	-	1404	-	731	-	442	-	288	-	207	-

4 RI-K

Following the approach of Weigend in Ref 1 (reference 12 in article), the 4-center-2-electron repulsion integral tensor is approximated as

$$(\mu\nu|\lambda\sigma) \approx \sum_{PQ} (\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma), \tag{1}$$

where $(P|Q)^{-1}$ denotes the matrix inverse of the auxiliary basis 2-center-2-electron repulsion integrals $(P|Q)$, and which becomes exact in the limit of complete auxiliary basis sets. Inserting Eq. 1 into the atomic orbital expression for the exchange matrix

$$K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\sigma|\lambda\nu)P_{\lambda\sigma}, \tag{2}$$

decomposing

$$(P|Q)^{-1} = \sum_{R} (P|R)^{-\frac{1}{2}}(R|Q)^{-\frac{1}{2}}, \tag{3}$$

and the density matrix

$$P_{\lambda\sigma} = \sum_{i} C_{\lambda i}C_{\sigma i}, \tag{4}$$

where $C_{\lambda i}$ denotes occupied MO coefficients (or, equivalently, Cholesky factors of $\mathbf{P}$), leads to the RI-K expression (cf. Eq. 4 of ref. 1)

$$K_{\mu\nu} = \sum_{\lambda\sigma i PQR} (\mu\sigma|P)(P|R)^{-\frac{1}{2}}(R|Q)^{-\frac{1}{2}}(Q|\lambda\nu)C_{\lambda i}C_{\sigma i}\,. \tag{5}$$

There are multiple viable approaches to evaluate Eq. 5: In the original implementation of Weigend, only $(P|Q)^{-\frac{1}{2}}$ is initially precomputed and the exchange matrix is then formed in each self-consistent field (SCF) iteration as:

$$\begin{aligned}
(i\mu|P) &= \sum_{\nu} C_{\nu i}(\nu\mu|P) &\quad (6)\\
B^{Q}_{i\mu} &= \sum_{P}(i\mu|P)(P|Q)^{-\frac{1}{2}} &\quad (7)\\
K_{\mu\nu} &= \sum_{iQ} B^{Q}_{i\mu}B^{Q}_{i\nu}\,. &\quad (8)
\end{aligned}$$

In an alternative approach, the untransformed 3-center-1-electron integrals $(\mu\nu|P)$ are transformed with $(P|Q)^{-\frac{1}{2}}$ in an additional precomputation step

$$B^{Q}_{\nu\mu} = \sum_{P}(\nu\mu|P)(P|Q)^{-\frac{1}{2}} \qquad (9)$$

and the exchange matrix is then obtained in each SCF iteration as

$$\begin{aligned}
B^{Q}_{i\mu} &= \sum_{\nu} C_{\nu i}B^{Q}_{\nu\mu} & & \quad (10)\\
K_{\mu\nu} &= \sum_{iQ} B^{Q}_{i\mu}B^{Q}_{i\nu} & \text{(same as Eq. 8).} & \quad (11)
\end{aligned}$$

This approach avoids the second transformation step (Eq. 7) within each SCF cycle at the cost of an additional preparatory step and a higher memory demand (storage of $B^{Q}_{\nu\mu}$).
In practice, the second approach is highly beneficial for iterative procedures like the SCF method, where the cost of the preparation is overall less significant. For F12-type-Fock builds, however, only a single K-build needs to be performed, so the cost of the precomputation in Eq. 9 outweighs the savings from avoiding the evaluation of Eq. 7. Therefore, we decided to use the precomputation RI-K method (second approach) for the SCF calculation and the integral-direct variant (first approach) for the F12-type Fock build.

References

(1) Weigend, F. A fully direct RI-HF algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency. *Phys. Chem. Chem. Phys.* **2002**, *4*, 4285–4291.

3.6 Publication VI: An improved molecular partitioning scheme for numerical quadratures in density functional theory

H. Laqua, J. Kussmann, C. Ochsenfeld

J. Chem. Phys. **149**, 204111 (2018).

Abstract

We present a modification to Becke's molecular partitioning scheme [A. D. Becke, J. Chem. Phys. **88**, 2547 (1988)] that provides substantially better accuracy for weakly bound complexes and allows for a faster and linear scaling grid generation without introducing a cutoff error. We present the accuracy of our new partitioning scheme for atomization energies of small molecules and for interaction energies of van der Waals complexes. Furthermore, the efficiency and scaling behavior of the grid generation are demonstrated for large molecular systems with up to 1707 atoms.

Reproduced from:

H. Laqua, J. Kussmann, C. Ochsenfeld
"An improved molecular partitioning scheme for numerical quadratures in density functional theory"
J. Chem. Phys. **149**, 204111 (2018).

with the permission of AIP Publishing.
`https://aip.scitation.org/doi/10.1063/1.5049435`

An improved molecular partitioning scheme for numerical quadratures in density functional theory

Henryk Laqua, Jörg Kussmann, and Christian Ochsenfeld[a)]
Department of Chemistry and Center for Integrated Protein Science (CIPSM), University of Munich (LMU), D-81377 München, Germany

(Received 21 July 2018; accepted 26 October 2018; published online 29 November 2018)

We present a modification to Becke's molecular partitioning scheme [A. D. Becke, J. Chem. Phys. **88**, 2547 (1988)] that provides substantially better accuracy for weakly bound complexes and allows for a faster and linear scaling grid generation without introducing a cutoff error. We present the accuracy of our new partitioning scheme for atomization energies of small molecules and for interaction energies of van der Waals complexes. Furthermore, the efficiency and scaling behavior of the grid generation are demonstrated for large molecular systems with up to 1707 atoms. *Published by AIP Publishing.*
https://doi.org/10.1063/1.5049435

I. INTRODUCTION

During the last decades, density functional theory (DFT),[1] in particular, in combination with empirical dispersion corrections (see, e.g., Ref. 2), has become the *de facto* standard for electronic structure theory calculations due to its excellent cost-performance ratio. However, the evaluation of typical density functionals requires the integration of the exchange-correlation (XC) energy density over the whole 3D-space of the molecule. Since there exists no general analytical integration scheme for this purpose, a numerical grid-based quadrature is typically employed. Although the use of evenly spaced Cartesian grids is possible in principle, specifically designed atom-centered integration grids are more effective due to the high variance of the XC energy density, in particular, near the nuclei.

The atomic grids are typically constructed as the product of an angular and a radial grid. For the former, the schemes of Lebedev and Laikov[3] are ubiquitously employed, whereas for the latter, a variety of different schemes have been proposed.[4–8]

Finally, the individual atomic grids need to be merged into the molecular grid requiring a molecular partitioning scheme to account for the overlap of the atomic grids. For this purpose, Becke[4] proposed a weighting scheme employing smooth Voronoi polyhedrons to partition the molecule into regions for each atom and scaling the corresponding atomic grids accordingly. This approach has later been adjusted by Stratmann *et al.*[9] to allow for a more efficient and asymptotically linear scaling construction of the grid.

However, both partitioning schemes lead to artificial oscillations in the energy surface at larger interatomic distances (see Sec. II) which we trace back to an insufficient sharpness of the partitioning profile in these situations (see Sec. III B). Therefore, we propose a simple modification to Becke's partitioning function, adjusting the sharpness at large (and only at large) distances, in this way completely removing the above-mentioned artifacts. Moreover, this adjustment introduces a locality into the weighting scheme that allows for a very efficient and linear scaling generation of the grid without the introduction of a cutoff error.

The paper is organized as follows: First, we motivate the development of our new partitioning scheme at the example of the argon-dimer dissociation in Sec. II. Subsequently, we briefly review and benchmark Becke's partitioning scheme[4] in Secs. III A and III B, respectively, describe our modified scheme in Sec. III C, and present a linear-scaling algorithm for the generation of the molecular partitioning weights in Sec. III D. Finally, we compare the accuracy of our modified scheme with the existing schemes of Becke[4] and Stratmann *et al.*[9] for a selection of test sets (G2,[10,11] L7,[12] S22 × 5[13]) in Sec. IV A and present the computational efficiency of our grid generation in Sec. IV B. Moreover, the full specification of the numerical integration grids employed in our FermiONs++ program[14–16] is given in the Appendix.

II. MOTIVATION: Ar–Ar DISSOCIATION CURVE

We compare the existing partitioning schemes of Becke[4] (denoted as "Becke") and Stratmann, Scuseria, and Frisch[9] (denoted as "SSF") with the modified scheme of the present work (denoted as "mod. Becke") for the argon-dimer dissociation curve, exemplifying a typical van der Waals complex. The functional of Perdew, Burke, and Ernzerhof (PBE)[17] and the def2-TZVP basis set[18] have been employed in all calculations as they are widely used today (see Sec. IV A for more computational details).

Figure 1 illustrates the shortcomings of existing partitioning schemes, exhibiting unphysical energy oscillations at large interatomic distances (considerably outside of the experimental van der Waals minimum at 3.76 Å[19]). This

[a)]Electronic mail: christian.ochsenfeld@uni-muenchen.de

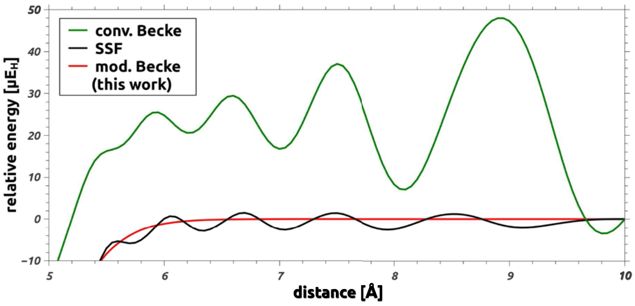


FIG. 1. Comparison of different partitioning schemes for the dissociation curve of the argon-dimer employing the FermiONs++ program with the "g5"-grid, the def2-TZVP basis set, and the PBE functional.

artifact is caused by the partitioning function not being sharp enough at larger distances. Therefore, grid points originating from one center but appearing close to another center have a non-zero weight and thus interfere with the atomic grid of the other center, resulting in a substantially increased grid error. Since the SSF partitioning scheme is sharper than the Becke scheme (see also discussions below), the effect is less pronounced for the SSF scheme. This observation motivated the development of our new weighting scheme, described in Secs. III A and III C, where the "smoothness" is capped at some given interatomic distance R_{cutoff}, in this way providing an excellent dissociation curve.

Since both the Becke and the SSF partitioning scheme are ubiquitously used throughout quantum chemistry (QC), we expect this artifact to also be present in other commonly used QC-programs. Therefore we compare our FermiONs++ program with the QChem,[20] Turbomole,[21] and ORCA[22] program packages in Fig. 2, employing default settings if not stated otherwise. To allow for a fair comparison, we choose grids of comparable sizes (about 20 000 points per atom), except for QChem where we use the unpruned [99/590]-grid since a comparable pruned grid is not available in our version. From Fig. 2, we note that energy oscillations at large interatomic distances are indeed ubiquitously present in commonly used quantum chemistry programs, which can, however, be removed completely by employing the new partitioning scheme of our present paper, which we describe below.

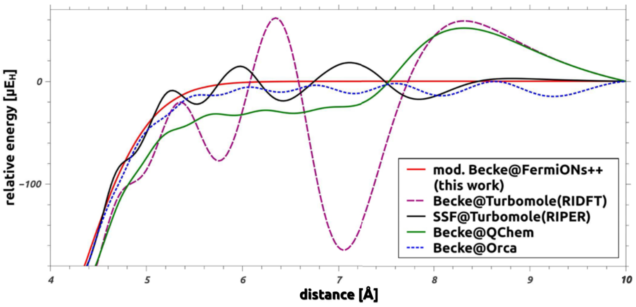


FIG. 2. Comparison of different program packages for the dissociation curve of the argon-dimer employing the def2-TZVP basis set and the PBE functional. The following grids have been employed: FermiONs++: "g5"; Turbomole: gridsize "5"; QChem: "99/590"; Orca: "GRID6."

III. THEORY

A. Becke's molecular partitioning scheme

For the molecular weighting scheme, the molecule is partitioned into smooth Voronoi polyhedrons (Dirichlet partitioning). We begin with the construction of the confocal elliptical coordinate μ defined as

$$\mu_{ij} = \frac{r_i - r_j}{R_{ij}}, \tag{1}$$

where r_i denotes the distance of the reference-point (i.e., the grid-point of interest) to the i-th atom and R_{ij} denotes the distance between atom i and atom j. Then, we apply the polynomial smoothing function given as

$$g(\mu_{ij}) = \underbrace{h(\ldots h(}_{k \text{ times}}\mu_{ij})\ldots), \tag{2}$$

with

$$h(x) = \frac{3}{2}x - \frac{1}{2}x^3, \tag{3}$$

resulting in a smoothly varying value of $g(\mu)$ in the interval $g \in [-1, 1]$. The result is then mapped linearly into the interval $s \in [0, 1]$ as

$$s(\mu_{ij}) = \frac{1}{2}[1 - g(\mu_{ij})]. \tag{4}$$

As pictured in Fig. 3, the sharpness of the partition is controlled by the integer parameter k in Eq. (2), where higher values for k result in a sharper partitioning. Becke[4] recommended the value of $k = 3$, which has been adopted in the studies of Refs. 5 and 6.

The unnormalized Voronoi polyhedron functions are subsequently constructed as

$$P_i(\mathbf{r}) = \prod_{i \neq j} s(\mu_{ij}) \tag{5}$$

and are finally normalized to yield the molecular partitioning weights

$$p_i(\mathbf{r}) = \frac{P_i(\mathbf{r})}{\sum_j P_j(\mathbf{r})}. \tag{6}$$

The molecular grid is then constructed from the individual atomic grids weighted by the corresponding partitioning function $p_p(\mathbf{r}_g)$ for each grid point $\mathbf{r}_g$, where p denotes the parent atom of the respective grid point (i.e., the center of the atomic grid the point originated from).

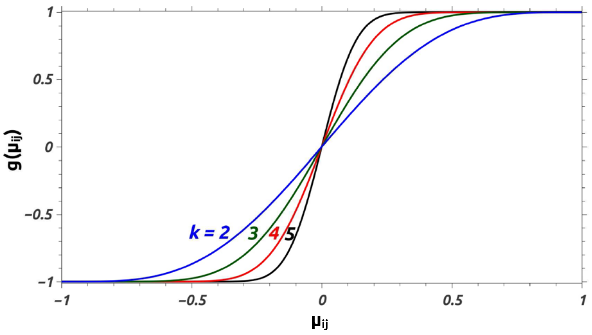


FIG. 3. Partitioning functions $g(\mu_{ij})$ for different values of k.

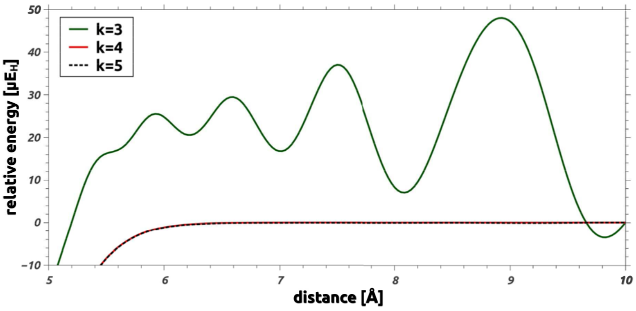


FIG. 4. Influence of different k values on the dissociation curve of the argon dimer with the "g5"-grid, the def2-TZVP basis set, and the PBE functional. Note that the curves for $k = 4$ and $k = 5$ are virtually identical in the represented region.

B. Accuracy benchmark of Becke's molecular partitioning scheme

Considering the shortcomings of the original Becke weighting scheme with $k = 3$ as discussed in Sec. II, the sharpness parameter k of Eq. (2) represents an obvious target for adjustments. First, we revisit the argon dimer dissociation curve in Fig. 4 and note that a sharper partitioning ($k \geq 4$) indeed provides substantially better (i.e., virtually perfect) results. Thus, only considering dissociation curves, one would question Becke's[4] initial choice of $k = 3$ and instead prefer to use a sharper partitioning profile with $k \geq 4$.

However, considering tightly bound molecules, e.g., the G2 test as represented in Fig. 5, leads to a different conclusion: here, the scheme with $k = 3$ outperforms all other schemes, particularly for coarser grids. This observation thus confirms Becke's[4] initial choice of $k = 3$.

Taking both of the above observations into account, we therefore conclude that at small distances the recommended value of $k = 3$ is indeed optimal, whereas at large interatomic separations a sharper partitioning profile is necessary for optimal results. Ideally, in the case of, e.g., supermolecular systems, one would choose a partitioning scheme with $k = 3$ for intramolecular atom pairs and $k = 4$ for intermolecular pairs. However, since a continuous transition between $k = 3$ and $k = 4$ for intermediate distances is not possible (k needs to be an integer), such a scheme would be impractical for general applications. Therefore, we propose a modified Becke scheme, which provides a continuous transition between both cases, below.

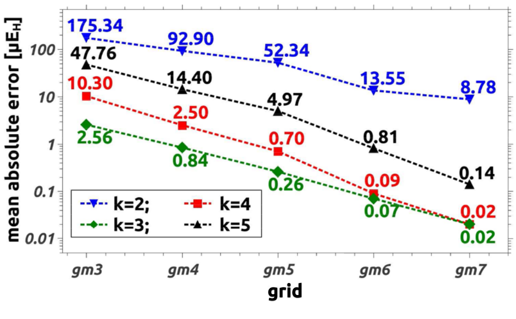


FIG. 5. Grid-induced mean absolute errors for the G2 test (atomization energies of small molecules) for different grids and sharpness parameters k at the PBE/def2-TZVP level compared to the very tight [150/2030]-reference grid (using the modified partitioning scheme of the present work). See also Sec. IV A for more computational details.

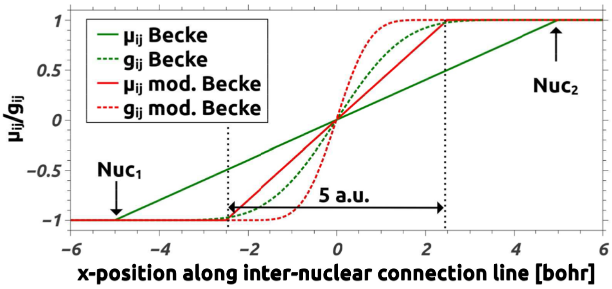


FIG. 6. Modified (new Becke) vs. unmodified (old Becke) μ_{ij} (continuous line) and g_{ij} (dashed line) along the nucleus-nucleus connection line for two atoms at 10 bohr distance.

C. Modification to the molecular partitioning scheme

The rationale behind our modified scheme is to leave the original partitioning profile with $k = 3$ unchanged at short distances R_{ij}, in this way preserving the superior accuracy for tightly bound molecules, and to choose a sharper profile for large interatomic separations, in this way providing smooth dissociation curves.

For this purpose, μ_{ij} of Eq. (1) is modified according to (see also Fig. 6 for a graphical representation)

$$\mu_{ij}^{\text{mod}} = \frac{r_i - r_j}{\min\left(R_{\text{cutoff}}, R_{ij}\right)}, \tag{7}$$

$$\mu_{ij}^{\text{mod, cutoff}} = \begin{cases} -1, & \mu_{ij}^{\text{mod}} \leq -1 \\ 1, & \mu_{ij}^{\text{mod}} \geq 1 \\ \mu_{ij}^{\text{mod}}, & \text{otherwise} \end{cases}. \tag{8}$$

Note that the partitioning profile remains unchanged for $R_{ij} \leq R_{\text{cutoff}}$.

By testing different values for R_{cutoff}, we found that $R_{\text{cutoff}} = 5$ bohrs provides optimal results in all our test cases and minor variations (±1 bohr) have virtually no influence on the accuracy of the weighting scheme. Furthermore, we want to emphasize that the modifications in Eqs. (7) and (8) require very little change to existing grid generation codes.

D. Efficient and linear scaling generation of the grid-weights

In addition to a superior accuracy, our new partitioning function allows for an efficient and linear-scaling generation of the molecular weight adjustments. Since the construction of the molecular grid scales formally as $\mathcal{O}(N_{grid}N_{at}^2)$, this step represents a computational bottleneck for the existing partitioning schemes, in particular, for periodic systems.[23–25] In previous work,[23–25] cutoff radii for significant atoms have been proposed, requiring a careful control of the so-introduced error.

However, our new partitioning function allows for a linear-scaling grid generation without such considerations due to the intrinsic locality of the new partitioning function given in Eqs. (7) and (8). Below, we present our linear-scaling grid generation algorithm in combination with the *formal* scaling behavior of each step:

```
 1: for all parent atoms p do
 2:   calculate distances to all other atoms $R_{pi}$                          $\mathcal{O}(N_{at}^2)$
 3:   sort $\{R_{pi}\}$ (in ascending order)                                   $\mathcal{O}(N_{at}^2 \log(N_{at}))$
 4:   for all grid points g ∈ atomic grid of p do
 5:     calculate $r_{gp}$                                                     $\mathcal{O}(N_{grid})$
 6:     initialize distance to nearest atom $r_{gn} = r_{gp}$                  $\mathcal{O}(N_{grid})$
 7:     for all atoms i sorted by $R_{pi}$ do
 8:       if $(R_{pi} > r_{gn} + r_{gp} + 2R_{\text{cutoff}})$ break
 9:       calculate $r_{gi}$                                                   $\mathcal{O}(N_{at}N_{grid})$
10:       update $r_{gn} = \min\left(r_{gn}, r_{gi}\right)$                   $\mathcal{O}(N_{at}N_{grid})$
11:     end for
12:     if $(r_{gn} > r_{gn} + R_{\text{cutoff}})$ continue
13:     sort $\{r_{gi}\}$ (in ascending order)                                 $\mathcal{O}(N_{at} \log(N_{at}) N_{grid})$
14:     for all atoms i sorted by $r_{gi}$ do
15:       if $(r_{gi} > r_{gn} + R_{\text{cutoff}})$ break
16:       for all atoms j > i sorted by $r_{gj}$ do
17:         if $(r_{gj} > r_{gi} + R_{\text{cutoff}})$ break
18:         calculate $s(\mu_{ij})$ according to Eqs. (1)–(4)                 $\mathcal{O}(N_{at}^2 N_{grid})$
19:         calculate $s(\mu_{ji}) = 1 - s(\mu_{ij})$ [utilize symmetry of $s(\mu_{ij})$]
                                                                               $\mathcal{O}(N_{at}^2 N_{grid})$
20:         calculate $P_i, P_j$ according to Eq. (5)                         $\mathcal{O}(N_{at}^2 N_{grid})$
21:     end for
22:   end for
23:     calculate molecular weight adjustment $p_p$ according
        to Eq. (6)                                                             $\mathcal{O}(N_{at}N_{grid})$
24:   end for
25: end for
```

Lines 18-20 typically represent the computational bottleneck in the grid generation due to their formal $\mathcal{O}(N_{at}^2 N_{grid})$ scaling. However, due to pre-sorting of the list of adjacent nuclei (line 13) and the loop-breaking in lines 15 and 17, only a constant number of nuclei need to be considered in these steps, which reduces the computational cost substantially and ensures an asymptotic linear-scaling behavior.

IV. ILLUSTRATIVE CALCULATIONS

A. Accuracy for benchmark test sets

Here, we compare the accuracy of the conventional partitioning schemes (Becke, SSF) with our modified Becke scheme for three different test sets, namely, the G2 test set (atomization energies of small molecules),[10,11] the S22 × 5 test set (weak interactions of small and medium sized molecules at five different distances),[13] and the L7 test set (weak interactions of seven larger dimers up to 101 atoms),[12] covering a variety of different cases commonly encountered in applications. All calculations are referenced toward a very tight [150/2030]-grid, where we employ the new weighting scheme of the present work.

We decided to use the PBE functional[17] for our benchmarks since it is widely used within the quantum chemistry community, particularly in the form of the PBE hybrid functional (PBEH),[26,27] and in combination with empirical dispersion correction (e.g., the DFT-D3 method by Grimme and co-workers[2]). We have also tested other functional and observed fully analogous behavior. Note that some highly fitted functionals exhibit significantly larger grid errors due to the high spatial variance of the energy density probably due to "overfitting" (see also the discussions in Refs. 28 and 29).

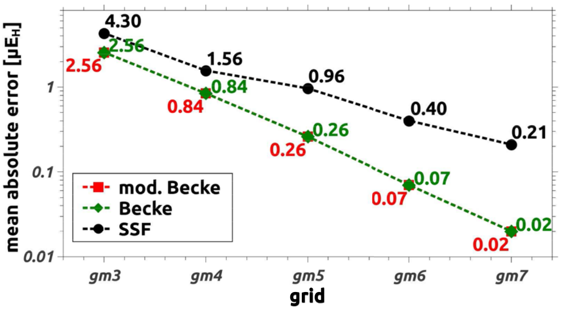


FIG. 7. Grid-induced absolute errors for the G2 test (small molecules) for different grids at the PBE/def2-TZVP level compared to the very tight [150/2030]-reference grid (using the modified partitioning scheme of the present work).

We note that for the small molecules of the G2-test set (Fig. 7), the original and the modified Becke scheme provide virtually identical results since the cutoff-condition of Eq. (7) is essentially never fulfilled for the small interatomic distances present in the G2 test set. As discussed in Sec. III B, this is indeed the desired behavior since a smooth partitioning function provides better results for tightly bound molecules. This explains the inferior accuracy of the sharper SSF weighting scheme in this test set.

However, for weak interaction and especially for stretched dimers, a sharper partitioning function is mandatory to avoid the energy oscillations pictured in Figs. 1, 2, and 4. Therefore, the original Becke scheme provides quite poor results for the weak interactions of both the L7 test set (Fig. 8) and the S22 × 5 test set (Fig. 9). This is particularly problematic since the results of the Becke scheme converge very slowly with the grid size. This observation matches our results for the argon dimer dissociation curve, where the oscillations appear even when employing quite large grids. Although the effect is supposed to disappear for infinitely fine grids in principle, sufficiently tight grids are substantially too large for practical calculations.

The SSF scheme performs much better for weak interaction, especially for larger grids, due to the above-mentioned higher sharpness. However, the modified Becke scheme of

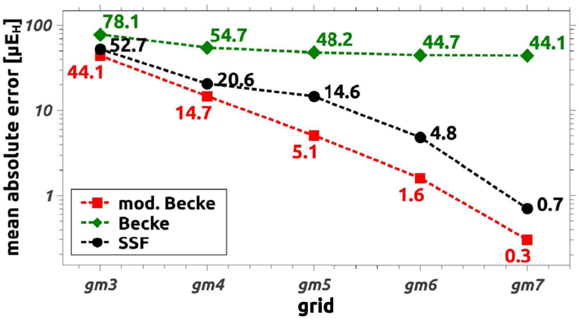


FIG. 8. Grid-induced errors in the interaction energies of the L7 test set (weak interactions of large dimers up to 101 atoms) at the PBE/def2-TZVP level referenced to the very tight [150/2030] grid (using the modified partitioning scheme of the present work). The average interaction energy [QCISD(T)/CBS results from Ref. 12] is given as 29 000 μE_h.

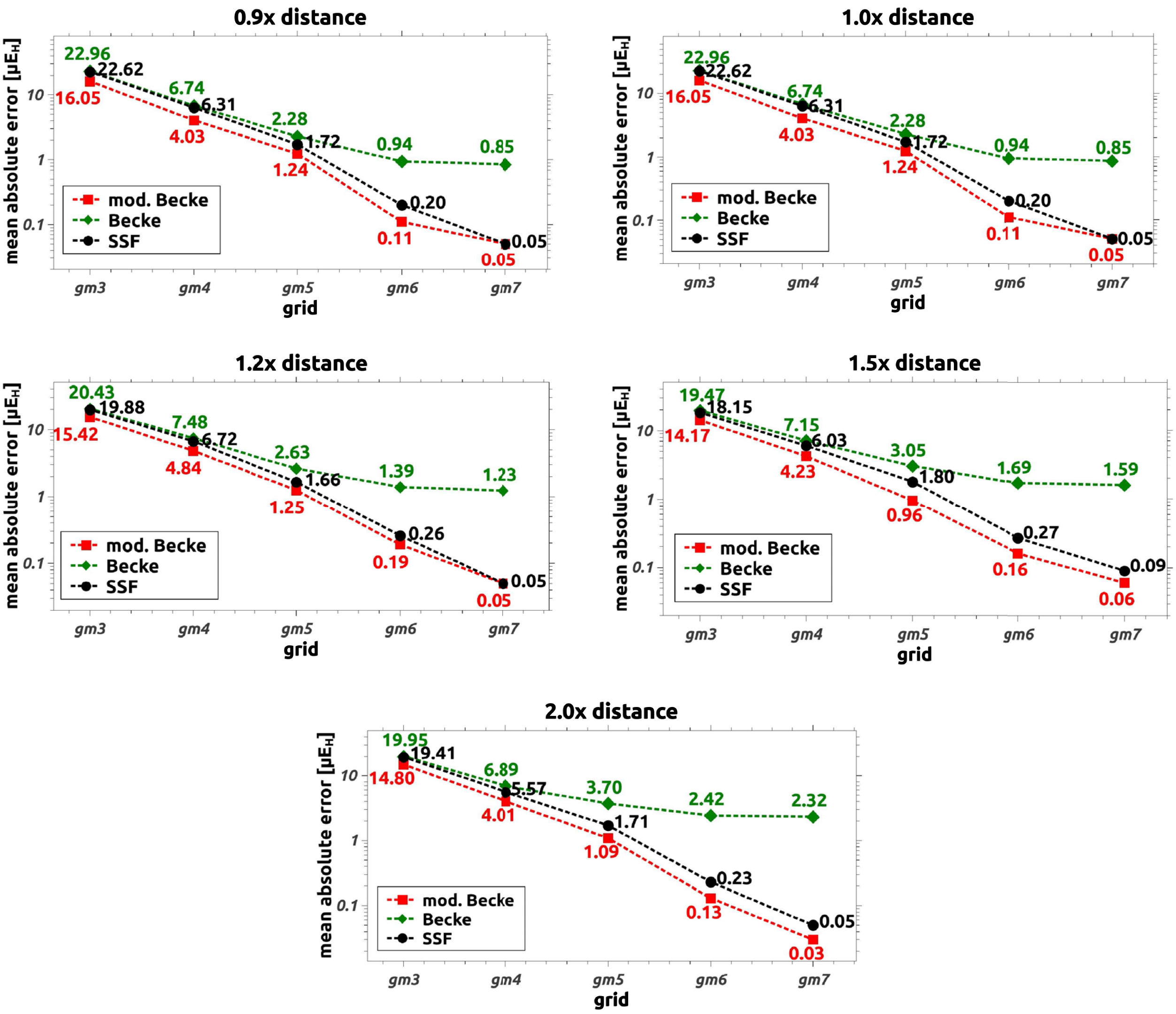


FIG. 9. Grid-induced errors in the interaction energies of the S22 × 5 test set (weak interactions of small and medium sized molecules at five shortened/stretched distances) at the PBE/def2-TZVP level referenced to the very tight [150/2030] grid (using the modified partitioning scheme of the present work). The average interaction energies [CISD(T)/CBS results from Ref. 13] are given as 9470 μE_{h} (0.9 × distance), 11 700 μE_{h} (1.0 × distance), 9200 μE_{h} (1.2 × distance), 4920 μE_{h} (1.5 × distance), 1860 μE_{h} (2.0 × distance).

the present work performs equally well or better than both of the other two schemes in any tested case, and we thus consider it to always be the better choice for DFT calculations. Moreover, the molecular grids from the new weighting scheme contain about 10%–30% less grid points, depending on the packing density of the specific system (higher savings for more densely packed systems), due to the higher locality of the partitioning function, in this way providing proportional savings of 10%–30% in the numeric evaluation of the density functional. Although the purpose of our modified Becke scheme is to improve energy surfaces, these savings in computational cost represent another additional benefit.

Note that even for the coarsest grids, none of the above presented errors are significant (≤1% of the investigated interaction energy) compared to typical functional or basis set errors (commonly ≥10%). However, the less systematic nature of the grid errors, in particular, when leading to oscillating energy surfaces, requires a considerable stricter error control. Since our new scheme yields overall smoother energy surfaces, we expect even more pronounced gains for geometry optimizations, which have been known to require larger grids for robust results and tight convergence. A further investigation of this and related topics (e.g., for vibrational frequencies) will be included in future work.

Overall, when employing the new weighting scheme, the smallest “gm3”-grid (ca. 2500 points/atom for the SCF and 9500 points/atom for the final energy) already provides an acceptable accuracy with the largest error being 102 μE_{h} for the coronene-dimer of the L7-test set, which only corresponds to 0.26% of the absolute interaction energy for this example. This is particularly promising with regard to local-hybrid functionals, where the exact-exchange energy density has to be evaluated for every grid-point, resulting in the numerical (grid-based) integration to represent the computational bottleneck, in contrast to conventional functionals, where the evaluation of the 4-center-2-electron integrals is typically the most time

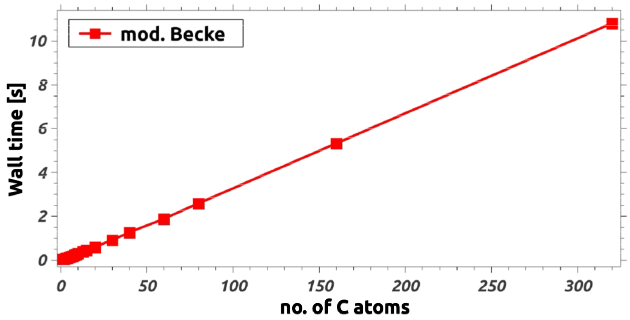


FIG. 10. Scaling behavior of the molecular partitioning scheme of the present work for linear alkanes employing the "g5" grid.

consuming step. Thus, the new weighting scheme is a valuable augmentation to our recently published semi-numerical local-hybrid scheme preLinX.[30]

B. Performance analysis of the grid generation

Having discussed the accuracy of our newly developed modified Becke scheme above, we now assess the efficiency and scaling behavior of our grid generation algorithm, described in Sec. III D. We present the results in Fig. 10 and Table I. All timings are given for the "g5" grid and employing an openMP[31] parallelized setup employing 12 cores (2 × Intel-E5645 at 2.40 GHz), where the parallelization is performed at the level of the parent atoms (line 1 in the pseudocode of Sec. III D). The code was compiled with the GNU compiler collection (GCC)[32] using compiler optimizations (-O3).

TABLE I. Timing comparisons for the molecular partitioning scheme of the present work employing the "g5" grid. The scaling behaviors are given with respect to the respective predecessing molecule.

System	No. of atoms	Time (s)	Scaling
	DNA fragments		
$(DNA)_1$	62	0.7	...
$(DNA)_2$	128	2.3	1.59
$(DNA)_4$	260	5.95	1.27
$(DNA)_8$	524	13.06	1.09
$(DNA)_{16}$	1052	27.57	1.05
	Fullerenes		
C_{60}	60	1.13	...
C_{100}	100	2.17	1.15
C_{180}	180	4.88	1.25
C_{240}	240	6.99	1.07
	Spherical water clusters		
$(H_2O)_{68}$	204	4.39	...
$(H_2O)_{142}$	426	11.83	1.29
$(H_2O)_{285}$	855	27.87	1.17
$(H_2O)_{569}$	1707	62.77	1.13
	LiF cutouts		
$(LiF)_{16}$	32	0.26	...
$(LiF)_{36}$	72	1.02	1.74
$(LiF)_{144}$	288	8.75	2.14
$(LiF)_{256}$	512	22.69	1.46
	Saturated diamond cutouts		
$C_{42}H_{60}$	102	3.44	...
$C_{252}H_{218}$	470	33.67	2.12

The asymptotic linear-scaling of our scheme is clearly demonstrated for both linear alkanes in Fig. 10 and a variety of different systems in Table I, and the grid generation does not represent a computational bottleneck in any of the test cases. We note that, in principle, Becke's partitioning scheme can also be implemented in an asymptotically linear-scaling fashion, by neglecting the contributions of distant nuclei (see, e.g., Ref. 23). However, as discussed in Sec. III D and in contrast to our modified partitioning function of the present work, this leads to an additional cutoff error which needs to be carefully controlled. Indeed, the disadvantageous $\mathcal{O}(N^3)$-scaling of Becke's weighting scheme motivated the development of the weighting scheme of Ref. 9 (SSF), whereas our new partitioning scheme completely removes the computational bottleneck of grid generation without any additional considerations regarding cutoff errors.

V. CONCLUSION AND OUTLOOK

We presented a modification to Becke's molecular partitioning scheme, which completely removes the shortcomings of existing schemes for weakly bound van der Waals complexes and allows for an efficient and linear-scaling generation of the molecular grid. The modified scheme of the present work provides superior accuracy compared to the existing schemes of Becke[4] and Stratmann *et al.*[9] in any test case and, in contrast to the existing schemes, leads to considerably smoother energy surfaces.

This superior cost-performance ratio is particularly significant for local-hybrid functionals[30] where the grid based integration of the exchange-correlation functional represents the major computational bottleneck. Furthermore, due to the linear-scaling algorithm presented in Sec. III D, the computation time for the grid generation can be reduced to virtually negligible cost, which is also promising regarding periodic boundary condition (PBC) calculations.

ACKNOWLEDGMENTS

The authors acknowledge the financial support from the *Deutsche Forschungsgemeinschaft* (DFG) in the *SFB* 749 "Dynamik und Intermediate molekularer Transformationen" and the DFG *cluster of excellence* (EXC114) "Center for Integrative Protein Science Munich" (CIPSM). C.O. further acknowledges financial support as Max-Planck-Fellow at the MPI-FKF Stuttgart.

APPENDIX: SPECIFICATION OF GRIDS USED IN THE PRESENT WORK

The grids employed in the present work are inspired by the approach of Treutler and Ahlrichs[6] as employed in the Turbomole program.[21] The molecular grids are constructed as a combination of the individual atomic grids, adjusted by the partitioning scheme described in Secs. III A and III C. The atomic grids are given as a product of radial (τ) and angular (σ) grids of the form

TABLE II. Optimal atomic radii R for the (M4) quadrature [Eqs. (A2)–(A4)] taken from Ref. 6 for all elements up to Kr. For all heavier elements, $R = 1.0$ is employed.

H	0.8	He	0.9	Li	1.8	Be	1.4	B	1.3	C	1.1	N	0.9	O	0.9	F	0.9	Ne	0.9
				Na	1.4	Mg	1.3	Al	1.3	Si	1.2	P	1.1	S	1.0	Cl	1.0	Ar	1.0
				K	1.5	Ca	1.4	Ga	1.1	Ge	1.0	As	0.9	Se	0.9	Br	0.9	Kr	0.9
Sc	1.3	Ti	1.2	V	1.2	Cr	1.2	Mn	1.2	Fe	1.2	Co	1.2	Ni	1.1	Cu	1.1	Zn	1.1

$$\mathbf{r}_{\text{atomic}} = r_\tau \mathbf{r}_\sigma, \qquad w_{\text{atomic}} = w_\tau w_\sigma, \tag{A1}$$

where r_τ denotes the radius of the radial shell, $\mathbf{r}_\sigma$ denotes the position of the angular grid point on a unit sphere, w_τ denotes the radial grid weight, and w_σ denotes the angular grid weight.

The Lebedev-Laikov grids[3] are employed for the angular grid ($\mathbf{r}_\sigma$, w_σ), and the (M4) quadrature of Ref. 6 is employed for the radial quadrature (r_τ, w_τ). The radial points of the (M4) quadrature are obtained as given in Eqs. (23)–(25) of Ref. 8 as

$$r_\tau = -R\frac{(1+x_\tau)^\alpha}{\ln(2)}\ln\left(\frac{1-x_\tau}{2}\right), \tag{A2}$$

$$\begin{aligned} w_\tau = R^3\frac{\pi}{n_{\text{rad}}+1}\frac{(1+x_\tau)^{3\alpha}}{\ln^3(2)}\Bigg[\sqrt{\frac{1+x_\tau}{1-x_\tau}}\ln^2\left(\frac{1-x_\tau}{2}\right) \\ -\alpha\sqrt{\frac{1-x_\tau}{1+x_\tau}}\ln^3\left(\frac{1-x_\tau}{2}\right)\Bigg], \end{aligned} \tag{A3}$$

$$x_\tau = \cos\left(\pi\frac{\tau}{n_{\text{rad}}+1}\right), \tag{A4}$$

where n_{rad} denotes the total amount of radial points and the optimal values for $\alpha = 0.6$ and the atomic radii R (given in Table II) have been taken from Ref. 6.

Since the electronic structure is typically more isotropic near the nuclei, a lower amount of angular points can be employed for the inner radial shells, saving a substantial amount of grid points with virtually no sacrifice in accuracy. For this purpose and analogous to Ref. 6, we partition the atomic grid into three regions by the conditions

$$\begin{gathered} \text{inner: } \tau \le \frac{n_{\text{rad}}}{3}, \qquad \text{medium: } \frac{n_{\text{rad}}}{3} < \tau \le \frac{n_{\text{rad}}}{2}, \\ \text{outer: } \tau > \frac{n_{\text{rad}}}{2}. \end{gathered} \tag{A5}$$

The number of radial points and the number of angular points n_{ang} for each region (inner/medium/outer) for the grids "g1"–"g7" employed in the present work are given in Table III. Furthermore, to account for their larger sizes, the numbers of radial points n_{rad} are increased by

TABLE III. Specification of the grids employed in the present work.

Grid	n_{rad}	n_{ang} (inner/medium/outer)	Points per C-atom
"g1"	35	14/50/110	2 586
"g2"	40	26/74/194	5 056
"g3"	50	38/110/302	9 564
"g4"	55	50/194/434	15 526
"g5"	60	50/194/590	21 330
"g6"	70	86/302/974	40 838
"g7"	80	110/434/1454	68 770

$$n_{\text{rad, extra}} = \begin{cases} 5 & \text{Li–Ne} \\ 10 & \text{Na–Ar} \\ 20 & \text{K–Kr} \\ 25 & \text{Rb–Xe} \\ 30 & \text{Cs–Og} \end{cases} \tag{A6}$$

for heavier elements.

In analogy to Ref. 6, we employ multi-grids ("gm3" to "gm7"), where the grid-size is reduced by two orders (e.g., "g3" for "gm5") during the SCF and the tighter final grid (e.g., "g5" for "gm5") is only employed for the calculation of the final energy and properties, saving a significant amount of computational effort during the SCF in this way.

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

[2] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, J. Chem. Phys. **132**, 154104 (2010).

[3] V. I. Lebedev and D. N. Laikov, Dokl. Akad. Nauk **366**, 741 (1999).

[4] A. D. Becke, J. Chem. Phys. **88**, 2547 (1988).

[5] C. W. Murray, N. C. Handy, and G. J. Laming, Mol. Phys. **78**, 997 (1993).

[6] O. Treutler and R. Ahlrichs, J. Chem. Phys. **102**, 346 (1995).

[7] M. E. Mura and P. J. Knowles, J. Chem. Phys. **104**, 9848 (1996).

[8] P. M. W. Gill and S.-H. Chien, J. Comput. Chem. **24**, 732 (2003).

[9] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).

[10] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople, J. Chem. Phys. **106**, 1063 (1997).

[11] R. Haunschild and W. Klopper, J. Chem. Phys. **136**, 164102 (2012).

[12] R. Sedlak, T. Janowski, M. Pitonak, J. Rezac, P. Pulay, and P. Hobza, J. Chem. Theory Comput. **9**, 3364 (2013).

[13] L. Grafova, M. Pitonak, J. Rezac, and P. Hobza, J. Chem. Theory Comput. **6**, 2365 (2010).

[14] J. Kussmann and C. Ochsenfeld, J. Chem. Phys. **138**, 134114 (2013).

[15] J. Kussmann and C. Ochsenfeld, J. Chem. Theory Comput. **11**, 918 (2015).

[16] J. Kussmann and C. Ochsenfeld, J. Chem. Theory Comput. **13**, 3153 (2017).

[17] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[18] F. Weigend and R. Ahlrichs, Phys. Chem. Chem. Phys. **7**, 3297 (2005).

[19] A. Bondi, J. Phys. Chem. **68**, 441 (1964).

[20] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kus, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. Woodcock III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. DiStasio, Jr., H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. D. Hanson-Heine, P. H. P. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Lao, A. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, E. Neuscamman, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ,

S. M. Sharada, S. Sharma, D. W. Small, A. Sodt, T. Stein, D. Stuck, Y.-C. Su, A. J. W. Thom, T. Tsuchimochi, V. Vanovschi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, J. Yang, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhao, B. R. Brooks, G. K. L. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. W. Gill, and M. Head-Gordon, Mol. Phys. **113**, 184 (2015), Q-Chem 4.4.1.

[21] R. Ahlrichs, M. Baer, M. Haeser, H. Horn, and C. Koelmel, Chem. Phys. Lett. **162**, 165 (1989), Turbomole V7.0.2.

[22] F. Neese, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **2**, 73 (2012), Orca 4.0.0.2.

[23] M. D. Towler, A. Zupan, and M. Causa, Comput. Phys. Commun. **98**, 181 (1996).

[24] K. N. Kudin and G. E. Scuseria, Phys. Rev. B **61**, 16440 (2000).

[25] A. M. Burow and M. Sierka, J. Chem. Theory Comput. **7**, 3097 (2011).

[26] C. Adamo and V. Barone, J. Chem. Phys. **110**, 6158 (1999).

[27] M. Ernzerhof and G. E. Scuseria, J. Chem. Phys. **110**, 5029 (1999).

[28] S. Dasgupta and J. M. Herbert, J. Comput. Chem. **38**, 869 (2017).

[29] N. Mardirossian and M. Head-Gordon, Mol. Phys. **115**, 2315 (2017).

[30] H. Laqua, J. Kussmann, and C. Ochsenfeld, J. Chem. Theory Comput. **14**, 3451 (2018).

[31] See http://www.openmp.org for OpenMP library, version 4.0.

[32] See http:/gcc.gnu.org for GNU compiler collection, version 4.8.5.

3.7 Publication VII: Highly Efficient Resolution-of-Identity Density Functional Theory Calculations on Central and Graphics Processing Units

J. Kussmann, H. Laqua, C. Ochsenfeld

J. Chem. Theory Comput. **17**, 1512 (2021).

Abstract

We present an efficient method to evaluate Coulomb potential matrices using the resolution of identity approximation and semilocal exchange-correlation potentials on central (CPU) and graphics processing units (GPU). The new GPU-based RI-algorithm shows a high performance and ensures the favorable scaling with increasing basis set size as the conventional CPU-based method. Furthermore, our method is based on the J-engine algorithm [White, Head-Gordon, *J. Chem. Phys.* **1996**, *7*, 2620], which allows for further optimizations that also provide a significant improvement of the corresponding CPU-based algorithm. Due to the increased performance for the Coulomb evaluation, the calculation of the exchange-correlation potential of density functional theory on CPUs quickly becomes a bottleneck to the overall computational time. Hence, we also present a GPU-based algorithm to evaluate the exchange-correlation terms, which results in an overall high-performance method for density functional calculations. The algorithms to evaluate the potential and nuclear derivative terms are discussed, and their performance on CPUs and GPUs is demonstrated for illustrative calculations.

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

Highly Efficient Resolution-of-Identity Density Functional Theory Calculations on Central and Graphics Processing Units

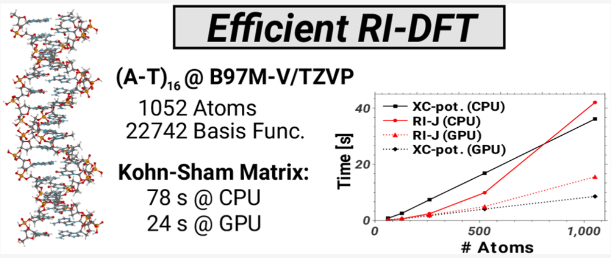
Jörg Kussmann,[†] Henryk Laqua,[†] and Christian Ochsenfeld*

Cite This: *J. Chem. Theory Comput.* 2021, 17, 1512−1521

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: We present an efficient method to evaluate Coulomb potential matrices using the resolution of identity approximation and semilocal exchange-correlation potentials on central (CPU) and graphics processing units (GPU). The new GPU-based RI-algorithm shows a high performance and ensures the favorable scaling with increasing basis set size as the conventional CPU-based method. Furthermore, our method is based on the J-engine algorithm [White, Head-Gordon, *J. Chem. Phys.* **1996**, 7, 2620], which allows for further optimizations that also provide a significant improvement of the corresponding CPU-based algorithm. Due to the increased performance for the Coulomb evaluation, the calculation of the exchange-correlation potential of density functional theory on CPUs quickly becomes a bottleneck to the overall computational time. Hence, we also present a GPU-based algorithm to evaluate the exchange-correlation terms, which results in an overall high-performance method for density functional calculations. The algorithms to evaluate the potential and nuclear derivative terms are discussed, and their performance on CPUs and GPUs is demonstrated for illustrative calculations.



1. INTRODUCTION

In the past decade, it has been shown that graphic processing units (GPUs) can significantly accelerate ab initio calculations.[1−14] While first implementations for GPUs were limited to smaller basis sets due to tight memory limitations, recent works extended the applicability of GPUs to calculations with basis sets containing higher *l*-quantum numbers for Coulomb-type terms[9] as well as exact-exchange terms using a seminumerical approach (sn-LinK[11]). The evaluation of exact-exchange terms in general remains the bottleneck of Hartree−Fock or hybrid density functional theory (DFT) calculations. However, the sn-LinK method[11] shows a favorable scaling with respect to increasing basis set sizes, while the GPU-based Coulomb calculations scale as N^4 with the size of the underlying basis set. To overcome this unfavorable scaling behavior, the resolution-of-identity approximation[15−19] (RI) is widely applied and provided by most program packages.

In this work, we present a J-engine based, integral-direct RI method to evaluate the Coulomb potential and the nuclear derivate of the Coulomb energy. Here, it should be stressed that Neese proposed a CPU-based improved RI Coulomb method (IRI)[20] that is also based on the J-engine algorithm. In contrast to the latter, however, we propose improvements to the 3-center (3c) and 2-center (2c) integral evaluation steps to reduce the computational workload. Our method reduces not only the number of floating-point operations (FLOPs), which additionally results in a better performance on CPUs, but also the amount of required local memory in GPU kernels.

As the performance of the Coulomb evaluation step is strongly improved, the evaluation of the semilocal exchange-correlation terms (XC) quickly becomes a bottleneck in GPU-based DFT calculations. Thus, we also discuss efficient, linearly scaling algorithms on both CPUs and GPUs for the computation of XC potentials and nuclear derivatives. With the proposed methods at hand, we show the significant performance improvements of RI-DFT calculations using large basis sets on both CPUs and GPUs. The scaling and performance of our methods are demonstrated at first illustrative calculations.

The theory and algorithmic considerations of our proposed methods are discussed in section 2. The performance analysis at first illustrative calculations is discussed in section 3.

2. THEORY

In this section, we present algorithms to evaluate the RI-based Coulomb and exchange-correlation contributions to the

Received: December 3, 2020
Published: February 22, 2021

Table 1. FLOP-Count of Selected Coulomb-type Integral Kernels for Regular and RI-based Evaluation[a]

l_{bra}	l_{ket}	regular $(\mu\nu|\lambda\sigma)$	RI (step 1) [eqs 2, 14] $(P|\lambda\sigma)$		RI (step 3) [eqs 4, 16] $(\mu\nu|Q)$		RI (step5) [eq 18] $(P|Q)$	
0	1	49	49	100.0%	46	93.9%	47	95.9%
0	2	79	79	100.0%	69	87.3%	70	88.6%
0	3	131	131	100.0%	107	81.7%	108	82.4%
0	4	214	214	100.0%	167	78.0%	168	78.5%
1	0	52	49	94.2%	48	92.3%	49	94.2%
2	0	88	76	86.4%	78	88.6%	76	86.4%
3	0	150	120	80.0%	130	86.7%	120	80.0%
4	0	302	189	62.6%	213	70.5%	189	62.6%
5	0	524	288	55.0%	336	64.1%	288	55.0%
4	4	3673	2290	62.3%	2268	61.7%	1578	43.0%
5	6	13676	7647	55.9%				
6	4	9834			5102	51.9%		

[a]The percentage is given with respect to the regular integral kernel.

Kohn−Sham matrix, as well as the corresponding nuclear derivatives. Specific considerations for an efficient execution of the algorithm on GPUs are discussed.

2.1. J-Engine Based Resolution-of-Identity Coulomb Potential Evaluation.

The Coulomb potential using the RI approximation is given as

$$J_{\mu\nu} = \sum_{\lambda\sigma} \rho_{\lambda\sigma}(\mu\nu|\lambda\sigma) \approx \sum_{\lambda\sigma PQ} \rho_{\lambda\sigma}(\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma) \tag{1}$$

where the integral-direct algorithm consists of three steps:

Step 1:

$$B_P = \sum_{\lambda\sigma} (P|\lambda\sigma)\rho_{\lambda\sigma} \tag{2}$$

Step 2:

$$B'_Q = \sum_P (Q|P)^{-1} B_P \tag{3}$$

Step 3:

$$J_{\mu\nu} = \sum_Q (\mu\nu|Q)B'_Q \tag{4}$$

with μ, ν, λ, and σ representing regular and P and Q auxiliary basis functions, respectively, and $\rho_{\lambda\sigma}$ being an element of the matrix representation of the one-electron density.

Our algorithm improves the first and third step of the calculation, while we use the Coulomb fitting method by Mintmire and Dunlap[21] in step 2. By default, the explicit inversion of the Coulomb metric is avoided by directly solving $(Q|P)x = B_P$ to obtain $x = (Q|P)^{-1}B_P = B'_Q$ for step 2, which only requires a Cholesky decomposition of $(Q|P)$.

For the two rate-determining steps that involve the evaluation of 3-center integrals, we propose to use the J-engine,[22−24] which is based on the McMurchie−Davidson algorithm[25] and delivers a high-performing algorithm on both CPUs and GPUs. In order to exploit all benefits of this approach, in particular on GPUs, different strategies have to be followed for steps 1 and 3.

2.1.1. Step 1: Evaluation of B_P.

Following the notation in ref 25, the first step in eq 2 is given by

$$\begin{aligned} J_P &= \sum_{\lambda\sigma} \rho_{\lambda\sigma}(P|\lambda\sigma) \\ &= \sum_{\lambda\sigma} \rho_{\lambda\sigma} \sum_{k=NML} D_P^k \sum_{l=N'M'L'} D_{\lambda\sigma}^l (-1)^{N+M+L} R^{kl}_{(N+N')(M+M')(L+L')} \\ &= \sum_{k=NML} D_P^k \sum_{l=N'M'L'} \rho^l (-1)^{N+M+L} R^{kl}_{(N+N')(M+M')(L+L')} \\ &= \sum_k D_P^k J_k \end{aligned} \tag{5}$$

with the Hermite factors D representing the product of expansion factors for the angular momentum information in the basis of Hermite polynomials (see eq 2.27 in ref 25); R^{kl} are the auxiliary integrals (section 4 in ref 25), and k and l represent combined indices for the angular information on the bra and ket terms, respectively.

The central point of the J-engine algorithm[24] is the preprocessing of the ket data to yield the factors ρ^l. For GPU-based calculations, the preprocessing step is done on CPUs, while the evaluation of the auxiliary integrals R^{kl} and their contraction with the Hermite factors are done on GPUs. The final postprocessing step ($J_P = \sum_k D_P^k J_k$) is done on CPUs again:

@CPU:

$$\rho^l = \sum_{\lambda\sigma} \rho_{\lambda\sigma} D_{\lambda\sigma}^l \tag{6}$$

@GPU:

$$J_k = \sum_{l=N'M'L'} \rho^l (-1)^{N+M+L} R^{kl}_{(N+N')(M+M')(L+L')} \tag{7}$$

@CPU:

$$J_P = \sum_k D_P^k J_k \tag{8}$$

where the CPU-based algorithm follows the same scheme.

Note that the Hermite factors $D_{\lambda\sigma}^l$ also shift angular information from the shell-pair center R_Q to the centers of the individual basis functions R_λ and R_σ, respectively. In eq 5, however, the bra side represents a single auxiliary function instead of a shell-pair, so that many of the corresponding factors D_P^k are zero, which in turn also reduces the number of required elements J_p. While for the regular elements $J_{\mu\nu}$ all integrals J_k^{lqn} with lqn = $\text{lqn}_{\mu+\nu}$, $\text{lqn}_{\mu+\nu} - 1$, ..., 0 have to be evaluated, here, we only have to evaluate every second batch, that is, J_k^{lqn} with lqn = lqn_P, $\text{lqn}_P - 2$, etc. For an h-shell (lqn =

5), for example, one only has to evaluate 34 out of the regular 56 integrals. Using a recursive algorithm in the integral generator, one can further reduce the number of intermediate integrals R^{kl} and therefore significantly reduce the overall number of floating point operations (FLOPs). Table 1 shows this reduction of FLOPs for a number of selected integral kernels for all types of 3c- and 2c-integrals in comparison to the regular, non-RI kernels. While both CPU- and GPU-based algorithms profit from the reduced number of FLOPs, the reduced number of integrals J_k also significantly reduces the amount of required local memory, especially shared memory, on GPUs. Considering the strict local memory limitations on GPUs,[9,10] this reduced shared memory requirement leads to less memory spills and no splitting of shell-pairs for functions with higher l-quantum numbers.

To ensure an efficient use of GPUs, a further optimization regarding the arrangement of computing threads is required. For the conventional non-RI algorithm, it has been shown that the use of 8 × 8 thread blocks is most efficient.[4,9] For the processing of eq 7, the number of auxiliary bra terms is significantly smaller than the number of regular shell-pair ket terms. Therefore, we use 1 × 64 thread blocks, that is, the parallelization within a block is over the ket data only. Finally, we can also reduce the size of the output vector J_k by executing the final transformation $\sum_k D_P^k J_k$ (eq 8) on GPUs. As a result, we only have to store $(\mathrm{lqn_{bra}} + 1)(\mathrm{lqn_{bra}} + 2)/2$ elements of J_P for a primitive shell, while the postprocessing step on CPUs is reduced to a simple sum over the primitives.

2.1.2. Step 3: Evaluation of $J_{\mu\nu}$.

The equations to evaluate step 3 in eq 4 are

$$\begin{aligned} J_{\mu\nu} &= \sum_Q B'_Q(\mu\nu|Q) \\ &= \sum_Q B'_Q \sum_{k=NML} D_{\mu\nu}^k \sum_{l=N'M'L'} D_Q^l (-1)^{N+M+L} R^{kl}_{(N+N')(M+M')(L+L')} \\ &= \sum_{k=NML} D_{\mu\nu}^k \sum_{l=N'M'L'} \rho^l (-1)^{N+M+L} R^{kl}_{(N+N')(M+M')(L+L')} \\ &= \sum_k D_{\mu\nu}^k J_k \end{aligned} \tag{9}$$

with $\rho^l = \sum_Q B'_Q D_Q^l$. The J-engine scheme is similar to eqs 6–8. As for the evaluation of the 3-center integrals in part 1, the overall number of FLOPs can be reduced. While we need all integrals J_p to form the final Coulomb integrals $J_{\mu\nu}$, many elements D_Q^l, and therefore ρ^l, are zero, which can be exploited again by reducing the number of intermediate integrals R^{kl} as shown in Table 1.

In contrast to step 1, however, the postprocessing is done the conventional way, that is, the contraction $\sum_k D_{\mu\nu}^k J_k$ is done on CPUs and all integrals J_p are required. For the GPU-based algorithm, we again have to consider the fact of a far larger number of shell-pairs $\mu\nu$ as compared to the number of auxiliary functions. Therefore, we found it most efficient to employ 64 × 1 thread blocks, that is, a parallelization over the bra data only within a block. This has the further advantage that we can completely avoid the use of shared memory in the GPU kernels, which again allows us to write kernels for shells with high l-quantum numbers without splitting the shell-pair data into multiple kernels.

2.2. J-Engine Based Resolution-of-Identity Coulomb Nuclear Gradients.

For the evaluation of nuclear gradients of the Coulomb energy, the derivatives of the auxiliary basis functions also have to be considered since the employed basis sets are usually far from complete. The Coulomb integral contributions to the nuclear derivatives are given as

$$\begin{aligned} \frac{\partial E_J}{\partial A_x} &= \sum_{\mu\nu} \rho_{\mu\nu} \sum_{\lambda\sigma} \rho_{\lambda\sigma} [([\mu\nu]^{A_x}|\lambda\sigma) + (\mu\nu|[\lambda\sigma]^{A_x})] \\ &= 2\sum_{\mu\nu} \rho_{\mu\nu} \sum_{\lambda\sigma} \rho_{\lambda\sigma} ([\mu\nu]^{A_x}|\lambda\sigma) \end{aligned} \tag{10}$$

where A_x represent the x-coordinate of nucleus A. With the RI approximation, the gradient is

$$\begin{aligned} \frac{\partial E_J}{\partial A_x} &\approx \sum_{\mu\nu} \rho_{\mu\nu} \sum_{\lambda\sigma} \rho_{\lambda\sigma} \sum_{PQ} [(\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma)]^{A_x} \\ &= \sum_{\mu\nu} \rho_{\mu\nu} \sum_{\lambda\sigma} \rho_{\lambda\sigma} \sum_{PQ} [2([\mu\nu]^{A_x}|P)(P|Q)^{-1}(Q|\lambda\sigma) \\ &\quad + 2(\mu\nu|P^{A_x})(P|Q)^{-1}(Q|\lambda\sigma) + (\mu\nu|P)[(P|Q)^{-1}]^{A_x} \\ &\quad (Q|\lambda\sigma)] \end{aligned} \tag{11}$$

The derivative of the inverse Coulomb metric can be evaluated from $\sum_R (P|R)(R|Q)^{-1} = \delta_{PQ}$ as

$$[(P|Q)^{-1}]^{A_x} = -(P|R)^{-1}[(R|S)]^{A_x}(S|Q)^{-1} \tag{12}$$

leading to

$$\begin{aligned} \frac{\partial E_J}{\partial A_x} &\approx \sum_{\mu\nu} \rho_{\mu\nu} \sum_{\lambda\sigma} \rho_{\lambda\sigma} \sum_{PQ} [2([\mu\nu]^{A_x}|P)(P|Q)^{-1}(Q|\lambda\sigma) \\ &\quad + 2(\mu\nu|P^{A_x})(P|Q)^{-1}(Q|\lambda\sigma) \\ &\quad - \sum_{RS} (\mu\nu|P)(P|R)^{-1}[(R|S)]^{A_x}(S|Q)^{-1}(Q|\lambda\sigma)] \end{aligned} \tag{13}$$

Therefore, the resulting algorithm is given as

Step 1:

$$B_P = \sum_{\lambda\sigma} (P|\lambda\sigma)\rho_{\lambda\sigma} \tag{14}$$

Step 2:

$$B'_Q = \sum_P (Q|P)^{-1} B_P \tag{15}$$

Step 3:

$$\frac{\partial E_J}{\partial A_x} = 2\sum_{\mu\nu} \rho_{\mu\nu} \sum_Q ([\mu\nu]^{A_x}|Q) B'_Q \tag{16}$$

Step 4:

$$\frac{\partial E_J}{\partial A_x} += 2\sum_{\mu\nu} \rho_{\mu\nu} \sum_Q (\mu\nu|Q^{A_x}) B'_Q \tag{17}$$

Step 5:

$$\frac{\partial E_J}{\partial A_x} += \sum_{PQ} B'_P [(P|Q)]^{A_x} B'_Q \tag{18}$$

While steps 1 and 2 are equivalent to the Coulomb potential algorithm, step 3 can be evaluated with the same integral kernel as in the potential case by evaluating the integrals J_k up to $\mathrm{lqn_{bra}} + 1$ and employing a postprocessing step on CPUs to evaluate the gradient contributions.[5] Step 4 can be processed in a similar fashion by employing the integral kernel used for step 1.

The final contribution in step 5 could also be evaluated with the kernels used for step 1. However, since both bra and ket terms stem from auxiliary basis functions, we can further reduce the FLOP count by exploiting the sparsity of both sets $\{D^k\}$ and $\{D^l\}$ (see Table 1):

@CPU:

$$\rho^l = \sum_Q B'_Q D^l_Q \tag{20}$$

@GPU:

$$J_k = \sum_{l=N'M'L'} \rho^l (-1)^{N+M+L} R^{kl}_{(N+N')(M+M')(L+L')} \tag{21}$$

@CPU:

$$\frac{\partial E_J}{\partial A_x} += \sum_{Bk} B'_P [D^k]_P^{A_x} J_k \tag{22}$$

Note that the execution of step 5 is significantly faster than the other steps since both bra and ket data are constructed from auxiliary functions.

It should be mentioned that all integral as well as pre- and postprocessing kernels are automatically generated, that is, no further manual optimizations of specific compute kernels were undertaken. The generator recursively determines the significant intermediates based on the required final quantities necessary for the objective integral type.

2.3. Evaluation of Exchange-Correlation Potential and Nuclear Forces on GPUs.

While an efficient CPU-based algorithm for the evaluation of the DFT exchange-correlation potential (XC) can compete with the regular GPU-based Coulomb evaluation, especially for larger basis sets, the RI-Coulomb evaluation on GPUs exposes the XC evaluation as the new bottleneck in the calculation. Thus, we discuss in this section our GPU-based algorithm to evaluate the XC potential and nuclear gradients of the XC energy.

For simplicity, we restrict ourselves here to the local density approximation (LDA) and provide the theory for the generalized gradient approximation (GGA) and the meta-generalized gradient approximation (mGGA) in the appendix, since their implementation is analogous. Moreover, we only discuss restricted closed shell calculations and note that the extension to open-shell calculations simply requires the evaluation for both spin channels.

2.3.1. Exchange-Correlation Potential. The LDA exchange-correlation (XC) energy is defined as the 3-dimensional integral

$$E_{\mathrm{XC}} = \int \varepsilon_{\mathrm{xc}}(\rho(\mathbf{r}))\, \mathrm{d}\mathbf{r} \tag{23}$$

where the XC energy density, $\varepsilon_{\mathrm{xc}}$, only depends on the electron density $\rho(\mathbf{r})$ at position $\mathbf{r}$. In practice, eq 23 is evaluated on a numerical integration grid[26,27] with grid points $\mathbf{r}_g$ and corresponding weights w_g:

$$E_{\mathrm{XC}} = \sum_g w_g \varepsilon_{\mathrm{xc}}(\rho(\mathbf{r}_g)) \tag{24}$$

The electron density $\rho(\mathbf{r}_g)$ can be obtained from the 1-particle reduced density matrix in the atomic orbital basis $\rho_{\mu\nu}$ as

$$\rho(\mathbf{r}_g) = \sum_{\mu\nu} \rho_{\mu\nu} \chi_\mu(\mathbf{r}_g) \chi_\nu(\mathbf{r}_g) \tag{25}$$

where $\chi_\mu(\mathbf{r}_g)$ is the value of the basis function χ_μ evaluated at the grid point $\mathbf{r}_g$. Equation 25 is evaluated in two steps:

$$F_{\nu g} = \sum_\mu \rho_{\mu\nu} \chi_\mu(\mathbf{r}_g) \tag{26}$$

$$\rho(\mathbf{r}_g) = \sum_\nu F_{\nu g} \chi_\nu(\mathbf{r}_g) \tag{27}$$

The evaluation of eq 26 formally scales as $\mathcal{O}(\mathrm{n}_{\mathrm{grid}} \mathrm{n}_{\mathrm{bas}}{}^2)$ and therefore typically represents the most time-consuming step. Since it can be evaluated by a single matrix−matrix multiplication, highly optimized linear-algebra libraries can be employed for this step, providing optimal performance without much implementation effort.

Within the Kohn−Sham DFT method, the XC potential matrix $\mathbf{V}^{\mathrm{XC}}$, which can be defined as

$$\mathbf{V}^{\mathrm{XC}}_{\mu\nu} = \frac{\partial E_{\mathrm{XC}}}{\partial \rho_{\mu\nu}} \tag{28}$$

is also required. Using the derivative chain rule, a grid-based expression for the LDA XC potential matrix is readily derived as

$$\mathbf{V}^{\mathrm{XC}}_{\mu\nu} = \sum_g w_g \frac{\partial \varepsilon_{\mathrm{xc}}}{\partial \rho}(\mathbf{r}_g) \chi_\mu(\mathbf{r}_g) \chi_\nu(\mathbf{r}_g) \tag{29}$$

which is also evaluated in two steps:

$$G_{\nu g} = w_g \frac{\partial \varepsilon_{\mathrm{xc}}}{\partial \rho}(\mathrm{r}_g) \chi_\nu(\mathrm{r}_g) \tag{30}$$

$$\mathbf{V}^{\mathrm{XC}}_{\mu\nu} = \sum_g \chi_\mu(\mathrm{r}_g) G_{\nu g} \tag{31}$$

Here, the slowest step is the evaluation of eq 31 with an identical computational cost as eq 26, which is again implemented as a matrix−matrix multiplication.

The algorithms for CPUs and GPUs are outlined in Algorithm 1 and 2, respectively, with the scaling for each

Algorithm 1 LDA DFT XC-potential implementation @ CPU.

```
for all grid-batches do                                                     ▷ multi-core parallel
    compute basis function values $\chi_\mu(\mathbf{r}_g)$                      ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    get batch-local density matrix $\rho_{\mu\nu}$                              ▷ $\mathcal{O}(\mathrm{n}^2_{\{\mu\}})$
    compute $F_{\nu g}$ (eq. 26)                                               ▷ GEMM,$\mathcal{O}(\mathrm{n}^2_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    compute $\rho(\mathbf{r}_g)$ (eq. 27)                                       ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    evaluate functional: $\{\rho(\mathbf{r}_g)\} \to \{\varepsilon(\mathbf{r}_g), \frac{\partial\varepsilon(\mathbf{r}_g)}{\partial\rho(\mathbf{r}_g)}\}$   ▷ $\mathcal{O}(\mathrm{n}_\mathrm{g})$
    compute $G_{\nu g}$ (eq. 30)                                               ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    compute batch-local exchange matrix $V_{\mu\nu}$ (eq. 31)                  ▷ GEMM,$\mathcal{O}(\mathrm{n}^2_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    add batch-local exchange matrix to global exchange matrix                  ▷ $\mathcal{O}(\mathrm{n}^2_{\{\mu\}})$
end for
```

Algorithm 2 LDA DFT XC-potential implementation @ GPU.

```
pre-allocate GPU memory
copy $P_{\mu\nu}$ to GPUs
for all grid-batches do                                                     ▷ multi-GPU/multi-stream parallel
    compute basis function values $\chi_\mu(\mathbf{r}_g)$                      ▷ GPU, $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    get batch-local density matrix $\rho_{\mu\nu}$                              ▷ GPU, $\mathcal{O}(\mathrm{n}^2_{\{\mu\}})$
    compute $F_{\nu g}$ (eq. 26)                                               ▷ GEMM@GPU,$\mathcal{O}(\mathrm{n}^2_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    compute $\rho(\mathbf{r}_g)$ (eq. 27)                                       ▷ GPU,$\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    evaluate functional: $\{\rho(\mathbf{r}_g)\} \to \{\varepsilon(\mathbf{r}_g), \frac{\partial\varepsilon(\mathbf{r}_g)}{\partial\rho(\mathbf{r}_g)}\}$   ▷ CPU,$\mathcal{O}(\mathrm{n}_\mathrm{g})$
    compute $G_{\nu g}$ (eq. 30)                                               ▷ GPU,$\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    compute batch-local exchange matrix $K_{\mu\nu}$ (eq. 31)                  ▷ GEMM@GPU,$\mathcal{O}(\mathrm{n}^2_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    add batch-local exchange matrix to global exchange matrix                  ▷ GPU,$\mathcal{O}(\mathrm{n}^2_{\{\mu\}})$
end for
copy $K_{\mu\nu}$ to CPU
```

https://dx.doi.org/10.1021/acs.jctc.0c01252

step in terms of the number of significant basis functions $n_{\{\mu\}}$ for a given batch of grid points n_g. Note that the overall scaling behavior is asymptotically linear, that is, for even-sized grid batches (constant number of grid points), we have an on-average constant number of significant basis functions and a linearly increasing number of batches.

All rate-determining steps that scale cubically $[O(\mathrm{n}_{\{\mu\}}{}^2\mathrm{n}_g)]$ with the constant dimension $(n_{\{\mu\}}{}^2, n_g)$ of the batch, are implemented using efficient matrix—matrix multiplications (GEMM) employing highly optimized linear algebra libraries (Intel Math Kernel Library (MKL) on CPUs,[28] cuBLAS,[29] or clBLAST[30]) for GPUs using CUDA or OpenCL, respectively.

The evaluation of the XC functional is in both cases done on CPU using the libXC library.[31] However, this step scales with $O(\mathrm{n}_g)$ only and is therefore computationally far less significant. Furthermore, only the density values (ρ, σ, τ) and the resulting functional value derivatives for the given batch of grid points have to be copied between the host and the GPU device, which amounts to a few megabytes at most. Thus, the impact of the data transfer is insignificant for the overall computational effort.

The parallelization is done in a straightforward fashion over even-sized batches of adjacent grid points using Hilbert curves as described in ref 32. For the GPU-based algorithm, the density $\boldsymbol{\rho}$ and XC potential matrices $\mathbf{V}^{\mathrm{XC}}$ are stored, where each GPU stream holds a private potential matrix to prevent data races. Note that the XC potential is local, that is, only a vector of the elements from significantly overlapping shell-pairs have to be stored. Considering that we use local Gaussian-type basis functions, the size of these vectors scales linearly with increasing system size and are significantly smaller than $N_{\mathrm{BF}}{}^2$ (e.g., for $(\mathrm{AT})_{16}$/TZVP, 108 MB instead of 3945 MB).

The algorithm for GGA and mGGA functionals is analogous to the scheme outlined in Algorithms 1 and 2 with additional terms arising from the gradient and Laplacian of the one-particle density ρ, which can be evaluated with the same subroutines. These additional steps increase the overall computational effort from LDA to mGGA. As mentioned before, the cubically scaling steps, that is, matrix multiplications, are rate-determining, so the performance for the different functional types are closely linked to the number of matrix multiplications; see Table 2.

Table 2. Comparison of the Number of Matrix Multiplications for LDA, GGA, and MGGA Functionals for Evaluation of the Energy, the XC Potential, and the XC Gradient

functional type	energy	XC potential	XC gradient
LDA	1	2	1
GGA	1	2	4
MGGA	4	8	4

The additional terms required for GGA and mGGA functionals can be found in the Appendix.

2.3.2. Nuclear Gradient of Exchange-Correlation Energy. The LDA contribution to nuclear forces is given as

$$E_{\mathrm{XC}}^{A_x} = \sum_g w_g \frac{\partial \varepsilon_{\mathrm{xc}}}{\partial \rho}(\mathrm{r}_g)\rho^{A_x}(\mathrm{r}_g) \tag{32}$$

where

$$\rho^{A_x}(\mathrm{r}_g) = 2\sum_{\mu\nu} \rho_{\mu\nu}\chi_\mu(\mathrm{r}_g)\chi_\nu^{A_x}(\mathrm{r}_g) \tag{33}$$

is the infinitesimal change of $\rho(\mathbf{r}_g)$ when changing a nuclear coordinate A_x. Note that the contribution from the relaxation of the electron density $(\rho_{\mu\nu}^{A_x})$ is considered in the Pulay term.[33] The quantity ρ^{A_x} $(\mathbf{r}_g)$ is evaluated from the intermediate quantity $F_{\nu g}$ without significant overhead as

$$\rho^{A_x}(\mathrm{r}_g) = 2\sum_{\nu} F_{\nu g}\chi_\nu^{A_x}(\mathrm{r}_g) \tag{34}$$

The perturbed basis functions $\chi_\nu^{A_x}(\mathbf{r}_g)$ can be obtained from the gradient of the basis function as

$$\chi_\nu^{A_x}(\mathbf{r}_g) = \begin{cases} -\dfrac{\partial \chi_\mu}{\partial x}(\mathrm{r}_g) & \text{if } \mu \in A \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

The algorithm for the CPU-implementation is shown in Algorithm 3, while the corresponding GPU-based algorithm is straightforward; that is, similar to the potential term the corresponding steps are evaluated on GPUs.

Algorithm 3 LDA DFT XC-forces implementation (CPU).

```
for all grid-batches do                                              ▷ multi-core parallel
    compute basis function values $\{\chi_\mu(\mathbf{r}_g), \nabla\chi_\mu(\mathbf{r}_g)\}$        ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    get batch-local density matrix $\rho_{\mu\nu}$                        ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}^2)$
    compute $F_{\nu g}$ (eq. 26)                                     ▷ GEMM,$\mathcal{O}(\mathrm{n}_{\{\mu\}}^2\mathrm{n}_\mathrm{g})$
    compute $\rho(\mathbf{r}_g)$ (eq. 27)                                  ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    compute $\rho^{A_x}(\mathbf{r}_g)$ (eq. 34)                               ▷ $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_\mathrm{g})$
    evaluate functional: $\{\rho(\mathbf{r}_g)\} \rightarrow \{\varepsilon(\mathbf{r}_g), \frac{\partial\varepsilon(\mathbf{r}_g)}{\partial\rho(\mathbf{r}_g)}\}$   ▷ $\mathcal{O}(\mathrm{n}_\mathrm{g})$
    compute $E_{XC}^{A_x}$ (eq. 32)                                   ▷ GEMV,$\mathcal{O}(\mathrm{n}_\mathrm{g}^2)$
end for
```

3. COMPUTATIONAL SETUP

All presented methods have been implemented in our FermiONs++ program package.[9,34,35] The binary has been compiled with the Intel Compiler 19.1[28] (flags: -Ofast -march=native [skylake-avx512]), CUDA kernels with CUDA 10.1[29] (flags: -O3, -use_fast_math), and OpenCL kernels with ROCm-3.8.0[36] (flags: -O3 -cl-mad-enable -cl-finite-math-only -cl-no-signed-zeros). CPU calculations were executed on 2 Intel Xeon Silver 4216 (32 cores/64 threads; 2.1 GHz); the GPU-calculations were executed on either a single NVIDIA GV100 or up to four AMD Radeon VII cards.

For the numerical evaluation of XC terms, we employ multigrids defined in ref 27, that is, a smaller grid within the SCF optimization and a larger grid for the final energy evaluation, generated with the modified Becke weighting scheme.[27] If not stated otherwise, all calculations employ the B97M-V functional.[37] The extent threshold to determine significant basis functions for a given grid batch[38] is set to $\epsilon = 10^{-9}$. The target batch size is 512 for CPU-based and 2048 for GPU-based calculations to ensure optimal performance on the respective architectures. Note that larger batches lead to a less tight screening, that is, more basis functions per grid batch. On the other hand, larger matrices result in better performance of the rate-determining linear algebra operations, especially on GPUs. All exchange-correlation evaluations employ four streams per GPU in order to maximize GPU utilization.

For all RI-Coulomb calculations the universal auxiliary basis "def2-universal-jfit" by Weigend[39] is used. In order to generate enough parallel workload, especially for GPUs, no further batching of the integrals is employed for a given *l*-quantum number combination (e.g., (*ss*|*s*)). If not stated otherwise, an

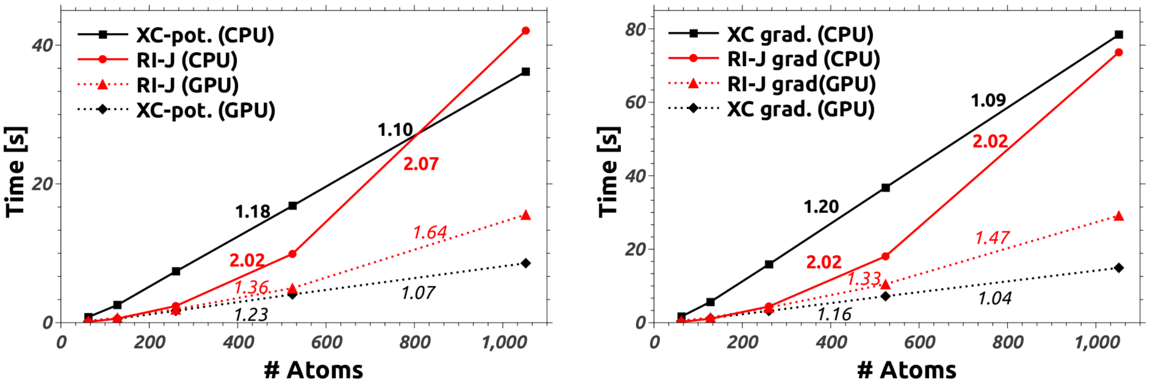


Figure 1. Program execution time for one Coulomb- or XC potential build averaged over all but the very first SCF cycles for AT-DNA-Fragments (B97M-V/TZVP/gm3; left). The timings for the evaluation of the nuclear gradient are shown on the right. The colored numbers denote the scaling to the respective predecessor. CPUs: 2 Intel Xeon Silver 4216 (32 cores @ 2.1 GHz). GPU: 1 NVIDIA GV100.

integral screening threshold of 10^{-10} is used, and SCF convergence is set to 10^{-7} for the norm of the commutator $||\mathbf{FPS} - \mathbf{SPF}||$. No incremental Kohn–Sham-builds were used throughout this work. All geometries are available for download online.[40,41]

4. ILLUSTRATIVE CALCULATIONS

We discuss the performance and scaling behavior of our presented methods with the example of a series of double-stranded DNA fragments of adenosine–thymine base pairs ($(\mathrm{AT})_x$). The scaling behavior with respect to the system as well as the basis set and DFT grid size is analyzed. Furthermore, the performance of CPU- and GPU-based calculations is shown.

Note that we restrict the discussion to the evaluation of the potential matrices only; the scaling behavior of the nuclear forces evaluation is virtually identical.

4.1. Scaling with Respect to the System Size.

While the scaling of the XC evaluation is expected to be linear, the scaling for RI-J computation should be quadratic with increasing system size. The wall times for a series of DNA fragments is shown in Figure 1 for both CPU- and GPU-based calculations, where the largest system contains 16 base pairs and 1052 atoms.

While the expected scaling is obtained for the CPU-based calculations, the scaling of RI-J on GPUs is clearly subquadratic on GPUs. This results from a lack of parallel workload for smaller systems, that is, larger systems show a better utilization of the GPU resources.

As shown in Figure 1, our CPU-based method is highly efficient and only takes 78 s to build the complete Kohn–Sham matrix for the largest system $(\mathrm{AT})_{16}$ with 1052 atoms and 22742 pure basis functions on a single compute node. On the GPU, this calculation takes 24 s, showing a speedup factor of 2.7 and 4.2 for RI-J and the XC evaluation, respectively. In contrast, the diagonalization step takes 316 s on CPUs or 135 s on the GPU, respectively, and is therefore by far the rate-determining step in the SCF calculation.

4.2. Scaling with Respect to the Basis Set Size.

The scaling with respect to an increasing size of the general basis set is shown in Figure 2 for an $(\mathrm{AT})_4$ fragment with increasing basis set size. In comparison to the unfavorable N^4 scaling of the conventional Coulomb integral evaluation without RI, the RI-J algorithm scales only quadratically $[N^2]$ with increasing basis set size, same as the XC evaluation. Note that usually the basis size scaling for RI-J is N^3. This holds true if one not only increases the size of general basis set, but also the auxiliary basis. However, as mentioned before, we always use the universal J-fit basis,[39] thereby obtaining an N^2 scaling.

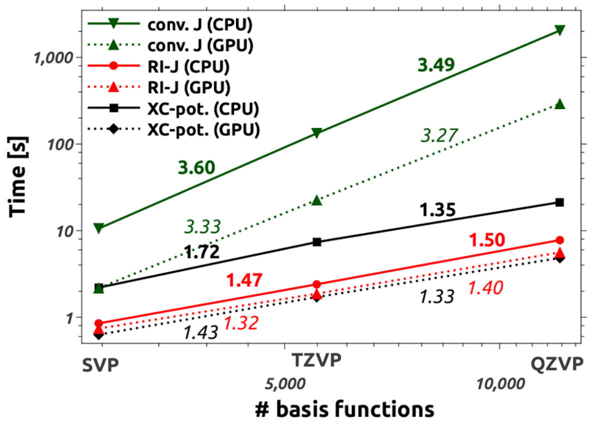


Figure 2. Program execution time for one Coulomb- (with and without RI) and XC potential build averaged over all but the very first SCF cycles for DNA$_4$/gm3 with different "def2-' basis sets. Double-logarithmic plot. Colored numbers denote scaling with respect to predecessor. CPUs: 2 Intel Xeon Silver 4216 (32 cores @ 2.1 GHz). GPU: 1 NVIDIA GV100.

For the largest basis set (def2-QZVP[42]), the speedup from conventional to RI-Coulomb evaluation is about 260-fold for $(\mathrm{AT})_4$.

As can be seen in Figure 2, the basis set scaling on GPUs for both RI-J and DFT is clearly subquadratic. For the RI-J evaluation, this might again be explained by an increasing workload, especially on GPUs. Furthermore, the second step in the algorithm (eq 3) only depends on the size of the auxiliary basis. In case of the DFT part, the reason might be larger numbers of significant basis functions $n_{\{\mu\}}$ for the given ngrid batches. Since the performance of the rate-determining matrix multiplications is usually better for larger matrices, the overall scaling is reduced. Furthermore, the terms scaling as $\mathcal{O}(\mathrm{n}_{\{\mu\}}\mathrm{n}_g)$, that is, linear with the basis set size, also contribute significantly to the overall computational time.

4.3. Comparison of Different Functional Types.

The performance of the three types of XC functionals is examined at $(AT)_4$/TZVP calculations of the potential and nuclear forces using Slater-Exchange/VWN-Correlation[43] (LDA), PBE[44] (GGA), and B97M-V[37] (mGGA). The results in Table 3 reconfirm that matrix multiplications are the rate-determining step, as the computational times directly correlate with the number of multiplications in Table 2 for both CPU- and GPU-based calculations.

Table 3. Program Execution Time [s] for One XC energy, One XC Potential, and One XC Gradient (Nuclear Forces) Calculation for DNA_4/TZVP/gm3[a]

functional type	energy	XC potential	XC gradient
	CPU		
LDA	2.9	2.1	3.8
GGA	3.7	2.5	12.4
MGGA	10.4	7.3	14.9
	GPU		
LDA	0.7 (4.2×)	0.5 (3.9×)	0.9 (4.4×)
GGA	1.0 (3.7×)	0.8 (3.2×)	2.7 (4.6×)
MGGA	2.0 (5.2×)	1.7 (4.3×)	3.2 (4.7×)

[a] The XC potential time is averaged over all but the very first SCF cycles. CPUs: 2 Intel Xeon Silver 4216 (32 cores @ 2.1 GHz). GPU: 1 NVIDIA GV100. CPU → GPU speedups are given in parentheses.

As expected the CPU to GPU speedups are highest for meta-GGAs due to the higher number of matrix multiplications. Note again that multigrids have been employed, that is, the grid for the final energy build is about 3.7 times larger than that for the potential evaluation. Thus, the final energy build takes longer than the potential evaluation although more matrix multiplications are involved for the latter.

4.4. Scaling with Respect to Grid Size. The gm3 grid employed so far in our calculations provides a grid error <1 μH (micro Hartree) per atom for the systems under investigation. For more accurate results, for example, accurate nuclear forces for structure optimizations, larger grids might be necessary. To analyze the increase in computational effort, timings for different grid sizes are shown in Figure 3. For a benchmark on the accuracy of the different grids, see ref 27. Note that the scaling is again better than the expected O(N) scaling. This is due to a more efficient screening if constant batch sizes are used, that is, the spatial extent of the grid batches decreases with increasing grid sizes, which results in a reduced number of significant basis functions $n_{\{\mu\}}$. Furthermore, the increasing number of grid batches results in a higher parallel workload.

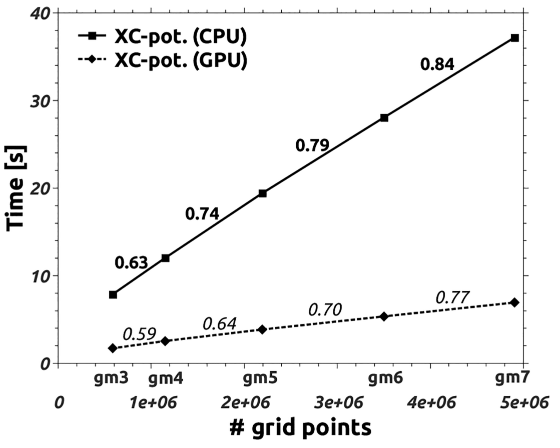


Figure 3. Program execution times for one XC potential build averaged over all but the very first SCF cycles for DNA_4/TZVP for different grids. The colored numbers denote the scaling to the respective predecessor. CPUs: 2 Intel Xeon Silver 4216 (32 cores @ 2.1 GHz). GPU: 1 NVIDIA GV100.

4.5. Multi-GPU Parallel Scaling. The scaling with the number of GPU devices is tested for a single compute node with one to four AMD Radeon VII GPUs for $(AT)_4$ and $(AT)_8$ using B97M-V/TZVP, shown in Figure 4. The numerical integration of the XC terms scales almost ideally with a high parallel efficiency. In contrast, the RI-J evaluation shows a poorer parallel efficiency due to the lack of parallel workload, where the computation with four GPUs for the smaller fragment is even slower than the calculation with only three GPUs. In comparison, the larger fragment $(AT)_8$ generates enough workload to at least ensure a steady speedup with up to four GPUs.

4.6. FLOP Utilization. The FLOP utilization as compared to the theoretical peak FLOP/s (TPF) of the corresponding CPU and GPU is shown in Figure 5 with the example of DNA base pairs $(AT)_x$ using B97M-V/TZVP. As discussed before, the rate-determining steps in the evaluation of the exchange-correlation terms are matrix multiplications, so that a high flop utilization of 58% of TPF can be obtained. From our experience, well optimized libraries (Intel MKL,[28] cuBLAS[29]) achieve for matrix multiplications roughly 75% of theoretical peak performance on both CPU and GPU, confirming again the dominance of the linear algebra steps within the exchange-correlation algorithm. Figure 5 also shows that the smaller systems show a lesser performance due to an overall smaller workload, while the saturation level is reached for the medium-sized $(AT)_4$ fragment on both CPU and GPU.

In contrast, the RI Coulomb evaluation on GPUs shows a rather poor FLOP utilization, which steadily increases with the system size up to $(AT)_{16}$. As mentioned before, this results from an insufficient workload for the GPUs even for larger systems, particularly for step 1 (eq 2). As an upside to this shortcoming, one should stress that the RI-Coulomb algorithm will show only slightly reduced performance on "gaming" GPUs with a comparably poor double-precision performance. On CPUs, however, we see a similar saturation in FLOP utilization as in case of the exchange-correlation evaluation with about 40% as compared to TPF. A similar picture results for the conventional analytical Coulomb algorithm on CPUs. Note that for both algorithms (RI and analytical) the J-engine is employed, which results in similar utilization numbers. Finally, the analytical Coulomb evaluation on GPU shows a far better FLOP utilization as compared to the RI algorithm as a result of a significantly higher workload.

5. CONCLUSION

We have presented a new improved RI-Coulomb algorithm that significantly reduces the computational workload by reducing the overall number of floating-point operations in the integral kernels of the J-engine algorithm for both CPU- and GPU-architectures. Furthermore, the reduced local memory necessary in GPU kernels allows for high performance evaluation on GPUs, shown with examples including basis sets up to quadruple-ζ quality. Thus, the evaluation of the XC potential became the bottleneck in nonhybrid DFT calculations, which required the development of an efficient GPU-

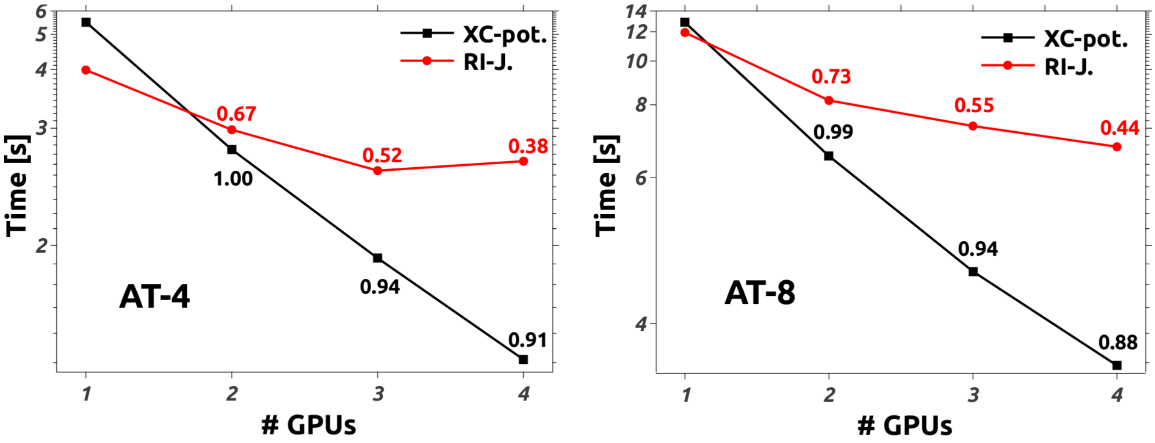


Figure 4. Program execution time (inverse plot) for one Coulomb or XC potential build averaged over all but the very first SCF cycles for DNA fragments with four ($(AT)_4$, left) and eight ($(AT)_8$, right) base pairs at B97M-V/TZVP using 1, 2, 3, or 4 AMD Radeon VII GPUs. The values in the graph denote the parallel efficiency compared to one GPU.

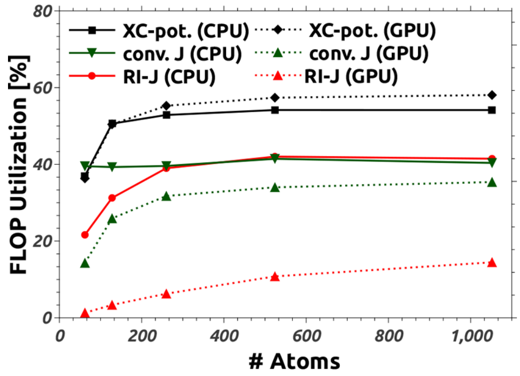


Figure 5. FLOP-utilization as percentage of theoretical peak performance of the employed CPU and GPU for one full Kohn–Sham build of AT-DNA-Fragments (B97M-V/TZVP/gm3). CPUs: 2 Intel Xeon Silver 4216 (32 cores @ 2.1 GHz, TPF = 1.075 TFLOP/s). GPU: 1 NVIDIA GV100 (TPF = 8.330 TFLOP/s).

based algorithm to evaluate XC contributions strictly in terms of highly efficient linear algebra operations.

For first test calculation, we have analyzed the overall performance and scaling behavior of the presented methods. Apart from an efficient GPU algorithm for nonhybrid DFT calculations, which allows a single Kohn–Sham matrix build for a DNA fragment with 1052 atoms and using a triple-ζ basis (22742 pure basis functions) in 24 s on a single GPU, our RI-DFT method also results in a highly efficient CPU code. Employing a single compute node with 32 cores, the calculation requires only 78 s for a complete Kohn–Sham matrix construction. This also enables, for example, ab initio molecular dynamics on regular CPU nodes without additional accelerator cards.

While this work ensures efficient calculations employing nonhybrid DFT, the most popular functionals nowadays remain hybrid or range-corrected functionals that also include exact-exchange contributions. Thus, the now by far rate-determining step in these calculations is the evaluation of the exact-exchange potential, even if the efficient GPU-accelerated sn-LinK method[11] is employed.

APPENDIX

A. Additional Terms for GGA and Meta-GGA

A.1. Additional Terms for GGA XC Potential. GGA functionals also depend on the gradient of the density $\nabla\rho$, which is obtained with only small overhead from the intermediate quantity $F_{\nu g}$ as

$$\nabla\rho(\mathbf{r}_g) = 2\sum_{\mu\nu} \rho_{\mu\nu}\chi_\mu(\mathbf{r}_g)\nabla\chi_\nu(\mathbf{r}_g) = 2\sum_{\nu} F_{\nu g}\nabla\chi_\nu(\mathbf{r}_g) \tag{36}$$

Therefore, an additional contribution to the XC potential arises as

$$V_{\mu\nu}^{\nabla\rho} = \sum_g w_g \frac{\partial\varepsilon_{\mathrm{xc}}}{\partial\nabla\rho}(\mathbf{r}_g)\cdot[\chi_\mu(\mathbf{r}_g)\nabla\chi_\nu(\mathbf{r}_g) + \nabla\chi_\mu(\mathbf{r}_g)\chi_\nu(\mathbf{r}_g)] \tag{37}$$

This term may best be evaluated by addition of another intermediate quantity $G_{\nu g}^{\mathrm{GGA}}$ to $G_{\nu g}$

$$G_{\nu g}^{\mathrm{GGA}} = w_g \frac{\partial\varepsilon_{\mathrm{xc}}}{\partial\nabla\rho}(\mathbf{r}_g)\cdot\nabla\chi_\nu(\mathbf{r}_g) \tag{38}$$

and evaluating eq 31 with the modified $G_{\nu g}$ and finally symmetrizing to account for the transpose in eq 37. Therefore, the GGA XC potential can be obtained at nearly the same cost (i.e., two $O(N^3)$ matrix multiplications) as the LDA XC potential.

A.2. Additional Terms for GGA XC Forces. The GGA part for XC forces is evaluated similarly to the LDA forces (eq 32) as

$$E_{\nabla\rho}^{A_x} = \sum_g w_g \frac{\partial\varepsilon_{\mathrm{xc}}}{\partial\nabla\rho}(\mathbf{r}_g)(\nabla\rho(\mathbf{r}_g))^{A_x} \tag{39}$$

where, analogously to eq 33,

$$(\nabla\rho(\mathbf{r}_g))^{A_x} = 2\sum_{\mu\nu} [\rho_{\mu\nu}\nabla\chi_\mu(\mathbf{r}_g)\chi_\nu^{A_x}(\mathbf{r}_g) + \chi_\mu(\mathbf{r}_g)(\nabla\chi_\nu(\mathbf{r}_g))^{A_x}] \tag{40}$$

is the infinitesimal change of $\nabla\rho(\mathbf{r}_g)$ when changing a nuclear coordinate A_x. The first term of eq 40 necessitates the computation of

$$\nabla F_{\nu g} = \sum_{\mu} \rho_{\mu\nu} \nabla \chi_{\mu}(\mathbf{r}_g) \tag{41}$$

requiring three additional $O(\mathrm{N}^3)$ steps (x, y, and z components of gradient). The second term of eq 40, however, is available directly from $F_{\nu g}$ and only requires the perturbed gradient of the basis functions, available from the second basis function derivatives:

$$\left(\frac{\partial \chi_{\mu}}{\partial x}(\mathbf{r}_g)\right)^{A_y} = \begin{cases} -\dfrac{\partial^2 \chi_{\mu}}{\partial x \partial y}(\mathbf{r}_g), & \text{if } \mu \in A \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

A.3. Additional Terms for Meta-GGA XC Potential. Meta-GGA functionals additionally depend on the kinetic energy density

$$\tau(\mathbf{r}) = \sum_{i} |\nabla \varphi_i(\mathbf{r})|^2 = \sum_{\mu\nu} \rho_{\mu\nu} \nabla \chi_{\mu}(\mathbf{r}) \cdot \nabla \chi_{\nu}(\mathbf{r}) \tag{43}$$

For the evaluation of eq 43, the computation of $\nabla F_{\nu g}$ (eq 41) cannot be avoided, resulting in three additional $O(\mathrm{N}^3)$ steps for the meta-GGA XC energy.

The meta-GGA XC potential also contains another additional term of the form:

$$V_{\mu\nu}^{\tau} = \sum_{g} w_g \frac{\partial \varepsilon_{\mathrm{xc}}}{\partial \tau}(\mathbf{r}_g) \nabla \chi_{\mu}(\mathbf{r}_g) \cdot \nabla \chi_{\nu}(\mathbf{r}_g) \tag{44}$$

which necessitates the computation of

$$\nabla G_{\nu g} = w_g \frac{\partial \varepsilon_{\mathrm{xc}}}{\partial \tau}(\mathbf{r}_g) \nabla \chi_{\nu}(\mathbf{r}_g) \tag{45}$$

and leads to three additional $O(\mathrm{N}^3)$ steps for the computation of $V_{\mu\nu}^{\tau}$:

$$V_{\mu\nu}^{\tau} = \sum_{g} \nabla \chi_{\mu}(\mathbf{r}_g) \cdot \nabla G_{\nu g} \tag{46}$$

A.4. Additional Terms for Meta-GGA XC Forces. The τ-dependent term of meta-GGA XC forces is evaluated analogously to eqs 32 and 39, requiring

$$\tau^{A_x}(\mathbf{r}_g) = 2 \sum_{\mu\nu} \rho_{\mu\nu} \nabla \chi_{\mu}(\mathbf{r}_g) \cdot (\nabla \chi_{\nu}(\mathbf{r}_g))^{A_x} \tag{47}$$

It can be directly computed from $\nabla F_{\nu g}$ (eq 41) and $(\nabla \chi_{\nu}(\mathbf{r}_g))^{A_x}$ (eq 42) without addition of another $O(\mathrm{N}^3)$ step.

AUTHOR INFORMATION

Corresponding Author

Christian Ochsenfeld − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 München, Germany*; orcid.org/0000-0002-4189-6558; Email: christian.ochsenfeld@uni-muenchen.de

Authors

Jörg Kussmann − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 München, Germany*; orcid.org/0000-0002-4724-8551

Henryk Laqua − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 München, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.0c01252

Author Contributions

[†]J.K. (RI-Coulomb) and H.L. (DFT) contributed equally to this work.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This article is dedicated to the memory of James B. Anderson (1935−2021). The authors acknowledge financial support by the "Deutsche Forschungsgemeinschaft" (DFG) via the SFB 1309-32587107 and the clusters of excellence (EXC2111-390814868) (Munich Center for Quantum Science and Technology; MCQST) and EXC 2089/1-390776260 (e-conversion). C.O. further acknowledges financial support as Max-Planck-Fellow at the MPI-FKF Stuttgart.

REFERENCES

(1) Yasuda, K. Two-electron integral evaluation on the graphics processor unit. *J. Comput. Chem.* **2008**, *29*, 334−42.

(2) Yasuda, K. Accelerating Density Functional Calculations with Graphics Processing Unit. *J. Chem. Theory Comput.* **2008**, *4*, 1230−1236.

(3) Ufimtsev, I. S.; Martínez, T. J. Graphical Processing Units for Quantum Chemistry. *Comput. Sci. Eng.* **2008**, *10*, 26−34.

(4) Ufimtsev, I. S.; Martínez, T. J. Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field Implementation. *J. Chem. Theory Comput.* **2009**, *5*, 1004−1015.

(5) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics. *J. Chem. Theory Comput.* **2009**, *5*, 2619−2628.

(6) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(7) 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.

(8) 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.

(9) Kussmann, J.; Ochsenfeld, C. Hybrid CPU/GPU Integral Engine for Strong-Scaling Ab Initio Methods. *J. Chem. Theory Comput.* **2017**, *13*, 3153−3159.

(10) Kussmann, J.; Ochsenfeld, C. Employing OpenCL to Accelerate Ab Initio Calculations on Graphics Processing Units. *J. Chem. Theory Comput.* **2017**, *13*, 2712−2716.

(11) Laqua, H.; Thompson, T. H.; Kussmann, J.; Ochsenfeld, C. Highly Efficient, Linear-Scaling Seminumerical Exact-Exchange Method for Graphic Processing Units. *J. Chem. Theory Comput.* **2020**, *16*, 1456−1468.

(12) Kalinowski, J.; Wennmohs, F.; Neese, F. Arbitrary Angular Momentum Electron Repulsion Integrals with Graphical Processing Units: Application to the Resolution of Identity Hartree−Fock Method. *J. Chem. Theory Comput.* **2017**, *13*, 3160−3170.

(13) Nitsche, M. A.; Ferreria, M.; Mocskos, E. E.; González Lebrero, M. C. GPU Accelerated Implementation of Density Functional Theory for Hybrid QM/MM Simulations. *J. Chem. Theory Comput.* **2014**, *10*, 959−967.

(14) Manathunga, M.; Miao, Y.; Mu, D.; Götz, A. W.; Merz, K. M. Parallel Implementation of Density Functional Theory Methods in the Quantum Interaction Computational Kernel Program. *J. Chem. Theory Comput.* **2020**, *16*, 4315−4326.

(15) Whitten, J. L. Coulombic potential energy integrals and approximations. *J. Chem. Phys.* **1973**, *58*, 4496−4501.

(16) Dunlap, B. I.; Connolly, J.; Sabin, J. On some approximations in applications of X α theory. *J. Chem. Phys.* **1979**, *71*, 3396−3402.

(17) Vahtras, O.; Almlöf, J.; Feyereisen, M. Integral approximations for LCAO-SCF calculations. *Chem. Phys. Lett.* **1993**, *213*, 514−518.

(18) Eichkorn, K.; Treutler, O.; Öhm, H.; Häser, M.; Ahlrichs, R. Auxiliary basis sets to approximate Coulomb potentials. *Chem. Phys. Lett.* **1995**, *240*, 283−290.

(19) Weigend, F. A fully direct RI-HF algorithm: Implementation, optimized auxiliary basis sets, demonstration of accuracy and efficiency. *Phys. Chem. Chem. Phys.* **2002**, *4*, 4285−4291.

(20) Neese, F. An improvement of the resolution of the identity approximation for the formation of the Coulomb matrix. *J. Comput. Chem.* **2003**, *24*, 1740−1747.

(21) Mintmire, J. W.; Dunlap, B. I. Fitting the Coulomb potential variationally in linear-combination-of-atomic-orbitals density-functional calculations. *Phys. Rev. A: At., Mol., Opt. Phys.* **1982**, *25*, 88−95.

(22) Reza Ahmadi, G.; Almlöf, J. The Coulomb operator in a Gaussian product basis. *Chem. Phys. Lett.* **1995**, *246*, 364−370.

(23) White, C. A.; Head-Gordon, M. A J matrix engine for density functional theory calculations. *J. Chem. Phys.* **1996**, *104*, 2620.

(24) Shao, Y.; Head-Gordon, M. An improved J matrix engine for density functional theory calculations. *Chem. Phys. Lett.* **2000**, *323*, 425−433.

(25) McMurchie, L. E.; Davidson, E. R. One- and Two-Electron Integrals over Cartesian Gaussian Functions. *J. Comput. Phys.* **1978**, *26*, 218−231.

(26) Becke, A. D. A multicenter numerical integration scheme for polyatomic molecules. *J. Chem. Phys.* **1988**, *88*, 2547.

(27) Laqua, H.; Kussmann, J.; Ochsenfeld, C. An improved molecular partitioning scheme for numerical quadratures in density functional theory. *J. Chem. Phys.* **2018**, *149*, 204111.

(28) *Intel C++ Compiler*, version 19.1.0.166, see https://software.intel.com/c-compilers, 2019.

(29) *CUDA Toolkit* 10.1, see https://developer.nvidia.com/cuda-10.1-download-archive-base, 2019.

(30) Nugteren, C. CLBlast: A Tuned OpenCL BLAS Library. *Proceedings of the International Workshop on OpenCL*; Association for Computing Machinery: New York, NY, 2018.

(31) Marques, M. A. L.; Oliveira, M. J. T.; Burnus, T. Libxc: A library of exchange and correlation functionals for density functional theory. *Comput. Phys. Commun.* **2012**, *183*, 2272−2281.

(32) Laqua, H.; Kussmann, J.; Ochsenfeld, C. Efficient and Linear-Scaling Seminumerical Method for Local Hybrid Density Functionals. *J. Chem. Theory Comput.* **2018**, *14*, 3451−3458.

(33) Pulay, P. Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules. *Mol. Phys.* **1969**, *17*, 197−204.

(34) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(35) 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.

(36) *ROCm* 3.8.0, see https://www.amd.com/en/graphics/servers-solutions-rocm, 2021.

(37) Mardirossian, N.; Head-Gordon, M. Mapping the genome of meta-generalized gradient approximation density functionals: The search for B97M-V. *J. Chem. Phys.* **2015**, *142*, 074111.

(38) Burow, A. M.; Sierka, M. Linear Scaling Hierarchical Integration Scheme for the Exchange-Correlation Term in Molecular and Periodic Systems. *J. Chem. Theory Comput.* **2011**, *7*, 3097−3104.

(39) Weigend, F. Accurate Coulomb-fitting basis sets for H to Rn. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1057−1065.

(40) Maurer, S. A.; Lambrecht, D. S.; Flaig, D.; Ochsenfeld, C. Distance-dependent Schwarz-based integral estimates for two-electron integrals: Reliable tightness vs. rigorous upper bounds. *J. Chem. Phys.* **2012**, *136*, 144107.

(41) Structure files are available for download at our homepage. https://www.cup.uni-muenchen.de/pc/ochsenfeld/download/, 2021.

(42) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.

(43) Vosko, S. H.; Wilk, L.; Nusair, M. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Can. J. Phys.* **1980**, *58*, 1200−11.

(44) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

3.8 Publication VIII: Communication: Density functional theory model for multi-reference systems based on the exact-exchange hole normalization

H. Laqua, J. Kussmann, C. Ochsenfeld

J. Chem. Phys. **148**, 121101 (2018).

Abstract

The correct description of multi-reference electronic ground states within Kohn-Sham density functional theory (DFT) requires an ensemble-state representation, employing fractionally occupied orbitals. However, the use of fractional orbital occupation leads to non-normalized exact-exchange holes, resulting in large fractional-spin errors for conventional approximative density functionals. In this communication, we present a simple approach to directly include the exact-exchange-hole normalization into DFT. Compared to conventional functionals, our model strongly improves the description for multi-reference systems, while preserving the accuracy in the single-reference case. We analyze the performance of our proposed method at the example of spin-averaged atoms and spin-restricted bond dissociation energy surfaces.

Reproduced from:

H. Laqua, J. Kussmann, C. Ochsenfeld
“Communication: Density functional theory model for multi-reference systems based on the exact-exchange hole normalization”
J. Chem. Phys. **148**, 121101 (2018).

with the permission of AIP Publishing.
`https://aip.scitation.org/doi/10.1063/1.5025334`

Communication: Density functional theory model for multi-reference systems based on the exact-exchange hole normalization

Henryk Laqua, Jörg Kussmann, and Christian Ochsenfeld[a)]
Department of Chemistry and Center for Integrated Protein Science (CIPSM), University of Munich (LMU), D-81377 München, Germany

(Received 8 February 2018; accepted 13 March 2018; published online 27 March 2018)

The correct description of multi-reference electronic ground states within Kohn-Sham density functional theory (DFT) requires an ensemble-state representation, employing fractionally occupied orbitals. However, the use of fractional orbital occupation leads to non-normalized exact-exchange holes, resulting in large fractional-spin errors for conventional approximative density functionals. In this communication, we present a simple approach to directly include the exact-exchange-hole normalization into DFT. Compared to conventional functionals, our model strongly improves the description for multi-reference systems, while preserving the accuracy in the single-reference case. We analyze the performance of our proposed method at the example of spin-averaged atoms and spin-restricted bond dissociation energy surfaces. *Published by AIP Publishing.* https://doi.org/10.1063/1.5025334

I. INTRODUCTION

The correct description of multi-reference electronic ground states is a major challenge within Kohn-Sham density functional theory (KS-DFT).[1,2] A formally exact solution exists in the form of ensemble-density-functional-theory (eDFT) employing fractionally occupied orbitals.[3,4]

One practical approach to eDFT is the thermally assisted occupation DFT (TAO-DFT).[5–7] An alternative solution is the representation of the ensemble-density by a linear-combination of multiple determinants in the spin-restricted-ensemble-referenced-Kohn-Sham (REKS)-method.[8–12]

Furthermore, designated strong-correlation functionals based on Becke's real-space correlation model[13–15] have been proposed.[16–19] Here, an effective exchange-hole normalization is obtained from a combination of semilocal DFT-ingredients and the exact-exchange potential. Subsequently, a model of static correlation is constructed using these hole-normalizations.

Fractionally occupied orbitals lead to local, but non-normalized exact-exchange holes. Therefore, the exact-exchange hole normalization can directly be used to model static correlation. Our present work is based on the strong-correlation model by Johnson,[18] but instead of the effective Becke-Roussel normalization[13–15] we employ the exact-exchange hole normalization. We briefly outline the methodology of Johnson's strong-correlation model[18] and present our method to obtain exchange-hole normalizations in Sec. II. Subsequently, we test our approach, denoted as exchange-hole-normalization DFT (xhn-DFT), on selected strong-correlation problems in Sec. III.

II. THEORY

Johnson proposed a non-empirical functional designed to deal with the fractional spin-error in spin-averaged atoms.[18] It defines a sum of spin-densities

$$\rho_s = \rho_\alpha + f_{\text{opp}}\rho_\beta, \tag{1}$$

assuming the α-spin to be the majority spin in the spin polarized case. The strong correlation factor f_{opp} was first introduced in Ref. 14 and is obtained from effective hole normalizations N_α, N_β as

$$f_{\text{opp}} = \min\left(\frac{1-N_\alpha}{N_\beta}, \frac{1-N_\beta}{N_\alpha}, 1\right), \tag{2}$$

where the effective hole normalizations are obtained from the reversed Becke-Roussel machinery.[13–15]

The final energy expression of the most sophisticated NDC2-scheme from Ref. 18 reads

$$\begin{aligned} E_{XC}^{\text{NDC2}} &= E_{\text{X}}^{\text{sl}}[\rho_s] + (1-f_{\text{opp}})E_{\text{X}}^{\text{sl}}[\rho_\beta] \\ &\quad + (1-f_{\text{opp}})E_{\text{C}}^{\text{dyn,opp}}[\rho_s,\rho_\beta] \\ &\quad + E_{\text{C}}^{\text{dyn,par}}[\rho_s] + (1-f_{\text{opp}})E_{\text{C}}^{\text{dyn,par}}[\rho_\beta], \end{aligned} \tag{3}$$

where any semilocal exchange functional $E_{\text{X}}^{\text{sl}}[\rho]$ and correlation functional $E_{\text{C}}^{\text{dyn}}[\rho]$ may be used in principle.

In the present work, we utilize the aforementioned NDC2-model of Eqs. (1)–(3), but employ the exact-exchange hole normalizations instead of the Becke-Roussel normalizations in Eq. (2).

A. Exact-exchange hole normalization for fractionally occupied orbitals

The exact-exchange hole is given in terms of fractionally occupied spin-orbitals $\varphi_{i\sigma}$ as

$$h_{X\sigma}^{\text{ex}}(\mathbf{r}_1,\mathbf{r}_2) = \frac{\sum_{ij} f_{i\sigma} f_{j\sigma} \varphi_{i\sigma}^*(\mathbf{r}_1)\varphi_{j\sigma}(\mathbf{r}_1)\varphi_{i\sigma}(\mathbf{r}_2)\varphi_{j\sigma}^*(\mathbf{r}_2)}{\rho_\sigma(\mathbf{r}_1)}, \tag{4}$$

[a)]Electronic mail: christian.ochsenfeld@uni-muenchen.de with spin-orbital occupation numbers $0 \leq f_{i\sigma} \leq 1$ and the one-particle density ρ_σ of spin $\sigma = \alpha, \beta$. In the case of non-integer orbital occupations, the exact-exchange hole normalization

$$N^{\text{ex}}_{X\sigma}(\mathbf{r}_1) = \int h^{\text{ex}}_{X,\sigma}(\mathbf{r}_1, \mathbf{r}_2) d^3\mathbf{r}_2 \tag{5}$$

is in general not 1 anymore.

Expanding Eqs. (4) and (5) with a linear-combination of atomic orbitals (LCAO) and assuming real molecular orbitals yields

$$\begin{aligned} N^{\text{ex}}_{X,\sigma}(\mathbf{r}_1) &= \sum_{\mu\nu\kappa\lambda ij} f_{i\sigma} f_{j\sigma} C^\sigma_{\mu i} C^\sigma_{\nu i} C^\sigma_{\kappa j} C^\sigma_{\lambda j} \chi_\mu(\mathbf{r}_1)\chi_\lambda(\mathbf{r}_1) \\ &\quad \times \left[\int \chi_\nu(\mathbf{r}_2)\chi_\kappa(\mathbf{r}_2) d\mathbf{r}_2\right] \rho_\sigma(\mathbf{r}_1)^{-1} &(6) \\ &= \sum_{\mu\nu\kappa\lambda} \chi_\mu(\mathbf{r}_1) P^\sigma_{\mu\nu} S_{\nu\kappa} P^\sigma_{\kappa\lambda} \chi_\lambda(\mathbf{r}_1) \rho_\sigma(\mathbf{r}_1)^{-1}, &(7) \end{aligned}$$

where $C^\sigma_{\mu i}$ are the MO coefficients. In Eq. (7), the definition of the (spin-resolved) one-particle density matrix

$$P^\sigma_{\mu\nu} = \sum_i f_{i\sigma} C^\sigma_{\mu i} C^\sigma_{\nu i} \tag{8}$$

has been inserted. Eq. (7) may be evaluated on a numerical integration grid as

$$N_{X\sigma}(\mathbf{r}_g) = \sum_{\mu\lambda} \chi_\mu(\mathbf{r}_g)\, (\mathbf{PSP})^\sigma_{\mu\lambda}\, \chi_\lambda(\mathbf{r}_g) \rho_\sigma(\mathbf{r_g})^{-1}, \tag{9}$$

where $\mathbf{r}_g$ denotes a grid point.

Our proposition to use exact exchange hole normalizations instead of the effective Becke-Roussel normalizations leads to a number of advantages:

1. No evaluation of the exact exchange-energy density at every grid point is necessary, saving a significant amount of computational effort.
2. The computation of the complicated reverse Becke-Roussel machinery is avoided.
3. In the single-reference case, i.e., if no fractional orbital occupations are employed, the result is unchanged. Therefore, the well-appreciated performance of semilocal DFT in these cases is preserved, avoiding possible double-counting of static-correlation.

B. Treatment of semilocal DFT ingredients

For simplicity, only the spin-restricted closed-shell formalism will be discussed below, noting that the extension to open-shell problems is straightforward. In this case, Eq. (2) reduces to

$$f_{\text{opp}} = \min\left(\frac{1}{N} - 1, 1\right) = \min\left(\frac{\rho}{\tilde{\rho}}, 2\right) - 1, \tag{10}$$

defining

$$\tilde{\rho}(\mathbf{r}_g) = \sum_{\mu\lambda} \chi_\mu(\mathbf{r}_g)\, (\mathbf{PSP})_{\mu\lambda}\, \chi_\lambda(\mathbf{r}_g). \tag{11}$$

The sum of densities from Eq. (1) is then given as

$$\rho_s = \frac{1}{2}(1 + f_{\text{opp}})\rho = \frac{1}{2}\min\left(\frac{\rho}{\tilde{\rho}}, 2\right) \times \rho. \tag{12}$$

If a generalized gradient approximation (GGA)- or meta-GGA is used for $E^{\text{sl}}_{\text{XC}}[\rho]$, the semilocal DFT ingredients, which correspond to the sum of densities ρ_s, are also required. The gradient of the sum of densities is obtained as

$$\nabla\rho_s = \begin{cases} 0, & f_{\text{opp}} \geq 1 \\ \dfrac{\rho}{\tilde{\rho}}\nabla\rho - \dfrac{1}{2}\dfrac{\rho^2}{\tilde{\rho}^2}\nabla\tilde{\rho}, & f_{\text{opp}} < 1 \end{cases}. \tag{13}$$

In the case of meta-GGA functionals, the kinetic energy density τ_s is obtained analogously to ρ_s as

$$\tau_s = \frac{1}{2}(1 + f_{\text{opp}})\tau. \tag{14}$$

To summarize, we introduce exact-exchange hole normalizations obtained by Eq. (9) into the NDC2-scheme of Ref. 18 to account for static-correlation in the multi-reference limit. The so-derived method, denoted as exchange-hole-normalization-DFT (xhn-DFT), is assessed for selected multi-reference problems below.

III. RESULTS

A. Spin-averaged atoms

Spin-averaged atoms represent the spin-restricted homonuclear bond dissociation limit. Therefore, the energy of spin-polarized and spin-averaged atoms should be equal. This condition has been successfully employed to fit and assess the strong-correlation functionals of Refs. 16–19. To illustrate, consider the orbital occupation of the spin-polarized carbon-atom

$$(2p^\alpha_x)^1(2p^\alpha_y)^1(2p^\alpha_z)^0(2p^\beta_x)^0(2p^\beta_y)^0(2p^\beta_z)^0. \tag{15}$$

The corresponding spin-depolarized (spin-averaged) configuration is obtained by averaging α- and β-occupation numbers, leading to

$$(2p^\alpha_x)^{\frac{1}{2}}(2p^\alpha_y)^{\frac{1}{2}}(2p^\alpha_z)^0(2p^\beta_x)^{\frac{1}{2}}(2p^\beta_y)^{\frac{1}{2}}(2p^\beta_z)^0. \tag{16}$$

We test our xhn-DFT scheme on spin-averaged main group atoms in Table I, employing the local-density approximation (LDA) with VWN-parametrization,[20] the GGA-functional PBE,[21] and the meta-GGA functional TPSS.[22] The calculations are performed with our FermiONs++ program[23–25] employing orbitals from restricted open-shell-KS calculations with the def2-TZVP basis set.[26]

Conventional density functionals and especially Hartree-Fock cannot describe spin-averaged atoms correctly, due to their inherent multi-reference character, leading to large fractional spin errors. However, our xhn-DFT-scheme corrects for the largest part of the errors, only slightly overestimating strong correlation in most cases. Furthermore, the errors decrease when better functionals are employed [MAE(xhn-LDA) > MAE(xhn-PBE) > MAE(xhn-TPSS)]. The accuracy of the xhn-TPSS-method is similar to the NDC2-scheme of Ref. 18 and superior to the empirically fitted functionals B13[16,17] and KP16.[19] Note also that all xhn-DFT methods are exact (exhibit no fractional spin error) for the spin-averaged hydrogen atom.

B. Homo-nuclear bond dissociation

We investigate the transition from a single-reference to a multi-reference problem employing the example of

TABLE I. Fractional spin errors for spin-averaged main group atoms from H to Br and H to Cl in kcal mol^{-1}. ME = mean error; MAE = mean absolute error. Spin-polarized results are from self-consistent restricted open-shell calculations. The spin-depolarized results are obtained from the same set of orbitals as the spin-polarized results but with spin-averaged density matrices. If not stated otherwise, the results have been obtained using the def2-TZVP-basis and a [99/590]-grid. Errors for the individual atoms are provided in the supplementary material.

	H to Cl		H to Br	
Method	ME	MAE	ME	MAE
HF[a]	139.8	139.8	127.9	127.9
LDA	22.9	22.9	20.1	20.1
PBE	26.3	26.3	23.4	23.4
TPSS	30.0	30.0	26.6	26.6
xhn-LDA	−8.7	8.7	−9.2	9.2
xhn-PBE	−4.1	4.1	−4.3	4.3
xhn-TPSS	−2.5	2.6	−3.0	3.1
J13 (NDC2)[a]	−0.2	1.5	−2.6	3.5
B13[b]	3.9	5.0	−0.8	6.9
KP16 (PMF3)[c]	(−8.0)	(8.1)		

[a] Post-LDA (complete basis set-limit; NUMOL) from Ref. 18.

[b] Post-LDA (complete basis set-limit; NUMOL) from Ref. 17.

[c] Self-consistent values (G3-large basis; QChem) from Ref. 19. Only values for H, N, O, F, Si, and Cl are available.

spin-restricted bond dissociation energy surfaces. Here, fractional occupation numbers are obtained from Fermi-smearing,[27] i.e., the occupation numbers are obtained from the Fermi-Dirac distribution as

$$f_i = \frac{1}{1+\exp(\frac{\varepsilon_i-\mu}{k_BT})}, \tag{17}$$

where f_i is the occupation of the ith molecular orbital (MO), k_B is the Boltzmann constant, T is the temperature (in Kelvin), ε_i is the energy of the ith MO, and μ is the Fermi level. Note that Fermi-smearing is only used to model static-correlation[28] and should not be confused with a thermodynamical treatment at finite-temperatures.

First the bond dissociation of H_2 is investigated using our xhn-TPSS scheme for different temperatures in Fig. 1. Note that the xhn-TPSS approach predicts the correct dissociation limit for any non-zero temperature since both orbitals are half-occupied in the case of a vanishing HOMO-LUMO gap. However, the dissociation curves exhibit an unphysical maximum at intermediate bond distances if low temperatures are employed. This is due to non-optimal occupation numbers. Nevertheless, T = 10 000 K yields a greatly improved dissociation curve compared to the conventional TPSS-curve (which is identical to the 0 K-curve) without significantly altering the results for the minimum structure.

Moreover, Fig. 2 shows that the occupation numbers at 10 000 K match the natural orbital occupation numbers (NOONs) from the full-CI reference best. Overall, our xhn-TPSS approach predicts the correct spin-restricted dissociation limit. However, the correct description at intermediate distances, although significantly improved compared to conventional DFT, remains challenging.

Regarding the temperature-dependence, it would be more desirable to directly determine the optimal occupation numbers within a variational scheme. Preliminary tests of such an optimization scheme have been performed for the xhn-LDA method; however, multiple local minima of the energy with respect to occupation numbers complicate a straightforward application of this approach. We are currently investigating improvements to such optimization schemes.

Furthermore, we show that the xhn-TPSS method of the present work is also capable of describing breaking of multiple bonds at the example of N_2 in Fig. 3. We compare our results to the full-CI/6-31G* reference[29] and find our results to be in good agreement at intermediate and long distances for T = 10 000 K. Note that for the equilibrium geometry the full-CI energy is too high due to the importance of dynamic

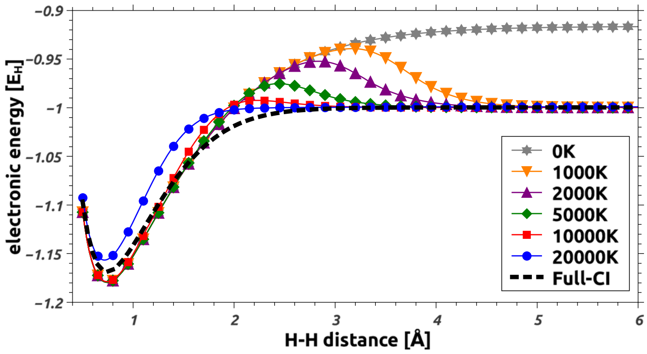


FIG. 1. H_2-dissociation-curve from xhn-TPSS for different temperatures compared to a full-CI reference and employing the def2-TZVP basis. The 0 K-curve is identical to the conventional TPSS-curve.

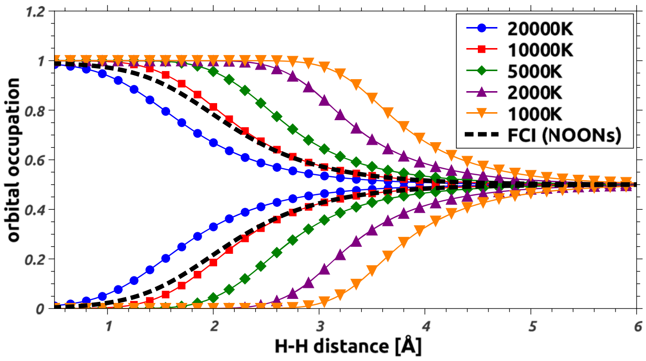


FIG. 2. Fractional occupation numbers at different H–H distances and various temperatures compared to the natural orbital occupation numbers (NOONs) from the full-CI reference.

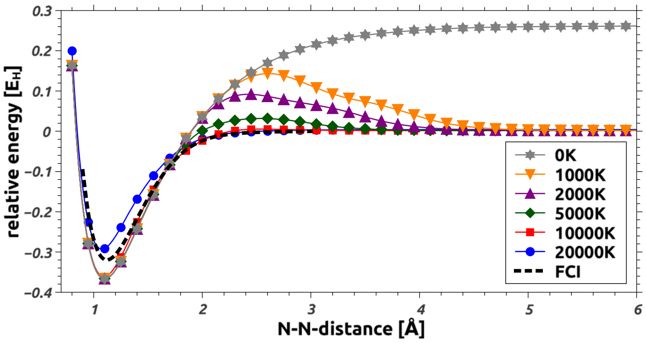


FIG. 3. N_2-dissociation-curve from xhn-TPSS with different temperatures and employing the def2-TZVP basis compared to a full-CI/6-31G* reference.[29] The relative energies are referenced to the energies of the spin-polarized atoms.

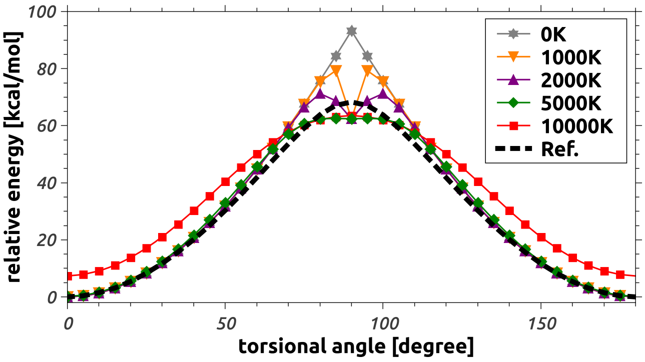


FIG. 4. Relative energies for ethylene at different torsional angels employing our xhn-TPSS-method at various temperatures in comparison to the MR-ccCA[30,31] reference values from Ref. 32. The xhn-TPSS calculations employ the def2-TZVP-basis and are referenced to the value at 0 K in the minimum (0° torsional angle). The geometries of Ref. 32 have been employed.

correlation effects that cannot be accurately described with the small 6-31G* basis within full-CI.

C. Rotational barrier of ethylene

Finally, the breaking of the C–C π-bond during rotation around the C–C-bond axis in ethylene is investigated in Fig. 4. Conventional DFT methods, as represented in the 0 K-curve, exhibit a discontinuity at the transition state (90° torsion) due to the inability to properly describe the multi-reference character of the wave-function in this case. This also leads to a significantly overestimated rotational barrier. In contrast, our xhn-TPSS method yields a smooth curve for sufficiently large temperatures ($\geq$5000 K). Similar to the bond dissociation curves of H_2 and N_2, an unphysical maximum at intermediate torsional angles is observed for low temperatures due to the non-optimal orbital occupations. Overall, our xhn-TPSS method strongly improves the description of the rotation-energy profile and is in good agreement with the multi-reference calculation in Refs. 30–32.

IV. CONCLUSION AND OUTLOOK

In the present work, we proposed the use of the exact-exchange hole normalization as a DFT ingredient. Our xhn-DFT method allows for a significantly improved description of multi-reference problems at a similar cost as conventional semilocal KS-DFT and without employing empirically fitted parameters. A smooth transition from the single- to the multi-reference limit is obtained when Fermi-smearing with T = 5000–10 000 K is applied, which is in accordance with the proposed temperatures of Ref. 28.

A reliable optimization scheme for orbital occupation numbers and a generalization of our xhn-DFT method to (local-)hybrid-functionals are currently under investigation. Finally, we hope that our work encourages further development of density functionals for strongly correlated systems.

SUPPLEMENTARY MATERIAL

See supplementary material for the fractional spin errors of the individual atoms.

ACKNOWLEDGMENTS

The authors acknowledge the financial support by the *Deutsche Forschungsgemeinschaft* (DFG) in the *SFB* 749 “Dynamik und Intermediate molekularer Transformationen” and the DFG *cluster of excellence* (EXC114) “Center for Integrative Protein Science Munich” (CIPSM). C.O. further acknowledges financial support as Max-Planck-Fellow at the MPI-FKF Stuttgart.

[1] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[2] D. Cremer, M. Filatov, V. Polo, E. Kraka, and S. Shaik, Int. J. Mol. Sci. **3**, 604 (2002).
[3] S. M. Valone, J. Chem. Phys. **73**, 4653 (1980).
[4] E. H. Lieb, Int. J. Quantum Chem. **24**, 243 (1983).
[5] J.-D. Chai, J. Chem. Phys. **136**, 154104-1 (2012).
[6] J.-D. Chai, J. Chem. Phys. **140**, 18A521-1 (2014).
[7] J.-D. Chai, J. Chem. Phys. **146**, 044102-1 (2017).
[8] M. Filatov and S. Shaik, Chem. Phys. Lett. **288**, 689 (1998).
[9] M. Filatov and S. Shaik, Chem. Phys. Lett. **304**, 429 (1999).
[10] M. Filatov and S. Shaik, J. Chem. Phys. **110**, 116 (1999).
[11] M. Filatov, M. Huix-Rotllant, and I. Burghardt, J. Chem. Phys. **142**, 184104-1 (2015).
[12] M. Filatov, F. Liu, K. S. Kim, and T. J. Martinez, J. Chem. Phys. **145**, 244104-1 (2016).
[13] A. D. Becke and M. R. Roussel, Phys. Rev. A **39**, 3761 (1989).
[14] A. D. Becke, J. Chem. Phys. **119**, 2972 (2003).
[15] A. D. Becke, J. Chem. Phys. **122**, 064101-1 (2005).
[16] A. D. Becke, J. Chem. Phys. **138**, 074109-1 (2013).
[17] A. D. Becke, J. Chem. Phys. **138**, 161101-1 (2013).
[18] E. R. Johnson, J. Chem. Phys. **139**, 074110-1 (2013).
[19] J. Kong and E. Proynov, J. Chem. Theory Comput. **12**, 133 (2016).
[20] S. H. Vosko, L. Wilk, and M. Nusair, Can. J. Phys. **58**, 1200 (1980).
[21] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[22] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, Phys. Rev. Lett. **91**, 146401-1 (2003).
[23] J. Kussmann and C. Ochsenfeld, J. Chem. Phys. **138**, 134114-1 (2013).
[24] J. Kussmann and C. Ochsenfeld, J. Chem. Theory Comput. **11**, 918 (2015).
[25] J. Kussmann and C. Ochsenfeld, J. Chem. Theory Comput. **13**, 3153 (2017).
[26] F. Weigend and R. Ahlrichs, Phys. Chem. Chem. Phys. **7**, 3297 (2005).
[27] N. D. Mermin, Phys. Rev. **137**, A1441 (1965).
[28] S. Grimme and A. Hansen, Angew. Chem., Int. Ed. **54**, 12308 (2015).
[29] D. Robinson, J. Comput. Chem. **34**, 2625 (2013).
[30] B. Mintz, T. G. Williams, L. Howard, and A. K. Wilson, J. Chem. Phys. **130**, 234104-1 (2009).
[31] G. A. Oyedepo and A. K. Wilson, J. Phys. Chem. A **114**, 8806 (2010).
[32] W. Jiang, C. C. Jeffrey, and A. K. Wilson, J. Phys. Chem. A **116**, 9969 (2012).

Supporting information to the paper: Density functional theory model for multi-reference systems based on the exact-exchange hole normalization

Henryk Laqua,[1] Jörg Kussmann,[1] and Christian Ochsenfeld[1, a)]

Department of Chemistry and Center for Integrated Protein Science (CIPSM), University of Munich (LMU), D-81377 München, Germany

(Dated: 8 February 2018)

[a)] Electronic mail: christian.ochsenfeld@uni-muenchen.de

TABLE S1. Fractional spin errors for spin-averaged main group atoms in kcal mol^{-1} employing the def2-TZVP basis and a [99/590]-grid. ME = mean error; MAE = mean absolute error. Spin polarized results are from self-consistent restricted open shell calculations. The spin depolarized results are obtained from the same set of orbitals as the spin polarized results, but with spin-averaged spin densities.

atom	LDA	PBE	TPSS	xhn-LDA	xhn-PBE	xhn-TPSS
H	21.5	26.7	27.1	0.0	0.0	0.0
Li	5.6	6.9	6.3	-0.8	-1.2	-0.9
B	9.3	11.2	14.5	-4.1	-2.0	0.4
C	31.6	34.1	40.5	-8.53	-3.9	-0.7
N	70.9	72.9	83.8	-12.6	-4.3	-0.5
O	43.6	49.7	59.0	-18.0	-6.4	-2.2
F	18.2	23.1	27.3	-15.1	-6.9	-5.4
Na	4.8	5.0	4.7	-1.4	-1.2	-0.6
Al	5.5	7.4	8.5	-3.6	-1.8	-0.9
Si	17.5	21.4	24.1	-8.4	-4.1	-2.7
P	37.8	43.6	48.5	-13.9	-6.4	-4.9
S	22.6	27.6	31.5	-15.4	-8.5	-7.6
Cl	9.3	12.1	14.1	-11.3	-7.2	-7.0
K	3.4	3.5	3.2	-1.6	-1.2	-0.7
Ga	5.3	7.1	7.9	-4.5	-1.9	-1.3
Ge	16.2	20.0	22.3	-10.3	-3.9	-2.8
As	33.0	39.3	43.5	-17.0	-5.8	-4.7
Se	19.1	23.8	26.6	-16.5	-8.2	-8.0
Br	7.6	10.1	11.4	-11.4	-6.9	-7.1
ME (H-Cl)	22.9	26.3	30.0	-8.7	-4.1	-2.5
MAE (H-Cl)	22.9	26.3	30.0	8.7	4.1	2.6
ME (H-Br)	20.1	23.4	26.6	-9.2	-4.3	-3.0
MAE (H-Br)	20.1	23.4	26.6	9.2	4.3	3.1

3.9 Publication IX: Range-Separated Density-Functional Theory in Combination with the Random Phase Approximation: An Accuracy Benchmark

A. Kreppel, D. Graf, H. Laqua, C. Ochsenfeld

J. Chem. Theory Comput. **16**, 2985 (2020).

Abstract

A formulation of range-separated random phase approximation (RPA) based on our efficient ω-CDGD-RI-RPA [*J. Chem. Theory Comput.* **2018**, *14*, 2505] method and a large scale benchmark study are presented. By application to the GMTKN55 data set, we obtain a comprehensive picture of the performance of range-separated RPA in general main group thermochemistry, kinetics, and noncovalent interactions. The results show that range-separated RPA performs stably over the broad range of molecular chemistry included in the GMTKN55 set. It improves significantly over semilocal DFT but it is still less accurate than modern dispersion corrected double-hybrid functionals. Furthermore, range-separated RPA shows a faster basis set convergence compared to standard full-range RPA making it a promising applicable approach with only one empirical parameter.

Reprinted with permission from:

H. Laqua, J. Kussmann, C. Ochsenfeld
"Range-Separated Density-Functional Theory in Combination with the Random Phase Approximation: An Accuracy Benchmark"
J. Chem. Theory Comput. **16**, 2985 (2020).

`https://pubs.acs.org/doi/10.1021/acs.jctc.9b01294`

JCTC Journal of Chemical Theory and Computation

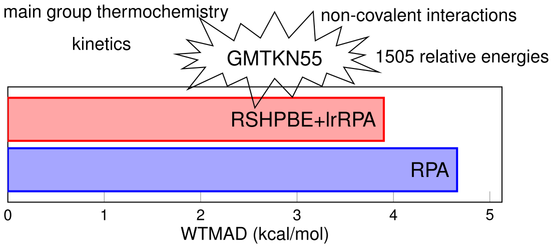
pubs.acs.org/JCTC Article

Range-Separated Density-Functional Theory in Combination with the Random Phase Approximation: An Accuracy Benchmark

Andrea Kreppel, Daniel Graf, Henryk Laqua, and Christian Ochsenfeld*

Cite This: *J. Chem. Theory Comput.* 2020, 16, 2985−2994 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: A formulation of range-separated random phase approximation (RPA) based on our efficient ω-CDGD-RI-RPA [*J. Chem. Theory Comput.* **2018**, *14*, 2505] method and a large scale benchmark study are presented. By application to the GMTKN55 data set, we obtain a comprehensive picture of the performance of range-separated RPA in general main group thermochemistry, kinetics, and noncovalent interactions. The results show that range-separated RPA performs stably over the broad range of molecular chemistry included in the GMTKN55 set. It improves significantly over semilocal DFT but it is still less accurate than modern dispersion corrected double-hybrid functionals. Furthermore, range-separated RPA shows a faster basis set convergence compared to standard full-range RPA making it a promising applicable approach with only one empirical parameter.

1. INTRODUCTION

The random phase approximation (RPA)[1−6] has become an increasingly popular post-Kohn−Sham (KS)[7] approach. RPA can be considered as a parameter-free density functional and it stands on the fifth and highest rung of the Jacob's ladder of density-functional theory (DFT).[8] RPA overcomes several failures of semilocal density functionals, among which one of the most important issues are the poorly described long-range van der Waals interactions.[9] This means that RPA gives more accurate interaction and cohesion energies.[10−14] Even though the long-range part of the dispersion interactions is described well, RPA gives a poor approximation for small interelectronic distances.[3,15−17]

For this reason the idea of treating the short-range interactions with semilocal DFT arose some time ago.[16,18−20] Recently, a scheme that combines the long-range part of the RPA correlation energy with the short-range part of a density functional via the error function has been established.[21−23] This range-separated RPA approach has been shown to improve the RPA correlation energy in various cases. One example is the improvement of dissociation curves for rare-gas dimers and alkaline-earth dimers compared to full-range RPA.[22,23] It also has been shown that the range-separation approach provides accurate interaction energies for a range of noncovalent complexes.[24,25] Furthermore, the range-separation scheme improves atomization energies and barrier heights of small test sets.[26]

Here, we present a range-separated RPA method which is based on our efficient linear-scaling ω-CDGD-RI-RPA method[27−30] in the local atomic orbital space that uses a Cholesky decomposed ground state density (CDGD) and makes use of the resolution-of-the-identity (RI) with the attenuated ω-Coulomb metric.[31] The use of our efficient ω-CDGD-RI-RPA algorithm within the range-separation approach enables us to test range-separated RPA on a large scale and to provide a comprehensive picture of the performance of range-separated RPA. Hence, we compare range-separated RPA to full-range RPA for the GMTKN55 data set.[32] This large benchmark set comprises 1505 relative energies based on 2462 single-point calculations on molecules with up to 72 atoms and gives a broad overview of general main group thermochemistry, kinetics, and noncovalent interactions.

2. THEORY

Several schemes for range-separated RPA have been proposed so far.[22,23,23] The formalism of the range-separation scheme used in this work is described by Toulouse et al. in detail in ref 23. Here, we give a brief overview and rather focus on the description of the long-range formulation of our ω-CDGD-RI-RPA method.[28] In the subsequent description μ, ν, λ, σ refer to atomic orbitals (AOs) i, j and a, b refer to occupied and virtual molecular orbitals (MOs), respectively, and $\underline{i}$, $\underline{j}$ refer to Cholesky orbitals. M, N, P, Q denote auxiliary RI functions. Moreover, Einstein's sum convention[34] is used.

Received: December 29, 2019
Published: April 24, 2020

https://dx.doi.org/10.1021/acs.jctc.9b01294

2.1. Range Separation.

The separation of the electron–electron interaction into long-range (lr) and short-range (sr) contributions can be achieved by dividing the electron–electron operator v_{ee} into a long-range electron–electron operator $v_{\text{ee}}^{\text{lr}}$ and a short-range electron–electron operator $v_{\text{ee}}^{\text{sr}}$ using the error function and its complementary function as

$$v_{\text{ee}} = v_{\text{ee}}^{\text{lr}} + v_{\text{ee}}^{\text{sr}} = \frac{\text{erf}(\mu r_{12})}{r_{12}} + \frac{\text{erfc}(\mu r_{12})}{r_{12}} \tag{1}$$

where the adjustable range-separation parameter μ defines the range of the separation.

Until now, multiple formulations of short-range PBE were presented in the literature.[35−37] In this work the range-separated hybrid PBE functional (RSHPBE) of Goll et al.[38] is used, which utilizes the range-separation scheme in eq 1. A detailed description of this functional is given in ref 38. Its energy

$$E^{\text{RSHPBE}} = E_{\text{H}} + E_{\text{x}}^{\text{PBE,sr}} + E_{\text{x}}^{\text{HF,lr}} + E_{\text{c}}^{\text{PBE,sr}} \tag{2}$$

is composed of the Hartree energy E_{H}, the short-range exchange $E_{\text{x}}^{\text{PBE,sr}}$, and correlation energy $E_{\text{c}}^{\text{PBE,sr}}$ given by the short-range PBE-like functional and the long-range exact exchange energy $E_{\text{x}}^{\text{HF,lr}}$. E^{RSHPBE} lacks long-range correlation effects and thus can be corrected with the long-range part of the RPA correlation energy $E_{\text{c}}^{\text{RPA,lr}}$ in a *post*-KS calculation:

$$E^{\text{RSHPBE+lrRPA}} = E^{\text{RSHPBE}} + E_{\text{c}}^{\text{RPA,lr}} \tag{3}$$

2.2. Long-Range Formulation of the RPA Correlation Energy.

The standard full-range RPA total energy within the adiabatic connection formalism[39] is given by

$$E^{\text{RPA}} = E^{\text{HF}} + E_{\text{c}}^{\text{RPA}} \tag{4}$$

where E^{HF} is the Hartree–Fock energy evaluated non-self-consistently on the reference orbitals and $E_{\text{c}}^{\text{RPA}}$ is the RPA correlation energy. Using the fluctuation–dissipation theorem together with the RI approximation, the RPA correlation energy can be expressed after coupling-strength integration as[4−6]

$$E_{\text{c}}^{\text{RPA}} = \frac{1}{2\pi}\int_0^{+\infty} \text{d}\omega\ \text{Tr}[\ln(1 - \mathbf{X}_0(\text{i}\omega)\mathbf{V}) + \mathbf{X}_0(\text{i}\omega)\mathbf{V}] \tag{5}$$

where

$$V_{MN} = (\mathbf{C}^{-1})_{MP}\tilde{V}_{PQ}(\mathbf{C}^{-1})_{QN} \tag{6}$$

represents the Coulomb operator in the auxiliary basis with

$$C_{MN} = (M|m_{12}|N) \tag{7}$$

$$\tilde{V}_{MN} = (M|v_{\text{ee}}(r_{12})|N) \tag{8}$$

and the RI metric m_{12}. In the presented method the attenuated Coulomb metric

$$m_{12} = \frac{\text{erfc}(\omega_{\text{att}} r_{12})}{r_{12}} \tag{9}$$

with $\omega_{\text{att}} = 0.1\ a_0^{-1}$ is used, since it has been shown to constitute a good trade-off between accuracy and locality for fitting the full-range Coulomb operator.[31] $\mathbf{X}_0$ denotes the noninteracting density–density response function in the zero-temperature case, also represented in the auxiliary basis. For the sake of efficiency, $\mathbf{X}_0$ is calculated in the imaginary time domain

$$X_{0,MN}(\text{i}\tau) = G_{0,\mu\nu}(-\text{i}\tau)B_{\nu\lambda}^{M}G_{0,\lambda\sigma}(\text{i}\tau)B_{\sigma\mu}^{N} \tag{10}$$

where $\mathbf{G}_0(\text{i}\tau)$ is the one-particle Green's function

$$\mathbf{G}_0(\text{i}\tau) = \mathbf{\Theta}(-\text{i}\tau)\underline{\mathbf{G}}_0(\text{i}\tau) + \mathbf{\Theta}(\text{i}\tau)\overline{\mathbf{G}}_0(\text{i}\tau) \tag{11}$$

$$\underline{G}_{0,\mu\nu}(\text{i}\tau) = C_{\mu i}C_{\nu i}\exp(-(\varepsilon_i - \varepsilon_{\text{F}})\tau)$$

$$\overline{G}_{0,\mu\nu}(\text{i}\tau) = -C_{\mu a}C_{\nu a}\exp(-(\varepsilon_a - \varepsilon_{\text{F}})\tau)$$

with the Heaviside step function $\Theta(\text{i}\tau)$, the MO coefficients $C_{\mu i}$ and $C_{\mu a}$, as well as the MO energies ε_i and ε_a of the occupied and unoccupied MOs, respectively, and the Fermi level ε_{F}. The three-center integrals $B_{\mu\nu}^{M}$ are given in Mulliken notation by

$$B_{\mu\nu}^{M} = (\mu\nu|m_{12}|M) \tag{12}$$

The response function of eq 10 is then transformed into the imaginary frequency domain by a contracted double Laplace[27] or, equivalently, cosine[40] transform according to

$$\mathbf{X}_0(\text{i}\omega) = \int_{-\infty}^{+\infty} \text{d}\tau\ \cos(\omega\tau)\mathbf{X}_0(\text{i}\tau) \tag{13}$$

to perform the final frequency integration.

The main drawback of pure AO formulations is the unfavorable scaling with the size of the basis set compared to MO formulations. To address this problem, pivoted Cholesky decomposition[41−43] can be applied to density-type matrices[28,31] in order to obtain local Cholesky vectors/orbitals which can then be used to transform important quantities in the time-determining steps. In the following, pivoted Cholesky decomposition of a given matrix $\mathbf{A}$ is abbreviated by $\mathbf{A} = \mathbf{L}\mathbf{L}^T$.

Since the one-particle Green's function in the negative imaginary time domain is invariant with respect to projection onto the occupied space, eq 10 can equivalently be expressed as

$$X_{0,MN}(\text{i}\tau) = \text{Tr}(\mathbf{PSG}_0(-\text{i}\tau)\mathbf{SPB}^M\mathbf{G}_0(\text{i}\tau)\mathbf{B}^N) \tag{14}$$

Cholesky decomposition of the ground state density matrix $\mathbf{P}$ and cyclic permutation within the trace result in

$$X_{0,MN}(\text{i}\tau) = \text{Tr}(\mathbf{L}^T\mathbf{SG}_0(-\text{i}\tau)\mathbf{SLL}^T\mathbf{B}^M\mathbf{G}_0(\text{i}\tau)\mathbf{B}^N\mathbf{L}) \tag{15}$$

and allow the dimensions of the important quantities to be reduced yielding

$$X_{0,MN}(\text{i}\tau) = G_{0,\underline{j}\,\underline{i}}(-\text{i}\tau)B_{\underline{i}\nu}^{M}G_{0,\nu\mu}(\text{i}\tau)B_{\mu\underline{j}}^{N} \tag{16}$$

where we defined

$$G_{0,\underline{j}\,\underline{i}}(-\text{i}\tau) = (\mathbf{L}^T\mathbf{S})_{\underline{j}\mu}G_{0,\mu\nu}(-\text{i}\tau)(\mathbf{SL})_{\nu\underline{i}} \tag{17}$$

$$B_{\underline{i}\nu}^{M} = (\mathbf{L}^T)_{\underline{i}\mu}B_{\mu\nu}^{M} \tag{18}$$

The final and most expensive step in the calculation of the response function is then given by

$$X_{0,MN}(\text{i}\tau) = B_{\underline{j}\mu}^{M}(\text{i}\tau)B_{\mu\underline{j}}^{N} \tag{19}$$

with

$$B_{\underline{j}\mu}^{M}(\text{i}\tau) = G_{0,\underline{j}\,\underline{i}}(-\text{i}\tau)B_{\underline{i}\nu}^{M}G_{0,\nu\mu}(\text{i}\tau) \tag{20}$$

https://dx.doi.org/10.1021/acs.jctc.9b01294

The evaluation of eq 19 formally scales as $\mathcal{O}(N_{\text{aux}}^2 N_{\text{basis}} N_{\text{occ}})$ but can be implemented in an asymptotically linear-scaling fashion using sparse matrix algebra.

To account for the long-range part of the RPA correlation energy only, as required by the presented range-separated functional, the standard Coulomb operator in eq 8 is substituted by the long-range electron−electron operator defined in eq 1 to obtain

$$\tilde{V}_{MN}^{\text{lr}} = (M|v_{\text{ee}}^{\text{lr}}(r_{12})|N) \tag{21}$$

and hence

$$V_{MN}^{\text{lr}} = (\mathbf{C}^{-1})_{MP}\tilde{V}_{PQ}^{\text{lr}}(\mathbf{C}^{-1})_{QN} \tag{22}$$

This long-range Coulomb operator in the auxiliary basis $\mathbf{V}^{\text{lr}}$ is then used in the final expression for the long-range RPA correlation energy according to

$$E_{\text{c}}^{\text{RPA,lr}} = \frac{1}{2\pi}\int_0^{+\infty} \mathrm{d}\omega\ \text{Tr}[\ln(1 - \mathbf{X}_0(\text{i}\omega)\mathbf{V}^{\text{lr}}) + \mathbf{X}_0\mathbf{V}^{\text{lr}}] \tag{23}$$

In our standard full-range RPA algorithm, the trace of the matrix logarithm is evaluated using Cholesky decomposition of $\mathbf{V}$ in combination with the Mercator series for $\ln(1 + x)$ according to

$$\text{Tr}[\ln(1 + \mathbf{X}_0(\text{i}\omega)\mathbf{V})] = \text{Tr}[\ln(1 + \mathbf{L}^T\mathbf{X}_0(\text{i}\omega)\mathbf{L})] \tag{24}$$

$$= 2\ln\left(\prod_n L'_{nn}\right) \tag{25}$$

where we absorbed the minus sign into the response function and abbreviated the Cholesky decomposition of $\mathbf{1} + \mathbf{L}^T\mathbf{X}_0(\text{i}\omega)\mathbf{L}$ by $\mathbf{L}'$. In the presented range-separated RPA algorithm, Cholesky decomposition of the long-range Coulomb operator $\mathbf{V}^{\text{lr}}$ has turned out to be problematic in some cases due to very small negative eigenvalues occurring as a reason for numerical inaccuracies. Therefore, Cholesky decomposition of $\mathbf{V}^{\text{lr}}$ is avoided by evaluating the trace of the matrix logarithm according to

$$\text{Tr}[\ln(1 + \mathbf{X}_0(\text{i}\omega)\mathbf{V}^{\text{lr}})] = \text{Tr}[\ln(1 + (\mathbf{V}^{\text{lr}})^{1/2}\mathbf{X}_0(\text{i}\omega)(\mathbf{V}^{\text{lr}})^{1/2})] \tag{26}$$

$$= 2\ln\left(\prod_n L_{nn}\right) \tag{27}$$

where this time $\mathbf{L}$ stems from Cholesky-decomposing $1 + (\mathbf{V}^{\text{lr}})^{1/2}\mathbf{X}_0(\text{i}\omega)(\mathbf{V}^{\text{lr}})^{1/2}$. Another alternative avoiding Cholesky decomposition of $\mathbf{V}^{\text{lr}}$ is, of course, to simply evaluate the matrix logarithm via diagonalization, which works in any case but comes along with an increased computational cost.

3. COMPUTATIONAL DETAILS

All calculations were performed using the FermiONs++ program package.[44−46] The self-consistent range-separated hybrid DFT calculations were performed using the short-range PBE functional of ref 38, which was implemented in a development-version of libxc,[47] and long-range exact exchange. This approach is referred to as "RSHPBE" in the following. The long-range RPA correlation correction to the RSHPBE energy was calculated based on these RSHPBE reference orbitals using the long-range formulation of the ω-CDGD-RI-RPA method as described above. This range-separated RPA approach is termed "RSHPBE+lrRPA". For all range-separated calculations a range-separation parameter of $\mu = 0.5\ a_0^{-1}$ was used (see also discussion below), unless stated otherwise. Full-range RPA calculations performed on PBE[48,49] reference orbitals are simply named "RPA" in the following.

All calculations on the GMTKN55 were performed with the Ahlrichs-type split-valence triple-ζ basis set def2-TZVP[50] and the corresponding auxiliary basis set.[51] The basis set was augmented by diffuse functions for the WATER27, G21EA, AHB21, and IL16 subsets in the same way as for the original calculations on the GMTKN55[32] to ensure best possible comparability to already existing results of other density functionals. In the WATER27 test set, Dunning's diffuse s and p functions were applied to oxygen; diffuse s and p functions were applied to non-hydrogen atoms and diffuse s functions to hydrogen in the G21EA, AHB21, and IL16 sets.

Effective-core potentials[50] were used to replace the core electrons of heavy elements in the HEAVYSB11, HEAVY28, and HAL59 subsets.

For all molecules in the singlet state, closed-shell calculations were performed.

4. RESULTS AND DISCUSSION

4.1. Choice of the Basis Set.

Several investigations on the basis set dependence of RSHPBE+lrRPA indicated that within the range-separated framework a smaller number of basis functions is required for convergence of the RPA energy with respect to the basis set size.[22,23,26,52] This convergence behavior is caused by the expected exponential convergence of the long-range part of the RPA correlation energy[53] and the replacement of the relatively slowly converging short-range part of the RPA correlation by faster converging PBE. As the studies concerning basis set behavior of range-separated RPA rely on a small number of molecular systems, we investigate here the basis set convergence of range-separated RPA energies compared to full-range RPA energies using a larger set of molecules.

We compared RSHPBE+lrRPA to full-range RPA on the BH76 (barrier heights), BH76RC (reaction energies), and S22 (noncovalent interactions) test sets for different basis sets (detailed results can be found in the Supporting Information). For full-range RPA a rather pronounced basis set dependence can be observed (Figure 1) as the MAD decreases significantly for each of the three subsets going from the triple- to quadruple-ζ basis. The MADs for RSHPBE+lrRPA, in contrast, vary at most in a range of 0.17 kcal/mol going from def2-TZVP to the larger quadruple-ζ basis set and thus can be considered as sufficiently converged with the def2-TZVP basis set. Further, we want to note that the introduced error by fitting the long-range Coulomb operator with the short-range Coulomb metric is, like for fitting the full-range Coulomb operator, orders of magnitude below the orbital basis set error and the intrinsic error of RPA (see Table S2, Supporting Information). Therefore, the dependence of the results on the quality of the auxiliary basis is assumed to be similar to that of standard RI-RPA which was investigated in ref 5.

Even though the results of full-range RPA are clearly not yet converged with the triple-ζ basis sets, we compare both methods using def2-TZVP as we want to have a fair comparison for practical usage. This means using a basis set that is affordable for many applications. For the performance of

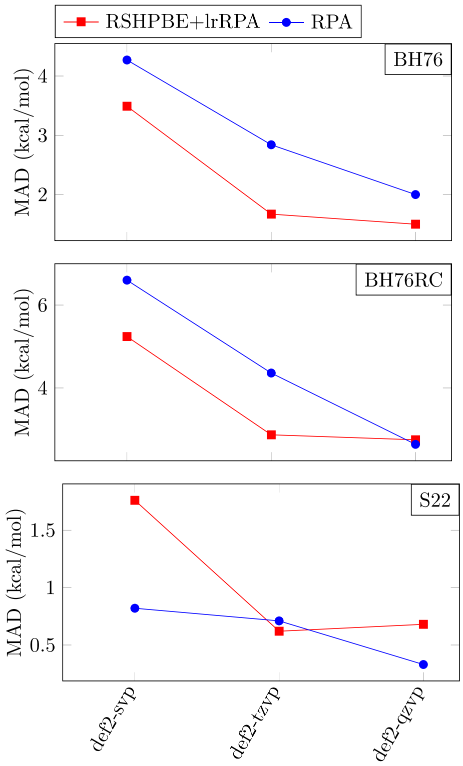


Figure 1. Basis set dependence of the mean absolute deviation (MAD) in kcal/mol for the BH76, BH76RC, and S22 data sets of range-separated RSHPBE+lrRPA (red) and full-range RPA (blue).

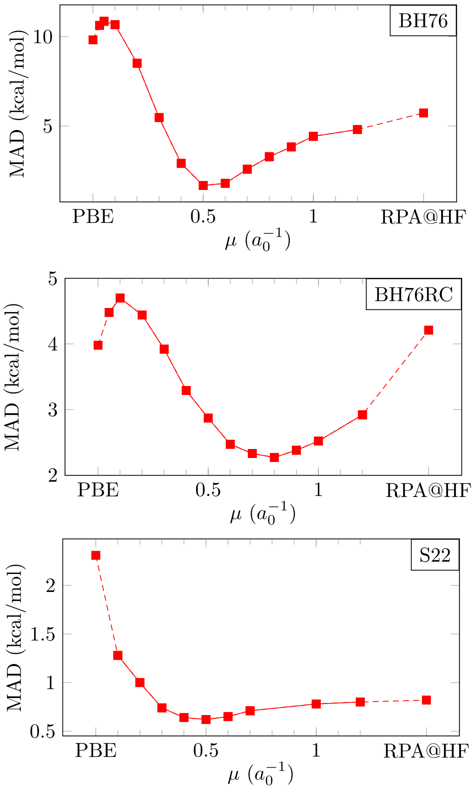


Figure 2. Mean absolute deviation (MAD) for the BH76, BH76RC, and S22 data sets as a function of the range-separation parameter μ for range-separated RSHPBE+lrRPA calculations using the def2-TZVP basis set. In the limit of $\mu \to \infty$ RSHPBE+lrRPA converges to standard RPA evaluated on HF reference orbitals (RPA@HF) and for $\mu \to 0$ it corresponds to PBE.

full-range RPA with larger basis sets we refer the interested reader to already existing benchmarks.[6,54−57]

4.2. Choice of the Range-Separation Parameter.

Prior studies investigating the range-separation parameter μ in range-separated methods revealed that its optimal value lies around 0.5 a_0^{-1}. These prior studies comprise the investigation of the enthalpies of formation for a series of molecules with a combination of srLDA and lrHF exchange[58] and calculations on atomization energy and barrier height data sets with range-separated RPA.[26]

It is worth noting here that in the limit $\mu \to \infty$ the results of RSHPBE+lrRPA do not converge to the results of conventional full-range RPA based on PBE reference orbitals. In fact, the lrRPA$_{\mu\to\infty}$ correlation energy formally corresponds to the full-range formulation, but the RSHPBE (see eq 2) reference orbitals converge to HF orbitals rather than PBE orbitals for $\mu \to \infty$. This means that RSHPBE+lrRPA$_{\mu\to\infty}$ is equal to full-range RPA using HF reference orbitals (RPA@HF, see Figure 2). In the limit of $\mu \to 0$ the lrRPA correlation energy approaches 0. Thus, RSHPBE+lrRPA$_{\mu\to 0}$ approaches the energy of the RSHPBE reference orbitals, which are identical to those of full-range PBE in the case of $\mu \to 0$.

To investigate whether a range-separation parameter of 0.5 a_0^{-1} is indeed an appropriate choice for a broader range of molecules and properties of molecular systems, we complemented these studies by calculations on the BH76RC, BH76, and S22 data sets with varying range-separation parameter in RSHPBE+lrRPA. The results (Figure 2, detailed results can be found in the Supporting Information) reveal that the optimal value for μ slightly varies depending on the examined property or system. While for the BH76 and S22 test sets the optimum of μ lies at 0.5 a_0^{-1}, it is shifted to a slightly higher value of 0.8 a_0^{-1} for the BH76RC test set. A shift of the optimal value of the range-separation parameter to a larger value has also been observed for calculations on reaction energies with a range-separated RPA variant.[59]

Since the results show a quite distinct dependence of the optimal range-separation parameter on the molecular system, we decided to investigate the parameter for an even broader range of molecular systems. We therefore created the set “RAND2x55” which contains two randomly chosen items of each subset of the GMTKN55. The detailed list of contained relative energies can be found in the Supporting Information (Table S1). The absolute values of the relative energies |Δ*E*| contained in this test set vary significantly as these describe completely different chemical properties. Items with larger |Δ*E*| are expected to give a larger absolute deviation, which in turn leads to a larger change between different μ values. In order to consider each item of the RAND2x55 in the same way for obtaining an optimal range-separation parameter, the absolute deviations of every item are weighted using the weighting factors of weighting scheme 1 of ref 32 for the respective subset. The weighted MADs of the RAND2x55 subset show that there is a broad minimum around μ = 0.45 a_0^{-1} (see Figure 3) with a deviation of maximally 0.1 kcal/mol in the MADs over the range μ = 0.4 a_0^{-1} to μ = 0.55 a_0^{-1}. On average, RSHPBE+lrRPA seems to be quite robust with respect to the choice of μ, reassuring us that the

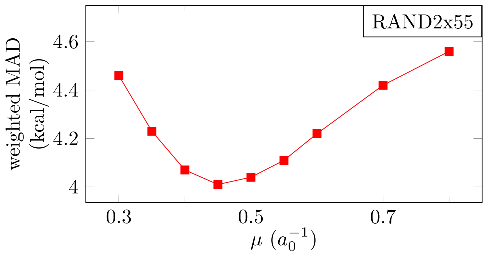


Figure 3. Weighted mean absolute deviation (MAD) for the RAND2×55 data set as a function of the range-separation parameter μ for range-separated RSHPBE+lrRPA calculations using the def2-TZVP basis set.

choice of $\mu = 0.5\ a_0^{-1}$ in previous studies[23,25,26,58] is reasonable. For this reason a range-separation parameter of 0.5 a_0^{-1} was used in the following.

4.3. Results of the GMTKN55 Data Set. The subsets included in the GMTKN55 data set can be grouped into five categories. The first category "basic + small" targets basic properties and reaction energies for small systems. The subsets of the second category "iso + large" comprise reaction energies for large systems and isomerizations. In the third category "barrier", barrier height test sets are united. The last two subcategories "intermol. NCIs" and "intramol. NCIs" focus on inter- and intramolecular noncovalent interactions, respectively.

As shown in Table 1, RSHPBE+lrRPA yields a weighted mean absolute deviation according to weighting scheme 1 of ref 32 (WTMAD-1) of 3.86 kcal/mol for the total GMTKN55 data set. With this result RSHPBE+lrRPA is among the 15% best density functionals tested in ref 32 using the def2-QZVP basis set (see Figure 4) and can be ranked in between the average hybrid and average double-hybrid density functional (see Table 1). It has to be further stressed that the compared (MP2-based) double-hybrid functionals are, due to the inclusion of exchange terms, computationally more expensive than the here presented RPA methods.

Table 1. Comparison of the WTMAD-1 for the GMTKN55 obtained by RSHPBE+lrRPA and Full-Range RPA to Density Functionals Grouped by the Rank of the Jacob's Ladder

RSHPBE+lrRPA	3.86[a]
RPA	4.72[a]
GGA	10.70[b]
meta-GGA	7.31[b]
hybrid	6.56[b]
double-hybrid	3.60[b]

[a]def-TZVP basis set, this work. [b]def2-QZVP basis set and no empirical dispersion correction. Average value taken from ref 32.

The results grouped by category (see Table 2 and Figure 5) show that RSHPBE+lrRPA is not as good as the average double-hybrid density functional for "basic + small" and "barriers" but is significantly better for NCIs. However, the deficiencies of double-hybrid density functionals in describing noncovalent interactions can be compensated by the inclusion of the empirical "D3" dispersion correction of Grimme.[60,61]

RSHPBE+lrRPA gives a slightly better result than full-range RPA (WTMAD-1 of 3.86 kcal/mol vs 4.72 kcal/mol) for the complete GMTKN55 test set. Furthermore, RSHPBE+lrRPA performs more stably over all categories. The WTMAD-1 of RSHPBE+lrRPA is for all categories about the same and does not show as high fluctuations as the full-range variant. In both cases, range-separated and full-range, the RPA correlation energy on average improves the results of the respective Kohn−Sham reference calculations, RSHPBE and PBE.

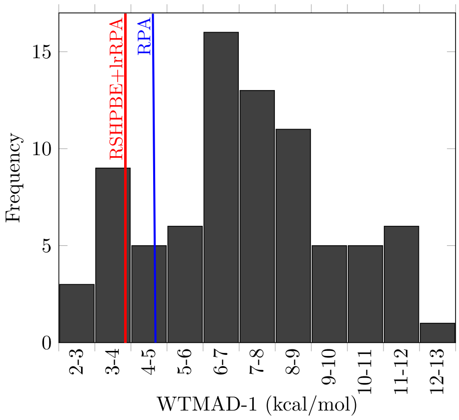


Figure 4. Histogram showing the WTMAD-1 distribution for all tested density functionals without empirical dispersion correction (def2-QZVP) in ref 32 on the total GMTKN55 test set. The red and blue lines illustrate where RSHPBE+lrRPA and full-range RPA def2-TZVP are placed among the density functionals according to the WTMAD-1.

Table 2. WTMAD-1 Values in kcal/mol for the GMTKN55 Test Set and Its Categories[a]

					average double-hybrid	
	RSHPBE	PBE	RSHPBE +lrRPA	RPA	no D3	D3
GMTKN55	8.33	8.17	3.86	4.72	3.60	2.05
basic + small	4.92	5.56	3.48	5.41	2.21	1.87
iso. + large	4.97	7.38	3.76	3.10	3.40	2.50
barriers	5.72	7.64	3.56	2.63	1.43	1.59
intermol. NCIs	13.87	10.41	4.27	6.54	5.90	2.02
intramlo. NCIs	13.13	11.64	4.40	4.16	5.17	2.39
all NCIs	13.55	10.94	4.33	5.52	5.59	2.18

[a]All calculations were performed using the def2-TZVP basis set. Values for the average double-hybrid functional with and without Grimme's D3 dispersion correction[60,61] were obtained using the def2-QZVP basis set and are taken from ref 32.

The improvement of the RPA approaches over the respective Kohn−Sham reference is most prominent for the categories concerning noncovalent interactions. Within the subsets of "intermol. NCIs" and "intramol. NCIs" the improvement is most obvious for the IDISP subset which targets intermolecular dispersion interactions (see Table 3). This is not surprising, as RSHPBE and PBE do not account for any dispersion interactions. Moreover, the remarkably high MAD of RSHPBE+lrRPA for the WATER27 (hydrogen bonds) subset has to be noted. Apparently, this test set is quite sensitive to the basis set size as all tested methods have a significant deviation in the MAD between the def2-TZVP and def2-QZVP results (see Table 3, values in brackets). This means that for this test set the results of all studied methods, including the references RSHPBE and PBE, are not sufficiently

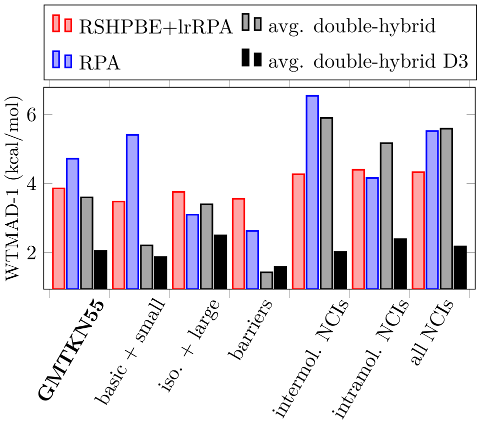


Figure 5. Graphical representation of the WTMAD-1 values for the GMTKN55 test set and its categories. The def2-TZVP basis set was used for RSHPBE+lrRPA and full-range RPA (this work). The average WTMAD-1s for all tested double-hybrid functionals in ref 32 with (avg. double-hybrid D3) and without (avg. double-hybrid) Grimme's D3 dispersion correction[60,61] were obtained using the def2-QZVP basis set and are taken from ref 32.

converged with respect to the basis set size at triple-ζ level and are thus not reliable.

For all noncovalent interactions (all NCIs, Table 2), RSHPBE+lrRPA has a slightly lower WTMAD-1 compared to full-range RPA. This is in line with the observation of Zhu et al.[25] that range-separated RPA improves interaction energies of weakly interacting intermolecular complexes. Also, several studies suggest[17,25,52,62] that a range-separated RPA approach improves interaction energies in rare-gas dimers which we can confirm by the results of the RG18 subset (Table 3).

For reaction barrier heights, varying results for RSHPBE+lrRPA were obtained. In fact, RSHPBE+lrRPA has a slightly lower MAD in some reaction barrier height subsets but has also a remarkably higher MAD for the two subsets, PX13 and WCPT18, containing reaction barriers of proton-transfer and -exchange reactions, where water−water interactions, which are also present in the WATER27 test set, play a crucial role. This suggests that the results of PX13 and WCPT18 might also be not sufficiently converged with respect to the basis set size at the triple-ζ level. This is one of the reasons why we have not observed a significant improvement in the description of reaction barrier heights for RSHPBE+lrRPA over full-range RPA, contradicting the finding of Mussard et al.[26] Another reason might be the larger test volume investigated in our present work.

For the category "iso. + large", a slightly inferior performance of RSHPBE+lrRPA compared to the full-range variant is observed (3.76 kcal/mol vs 3.10 kcal/mol). In this category, the MADs for the MB16-43 (decomposition of artificial molecules) and DARC (Diels−Alder reaction energies) subsets stand out in particular (Table 3). For the DARC test set the difference in the MADs between RSHPBE+lrRPA and full-range RPA is remarkable. It should be noted that the errors for this rather specialized test set are mainly systematic as all relative energies contained in this test set describe one single property: the relative stability of a C−C σ bond vs a C−C π bond. The low MAD of full-range RPA arises from a fortuitous error cancellation for this very specific type of reactions. PBE significantly underestimates the relative stability of C−C σ bonds (signed error +6.12 kcal/mol), and the addition of the full-range RPA correlation compensates this deficiency nearly exactly (signed error +0.48 kcal/mol). In contrast, RSHPBE already overestimates the relative strength of C−C σ bonds (signed error −1.27 kcal/mol), so that the addition of the long-range RPA correlation results in an even stronger comparative overbinding of σ bonds (signed error −6.79 kcal/mol). However, this error is not unusually large compared to other functionals. The average MAD for all double-hybrid functionals tested in ref 32 without empirical dispersion correction is 4.62 kcal/mol. We also tested the influence of the basis set on this specific test set employing the def2-QZVP basis set instead. The differences in the MADs of RSHPBE+lrRPA and full-range RPA, however, were found to be smaller than 1 kcal/mol, i.e., this test set is not dominated by basis set incompleteness errors.

For the MB16-43 test set large MADs are not unusual due to the large average of absolute energy differences $|\overline{\Delta E}|$ of 414.73 kcal/mol. The result of RSHPBE+lrRPA for this test set is as good as the average result of all double-hybrid functionals tested in ref 32 with 22.91 kcal/mol (without empirical dispersion correction). The MAD of full-range RPA, however, is exceptionally large displaying the deficiency of standard full-range RPA to describe the strength of covalent bonds which is well-known concerning atomization energies.[54,63−65]

RSHPBE+lrPBE seems to have an improved performance in basic properties as compared to full-range RPA ("basic + small", Table 2 and Figure 5). This difference in the WTMAD-1s arises from the stable performance of RSHPBE+lrRPA compared to the varying results of standard RPA. Here, especially the noticeable high MADs of the W4-11 (atomization energies), SIE4x4 (self-interaction-error related problems), and ALKBDE10 (dissociation energies of group-1 and -2 diatomics) subsets stand out. The obtained results for the atomization energies subset W4-11 are in line with those of Mussard et al.,[26] who also observed that range-separated RPA gives more precise atomization energies than the full-range variant. It has to be noted that the large MADs of full-range RPA for atomization energies and dissociation energies arise from the systematical underbinding of standard full-range RPA caused by deficiencies in the description of short-range correlation.[54,63,65] The poor performance of standard RPA for the self-interaction-error related problems is also not surprising as it is a well-known deficiency of direct RPA. However, the range-separation approach somewhat alleviates this problem, as indicated by the significantly better performance of RSHPBE+lrRPA in the SIE4x4 test set, confirming the findings of previous work on range-separated RPA.[33,66] In this context, range-separated RPA may also be regarded as a cost-effective alternative to beyond RPA methods.[29,67−72]

5. CONCLUSION

In this work we presented a range-separated RPA method, RSHPBE+lrRPA, based on our efficient linear-scaling ω-CDGD-RI-RPA algorithm.[28] Investigations on the basis set dependence revealed that energies obtained by this range-separated method converge faster with respect to the basis set size than full-range RPA energies. For most systems, RSHPBE+lrRPA yields reliable results with the def2-TZVP basis set. The weaker basis set dependence compared to full-range RPA and the fact that the presented RSHPBE+lrRPA method is exactly as efficient as the underlying ω-CDGD-RI-RPA algorithm opens up the possibility for efficiently applying

https://dx.doi.org/10.1021/acs.jctc.9b01294

Table 3. Detailed List of the Mean Absolute Deviation in kcal/mol for All Subsets of the GMTKN55 Data Base[a]

set	description	RSHPBE	PBE	RSHPBE+lrRPA	RPA
Basic Properties and Reaction Energies for Small Systems					
W4-11[b]	total atomization energies	15.34	14.69	6.94	27.06
G21EA	adiabatic electron affinities	6.43	2.80	3.66	3.39
G21IP	adiabatic ionization potentials	5.09	3.91	4.29	3.41
DIPCS10	double-ionization potentials of closed-shell systems	6.15	4.59	2.94	6.32
PA26	adiabatic proton affinities (incl. of amino acids)	2.53	1.92	1.29	3.88
SIE4x4	self-interaction-error related problems	4.64	23.73	8.63	22.19
ALKBDE10	dissociation energies in group-1 and -2 diatomics	6.19	4.93	4.83	25.00
YBDE18	bond-dissociation energies in ylides	6.99	5.68	2.56	5.28
AL2x6	dimerization energies of AlX_x compounds	6.27	4.04	1.79	2.82
HEAVYSB11	dissociation energies in heavy-element compounds	12.53	4.34	4.97	6.66
NBPRC	oligomerizations and H_2 fragmentation of NH_3/BH_3 systems	2.62	2.77	1.95	2.53
ALK8	dissociation and other reactions of alkaline compounds	7.09	3.05	3.69	7.79
RC21	fragmentations and rearrangements in radical cations	2.71	6.03	4.09	2.79
G2RC	reaction energies of selected G2/97 systems	5.48	7.50	5.67	7.04
BH76RC	reaction energies of the BH76 set	2.38	3.98	2.87	4.51
FH51	reaction energies in various (in-) organic systems	3.27	4.03	3.31	3.40
TAUT15	relative energies in tautomers	1.18	1.91	0.90	1.19
DC13	13 difficult cases for DFT methods	12.76	10.00	8.49	10.47
Reaction Energies for Large Systems and Isomerization Reactions					
MB16-43	decomposition energies of artificial molecules	49.92	24.24	21.72	60.96
DARC	reaction energies of Diels−Alder reactions	1.61	6.39	6.79	0.92
RSE43	radical-stabilization energies	0.46	3.16	0.53	0.48
BSR36	bond-separation reaction of satured hydrocarbons	8.43	8.15	0.90	1.88
CDIE20	double-bond isomerization energies in cyclic systems	1.00	1.90	0.69	0.46
ISO34	isomerization energies of small and medium-sized organic molecules	1.70	1.95	1.51	1.43
ISOL24	isomerization energies of large organic molecules	4.74	6.71	3.79	2.01
C60ISO	relative energies between C_{60} isomers	23.05	10.48	7.55	7.71
PArel	relative energies in protonated isomers	1.05	1.76	1.05	0.97
Reaction Barrier Heights					
BH76	barrier heights of hydrogen transfer, heavy atom transfer, nucleophilic substitution, unimolecular, and association reactions	3.17	9.82	1.67	2.84
BHPERI	barrier heights of pericyclic reactions	10.74	4.18	1.85	0.73
BHDIV10	diverse reaction barrier heights	5.10	8.24	1.39	1.89
INV24	inversion/racemization barrier heights	3.39	2.95	2.11	1.21
BHROT27	barrier heights for rotation around single bonds	0.90	0.54	0.70	0.75
PX13	proton-exchange barriers in H_2O, NH_3, and HF clusters	5.07	13.16	7.67	2.36
WCPT18	proton-transfer barriers in uncatalyzed and water-catalyzed reactions	3.59	9.66	3.19	1.68
Intermolecular Noncovalent Interactions					
RG18	interaction energies in rare-gas complexes	0.51	0.36	0.14	0.41
ADIM6	interaction energies of *n*-alkane dimers	4.54	3.37	1.24	0.30
S22	binding energies of noncovalently bound dimers	3.01	2.31	0.62	0.71
S66	binding energies of noncovalently bound dimers	2.57	1.94	0.72	0.42
HEAVY28	noncovalent interaction energies between heavy element hydrides	1.30	0.49	0.45	0.65
WATER27	binding energies in $(H_2O)_n$, $H^+(H_2O)_n$, and $OH^-(H_2O)_n$	2.27 (5.08)	9.06 (2.84)	11.64 (5.70)	0.89 (3.86)
CARBH12	hydrogen-bonded complexes between carbene analogues and H_2O, NH_3, or HCl	0.63	1.45	0.59	2.07
PNICO23	interaction energies in pnicogen-containing dimers	1.77	0.86	0.53	1.43
HAL59	binding energies in halogenated dimers (incl. halogen bonds)	1.94	1.36	0.37	1.62
AHB21	interaction energies in anion-neutral dimers	1.22	1.10	1.52	1.33
CHB6	interaction energies in cation-neutral dimers	1.76	1.34	1.68	0.87
IL16	interaction energies in anion−cation dimers	4.29	1.77	0.66	0.95
Intramolecular Dispersion Interactions					
IDISP	intramolecular disperison interaction	10.72	10.62	2.81	2.63
ICONF	relative energies in conformers of inorganic systems	0.79	0.41	0.43	0.46
ACONF	Relative energies of alkane conformers	0.92	0.58	0.19	0.06
AMINO20x4	Relative energies in amino acid conformers	0.62	0.47	0.27	0.35

Table 3. continued

set	description	RSHPBE	PBE	RSHPBE+lrRPA	RPA
Intramolecular Dispersion Interactions					
PCONF21	relative energies in tri- and tetrapeptide conformers	3.14	3.20	0.68	1.04
MCONF	relative energies in melatonin conformers	1.83	1.64	0.19	0.64
SCONF	Relative energies of sugar conformers	0.75	0.70	0.49	0.24
UPU23	relative energies between RNA-backbone conformers	2.51	1.87	0.83	0.55
BUT14DIOL	relative energies in butane-1,4-diol conformers	0.19	0.54	0.60	0.14

[a]All calculations were performed using the def2-TZVP basis set. For the WATER27 test set aug-def2-QZVP results are given in brackets. [b]Without the atomization energy of C_2.

range-separated RPA onto relevant systems with several hundred atoms, as illustrated for the ω-CDGD-RI-RPA method in ref 28 where the largest system comprised 902 atoms.

Investigations on the range-separation parameter μ revealed a shallow minimum between 0.4 a_0^{-1} and 0.55 a_0^{-1}, which is in good agreement with previous findings of μ = 0.5 a_0^{-1} to be optimal.[21,23,25,26,38,62]

To give a comprehensive picture of the performance of RSHPBE+lrRPA we compared this method to standard RPA on the GMTKN55 data set[32] and placed it among previously tested density functionals. The results for GMTKN55 show that RSHPBE+lrRPA yields stable results for a broad range of thermochemical and kinetic properties as well as noncovalent interactions. Although the overall performance of RSHPBE+lrRPA is comparable to that of full-range RPA, it shows less variance in the WTMAD-1s of the subcategories. It was found that the range-separation approach especially gives better results compared to those of the full-range variant for atomization energies (W4-11), problems that are prone to the self-interaction-error (SIE4x4), and systems containing group-1 and -2 elements (ALKBDE10, ALK8).

Overall, the results of RSHPBE+lrRPA are promising considering that only one empirical parameter was employed. In the future, the method could further be improved by including exchange into the response function, e.g., along the lines of the second order screened exchange (SOSEX) RPA method.[29,67,68,72] Alternatively, more empirical approaches could be explored in a similar fashion as done by Mardirossian and Head-Gordon,[73] i.e., employing more empirical semilocal exchange-correlation functionals (e.g., B97[74]), more complicated range-separation schemes, or adding empirical dispersion interaction corrections.

Due to the lower computational cost compared to standard MP2 and the stable results of range-separated RPA over a broad range of chemical problems, this avenue is in our opinion worth considering for future developments.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.9b01294.

Details on the "RAND2x55" test set (PDF)

Detailed list of the relative energies for all subsets (XLSX)

■ AUTHOR INFORMATION

Corresponding Author

Christian Ochsenfeld − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany; Max Planck Institute for Solid State Research, D-70569 Stuttgart, Germany*; orcid.org/0000-0002-4189-6558; Email: c.ochsenfeld@fkf.mpg.de

Authors

Andrea Kreppel − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany*

Daniel Graf − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany*

Henryk Laqua − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.9b01294

Funding

The authors acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) within the cluster of excellence (EXC2111) "Munich Center for Quantum Science and Technology", MCQST. C.O. further acknowledges financial support as Max-Planck-Fellow at the MPI-FKF Stuttgart.

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Bohm, D.; Pines, D. A collective description of electron interactions: III. Coulomb interactions in a degenerate electron gas. *Phys. Rev.* **1953**, *92*, 609−625.

(2) Langreth, D. C.; Perdew, J. P. The Exchange-Correlation Energy of a Metallic Surface. *Solid State Commun.* **1975**, *17*, 1425−1429.

(3) Langreth, D. C.; Perdew, J. P. Exchange-correlation energy of a metallic surface: Wave-vector analysis. *Phys. Rev. B* **1977**, *15*, 2884−2901.

(4) Furche, F. Developing the random phase approximation into a practical post-Kohn-Sham correlation model. *J. Chem. Phys.* **2008**, *129*, 114105.

(5) Eshuis, H.; Yarkony, J.; Furche, F. Fast computation of molecular random phase approximation correlation energies using resolution of the identity and imaginary frequency integration. *J. Chem. Phys.* **2010**, *132*, 234114.

(6) Eshuis, H.; Furche, F. Basis set convergence of molecular correlation energy differences within the random phase approximation. *J. Chem. Phys.* **2012**, *136*, 084105.

(7) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133−A1138.

(8) Perdew, J. P.; Schmidt, K. Jacob's ladder of density functional approximations for the exchange-correlation energy. *AIP Conf. Proc.* **2000**, *577*, 1−20.

(9) Andersson, Y.; Langreth, D. C.; Lundqvist, B. I. Van der Waals interactions in density-functional theory. *Phys. Rev. Lett.* **1996**, *76*, 102−105.

(10) Dobson, J. F.; Wang, J. Successful test of a seamless van der Waals density functional. *Phys. Rev. Lett.* **1999**, *82*, 2123−2126.

(11) Dobson, J. F.; Wang, J.; Dinte, B. P.; McLennan, K.; Le, H. M. Soft cohesive forces. *Int. J. Quantum Chem.* **2005**, *101*, 579−598.

(12) Harl, J.; Kresse, G. Cohesive energy curves for noble gas solids calculated by adiabatic connection fluctuation-dissipation theory. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2008**, *77*, 045136.

(13) Nguyen, H. V.; Galli, G. A first-principles study of weakly bound molecules using exact exchange and the random phase approximation. *J. Chem. Phys.* **2010**, *132*, 044109.

(14) Lebègue, S.; Harl, J.; Gould, T.; Ángyán, J. G.; Kresse, G.; Dobson, J. F. Cohesive properties and asymptotics of the dispersion interaction in graphite by the random phase approximation. *Phys. Rev. Lett.* **2010**, *105*, 196401.

(15) Singwi, K. S.; Tosi, M. P.; Land, R. H.; Sjolander, A. Electron Correlations at Metallic Densities. *Phys. Rev.* **1968**, *176*, 589−599.

(16) Kurth, S.; Perdew, J. Density-functional correction of random-phase-approximation correlation with results for jellium surface energies. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1999**, *59*, 10461−10468.

(17) Ángyán, J. G.; Liu, R. F.; Toulouse, J.; Jansen, G. Correlation energy expressions from the adiabatic-connection fluctuation-dissipation theorem approach. *J. Chem. Theory Comput.* **2011**, *7*, 3116−3130.

(18) Perdew, J. P. Local density and gradient-corrected functionals for short-range correlation: Antiparallel-spin and non-RPA contributions. *Int. J. Quantum Chem.* **1993**, *48*, 93−100.

(19) Savin, A. A Combined Density Functional and Configuration Interaction Method. *Int. J. Quantum Chem.* **1988**, *34*, 59−69.

(20) Kohn, W.; Meir, Y.; Makarov, D. E. Van der Waals Energies in Density Functional Theory. *Phys. Rev. Lett.* **1998**, *80*, 4153−4156.

(21) Toulouse, J.; Gerber, I. C.; Jansen, G.; Savin, A.; Ángyán, J. G. Adiabatic-connection fluctuation-dissipation density-functional theory based on range separation. *Phys. Rev. Lett.* **2009**, *102*, 096404.

(22) Janesko, B. G.; Henderson, T. M.; Scuseria, G. E. Long-range-corrected hybrids including random phase approximation correlation. *J. Chem. Phys.* **2009**, *130*, 081105.

(23) Toulouse, J.; Zhu, W.; Angyan, J. G.; Savin, A. Range-separated density-functional theory with random phase approximation: detailed formalism and illustrative applications. *Phys. Rev. A: At., Mol., Opt. Phys.* **2010**, *82*, 032502.

(24) Janesko, B. G.; Henderson, T. M.; Scuseria, G. E. Long-range-corrected hybrid density functionals including random phase approximation correlation: Application to noncovalent interactions. *J. Chem. Phys.* **2009**, *131*, 034110.

(25) Zhu, W.; Toulouse, J.; Savin, A.; Ángyán, J. G. Range-separated density-functional theory with random phase approximation applied to noncovalent intermolecular interactions. *J. Chem. Phys.* **2010**, *132*, 244108.

(26) Mussard, B.; Reinhardt, P.; Ángyán, J. G.; Toulouse, J. Spin-unrestricted random-phase approximation with range separation: Benchmark on atomization energies and reaction barrier heights. *J. Chem. Phys.* **2015**, *142*, 154123.

(27) Schurkus, H. F.; Ochsenfeld, C. Communication: An effective linear-scaling atomic-orbital reformulation of the random-phase approximation using a contracted double-Laplace transformation. *J. Chem. Phys.* **2016**, *144*, 031101.

(28) Graf, D.; Beuerle, M.; Schurkus, H. F.; Luenser, A.; Savasci, G.; Ochsenfeld, C. Accurate and Efficient Parallel Implementation of an Effective Linear-Scaling Direct Random Phase Approximation Method. *J. Chem. Theory Comput.* **2018**, *14*, 2505−2515.

(29) Beuerle, M.; Graf, D.; Schurkus, H. F.; Ochsenfeld, C. Efficient calculation of beyond RPA correlation energies in the dielectric matrix formalism. *J. Chem. Phys.* **2018**, *148*, 204104.

(30) Graf, D.; Beuerle, M.; Ochsenfeld, C. Low-Scaling Self-Consistent Minimization of a Density Matrix Based Random Phase Approximation Method in the Atomic Orbital Space. *J. Chem. Theory Comput.* **2019**, *15*, 4468−4477.

(31) Luenser, A.; Schurkus, H. F.; Ochsenfeld, C. Vanishing-Overhead Linear-Scaling Random Phase Approximation by Cholesky Decomposition and an Attenuated Coulomb-Metric. *J. Chem. Theory Comput.* **2017**, *13*, 1647−1655.

(32) Goerigk, L.; Hansen, A.; Bauer, C.; Ehrlich, S.; Najibi, A.; Grimme, S. A look at the density functional theory zoo with the advanced GMTKN55 database for general main group thermochemistry, kinetics and noncovalent interactions. *Phys. Chem. Chem. Phys.* **2017**, *19*, 32184−32215.

(33) Mezei, P. D.; Kállay, M. Construction of a Range-Separated Dual-Hybrid Direct Random Phase Approximation. *J. Chem. Theory Comput.* **2019**, *15*, 6678−6687.

(34) Einstein, A. Die Grundlage der allgemeinen Relativitätstheorie. *Ann. Phys.* **1916**, *354*, 769−822.

(35) Iikura, H.; Tsuneda, T.; Yanai, T.; Hirao, K. A long-range correction scheme for generalized-gradient-approximation exchange functionals. *J. Chem. Phys.* **2001**, *115*, 3540−3544.

(36) Toulouse, J.; Colonna, F.; Savin, A. Short-range exchange and correlation energy density functionals: Beyond the local-density approximation. *J. Chem. Phys.* **2005**, *122*, 014110.

(37) Laikov, D. N. Simple exchange hole models for long-range-corrected density functionals. *J. Chem. Phys.* **2019**, *151*, 094106.

(38) Goll, E.; Werner, H.-J.; Stoll, H.; Leininger, T.; Gori-giorgi, P.; Savin, A. A short-range gradient-corrected spin density functional in combination with long-range coupled-cluster methods: Application to alkali-metal rare-gas dimers. *Chem. Phys.* **2006**, *329*, 276−282.

(39) Gunnarsson, O.; Lundqvist, B. I. Exchange and correlation in atoms, molecules, and solids by the spin-density-functional formalism. *Phys. Rev. B* **1976**, *13*, 4274−4298.

(40) Kaltak, M.; Klimeš, J.; Kresse, G. Low scaling algorithms for the random phase approximation: Imaginary time and Laplace transformations. *J. Chem. Theory Comput.* **2014**, *10*, 2498.

(41) Koch, H.; Sánchez De Merás, A.; Pedersen, T. B. Reduced scaling in electronic structure calculations using Cholesky decompositions. *J. Chem. Phys.* **2003**, *118*, 9481−9484.

(42) Higham, N. J. Cholesky factorization. *Wiley Interdiscip. Rev. Comput. Stat.* **2009**, *1*, 251−254.

(43) Harbrecht, H.; Peters, M.; Schneider, R. On the low-rank approximation by the pivoted Cholesky decomposition. *Appl. Numer. Math.* **2012**, *62*, 428−440.

(44) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, 134114.

(45) 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.

(46) Kussmann, J.; Ochsenfeld, C. Hybrid CPU/GPU Integral Engine for Strong-Scaling Ab Initio Methods. *J. Chem. Theory Comput.* **2017**, *13*, 3153−3159.

(47) Marques, M. A. L.; Oliveira, M. J. T.; Burnus, T. Libxc: a library of exchange and correlation functionals for density functional theory. *Comput. Phys. Commun.* **2012**, *183*, 2272.

(48) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(49) Perdew, J. P.; Ernzerhof, M.; Burke, K. (ERRATA) Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(50) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.

(51) Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. RI-MP2: optimized auxiliary basis sets and demonstration of efficiency. *Chem. Phys. Lett.* **1998**, *294*, 143−152.

(52) Irelan, R. M.; Henderson, T. M.; Scuseria, G. E. Long-range-corrected hybrids using a range-separated Perdew-Burke-Ernzerhof

functional and random phase approximation correlation. *J. Chem. Phys.* **2011**, *135*, 094105.

(53) Franck, O.; Mussard, B.; Luppi, E.; Toulouse, J. Basis convergence of range-separated density-functional theory. *J. Chem. Phys.* **2015**, *142*, 074107.

(54) Eshuis, H.; Bates, J. E.; Furche, F. Electron correlation methods based on the random phase approximation. *Theor. Chem. Acc.* **2012**, *131*, 1084.

(55) Ren, X.; Rinke, P.; Scuseria, G. E.; Scheffler, M. Renormalized second-order perturbation theory for the electron correlation energy: Concept, implementation, and benchmarks. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *88*, 035120.

(56) Ruzsinszky, A.; Zhang, I. Y.; Scheffler, M. Insight into organic reactions from the direct random phase approximation and its corrections. *J. Chem. Phys.* **2015**, *143*, 144115.

(57) Grimme, S.; Steinmetz, M. A computationally efficient double hybrid density functional based on the random phase approximation. *Phys. Chem. Chem. Phys.* **2016**, *18*, 20926−20937.

(58) Gerber, I. C.; Ángyán, J. G. Hybrid functional with separated range. *Chem. Phys. Lett.* **2005**, *415*, 100−105.

(59) Heßelmann, A.; Ángyán, J. Assessment of a range-separated orbital-optimized random-phase approximation electron correlation method. *Theor. Chem. Acc.* **2018**, *137*, 155.

(60) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *J. Chem. Phys.* **2010**, *132*, 154104.

(61) Grimme, S.; Ehrlich, S.; Goerigk, L. Effect of the Damping Function in Dispersion Corrected Density Functional Theory. *J. Comput. Chem.* **2011**, *32*, 1456−1465.

(62) Toulouse, J.; Zhu, W.; Savin, A.; Jansen, G.; Ángyán, J. G. Closed-shell ring coupled cluster doubles theory with range separation applied on weak intermolecular interactions. *J. Chem. Phys.* **2011**, *135*, 084119.

(63) Furche, F. Molecular tests of the random phase approximation to the exchange-correlation energy functional. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2001**, *64*, 195120.

(64) Dobson, J. F.; Gould, T. Calculation of dispersion energies. *J. Phys.: Condens. Matter* **2012**, *24*, 073201.

(65) Chen, G. P.; Voora, V. K.; Agee, M. M.; Balasubramani, S. G.; Furche, F. Random-Phase Approximation Methods. *Annu. Rev. Phys. Chem.* **2017**, *68*, 421−445.

(66) Mezei, P. D.; Csonka, G. I.; Ruzsinszky, A.; Kállay, M. Construction and Application of a New Dual-Hybrid Random Phase Approximation. *J. Chem. Theory Comput.* **2015**, *11*, 4615−4626.

(67) Freeman, D. L. Coupled-cluster expansion applied to the electron gas: Inclusion of ring and exchange effects. *Phys. Rev. B* **1977**, *15*, 5512−5521.

(68) Grüneis, A.; Marsman, M.; Harl, J.; Schimka, L.; Kresse, G. Making the random phase approximation to electronic correlation accurate. *J. Chem. Phys.* **2009**, *131*, 154115.

(69) Bates, J. E.; Furche, F. Communication: Random phase approximation renormalized many-body perturbation theory. *J. Chem. Phys.* **2013**, *139*, 171103.

(70) Mussard, B.; Rocca, D.; Jansen, G.; Ángyán, J. G. Dielectric Matrix Formulation of Correlation Energies in the Random Phase Approximation: Inclusion of Exchange Effects. *J. Chem. Theory Comput.* **2016**, *12*, 2191−2202.

(71) Dixit, A.; Ángyán, J. G.; Rocca, D. Improving the accuracy of ground-state correlation energies within a plane-wave basis set: The electron-hole exchange kernel. *J. Chem. Phys.* **2016**, *145*, 104105.

(72) Beuerle, M.; Ochsenfeld, C. Short-range second order screened exchange correction to RPA correlation energies. *J. Chem. Phys.* **2017**, *147*, 204107.

(73) Mardirossian, N.; Head-Gordon, M. Survival of the most transferable at the top of Jacob's ladder: Defining and testing the ω B97M(2) double hybrid density functional. *J. Chem. Phys.* **2018**, *148*, 241736.

(74) Becke, A. D. Density-functional thermochemistry. V. Systematic optimization of exchange- correlation functionals. *J. Chem. Phys.* **1997**, *107*, 8554−8560.

Supporting Information for: Range-Separated Density-Functional Theory in Combination with the Random Phase Approximation: An Accuracy Benchmark

Andrea Kreppel,[†] Daniel Graf,[†] Henryk Laqua,[†] and Christian Ochsenfeld[*,†,‡]

[†]*Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), D-81377 Munich, Germany*

[‡]*Max Planck Institute for Solid State Research, Heisenbergstr. 1, D-70569 Stuttgart, Germany*

E-mail: c.ochsenfeld@fkf.mpg.de

The detailed results of RSHPBE, PBE, RSHBPE+lrRPA, and full-range RPA on the GMTKN55 obtained in this work can be found in the enclosed file 'GMTKN55.xlsx'.

Table S1: Detailed list of the RAND2x55 test set. For each item its number in the original subset (#) is given. The system names correspond to the geometry files of the corresponding test set. The reference values are given in kcal/mol. In the last column, the weighting factor of the corresponding test set in the WTMAD-1 scheme is given.

subset	#	systems	stoichiometry	ref.	$w1$
W4-11	8	sih si h	-1 1 1	73.921	0.1
W4-11	90	hocl h o cl	-1 1 1 1	166.229	0.1
G21EA	20	EA_20n EA_20	1 -1	9.5	1
G21EA	2	EA_o EA_o-	1 -1	33.7	1
G21IP	36	IP_80 48	1 -1	261.153	0.1
G21IP	22	IP_65 IP_n65	1 -1	234.107	0.1
DIPCS10	2	c2h6 c2h6_2+	-1 1	667.1	0.1
DIPCS10	7	h2s h2s_2+	-1 1	733	0.1
PA26	15	ch3cooh ch3coohp	1 -1	190.9	0.1
PA26	10	h2s h2sp	1 -1	174.3	0.1
SIE4x4	5	he he+ he2+_1.0	1 1 -1	56.9	1
SIE4x4	8	he he+ he2+_1.75	1 1 -1	19.1	1
ALKBDE10	2	beo be o	-1 1 1	106.6	0.1
ALKBDE10	7	lio li o	-1 1 1	82.5	0.1
YBDE18	6	me2s-ch2 me2s ch2	-1 1 1	51.74	1
YBDE18	16	ph3-ch2 ph3 ch2	-1 1 1	60.11	1
AL2x6	4	al2me4 alme2	-1 2	38.4	1
AL2x6	3	al2cl6 alcl3	-1 2	32.5	1
HEAVYSB11	11	br br2	2 -1	53.17	1
HEAVYSB11	4	sh h2s2	2 -1	67.85	1
NBPRC	7	BH3PH3 BH3 PH3	1 -1 -1	-25.2	1
NBPRC	5	nh2-bh2 bz h2	-3 1 3	-48.9	1

ALK8	6	li5_ch li4_c li_h	-1 1 1	66.28	1
ALK8	2	na8 na2	-1 4	53.15	1
RC21	5	3e 3p1 3p2	-1 1 1	57.93	1
RC21	13	6e 6p1 ethylene	-1 1 1	21.21	1
G2RC	10	58 59 57 60	-1 -1 1 1	-27.15	1
G2RC	6	128 13 126 22	-1 -1 1 1	-10.7	1
BH76RC	29	C2H6 NH2 C2H5 NH3	-1 -1 1 1	-6.52	1
BH76RC	13	hnc hcn	-1 1	-15.06	1
FH51	13	2-pentyne H2 trans-2-pentene	-1 -1 1	-44.82	1
FH51	4	C4H9SO2H H2O2 C4H9SO3H H2O	-1 -1 1 1	-82.55	1
TAUT15	9	6a 6b	-1 1	-0.17	10
TAUT15	10	6a 6c	-1 1	-0.87	10
DC13	12	o3 c2h4 o3_c2h4_add	-1 -1 1	-58.7	1
DC13	2	c20cage c20bowl	-1 1	-7.7	1
MB16-43	13	13 H2 CH4 N2 O2 MgH2 S2	-2 -5 4 4 2 2 2	19.8751	0.1
MB16-43	32	32 H2 LiH BH3 N2 F2 AlH3 SiH4 S2	-2 -11 2 6 1 2 2 2 1	685.5818	0.1
DARC	6	ethine chdiene P6	-1 -1 1	-49	1
DARC	3	ethene cpdiene P3	-1 -1 1	-29.9	1
RSE43	42	E44 P1 E1 P44	-1 -1 1 1	-6.7	1
RSE43	13	E15 P1 E1 P15	-1 -1 1 1	-6.4	1
BSR36	26	c2h6 r11 ch4	11 -1 -12	8.93	1
BSR36	21	c2h6 r6 ch4	7 -1 -7	9.78	1
CDIE20	6	R28 P26	-1 1	4	10
CDIE20	20	R60 P60	-1 1	8.6	10
ISO34	20	E20 P20	-1 1	18.12	1
ISO34	24	E24 P24	-1 1	12.26	1

ISOL24	24	i24e i24p	-1 1	15.4	1
ISOL24	9	i9e i9p	-1 1	21.09	1
C60ISO	8	1 9	-1 1	143.96	0.1
C60ISO	7	1 8	-1 1	142.18	0.1
PArel	19	c2cl43 c2cl42	-1 1	2.47	10
PArel	12	sugar0 sugar3	-1 1	3.21	10
BH76	75	C5H8 RKT22	-1 1	39.7	1
BH76	63	h H2S RKT16	-1 -1 1	3.9	1
BHPERI	15	13r_5 13_c2h4 13ts_5a	-1 -1 1	6.5	1
BHPERI	26	09r 00r 09ts	-1 -1 1	31.3	1
BHDIV10	1	ed1 ts1	-1 1	25.65	1
BHDIV10	5	ed5 ts5	-1 1	15.94	1
INV24	3	SO2 SO2_TS	-1 1	60.6	1
INV24	12	Dibenzocycloheptene Dibenzocycloheptene_TS	-1 1	10.3	1
BHROT27	24	ethylthiourea_180 ethylthiourea_TS1	-1 1	10.36	10
BHROT27	22	butadiene_strans butadiene_TS	-1 1	6.3	10
PX13	6	h2o_4 h2o_4_ts	-1 1	26.6	1
PX13	9	hf_2 hf_2_ts	-1 1	42.3	1
WCPT18	8	reac8 ts8	-1 1	28.97	1
WCPT18	7	reac7 ts7	-1 1	32	1
RG18	15	c2h6Ne ne c2h6	-1 1 1	0.24	10
RG18	17	bzNe ne bz	-1 1 1	0.4	10
ADIM6	5	AM6 AD6	2 -1	4.6	10
ADIM6	6	AM7 AD7	2 -1	5.55	10
S22	10	10 10a 10b	-1 1 1	1.448	10
S22	7	7 07a 07b	-1 1 1	16.66	10

S66	2	02A 02B 2	1 1 -1	5.59	10
S66	53	53A 53B 53	1 1 -1	4.36	10
HEAVY28	21	sbh3_nh3 sbh3 nh3	-1 1 1	2.84	10
HEAVY28	11	pbh4_hcl pbh4 hcl	-1 1 1	0.75	10
WATER27	20	OHmH2O OHm H2O	-1 1 1	26.687	0.1
WATER27	3	H2O4 H2O	-1 4	27.353	0.1
CARBH12	1	1O 1O_A 1O_B	-1 1 1	5.37	10
CARBH12	10	2CL 2CL_A 2CL_B	-1 1 1	10.483	10
PNICO23	5	5 5a 5b	-1 1 1	2.86	10
PNICO23	1	1 1a p1b	-1 1 1	1.43	10
HAL59	32	BrBr_FCCH BrBr FCCH	-1 1 1	0.74	10
HAL59	38	BrBr_OCH2 BrBr OCH2	-1 1 1	4.41	10
AHB21	15	15 15A 15B	1 -1 -1	-8.62	1
AHB21	5	5 5A 5B	1 -1 -1	-15.61	1
CHB6	6	27 27A 27B	1 -1 -1	-19.9	1
CHB6	3	24 24A 24B	1 -1 -1	-17.83	1
IL16	1	008 008A 008B	1 -1 -1	-100.41	0.1
IL16	7	187 187A 187B	1 -1 -1	-114	0.1
IDISP	1	antdimer ant	1 -2	-9.15	1
IDISP	4	undecan1 undecan2	1 -1	9.1	1
ICONF	3	N4H6_1 N4H6_2	-1 1	0.13	10
ICONF	4	N4H6_1 N4H6_3	-1 1	2.33	10
ACONF	8	H_ttt H_gtg	-1 1	1.178	10
ACONF	11	H_ttt H_g+x-t+	-1 1	2.632	10
AMINO20x4	59	PRO_xae PRO_xaf	-1 1	4.187	10
AMINO20x4	66	THR_xaq THR_xag	-1 1	3.08	10

PCONF	15	SER_ab SER_aR	-1 1	1.47	10
PCONF	7	99 412	-1 1	2.18	10
MCONF	51	1 52	-1 1	8.75	10
MCONF	41	1 42	-1 1	6.39	10
SCONF	8	C1 C9	-1 1	6.19	10
SCONF	1	C1 C2	-1 1	0.86	10
UPU23	2	2p u1b	-1 1	2.97	10
UPU23	17	2p 7p	-1 1	3.9	10
BUT14DIOL	45	B1 B46	-1 1	3.18	10
BUT14DIOL	21	B1 B22	-1 1	2.74	10

Table S2: Comparison of the WTMAD-1 (kcal/mol) for the RAND2x55 test set using the attenuated Coulomb metric ($\omega = 0.1$) and the standard Coulomb metric to fit the long-range Coulomb operator in the auxiliary basis for two different range-separation values.

μ	ω-Coulomb	Coulomb	Δ
0.45	4.00618	4.00590	-2.74E-04
0.5	4.04240	4.04217	-2.37E-04

3.10 Publication X: Lagrangian-Based Minimal-Overhead Batching Scheme for the Efficient Integral-Direct Evaluation of the RPA Correlation Energy

V. Drontschenko, D. Graf, H. Laqua, C. Ochsenfeld

J. Chem. Theory Comput. **17**, 5623 (2021).

Abstract

A highly memory-efficient integral-direct random phase approximation (RPA) method based on our ω-CDGD-RI-RPA method [Graf, D. et al. *J. Chem. Theory Comput.* **2018**, *14*, 2505] is presented that completely alleviates the memory bottleneck of storing the multidimensional three-center integral tensor, which severely limited the tractable system sizes. Based on a Lagrangian formulation, we introduce an optimized batching scheme over the auxiliary and basis-function indices, which allows to compute the optimal number of batches for a given amount of system memory, while minimizing the batching overhead. Thus, our optimized batching constitutes the best tradeoff between program runtime and memory demand. Within this batching scheme, the half-transformed three-center integral tensor $B_{i\mu}^{M}$ is recomputed for each batch of auxiliary and basis functions. This allows the computation of systems that were out of reach before. The largest system within this work consists of a DNA fragment comprising 1052 atoms and 11 230 basis functions calculated on a single node, which emphasizes the new possibilities of our integral-direct RPA method.

Reprinted with permission from:

V. Drontschenko, D. Graf, H. Laqua, C. Ochsenfeld
"Lagrangian-Based Minimal-Overhead Batching Scheme for the Efficient Integral-Direct Evaluation of the RPA Correlation Energy"
J. Chem. Theory Comput. **17**, 5623 (2021).

https://pubs.acs.org/doi/10.1021/acs.jctc.1c00494

JCTC Journal of Chemical Theory and Computation

Lagrangian-Based Minimal-Overhead Batching Scheme for the Efficient Integral-Direct Evaluation of the RPA Correlation Energy

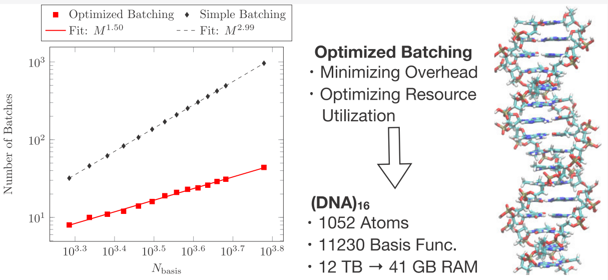
Viktoria Drontschenko, Daniel Graf, Henryk Laqua, and Christian Ochsenfeld*

Cite This: *J. Chem. Theory Comput.* 2021, 17, 5623−5634

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: A highly memory-efficient integral-direct random phase approximation (RPA) method based on our ω-CDGD-RI-RPA method [Graf, D. et al. *J. Chem. Theory Comput.* **2018**, *14*, 2505] is presented that completely alleviates the memory bottleneck of storing the multidimensional three-center integral tensor, which severely limited the tractable system sizes. Based on a Lagrangian formulation, we introduce an optimized batching scheme over the auxiliary and basis-function indices, which allows to compute the optimal number of batches for a given amount of system memory, while minimizing the batching overhead. Thus, our optimized batching constitutes the best tradeoff between program runtime and memory demand. Within this batching scheme, the half-transformed three-center integral tensor $B^M_{i\mu}$ is recomputed for each batch of auxiliary and basis functions. This allows the computation of systems that were out of reach before. The largest system within this work consists of a DNA fragment comprising 1052 atoms and 11 230 basis functions calculated on a single node, which emphasizes the new possibilities of our integral-direct RPA method.



1. INTRODUCTION

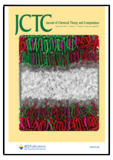
Density-functional theory (DFT) has become one of the most applied theoretical techniques for electronic structure calculations of molecules,[1−3] surfaces,[4−6] and crystals[7−9] in the fields of solid-state physics, computational chemistry, and materials science.[10] Its remarkable success can be largely attributed to the excellent cost performance ratios and good accuracies for various properties and compounds, which make DFT applicable to systems containing up to several thousand atoms.[11,12] However, despite the vast benefits of DFT, it is subject to several well-known deficiencies. The accurate description of long-range electron correlation, particularly including van der Waals (vdW) interactions, represents a challenging task in the modeling of molecules and materials.[11,13−18] This makes the development of more broadly applicable correlation models a necessity.

The random phase approximation (RPA) is one of the most promising methods to obtain accurate correlation energies,[19,20] which is reflected by the increased interest over the last decades.[13,21−34] It yields a good description of bonding types, including covalent, ionic, and metallic bonding.[19] Additionally, due to its nonlocality, RPA correlation is able to describe vdW interactions exceptionally well.[35]

RPA is usually implemented as a post-Kohn−Sham method[36] and was first introduced by Bohm and Pines in 1953.[37] It was later formulated within the framework of DFT using the adiabatic-connection fluctuation-dissipation theorem.[20,38,39] However, in its original formulation, the calculation of RPA correlation energies scales as $\mathcal{O}(M^6)$ with the system size M, limiting its applicability to systems comprising only tens of atoms. In 2010, Furche and co-workers introduced the resolution-of-the-identity (RI) approximation to RPA, reducing the scaling to $\mathcal{O}(M^4)$.[27,40,41] This opened the way for applications beyond the few atoms scale. In 2016, Schurkus and Ochsenfeld[32] reformulated the RPA correlation energy in the atomic orbital (AO) space, thus extending the applicability of the RPA to molecules comprising thousands of atoms. Further improvements were introduced by Luenser et al.[33] and later by Graf et al.[34] by employing an attenuated Coulomb metric, Cholesky decomposition of the ground-state density matrix, and an improved quadrature for the cosine transform in the framework of the ω-CDGD-RI-RPA method.

Within the ω-CDGD-RI-RPA method, the most demanding step regarding the computational effort and memory requirements constitutes the calculation of the response function in the auxiliary basis. For this step, the half-transformed three-center integral tensor $B^M_{i\mu}$, whose storage requirements formally scale as $\mathcal{O}(N_{\text{aux}}N_{\text{basis}}N_{\text{occ}})$ with the number of auxiliary

Received: May 18, 2021
Published: August 25, 2021



https://doi.org/10.1021/acs.jctc.1c00494

functions N_{aux}, AO basis functions N_{basis}, and occupied molecular orbitals (MOs) N_{occ}, has to be stored in memory. Consequently, when approaching large systems, the memory requirement of the three-center integral tensor easily exceeds the available system memory on a single computing node, thereby severely limiting the tractable system sizes. This memory bottleneck was previously addressed by Graf et al.[34] utilizing a hybrid parallelization scheme, thus reducing the memory requirements of the three-center integral tensor per node. However, depending on the targeted system size, this requires medium to large computing clusters. To avoid these demanding computing requirements, a method utilizing a single server or workstation, which is readily available in research groups, is desirable.

In this work, we introduce a Lagrange formulation for a minimal batching overhead and an optimal exploitation of computing resources. The memory-efficient integral-direct RPA method completely eliminates the storage bottleneck of the three-center integral tensor by computing the response function within an optimized batching scheme over both the auxiliary and AO basis-function index at the same time. The available system memory is utilized in the most efficient way and with minimal overhead. The three-center integral tensor is recomputed and transformed "on the fly" for only the respective batch (integral-direct), thereby reducing its memory requirement by a factor of $b_{\text{aux}}b_{\text{AO}}$, where b_{aux} denotes the number of auxiliary function batches and b_{AO} the number of AO basis-function batches. This redundant on the fly recomputation comes, however, at the cost of an increased program runtime. Hence, a compromise between memory demand and program runtime has to be made. In this context, the here presented optimized batching represents, by design, the optimal compromise requiring the smallest amount of recomputation, and consequently the lowest runtime, for any given amount of available system memory. In this way, our integral-direct RPA implementation extends the applicability of RPA to considerably larger systems.

This work is structured as follows: We begin with a brief review of the ω-CDGD-RI-RPA method in Section 2. Next, we derive a batching method for the calculation of the response function in Section 3. In this regard, we begin with the trivial approach of batching with respect to auxiliary functions in Section 3.1 and subsequently extend this batching scheme by additionally including batching over the AO basis functions and Laplace quadrature points in Section 3.2, where we arrive at the optimal batching formalism. We proceed to compare both batching methods in Section 3.3 and present calculations to support our considerations. In Section 4, we first establish why integral-direct RPA is best suited to reach very large systems by addressing two approaches typically used for the assessment of large integral tensors such as the three-center integrals, namely, the integral-direct approach and retrieving the three-center integrals from disk in Section 4.1. In Section 4.2, the scaling for integral-direct RPA is analyzed and systematically improved using shell pair and integral screening methods, sparse matrix algebra as well as switching from the Coulomb metric to the Coulomb metric attenuated by a complementary error function. Furthermore, calculations are presented to support our theoretical considerations. Computational details are given in Section 5 and the performance of our integral-direct RPA implementation is evaluated for chemically relevant systems in Section 6. Finally, the conclusion and outlook are presented in Section 7.

2. ω-CDGD-RI-RPA THEORY

In this section, we intend to give a brief overview of the theory underlying the ω-CDGD-RI-RPA method.[34] For a more detailed derivation, we refer the reader to previous publications.[27,32−34,40,41]

Throughout, the following notation has been adopted: μ, ν, λ, and σ denote atomic orbitals (AOs); i and j refer to occupied molecular orbitals (MOs); a and b refer to virtual MOs; $\underline{i}$ and $\underline{j}$ denote Cholesky orbitals, and M, N, P, and Q denote auxiliary functions. The number of auxiliary functions is represented by N_{aux}, the number of AO basis functions by N_{basis}, the number of Laplace quadrature points by N_τ, and the numbers of occupied and virtual MOs by N_{occ} and N_{virt}, respectively. For two-, three-, and four-center integrals, the Mulliken notation is used. Furthermore, Einstein's sum convention[42] is employed. The spin index is dropped for convenience and matrix operations are to be taken before indexing in this work.

The total energy of the electronic ground state can be expressed within the adiabatic-connection formalism[39] as[20,38]

$$E = E_{\text{h}}[\{\phi_{\text{KS}}\}] + E_{\text{J}}[\{\phi_{\text{KS}}\}] + E_{\text{X}}[\{\phi_{\text{KS}}\}] + E_{\text{C}} \tag{1}$$

where E_{h}, E_{J}, and E_{X} denote the one-electron, Coulomb, and exact exchange energies, respectively. An expression for the correlation energy[24] E_{C} can be derived by applying the zero-temperature fluctuation-dissipation theorem and the RPA[34] as well as the RI approximation[27,40,41]

$$E_{\text{C}} = \frac{1}{2\pi}\int_0^{+\infty} \mathrm{d}\omega \mathrm{Tr}[\ln(\mathbf{1}-\mathbf{X}_0(\mathrm{i}\omega)\mathbf{V}) + \mathbf{X}_0(\mathrm{i}\omega)\mathbf{V}] \tag{2}$$

with the electron−electron interaction operator in the auxiliary basis

$$V_{MN} = (M|m_{12}|P)^{-1}(P|r_{12}^{-1}|Q)(Q|m_{12}|N)^{-1} \tag{3}$$

where m_{12} denotes the RI metric and r_{12} the interelectronic distance. $\mathbf{X}_0$ represents the noninteracting density−density response function in the auxiliary basis in the zero-temperature case.[43] For efficiency reasons, the response function is calculated in the imaginary time domain according to[29,34]

$$X_{0,MN}(\mathrm{i}\tau) = \mathrm{Tr}[\underline{\mathbf{G}}_0(-\mathrm{i}\tau)\mathbf{B}^M\overline{\mathbf{G}}_0(\mathrm{i}\tau)\mathbf{B}^N] \tag{4}$$

$$X_{0,MN}(\mathrm{i}\tau) = \underline{G}_{0,\mu\nu}(-\mathrm{i}\tau)B_{\nu\lambda}^M\overline{G}_{0,\lambda\sigma}(\mathrm{i}\tau)B_{\sigma\mu}^N \tag{5}$$

with the one-particle Green's function in the imaginary time domain

$$\mathbf{G}_0(\mathrm{i}\tau) = \Theta(-\mathrm{i}\tau)\underline{\mathbf{G}}_0(\mathrm{i}\tau) + \Theta(\mathrm{i}\tau)\overline{\mathbf{G}}_0(\mathrm{i}\tau) \tag{6}$$

$$\underline{\mathbf{G}}_{0,\mu\nu}(\mathrm{i}\tau) = C_{\mu i}C_{\nu i}\exp(-(\epsilon_i - \epsilon_F)\tau) \tag{7}$$

$$\overline{\mathbf{G}}_{0,\mu\nu}(\mathrm{i}\tau) = -C_{\mu a}C_{\nu a}\exp(-(\epsilon_a - \epsilon_F)\tau) \tag{8}$$

where $C_{\mu i}$ and $C_{\mu a}$ denote the occupied and unoccupied MO coefficients, respectively, and ϵ_{F} the Fermi level. The three-center integral matrix $\mathbf{B}^M$ is given by

$$B_{\nu\lambda}^M = (\nu\lambda|m_{12}|M) \tag{9}$$

A drawback of AO compared to MO formulations is the increased scaling with the size of the atom-centered basis. However, this drawback can be addressed by utilizing pivoted Cholesky decomposition of density-type matrices, thereby reintroducing the occupied index.[33] Furthermore, a memory-efficient expression for $\mathbf{X}_0(\mathrm{i}\tau)$ can be obtained using the idempotency relation of the ground-state density matrix $\mathbf{P}$

$$\mathbf{P} = \mathbf{PSP} \tag{10}$$

with the two-center overlap matrix $\mathbf{S}$, and the analogous expression for the one-particle Green's function in the negative imaginary time domain

$$\underline{\mathbf{G}}_0(-\mathrm{i}\tau) = \underline{\mathbf{G}}_0(-\mathrm{i}\tau)\mathbf{SP} \tag{11}$$

leading to[34]

$$X_{0,MN}(\mathrm{i}\tau) = \mathrm{Tr}[\mathbf{L}^{\mathrm{T}}\mathbf{S}\underline{\mathbf{G}}_0(-\mathrm{i}\tau)\mathbf{SLL}^{\mathrm{T}}\mathbf{B}^M\overline{\mathbf{G}}_0(\mathrm{i}\tau)\mathbf{B}^N\mathbf{L}] \tag{12}$$

$$X_{0,MN}\ (\mathrm{i}\tau) = \underline{G}_{0,\underline{j}\,\underline{i}}(-\mathrm{i}\tau)B^{M}_{\underline{i}\nu}\overline{G}_{0,\nu\mu}(\mathrm{i}\tau)B^{N}_{\mu\underline{j}} \tag{13}$$

where the pivoted Cholesky factorization of a matrix $\mathbf{A}$ is abbreviated by $\mathbf{A} = \mathbf{LL}^{\mathrm{T}}$. Each $\mathbf{B}^M$ is precontracted with the Cholesky factor $\mathbf{L}$ of the occupied one-particle density $\mathbf{P}$, which is independent of the Laplace points. This reduces the memory requirement for storing the three-center integrals from $(N_{\mathrm{aux}}N^2_{\mathrm{basis}})$ to $(N_{\mathrm{aux}}N_{\mathrm{basis}}N_{\mathrm{occ}})$. The final expression for $\mathbf{X}_0(\mathrm{i}\tau)$ reads

$$X_{0,MN}(\mathrm{i}\tau) = B^{M}_{\underline{j}\mu}(\mathrm{i}\tau)B^{N}_{\mu\underline{j}} \tag{14}$$

with

$$B^{M}_{\underline{j}\mu}(\mathrm{i}\tau) = \underline{G}_{0,\underline{j}\,\underline{i}}(-\mathrm{i}\tau)B^{M}_{\underline{i}\nu}\overline{G}_{0,\nu\mu}(\mathrm{i}\tau) \tag{15}$$

and the transformed three-center integrals[32−34]

$$B^{N}_{\mu\underline{j}} = B^{N}_{\mu\nu}L_{\nu\underline{j}} \tag{16}$$

From here on, we will refer to the transformed three-center integrals $B^{N}_{\mu\underline{i}}$ (eq 16) also as the three-center integrals.

After obtaining the response function in the imaginary time domain, it is transformed into the imaginary frequency domain with a contracted double-Laplace[32,34] or cosine transform[44] according to

$$\mathbf{X}_0(\mathrm{i}\omega) = \int_{-\infty}^{+\infty} \mathrm{d}\tau\ \cos(\omega\tau)\mathbf{X}_0(\mathrm{i}\tau) \tag{17}$$

The ω-CDGD-RI-RPA method scales formally as $O(N^2_{\mathrm{aux}}N_{\mathrm{basis}}N_{\mathrm{occ}} \propto M^4)$; it can, however, be implemented in an asymptotically linear scaling fashion.[34]

3. MINIMAL-OVERHEAD BATCHING

Within the calculation of the RPA correlation energy, the most demanding step in terms of memory requirements is the calculation of the response function in the imaginary time domain. The response function $\mathbf{X}_0(\mathrm{i}\tau)$ is calculated within the standard algorithm according to eq 14 for one Laplace point at a time. Therefore, the Laplace point-dependent three-center integrals $B^{M}_{\underline{i}\mu}(\mathrm{i}\tau)$ as well as the three-center integrals $B^{N}_{\mu\underline{i}}$ have to be stored in memory, which requires $(2N_{\mathrm{aux}}N_{\mathrm{basis}}N_{\mathrm{occ}})$ memory. Further, taking into account the memory requirements of the response function with dimensions $(N_{\mathrm{aux}} \times N_{\mathrm{aux}} \times N_{\tau})$, it becomes apparent that for large systems the memory requirements easily exceed the available system memory on a workstation or server. Thus, to overcome the limiting storage requirements within the calculation of the response function, a batching algorithm is necessary.

In this section, we first derive a simple batching method where only batching over the auxiliary function index is employed. Subsequently, we increase the complexity of our batching method by additionally batching over the AO basis-function index as well as the Laplace quadrature points. For the latter method, we derive an expression for the optimal number of batches using a Lagrange formalism. Finally, we compare both algorithms in terms of their scaling behavior and present computational results supporting our theoretical studies. Please note that we use the def2-SVP basis set for the calculations in this and the subsequent section (Sections 3 and 4) for illustration purposes only. Since our objective is to, first, demonstrate the scaling with the system size, the completeness of the basis set is not relevant in this context. However, for practical applications, where the objective is to obtain high-quality results, larger basis sets are typically required, which are presented in Section 6.

3.1. Trivial Batching.

In the following, we introduce the approach of batching over the auxiliary function index, which we will refer to as trivial batching. The pseudocode for this implementation is shown in Algorithm 1.

Algorithm 1 Trivial Batching

```
1:  for aux-batch1 do
2:    for $M \in$ aux-batch1 do
3:      read/recalculate $B^{M}_{\underline{j}\nu}$   $\forall \underline{j}, \nu$
4:      for all $\tau$ do
5:        $B^{M}_{\underline{i}\mu}(\mathrm{i}\tau) = \underline{G}_{0,\underline{i}\underline{j}}(-\mathrm{i}\tau)B^{M}_{\underline{j}\nu}\overline{G}_{0,\nu\mu}(\mathrm{i}\tau)$   $\forall \underline{i}, \mu$
6:      end for
7:    end for
8:    for aux-batch2 $\geq$ aux-batch1 do
9:      for $N \in$ aux-batch2 do
10:       read/recalculate $B^{N}_{\underline{i}\mu}$   $\forall \underline{i}, \mu$
11:     end for
12:     for all $\tau$ do
13:       for $M \in$ aux-batch1 do
14:         for $N \in$ aux-batch2 do
15:           $X_{0,MN}(\mathrm{i}\tau) = B^{M}_{\underline{i}\mu}(\mathrm{i}\tau)B^{N}_{\underline{i}\mu}$
16:         end for
17:       end for
18:     end for
19:   end for
20:   for $M \in$ aux-batch1 do
21:     for all $\tau$ do
22:       write on disk $X_{0,MN}(\mathrm{i}\tau)$   $\forall N$
23:     end for
24:   end for
25: end for
```

In the context of index batching, reading from disk or recomputing from scratch are analogous. That is, both variants yield a given set of tensor elements at a cost that is proportional to the amount of requested elements. Thus, the two possible variants for accessing the three-center integrals, namely, reading or recalculating (lines 3 and 10) both require the same batching and can therefore be discussed separately in Section 4.

In Algorithm 1, first, the tensor elements of the three-center integrals $B^{M}_{\underline{j}\nu}$ are accessed for one auxiliary function within the respective auxiliary batch (aux-batch) (line 3) and subsequently used to compute $B^{M}_{\underline{i}\mu}(\mathrm{i}\tau)$ (line 5). Next, within the second aux-batch loop (line 8), the tensor elements of the three-center integrals $B^{N}_{\underline{i}\mu}$ are accessed for a second time (line 10) and subsequently contracted with $B^{M}_{\underline{i}\mu}(\mathrm{i}\tau)$ for each Laplace point τ to form $X_{0,MN}(\mathrm{i}\tau)$ (line 15). Please note that, due to the symmetry of the response function, only $\frac{b'_{\mathrm{aux}}+1}{2}$ aux-batches are considered for the second aux-batch loop (line 8), where b'_{aux} denotes the number of aux-batches. Further, for performance reasons, the operations in line 5 as well as lines 13−17 are implemented as matrix multiplications to utilize the high performance of dense matrix algebra routines provided by basic linear algebra subroutine (BLAS) libraries. To further reduce the memory requirements of the algorithm, the response function is written on disk (line 22) by batching over the first auxiliary function index. However, the storage of the response function only becomes problematic for extremely large systems, since the memory demand scales as $\mathcal{O}(M^2)$ compared to the $\mathcal{O}(M^3)$ scaling of the three-center integral tensor.

3.2. Optimized Batching.

In this section, we extend the trivial batching algorithm: First, in addition to batching with respect to auxiliary functions, we also incorporate batching with respect to basis functions as well as Laplace points. Second, we use the method of Lagrange multipliers to minimize the number of three-center integral tensor accesses for a given amount of available memory. This allows the optimal utilization of the available memory with minimal overhead. Thus, we refer to this batching algorithm as optimized batching. The pseudocode for the optimized batching algorithm is shown in Algorithm 2.

Algorithm 2 Optimized Batching

```
 1: for aux-batch1 do
 2:    for AO-batch do
 3:       for $\tau$-batch do
 4:          for $M \in$ aux-batch1 do
 5:             read/recalculate $B_{\underline{j}\nu}^{M} \quad \forall \underline{j}, \nu$
 6:             for $\tau \in \tau$-batch do
 7:                $B_{\underline{i}\mu'}^{M}(\mathrm{i}\tau) = \underline{G}_{0,\underline{ij}}(-\mathrm{i}\tau) B_{\underline{j}\nu}^{M} \overline{G}_{0,\nu\mu'}(\mathrm{i}\tau) \quad \forall \underline{i}, \mu' \in$ AO-batch
 8:             end for
 9:          end for
10:          for aux-batch2 $\geq$ aux-batch1 do
11:             for $N \in$ aux-batch2 do
12:                read/recalculate $B_{\underline{i}\mu'}^{N} \quad \forall \underline{i}, \mu' \in$ AO-batch
13:             end for
14:             for $\tau \in \tau$-batch do
15:                for $M \in$ aux-batch1 do
16:                   for $N \in$ aux-batch2 do
17:                      $X_{0,MN}(\mathrm{i}\tau)$ += $B_{\underline{i}\mu'}^{M}(\mathrm{i}\tau) B_{\underline{i}\mu'}^{N}$
18:                   end for
19:                end for
20:             end for
21:          end for
22:       end for
23:    end for
24:    for $M \in$ aux-batch1 do
25:       for all $\tau$ do
26:          write on disk $X_{0,MN}(\mathrm{i}\tau) \quad \forall N$
27:       end for
28:    end for
29: end for
```

In this work, the following abbreviations are introduced: The number of aux-batches is denoted by b_{aux}, the number of AO-batches by b_{AO}, and the number of τ-batches by b_τ. Please note that b_{aux} denotes the number of aux-batches within the optimized batching, while b'_{aux} represents the number of aux-batches within the trivial batching algorithm. Further, the following approximations are used for simplicity: The number of auxiliary functions in an aux-batch is given by $\frac{N_{\text{aux}}}{b_{\text{aux}}}$, the number of basis functions in an AO-batch by $\frac{N_{\text{basis}}}{b_{\text{AO}}}$, and the number of Laplace points in a τ-batch is given by $\frac{N_\tau}{b_\tau}$. Please note that within this approximation, the number of functions in the respective batches constitutes a rational number; therefore, it needs to be rounded down to an integer for practical applications. For large systems, however, this rounding makes little difference.

In Algorithm 2, the most prominent changes compared to the trivial batching in Algorithm 1 include the loop over the basis-function batches ranging from lines 2 to 23. Accordingly, $B_{\underline{i}\mu'}^{M}(\mathrm{i}\tau)$ (line 7) and $B_{\underline{i}\mu'}^{N}$ (line 12) show decreased memory requirements, considering the batched aux- and basis-function index. Further, $B_{\underline{i}\mu'}^{M}(\mathrm{i}\tau)$ (line 7) as well as $X_{0,MN}(\mathrm{i}\tau)$ (line 17) are evaluated for one τ-batch.

In Table 1, the memory requirements for an implementation without any batching, the trivial batching, as well as the optimized batching scheme are compared. It follows that the memory requirements of the largest quantities within the response function calculation can be significantly reduced by employing either of the batching schemes. However, the optimized batching scheme provides a larger range of batching configurations for the same ratio, while for the trivial batching there is only one possibility to achieve a specific ratio.

As seen in Algorithm 1 (lines 3 and 10) and Algorithm 2 (lines 5 and 12), each element $B_{\underline{i}\mu}^{M}$ needs to be read/recalculated redundantly. Therefore, the reduced memory requirements come at the cost of a batching overhead, which is proportional to the number of batches. Consequently, a minimal amount of batches is required to minimize the batching overhead for a fixed amount of the available system memory. This can be achieved by employing the method of Lagrange multipliers. Please note that for the rest of this section we will refer to the amount of redundant integral reads or recalculations more generally as redundant integral tensor accesses.

Therefore, the rest of this section is structured as follows: At first, an expression for the number of redundant integral tensor element accesses is derived, followed by an expression for the constraint function. Subsequently, the number of redundant tensor accesses is minimized with respect to the number of

Table 1. Largest Quantities within the Response Function Calculation with Their Respective Memory Requirements for an Implementation without Any Batching, the Trivial Batching Algorithm (Algorithm 1), and the Optimized Batching Algorithm (Algorithm 2)[a]

quantity	memory			ratio	
	not batched[b]	trivial[c]	optimized[d]	trivial	optimized
$B_{\underline{i}\mu}^{M}(\mathrm{i}\tau)$	$N_{\text{occ}}N_{\text{aux}}N_{\text{basis}}$	$\frac{N_{\text{aux}}}{b'_{\text{aux}}}N_{\text{basis}}N_{\text{occ}}N_\tau$	$\frac{N_{\text{aux}}}{b_{\text{aux}}}\frac{N_{\text{basis}}}{b_{\text{AO}}}\frac{N_\tau}{b_\tau}N_{\text{occ}}$	$\frac{N_\tau}{b'_{\text{aux}}}$	$\frac{N_\tau}{b_{\text{aux}}b_{\text{AO}}b_\tau}$
$B_{\underline{i}\mu}^{N}$	$N_{\text{aux}}N_{\text{basis}}N_{\text{occ}}$	$\frac{N_{\text{aux}}}{b'_{\text{aux}}}N_{\text{basis}}N_{\text{occ}}$	$\frac{N_{\text{aux}}}{b_{\text{aux}}}\frac{N_{\text{basis}}}{b_{\text{AO}}}N_{\text{occ}}$	$\frac{1}{b'_{\text{aux}}}$	$\frac{1}{b_{\text{aux}}b_{\text{AO}}}$
$X_{0,M\,N}(\mathrm{i}\tau)$	$N_{\text{aux}}N_{\text{aux}}N_\tau$	$\frac{N_{\text{aux}}}{b'_{\text{aux}}}N_{\text{aux}}N_\tau$	$\frac{N_{\text{aux}}}{b_{\text{aux}}}N_{\text{aux}}N_\tau$	$\frac{1}{b'_{\text{aux}}}$	$\frac{1}{b_{\text{aux}}}$

[a] The ratio of the memory for the trivial batching and optimized batching algorithm to an algorithm without any batching is given for each quantity. For illustrative purposes, all tensors are represented by their tensor elements. [b] Evaluated according to eq 14 per Laplace point. [c] Evaluated according to Algorithm 1. [d] Evaluated according to Algorithm 2.

aux-, AO-, and τ-batches using the method of Lagrange multipliers to comply with the constraint.

3.2.1. Number of Integral Tensor Accesses. According to Algorithm 2, the number of integral tensor accesses $N'_{\mathrm{acc}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)$ for the elements of $B^M_{\underline{i}\mu}$ (line 5) is given by

$$N'_{\mathrm{acc}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = b_{\mathrm{AO}} b_\tau N_{\mathrm{aux}} N_{\mathrm{basis}} N_{\mathrm{occ}} \tag{18}$$

and the number of integral tensor accesses N''_{acc} for $B^N_{\underline{i}\mu'}$ (line 12) is given by

$$N''_{\mathrm{acc}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = \frac{b_{\mathrm{aux}} + 1}{2} b_\tau N_{\mathrm{aux}} N_{\mathrm{basis}} N_{\mathrm{occ}} \tag{19}$$

where the term $\frac{b_{\mathrm{aux}}+1}{2}$ stems from exploiting the symmetry of the response function in line 10. The total number of integral tensor accesses $N^{\mathrm{total}}_{\mathrm{acc}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)$ can be obtained by adding eqs 18 and 19, which leads to

$$N^{\mathrm{total}}_{\mathrm{acc}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = \bar{N} b_\tau \left(\frac{b_{\mathrm{aux}} + 1}{2} + b_{\mathrm{AO}} \right) \tag{20}$$

with

$$\bar{N} = N_{\mathrm{aux}} N_{\mathrm{basis}} N_{\mathrm{occ}} \tag{21}$$

3.2.2. Constraint Function. The constraint function $C(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)$ can be expressed as

$$C(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = \mathrm{mem}_{\mathrm{avail}} - \mathrm{mem}_{\mathrm{req}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = 0 \tag{22}$$

Table 2. Quantities That have to be Stored in Memory during the Calculation of the Response Function with Their Respective Memory Requirements[a]

quantity	memory
$B^M_{\underline{i}\mu'}(\mathrm{i}\tau)$	$\frac{N_{\mathrm{aux}}}{b_{\mathrm{aux}}} \frac{N_{\mathrm{basis}}}{b_{\mathrm{AO}}} \frac{N_\tau}{b_\tau} N_{\mathrm{occ}}$
$B^N_{\underline{i}\mu'}$	$\frac{N_{\mathrm{aux}}}{b_{\mathrm{aux}}} \frac{N_{\mathrm{basis}}}{b_{\mathrm{AO}}} N_{\mathrm{occ}}$
$\underline{G}_{0,\mu\nu}(-\mathrm{i}\tau)$	$N^2_{\mathrm{basis}}\ N_\tau$
$\overline{G}_{0,\mu\nu}(\mathrm{i}\tau)$	$N^2_{\mathrm{basis}}\ N_\tau$
V_{MN}	N^2_{aux}

[a] For illustrative purposes, all tensors are represented by their tensor elements. Note that $B^M_{\underline{i}\mu'}(\mathrm{i}\tau)$ and $B^N_{\underline{i}\mu'}$ are evaluated within Algorithm 2, lines 7 and 12, respectively.

where $\mathrm{mem}_{\mathrm{avail}}$ denotes the available system memory and $\mathrm{mem}_{\mathrm{req}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)$ the memory required for the algorithm. For the latter, the relevant quantities that have to be stored in memory during the algorithm are shown along with their memory requirements in Table 2. Please note that the memory requirements of the batched response function are not considered in Table 2 since its size is not significant as explained in Section 3.1. Using Table 2, $\mathrm{mem}_{\mathrm{req}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)$ can be written as

$$\begin{aligned}\mathrm{mem}_{\mathrm{req}}&(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) \\ &= \frac{N_{\mathrm{aux}}}{b_{\mathrm{aux}}} \frac{N_{\mathrm{basis}}}{b_{\mathrm{AO}}} N_{\mathrm{occ}} \left(\frac{N_\tau}{b_\tau} + 1 \right) + N^2_{\mathrm{aux}} + 2N^2_{\mathrm{basis}} N_\tau\end{aligned} \tag{23}$$

$$\mathrm{mem}_{\mathrm{req}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = \frac{\bar{N}}{b_{\mathrm{aux}} b_{\mathrm{AO}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) + \overline{\overline{N}} \tag{24}$$

with

$$\overline{\overline{N}} = N^2_{\mathrm{aux}} + 2N^2_{\mathrm{basis}} N_\tau \tag{25}$$

By inserting eq 24 into eq 22, the constraint function can be expressed as

$$C(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) = \overline{\mathrm{mem}}_{\mathrm{avail}} - \frac{\bar{N}}{b_{\mathrm{aux}} b_{\mathrm{AO}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) = 0 \tag{26}$$

where

$$\overline{\mathrm{mem}}_{\mathrm{avail}} = \mathrm{mem}_{\mathrm{avail}} - \overline{\overline{N}} \tag{27}$$

3.2.3. Minimizing the Number of Integral Tensor Accesses using the Method of Lagrange Multipliers. To obtain the optimal number of batches, the total number of integral tensor accesses $N^{\mathrm{total}}_{\mathrm{acc}}$ (eq 20) has to be minimized with respect to the number of aux-, AO-, and τ-batches, while not exceeding the available memory. The Lagrange function hence reads

$$\begin{aligned}\mathcal{L}&(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau, \lambda) \\ &= N^{\mathrm{total}}_{\mathrm{acc}}(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau) - \lambda C(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)\end{aligned} \tag{28}$$

where λ denotes the Lagrange multiplier. Inserting the expression for $N^{\mathrm{total}}_{\mathrm{acc}}$ according to eq 20 and $C(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau)$ according to eq 26 yields

$$\begin{aligned}\mathcal{L}&(b_{\mathrm{aux}}, b_{\mathrm{AO}}, b_\tau, \lambda) \\ &= \bar{N} b_\tau \left(\frac{b_{\mathrm{aux}} + 1}{2} + b_{\mathrm{AO}} \right) - \lambda \left(\overline{\mathrm{mem}}_{\mathrm{avail}} - \frac{\bar{N}}{b_{\mathrm{aux}} b_{\mathrm{AO}}} \right. \\ &\quad \left. \left(\frac{N_\tau}{b_\tau} + 1 \right) \right)\end{aligned} \tag{29}$$

Partial differentiation of eq 29 with respect to b_{aux}, b_{AO}, and λ gives the first-order conditions for the minimization

$$\frac{\partial \mathcal{L}}{\partial b_{\mathrm{aux}}} = \frac{\bar{N}}{2} b_\tau - \lambda \left(\frac{\bar{N}}{b^2_{\mathrm{aux}} b_{\mathrm{AO}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) \right) \overset{!}{=} 0 \tag{30}$$

$$\frac{\partial \mathcal{L}}{\partial b_{\mathrm{AO}}} = \bar{N} b_\tau - \lambda \frac{\bar{N}}{b_{\mathrm{aux}} b^2_{\mathrm{AO}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) \overset{!}{=} 0 \tag{31}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -\overline{\mathrm{mem}}_{\mathrm{avail}} + \frac{\bar{N}}{b_{\mathrm{aux}} b_{\mathrm{AO}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) + \frac{N^2_{\mathrm{aux}}}{b_{\mathrm{aux}}} \frac{N_\tau}{b_\tau} \overset{!}{=} 0 \tag{32}$$

To obtain a relation between b_{aux} and b_{AO}, eq 31 can be rewritten as

$$\bar{N} b_\tau = \lambda \frac{\bar{N}}{b_{\mathrm{aux}} b^2_{\mathrm{AO}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) \tag{33}$$

and inserted into eq 30, which leads to the following relation

$$b_{\mathrm{AO}} = \frac{1}{2} b_{\mathrm{aux}} \tag{34}$$

An expression for b_{aux} can be obtained by inserting eq 34 into eq 32 according to

$$\overline{\mathrm{mem}}_{\mathrm{avail}} - \frac{2\bar{N}}{b^2_{\mathrm{aux}}} \left(\frac{N_\tau}{b_\tau} + 1 \right) = 0 \tag{35}$$

leading to

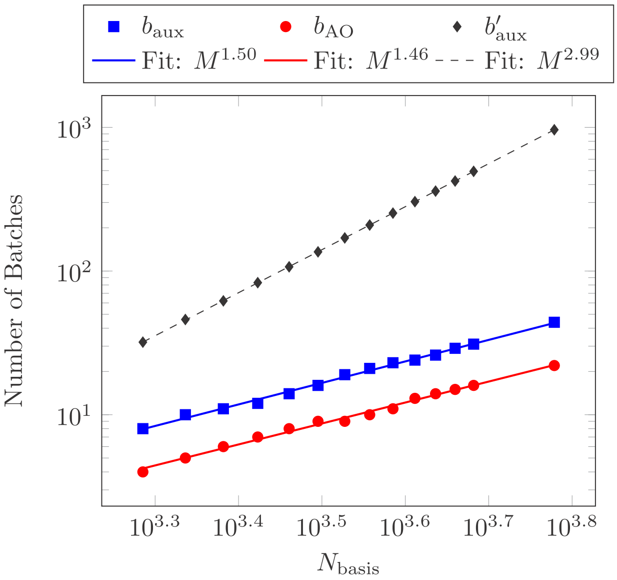


Figure 1. Log−log plot of the number of aux-batches b_{aux} and AO-batches b_{AO} for the optimized batching and number of aux-batches b'_{aux} for the trivial batching against the number of basis functions. In addition, the scaling fits are given. Note that for illustrative purposes, $\mathrm{mem}_{\mathrm{avail}}$ = 10 GB as well as $\overline{\overline{N}}$ = 0 (eq 25) was used.

$$b_{\mathrm{aux}} = \pm\sqrt{2}\sqrt{\bar{N}\left(\frac{N_\tau}{b_\tau} + 1\right)\frac{1}{\overline{\mathrm{mem}}_{\mathrm{avail}}}} \tag{36}$$

However, eq 36 is still dependent on b_τ. To derive an expression for b_τ, eq 34 can be inserted into eq 29

$$\mathcal{L}(b_{\mathrm{aux}}, b_\tau, \lambda) = \bar{N}b_\tau\left(b_{\mathrm{aux}} + \frac{1}{2}\right) - \lambda\left(\overline{\mathrm{mem}}_{\mathrm{avail}} - \frac{2\bar{N}}{b_{\mathrm{aux}}^2}\left(\frac{N_\tau}{b_\tau} + 1\right)\right) \tag{37}$$

and using eq 36 leads to

$$\mathcal{L}(b_\tau) = \bar{N}\sqrt{2}\sqrt{\frac{\bar{N}}{\overline{\mathrm{mem}}_{\mathrm{avail}}}b_\tau(N_\tau + b_\tau)} + \frac{1}{2}\bar{N}b_\tau \tag{38}$$

Equation 38 is minimized if b_τ is minimal. Therefore, it follows that

$$b_\tau = 1 \tag{39}$$

Inserting eq 39 into eq 36, the final expression for b_{aux} reads

$$b_{\mathrm{aux}} = \sqrt{2}\sqrt{\frac{\bar{N}}{\overline{\mathrm{mem}}_{\mathrm{avail}}}(N_\tau + 1)} \tag{40}$$

$$b_{\mathrm{aux}} = \sqrt{2}\sqrt{\frac{N_{\mathrm{aux}}N_{\mathrm{basis}}N_{\mathrm{occ}}}{\overline{\mathrm{mem}}_{\mathrm{avail}}}(N_\tau + 1)} \quad \propto \mathcal{O}(M^{3/2}) \tag{41}$$

and b_{AO} can be written using the relation in eq 34 as

$$b_{\mathrm{AO}} = \frac{1}{\sqrt{2}}\sqrt{\frac{\bar{N}}{\overline{\mathrm{mem}}_{\mathrm{avail}}}(N_\tau + 1)} \tag{42}$$

$$b_{\mathrm{AO}} = \frac{1}{\sqrt{2}}\sqrt{\frac{N_{\mathrm{aux}}N_{\mathrm{basis}}N_{\mathrm{occ}}}{\overline{\mathrm{mem}}_{\mathrm{avail}}}(N_\tau + 1)} \quad \propto \mathcal{O}(M^{3/2}) \tag{43}$$

It follows from eqs 41 and 43 that the number of batches scales as $\mathcal{O}(M^{3/2})$ or, equivalently, $\mathcal{O}(M^{1.5})$ with the system size and $\mathcal{O}(\mathrm{mem}_{\mathrm{avail}}^{-0.5})$ with respect to the available system memory, since the number of Laplace points N_τ is independent of the system size.

To summarize the results of the optimization, the optimal setting employs one τ-batch containing all Laplace points (eq 39), there are twice as many aux-batches as AO-batches (eq 34), and the number of batches scales as $\mathcal{O}(\mathrm{mem}_{\mathrm{avail}}^{-0.5})$ with respect to the available system memory and $\mathcal{O}(M^{1.5})$ with respect to the system size (eqs 41 and 43).

3.3. Comparing the Optimized Batching and the Trivial Batching.

In the following, the scaling behavior for the number of batches as well as the number of integral tensor accesses is analyzed for the trivial and the optimized batching.

3.3.1. Number of Batches. In the context of optimizing the batch sizes, the previously introduced trivial batching scheme can be regarded as a nonoptimal variant, where b_{AO} and b_τ were set equal to 1. Thus, for the trivial batching, the number of aux-batches b'_{aux} can be obtained using the constraint function in eq 26. Setting b_{AO} and b_τ equal to 1 leads to

$$C(b'_{\mathrm{aux}}) = \overline{\mathrm{mem}}_{\mathrm{avail}} - \frac{\bar{N}}{b'_{\mathrm{aux}}}(N_\tau + 1) = 0 \tag{44}$$

Rewriting eq 44 gives the optimal number of aux-batches b'_{aux}

$$b'_{\mathrm{aux}} = \frac{\bar{N}}{\overline{\mathrm{mem}}_{\mathrm{avail}}}(N_\tau + 1) \tag{45}$$

For the trivial batching, the optimal number of batches grows as $\mathcal{O}(M^{3.0})$, while the optimized batching shows a more favorable scaling of $\mathcal{O}(M^{1.5})$ (eqs 41 and 43).

To verify these theoretical considerations, we first carried out calculations on simple linear n-alkanes of increasing size using the def2-SVP basis set.[45−47] In Figure 1, a log−log plot of the numbers of aux- and AO-batches against the number of

AO basis functions is shown. The obtained scaling behavior is in very good agreement with the theoretical scaling. The fluctuations around the optimized batching arise solely from the rounding of the batch dimension to the nearest lower integer.

3.3.2. Number of Integral Tensor Accesses. The scaling behavior for both batching algorithms can be obtained using the expression for the number of integral tensor accesses in eq 20 and inserting the expression for the optimized number of batches.

For the optimized batching, we insert the expressions for b_{aux} (eq 41), b_{AO} (eq 43), and b_τ (eq 39) leading to

$$N_{\text{acc}}^{\text{opt}}(b_{\text{aux}}, b_{\text{AO}}, b_\tau) = \bar{N}\sqrt{2}\sqrt{\frac{\bar{N}}{\text{mem}_{\text{avail}}}(N_\tau + 1)} + \frac{\bar{N}}{2} \tag{46}$$

It follows that the number of integral tensor accesses grows as $O(M^{4.5})$.

Analogously, for the trivial batching, the number of integral tensor accesses N'_{acc} can be obtained by inserting b'_{aux} (eq 45) into eq 20 as well as setting b_{AO} and b_τ equal to 1, resulting in

$$N_{\text{acc}}^{\text{triv}} = \frac{\bar{N}^2}{\text{mem}_{\text{avail}}}(N_\tau + 1) + \frac{3}{2}\bar{N} \tag{47}$$

For the trivial batching, the number of integral tensor accesses thus grows as $O(M^{6.0})$.

In Figure 2, the corresponding log–log plot of the number of integral tensor accesses against the number of AO basis functions is shown for the trivial and the optimized batching, which confirms the theoretical scaling of $O(M^{4.5})$ for the optimized batching (eq 46) and $O(M^{6.0})$ for the trivial batching (eq 47).

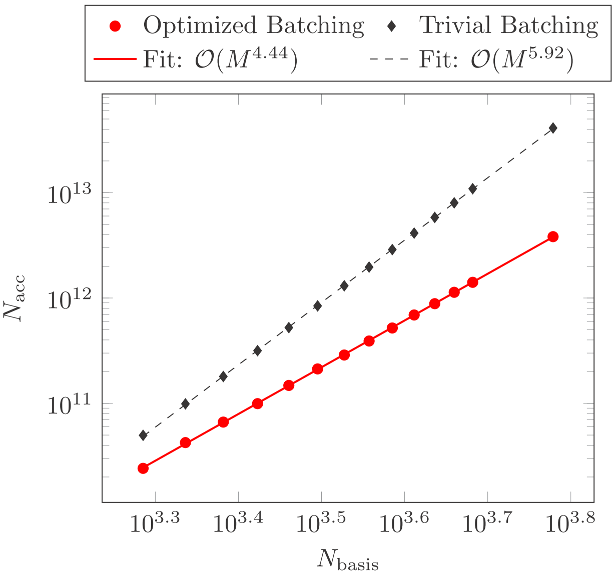


Figure 2. Log–log plot of the number of integral tensor accesses N_{acc} for the optimized batching and the trivial batching against the number of basis functions. In addition, the scaling fits are given. Note that for illustrative purposes, $\text{mem}_{\text{avail}}$ = 10 GB as well as $\bar{\bar{N}} = 0$ (eq 25) was uesd.

In conclusion, the optimized batching shows a more favorable scaling with respect to the number of batches as well as the number of integral tensor accesses compared to the trivial batching. Since the number of integral tensor accesses is proportional to the batching overhead, we can also conclude that our optimized batching is more effective in reducing the batching overhead than the trivial batching. Especially when aiming for very large systems requiring a high number of batches, the advantages of our optimized batching become apparent. In essence, the optimized batching represents the best compromise between program runtime and demand for system memory.

4. INTEGRAL-DIRECT RPA

As mentioned in Section 3.1, there are two approaches for accessing the elements of the three-center integral tensor $B_{i\mu}^{M}$, namely, reading and recomputing (Algorithm 1, lines 3 and 10 and Algorithm 2, lines 5 and 12). In this section, we will first compare both approaches and establish why integral-direct RPA (recomputation) is best suited for the computation of very large systems. We will then analyze the scaling behavior for integral-direct RPA and systematically improve upon it.

4.1. Hard-Disk IO vs On the Fly Computation of the three-center Integrals. For the first approach to access the elements of the three-center integrals, the integrals are stored on disk and the tensor elements $B_{i\mu}^{M}$ are read into memory in batches. Thus, the batching overhead is determined by the amount of input/output operations on a physical disk (disk I/O). However, since the three-center integrals with dimensions ($N_{\text{aux}} \times N_{\text{basis}} \times N_{\text{occ}}$) have to be stored on disk, the algorithm is limited by the available disk space. The storage limitation problem can be overcome entirely using an integral-direct scheme for the three-center integrals, which we will refer to as integral-direct RPA.

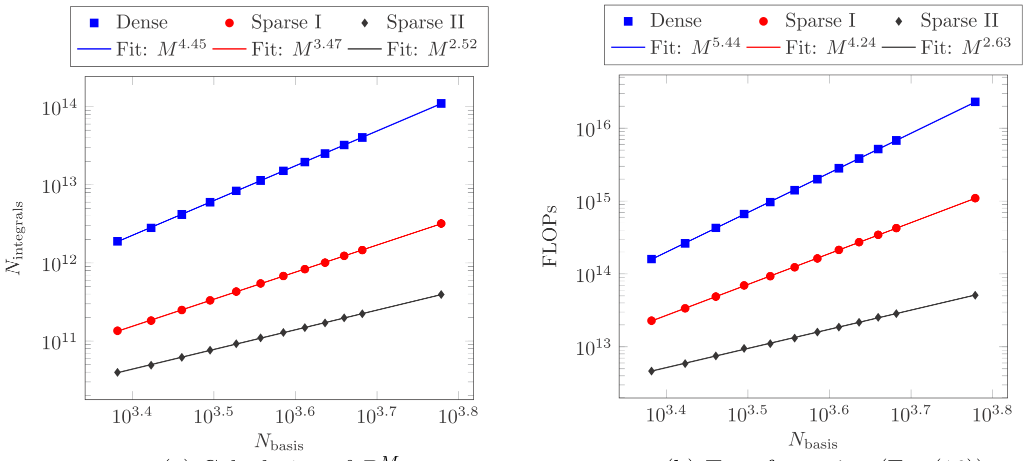
In Table 3, both approaches as well as an implementation without any batching are compared with regard to the feasible system size. To this end, the memory and disk space requirements for all methods are shown for different system sizes using the def2-SVP basis set.[45−47] Please note again that we use this basis set for illustration purposes only. Further, to demonstrate the scope of the methods from a practical viewpoint, it is noted whether the respective system is accessible on a computing node using 200 GB of memory space and 2500 GB of disk space. As expected, when reading the three-center integrals from disk, we are able to access much larger system sizes than without utilizing any batching, since this approach is not limited by the available system memory. However, this shifts the bottleneck to the disk space requirements such that storing the three-center integrals becomes the limiting factor and, therefore, larger systems are not accessible. In contrast to that, integral-direct RPA opens the way to access all listed systems, since this method is not limited by the disk space requirements of the three-center integral tensor. Within integral-direct RPA, only the response function has to be stored on disk. However, the response function with dimensions ($N_{\text{aux}} \times N_{\text{aux}} \times N_\tau$) is orders of magnitude smaller than the three-center integral tensor with dimensions ($N_{\text{aux}} \times N_{\text{basis}} \times N_{\text{occ}}$). Thus, it does not constitute the limiting factor for the systems shown in Table 3.

4.2. Scaling. The calculation of the response function comprises four major steps: The computation of the three-center integrals in the AO basis $B_{\mu\nu}^{M}$ and its subsequent

Table 3. Required Memory and Disk Space for Various Systems Utilizing an Implementation without any Batching (NB) and the Reading Variant of the RPA Batching Routine (Read) as well as the Integral-Direct RPA (Int-Dir)[a]

	memory (GB)		disk space (GB)		accessibility[f]		
system	NB[b,c]	Read/Int-Dir[c]	Read[d,e]	Int-Dir[d]	NB	Read	Int-Dir
$C_{100}H_{202}$	96.3	1.6	50.3	6.1	√	√	√
$C_{110}H_{222}$	127.1	2.0	66.2	7.3	√	√	√
$C_{120}H_{242}$	163.8	2.4	85.1	8.7	√	√	√
$C_{130}H_{262}$	207.0	2.8	107.2	10.2	×	√	√
$C_{300}H_{602}$	1177.9	15.0	608.6	54.2	×	√	√
$C_{400}H_{802}$	2373.3	26.6	1221.5	96.3	×	√	√
$C_{500}H_{1002}$	4145.3	41.6	2127.1	150.4	×	√	√
$C_{600}H_{1202}$	9123.7	59.8	4640.2	216.5	×	×	√
$C_{1000}H_{2002}$	62 189.4	166.1	31 312.2	601.1	×	×	√
$(DNA)_4$	183.4	2.5	95.5	9.9	√	√	√
$(DNA)_8$	1488.3	10.2	759.6	41.0	×	√	√
$(DNA)_{16}$	11 987.0	41.4	6056.1	166.5	×	×	√

[a] Note that the disk space requirements for an implementation without batching are not shown, since no quantities are stored on disk. Further, it is assessed whether the respective system is accessible, employing either method on a computing node using 200 GB of memory space and 2500 GB of disk space. [b] For storing $B_{i\mu}{}^{M}(i\tau)$ per Laplace point (eq 14). [c] For storing $\underline{G}_{0,\mu\nu}(-i\tau)$, $\overline{G}_{0,\mu\nu}(i\tau)$, and V_{MN} (Table 2). [d] For storing the response function. [e] For storing the three-center integrals. [f] Memory: 200 GB; disk space: 2500 GB.



(a) Calculation of $B^{M}_{\mu\nu}$ (b) Transformation (Eq. (16))

Figure 3. Log−log plots of the number of calculated primitive integrals $N_{\text{integrals}}$ against the number of basis functions (a) and the number of FLOPs needed for the transformation of the three-center integrals (b). In addition, the scaling fits are given. Note that for illustrative purposes, $\text{mem}_{\text{avail}}$ = 10 GB as well as $\overline{\overline{N}} = 0$ (eq 25) was used.

transformation (eq 16) as well as the matrix multiplications to obtain $B^{M}_{i\mu}(i\tau)$ (eq 15) and $X_{0,MN}(i\tau)$ (eq 14).

4.2.1. Without Batching. When no batching is employed, the calculation of the three-center integrals $B^{M}_{\mu\nu}$ formally scales as $O(N_{\text{aux}}N^{2}_{\text{basis}} \propto M^{3.0})$, while the transformation shows an $O(N_{\text{aux}}N^{2}_{\text{basis}}N_{\text{occ}} \propto M^{4.0})$ scaling. The matrix multiplications both scale as $O(M^{4.0})$. As mentioned in Section 2, the calculation of the response function can also be implemented in an asymptotically linear scaling fashion,[34] which, however, will not be discussed further in this work.

4.2.2. Dense. When there is not enough available system memory, batching has to be employed (integral-direct RPA). Please note that for our integral-direct RPA implementation, we use dense matrix algebra and hence presently do not aim for linear scaling. Our method could also be implemented using sparse matrix algebra; however, this would significantly complicate the determination of optimal batch sizes since the exact memory demand is not known a priori in case of sparse matrices.

Within integral-direct RPA, the response function is calculated in the batching routine (Algorithm 2), where the three-center integral tensor is recomputed on the fly. The recomputation (lines 5 and 12) is comprised of the computation of the three-center integrals in the AO basis $B^{M}_{\mu\nu}$ and the subsequent transformation with the Cholesky factor of the occupied one-particle density matrix (eq 16) for each batch only. It follows that the formal scaling for the computation of $B^{M}_{\mu\nu}$ as well as the transformation is increased by a factor of $O(M^{1.5})$, which accounts for the scaling with respect to the numbers of batches (eqs 41 and 43). The scaling for the matrix multiplications in lines 7 and 17, however, remains unchanged since it is independent of the batching as can be deduced from Algorithm 2. Thus, within integral-direct RPA, the integral calculation formally scales as $O(M^{4.5})$ and the transformation as $O(M^{5.5})$. To confirm these considerations, we carried out

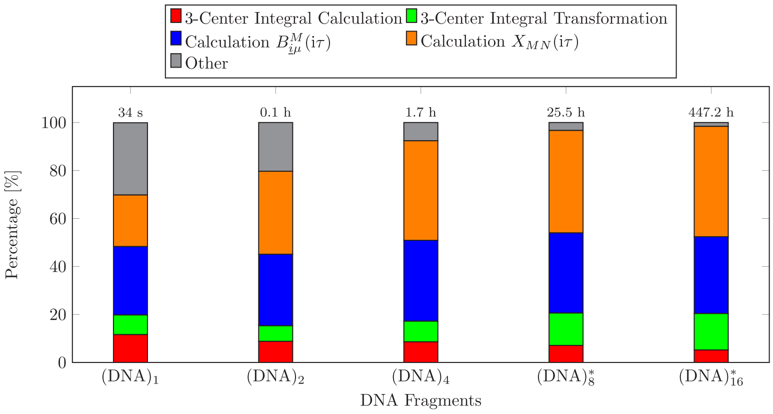


Figure 4. Contributions to the total time for the integral-direct calculation of the RPA correlation energy for DNA fragments using the def2-SVP basis set. Specifically, timings for the following operations are shown: three-center integral calculation (Algorithm 2, lines 5 and 12), three-center integral transformation (Algorithm 2, lines 5 and 12), calculation of $B^{M}_{i\mu}(\mathrm{i}\tau)$ (Algorithm 2, line 7), and calculation of $X_{MN}(\mathrm{i}\tau)$ (Algorithm 2, line 17). Systems that are marked with an asterisk (*) can only be computed using our integral-direct RPA method since the memory requirements for the unbatched variant would exceed the available system memory.

calculations on linear n-alkanes of increasing size using the def2-SVP basis set.[45−47] The results are summarized in Figure 3 in the blue graphs (denoted by "Dense"), where a log−log plot of the number of primitive integral calculations against the number of basis functions (a) as well as a log−log plot of the numbers of floating point operations (FLOPs) needed for the transformation of the three-center integrals against the number of basis functions (b) are shown.

4.2.3. Sparse I. Due to the exponential decay of the overlapping Gaussian-type basis functions of the charge distribution ($\mu\nu$), the number of significant three-center integrals $B^{M}_{\mu\nu}$ scales asymptotically only as $\mathcal{O}(M^{3.5})$ with a relatively early onset of the reduced scaling (only a few Ångström coupling distance between the AO basis functions μ and ν). Furthermore, the sparsity of ($\mu\nu$) can also be exploited in the transformation of the three-center integrals using block-sparse matrix multiplication. This reduces the scaling for the calculation and transformation by a factor of M, leading to an asymptotic scaling of $\mathcal{O}(M^{3.5})$ for the calculation and $\mathcal{O}(M^{4.5})$ for the transformation. To verify this, we edited the code of our implementation by incorporating shell pair screening for the integral calculation and using sparse matrix algebra for the transformation, where we only exploit the sparsity of the charge distribution ($\mu\nu$). The results are summarized in the red graphs (denoted by "Sparse I") in Figure 3. It can be observed that the obtained scaling for the calculation (Figure 3a) and the transformation (Figure 3b) are in good agreement with the theoretical scaling.

4.2.4. Sparse II. The scaling for the calculation can be further reduced by employing the Coulomb metric attenuated by the complementary error function (see eq 9 with $m_{12} = \frac{\mathrm{erfc}(\omega_{\mathrm{att}} r_{12})}{r_{12}}$), which decreases the range of coupling between the charge distribution ($\mu\nu$) and the auxiliary functions. This reduces the asymptotic scaling for the calculation of the three-center integrals $B^{M}_{\mu\nu}$ to $\mathcal{O}(M^{2.5})$. Furthermore, the scaling for the transformation of the three-center integrals can be reduced by additionally exploiting the sparsity of the Cholesky factor, leading to an asymptotic $\mathcal{O}(M^{2.5})$ scaling. For our implementation, we switched to the Coulomb metric attenuated by the complementary error function with the attenuation parameter $w_{\mathrm{att}} = 0.1$ a.u.[33] and used the approximate integral partition bounds (aIPBs)[48] developed by our group for screening the three-center integral computation. The results are shown in the black graphs (denoted by "Sparse II") in Figure 3.

To summarize, within integral-direct RPA, the computation of $B^{M}_{\mu\nu}$ scales formally as $\mathcal{O}(M^{4.5})$, the subsequent transformation as $\mathcal{O}(M^{5.5})$, albeit with a small prefactor depending on the available system memory, and the matrix multiplications for obtaining $B^{M}_{i\mu}(\mathrm{i}\tau)$ (line 7) and $X_{0,MN}(\mathrm{i}\tau)$ (line 17) scale formally as $\mathcal{O}(M^{4.0})$. However, the scaling for the calculation and transformation of the three-center integral tensor can be reduced to $\mathcal{O}(M^{2.5})$ by employing shell pair screening, a local metric for the three-center integral tensor, and integral screening, as well as using sparse matrix algebra for the transformation of the three-center integrals. As a result, these redundant on the fly recomputations of the three-center integrals do not represent a significant bottleneck compared to the computation of $B^{M}_{i\mu}(\mathrm{i}\tau)$ (line 7) and $X_{0,MN}(\mathrm{i}\tau)$ (line 17) in practice.

5. COMPUTATIONAL DETAILS

Our new integral-direct RPA method was implemented in the FermiONs++ program package.[49−51] The Kohn−Sham orbitals used for the RPA energy calculations were obtained by preceding DFT calculations employing the generalized gradient approximation of Perdew, Burke, and Ernzerhof (PBE).[52,53] The atomic basis sets def2-SVP and def2-QZVP are used.[45,46] The RI approximation, which is applied to the four-center integrals in the correlation part of the RPA energy, employs the corresponding auxiliary basis sets[47,54] with the attenuated Coulomb metric and the attenuation parameter $\omega_{\mathrm{att}} = 0.1$ a.u.[33] For the Laplace expansion, 13−15 quadrature points were used.[34] All calculations were carried out on an Intel Xeon E5-2667 processor using 16 threads.

6. PERFORMANCE

In the following, we investigate the performance of our integral-direct RPA method by considering the contribution of the batching overhead to the total computation time. In respect thereof, we calculated DNA fragments of increasing size. To obtain physically meaningful energies, we carried out calculations using the def2-QZVP basis set, which has shown to yield very accurate results within RPA.[13,27,32−34,55−58] Furthermore, for illustration purposes, we carried out calculations using the def2-SVP basis set to compute large systems in a reasonable amount of time.

In Figure 4, the contributions of the most time-consuming operations to the overall time are shown for DNA fragments using the def2-SVP basis set. The timings for the integral calculation and transformation as well as its contribution to the total time are considerably lower and grow less rapidly with increased system size as compared to the calculation of $B^M_{i\mu}(i\tau)$ (line 7) and $X_{MN}(i\tau)$ (line 17). It follows that the timings for the calculation and transformation observed for the largest systems in Figure 4 are a direct consequence of exploiting the sparsity of the system as explained in Section 4. Thus, the contribution of the batching overhead is relatively low and the total computation time is dominated by the calculation of $B^M_{i\mu}(i\tau)$ (Algorithm 2, line 7) and $X_{MN}(i\tau)$ (Algorithm 2 line 17).

In Figure 5, the corresponding results for DNA fragments using the def2-QZVP basis set are shown. Without batching (integral-direct RPA), we would only be able to compute $(\mathrm{DNA})_1$, since for larger systems the memory demand for the unbatched variant already exceeds the available system memory. In contrast, using the def2-SVP basis (Figure 4), systems up to $(\mathrm{DNA})_4$ were accessible without batching. Consequently, our integral-direct RPA method is of even higher relevance for large basis set calculations (which are required to obtain high-quality results in practice), where the memory limitation problem (which our proposed batching solves in an optimized fashion) already emerges for much smaller molecules. The timings shown in Figure 5 indicate that the calculation of the three-center integrals is considerably more demanding compared to the def2-SVP basis set results (Figure 4). For larger basis sets, the three-center integrals $B^M_{\mu\nu}$ show less sparsity and thus the computational cost increases since shell pair screening and integral screening methods cannot significantly decrease the number of significant elements for the present systems.

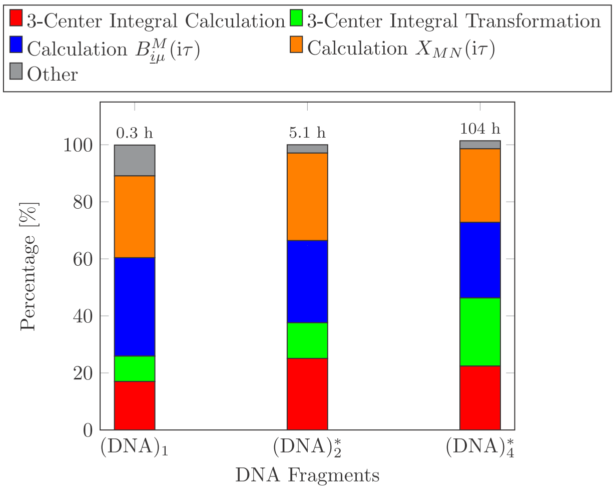


Figure 5. Contributions to the total time for the integral-direct calculation of the RPA correlation energy for DNA fragments using the def2-QZVP basis set. Specifically, timings for the following operations are shown: three-center integral calculation (Algorithm 2, lines 5 and 12), three-center integral transformation (Algorithm 2, lines 5 and 12), calculation of $B^M_{i\mu}(i\tau)$ (Algorithm 2, line 7), and calculation of $X_{MN}(i\tau)$ (Algorithm 2, line 17). Systems that are marked with an asterisk (*) can only be computed using our integral-direct RPA method, since the memory requirements for the unbatched variant would exceed the available system memory.

7. CONCLUSIONS AND OUTLOOK

We presented a memory-efficient integral-direct RPA algorithm based on our ω-CDGD-RI-RPA method by employing an optimized batching scheme, which, by construction via a Lagrangian formalism, allows for the most effective utilization of the available system memory, while minimizing the number of three-center integral tensor calculations.

We showed that our optimized batching scheme over the auxiliary and basis functions is able to minimize the batching overhead for a given amount of memory considerably better than an implementation where only batching with respect to auxiliary functions is employed by considering their scaling behavior with the system size M. For our optimized batching, the number of batches, which are proportional to the batching overhead, scale only as $\mathcal{O}(M^{1.5})$, which is a considerable improvement compared to the $\mathcal{O}(M^{3.0})$ scaling for a simple batching implementation over the auxiliary functions only. Furthermore, we have shown that the utilization of an integral-direct scheme for the three-center integral tensor, as opposed to reading the three-center integrals from disk, completely alleviates the storage bottleneck of the three-center integral tensor, thereby allowing the calculation of large systems, which were previously intractable. For the performance assessment of our integral-direct RPA method, we calculated DNA fragments of increasing size, showing that the batching overhead has a relatively small contribution on the total time. In this regard, we calculated the DNA fragment $(\mathrm{DNA})_{16}$ comprised of 1052 atoms and 11 230 basis functions.

In the future, our method could in principle be extended to asymptotically linear scaling schemes using sparse matrix algebra. However, for the computation of the optimized number of batches, the precise sparsity of the relevant matrices has to be known beforehand, which is only determined at program runtime, so that efficient estimates will be required.

Moreover, it has been shown that significant performance gains can be obtained by porting computer-intensive code to the graphics processing unit (GPU). However, special algorithms are necessary for the optimal exploitation of the scarce memory resources of GPUs as well as to reduce the high-cost data transfer between the GPU and the central processing unit (CPU). Our integral-direct RPA method is able to address both challenges: We compute the optimal amount of batches for a given amount of GPU memory. Further, all quantities needed for the computation of the response function could be computed directly on the GPU, thereby minimizing the data transfer between the GPU and CPU. Since the computation of the response function is the computationally most expensive part of the total calculation, significant performance gains are expected by porting our integral-direct RPA algorithm to the GPU.

Lastly, we would like to emphasize the applicability of the underlying concepts of our integral-direct RPA method (such as the derivation of the optimized batching method using the method of Lagrange multipliers) to related correlation methods such as SOS-MP2 and Coupled-Cluster variants.

AUTHOR INFORMATION

Corresponding Author

Christian Ochsenfeld − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), 81377 Munich, Germany; Max Planck Institute for Solid State Research, D-70569 Stuttgart, Germany;* orcid.org/0000-0002-4189-6558; Email: christian.ochsenfeld@cup.uni-muenchen.de

Authors

Viktoria Drontschenko − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), 81377 Munich, Germany*

Daniel Graf − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), 81377 Munich, Germany*

Henryk Laqua − *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), 81377 Munich, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.1c00494

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) within the cluster of excellence (EXC2111-390814868) "Munich Center for Quantum Science and Technology", MCQST. C.O. further acknowledges financial support as Max-Planck Fellow at the MPI-FKF Stuttgart.

REFERENCES

(1) Ciofini, I.; Daul, C. A. DFT calculations of molecular magnetic properties of coordination compounds. *Coord. Chem. Rev.* **2003**, *238–239*, 187.

(2) Li, X.; Cai, Z.; Sevilla, M. D. Investigation of proton transfer within DNA base pair anion and cation radicals by density functional theory (DFT). *J. Phys. Chem. B* **2001**, *105*, 10115.

(3) Jacquemin, D.; Wathelet, V.; Perpete, E. A.; Adamo, C. Extensive TD-DFT benchmark: singlet-excited states of organic molecules. *J. Chem. Theory Comput.* **2009**, *5*, 2420−2435.

(4) Armiento, R.; Mattsson, A. E. Functional designed to include surface effects in self-consistent density functional theory. *Phys. Rev. B* **2005**, *72*, No. 085108.

(5) Nolan, M.; Grigoleit, S.; Sayle, D. C.; Parker, S. C.; Watson, G. W. Density functional theory studies of the structure and electronic structure of pure and defective low index surfaces of ceria. *Surf. Sci.* **2005**, *576*, 217.

(6) Buchwald, J.; Hennes, M. Adsorption and diffusion of Au, Pt, and Co adatoms on SrTiO3 (001) surfaces: A density functional theory study. *Surf. Sci.* **2020**, No. 121683.

(7) Widdifield, C. M.; Farrell, J. D.; Cole, J. C.; Howard, J. A.; Hodgkinson, P. Resolving alternative organic crystal structures using density functional theory and NMR chemical shifts. *Chem. Sci.* **2020**, *11*, 2987.

(8) Sasmal, A.; Shit, S.; Rizzoli, C.; Wang, H.; Desplanches, C.; Mitra, S. Framework solids based on copper (II) halides (Cl/Br) and methylene-bridged bis (1-hydroxybenzotriazole): Synthesis, crystal structures, magneto-structural correlation, and density functional theory (DFT) studies. *Inorg. Chem.* **2012**, *51*, 10148.

(9) Campbell, J.; Mercier, H. P.; Franke, H.; Santry, D. P.; Dixon, D. A.; Schrobilgen, G. J. Syntheses, Crystal Structures, and Density Functional Theory Calculations of the c loso-[1-M (CO) 3 (η4-E9)] 4-(E= Sn, Pb; M= Mo, W) Cluster Anions and Solution NMR Spectroscopic Characterization of [1-M (CO) 3 (η4-Sn9)] 4-(M= Cr, Mo, W). *Inorg. Chem.* **2002**, *41*, 86.

(10) Schleder, G. R.; Padilha, A. C.; Acosta, C. M.; Costa, M.; Fazzio, A. From DFT to machine learning: recent approaches to materials science–a review. *J. Phys. Mater.* **2019**, *2*, No. 032001.

(11) Stöhr, M.; Van Voorhis, T.; Tkatchenko, A. Theory and practice of modeling van der Waals interactions in electronic-structure calculations. *Chem. Soc. Rev* **2019**, *48*, 4118.

(12) Kussmann, J.; Beer, M.; Ochsenfeld, C. Linear-scaling self-consistent field methods for large molecules. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2013**, *3*, 614.

(13) Ren, X.; Rinke, P.; Joas, C.; Scheffler, M. Random-phase approximation and its applications in computational chemistry and materials science. *J. Mater. Sci.* **2012**, *47*, 7447.

(14) Zhu, W.; Toulouse, J.; Savin, A.; Ángyán, J. G. Range-separated density-functional theory with random phase approximation applied to noncovalent intermolecular interactions. *J. Chem. Phys.* **2010**, *132*, No. 244108.

(15) Ehrlich, S.; Moellmann, J.; Reckien, W.; Bredow, T.; Grimme, S. System-Dependent Dispersion Coefficients for the DFT-D3 Treatment of Adsorption Processes on Ionic Surfaces. *ChemPhysChem.* **2011**, *12*, 3414.

(16) Caldeweyher, E.; Bannwarth, C.; Grimme, S. Extension of the D3 dispersion coefficient model. *J. Chem. Phys.* **2017**, *147*, No. 034112.

(17) Moellmann, J.; Grimme, S. DFT-D3 study of some molecular crystals. *J. Phys. Chem. C* **2014**, *118*, 7615.

(18) Sure, R.; Antony, J.; Grimme, S. Blind prediction of binding affinities for charged supramolecular host–guest systems: achievements and shortcomings of DFT-D3. *J. Phys. Chem. B* **2014**, *118*, 3431.

(19) Kaltak, M.; Klimeš, J.; Kresse, G. Cubic scaling algorithm for the random phase approximation: Self-interstitials and vacancies in Si. *Phys. Rev. B* **2014**, *90*, No. 054115.

(20) Langreth, D. C.; Perdew, J. P. Exchange-correlation energy of a metallic surface: Wave-vector analysis. *Phys. Rev. B* **1977**, *15*, No. 2884.

(21) Kurth, S.; Perdew, J. P. Density-functional correction of random-phase-approximation correlation with results for jellium surface energies. *Phys. Rev. B* **1999**, *59*, No. 10461.

(22) Furche, F. Molecular tests of the random phase approximation to the exchange-correlation energy functional. *Phys. Rev. B* **2001**, *64*, No. 195120.

(23) Fuchs, M.; Gonze, X. Accurate density functionals: Approaches using the adiabatic-connection fluctuation-dissipation theorem. *Phys. Rev. B* **2002**, *65*, No. 235109.

(24) Niquet, Y.; Fuchs, M.; Gonze, X. Exchange-correlation potentials in the adiabatic connection fluctuation-dissipation framework. *Phys. Rev. A* **2003**, *68*, No. 032507.

(25) Fuchs, M.; Niquet, Y.-M.; Gonze, X.; Burke, K. Describing static correlation in bond dissociation by Kohn−Sham density functional theory. *J. Chem. Phys.* **2005**, *122*, No. 094116.

(26) Furche, F.; Van Voorhis, T. Fluctuation-dissipation theorem density-functional theory. *J. Chem. Phys.* **2005**, *122*, No. 164106.

(27) Eshuis, H.; Furche, F. Basis set convergence of molecular correlation energy differences within the random phase approximation. *J. Chem. Phys.* **2012**, *136*, No. 084105.

(28) Heßelmann, A.; Görling, A. Random phase approximation correlation energies with exact Kohn−Sham exchange. *Mol. Phys.* **2010**, *108*, 359−372.

(29) Graf, D.; Beuerle, M.; Ochsenfeld, C. Low-Scaling Self-Consistent Minimization of a Density Matrix Based Random Phase Approximation Method in the Atomic Orbital Space. *J. Chem. Theory Comput.* **2019**, *15*, 4468.

(30) Beuerle, M.; Graf, D.; Schurkus, H. F.; Ochsenfeld, C. Efficient calculation of beyond RPA correlation energies in the dielectric matrix formalism. *J. Chem. Phys.* **2018**, *148*, No. 204104.

(31) Graf, D.; Ochsenfeld, C. A range-separated generalized Kohn−Sham method including a long-range nonlocal random phase approximation correlation potential. *J. Chem. Phys.* **2020**, *153*, No. 244118.

(32) Schurkus, H. F.; Ochsenfeld, C. Communication: An effective linear-scaling atomic-orbital reformulation of the random-phase approximation using a contracted double-Laplace transformation. *J. Chem. Phys.* **2016**, *144*, No. 031101.

(33) Luenser, A.; Schurkus, H. F.; Ochsenfeld, C. Vanishing-overhead linear-scaling random phase approximation by Cholesky decomposition and an attenuated Coulomb-metric. *J. Chem. Theory Comput.* **2017**, *13*, 1647.

(34) Graf, D.; Beuerle, M.; Schurkus, H. F.; Luenser, A.; Savasci, G.; Ochsenfeld, C. Accurate and Efficient Parallel Implementation of an Effective Linear-Scaling Direct Random Phase Approximation Method. *J. Chem. Theory Comput.* **2018**, *14*, 2505.

(35) Paier, J.; Janesko, B. G.; Henderson, T. M.; Scuseria, G. E.; Grüneis, A.; Kresse, G. Hybrid functionals including random phase approximation correlation and second-order screened exchange. *J. Chem. Phys.* **2010**, *132*, No. 094103.

(36) Kohn, W.; Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **1965**, *140*, No. A1133.

(37) Bohm, D.; Pines, D. A collective description of electron interactions: III. Coulomb interactions in a degenerate electron gas. *Phys. Rev.* **1953**, *92*, No. 609.

(38) Langreth, D. C.; Perdew, J. P. The exchange-correlation energy of a metallic surface. *Solid State Commun.* **1975**, *17*, 1425.

(39) Gunnarsson, O.; Lundqvist, B. I. Exchange and correlation in atoms, molecules, and solids by the spin-density-functional formalism. *Phys. Rev. B* **1976**, *13*, No. 4274.

(40) Furche, F. Developing the random phase approximation into a practical post-Kohn−Sham correlation model. *J. Chem. Phys.* **2008**, *129*, No. 114105.

(41) Eshuis, H.; Yarkony, J.; Furche, F. Fast computation of molecular random phase approximation correlation energies using resolution of the identity and imaginary frequency integration. *J. Chem. Phys.* **2010**, *132*, No. 234114.

(42) Einstein, A. The foundation of the general theory of relativity. *Ann. Phys.* **1916**, *354*, 769.

(43) Ullrich, C. A. *Time-Dependent Density-Functional Theory: Concepts and Applications*; Oxford University Press: Oxford, 2011.

(44) Kaltak, M.; Klimes, J.; Kresse, G. Low scaling algorithms for the random phase approximation: Imaginary time and Laplace transformations. *J. Chem. Theory Comput.* **2014**, *10*, 2498.

(45) Weigend, F.; Furche, F.; Ahlrichs, R. Gaussian basis sets of quadruple zeta valence quality for atoms H−Kr. *J. Chem. Phys.* **2003**, *119*, 12753−12762.

(46) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.

(47) Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. RI-MP2: optimized auxiliary basis sets and demonstration of efficiency. *Chem. Phys. Lett.* **1998**, *294*, 143−152.

(48) Thompson, T. H.; Ochsenfeld, C. Integral partition bounds for fast and effective screening of general one-, two-, and many-electron integrals. *J. Chem. Phys.* **2019**, *150*, No. 044101.

(49) Kussmann, J.; Ochsenfeld, C. Pre-selective screening for matrix elements in linear-scaling exact exchange calculations. *J. Chem. Phys.* **2013**, *138*, No. 134114.

(50) 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.

(51) Kussmann, J.; Ochsenfeld, C. Hybrid CPU/GPU integral engine for strong-scaling ab initio methods. *J. Chem. Theory Comput.* **2017**, *13*, 3153−3159.

(52) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, No. 3865.

(53) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple [Phys. Rev. Lett. 77, 3865 (1996)]. *Phys. Rev. Lett* **1997**, *78*, No. 1396.

(54) Hättig, C. Optimization of auxiliary basis sets for RI-MP2 and RI-CC2 calculations: Core−valence and quintuple-ζ basis sets for H to Ar and QZVPP basis sets for Li to Kr. *Phys. Chem. Chem. Phys.* **2005**, *7*, 59−66.

(55) Kreppel, A.; Graf, D.; Laqua, H.; Ochsenfeld, C. Range-separated density-functional theory in combination with the random phase approximation: An accuracy benchmark. *J. Chem. Theory Comput.* **2020**, *16*, 2985−2994.

(56) Yang, Y.; van Aggelen, H.; Steinmann, S. N.; Peng, D.; Yang, W. Benchmark tests and spin adaptation for the particle-particle random phase approximation. *J. Chem. Phys.* **2013**, *139*, No. 174110.

(57) Mussard, B.; Reinhardt, P.; Ángyán, J. G.; Toulouse, J. Spin-unrestricted random-phase approximation with range separation: Benchmark on atomization energies and reaction barrier heights. *J. Chem. Phys.* **2015**, *142*, No. 154123.

(58) Heßelmann, A.; Görling, A. Random-phase approximation correlation methods for molecules and solids. *Mol. Phys.* **2011**, *109*, 2473−2500.

3.11 Publication XI: An effective sub-quadratic scaling atomic-orbital reformulation of the scaled opposite-spin RI-CC2 ground-state model using Cholesky-decomposed densities and an attenuated Coulomb-metric

F. Sacchetta, D. Graf, H. Laqua, M. A. Ambroise, J. Kussmann, A. Dreuw, C. Ochsenfeld

J. Chem. Phys. **157**, 104104 (2022).

Abstract

An atomic-orbital reformulation of the Laplace-transformed scaled opposite-spin (SOS) coupled cluster singles and doubles (CC2) model within the resolution of the identity (RI) approximation (SOS-RI-CC2) is presented that extends its applicability to molecules with several hundreds of atoms and triple-zeta basis sets. We exploit sparse linear algebra and an attenuated Coulomb metric to decrease the disk space demands and the computational efforts. In this way, an effective sub-quadratic computational scaling is achieved with our ω-SOS-CDD-RI-CC2 model. Moreover, Cholesky decomposition of the ground-state one-electron density matrix reduces the prefactor, allowing for an early crossover with the molecular orbital formulation. The accuracy and performance of the presented method are investigated for various molecular systems.

Reproduced from:

F. Sacchetta, D. Graf, H. Laqua, M. A. Ambroise, J. Kussmann, A. Dreuw,
C. Ochsenfeld
"An effective sub-quadratic scaling atomic-orbital reformulation of the scaled opposite-spin RI-CC2 ground-state model using Cholesky-decomposed densities and an attenuated Coulomb-metric"
J. Chem. Phys. **157**, 104104 (2022).

with the permission of AIP Publishing.
`https://aip.scitation.org/doi/10.1063/5.0098719`

The Journal of Chemical Physics

An effective sub-quadratic scaling atomic-orbital reformulation of the scaled opposite-spin RI-CC2 ground-state model using Cholesky-decomposed densities and an attenuated Coulomb metric

Cite as: J. Chem. Phys. **157**, 104104 (2022); https://doi.org/10.1063/5.0098719
Submitted: 11 May 2022 • Accepted: 28 July 2022 • Published Online: 08 September 2022

F. Sacchetta, D. Graf, H. Laqua, et al.

View Online Export Citation CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Accurate property prediction by second order perturbation theory: The REMP and OO-REMP hybrids
The Journal of Chemical Physics **157**, 104111 (2022); https://doi.org/10.1063/5.0105628

Molecular dipole moment learning via rotationally equivariant derivative kernels in molecular-orbital-based machine learning
The Journal of Chemical Physics **157**, 104109 (2022); https://doi.org/10.1063/5.0101280

Charge delocalization error in Piris natural orbital functionals
The Journal of Chemical Physics **157**, 104113 (2022); https://doi.org/10.1063/5.0102310

An effective sub-quadratic scaling atomic-orbital reformulation of the scaled opposite-spin RI-CC2 ground-state model using Cholesky-decomposed densities and an attenuated Coulomb metric

Cite as: J. Chem. Phys. 157, 104104 (2022); doi: 10.1063/5.0098719
Submitted: 11 May 2022 • Accepted: 28 July 2022 •
Published Online: 8 September 2022

View Online Export Citation CrossMark

F. Sacchetta,[1] D. Graf,[1] H. Laqua,[1] M. A. Ambroise,[2] J. Kussmann,[1] A. Dreuw,[2] and C. Ochsenfeld[1,a)]

AFFILIATIONS

[1] Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), Munich, Germany
[2] Chair of Theoretical and Computational Chemistry, Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany

[a)] Author to whom correspondence should be addressed: christian.ochsenfeld@cup.uni-muenchen.de

ABSTRACT

An atomic-orbital reformulation of the Laplace-transformed scaled opposite-spin (SOS) coupled cluster singles and doubles (CC2) model within the resolution of the identity (RI) approximation (SOS-RI-CC2) is presented that extends its applicability to molecules with several hundreds of atoms and triple-zeta basis sets. We exploit sparse linear algebra and an attenuated Coulomb metric to decrease the disk space demands and the computational efforts. In this way, an effective sub-quadratic computational scaling is achieved with our ω-SOS-CDD-RI-CC2 model. Moreover, Cholesky decomposition of the ground-state one-electron density matrix reduces the prefactor, allowing for an early crossover with the molecular orbital formulation. The accuracy and performance of the presented method are investigated for various molecular systems.

I. INTRODUCTION

Coupled cluster (CC) theory[1,2] is among the most successful approaches to obtain accurate correlation energies.[3,4] However, the large computational and memory demands prohibit its routine application to large molecular systems. Among the most popular models are, e.g., CCSD and CCSD(T), whose computational effort scales as $\mathcal{O}(M^6)$ and $\mathcal{O}(M^7)$, respectively,[5,6] where M is a measure for the system size. In an effort to reduce the computational scaling associated with the accurate CCSD model, the second-order approximate coupled cluster singles and doubles model (CC2) was proposed by Christiansen *et al.*[7] in 1995. The CC2 ground-state energy is expected to be of similar quality as the MP2 energy[7] and is computed with $\mathcal{O}(M^5)$ computational scaling (as canonical MP2). Moreover, despite being the simplest approximation of the coupled cluster hierarchy, the space demand and I/O effort needed for CC2 are severe bottlenecks, scaling as $\mathcal{O}(M^4)$. While low scaling MP2 methods[8–18] are now widely used to compute ground-state properties of systems with hundreds of atoms and large basis sets, CC2 is not limited to ground-state properties, highlighting the importance of developing efficient low scaling CC2 methods. An important step in the CC2 progress has been the use of the resolution of the identity (RI) approximation,[19,20] largely reducing the storage requirements needed for two-electron integrals while maintaining $\mathcal{O}(M^5)$ computational scaling.[21] As an alternative, the Cholesky decomposition of the two-electron integral matrix can be used.[22–25] In 2004, a simplified variant of Grimme's spin-component scaled (SCS)-MP2 method[26] was proposed by Jung *et al.*[8] in order to reduce the computational complexity. The so-called scaled opposite-spin (SOS)-MP2 method[8] completely neglects the calculation of the same-spin contributions and scales the opposite spin part by a factor $c_{os} = 1.3$, providing an accuracy comparable to that of the unscaled models.[8,9] Later, the combination of this SOS approximation[8,9] employing also the Laplace transform of the energy denominator[27–29] within the RI-approximation led to an efficient fourth-order scaling CC2 implementation by Winter and Hättig,[30] providing computational and memory/disk space savings already for small systems. Although the promising scaling behavior of the SOS-RI-CC2 model allows routine calculations of energies for systems with a few hundreds of atoms and large basis sets, it is essential to develop low or even linear scaling formulations in order to extend the CC2 applicability to even larger systems. To that end, approaches based on local molecular orbitals[31–34] and natural orbitals[35–37] have been used in the context of CC2 showing reduced computational scaling. However, to the best of our knowledge, atomic-orbital (AO) reformulations of CC2 have not yet been considered in literature and will be proposed in the present work for the ground-state energies as a first step toward reformulating CC2 excited state energies and properties. We reformulate the ground-state SOS-RI-CC2 equations presented by Winter and Hättig[30] in a pure atomic-orbital basis (Sec. II B), taking as reference the previous studies on RI-MP2,[14,16,18,38,39] CC,[40] and CPSCF[41] theory. The AO approach has the crucial advantage of depending explicitly on the one-particle density matrix **P** (and its virtual counterpart **Q**) that contains all the necessary information regarding the reference determinant and becomes sparse for growing system sizes in contrast to molecular orbitals. Such an AO-SOS-RI-CC2 approach can be further improved by introducing the Cholesky factorization of the one-particle density matrix, as previously shown by our group.[14,18,41–44] The resulting SOS-CDD-RI-CC2 model, described in Sec. II D, is based on so-called Cholesky orbitals inheriting the locality from the density matrix and whose number is equal to the number of MOs in the occupied space. Consequently, the SOS-CDD-RI-CC2 implementation features a lower prefactor and reduced memory demands[14,18] with respect to AO-SOS-RI-CC2 and shows an early crossover with the MO formulation. Additional improvements in terms of locality (and, therefore, performance) are gained by employing the RI Coulomb metric attenuated by the complementary error function that introduces only small errors.[18,45] As a result, quadratic or even sub-quadratic scaling is obtained by exploiting the locality of the atomic/Cholesky orbitals and efficient sparse algebra routines, bypassing localization procedures of any kind. This allows for a more direct control of the accuracy, contrary to previous studies on local CC2 implementations[31,32,46] based on the Pulay and Saebø approach,[10,11] where excitations between spatially distant orbitals are neglected if their contributions to the correlation energy are negligible. Accordingly, *a priori* restricted local MO (LMO) pair lists and pair specific excitation subspaces of projected AO (PAO) are specified and amplitudes outside these lists are neglected.[31,32] The downsides are the strong dependence on a successful localization procedure and the larger size of the correlation domains required for small errors.[46]

This article is structured as follows: First, in Sec. II A, we summarize the key components of the ground-state MO-SOS-RI-CC2 model proposed by Winter and Hättig.[30] Section II B describes the reformulation of the equations in the atomic-orbital basis. In Sec. II C, we introduce the Coulomb metric attenuated by the complementary error function for the RI-approximation, which moves the scaling toward sub-quadratic. The Cholesky decomposition of density matrices is introduced in Sec. II D, significantly reducing the prefactor of the AO implementation. Then, we provide an outline of our ω-SOS-CDD-RI-CC2 method in Sec. II E and discuss the accuracy (Sec. IV A), scaling behavior (Sec. IV B), and performance (Sec. IV C). Finally, we show the performance of our ω-SOS-CDD-RI-CC2 method when applied to systems of chemical interest.

II. THEORY

The CC2 model has been introduced by Christiansen *et al.*[7] as an approximation to the CCSD model, where the singles are treated as zeroth-order parameters in terms of the fluctuation potential. On the other hand, the double excitations are treated at first order in the fluctuation potential as in the MP2 theory.[7] The SOS-CC2 energy is given by

$$E^{\text{sos}} = \langle \text{HF}|\hat{H} + c_{\text{os}}[\hat{H}, T_2^{\text{os}}]|\text{HF}\rangle, \tag{1}$$

where $\hat{H}$ is the similarity-transformed Hamiltonian and T_2^{os} is the two-electron excitation operator, which acts on two electrons with different spins.[8,30] The cluster amplitudes are determined by solving the coupled cluster equations,

$$\Omega_{\mu_1} = \langle \mu_1|\hat{H} + c_{\text{os}}[\hat{H}, T_2^{\text{os}}]|\text{HF}\rangle = 0, \tag{2}$$

$$\Omega_{\mu_2} = \langle \mu_2^{\text{os}}|\hat{H} + [F, T_2^{\text{os}}]|\text{HF}\rangle = 0, \tag{3}$$

where Ω_μ are the so-called coupled cluster vector functions and F is the Fock operator. We use the same scaling factor as for SOS-MP2,[8,30] $c_{\text{os}} = 1.3$. Since the CC2 singles (t_{μ_1}) and doubles (t_{μ_2}) amplitudes depend on each other, the nonlinear Eqs. (2) and (3) must be solved iteratively. The CC2 calculations start with an initial guess for the singles amplitudes that are usually set to zero. The optimization scheme is based on the quasi-Newton equation for the nth iteration,[47]

$$\Delta t_{ai}^{(n)} = -\frac{\Omega_{ai}(t^{(n)})}{\varepsilon_a - \varepsilon_i}. \tag{4}$$

The correction $\Delta t_{ai}^{(n)}$ is then used to obtain the new singles amplitudes $t_{ai}^{(n+1)}$ by

$$t_{ai}^{(n+1)} = t_{ai}^{(n)} + \Delta t_{ai}^{(n)}. \tag{5}$$

The convergence may be improved significantly by application of the DIIS acceleration scheme[48] and it is reached when the norm of Ω_{μ_1} and the variation of the energy are below the given thresholds ϑ_{oconv} (i.e., 10^{-5}) and ϑ_{econv} (i.e., 10^{-6}), respectively.

A. MO-SOS-RI-CC2

In 2011, Winter and Hättig proposed an efficient fourth-order scaling implementation to compute the SOS-RI-CC2 ground-state energies,[30] given as the sum of the HF energy (E_{HF}) and the disconnected (E_{D}) and connected (E_{C}) double amplitudes contributions as follows:

$$\begin{aligned} E_{\text{CC2}}^{\text{SOS}} &= E_{\text{HF}} + E_{\text{D}} + E_{\text{C}} \\ &= E_{\text{HF}} + c_{\text{os}}\sum_{aibj} t_{ai}t_{bj}(ai|bj) + c_{\text{os}}\sum_{aibj} t_{aibj}(ai|bj). \end{aligned} \tag{6}$$

In order to solve Eq. (6) with a fourth-order scaling, the t_{aibj} orbital energy denominator is factorized using the Laplace transform technique followed by a numerical quadrature according to

$$\frac{1}{\varepsilon_{aibj}} = \int_0^\infty e^{-\varepsilon_{aibj}} dt = \sum_\tau^{N_\tau} w_\tau e^{-\varepsilon_{aibj} t_\tau}, \tag{7}$$

where $\varepsilon_{aibj} = \varepsilon_a - \varepsilon_i + \varepsilon_b - \varepsilon_j$ is the MO energy denominator and w_τ and the t_τ are the weights and grid points of the numerical quadrature procedure, respectively, which are optimized using, e.g., the minimax approximation in order to reduce the error.[49,50] In addition, the RI-approximation decomposes the four-index electron repulsion integrals (ERIs) to bypass an expensive four-index AO to MO transformation,[20,21,30]

$$(pq|rs) = \sum_P B^P_{pq} B^P_{rs}, \tag{8}$$

$$B^P_{pq} = \sum_{\mu\nu,Q} C_{\mu p} C_{\nu q} (\mu\nu|Q) J^{-1/2}_{QP}, \tag{9}$$

$$J_{PQ} = \left(P\left|\frac{1}{r_{12}}\right|Q\right), \tag{10}$$

using the Mulliken notation for two-, three-, and four-center integrals. In Eqs. (8) and (9), p, q, r, and s are MO indices, μ, ν are AO indices, and P, Q are auxiliary functions for the RI-approximation. $\mathbf{C}$ is the MO coefficient matrix. Moreover, some of the CC2 three-center integrals are modified according to

$$\hat{B}^P_{pq} = \sum_{\mu\nu} \Lambda^{\rm p}_{\mu p} \Lambda^{\rm h}_{\nu q} (\mu\nu|P), \tag{11}$$

with the transformation matrices $\Lambda^{\rm p}$ and $\Lambda^{\rm h}$ given by

$$\Lambda^{\rm p} = \mathbf{C}(\mathbf{I} - \mathbf{t}_1^T) \qquad \Lambda^{\rm h} = \mathbf{C}(\mathbf{I} + \mathbf{t}_1) \qquad \mathbf{t}_1 = \begin{pmatrix} 0 & 0 \\ \{t_{ai}\} & 0 \end{pmatrix}. \tag{12}$$

The t_1-dependent doubles amplitudes are then calculated as

$$t^{ab}_{ij} = -\sum_\tau^{N_\tau} w_\tau \sum_P \hat{B}^P_{ai} e^{-\varepsilon_{ai} t_\tau} \cdot \hat{B}^P_{bj} e^{-\varepsilon_{bj} t_\tau}, \tag{13}$$

and the expression for the MO-SOS-RI-CC2 correlation energy can be rewritten as

$$E_{\rm D} = c_{\rm os} \sum_{aibj} t_{ai} t_{bj} \sum_Q B^Q_{ai} B^Q_{bj}, \tag{14}$$

$$\begin{aligned} E_{\rm C} &= -c_{\rm os} \sum_\tau w_\tau \sum_{aibj} \sum_{PQ} \hat{B}^P_{ai} \hat{B}^P_{bj} e^{-\varepsilon_{ai} t_\tau} e^{-\varepsilon_{bj} t_\tau} B^Q_{ai} B^Q_{bj} \\ &= -c_{\rm os} \sum_\tau w_\tau \sum_{PQ} N^{QP}_\tau N^{QP}_\tau, \end{aligned} \tag{15}$$

where N^{QP}_τ is given in Table I. The converged t_1-amplitudes are obtained at the end of the iterative optimization, where, in each iteration, doubles amplitudes are computed "on-the-fly" by inserting Eq. (13) into the singles part of the vector function [Eq. (2)], yielding the working expressions in terms of computational convenient intermediates,

$$\Omega_{ai}(t^{(n)}) = \Omega^G_{ai}(t^{(n)}) + \Omega^H_{ai}(t^{(n)}) + \Omega^I_{ai}(t^{(n)}) + \Omega^J_{ai}(t^{(n)}), \tag{16}$$

TABLE I. Explicit expressions of the singles amplitudes vector function terms for SOS-RI-CC2 in the MO and the SOS-CDD-RI-CC2 formulation. The Einstein notation is used.

MO-SOS-RI-CC2	SOS-CDD-RI-CC2
$\Omega^G_{ai} = \hat{B}^Q_{ac} \hat{Y}^Q_{ci}$	$\Omega^G_{\mu\nu} = \hat{Q}_{\mu\mu'} (B^Q_{\mu'\nu'} \hat{Y}^Q_{\nu'\underline{i}}) L_{\nu\underline{i}}$
$\Omega^H_{ai} = -\hat{Y}^Q_{ak} \hat{B}^Q_{ki}$	$\Omega^H_{\mu\nu} = (-\hat{Y}^Q_{\mu\underline{k}} \hat{B}^Q_{\underline{k}\underline{i}}) L_{\nu\underline{i}}$
$\Omega^I_{ai} = -c_{\rm os} w_\tau n^P_\tau \hat{B}^P_{ai} e^{-\varepsilon_{ai} t_\tau}$	$\Omega^I_{\mu\nu} = (-c_{\rm os} \tilde{n}^P_\tau\, {}^{\tau}\hat{B}^P_{\mu\underline{i}}) L_{\nu\underline{i}}$
$\Omega^J_{ai} = \hat{F}_{ai}$	$\Omega^J_{\mu\nu} = \hat{Q}_{\mu\mu'} \hat{F}_{\mu'\nu'} \hat{\bar{P}}_{\nu'\nu}$
$\hat{Y}^Q_{ai} = -c_{\rm os} w_\tau \hat{B}^P_{ai} N^{QP}_\tau e^{-\varepsilon_{ai} t_\tau}$	$\hat{Y}^Q_{\mu\underline{i}} = -c_{\rm os}\, {}^{\tau}\hat{B}^P_{\mu\underline{i}} \tilde{N}^{QP}_\tau$
$N^{QP}_\tau = B^Q_{bj} \hat{B}^P_{bj} e^{-\varepsilon_{bj} t_\tau}$	$N^{RS}_\tau = B^R_{\mu j}\, {}^{\tau}\hat{B}^S_{\mu\underline{j}}$
$n^P_\tau = \hat{B}^P_{bj} \hat{F}_{jb} e^{-\varepsilon_{bj} t_\tau}$	$n^R_\tau = {}^{\tau}\hat{B}^{\bar{R}}_{\mu\underline{j}} \hat{F}_{\underline{j}\mu}$
$\hat{B}^P_{ai} = \Lambda^p_{\mu a} \Lambda^h_{\nu i} B^P_{\mu\nu} J^{-1/2}_{PQ}$	$\tilde{N}^{QP}_\tau = J^{-1}_{QR} N^{RS}_\tau J^{-1}_{SP}$
	$\tilde{n}^P_\tau = J^{-1}_{PR} n^R_\tau$

whose explicit expressions are listed in Table I. The solution of Eq. (16) requires a considerable amount of memory and disk space for the calculation of intermediates. The memory limitations are overcome by employing batching algorithms to evaluate the three-center integrals and the intermediates (see the algorithm proposed by Winter and Hättig[30]).

The MO-SOS-RI-CC2 equations in Table I have been implemented in the FermiONs++[51–53] program as proposed by Winter and Hättig.[30] In contrast, we do not batch over the Laplace quadrature points (see the supplementary material).

B. Reformulation of the SOS-RI-CC2 method in the AO basis

In order to extend the applicability of the SOS-RI-CC2 model to systems with hundreds of atoms and large basis sets, we reformulated the MO-expressions summarized in Table I in the AO basis. The quasi-Newton expression for the correction of the CC2 singles amplitudes [Eq. (4)] has an important role during this reformulation. Indeed, we can back-transform $\Delta t^{(n)}_{ai}$ [Eq. (4)] to the AO basis [Eq. (17)] via the Laplace transform of the energy denominator $(\varepsilon_a - \varepsilon_i)$, along the lines of Beer and Ochsenfeld[41] for the U^x_{ai} matrix,

$$\sum_{ai} C_{\mu a} \Delta t^{(n)}_{ai} C_{\nu i} = -\sum_\alpha^{N_\alpha} w_\alpha \sum_{ai} e^{-\varepsilon_a t_\alpha} C_{\mu a} \Omega_{ai} C_{\nu i} e^{\varepsilon_i t_\alpha}. \tag{17}$$

In that way, the SOS-RI-CC2 equations are rewritten in the AO basis employing a convenient transformation that generates a formulation where all integrals and excitation amplitudes are written with AO indices. Even though this procedure allows us to completely avoid the canonical MO basis,[41] it increases the computational effort by a factor of N_α, that is, the number of Laplace quadrature points employed in [Eq. (17)] (generally 6–10 points). In order to overcome this drawback, we limited the back-transformation to the vector function for the singles amplitudes [the numerator in Eq. (4)],

$$\sum_{ai} C_{\mu a}\Omega_{ai}(t^{(n)})C_{\nu i} = \Omega_{\mu\nu}(t^{(n)}) \\ = \Omega^{G}_{\mu\nu}(t^{(n)}) + \Omega^{H}_{\mu\nu}(t^{(n)}) + \Omega^{I}_{\mu\nu}(t^{(n)}) + \Omega^{J}_{\mu\nu}(t^{(n)}). \tag{18}$$

Such a strategy enables the reformulation of the intermediates in the AO basis, while the use of canonical MOs is limited only to Eq. (4) and does not affect the overall efficiency of the method. In the following, the most important steps in the reformulation of the CC2 method in the AO basis will be presented. The computational scaling of the main SOS-RI-CC2 steps is reported in Table II. The Einstein summation convention is used in the following equations.

The first two contributions to the vector function are reformulated according to

$$\Omega^{G}_{\mu\nu} = C_{\mu a}\Omega^{G}_{ai}C_{\nu i} = C_{\mu a}\hat{B}^{Q}_{ac}\hat{Y}^{Q}_{ci}C_{\nu i} = \hat{Q}_{\mu\sigma}B^{Q}_{\sigma\lambda}\hat{Y}^{Q}_{\lambda\nu}, \tag{19}$$

$$\Omega^{H}_{\mu\nu} = C_{\mu a}\Omega^{H}_{ai}C_{\nu i} = -C_{\mu a}\hat{Y}^{Q}_{ak}\hat{B}^{Q}_{ki}C_{\nu i} = -\hat{Y}^{Q}_{\mu\sigma}B^{Q}_{\sigma\lambda}\hat{P}_{\lambda\nu}, \tag{20}$$

with the CC2 virtual $(\hat{Q})$ and occupied $(\hat{P})$ densities given by

$$\hat{Q}_{\mu\nu} = C_{\mu d}\Lambda^{p}_{\nu d} = Q_{\mu\nu} - Q_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}P_{\nu'\nu}, \tag{21}$$

$$\hat{P}_{\mu\nu} = \Lambda^{h}_{\mu l}C_{\nu l} = P_{\mu\nu} + Q_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}P_{\nu'\nu}, \tag{22}$$

where $Q_{\mu\nu} = \sum_a C_{\mu a}C_{\nu a}$ and $P_{\mu\nu} = \sum_i C_{\mu i}C_{\nu i}$ are the virtual and occupied ground-state densities and $\mathbf{S}$ is the overlap matrix. The formation of the densities scales cubically with the system size, but its computational cost is negligible compared to other steps. The intermediate $\hat{\mathbf{Y}}$ is computed as

$$\hat{Y}^{Q}_{\mu\nu} = -c_{\text{os}}\,{}^{\tau}\hat{B}^{P}_{\mu\nu}\tilde{N}^{QP}_{\tau}, \tag{23}$$

where the three-center integrals and the intermediates $\tilde{N}^{QP}_{\tau}$ depending on the Laplace quadrature points τ are given by

$${}^{\tau}\hat{B}^{P}_{\mu\nu} = \hat{Q}^{\tau}_{\mu\mu'}B^{P}_{\mu'\nu'}\hat{P}^{\tau}_{\nu'\nu}, \tag{24}$$

$$\tilde{N}^{QP}_{\tau} = J^{-1}_{QR}N^{RS}_{\tau}J^{-1}_{SP}, \tag{25}$$

$$N^{RS}_{\tau} = B^{R}_{bj}\hat{B}^{S}_{bj}e^{-\varepsilon_{bj}t_{\tau}} = B^{R}_{\mu\nu}\hat{Q}^{\tau}_{\mu\sigma}B^{S}_{\sigma\lambda}\hat{P}^{\tau}_{\lambda\nu} = B^{R}_{\mu\nu}\,{}^{\tau}\hat{B}^{S}_{\mu\nu}. \tag{26}$$

The matrices $\hat{Q}^{\tau}$ and $\hat{P}^{\tau}$ are the CC2 virtual and occupied pseudodensities, respectively,

$$\hat{Q}^{\tau}_{\mu\nu} = w^{\frac{1}{4}}_{\tau}C_{\mu d}e^{-\varepsilon_{d}t_{\tau}}\Lambda^{p}_{\nu d} = Q^{\tau}_{\mu\nu} - Q^{\tau}_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}P_{\nu'\nu}, \tag{27}$$

$$\hat{P}^{\tau}_{\mu\nu} = w^{\frac{1}{4}}_{\tau}\Lambda^{h}_{\mu l}e^{\varepsilon_{l}t_{\tau}}C_{\nu l} = P^{\tau}_{\mu\nu} + Q_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}P^{\tau}_{\nu'\nu}, \tag{28}$$

with

$$Q^{\tau}_{\mu\nu} = w^{\frac{1}{4}}_{\tau}C_{\mu a}e^{-\varepsilon_{a}t_{\tau}}C_{\nu a}, \tag{29}$$

$$P^{\tau}_{\mu\nu} = w^{\frac{1}{4}}_{\tau}C_{\mu i}e^{\varepsilon_{i}t_{\tau}}C_{\nu i}. \tag{30}$$

An example of the sparsity pattern of $\hat{Q}^{\tau}_{\mu\nu}$ for the first Laplace quadrature point is displayed in Fig. 1 for the linear alkane $C_{320}H_{642}$. The remaining contributions to the singles vector function are given by

$$\Omega^{I}_{\mu\nu} = C_{\mu a}\Omega^{I}_{ai}C_{\nu i} = -c_{\text{os}}\tilde{n}^{P\tau}_{\tau}\hat{B}^{P}_{\mu\nu}, \tag{31}$$

$$\Omega^{J}_{\mu\nu} = C_{\mu a}\Omega^{J}_{ai}C_{\nu i} = \hat{Q}_{\mu\mu'}\hat{F}_{\mu'\nu'}\hat{P}_{\nu'\nu}, \tag{32}$$

where

$$\tilde{n}^{P}_{\tau} = n^{R}_{\tau}J^{-1}_{RP}, \tag{33}$$

$$n^{R}_{\tau} = \hat{F}_{jb}\hat{B}^{R}_{bj}e^{-\varepsilon_{bj}t_{\tau}} = \hat{F}_{\mu\nu}\hat{Q}^{\tau}_{\nu\sigma}B^{R}_{\sigma\lambda}\hat{P}^{\tau}_{\lambda\mu} = \hat{F}_{\mu\nu}\,{}^{\tau}\hat{B}^{R}_{\mu\nu}. \tag{34}$$

Moreover, the modified Fock matrix is built from the CC2 density, showing an asymptotically quadratic scaling for the calculation of the Coulomb contributions:

$$\hat{F}_{\mu\nu} = h_{\mu\nu} + \sum_{\sigma\lambda}\hat{P}_{\sigma\lambda}[2(\mu\nu|\lambda\sigma) - (\mu\sigma|\lambda\nu)]. \tag{35}$$

Due to the long-range nature of the electron–electron interaction operator $(\frac{1}{r_{12}})$, the formation of the three- and two-center integrals ($B^{P}_{\mu\nu}$ and J_{PQ}) shows a quadratic scaling with system size while the inversion of the two-center integrals scales cubically. These steps are carried out only once and the time demands are not significant as compared to the rest of the CC2 calculation. In order to retain the sparsity of the three-center integrals, the multiplication with the dense J^{-1}_{PQ} matrix is delayed until the N^{RS}_{τ} and n^{R}_{τ} intermediates are formed.

TABLE II. Formal and asymptotic computational scaling (with the number of orbitals *N*) for key steps of ω-SOS-CDD-RI-CC2 models within the RI standard Coulomb metric $(\omega = 0)$ and overlap metric $(\omega \to \infty)$.

		Asymptotic scaling	
Step	Formal scaling	$\omega = 0$	$\omega \to \infty$
Cholesky decompose **P**	N^3	$\mathcal{O}(N)$	$\mathcal{O}(N)$
Form **J**	N^2	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
Invert **J**	N^3	$\mathcal{O}(N^3)$	$\mathcal{O}(N^3)$
Form $B^{P}_{\mu\nu}$	N^2	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
Form $\hat{B}^{P}_{\mu\underline{i}}$ and $B^{P}_{\mu\underline{i}}$	N^4	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
Form ${}^{\tau}\hat{B}^{P}_{\mu\underline{i}}$	N^3	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
Form N^{RS}_{τ}	N^4	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
Form n^{R}_{τ}	N^3	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
Form $\tilde{N}^{QP}_{\tau}$	N^3	$\mathcal{O}(N^3)$	$\mathcal{O}(N^3)$
Form $\tilde{n}^{P}_{\tau}$	N^2	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
Form $\Omega^{I}_{\mu\nu}$	N^3	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
Form $\hat{Y}^{Q}_{\mu\underline{i}}$	N^4	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
Form $\hat{B}^{Q}_{\underline{k}\underline{i}}$	N^4	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
Form $\Omega^{G}_{\mu\nu}$	N^4	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
Form $\Omega^{H}_{\mu\nu}$	N^4	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
Form $\Omega^{J}_{\mu\nu}$	N^3	$\mathcal{O}(N^3)$	$\mathcal{O}(N^3)$

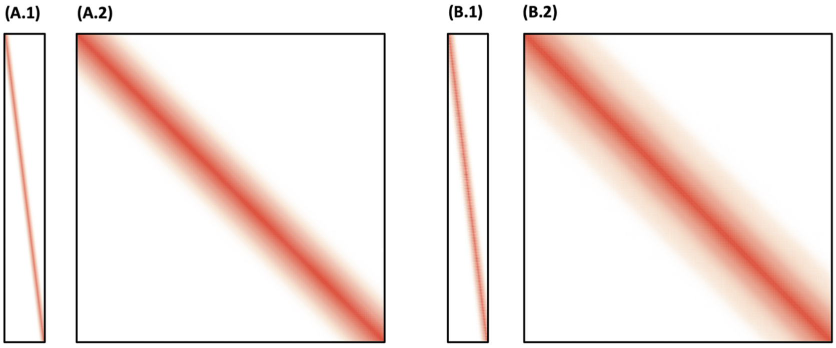


FIG. 1. Sparsity patterns of occupied and virtual CC2 pseudo-densities for the linear alkane $C_{320}H_{642}$, in the def2-svp (a) and def-tzvp (b) basis sets with a cutoff threshold equal to 10^{-10}. (a.1, b.1) $\hat{P}^{\tau}_{\mu\underline{i}}$ from Eq. (46) and (a.2, b.2) $\hat{Q}^{\tau}_{\mu\nu}$ from Eq. (27).

Once all the contributions to $\Omega_{\mu\nu}$ are computed, we transform the vector function to the MO basis according to

$$\Omega_{ai}(t^{(n)}) = C_{\mu a} S_{\mu\mu'} \Omega_{\mu'\nu'}(t^{(n)}) S_{\nu'\nu} C_{\nu i}, \tag{36}$$

and the correction Δt_{ai} is then given as in the canonical MO implementation [Eq. (4)]. The updated CC2 single amplitudes in the AO basis are finally given by

$$\Delta t^{(n)}_{\mu\nu} = C_{\mu a} \Delta t^{(n)}_{ai} C_{\nu i}, \tag{37}$$

$$t^{(n+1)}_{\mu\nu} = t^{(n)}_{\mu\nu} + \Delta t^{(n)}_{\mu\nu}. \tag{38}$$

C. Using the erfc-attenuated Coulomb metric: ω-SOS-AO-RI-CC2

The presented reformulation in the atomic-orbital basis results in a cubic asymptotic scaling $\mathcal{O}(M^3)$ with molecular size, since the auxiliary functions couple to the AO basis-function pairs with a $\frac{1}{r_{12}}$ decay within the standard Coulomb metric (see Table II). Density fitting calculations are generally performed with the Coulomb metric because it yields the most accurate results for the commonly employed auxiliary basis sets.[9,43,45] The slow long-range decay of this metric has no disadvantage when transforming to the canonical MO basis. However, in local bases (i.e., Cholesky and atomic orbitals), no sparsity can be gained. Therefore, the overlap metric represents a promising alternative due to the increased locality and sparsity.[43,45] The drawback is the decreased accuracy. A third choice would be a metric that combines the accuracy of the Coulomb metric and the sparsity of the local overlap metric. The attenuated Coulomb metric given by

$$g = \frac{\mathrm{erfc}(\omega r_{12})}{r_{12}} \tag{39}$$

has this property, as described by Jung *et al.*[45] in SOS-MP2 calculations. The three-, two-, and four-center integrals are then given by

$$(\mu\nu|P) = \left(\mu\nu\middle|\frac{\mathrm{erfc}(\omega r_{12})}{r_{12}}\middle|P\right), \tag{40}$$

$$(S_\omega)_{PQ} = \left(P\middle|\frac{\mathrm{erfc}(\omega r_{12})}{r_{12}}\middle|Q\right), \tag{41}$$

$$\mathbf{J}_{\boldsymbol{\omega}} = \mathbf{S}^{-1}_{\omega}\mathbf{J}\mathbf{S}^{-1}_{\omega}, \tag{42}$$

$$(\mu\nu|\sigma\lambda) = (\mu\nu|P) J^{PQ}_{\omega} (Q|\sigma\lambda). \tag{43}$$

As with the standard Coulomb metric, we postpone the multiplication with the two-center integrals to the step in Eq. (33) in order to preserve locality through the previous time-determining steps. The extent of locality is controlled by the parameter ω, recovering the Coulomb metric at $\omega = 0$ and approaching the overlap metric as ω increases. If $\omega \to \infty$, we recover the overlap metric. Of course, the larger ω is, the less accurate are the results[45] (at least when standard RI basis sets are employed). In the present work, we employ the erfc-Coulomb metric within CC2 for the first time, increasing the sparsity of the intermediates and allowing a further reduction of the scaling in all time-determining steps. Indeed, the effective computational scaling of ω-SOS-CDD-RI-CC2 is sub-quadratic for $\omega > 0$.

D. Reduction of the basis set scaling: ω-SOS-CDD-RI-CC2

Despite the above formulation being suited for large systems and moderate basis sets, its applicability to large basis sets is hampered by the scaling with the basis set size N_{basis} and auxiliary basis set size N_{aux}. The formal scaling is increased from $\mathcal{O}(N_{\text{virt}} N_{\text{occ}} N^2_{\text{aux}})$ to $\mathcal{O}(N^2_{\text{basis}} N^2_{\text{aux}})$ for a fixed molecular size. In order to reduce the complexity and improve the performance, we employ Cholesky decomposition of the occupied ground-state density matrix with complete pivoting[54,55] and the idempotency relation of the occupied pseudo-density matrix, as proposed by Graf *et al.*[44] and Glasbrenner *et al.*,[18]

$$\mathbf{P} = \mathbf{L}\mathbf{L}^T \quad \mathbf{P}^{\boldsymbol{\tau}} = \mathbf{P}^{\boldsymbol{\tau}}\mathbf{S}\mathbf{P} = \mathbf{P}^{\boldsymbol{\tau}}\mathbf{S}\mathbf{L}\mathbf{L}^T. \tag{44}$$

The Cholesky factorization scales formally as N^3, but it has a very low prefactor. Moreover, it can be carried out with asymptotic linear scaling.[56] The columns of $\mathbf{L}$ can be considered as the coefficients of localized occupied MOs that we will call Cholesky orbitals[14] and tag with $\underline{i}, \underline{j}, \underline{k}$ indices. They inherit the locality from the density matrix[57] and their number is equal to the number of MOs. Like the pseudo-density matrices, the CC2 occupied density matrix is invariant with respect to projection onto the occupied space and can now be written as

$$\hat{P}^{\tau}_{\mu\nu} = \hat{P}^{\tau}_{\mu\nu} S_{\nu\sigma} L_{\sigma\underline{i}} L_{\nu\underline{i}} = \hat{P}^{\tau}_{\mu\underline{i}} L_{\nu\underline{i}}, \tag{45}$$

$$\begin{aligned}\hat{P}^{\tau}_{\mu\underline{i}} &= P_{\mu\sigma}S_{\sigma\lambda}P^{\tau}_{\lambda\underline{i}} + Q_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}P_{\nu'\sigma'}S_{\sigma'\lambda'}P^{\tau}_{\lambda'\underline{i}} \\ &= (L_{\mu\underline{j}} + Q_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}L_{\nu'\underline{j}})L_{\sigma'\underline{j}}S_{\sigma'\lambda'}P^{\tau}_{\lambda'\underline{i}},\end{aligned} \tag{46}$$

and its sparsity pattern is depicted in Fig. 1 for $C_{320}H_{642}$. The derived ω-SOS-CDD-RI-CC2 expressions for the vector function contributions are given in Table I, where the three-center integrals, then, read

$$B^{Q}_{\mu\underline{i}} = B^{Q}_{\mu\nu}L_{\nu\underline{i}}, \tag{47}$$

$$\hat{B}^{Q}_{\underline{k}\underline{i}} = L_{\mu\underline{k}}B^{Q}_{\mu\nu}\hat{P}_{\nu\underline{i}}, \tag{48}$$

$$^{\tau}\hat{B}^{P}_{\mu\underline{i}} = \hat{Q}^{\tau}_{\mu\mu'}B^{P}_{\mu'\nu'}\hat{P}^{\tau}_{\nu'\underline{i}}. \tag{49}$$

The introduction of Cholesky orbitals reduces the formal scaling behavior to $\mathcal{O}(N_{\text{basis}}N_{\text{occ}}N^2_{\text{aux}})$. The half-transformed three-center integrals $B^{Q}_{\mu\underline{i}}$ do not depend on the singles amplitudes and, hence, can be computed only once at the beginning and stored on disk. On the other hand, $^{\tau}\hat{B}^{P}_{\mu\underline{i}}$ in Eq. (49) is computed for each Laplace point and each iteration. We decided to use the idempotency relation in Eq. (44) not only to improve the efficiency but also to reduce the memory requirements. Each $B^{Q}_{\mu\nu}$ matrix is precontracted as shown in Eq. (50) and the resulting half-transformed $\hat{B}^{P}_{\mu\underline{i}}$ quantity is independent of the Laplace points. The Laplace point-dependent three-center integrals are then obtained from the half-transformed integrals reducing both I/O and computational effort, as shown in Eq. (51),

$$\hat{B}^{P}_{\mu\underline{j}} = B^{P}_{\mu\nu}(L_{\nu\underline{j}} + Q_{\nu\nu'}S_{\nu'\sigma}t_{\sigma\lambda}S_{\lambda\nu''}L_{\nu''\underline{j}}), \tag{50}$$

$$^{\tau}\hat{B}^{P}_{\mu\underline{i}} = \hat{Q}^{\tau}_{\mu\mu'}B^{Q}_{\mu'\underline{j}}(L_{\sigma\underline{j}}S_{\sigma\lambda}P^{\tau}_{\lambda\underline{i}}). \tag{51}$$

The reason why we avoided Cholesky factorization of the virtual density matrix is twofold. First, the sparsity of the virtual density is not well preserved and, hence, the rank reduction is often counteracted by this loss of sparsity. Second, correlation methods such as CC2 require large basis sets for accurate results, in which case, $\text{N}_{\text{basis}} \approx \text{N}_{\text{virt}}$ and the rank reduction from factorization is negligible.

Finally, the ω-SOS-CDD-RI-CC2 ground-state energy is computed as

$$E^{\text{SOS}}_{\text{CC2}} = E_{\text{D}} + E_{\text{C}} = c_{\text{os}}t_{\mu\underline{i}}B^{P}_{\mu\underline{i}}J^{PQ}_{\omega}t_{\nu\underline{j}}B^{Q}_{\nu\underline{j}} - c_{\text{os}}\tilde{N}^{QP}_{\tau}N^{QP}_{\tau}, \tag{52}$$

with

$$t_{\mu\underline{i}} = Q_{\mu\mu'}S_{\mu'\sigma}t_{\sigma\lambda}S_{\lambda\nu'}L_{\nu'\underline{i}} \tag{53}$$

and $\tilde{N}^{QP}_{\tau}$ and N^{QP}_{τ} given by Eqs. (25) and (26).

E. Outline of the low scaling implementation: A minimal-overhead batching

The available memory on a single computing node is easily exceeded by CC2 memory requirements in both MO and AO basis. Therefore, we introduced batching schemes for evaluating intermediates as three-center integrals, intermediates N^{QP}_{τ}, $\hat{Y}^{Q}_{\mu\underline{i}}$, and contributions to the vector function. The MO implementation is not discussed here; its algorithms are provided in the supplementary material.

First, we compute the $B^{P}_{\mu\nu}$ and half-transformed $B^{P}_{\underline{i}\mu}$ matrices and store them on disk. These three-center integrals are computed only once because they do not depend on the single amplitudes. Moreover, in each iteration, the half-transformed three-center integrals $\hat{B}^{P}_{\mu\underline{i}}$ are computed and stored on disk in order to alleviate the memory limitation problem.

We introduce an optimized batching scheme based on a Lagrangian formulation, where the optimal number of batches is computed by minimizing the batching overhead.[58]

- As proposed in the optimal batching scheme by Drontschenko *et al.*[58] for the response function in RPA, we compute the intermediates N^{QP}_{τ} (and n^{P}_{τ}) as shown

```
for aux-batch-1 do
    for P ∈ aux-batch-1 do
        read $B^{P}_{\mu\nu}$ $\forall\mu,\nu$
        $B^{Q}_{\underline{i}\mu} = L_{\mu\underline{i}}B^{P}_{\mu\nu}$ $\forall\underline{i},\mu$;
        $\hat{B}^{P}_{\mu\underline{j}} = B^{P}_{\mu\nu}(L_{\nu\underline{j}} + Q_{\nu\nu'}S_{\nu'\sigma}t_{\sigma\lambda}S_{\lambda\nu''}L_{\nu''\underline{j}})$ $\forall\underline{j},\mu$
        write $B^{Q}_{\underline{i}\mu}$ and $\hat{B}^{P}_{\mu\underline{j}}$ on disk
    end for
end for
for aux-batch-1 do
    for AO-batch do
        for P ∈ aux-batch-1 do
            read $\hat{B}^{P}_{\mu'\underline{j}}$ $\forall\mu',\underline{j}$
            for all $\tau$ do
                $^{\tau}\hat{B}^{P}_{\mu\underline{i}} = \hat{Q}^{\tau}_{\mu\mu'}\hat{B}^{P}_{\mu'\underline{j}}(L_{\sigma\underline{j}}S_{\sigma\lambda}P^{\tau}_{\lambda\underline{i}})\forall\underline{i},\mu \in$ AO-batch;
            end for
        end for
        for all $\tau$ do
            for P ∈ aux-batch-1 do
                $n^{P}_{\tau} += {}^{\tau}\hat{B}^{P}_{\mu\underline{i}}\hat{F}_{\mu\underline{i}}\forall\underline{i},\mu \in$ AO-batch
            end for
        end for
        for aux-batch-2 do
            for Q ∈ aux-batch-2 do
                read $B^{Q}_{\underline{i}\mu}\forall\underline{i},\mu \in$ AO-batch;
            end for
            for all $\tau$ do
                for i ∈ rank_occ do
                    $N^{QP}_{\tau} += B^{Q}_{\mu\underline{i}}\,{}^{\tau}\hat{B}^{P}_{\mu\underline{i}}$ $\forall\mu \in$ AO-batch and $P,Q \in$ aux-batch-1/-2
                end for
            end for
        end for
    end for
end for
$\tilde{N}^{QP}_{\tau} = J^{QR}_{\omega}N^{RS}_{\tau}J^{SP}_{\omega}$ $\forall\tau$
$\tilde{n}^{P}_{\tau} = n^{R}_{\tau}J^{RP}_{\omega}$ $\forall\tau$
```

FIG. 2. Algorithm for the calculation of $\tilde{N}^{QP}_{\tau}$ and $\tilde{n}^{P}_{\tau}$ intermediates within the ω-SOS-CDD-RI-CC2 implementation.

in Fig. 2 by reading $\hat{B}^P_{\mu\underline{i}}$ (line 12) and $B^Q_{\underline{i}\mu}$ (line 24) by batches of auxiliary and basis functions indices, at the cost of a batching overhead proportional to the number of auxiliary and basis functions batches $\mathrm{b_{aux}}$ and $\mathrm{b_{AO}}$, respectively. In the optimal batching, $\mathrm{b_{aux}}$ and $\mathrm{b_{AO}}$ are equal. Once the intermediates are formed, we multiply them with $\mathbf{J}_{\omega}$.

- The intermediates $\hat{Y}^Q_{\mu\underline{i}}$ and $\Omega^I_{\mu\underline{i}}$ are computed by batching over auxiliary and occupied indices. The three-center integrals $\hat{B}^P_{\mu\underline{j}}$ are read with an overhead proportional to the number of occupied batches $\mathrm{b_{occ}}$, while $\hat{Y}^Q_{\mu\underline{i}}$ is read $\mathrm{b_{aux}}$ times (see Fig. 3). Notice that $\mathrm{b_{aux}}$ is generally smaller than $\mathrm{b_{occ}}$ in order to minimize the I/O effort. In addition, lines 1, 12, and 13 are optional and limited to cases with minor sparsity in the three-center integrals matrices. At the end, the contribution to the vector function is scaled by $-\mathrm{c_{os}}$.
- The G and H contributions to the vector function are computed in batches of auxiliary indices. In this case, the three-center integrals $B^Q_{\underline{k}\underline{i}}$ and $B^Q_{\mu\nu}$ are not read redundantly, as reported in Fig. 4. At the end, the contributions to the vector function are scaled by $-\mathrm{c_{os}}$.

```
1:  for aux-batch-1 do
2:      for occ-batch do
3:          for all P ∈ aux-batch-1 do
4:              read $\hat{B}^P_{\mu'\underline{j}}$ $\forall\mu',\underline{j}$
5:              for all $\tau$ do
6:                  ${}^{\tau}\hat{B}^P_{\mu\underline{i}} = \hat{Q}^{\tau}_{\mu\mu'}\hat{B}^P_{\mu'\underline{j}}(L_{\sigma\underline{j}}S_{\sigma\lambda}P^{\tau}_{\lambda\underline{i}})\forall\mu,\underline{i}\in$ occ-batch;
7:              end for
8:          end for
9:          for all $\tau$ do
10:             $\Omega^I_{\mu\underline{i}} += {}^{\tau}\hat{B}^P_{\mu\underline{i}}n^P_{\tau}\forall\mu,\underline{i}\in$ occ-batch, $\forall P\in$ aux-batch-1
11:         end for
12:         for aux-batch-2 do
13:             read $\hat{Y}^Q_{\mu\underline{i}}$ $\forall Q\in$ aux-batch-2, $\forall\underline{i}\in$ occ-batch
14:             for all $\tau$ do
15:                 for all $\underline{i}\in$ occ-batch do
16:                     $\hat{Y}^Q_{\mu\underline{i}} += {}^{\tau}\hat{B}^P_{\mu\underline{i}}\tilde{N}^{QP}_{\tau}\forall P,Q\in$ aux-batch-1 and -2
17:                 end for
18:             end for
19:             write $\hat{Y}^Q_{\mu\underline{i}}$ $\forall Q\in$ aux-batch-2, $\forall\mu,\underline{i}\in$ occ-batch
20:         end for
21:     end for
22: end for
23: scale $\Omega^I_{\mu\underline{i}}$ by $-c_{os}$
```

FIG. 3. Algorithm for the calculation of $\hat{Y}^Q_{\mu\underline{i}}$ and $\Omega^I_{\mu\underline{i}}$ intermediates within the ω-SOS-CDD-RI-CC2 implementation.

```
1:  for aux-batch do
2:      for Q ∈ aux-batch do
3:          read $\hat{B}^Q_{\underline{k}\underline{i}}$ and $\hat{Y}^Q_{\mu\underline{k}}$ $\forall\underline{k},\underline{i},\mu$
4:      end for
5:      for Q ∈ aux-batch do
6:          $\Omega^H_{\mu\underline{i}} += \hat{Y}^Q_{\mu\underline{k}}\hat{B}^Q_{\underline{k}\underline{i}}\forall\mu,\underline{i}$
7:      end for
8:      for Q ∈ aux-batch do
9:          read $B^Q_{\mu\nu}\forall\mu,\nu$
10:         $\bar{\Omega}^G_{\mu\underline{i}} += B^Q_{\mu\nu}\hat{Y}^Q_{\nu\underline{i}}\forall\mu,\underline{i}$
11:     end for
12: end for
13: $\Omega^G_{\mu\underline{i}} = \hat{Q}_{\mu\mu'}\bar{\Omega}^G_{\mu'\underline{i}}$
14: scale $\Omega^G_{\mu\underline{i}}$ and $\Omega^H_{\mu\underline{i}}$ by $-c_{os}$
```

FIG. 4. Algorithm for the calculation of $\Omega^G_{\mu\underline{i}}$ and $\Omega^H_{\mu\underline{i}}$ intermediates within the ω-SOS-CDD-RI-CC2 implementation.

As can be seen in Figs. 2 and 3, the minimal overhead is obtained if there is only one τ-batch containing all Laplace quadrature points.[58] Finally, in order to increase the efficiency, the three-center integrals are read and simultaneously transformed in parallel using all the available threads (i.e., line 11 in Fig. 2 and line 3 in Fig. 3).

III. COMPUTATIONAL DETAILS

Our ω-SOS-CDD-RI-CC2 method as well as the MO-SOS-RI-CC2 equations by Winter and Hättig[30] were implemented in the FermiONs++ program.[51–53] We checked our MO-SOS-RI-CC2 implementation against the implementation in Turbomole7.3[59] to verify comparable performance and accuracy (error in energy in the range of 10^{-4}–10^{-5} a.u. and very similar computational times). The underlying Hartree–Fock calculations have been converged to a maximum element of the error matrix in the direct inversion in the iterative subspace (DIIS) procedure below 10^{-7}. We employed the RI-approximated integrals by Kussmann *et al.*[60] for the evaluation of the Coulomb and the $\mathcal{O}(N)$ semi-numerical sn-LinK method by Laqua *et al.*[61,62] to compute the exchange integrals of the Fock matrix.

Furthermore, our CC2 model does not make use of any explicit integral screening. The reduction of the scaling and the consequent performance improvement are based on the use of efficient sparse matrix algebra in the steps involving the three-center integrals. The present implementation exploits block-sparse (BS) matrices, which divide the matrices in smaller blocks, whose maximum size is 96 × 96. The screening is twofold: First, we employ a sparsity criterion (ϑ_a) that screens the matrices upon allocation with a default threshold $\vartheta_a = 10^{-7}$. This means that every block with a

L2-norm lower than ϑ_a is discarded. Consequently, both memory and disk space requirements are reduced. Second, the threshold ϑ_m is used to improve the performance of matrix–matrix multiplications. In fact, if the product of the L2-norms of two multiplied matrix-blocks is lower than a given threshold, that multiplication step is not performed. The default value is $\vartheta_m = 10^{-9}$. Additional information about our BS matrices and the algorithm for the matrix–matrix multiplication are provided in the supplementary material.

The optimization of the singles cluster amplitudes is carried out via the DIIS procedure, which terminates when the L2-norm of the singles vector function is lower than 10^{-5}. As atomic basis sets, the def2-SVP and the def2-TZVP basis sets[63,64] are employed. For the resolution of identity used to approximate the four-center integrals, the corresponding auxiliary basis sets[9,45,65] are used. If nothing else is indicated, we use optimized minimax grids with seven quadrature points for the Laplace expansion. In addition, we set $\omega = 0.1$ a.u. for the attenuated RI-metric, which has been found sufficient for the metric[43] to start being local while there is no significant loss in accuracy. All calculations are performed using multi-core computing nodes and an OpenMP parallelized code. We used a computing node with one Dual AMD EPYC 7302 32-Core 3.0 GHz CPUs, 1 TB of RAM, and 5.5 TB of disk space. All runtimes given are wall times, not CPU times.

IV. RESULTS

A. Accuracy

We performed benchmark calculations on the S22 and L7 test sets of complexes.[66,67] The interaction energies obtained with the ω-SOS-CDD-RI-CC2 method are compared to a MO-SOS-RI-CC2 reference and the errors are summarized in Table III. The use of sparse algebra yields results as accurate as standard dense algebra for both test sets since the sparsity of the density matrices associated with these systems is low. The error introduced by the attenuation factor is negligible for the S22 set, with MAE and MAX equal to 0.002 and 0.007 kcal mol^{-1}, respectively, and slightly increases for the larger systems in the L7 set. In fact, the error for L7 samples is 0.02 and 0.03 kcal mol^{-1} in MAE and RMSD, respectively, and 0.06 kcal mol^{-1} in MAX. For the C3A, C3GC (and the monomers A, GC, and C3) systems, using the RI-approximation to compute the Coulomb contribution to the Fock matrix caused numerical instability during both HF and CC2 iterative procedures. Specifically, although HF

TABLE III. Root mean square deviation (RMSD), mean absolute error (MAE), and maximum error (MAX) for interaction energies computed with ω-SOS-CDD-RI-CC2 in the def2-TZVP basis compared to MO-SOS-RI-CC2 results for the S22[66] and L7[67] test sets with different RI-attenuation factors ω, and both sparse/dense linear algebra.

	$\vartheta_a = 10^{-7}, \vartheta_m = 10^{-9}$		$\vartheta_a = 0.0, \vartheta_m = 0.0$	
	$\omega = 0.0$	$\omega = 0.1$	$\omega = 0.0$	$\omega = 0.1$
		S22		
RMSD (μH)	0.3	5.1	0.3	5.0
MAE (μH)	0.3	3.9	0.3	3.8
MAX (μH)	0.8	11.3	0.8	11.3
		L7		
RMSD (μH)	0.7	49.8	0.6	50.0
MAE (μH)	0.5	36.9	0.5	37.0
MAX (μH)	1.5	96.0	1.2	96.3

TABLE IV. Absolute energy error of ω-SOS-CDD-RI-CC2 with respect to MO-SOS-RI-CC2 results for different attenuation factors ω, as well as for sparse and dense linear algebra. The number of basis functions is given for each system. The structure files of the selected systems are available for download from our website.[68]

		Error (μH)		Error (μH)	
	No. of bf	$\vartheta_a = 10^{-7}, \vartheta_m = 10^{-9}$		$\vartheta_a = 0.0, \vartheta_m = 0.0$	
Sample	def2-TZVP	$\omega = 0.0$	$\omega = 0.1$	$\omega = 0.0$	$\omega = 0.1$
$C_{40}H_{82}$	1732	0.02	23.00	0.03	7.90
$C_{80}H_{162}$	3452	0.50	48.20	0.50	15.30
$C_{100}H_{202}$	4312	0.71	62.60	0.74	18.42
$C_{160}H_{322}$	6892	1.06	108.00	1.36	28.90
AT01	1247	0.57	7.25	0.60	7.30
AT02	2680	0.80	17.50	0.70	16.90
Diamond 102	1662	0.004	28.00	0.03	28.01
Water 68	2924	0.14	19.30	0.14	19.30
Angiotensin	2751	0.05	31.60	0.10	30.50
Angiotensin deprotonated	2739	0.40	51.50	0.50	50.10
Angiotensin zwitterion	2751	0.50	40.00	0.50	39.00
K2 vitamine (90°)	1263	0.60	0.50	0.57	0.56
K2 vitamine (180°)	1263	0.50	1.40	0.50	1.50
CNT (C_{20})	680	1.19	3.77	1.21	3.78

calculations converged using tighter thresholds and larger auxiliary basis set for the three-center integrals, the CC2 calculations did not reach convergence for monomers or dimers if the RI-approximation was used in Eq. (35). Moreover, even using the RI-approximation only for HF calculations, and so avoiding it in Eq. (35), the CC2 calculations did not converge for monomers or dimers.

Therefore, for the mentioned systems, we did not employ the RI-approximation during neither HF nor CC2, since the computation of the Coulomb term with J-engine is not a computational bottleneck regardless. In addition, we used ten Laplace quadrature points for the C3 monomer for the same reason. In order to further investigate the behavior of ω-SOS-CDD-RI-CC2, we computed the absolute energies for random large systems from our own benchmark set[68] (see Table IV). In contrast to the observations for the S22 and the L7 test sets, the use of sparse linear algebra slightly decreases the accuracy of the obtained results. Thus, the accuracy of ω-SOS-CDD-RI-CC2 with both dense and sparse algebra is compared in order to display the influence of the sparse linear algebra screening and the RI-metric attenuation factor ω.

Again, it is clear from Table IV that the smaller systems are not affected by sparse algebra and yield results similar to the dense algebra implementation, as for the L7 and S22 sets. Moreover, the combined use of sparse algebra and the moderately attenuated Coulomb metric ($\omega = 0.1$) reduces the accuracy by a maximum of 0.1 kcal/mol. Therefore, the default screening thresholds ensure accurate results with both metrics; hence, they will be used in all further calculations. Of course, larger molecules are more sensitive to the choice of the RI-metric and are, in general, more sparse. For instance, the error introduced by either sparse linear algebra or the attenuation factor increases with the system size for the linear alkanes. Nonetheless, the accuracy of ω-SOS-CDD-RI-CC2 is under complete control using the two screening parameters ϑ_a and ϑ_m, and the attenuation factor ω.

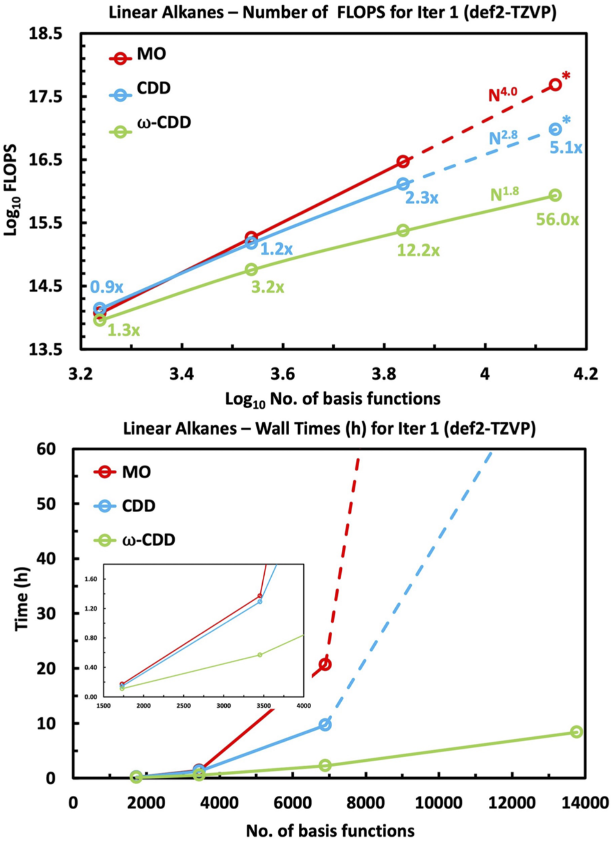


FIG. 5. Plots of FLOPS (top) and wall times (bottom) against number of basis functions showing the computational complexity of the MO and ω-SOS-CDD-RI-CC2 formulations (ω = 0: "CDD" and ω = 0.1: "ω-CDD") for linear alkanes in the def2-TZVP basis. In the top plot, speedups with respect to MO calculations are given. We used ϑ_a and ϑ_m equal to 10^{-7} and 10^{-9}, respectively. Top: Log-log plot. Bottom: Linear plot. The dashed lines and asterisks indicate that the points have been extrapolated.

B. Scaling behavior: Linear alkanes

The sparsity of the one-electron densities is closely related to the HOMO–LUMO gap of molecular systems.[69] Especially, the asymptotic linear scaling behavior holds only for systems with a nonvanishing HOMO–LUMO gap.[69,70] Accordingly, we investigated the computational and storage scaling of ω-SOS-CDD-RI-CC2 with $\omega = 0$ and $\omega = 0.1$ (with $\frac{1}{r_{12}}$ and $\frac{\mathrm{erfc}(\omega r_{12})}{r_{12}}$ operators for the RI-approximation, respectively) on electronically local systems such as linear alkanes. Of course, they represent optimal systems for calculations in a local basis (as shown by Fig. 1), but the same behavior is transferred to three-dimensional systems that are large enough. We carried out the analysis of the computational scaling, taking into account the number of floating-point operations (FLOPS) during the first iteration (with $t_{ai} \neq 0$). The results obtained using the def2-TZVP basis set are summarized in Fig. 5. As can be seen, the scaling exponents meet the expectations for both standard and ω-Coulomb ($\omega = 0.1$) metrics with values equal to ~2.8 and ~1.8 for the largest system. In the asymptotic limit, the computational scaling is cubic for $\omega = 0$. With a moderate attenuation factor ($\omega = 0.1$), the asymptotic scaling of almost all steps is reduced to linear. However, the number of FLOPS scales sub-quadratically $\mathcal{O}(N^{1.9})$ for the time-determining calculation of the $\hat{Y}^{Q}_{\underline{\mu} i}$ intermediate [Eq. (23)]. This is due to the multiplications of N^{QP}_{τ} matrices with $\mathbf{J}_{\omega}$ [Eq. (25)], which is not sparse. Notice that there are other steps with cubic or quadratic scaling in the algorithm (see Table II). Among these, the quadratic calculation of all $B_{\mu\nu}$ integrals is performed only once at the beginning and does not affect the overall efficiency (see Fig. 6). Some cubic and quadratic steps are repeated in each iteration (i.e., the formation of the Fock matrix and pseudo-densities). Nonetheless, it can be seen in Fig. 6 that these nonlinear scaling steps do not affect the overall efficiency of our method because the time demands are negligible if compared to steady times of one iteration or the entire CC2 calculation. The same behavior appears in the disk space requirements that are summarized in Table V. Since the number of basis functions is close to the number of virtual orbitals, the disk space demands of ω-SOS-CDD-RI-CC2 are formally similar to MO-SOS-RI-CC2 for smaller systems. Of course, as soon as the systems become large enough, the use of sparse matrices reduces the storage requirements along with the CPU costs. Indeed, the space needed for the matrices of the half-transformed

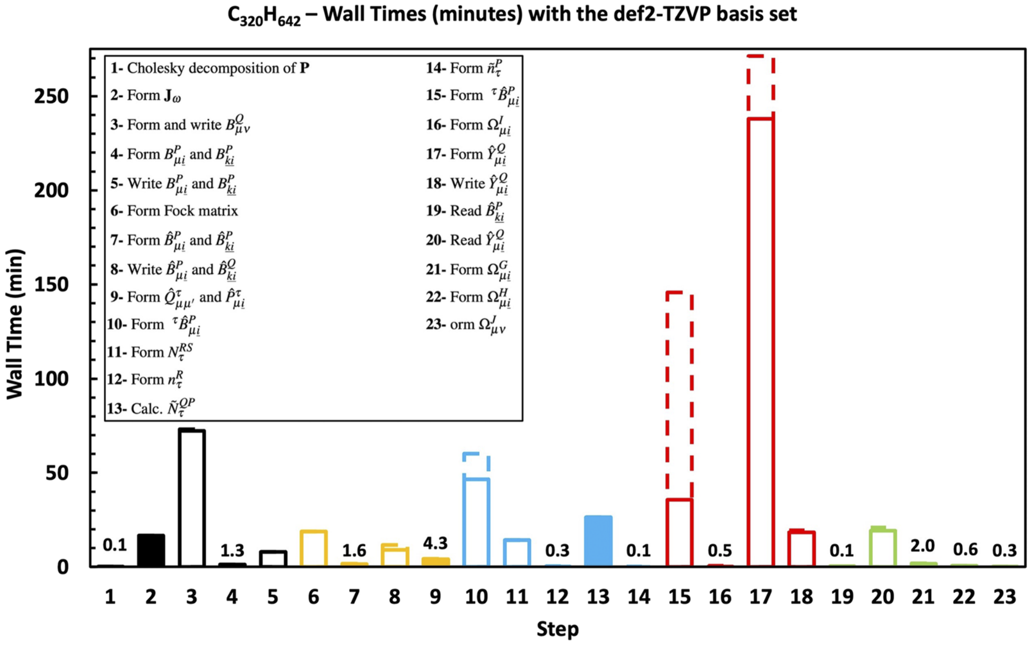


FIG. 6. Plot of wall times (minutes) for each step of ω-SOS-CDD-RI-CC2 $(\omega = 0.1)$ for the linear alkane $C_{320}H_{642}$. Fully colored rectangles indicate the use of dense matrices for that specific step. In black (1–5): Steps performed only once at the beginning. In orange (6–9): Steps performed in each iteration as preparation of the following time-determining calculations. In light blue (10–14): Time-determining steps involved in Fig. 2 where Step 10 refers to lines 11–16. In red (15–18): Time-determining steps involved in Fig. 3 where Step 15 refers to lines 3–8. In green (19–23): Steps involved in Fig. 4. The dashed lines indicate the timings obtained when we reduce the available memory from ∼900 to ∼450 GB.

TABLE V. Disk space demands (GB) for MO-SOS-RI-CC2 ("MO") and ω-SOS-CDD-RI-CC2 with $\omega = 0$ ("CDD") and $\omega = 0.1$ ("ω-CDD"). We do not store the AO three-center integrals in the MO implementation. The $\ast$ highlights when sparse matrices are not used to store them on disk. The sparsity thresholds ϑ_a and ϑ_m are equal to 10^{-7} and 10^{-9}, respectively.

	No. of bf	$B^P_{\mu\nu}$ (GB)		$\hat{B}^P_{ai}/\hat{B}^P_{\mu\underline{i}}$ (GB)			$\hat{Y}^Q_{ai}/\hat{Y}^Q_{\mu\underline{i}}$ (GB)		
Sample	def2-SVP	CDD	ω-CDD	MO	CDD	ω-CDD	MO	CDD	ω-CDD
$C_{40}H_{82}$	970	2.5	*2.5	2.3	2.6	1.7	2.2	2.7	2.7
$C_{80}H_{162}$	1930	10.0	7.2	17.6	16.0	6.3	17.6	17.5	17.2
$C_{160}C_{322}$	3850	40.2	15.0	140.2	68.5	15.8	140.2	80.9	69.2

	No. of bf	$B^P_{\mu\nu}$ (GB)		$\hat{B}^P_{ai}/\hat{B}^P_{\mu\underline{i}}$ (GB)			$\hat{Y}^Q_{ai}/\hat{Y}^Q_{\mu\underline{i}}$ (GB)		
Sample	def2-TZVP	CDD	ω-CDD	MO	CDD	ω-CDD	MO	CDD	ω-CDD
$C_{40}H_{82}$	1732	10.6	*10.6	6.1	6.4	4.0	6.1	6.7	6.7
$C_{80}H_{162}$	3452	44.5	25.0	47.8	42.7	16.8	47.8	52.7	52.7
$C_{160}C_{322}$	6892	176.3	52.0	380.6	220.0	54.4	380.6	378.0	338.3

three-center integrals ($\hat{B}^P_{\mu\underline{i}}$) is significantly decreased and scales linearly in the asymptotic limit (with $\omega = 0.1$). On the other hand, the disk space requirements for the $\hat{Y}^Q_{\mu\underline{i}}$ matrices are only reduced for the largest system because of their decreased sparsity and scale as $\mathcal{O}(N^2)$. The symmetric AO three-center integrals are treated differently. For MO-SOS-RI-CC2, these integrals are computed and transformed in each iteration. Thus, they are not stored on disk. For ω-SOS-CDD-RI-CC2 with $\omega = 0.0$, we store only the upper triangles that require $\frac{1}{2}\mathrm{N}_{\mathrm{aux}}\mathrm{N}_{\mathrm{basis}}(\mathrm{N}_{\mathrm{basis}} + 1)$ of disk space. On the other hand, for $\omega = 0.1$, we store the significant blocks within the upper triangle of the BS matrices, showing an asymptotic linear scaling $\mathcal{O}(N)$. The integral-direct transformation of the AO three-center integrals is possible also for ω-SOS-CDD-RI-CC2, avoiding the storage of this quantity.

C. Timings

1. Linear alkanes

The use of sparse linear algebra reduces the number of FLOPS carried out in one iteration and considerable runtime speedups over MO-SOS-RI-CC2 are expected when our ω-SOS-CDD-RI-CC2 is used. We performed the calculations in the def2-TZVP basis and the results are summarized in Table VI and Fig. 5. Moreover, all calculations were performed with the same number of batches. The runtime speedups are always smaller than the speedups obtained when comparing FLOPS, due to a runtime overhead of ~1.6 associated with the use of our block-matrices. Nevertheless, the crossover with the MO implementation is at ~50 carbon atoms, as one can see in Fig. 5. Indeed, our ω-SOS-CDD-RI-CC2 method ($\omega = 0.1$) is already twice as fast for $C_{80}H_{162}$. Doubling the size, our implementation is ~9 times faster than the MO formulation. Whether or not the runtime speedups meet the trend in the FLOPS speedups critically depends on the number of batches (b_{ao} and b_{occ}) in Figs. 2 and 3. In fact, although efficient, in general, the formation of the Laplace point-dependent three-center integrals (line 12, Fig. 2, and line 3, Fig. 3) in ω-SOS-CDD-RI-CC2 is negatively affected by the batching overhead. However, since $\hat{B}^{P}_{\underline{\mu}j}$ is sparse, this downside is mitigated until b_{ao} and b_{occ} get large, i.e., when the memory requirements exceed the available memory of the computing node by several times. The dashed lines in Fig. 6 display such a behavior for the linear alkane $C_{320}H_{642}$, where the computation time for Step 15 is quadrupled when the available memory is halved. The AO reformulation is ~37 times faster with $\omega = 0.1$ (see Table VI) and it outperforms the MO implementation even if the number of batches is increased. Furthermore, we want to stress that one can decrease the batching overhead (and increase the runtime speedups) either by using a computing node with a larger memory or by performing CC2 calculations on multiple nodes, distributing both CPU and I/O efforts. However, we did not exploit the second solution in the present paper.

TABLE VI. Wall times (h) for the first iteration ($t_{ai} \neq 0$) of MO and ω-SOS-CDD-RI-CC2 formulations (ω = 0: "CDD" and ω = 0.1: "ω-CDD") for linear alkanes in the def2-TZVP basis. We employ sparsity thresholds ϑ_a and ϑ_m equal to 10^{-7} and 10^{-9}, respectively. Values marked with an asterisk (*) are extrapolated conservatively.

	No. of bf	MO	CDD		ω-CDD	
Sample	def2-TZVP	Time (h)	Time (h)	Speedup	Time (h)	Speedup
$C_{40}H_{82}$	1 732	0.17	0.15	×1.2	0.11	×1.6
$C_{80}H_{162}$	3 452	1.37	1.29	×1.1	0.57	×2.2
$C_{160}C_{322}$	6 892	20.63	9.62	×2.1	2.27	×9.1
$C_{320}H_{642}$	13 772	*309.00	*85.00	×3.6	8.38	×36.9

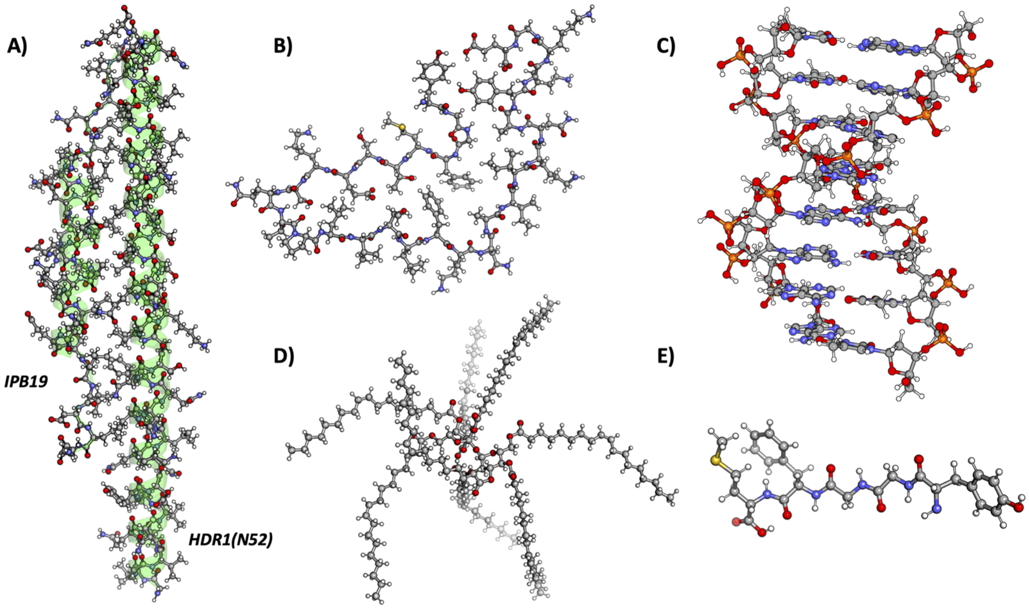


FIG. 7. Test set for the performance of the ω-SOS-CDD-RI-CC2 model (ω = 0.1): (a) IPB19/N52 complex[71] (its alpha-helix structure is highlighted in green), (b) beta-endorphin,[72] (c) AT08 pairs,[68] (d) olestra,[73] (e) metenkephalin.[34] We employed MolProbity[74] with default settings in order to add the missing hydrogens to the IPB19/N52 complex.

TABLE VII. Runtime speedups for the first iteration of ω-SOS-CDD-RI-CC2 ($\omega = 0.1$) against MO-SOS-RI-CC2 in both def2-SVP and def2-TZVP bases. We employ sparsity thresholds ϑ_a and ϑ_m equal to 10^{-7} and 10^{-9}, respectively.

Sample	No. of atoms	No. of bf def2-SVP	MO-SOS-RI-CC2 Time (h)	ω-SOS-CDD-RI-CC2 Time (h)	 Speed up
HDR1(N52)	746	7069	66.76	11.93	×5.6
DNA8	524	5574	29.25	18.42	×1.6
Beta-endorphin	495	4675	14.07	6.55	×2.1
Olestra	453	3840	5.27	3.94	×1.3
IPB19	441	4149	8.77	3.48	×2.5
Metenkephalin	75	739	0.02	0.03	×0.7

Sample	No. of atoms	No. of bf def2-TZVP	MO-SOS-RI-CC2 Time (h)	ω-SOS-CDD-RI-CC2 Time (h)	 Speed up
Beta-endorphin	495	9076	61.32	46.24	×1.3
Olestra	453	7093	23.68	18.45	×1.3
IPB19	441	8046	40.27	26.04	×1.5
Metenkephalin	75	1456	0.08	0.09	×0.8

2. Three-dimensional systems

The ω-SOS-CDD-RI-CC2 method has proven to be efficient if applied to optimal systems such as linear alkanes. In fact, the introduction of the Cholesky-decomposed density matrices resulted in an early crossover with the MO-SOS-RI-CC2 model, as shown in Fig. 5. A reduction of the number of FLOPS and a consequent runtime speedup are also obtained when ω-SOS-CDD-RI-CC2 is applied to real-life organic systems, which do not always display sparse density matrices. We used as test set six different systems illustrated in Fig. 7, but it was not possible to perform calculations in the def2-TZVP basis for some systems due to disk space limitations. Table VII shows the wall times and speedups (with $\omega = 0.1$) for the first iteration in both def2-SVP and def2-TZVP basis sets. For HDR1(N52) in the def2-SVP basis, our ω-SOS-CDD-RI-CC2 provides a speedup of 5.6, so that one iteration is carried out in ~12 h instead of ~67 h. The speedups for the def2-TZVP basis set are lower due to the use of more diffuse functions. For instance, the ω-SOS-CDD-RI-CC2 speedups for the beta-endorphin are 2.1 and 1.3 with def2-SVP and def2-TZVP basis, respectively. In general, the ω-SOS-CDD-RI-CC2 model provides speedups that are expected to become larger for increasing system sizes due to the reduced scaling. On the other hand, ω-SOS-CDD-RI-CC2 timings are comparable to the MO-based implementation for the smaller systems (e.g., metenkephalin).

V. SUMMARY

We presented a reformulation of the SOS-RI-CC2 method in the AO basis that shows cubic scaling in the asymptotic limit. We further employed an attenuated Coulomb metric for the RI-approximation decreasing the scaling to sub-quadratic for $\omega = 0.1$. The Cholesky decomposition of the ground-state occupied density matrix provides local occupied molecular orbitals that allow for the reduction of the basis set scaling. Moreover, it leads to a reduced prefactor and an early crossover with the MO implementation. Our memory-efficient ω-SOS-CDD-RI-CC2 method is based on a minimal-overhead batching scheme and on efficient sparse linear algebra routines, providing complete control of the error via the ϑ_a and ϑ_m thresholds while significantly speeding up the calculations. Such control results in small errors as shown for calculations on the S22 and L7 test sets and a selection of systems from our own benchmark.[68] The performance of our reformulation has been assessed for both SVP and TZVP basis sets with three-dimensional systems of up to 700 atoms. The timings show that our method provides considerable advantages if there is enough sparsity to be exploited. On the other hand, ω-SOS-CDD-RI-CC2 timings are comparable to MO-SOS-RI-CC2 wall times for smaller systems, whose density matrices are not sparse.

The disk space demand will be further reduced by implementing an integral-direct algorithm, which avoids the storage of some three-dimensional tensors and will be subject of a future publication. In addition, increasing the runtime speedups may be possible by distributing the computational and I/O efforts among multiple nodes. Finally, we want to stress that the presented ω-SOS-CDD-RI-CC2 approach provides the basis for the AO reformulation of the SOS-RI-CC2 and ADC(2) equations for excited states energies. An implementation is currently in progress in our group and will be the subject of a forthcoming publication.

SUPPLEMENTARY MATERIAL

See the supplementary material for details about the reformulation of cluster equations in the AO basis, the operation of our block-sparse matrices, the scaling behavior, and our implementation of MO-SOS-RI-CC2.

ACKNOWLEDGMENTS

This project received funding from the European Union's Horizon 2020 research and innovation program under Marie Skłodowska-Curie Grant Agreement No 765739. C.O. further acknowledges financial support from the Max-Planck Fellowship Program at the MPI-FKF Stuttgart.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

F. Sacchetta: Investigation (equal); Methodology (equal); Writing – original draft (equal). **D. Graf**: Conceptualization (equal); Investigation (equal); Writing – original draft (equal). **H. Laqua**: Conceptualization (equal); Methodology (equal); Writing – review & editing (equal). **M. A. Ambroise**: Conceptualization (equal); Investigation (equal). **J. Kussmann**: Conceptualization (equal); Investigation (equal); Methodology (equal); Writing – review & editing (equal). **A. Dreuw**: Conceptualization (equal); Supervision (equal); Writing – review & editing (equal). **C. Ochsenfeld**: Conceptualization (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material.

REFERENCES

[1] J. Čížek and J. Paldus, "Correlation problems in atomic and molecular systems III. Rederivation of the coupled-pair many-electron theory using the traditional quantum chemical methods," Int. J. Quantum Chem. **5**, 359–379 (1971).

[2] J. Čížek, "On the use of the cluster expansion and the technique of diagrams in calculations of correlation effects in atoms and molecules," Adv. Chem. Phys. **14** 35–89 (1969).

[3] R. J. Bartlett, "The coupled-cluster revolution," Mol. Phys. **108**, 2905–2920 (2010).

[4] T. D. Crawford and H. F. Schaefer, "An introduction to coupled cluster theory for computational chemists," Rev. Comput. Chem. **14**, 33–136 (2000).

[5] G. D. Purvis III and R. J. Bartlett, "A full coupled-cluster singles and doubles model: The inclusion of disconnected triples," J. Chem. Phys. **76**, 1910–1918 (1982).

[6] H. Koch, A. Sánchez de Merás, T. Helgaker, and O. Christiansen, "The integral-direct coupled cluster singles and doubles model," J. Chem. Phys. **104**, 4157–4165 (1996).

[7] O. Christiansen, H. Koch, and P. Jørgensen, "The second-order approximate coupled cluster singles and doubles model CC2," Chem. Phys. Lett. **243**, 409–418 (1995).

[8] Y. Jung, R. C. Lochan, A. D. Dutoi, and M. Head-Gordon, "Scaled opposite-spin second order Møller-Plesset correlation energy: An economical electronic structure method," J. Chem. Phys. **121**, 9793–9802 (2004).

[9] Y. Jung, Y. Shao, and M. Head-Gordon, "Fast evaluation of scaled opposite spin second-order Møller-Plesset correlation energies using auxiliary basis expansions and exploiting sparsity," J. Comput. Chem. **28**, 1953–1964 (2007).

[10] P. Pulay, "Localizability of dynamic electron correlation," Chem. Phys. Lett. **100**, 151–154 (1983).

[11] P. Pulay and S. Saebø, "Orbital-invariant formulation and second-order gradient evaluation in Møller-Plesset perturbation theory," Theor. Chim. Acta **69**, 357–368 (1986).

[12] S. Saebø and P. Pulay, "A low-scaling method for second order Møller-Plesset calculations," J. Chem. Phys. **115**, 3975–3983 (2001).

[13] H.-J. Werner, F. R. Manby, and P. J. Knowles, "Fast linear scaling second-order Møller-Plesset perturbation theory (MP2) using local and density fitting approximations," J. Chem. Phys. **118**, 8149–8160 (2003).

[14] S. A. Maurer, L. Clin, and C. Ochsenfeld, "Cholesky-decomposed density MP2 with density fitting: Accurate MP2 and double-hybrid DFT energies for large systems," J. Chem. Phys. **140**, 224112 (2014).

[15] P. Y. Ayala and G. E. Scuseria, "Linear scaling second-order Moller-Plesset theory in the atomic orbital basis for large molecular systems," J. Chem. Phys. **110**, 3660–3671 (1999).

[16] S. A. Maurer, D. S. Lambrecht, J. Kussmann, and C. Ochsenfeld, "Efficient distance-including integral screening in linear-scaling Moller-Plesset perturbation theory," J. Chem. Phys. **138**, 014101 (2013).

[17] S. Schweizer, B. Doser, and C. Ochsenfeld, "An atomic orbital-based reformulation of energy gradients in second-order Møller-Plesset perturbation theory," J. Chem. Phys. **128**, 154101 (2008).

[18] M. Glasbrenner, D. Graf, and C. Ochsenfeld, "Efficient reduced-scaling second-order Møller-Plesset perturbation theory with Cholesky-decomposed densities and an attenuated Coulomb metric," J. Chem. Theory Comput. **16**, 6856–6868 (2020).

[19] R. A. Kendall and H. A. Früchtl, "The impact of the resolution of the identity approximate integral method on modern *ab initio* algorithm development," Theor. Chem. Acc. **97**, 158–163 (1997).

[20] M. Feyereisen, G. Fitzgerald, and A. Komornicki, "Use of approximate integrals in *ab initio* theory. An application in MP2 energy calculations," Chem. Phys. Lett. **208**, 359–363 (1993).

[21] C. Hättig and F. Weigend, "CC2 excitation energy calculations on large molecules using the resolution of the identity approximation," J. Chem. Phys. **113**, 5154–5161 (2000).

[22] H. Koch, A. Sánchez de Merás, and T. B. Pedersen, "Reduced scaling in electronic structure calculations using Cholesky decompositions," J. Chem. Phys. **118**, 9481–9484 (2003).

[23] J. Boström, M. Pitoňák, F. Aquilante, P. Neogrády, T. B. Pedersen, and R. Lindh, "Coupled cluster and Møller–Plesset perturbation theory calculations of noncovalent intermolecular interactions using density fitting with auxiliary basis sets from Cholesky decompositions," J. Chem. Theory Comput. **8**, 1921–1928 (2012).

[24] P. Baudin, J. S. Marín, I. G. Cuesta, and A. M. J. Sánchez de Merás, "Calculation of excitation energies from the CC2 linear response theory using Cholesky decomposition," J. Chem. Phys. **140**, 104111 (2014).

[25] S. D. Folkestad, E. F. Kjønstad, L. Goletto, and H. Koch, "Multilevel CC2 and CCSD in reduced orbital spaces: Electronic excitations in large molecular systems," J. Chem. Theory Comput. **17**, 714–726 (2021).

[26] S. Grimme, "Improved second-order Møller-Plesset perturbation theory by separate scaling of parallel-and antiparallel-spin pair correlation energies," J. Chem. Phys. **118**, 9095–9102 (2003).

[27] M. Häser, "Møller-Plesset (MP2) perturbation theory for large molecules," Theor. Chim. Acta **87**, 147–173 (1993).

[28] J. Almlöf, "Elimination of energy denominators in Møller-Plesset perturbation theory by a Laplace transform approach," Chem. Phys. Lett. **181**, 319–320 (1991).

[29] M. Häser and J. Almlöf, "Laplace transform techniques in Møller-Plesset perturbation theory," J. Chem. Phys. **96**, 489–494 (1992).

[30] N. O. C. Winter and C. Hättig, "Scaled opposite-spin CC2 for ground and excited states with fourth order scaling computational costs," J. Chem. Phys. **134**, 184101 (2011).

[31] D. Kats, T. Korona, and M. Schütz, "Local CC2 electronic excitation energies for large molecules with density fitting," J. Chem. Phys. **125**, 104106 (2006).

[32] D. Kats and M. Schütz, "A multistate local coupled cluster CC2 response method based on the Laplace transform," J. Chem. Phys. **131**, 124117 (2009).

[33] K. Ledermüller and M. Schütz, "Local CC2 response method based on the Laplace transform: Analytic energy gradients for ground and excited states," J. Chem. Phys. **140**, 164113 (2014).

[34] P. Baudin and K. Kristensen, "LoFEx—A local framework for calculating excitation energies: Illustrations using RI-CC2 linear response theory," J. Chem. Phys. **144**, 224106 (2016).
[35] F. Neese, A. Hansen, and D. G. Liakos, "Efficient and accurate approximations to the local coupled cluster singles doubles method using a truncated pair natural orbital basis," J. Chem. Phys. **131**, 064103 (2009).
[36] F. Neese, F. Wennmohs, and A. Hansen, "Efficient and accurate local approximations to coupled-electron pair approaches: An attempt to revive the pair natural orbital method," J. Chem. Phys. **130**, 114108 (2009).
[37] B. Helmich and C. Hättig, "A pair natural orbital implementation of the coupled cluster model CC2 for excitation energies," J. Chem. Phys. **139**, 084114 (2013).
[38] F. Weigend and M. Häser, "RI-MP2: First derivatives and global consistency," Theor. Chem. Acc. **97**, 331–340 (1997).
[39] R. C. Lochan, Y. Shao, and M. Head-Gordon, "Quartic-scaling analytical energy gradient of scaled opposite-spin second-order Møller- Plesset perturbation theory," J. Chem. Theory Comput. **3**, 988–1003 (2007).
[40] G. E. Scuseria and P. Y. Ayala, "Linear scaling coupled cluster and perturbation theories in the atomic orbital basis," J. Chem. Phys. **111**, 8330–8343 (1999).
[41] M. Beer and C. Ochsenfeld, "Efficient linear-scaling calculation of response properties: Density matrix-based Laplace-transformed coupled-perturbed self-consistent field theory," 2008.
[42] M. Beuerle, D. Graf, H. F. Schurkus, and C. Ochsenfeld, "Efficient calculation of beyond RPA correlation energies in the dielectric matrix formalism," J. Chem. Phys. **148**, 204104 (2018).
[43] A. Luenser, H. F. Schurkus, and C. Ochsenfeld, "Vanishing-overhead linear-scaling random phase approximation by Cholesky decomposition and an attenuated Coulomb-metric," J. Chem. Theory Comput. **13**, 1647–1655 (2017).
[44] D. Graf, M. Beuerle, and C. Ochsenfeld, "Low-scaling self-consistent minimization of a density matrix based random phase approximation method in the atomic orbital space," J. Chem. Theory Comput. **15**, 4468–4477 (2019).
[45] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon, "Auxiliary basis expansions for large-scale electronic structure calculations," Proc. Natl. Acad. Sci. U. S. A. **102**, 6692–6697 (2005).
[46] T. D. Crawford, A. Kumar, A. P. Bazanté, and R. Di Remigio, "Reduced-scaling coupled cluster response theory: Challenges and opportunities," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **9**, e1406 (2019).
[47] R. J. Bartlett and G. D. Purvis, "Many-body perturbation theory, coupled-pair many-electron theory, and the importance of quadruple excitations for the correlation problem," Int. J. Quantum Chem. **14**, 561–581 (1978).
[48] P. Pulay, "Improved SCF convergence acceleration," J. Comput. Chem. **3**, 556–560 (1982).
[49] A. Takatsuka, S. Ten-No, and W. Hackbusch, "Minimax approximation for the decomposition of energy denominators in Laplace-transformed Møller-Plesset perturbation theories," J. Chem. Phys. **129**, 044112 (2008).
[50] B. Helmich-Paris and L. Visscher, "Improvements on the minimax algorithm for the Laplace transformation of orbital energy denominators," J. Comput. Phys. **321**, 927–931 (2016).
[51] J. Kussmann and C. Ochsenfeld, "Pre-selective screening for matrix elements in linear-scaling exact exchange calculations," J. Chem. Phys. **138**, 134114 (2013).
[52] J. Kussmann and C. Ochsenfeld, "Preselective screening for linear-scaling exact exchange-gradient calculations for graphics processing units and general strong-scaling massively parallel calculations," J. Chem. Theory Comput. **11**, 918–922 (2015).
[53] J. Kussmann and C. Ochsenfeld, "Hybrid CPU/GPu integral engine for strong-scaling *ab initio* methods," J. Chem. Theory Comput. **13**, 3153–3159 (2017).
[54] N. J. Higham, "Cholesky factorization," Wiley Interdiscip. Rev.: Comput. Stat. **1**, 251–254 (2009).
[55] H. Harbrecht, M. Peters, and R. Schneider, "On the low-rank approximation by the pivoted Cholesky decomposition," Appl. Numer. Math. **62**, 428–440 (2012).
[56] S. Schweizer, J. Kussmann, B. Doser, and C. Ochsenfeld, "Linear-scaling Cholesky decomposition," J. Comput. Chem. **29**, 1004–1010 (2008).
[57] F. Aquilante, T. Bondo Pedersen, A. Sánchez de Merás, and H. Koch, "Fast noniterative orbital localization for large molecules," J. Chem. Phys. **125**, 174101 (2006).
[58] V. Drontschenko, D. Graf, H. Laqua, and C. Ochsenfeld, "Lagrangian-based minimal-overhead batching scheme for the efficient integral-direct evaluation of the RPA correlation energy," J. Chem. Theory Comput. **17**, 5623–5634 (2021).
[59] V. Turbomole, 7.3, TURBOMOLE GmbH 2018, TURBOMOLE is a development of University of Karlsruhe and Forschungszentrum Karlsruhe 2007, 1989.
[60] J. Kussmann, H. Laqua, and C. Ochsenfeld, "Highly efficient resolution-of-identity density functional theory calculations on central and graphics processing units," J. Chem. Theory Comput. **17**, 1512–1521 (2021).
[61] H. Laqua, T. H. Thompson, J. Kussmann, and C. Ochsenfeld, "Highly efficient, linear-scaling seminumerical exact-exchange method for graphic processing units," J. Chem. Theory Comput. **16**, 1456–1468 (2020).
[62] H. Laqua, J. Kussmann, and C. Ochsenfeld, "Accelerating seminumerical Fock-exchange calculations using mixed single-and double-precision arithmethic," J. Chem. Phys. **154**, 214116 (2021).
[63] F. Weigend, F. Furche, and R. Ahlrichs, "Gaussian basis sets of quadruple zeta valence quality for atoms H–Kr," J. Chem. Phys. **119**, 12753–12762 (2003).
[64] F. Weigend and R. Ahlrichs, "Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy," Phys. Chem. Chem. Phys. **7**, 3297–3305 (2005).
[65] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, "RI-MP2: Optimized auxiliary basis sets and demonstration of efficiency," Chem. Phys. Lett. **294**, 143–152 (1998).
[66] P. Jurečka, J. Šponer, J. Černý, and P. Hobza, "Benchmark database of accurate (MP2 and CCSD (T) complete basis set limit) interaction energies of small model complexes, DNA base pairs, and amino acid pairs," Phys. Chem. Chem. Phys. **8**, 1985–1993 (2006).
[67] R. Sedlak, T. Janowski, M. Pitoňák, J. Řezáč, P. Pulay, and P. Hobza, "Accuracy of quantum chemical methods for large noncovalent complexes," J. Chem. Theory Comput. **9**, 3364–3374 (2013).
[68] Structures available online from http://www.cup.lmu.de/pc/ochsenfeld/.
[69] P. E. Maslen, C. Ochsenfeld, C. A. White, M. S. Lee, and M. Head-Gordon, "Locality and sparsity of *ab initio* one-particle density matrices and localized orbitals," J. Phys. Chem. A **102**, 2215–2222 (1998).
[70] C. Ochsenfeld, J. Kussmann, and D. S. Lambrecht, "Linear-scaling methods in quantum chemistry," Rev. Comput. Chem. **23**, 1 (2007).
[71] D. Yu, Y. Zhu, T. Jiao, T. Wu, X. Xiao, B. Qin, Y. Hu, H. Chong, X. Lei, L. Ren *et al.*, "Structure-based design and characterization of novel fusion-inhibitory lipopeptides against SARS-CoV-2 and emerging variants," Emerging Microbes Infect. **10**, 1227–1240 (2021).
[72] C. Seuring, J. Verasdonck, J. Gath, D. Ghosh, N. Nespovitaya, M. A. Wälti, S. K. Maji, R. Cadalbert, P. Güntert, B. H. Meier, and R. Riek, "The three-dimensional structure of human β-endorphin amyloid fibrils," Nat. Struct. Mol. Biol. **27**, 1178–1184 (2020).
[73] Structures available online from http://www.petachem.com/products.html.
[74] C. J. Williams, J. J. Headd, N. W. Moriarty, M. G. Prisant, L. L. Videau, L. N. Deis, V. Verma, D. A. Keedy, B. J. Hintze, V. B. Chen *et al.*, "Molprobity: More and better reference data for improved all-atom structure validation," Protein Sci. **27**, 293–315 (2018).

An effective sub-quadratic scaling atomic-orbital reformulation of the scaled opposite-spin RI-CC2 ground-state model using Cholesky-decomposed densities and an attenuated Coulomb-metric (Supporting Information).

F. Sacchetta,[1] D. Graf,[1] H. Laqua,[1] M. A. Ambroise,[2] J. Kussmann,[1] A. Dreuw,[2] and C. Ochsenfeld[1, a)]

[1)] *Chair of Theoretical Chemistry, Department of Chemistry, University of Munich (LMU), Munich, Germany.*

[2)] *Chair of Theoretical and Computational Chemistry, Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany.*

(Dated: 25 July 2022)

[a)] Electronic mail: christian.ochsenfeld@cup.uni-muenchen.de

I. DERIVATION OF AO-BASED EXPRESSIONS FOR THE CONTRIBUTIONS TO THE SINGLES VECTOR FUNCTION.

In this section we provide the steps for reformulating the vector function terms of MO-SOS-RI-CC2 in the AO basis. The explicit expressions for the ground state densities can be found in the article.

$$\begin{aligned}
\Omega^G_{\mu\nu} &= \sum_{ai} C_{\mu a}\Omega^G_{ai}C_{\nu i} = -c_{os}\sum_{aci,Q} C_{\mu a}C_{\nu i}\hat{B}^Q_{ac}Y^Q_{ci} \\
&= -c_{os}\sum_{aci,Q}\sum_{\tau}^{n} w_\tau \sum_{P} N^{QP}_\tau \sum_{\substack{\mu'\nu'\\ \sigma\lambda}} C_{\mu a}\Lambda^p_{\mu' c}\Lambda^h_{\nu' i}B^P_{\mu'\nu'}\Lambda^p_{\sigma a}C_{\lambda c}B^Q_{\sigma\lambda}e^{-\varepsilon_{ci}t_\tau}C_{\nu i} \\
&= -c_{os}\sum_{\tau}^{n}\sum_{P}\sum_{\substack{\mu'\nu'\\ \sigma\lambda}} \hat{Q}^\tau_{\lambda\mu'}B^P_{\mu'\nu'}\hat{P}^\tau_{\nu'\nu}N^{QP}_\tau\hat{Q}_{\mu\sigma}B^Q_{\sigma\lambda} \\
&= \sum_{\lambda,Q}\hat{B}^Q_{\mu\lambda}Y^Q_{\lambda\nu}
\end{aligned}\tag{1}$$

$$\begin{aligned}
\Omega^H_{\mu\nu} &= \sum_{ai} C_{\mu a}\Omega^H_{ai}C_{\nu i} = -c_{os}\sum_{aik,Q} C_{\mu a}C_{\nu i}Y^Q_{ak}\hat{B}^Q_{ki} \\
&= -c_{os}\sum_{aik,Q}\sum_{\tau}^{n} w_\tau \sum_{P} N^{QP}_\tau \sum_{\substack{\mu'\nu'\\ \sigma\lambda}} C_{\mu a}\Lambda^p_{\mu' a}\Lambda^h_{\nu' k}B^P_{\mu'\nu'}C_{\sigma k}\Lambda^h_{\lambda i}B^Q_{\sigma\lambda}e^{-\varepsilon_{ak}t_\tau}C_{\nu i} \\
&= -c_{os}\sum_{\tau}^{n}\sum_{P}\sum_{\substack{\mu'\nu'\\ \sigma\lambda}} \hat{Q}^\tau_{\mu\mu'}B^P_{\mu'\nu'}\hat{P}^\tau_{\nu'\sigma}N^{QP}_\tau B^Q_{\sigma\lambda}\hat{P}_{\lambda\nu} \\
&= \sum_{\sigma,Q}Y^Q_{\mu\sigma}\hat{B}^Q_{\sigma\nu}
\end{aligned}\tag{2}$$

$$\begin{aligned}
\Omega^I_{\mu\nu} &= \sum_{ai} C_{\mu a}\Omega^I_{ai}C_{\nu i} = -c_{os}\sum_{\tau}^{n} w_\tau \sum_{ai,P} C_{\mu a}n^P_\tau\hat{B}^P_{ai}C_{\nu i} \\
&= -c_{os}\sum_{\tau}^{n} w_\tau \sum_{\mu'\nu',P}\sum_{ai} C_{\mu a}n^P_\tau\Lambda^p_{\mu' a}\Lambda^h_{\nu' i}B^P_{\mu'\nu'}e^{-\varepsilon_{ai}t_\tau}C_{\nu i} \\
&= -c_{os}\sum_{\tau}^{n}\sum_{\mu'\nu',P} n^P_\tau\hat{Q}^\tau_{\mu\mu'}B^P_{\mu'\nu'}\hat{P}^\tau_{\nu'\nu} = -c_{os}\sum_{\tau}^{n}\sum_{P} n^P_\tau B^P_{\mu\nu}
\end{aligned}\tag{3}$$

$$\begin{aligned}
\Omega^J_{\mu\nu} &= \sum_{ai} C_{\mu a}\Omega^J_{ai}C_{\nu i} = \sum_{ai,\mu'\nu'} C_{\mu a}\Lambda^p_{\mu' a}\hat{F}_{\mu'\nu'}\Lambda^h_{\nu' i}C_{\nu i} \\
&= \sum_{\mu'\nu'}\hat{Q}_{\mu\mu'}\hat{F}_{\mu'\nu'}\hat{P}_{\nu'\nu} = \hat{F}_{\mu\nu}
\end{aligned}\tag{4}$$

II. BLOCK-SPARSE MATRICES

Our block-sparse algebra are implemented to efficiently control memory demands, accuracy, and performance when the matrices are sparsely occupied. These matrices are divided in blocks of defined size, whose maximum is 96x96 in the present work. The allocation of each block is carried out by employing a *block allocator* which stores the block in one large, and dynamically growing *memory pool*, improving the performance for the allocation substantially (>20x). A *memory pool* is unique for each matrix, however, in case of three-dimensional tensors T^{l}_{ij} (l matrices with i rows and k columns), we allocate the blocks from all l matrices in the same pool.

Whether or not a block is allocated depends on the allocation threshold ϑ_a. Thus, only the blocks with L2-norm $\geq \vartheta_a$ are stored. The second screening threshold ϑ_m is used within the matrix-matrix multiplication routine, whose pseudo-code is summarized in Algorithm 1. Within this algorithm we multiply only the elements of the blocks that meet the screening criterium (line 6-9). The loops in lines 1-2 can be parallelized in a single loop over all ib- and jb indices. If the workload is not enough, the loop over k indices (line 3) is also parallelized and each thread computes its local Z block (line 7) according to the multiplication in line 8.

Algorithm 1 BSMat - Multiplication of two three-dimensional tensors: $C_{ij} = \sum_k A_{ik}\ B_{kj}$. nb_i = number of row-blocks of A and C, nb_j = number of column-blocks of B and C, nb_k = number of column- and row-blocks of A and B respectively,

```
1:  for block jb $\in nb_j$ do
2:    for block ib $\in nb_i$ do
3:      for block kb $\in nb_k$ do
4:        X = A.block(ib,kb)
5:        Y = B.block(kb,jb)
6:        if $(||X||*||Y||) \geq \vartheta_m$ then
7:          Z = C.block(ib,jb)
8:          $Z_{pq} += \sum_r X_{pr} Y_{rq}$
9:        end if
10:     end for
11:   end for
12: end for
```

III. THEORETICAL COMPUTATIONAL SCALING AND SPEEDUPS FOR LINEAR ALKANES

TABLE I. Computational scaling of a single optimization iteration for linear alkanes. We take into account the number of FLOPS of MO-SOS-RI-CC2 and ω-SOS-CDD-RI-CC2 in the def2-TZVP basis and two different density fitting metrics. We employ sparsity thresholds ϑ_a and ϑ_m equal to 10^{-7} and 10^{-9}, respectively. Values marked with an asterisk (*) are extrapolated conservatively.

Sample	No. of bf	MO-SOS-RI-CC2	ω-SOS-CDD-RI-CC2 ($\omega = 0$)			ω-SOS-CDD-RI-CC2 ($\omega = 0.1$)		
	def2-TZVP	FLOPS	FLOPS	Scaling	Speed Up	FLOPS	Scaling	Speed Up
$C_{40}H_{82}$	1732	1.2E+14	1.4E+14	—	x0.9	8.98E+13	—	x1.3
$C_{80}H_{162}$	3452	1.8E+15	1.5E+15	**3.50**	x1.2	5.67E+14	**2.70**	x3.2
$C_{160}C_{322}$	6892	2.9E+16	1.3E+16	**3.10**	x2.3	2.4E+15	**2.10**	x12.2
$C_{320}H_{642}$	13772	*4.8E+17	*9.5E+16	**2.80**	x5.1	8.6E+15	**1.85**	x56.0

IV. MO-SOS-RI-CC2 ALGORITHMS

```
for occ-batch do
    for all P $\in$ aux do
        read $B^Q_{bj}$ $\forall b, j \in$ occ-batch
    end for
    for all P $\in$ aux do
        read $\hat{B}^P_{bj}$ $\forall b$
        for all $\tau$ do
            ${}^{\tau}\hat{B}^P_{bj} = \hat{B}^P_{bj} e^{-\varepsilon_{bj} t_\tau} \forall b, j \in$ occ-batch;
        end for
    end for
    for all $\tau$ do
        for all P $\in$ aux do
            $n^P_\tau += {}^{\tau}\hat{B}^P_{bj}\hat{F}_{jb} \forall b, j \in$ occ-batch;
        end for
    end for
    for all $\tau$ do
        for $j \in$ occ-batch do
            $N^{QP}_\tau += B^Q_{bj}\,{}^{\tau}\hat{B}^P_{bj} \forall P, Q, b$
        end for
    end for
end for
scale $n^P_\tau$ and $N^{QP}_\tau$ by $-c_{os} w_\tau$
```

FIG. 1. Algorithm for the calculation of N^{QP}_τ and n^P_τ intermediates within the MO-SOS-RI-CC2 implementation.

```
for occ-batch do
    for all P $\in$ aux do
        read $\hat{B}^P_{ai}$ $\forall b$
        for all $\tau$ do
            ${}^{\tau}\hat{B}^P_{ai} = \hat{B}^P_{ai} e^{-\varepsilon_{ai} t_\tau} \forall a, i \in$ occ-batch;
        end for
    end for
    for all $\tau$ do
        for p $\in$ aux do
            $\Omega^I += {}^{\tau}\hat{B}^P_{ai} n^P_\tau \forall a, i \in$ occ-batch
        end for
    end for
    for all $\tau$ do
        for all i $\in$ occ-batch do
            $Y^Q_{ai} += {}^{\tau}\hat{B}^P_{ai} N^{QP}_\tau \forall P, Q, \mu$
        end for
    end for
    for all $i \in$ occ-batch do
        write $Y^Q_{ai} \forall Q, \mu$
    end for
end for
```

FIG. 2. Algorithm for the calculation of Y^Q_{ai} Ω^I_{ai} intermediates within the MO-SOS-RI-CC2 implementation.

```
for aux-batch do
    for Q $\in$ aux-batch do
        read $Y^{Q}_{ai} \forall a, i$
        $Y^{Q}_{\mu i} = C_{\mu a} Y^{Q}_{ai} \forall \mu, i$
    end for
    for Q $\in$ aux-batch do
        calc. $B^{Q}_{\mu\nu} \forall \mu, \nu$
        $\bar{\Omega}^{G}_{\mu i} = B^{Q}_{\mu\nu} Y_{\nu i} \forall \mu, i$
    end for
    for Q $\in$ aux-batch do
        read $B^{Q}_{ki} \forall \mu, \nu$
        $\Omega^{H}_{ai} = Y^{Q}_{ak} \hat{B}^{Q}_{ki} \forall a, i$
    end for
end for
$\Omega^{G}_{ai} = \Lambda^{p}_{\mu a} \bar{\Omega}^{G}_{\mu i}$
```

FIG. 3. Algorithm for the calculation of Ω^{G}_{ai} and Ω^{H}_{ai} contribution in the MO-SOS-RI-CC2 implementation.

4 Conclusion

This thesis comprises various methods to accelerate electronic structure theory calculations, namely seminumerical integration, integral screening, GPU acceleration, the resolution of the identity approximation, memory-optimized batching schemes, and the Cholesky decomposition of the ground state density.

Since the computation of Fock-exchange contributions typically represents the major bottleneck in Hartree-Fock and hybrid DFT calculations, the most significant contribution of this work is introduction of the sn-LinK method, which utilizes optimized integration grids, batch-wise integral screening, machine-optimized integral kernels, mixed precision arithmetic, and GPU acceleration to substantially accelerate (>1000× in some cases) the seminumerical evaluation of Fock-exchange. This facilitates the application of hybrid DFT to much larger molecules with much larger basis sets. In particular, the combination of the reduced (quadratic vs. quartic) *formal* scaling with respect to the basis set size and the *asymptotic* linear-scaling with respect to the system size, makes the method ideally suited for accurate calculations on large systems comprising hundreds or even thousands of atoms.

In addition, the evaluation of the Coulomb interaction and the semilocal exchange-correlation (XC) functional was also greatly accelerated: The former was treated with the resolution of the identity approximation (RI-J) and a modified J-engine algorithm, whereas the latter was reformulated in terms of matrix-matrix multiplications to achieve optimal performance, especially on GPUs. In both cases, the asymptotic scaling behavior was reduced by disregarding insignificant contributions, resulting in quadratic scaling for RI-J and linear-scaling for the semilocal XC integration. Thus, in combination with the sn-LinK method for the Fock-exchange, the three most expensive steps within hybrid-DFT applications were addressed.

In addition to Kohn-Sham density functional theory, the post-Kohn-Sham RPA and the post-Hartree-Fock SOS-CC2 method were also studied. The accuracy and the basis set dependence of RPA was significantly improved by employing range-separation to combine it with the semi-local PBE correlation functional. Moreover, the memory demand of RPA calculations was greatly reduced by incorporating an optimized batching scheme leading to an optimal memory vs. runtime trade-off. Furthermore, application of a variant of this batching scheme together with the Cholesky decomposition of the ground state density resulted in an efficient and asymptotically linear-scaling SOS-CC2 method.

Generally, all contributions within this thesis had one common objective: Enabling accurate electronic structure theory calculations on large molecules at low computational cost. In this way, more complex molecular environments, e.g., explicit solvent effects, protein catalysis, DNA interactions, supramolecular host-guest complexes etc., can now be tackled at higher accuracy (e.g., larger basis sets, better density functional approximations, more sampling of the configuration space) than before. Therefore, the present work represents a substantial step towards the goal of solving the Schrödinger equation as accurately as possible while also utilizing the available computing hardware as efficiently as possible, opening the way to gain novel access into large and complex systems.

Bibliography

[1] E. Schrödinger, *Phys. Rev.* **1926**, *28*, 1049–1070.

[2] W. Kohn, L. J. Sham, *Phys. Rev.* **1965**, *140*, A1133–A1138.

[3] A. D. Becke, *J. Chem. Phys.* **1993**, *98*, 1372–1377.

[4] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, M. J. Frisch, *J. Phys. Chem.* **1994**, *98*, 11623–11627.

[5] C. Adamo, V. Barone, *J. Chem. Phys.* **1999**, *110*, 6158–6170.

[6] M. Ernzerhof, G. E. Scuseria, *J. Chem. Phys.* **1999**, *110*, 5029–5036.

[7] N. Mardirossian, M. Head-Gordon, *J. Chem. Phys.* **2016**, *144*, 214110.

[8] R. A. Friesner, *Chem. Phys. Lett.* **1985**, *116*, 39–43.

[9] R. A. Friesner, *J. Chem. Phys.* **1986**, *85*, 1462–1468.

[10] R. A. Friesner, *J. Chem. Phys.* **1987**, *86*, 3522–3531.

[11] R. A. Friesner, *J. Phys. Chem.* **1988**, *92*, 3091–3096.

[12] M. N. Ringnalda, M. Belhadj, R. A. Friesner, *J. Chem. Phys.* **1990**, *93*, 3397–3407.

[13] R. A. Friesner, J. A. Bentley, M. Menou, C. Leforestier, *J. Chem. Phys.* **1993**, *99*, 324–335.

[14] R. B. Murphy, Y. Cao, M. D. Beachy, M. N. Ringnalda, R. A. Friesner, *J. Chem. Phys.* **2000**, *112*, 10131–10141.

[15] Y. Cao, M. D. Beachy, D. A. Braden, L. Morrill, M. N. Ringnalda, R. A. Friesner, *J. Chem. Phys.* **2005**, *122*, 224116.

[16] F. Neese, F. Wennmohs, A. Hansen, U. Becker, *Chem. Phys.* **2009**, *356*, 98–109.

[17] P. Plessow, F. Weigend, *J. Comput. Chem.* **2012**, *33*, 810–816.

[18] A. D. Bochevarov, E. Harder, T. F. Hughes, J. R. Greenwood, D. A. Braden, D. M. Philipp, D. Rinaldo, M. D. Halls, J. Zhang, R. A. Friesner, *Int. J. Quantum Chem.* **2013**, *113*, 2110–2142.

[19] H. Bahmann, M. Kaupp, *J. Chem. Theory Comput.* **2015**, *11*, 1540–1548.

[20] T. M. Maier, H. Bahmann, M. Kaupp, *J. Chem. Theory Comput.* **2015**, *11*, 4226–4237.

[21] Y. Cao, T. Hughes, D. Giesen, M. D. Halls, A. Goldberg, T. R. Vadicherla, M. Sastry, B. Patel, W. Sherman, A. L. Weisman, R. A. Friesner, *J. Comput. Chem.* **2016**, *37*, 1425–1441.

[22] S. Klawohn, H. Bahmann, M. Kaupp, *J. Chem. Theory Comput.* **2016**, *12*, 4254–4262.

[23] F. Liu, J. Kong, *J. Chem. Theory Comput.* **2017**, *13*, 2571–2580.

[24] F. Liu, J. Kong, *Chem. Phys. Lett.* **2018**, *703*, 106–111.

[25] G. L. Stoychev, A. A. Auer, R. Izsak, F. Neese, *J. Chem. Theory Comput.* **2018**, *14*, 619–637.

[26] T. M. Maier, Y. Ikabata, H. Nakai, *J. Chem. Theory Comput.* **2019**, *15*, 4745–4763.

[27] R. Grotjahn, F. Furche, M. Kaupp, *J. Chem. Theory Comput.* **2019**, *15*, 5508–5522.

[28] C. Holzer, *J. Chem. Phys.* **2020**, *153*, 184115.

[29] B. Helmich-Paris, B. de Souza, F. Neese, R. Izsak, *J. Chem. Phys.* **2021**, *155*, 104109.

[30] J. C. Burant, G. E. Scuseria, M. J. Frisch, *J. Chem. Phys.* **1996**, *105*, 8969–8972.

[31] E. Schwegler, M. Challacombe, *J. Chem. Phys.* **1996**, *105*, 2726–2734.

[32] E. Schwegler, M. Challacombe, M. Head-Gordon, *J. Chem. Phys.* **1997**, *106*, 9708–9717.

[33] M. Challacombe, E. Schwegler, *J. Chem. Phys.* **1997**, *106*, 5526–5536.

[34] C. Ochsenfeld, C. A. White, M. Head-Gordon, *J. Chem. Phys.* **1998**, *109*, 1663–1669.

[35] C. Ochsenfeld, *Chem. Phys. Lett.* **2000**, *327*, 216–223.

[36] J. Kussmann, C. Ochsenfeld, *J. Chem. Phys.* **2013**, *138*, 134114.

[37] J. Kussmann, C. Ochsenfeld, *J Chem Theory Comput* **2015**, *11*, 918–922.

[38] J. Kussmann, C. Ochsenfeld, *J Chem Theory Comput* **2017**, *13*, 3153–3159.

[39] T. H. Thompson, C. Ochsenfeld, *J. Chem. Phys.* **2019**, *150*, 044101.

[40] *ANSI/IEEE Std 754-1985* **1985**, 1–20.

[41] J. Almloef, K. Faegri, Jr., K. Korsell, *J. Comput. Chem.* **1982**, *3*, 385–399.

[42] M. Haeser, R. Ahlrichs, *J. Comput. Chem.* **1989**, *10*, 104–11.

[43] J. Noga, J. Simunek, *Chem. Phys.* **2009**, *356*, 1–6.

[44] W. Klopper, W. Kutzelnigg, *Chem. Phys. Lett.* **1987**, *134*, 17–22.

[45] V. Termath, W. Klopper, W. Kutzelnigg, *J. Chem. Phys.* **1991**, *94*, 2002–2019.

[46] S. Ten-no, *Chem. Phys. Lett.* **2004**, *398*, 56–61.

[47] T. B. Adler, G. Knizia, H.-J. Werner, *J. Chem. Phys.* **2007**, *127*, 221106.

[48] A. D. Becke, *J. Chem. Phys.* **1988**, *88*, 2547–2553.

[49] J. W. Mintmire, B. I. Dunlap, *Phys. Rev. A* **1982**, *25*, 88–95.

[50] C. A. White, M. Head-Gordon, *J. Chem. Phys.* **1996**, *104*, 2620–2629.

[51] F. Neese, *J. Comput. Chem.* **2003**, *24*, 1740–1747.

[52] F. Weigend, R. Ahlrichs, *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297–3305.

[53] V. V. Karasiev, *J. Chem. Phys.* **2003**, *118*, 8576–8583.

[54] A. D. Becke, *J. Chem. Phys.* **2003**, *119*, 2972–2977.

[55] A. D. Becke, *J. Chem. Phys.* **2005**, *122*, 064101.

[56] J. P. Perdew, V. N. Staroverov, J. Tao, G. E. Scuseria, *Phys. Rev. A* **2008**, *78*, 052513.

[57] E. R. Johnson, *J. Chem. Phys.* **2013**, *139*, 074110.

[58] A. D. Becke, *J. Chem. Phys.* **2013**, *138*, 074109.

[59] J. Kong, E. Proynov, *J. Chem. Theory Comput.* **2016**, *12*, 133–143.

[60] J. Kong, E. Proynov, J. Yu, R. Pachter, *J. Phys. Chem. Lett.* **2017**, *8*, 3142–3146.

[61] D. Bohm, D. Pines, *Phys. Rev.* **1953**, *92*, 609–625.

[62] M. Gell-Mann, K. A. Brueckner, *Phys. Rev.* **1957**, *106*, 364–368.

[63] D. C. Langreth, J. P. Perdew, *Phys. Rev. B* **1977**, *15*, 2884–2901.

[64] J. P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.

[65] J. Toulouse, F. Colonna, A. Savin, *J. Chem. Phys.* **2005**, *122*, 014110.

[66] J. Toulouse, I. C. Gerber, G. Jansen, A. Savin, J. G. Angyan, *Phys. Rev. Lett.* **2009**, *102*, 096404.

[67] J. Toulouse, W. Zhu, J. G. Angyan, A. Savin, *Phys. Rev. A* **2010**, *82*, 032502.

[68] O. Franck, B. Mussard, E. Luppi, J. Toulouse, *J. Chem. Phys.* **2015**, *142*, 074107.

[69] A. Luenser, H. F. Schurkus, C. Ochsenfeld, *J. Chem. Theory Comput.* **2017**, *13*, 1647–1655.

[70] D. Graf, M. Beuerle, H. F. Schurkus, A. Luenser, G. Savasci, C. Ochsenfeld, *J. Chem. Theory Comput.* **2018**, *14*, 2505–2515.

[71] O. Christiansen, H. Koch, P. Jorgensen, *Chem. Phys. Lett.* **1995**, *243*, 409–418.

[72] N. O. C. Winter, C. Haettig, *J. Chem. Phys.* **2011**, *134*, 184101.

[73] M. Born, R. Oppenheimer, *Ann. Phys.* **1927**, *389*, 457–484.

[74] D. R. Hartree, *Math. Proc. Cambridge Philos. Soc.* **1928**, *24*, 89–110.

[75] W. Pauli, *Z. Phys.* **1925**, *31*, 765–783.

[76] J. C. Slater, *Phys. Rev.* **1929**, *34*, 1293–1322.

[77] V. Fock, *Z. Phys.* **1930**, *61*, 126–148.

[78] C. C. J. Roothaan, *Rev. Mod. Phys.* **1951**, *23*, 69–89.

[79] D. K. Jordan, D. A. Mazziotti, *J. Chem. Phys.* **2005**, *122*, 084114.

[80] A. M. N. Niklasson, V. Weber, M. Challacombe, *J. Chem. Phys.* **2005**, *123*, 044107.

[81] E. H. Rubensson, E. Rudberg, P. Salek, *J. Chem. Phys.* **2008**, *128*, 074106.

[82] P. Suryanarayana, *Chem. Phys. Lett.* **2013**, *555*, 291–295.

[83] J. P. Perdew, K. Schmidt in Density Functional Theory and its Application to Materials, **2001**, pp. 1–20.

[84] P. A. M. Dirac, *Math. Proc. Cambridge Philos. Soc.* **1930**, *26*, 376–385.

[85] S. H. Vosko, L. Wilk, M. Nusair, *Can. J. Phys.* **1980**, *58*, 1200–1211.

[86] Perdew, Wang, *Phys. Rev. B* **1992**, *45*, 13244–13249.

[87] A. D. Becke, *Phys. Rev. A* **1988**, *38*, 3098–3100.

[88] C. Lee, W. Yang, R. G. Parr, *Phys. Rev. B* **1988**, *37*, 785.

[89] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, C. Fiolhais, *Phys. Rev. B* **1992**, *46*, 6671–6687.

[90] A. D. Becke, M. R. Roussel, *Phys. Rev. A* **1989**, *39*, 3761–3767.

[91] J. Tao, J. P. Perdew, V. N. Staroverov, G. E. Scuseria, *Phys. Rev. Lett.* **2003**, *91*, 146401.

[92] N. Mardirossian, M. Head-Gordon, *J. Chem. Phys.* **2015**, *142*, 074111.

[93] C. Moller, M. S. Plesset, *Phys. Rev.* **1934**, *46*, 618–622.

[94] S. Grimme, F. Neese, *J. Chem. Phys.* **2007**, *127*, 154116.

[95] L. Goerigk, S. Grimme, *J. Chem. Theory Comput.* **2011**, *7*, 291–309.

[96] S. Grimme, M. Steinmetz, *Phys. Chem. Chem. Phys.* **2016**, *18*, 20926–20937.

[97] N. Mardirossian, M. Head-Gordon, *J. Chem. Phys.* **2018**, *148*, 241736.

[98] R. E. Stratmann, G. E. Scuseria, M. J. Frisch, *Chem. Phys. Lett.* **1996**, *257*, 213–223.

[99] P. Pulay, *Mol. Phys.* **1969**, *17*, 197–204.

[100] F. Weigend, *Phys. Chem. Chem. Phys.* **2006**, *8*, 1057–1065.