Received: 10 November 2021 | Revised: 13 May 2022 | Accepted: 27 June 2022

DOI: 10.1002/jcc.26968

RESEARCH ARTICLE

Journal of COMPUTATIONAL CHEMISTRY WILEY

The numerical evaluation of Slater integrals on graphics processing units

Duy-Khoi Dang[1] | Leighton W. Wilson[1,2] | Paul M. Zimmerman[1]

[1]Department of Chemistry, University of Michigan, Ann Arbor, Michigan, USA

[2]Department of Mathematics, University of Michigan, Ann Arbor, Michigan, USA

Correspondence
Paul M. Zimmerman, Department of Chemistry, University of Michigan, 930 N. University Ave, Ann Arbor, MI 48109, USA.
Email: paulzim@umich.edu

Abstract

This article presents SlaterGPU, a graphics processing unit (GPU) accelerated library that uses OpenACC to numerically compute Slater-type orbital (STO) integrals. The electron repulsion integrals (ERI) are computed under the RI approximation using the Coulomb potential of the Slater basis function. To fully realize the performance capabilities of modern GPUs, the Slater integrals are evaluated in mixed-precision, resulting in speedups for the ERIs of over 80×. Parallelization on multiple GPUs allows for integral throughput of over 3 million integrals per second. This places STO integral throughput within reach of single-threaded, conventional Gaussian integration schemes. To test the quality of the integrals, the fluorine exchange reaction barrier in fluoromethane was computed using heat-bath configuration interaction (HBCI). In addition, the singlet-triplet gap of cyclobutadiene was examined using HBCI in a triple-ζ, polarized basis set. These benchmarks demonstrate the library's ability to generate the full set of integrals necessary for configuration interaction with up to 6*h* functions in the auxiliary basis.

KEYWORDS

configuration interaction, GPU, integrals, Slater orbitals

1 | INTRODUCTION

Advances in quantum chemistry and computer hardware have facilitated the routine use of electronic structure simulations for chemical applications. Some of the most widely used theories make use of one-electron, atom-centered basis functions[1] to represent the electron density. The simplest wave function that approximately solves the Schrödinger equation is Hartree–Fock (HF), which represents the wave function using a single Slater determinant. While HF is not a quantitatively accurate theory, it forms the basis for more sophisticated theories. In many canonical, post-HF methods, evaluation of Hamiltonian elements in the Schrödinger picture requires computing integrals of the form

$$O_{\mu\nu} = \langle \chi_\mu | \widehat{O}_1 | \chi_\nu \rangle = \int \chi_\mu(\mathbf{r}) \widehat{O}(\mathbf{r}) \chi_\nu(\mathbf{r}) d\mathbf{r}, \tag{1}$$

$$\begin{aligned} O_{\mu\nu\lambda\sigma} &= \langle \chi_\mu(1)\chi_\nu(1) \big| \widehat{O}_2 \big| \chi_\lambda(2)\chi_\sigma(2) \rangle \\ &= \iint \chi_\mu(\mathbf{r}_1)\chi_\nu(\mathbf{r}_1) \widehat{O}_2 \chi_\lambda(\mathbf{r}_2)\chi_\sigma(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2. \end{aligned} \tag{2}$$

The first equation denotes 1-electron quantities such as the overlap $\widehat{O}_1 = 1$, the kinetic energy $\widehat{O}_1 = -\frac{1}{2}\nabla^2$, and the nuclear attraction $\widehat{O}_1 = \frac{Z_A}{R_{1A}}$ operators. 2-electron operators include the Coulomb repulsion $\widehat{O}_2 = \frac{1}{r_{12}}$, where r_{12} is the distance between electrons 1 and 2. Derivatives of these terms, for example with respect to nuclear position, are also quantities of interest.

Amongst these integrals, the electron repulsion integrals (ERIs) are the most difficult (and numerous) to evaluate, being 2-electron quantities that require six-dimensional integration. In addition, the $\frac{1}{r_{12}}$ operator contains a singularity at every point in three-dimensional space, further challenging their integration. Consequently, the choice of basis is important for not only accurate representation of the molecular wave function, but also for computational evaluation of integrals.

One physically motivated choice are Slater-type orbitals (STOs), which are hydrogen-like orbitals of the form

$$S(\zeta,n,l,m,r,\boldsymbol{\theta},\phi) = N^{\mathrm{STO}} r^{n-1} e^{-\zeta r} Z_{lm}(\boldsymbol{\theta},\phi), \tag{3}$$

where ζ is the exponent, n,l,m are the usual atomic quantum numbers, r,θ,ϕ are spherical coordinates, N^{STO} is the normalization constant, and Z_{lm} are the spherical harmonics.[1–3] The STOs can satisfy the Kato cusp and exponential decay of atomic wave functions,[1,4,5] making them a natural basis choice for quantum chemical calculations. However, the ERIs over STOs do not have a known general analytic form.

The difficulties of STO integration led to the expansion of STOs in terms of Gaussian-type orbitals (GTOs)[6]

$$G(\alpha,n,l,m,r,\boldsymbol{\theta},\phi)=N^{\mathrm{GTO}}e^{-\alpha r^2}S_{lm}(r,\boldsymbol{\theta},\phi), \tag{4}$$

where the S_{lm} are the real solid harmonics.[1] The GTOs benefit from the Gaussian product rule, that is, the product of two GTOs is again a GTO, which simplifies Equation (2) from a 4-center, 2-electron integral to 2-center, 2-electron. The 2-center, 2-electron integral can be evaluated over the Coulomb potential of one GTO reducing a six-dimensional integral to three-dimensions. These nice analytical properties of GTOs facilitated the development of fast analytical integral evaluation.[7–11]

While GTOs can be quickly evaluated using modern integral libraries, they do not contain the correct short- and long-range behaviors expected in molecular wave functions.[12] For example, the cusp near the nucleus is important for computing properties such as nuclear magnetic resonance shifts and polarizabilities,[13,14] but the cusp is not present in the GTO basis, and only crudely treated by using contracted sets of GTOs. Exponential decay of the wave function for an accurate description is required for precise quantification of the HOMO energy, but this behavior is also absent in GTOs.[15]

The imperfections of GTO basis sets have left room for the continued development and use of STOs for quantum chemical applications. Several schemes have been developed to compute general STO integrals. One approach is to expand each STO in a very large number of GTOs and compute the GTO integrals analytically.[16–18] In addition, Monte Carlo has been used to correct integrals over Gaussian expansions to evaluate the Slater quantity.[19] These schemes are prohibitively expensive for routine use. While the focus of this article is on the use of STOs in integrals such as Equations (1) and (2), STOs have seen frequent use in quantum Monte Carlo wave functions, where the 1- and 2-electron integrals are not important.[20–23]

An attractive alternative to explicit evaluation of STO ERIs involves density fitting—in particular the resolution of the identity (RI) approximation (see Section 2)—which allows Equation (2) to be approximated as a tensor product of 2- and 3-index ERIs. Within the RI approximation, one of the two electrons is described by a single basis function. This facilitates the use of a Coulomb potential to represent one electron without relying on an explicit basis set product rule—which does not exist for STOs—to condense multiple centers. This simplification, which is only necessary for systems with at least four distinct atomic centers, allows STO integration of ERIs to be amenable to numerical quadrature schemes. The Amsterdam Density Functional (ADF) package and other density functional theory (DFT) codes implement a density fitting approach to use STOs in DFT.[3,13,24,25] Other density fitting frameworks have allowed STOs to be used in approximate MP2,[26] double-hybrid DFT,[27] and Green's function methods.[28,29] These previous STO studies, however, did not generate the full complement of ERIs required for correlated methods such as those based on configuration interaction,[30–33] multiconfigurational self-consistent field[34–37] and coupled cluster.[38–41]

This study introduces and benchmarks a graphics processing unit (GPU) library for evaluating STO integrals for wave function theories. The article will show that these can be accurately and efficiently evaluated using numerical integration by combining the RI approximation with the STO Coulomb potential. The large number of processing cores and high memory bandwidth make modern GPUs the architecture of choice for evaluating and summing numerical grids. For additional performance, the integrals are also computed using mixed-precision evaluation. Timings suggest that this library allows STOs to be useful alongside strongly correlated wave function theories. Accuracy benchmarks indicate minimal loss in accuracy from using mixed-precision relative to double-precision. The resulting code, called SlaterGPU,[42] is the first reported library to use GPUs to accelerate STO integrals and evaluate the full set of 1- and 2-electron STO integrals up to the *6h* subshell as well as *5g* for first derivatives for the auxiliary basis.

2 | THEORY AND COMPUTATIONAL DETAILS

The present STO integral scheme relies on numerical integration over atom-centered grids. Grid-based integration can make use of single instruction, multiple data parallelism and therefore can leverage GPU hardware for acceleration. Even with this acceleration, the six-dimensional ERIs remain too costly for routine computations. The dimensionality of integration can be reduced, however, by employing the RI approximation,[43–45] where the Coulomb potentials for the auxiliary basis functions are known analytically. The various components of the STO integral algorithm are explained in the following sections: the RI, Grid Construction, Implementation on GPU, and Computational Details.

2.1 | Resolution of the identity

This section focuses on simplifying the challenging ERIs for numerical evaluation. The expressions for numerically evaluating the 1-electron integrals are listed in Section S1 of the Supporting Information. In the RI approximation, the 4-index ERIs $(\mu\nu|\lambda\kappa)$ are decomposed into tensor products of 2- and 3-index integrals by representing the density in terms of an auxiliary basis. Using the Coulomb metric, the integral can be approximated with the expression[43–45]

$$(\mu\nu|\lambda\kappa)\approx\sum_{PQ}(\mu\nu|P)(PQ)^{-1}(Q|\lambda\kappa)=\sum_{Q}B^{Q}_{\mu\nu}B^{Q}_{\lambda\kappa}, \tag{5}$$

where

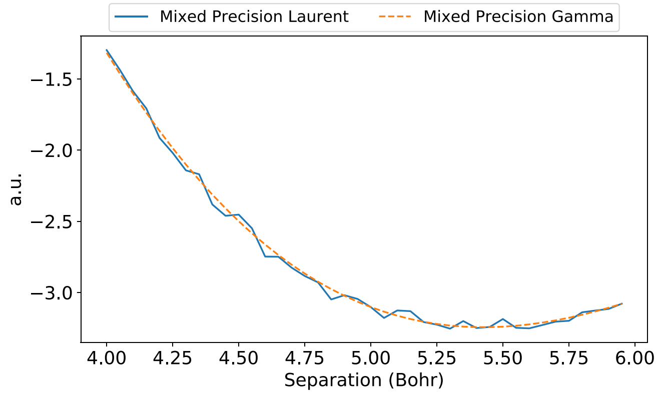

FIGURE 1 The $(6H|6H)$ integral is scanned in the $(0.370, 0.370, 0.853)$ direction with the left center at the origin. Evaluations are in mixed precision using either the Laurent polynomial expansion of Equation (10) or the lower incomplete gamma function, where mixed precision is defined similarly to Equation (14). Both basis functions have $m = 0$ and $\zeta = 1$

$$B_{\mu\nu}^{Q} = \sum_{P} (\mu\nu|P)(PQ)^{-1/2}. \tag{6}$$

ALGORITHM 1 **GPU compute structure for generating 3-center ERIs**

```
#pragma omp parallel for schedule(dynamic)
for $A, B, C$ in Atom List   $//A, B$ overall Atoms, $C \geq B$.
    Generate $x_A, x_B, x_C, w(x_A), w(x_B), w(x_C)$
    $x \leftarrow x_A \cup x_B \cup x_C$
    $w(x) \leftarrow w(x_A) \cup w(x_B) \cup w(x_C)$
    for $P_i$ in Aux($A$).
        Compute $V_{P_i}(x)$.
    for $\chi_{\mu_j}$ in Basis($B$).
        Compute $\chi_{\mu_j}(x)$.
    for $\chi_{\nu_k}$ in Basis($C$).
        Compute $\chi_{\nu_k}(x)$.
    for $P_i \in \text{Aux}(A), \mu_j \in \text{Basis}(B), \nu_k \in \text{Basis}(C)$.
        $(P_i|\mu_j\nu_k) \leftarrow \sum_x V_{P_i}(x)\chi_{\mu_j}(x)\chi_{\nu_k}(x)w(x)$.
```

Aux(·) and Basis(·) denote the set of auxiliary and main basis functions centered at ·, respectively.

In a numerical integration scheme, Equation (5) not only reduces the count of numerical integrals for a given basis set size (N) from $O(N^4)$ to $O\left(N^3\right)$, it also has a secondary consequence that is useful in the context of STO basis functions. Specifically, the integral

$$(P|\mu\nu) = \iint \chi_P(\mathbf{r}_1)\frac{1}{r_{12}}\chi_\mu(\mathbf{r}_2)\chi_\nu(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{7}$$

can be simplified to

$$(P|\mu\nu) = \int V_C^P(\mathbf{r})\chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r})d\mathbf{r} \tag{8}$$

by using the known analytical form of the single-Slater Coulomb potential. In spherical coordinates, this potential has the form[3]

$$V_C(\zeta, n, l, m, r, \boldsymbol{\theta}, \phi) = \frac{4\pi(2\zeta)^{n+(1/2)}}{\sqrt{(2n)!}(2l+1)} Z_{lm}(\boldsymbol{\theta}, \phi) I_{nl}(r), \tag{9}$$

where

$$I_{nl}(r) = r^{-l-1}\int_0^r (r')^{n+l+1} e^{-\zeta r'} dr' + r^l \int_r^\infty (r')^{n-l} e^{-\zeta r'} dr'. \tag{10}$$

I_{nl} has analytic expressions using finite Laurent polynomials for each n, l of interest.

For large angular momentum l, the Laurent expressions (see Section S2 of the Supporting Information)—and especially their derivatives—exhibit numerical instability, especially when using mixed-precision arithmetic, which is essential for high performance integral evaluation. This can result in nonsmooth integrals as shown in Figure 1, which can in turn result in nonsmooth or discontinuous energies. Instead of applying the Laurent expressions, Equation (10) can be evaluated using lower incomplete gamma functions, which have fast, numerically precise implementations.[46] The final form of Equation (10) used in the current implementation of SlaterGPU is

$$I_{nl}(r) = r^{-l-1}\zeta^{-l-n-2} \left\{ (r\zeta)^{2l+1}[(-l+n)! - \gamma(-l+n+1, r\zeta)] + \gamma(l+n+2, r\zeta) \right\}, \tag{11}$$

where $\gamma(s, x)$ is the lower incomplete gamma function,

$$\gamma(s, x) = \int_0^x t^{s-1} e^{-t} dt. \tag{12}$$

After evaluation of all 2- and 3-center Coulomb integrals, the full set of 4-index ERIs can be reconstructed using Equation (5). SlaterGPU therefore uses the RI approximation for Slater integrals, similar to prior implementations for DFT applications,[24,25] but further provides all 4-index integrals, $(ij|kl)$, which are not generated or required for DFT. This allows the SlaterGPU library to be useful for wave function theories, which require a larger set of ERIs. In particular, while prior codes demonstrated applicability to $l \leq 3$,[25,29] SlaterGPU is shown here to be useful for $l \leq 5$.

2.2 | Grid construction

When numerically evaluating integrals over atomic orbitals (AOs), the choice of grid is important. The atom-centered grids used here borrow their core concept from prior studies, especially those involving integration of DFT functionals.[24,47–51] The accepted

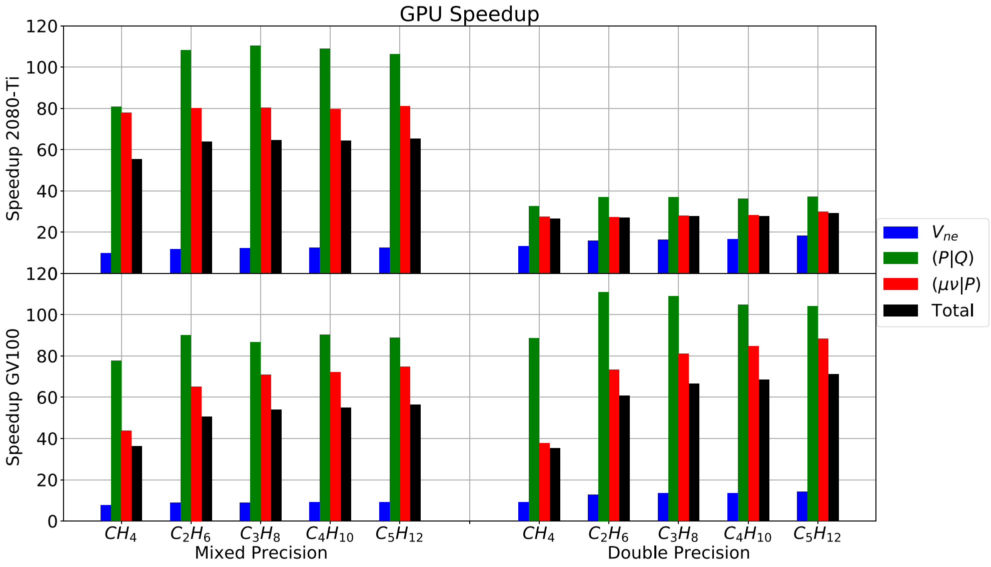

FIGURE 2 The GPU speedups over CPU for integral evaluation on the 2080-Ti (top) and GV100 (bottom) for various alkanes using the DZP basis from ADF. The speedups are partitioned into the various integrals. Speedups for mixed (left) and double (right) precision evaluations are also shown

TABLE 1 Double-precision timing data (in seconds) for various alkanes

	Basis size		CPU time			2080-Ti time			V100 time		
Molecule	Main	Aux	V_{ne}	$(P|Q)$	$(\mu\nu|Q)$	V_{ne}	$(P|Q)$	$(\mu\nu|Q)$	V_{ne}	$(P|Q)$	$(\mu\nu|Q)$
CH_4	35	224	3.282	14.02	56.88	0.2483	0.4316	2.062	0.3583	0.1581	1.513
C_3H_8	85	516	40.28	67.98	786.2	2.452	1.835	28.00	2.991	0.6237	9.693
C_5H_{12}	135	808	166.1	163.5	3286	9.125	4.385	110.1	11.71	1.570	37.20

Note: Each atom contributes 46,200 grid points.

route for integrating the exchange-correlation energy is to build atomic grids as products of radial and angular grids, then reweight these using Voronoi polyhedra centered about the nuclei. The atom-centered grids are necessary to capture the spherical harmonics and radial decay of AOs, while partitioning three-dimensional space into polyhedra divides the grid into volumes centered around each nucleus. The Voronoi boundaries are smoothed and reweighted to avoid double counting of volume elements.[48,49,51] This same framework is used in SlaterGPU, though only a maximum of three atom-centered grids are required for any given integral, since the ERIs only involve up to three centers at a time in the RI approximation. In a polyatomic system, this greatly simplifies the form of the integration grid, keeping each integral grid small enough to be efficiently evaluated. The grids chosen for this implementation are the "Log3" grid from Mura and Knowles[51] for the radial component, and the Lebedev grid[52] for the angular component. Both grids are widely used in electronic structure codes. Once each atom-centered grid is generated, and the Becke partitioning scheme[48] is applied, the 3-center integral $(\mu\nu|P)$ can be evaluated over the grid points x and grid weights $w(x)$ as

$$(\mu\nu|P) = N^{STO}_{V_C} N^{STO}_{\chi_\mu} N^{STO}_{\chi_\nu} \sum_x \overline{V}_C(x)\overline{\chi}_\mu(x)\overline{\chi}_\nu(x)w(x), \qquad (13)$$

where $\overline{V}_C$, $\overline{\chi}_\mu$, $\overline{\chi}_\nu$ are the Coulomb potential and basis functions with their respective normalization constants, $N^{STO}_{V_C}$, $N^{STO}_{\chi_\mu}$, and $N^{STO}_{\chi_\nu}$, factored out. While Lebedev and Mura–Knowles grids are used with Becke weights, any quadrature grid and weighting scheme can be used in Equation (13). The 2-index integrals $(P|Q)$ are evaluated in a similar manner.

2.3 | Implementation on GPU

All integral code in this study is written in C++ using OpenACC for GPU acceleration, which has the advantage of being based on pragma directives allowing the same code base to be compiled to run on CPUs or GPUs. When evaluating Equation (10), a modified version from the Cephes library[53] was used for the lower incomplete gamma function, noting that OpenACC allows these implementations to be used directly. Most GPUs contain more single precision compute units than double precision, so mixed precision operations are an attractive choice in a practical implementation.[54–57] For example, the 2080-Ti contains $\frac{1}{32}$ the double precision units compared with single precision units, while the GV100 contains $\frac{1}{2}$ the double precision units compared with single precision units. In SlaterGPU, mixed precision is available, where evaluations over the grid are performed using single-precision arithmetic, and the final summation occurs in double precision. In mixed precision, Equation (13) becomes

$$(\mu\nu|P)_{64} = N^{STO}_{V_C} N^{STO}_{\chi_\mu} N^{STO}_{\chi_\nu} \sum_x \overline{V}_P(x)_{32}\overline{\chi}_\mu(x)_{32}\overline{\chi}_\nu(x)_{32}w(x)_{32}, \qquad (14)$$

TABLE 2 Mixed-precision timing data (in seconds) for various alkanes

	Basis size		CPU time			2080-Ti time			V100 time		
Molecule	Main	Aux	V_{ne}	$(P\|Q)$	$(\mu\nu\|Q)$	V_{ne}	$(P\|Q)$	$(\mu\nu\|Q)$	V_{ne}	$(P\|Q)$	$(\mu\nu\|Q)$
CH_4	35	224	2.095	8.686	34.34	0.2139	0.1076	0.4415	0.2708	0.1117	0.7867
C_3H_8	85	516	24.95	42.34	484.8	2.037	0.3838	6.029	2.795	0.4892	6.832
C_5H_{12}	135	808	95.62	98.58	1957	7.680	0.9275	24.12	10.58	1.109	26.21

Note: Each atom contributes 46,200 grid points.

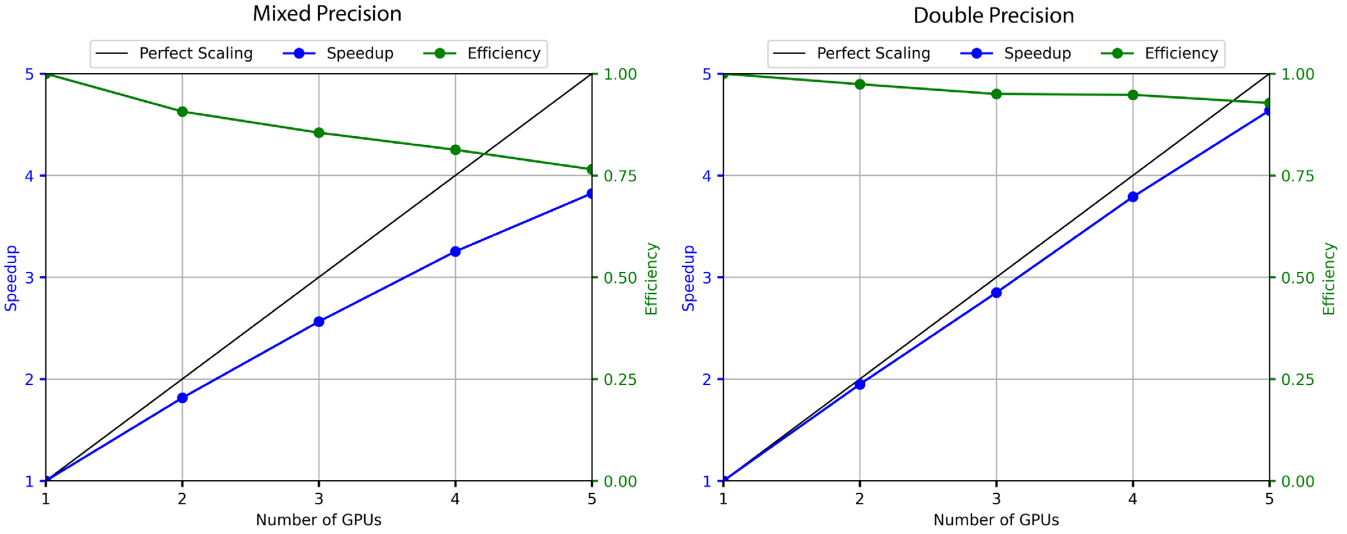

FIGURE 3 Multi-GPU speedups over single GPU and parallel efficiency for mixed (left) and double (right) precision evaluation of the 3-center ERIs for C_9H_{20}. There are a total of 76,873,200 3-center integrals. Perfect scaling is plotted as a solid black line. All GPUs are co-located on a single compute node. Single GPU run times were 125 and 542 s for mixed- and double-precision implementations, respectively

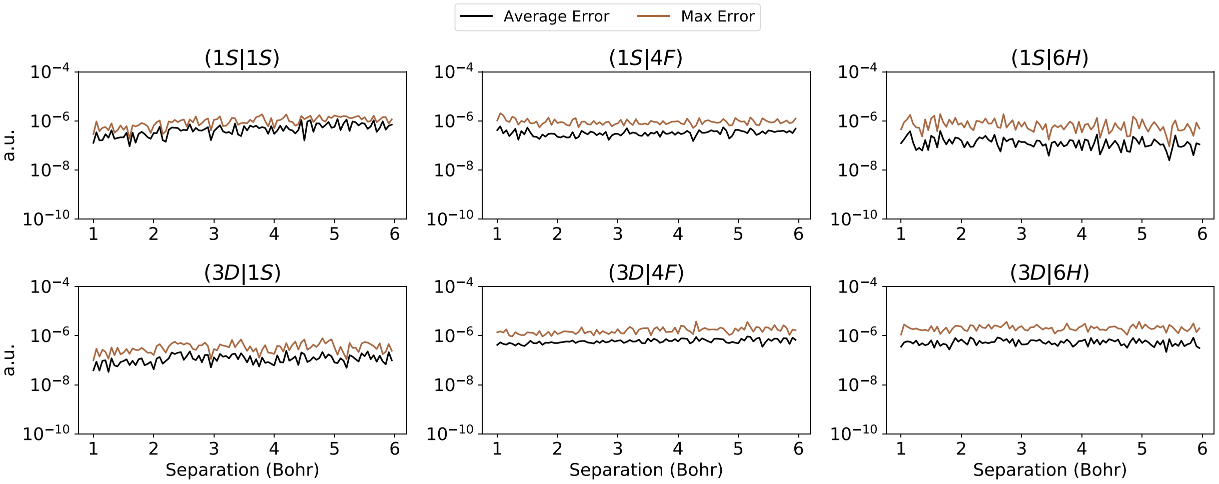

FIGURE 4 The max and average errors between mixed- and double-precision integral evaluation are plotted for various basis functions. All basis functions have $\zeta = 1$ and $m = 0$. The max and average errors are computed over internuclear distance scans based on the 16 all-positive directions of a Lebedev grid

where the subscript refers to the bits of precision of the quantity. Factorization of the normalization constants reduces the number of floating point operations required, which is essential for high performance.

In addition to making use of the greater quantity of single-precision compute units in GPUs, single precision also reduces storage and memory bandwidth demands by a factor of 2. A generalization of the

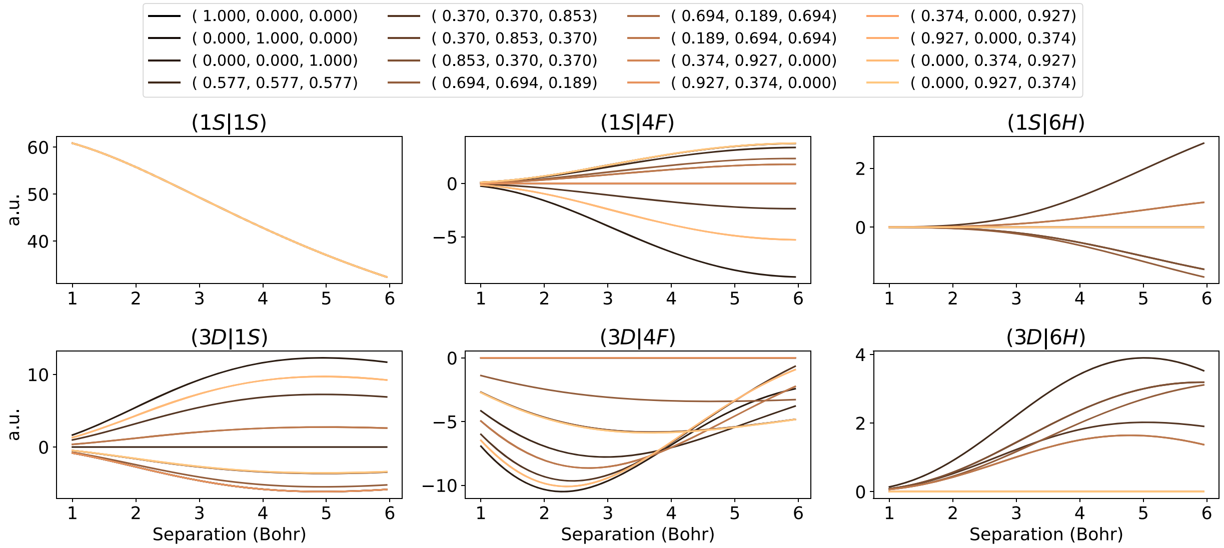

FIGURE 5 The value of 2-center ERIs are evaluated in mixed precision. All basis functions have $\zeta = 1$ and the right basis is scanned radially away from the origin in various directions. The directions selected are provided in the legend and were selected using the 16 all-positive directions of an 86-point Lebedev grid. For all integrals shown here, $m = 0$. The legend entries are direction unit vectors

mixed precision procedure would be to adaptively determine which integrals to evaluate at each level of precision, as has been done in (analytic) GTO integration.[57] This is not done here; instead, the accuracy of the mixed precision approach is evaluated in comparison to double precision integration.

In the GPU computing framework, data transfers between CPU and GPU incur large overhead penalties, and thus it is necessary to minimize these transactions for maximum performance. As such, all quantities in Equation (14) are generated and evaluated directly on GPU. The grid x and its weights $w(x)$ only depend on the set of atoms and not the basis functions, so these are generated once for each unique triad of atoms as described in Algorithm 1. Additional computations may be avoided by evaluating each V_P, χ_μ, and χ_ν on the grid only once per triad of atoms. In other words, when evaluating $(P_i|\mu_j\nu_k)$, the quantities $V_P(x)$, $\chi_\mu(x)$, and $\chi_\nu(x)$ are all computed and stored as GPU arrays for all P_i on atom A, μ_j on atom B and ν_k on atom C to avoid duplicating computations. These arrays can then be contracted all at once in a single tensor operation as indicated in lines 11–12 of Algorithm 1. With OpenACC, the contraction on line 12 can be handled using a single *pragma* directive containing the parallel and reduction clauses. Sample OpenACC code is provided in Section S8 of the Supporting Information. As the grid and weights are generated directly on GPU, reuse of the grid benefits from the high memory bandwidth of the GPU (~600 GB/s on the 2080-Ti). Once the integrals are computed on GPU, a single data transfer step returns the integrals to CPU memory. The code for numerically computing the STO integrals is freely available on GitHub under an Apache 2.0 license with Commons Clause as noted in the Data Availability Statement.

Multi-GPU parallelization is also implemented for a single node, using OpenMP to manage the GPU processes. Each OpenMP thread is assigned a GPU, and a manager-worker scheme is used for load balancing, where the work is partitioned using sets of atoms to take advantage of grid/weight reuse. The parallelization occurs over the loop in Line 2 of Algorithm 1 and can be accomplished with a single *pragma* directive, shown in Line 1.

TABLE 3 HF energies computed for several small molecules are listed

Molecule	DZP (32)	DZP (64)	6-31G*
CH_4	−40.199728	−40.199730	−40.194806
C_2H_6	−79.232585	−79.232593	−79.227194
C_3H_8	−118.269381	−118.269391	−118.261168
C_4H_{10}	−157.305952	−157.305972	−157.294705
C_5H_{12}	−196.342295	−196.342322	−196.328158
BH_3	−26.395615	−26.395617	−26.390665
BF_3	−323.166750	−323.166775	−323.142633
CF_4	−435.667561	−435.667608	−435.642948
$Cr(CO)_6$	−1714.832816	−1714.832901	−1714.469310

Note: The STO basis sets and grid used are described in the computational details. The numbers in parenthesis in the header denotes the bits of precision used for integral evaluation.

2.4 | Computational details

An all-electron double-zeta STO basis set with polarization functions[58] was used (denoted DZP) as the primary AO basis. The auxiliary basis sets were taken from the same source. Full specification for the primary and auxiliary basis sets are provided in Section S3 of the Supporting Information. Unless otherwise specified, the integration grid was a direct product of 60 radial points and 770 angular points (Lebedev order 18). HF and heat-bath configuration interaction

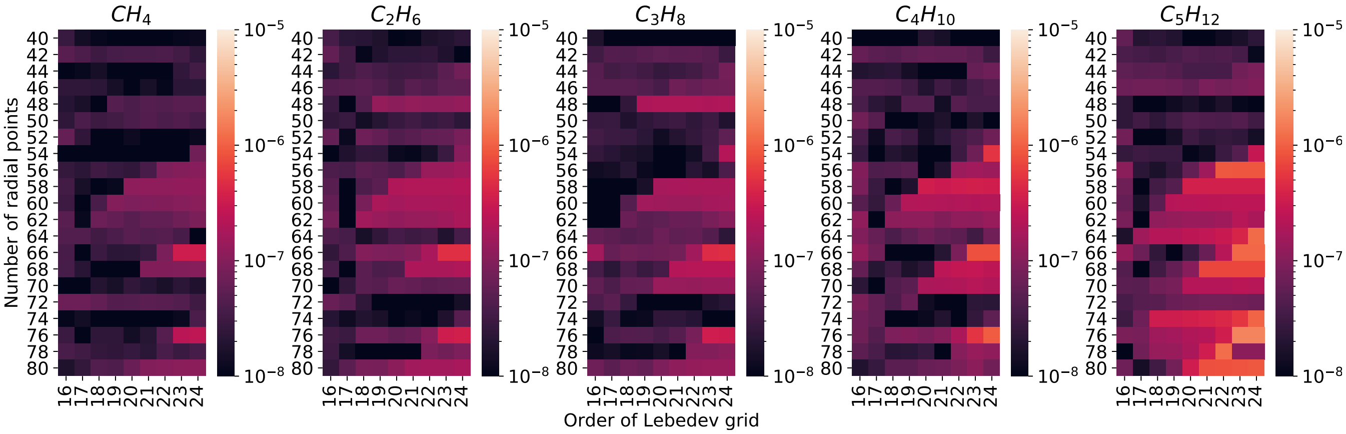

FIGURE 6 Heatmaps of the relative error of HF energies when using mixed- versus double-precision integral evaluation are shown for various alkanes using different angular and radial grid sizes

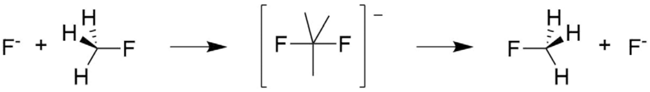

FIGURE 7 The S_N2 reaction for fluoride exchange of fluoromethane

(HBCI)[30,59–62] were used as representative electronic structure methods. The HBCI parameters used are detailed in the following section. For GTOs, the 6-31G* basis with the RI-cc-pVTZ auxiliary basis was used. All GTO integral evaluation was performed using the Libcint library.[63] Molecules were placed in standard nuclear orientations.[64] The Nvidia HPC SDK 20.7 compiler suite with CUDA 11.0 was used to compile all code. CPU code was run on Intel Xeon Gold 6242 processors clocked at 2.8 GHz and GPU code was executed using the Nvidia RTX 2080-Ti and GV-100 GPUs.

3 | RESULTS AND DISCUSSION

3.1 | Performance analysis

High throughput integral evaluation is necessary for any electronic structure theory code, regardless of basis set type. Grid-based numerical integration, however, requires orders of magnitude more floating point operations than analytical integration. To achieve the integral performance required, GPUs are used in this study for numerical integration of STO integrals. These integrals include all of the common 2-center integrals (overlap, electron-nuclear attraction, and kinetic energy) as well as the 2- and 3-center Coulomb integrals needed for the RI approximation. The relative speedup for numerical GPU integration compared with CPU integration is visualized in Figure 2. In double precision, the 2080-Ti can achieve over 30× speedup and the GV100 achieves ∼70× speedup, allowing for tractable wall times for the integrals as listed in Table 1. Even further performance can be gained by utilizing mixed precision, showing speedups of over 60× and overall integral throughput increasing by a factor of ∼4 for the 2080-Ti (see Table 2). The speedup relative to CPU drops slightly to ∼55× on the GV100. The performance behavior is a consequence of the hardware configuration and the integral kernels being compute bound, with a detailed analysis provided in the Supporting Information (Section S5).

Faster integral evaluation is also possible by distributing the workload across multiple GPUs. To test multi-GPU scaling, 3-center ERIs were evaluated for C_9H_{20}, which has 76.9 million ERIs, taking 125 s to compute in mixed precision and 542 s in double-precision. Figure 3 shows the strong scaling performance when evaluating the 3-center ERIs for C_9H_{20}, which maintains parallel efficiency greater than 75% on up to five GPUs for mixed-precision evaluation and greater than 90% for double-precision evaluations in this benchmark. Due to the reduced computational demand of mixed-precision integration, the serial components and communication overhead take up a proportionally larger amount of computational time. Consequently, the parallel efficiency for mixed-precision integral evaluation drops off more rapidly than for double precision in strong scaling tests. However, this parallelization scheme still allows STO integration to achieve greater than 75% parallel efficiency and overall integral throughput greater than 3 million integrals per second in mixed precision on five GPUs. For comparison, Sun reported a throughput of approximately 6–8 million explicitly calculated integrals per second per thread with Libcint,[63] thus placing STO integral throughput within reach of analytical integral evaluation for GTOs. While the Libcint performance was reported for 4-center ERIs, the comparison demonstrates the feasibility of Slater integration under the RI approximation. Additional developments in code optimization[65] and screening protocols[55] may narrow this gap further. Other grid-based STO integral evaluation implementations[13,25–29] do not report timings nor do they use GPU acceleration. The closest available performance comparison is an example where a 9-Gaussian expansion was used to approximate STO integrals,[18] which would reduce throughput, relative to GTOs, by approximately a factor of 700 under the RI approximation.

TABLE 4 HF and HBCI activation energies (kcal/ mol) of CH_3F fluoride exchange at various grid sizes using single- and double-precision integral evaluations

	Double precision			Mixed precision		
Angular Radial	17 (590)	18 (770)	19 (974)	17 (590)	18 (770)	19 (974)
			HF			
50	18.4	18.4	18.4	18.4	18.4	18.4
60	18.4	18.4	18.4	18.4	18.4	18.4
			HBCI			
50	13.7	13.6	13.8	13.6	13.8	13.7
60	13.9	13.8	13.7	13.8	13.6	13.6

Note: The number of radial points and Lebedev order are provided for the radial and angular grids. The size of the angular grid is given in parenthesis next to the Lebedev order.

TABLE 5 Relative energies of cyclobutadiene at D_{2h} and D_{4h} geometries (kcal/mol)

	Double precision		Mixed precision	
	D_{2h}	D_{4h}	D_{2h}	D_{4h}
Singlet	0.0	9.4	0.0	9.3
Triplet	36.2	14.2	36.2	14.2
Gap	36.2	4.8	36.2	4.9

The large performance gain (~4× speedup) when using mixed precision on the 2080-Ti units necessarily comes with some loss in accuracy compared with double precision arithmetic. Therefore tests of the mixed-precision integral evaluation are needed, in order to gauge the quantitative tradeoff between accuracy and speed.

3.2 | Mixed-precision evaluation

Numerical evaluation of integrals, whether done in single, double, or mixed precision, will necessarily contain some residual error with any finite grid. While this is expected with grid-based integration, estimates of the error and smoothness of the resulting integrals are necessary to test the accuracy of the procedure. First, a selection of 2-center ERIs were evaluated to determine the relative loss in precision when using the mixed-precision procedure. For each $(P|Q)$, the center Q was scanned radially away from center P in 16 directions corresponding to all-positive vectors of an 86-point Lebedev grid. Figure 4 plots the max and average absolute errors between mixed- and double-precision integrals at each distance. This indicates that the error of individual integrals are similarly sized across various distances, with the errors all being less than 8×10^{-5}.

The next measure of performance for mixed-precision integration is to evaluate the smoothness of the integrals with respect to changes in nuclear position. Therefore 2-center ERIs were evaluated in mixed precision as center Q is scanned radially for the same 16 directions as before. These yield qualitatively smooth plots, as seen in Figure 5. Additional plots for other basis set pairings are provided in Section S6 of the Supporting Information and show the same qualitative behavior as this figure.

3.3 | Hartree–Fock and HBCI

Two levels of wave function-based electronic structure theory were selected to provide practical tests for the Slater GPU integrals. First, the HF energies for a set of benchmark molecules were computed and these are listed in Table 3 (see Table S2 of the Supporting Information for timing information). The DZP basis set, corresponding auxiliary basis sets, and grid described in the Computational Details were used for these tests. Energies using the 6-31G* and RI-cc-pVTZ auxiliary GTO basis sets are also reported, to provide a baseline for comparison. The HF results for alkanes (C_nH_{2n+2}) show a slight increase in the mixed-precision error as the chain length increases. This is shown in Figure 6, which depicts the relative error of the HF energy when using mixed- and double-precision at various grid sizes. The roughly constant relative error as system size grows suggests that the mixed-precision error is size extensive. Combined with Figures 4 and 5, Figure 6 indicates that errors due to using mixed-precision integrals may largely result in error cancellation.

The small error margins for STO integrals—as measured at the HF level of theory—suggest that thermochemical properties can be precisely evaluated. To test this hypothesis, an S_N2 reaction involving fluoride exchange in fluoromethane was evaluated (Figure 7). Since the HF level of theory is not expected to be quantitative, activation energies were computed not only with HF, but also with the HBCI method, with ε_1 set to 1.0 mHa and ε_2 set to 1.0 μHa. HBCI provides a close approximation to full CI, and importantly, is tractable for the 20e$^-$ in 60 orbital system of interest here. The activation energies of the exchange reaction using various grids are reported in Table 4. At the grids considered, the change in activation energy at the HF level is negligible between mixed and double precision as well as between grid sizes. At the HBCI level, more integrals contribute to the total energy. Consequently, the variation in the activation energy is larger for HBCI relative to HF. However, the range of activation barriers for HBCI is still less than half a kcal/mol.

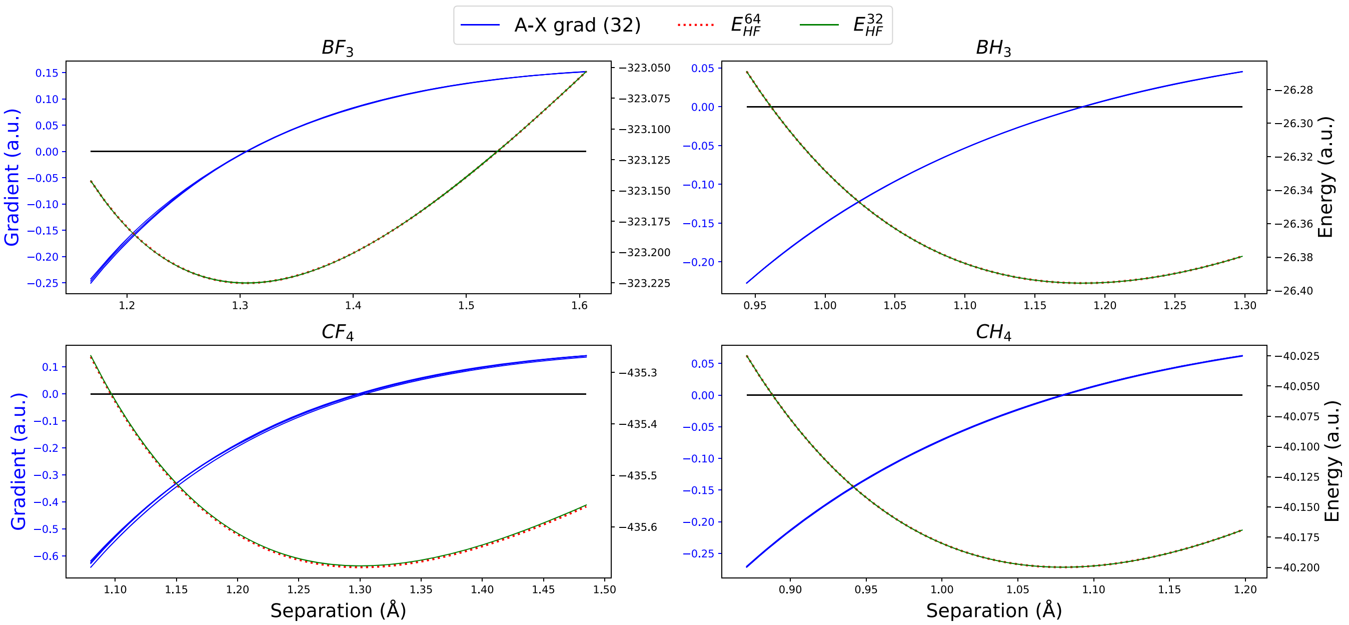

FIGURE 8 The Hartree–Fock geometric gradient projections (solid blue lines) of molecules with D_{3h} and T_d point groups are plotted as the A–X bond distance is scanned, where A = B,C and X = H,F. Gradients were computed in mixed precision. The mixed-precision (solid green) and double-precision (dotted red) Hartree–Fock energies at each point are also plotted. For CF_4, the auxiliary basis for fluorine is extended with additional 2*p*, 3*d*, 4*f*, and 5*g* functions

Another test using the HBCI method was the calculation of the singlet-triplet gaps of cyclobutadiene at its D_{2h} and D_{4h} geometries using a triple-ζ polarized basis set. Cyclobutadiene has a multireference singlet ground state, due to its degenerate π orbitals in the D_{4h} geometry. The results, for HBCI parameters of $\varepsilon_1 = 1.0$ mHa and $\varepsilon_2 = 0.1$ μHa, are given in Table 5. These demonstrate mixed-precision errors of less than 0.1 kcal/mol. Furthermore, the singlet-triplet gap at the D_{4h} is consistent with prior full-CI using GTOs.[31]

One final test will further show the utility of the Slater GPU integrals in quantum chemistry. Specifically, the geometric gradients—which are essential in studying chemical reactions—were evaluated using analytical nuclear derivatives of the quantities V_P, μ, and ν in Equation (13). As a benchmark, the fully symmetric BF_3, BH_3, CF_4, and CH_4 molecules were symmetrically stretched. The HF energies as well as the projection of the mixed-precision HF geometric gradient onto each A–X bond (A = B,C; X = H,F) are plotted in Figure 8. As before, the mixed- and double-precision energies overlap with one another. As for the gradient, the magnitude along each A–X bond should be identical for all distances. This is largely achieved in these test cases, however, there is some variation when fluorine is present. For CF_4, using the ADF fitting basis led to large gradient errors, thus the auxiliary basis of fluorine was extended with additional functions (see Section S3 of the Supporting Information for additional details). Since this addition resulted in substantially improved gradients, the remaining variations for BF_3 and CF_4 are attributed to an incomplete RI auxiliary basis. While this work has not examined the choice of RI basis in detail, this subject will need to be revisited in a future study.

4 | CONCLUSIONS

The SlaterGPU integral code is herein shown capable of evaluating the full complement of ERIs needed for HF and post-HF theories. Modern computer architectures combined with the RI approximation have allowed STO integrals to be feasible even though analytic expressions are currently unavailable. The use of mixed-precision integration allows further performance gains—achieving speedups greater than 80× for the ERIs—with minimal loss to accuracy. In the future, computing select integrals in double precision may mitigate errors due to using mixed-precision integrals. The combination of GPU acceleration, multi-GPU parallelization, and mixed-precision integration make SlaterGPU competitive with single-threaded GTO integration with the possibility of tuning SlaterGPU for additional performance.

The current implementation and basis sets are adequate for performing correlated electronic structure computations at the full CI level, however room for improvement remains in the STO RI gradients, where the available auxiliary basis sets appear to be inadequate. Further development of auxiliary basis sets will be required before STO integrals are generally useful for gradient computations.

ACKNOWLEDGMENTS

The authors acknowledge David Braun for loaning a GV100 for testing. Kris Keipert and Marc West of Nvidia were instrumental in assisting the authors with OpenACC. This work is supported by a DOE CCS award, DE-SC0022241. Duy-Khoi Dang thanks the National Science Foundation for providing a Graduate Research Fellowship under grant number DGE 1841052.

DATA AVAILABILITY STATEMENT

Basis sets and geometries are available in the Supporting Information. The SlaterGPU integral library is available on Github (https://github.com/ZimmermanGroup/SlaterGPU) under an Apache 2.0 License with Commons Clause.

ORCID

Duy-Khoi Dang https://orcid.org/0000-0001-7530-0540
Leighton W. Wilson https://orcid.org/0000-0003-1676-8156
Paul M. Zimmerman https://orcid.org/0000-0002-7444-1314

REFERENCES

[1] T. Helgaker, P. Jørgensen, J. Olsen, *Molecular Electronic Structure Theory*, John Wiley & Sons, Ltd, England **2000**.
[2] J. C. Slater, *Phys. Rev.* **1928**, *31*, 333.
[3] A. J. Cohen, N. C. Handy, *J. Chem. Phys.* **2002**, *117*, 1470v1478.
[4] T. Kato, *Commun. Pure Appl. Math.* **1957**, *10*, 151.
[5] P. Reinhardt, P. E. Hoggan, *Int. J. Quantum Chem.* **2009**, *109*, 3191.
[6] S. F. Boys, A. C. Egerton, *Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci.* **1950**, *200*, 542.
[7] L. E. McMurchie, E. R. Davidson, *J. Comput. Phys.* **1978**, *26*, 218.
[8] S. Obara, A. Saika, *J. Chem. Phys.* **1986**, *84*, 3963.
[9] S. Obara, A. Saika, *J. Chem. Phys.* **1988**, *89*, 1540.
[10] M. Head-Gordon, J. A. Pople, *J. Chem. Phys.* **1988**, *89*, 5777.
[11] P. M. W. Gill, M. Head-Gordon, J. A. Pople, *J. Phys. Chem.* **1990**, *94*, 5564.
[12] B. Kanungo, P. M. Zimmerman, V. Gavini, *Nat. Commun.* **2019**, *10*, 4497.
[13] M. A. Watson, N. C. Handy, A. J. Cohen, T. Helgaker, *J. Chem. Phys.* **2004**, *120*, 7252.
[14] P. E. Hoggan, *Int. J. Quantum Chem.* **2004**, *100*, 214.
[15] P. W. Ayers, R. C. Morrison, R. G. Parr, *Mol. Phys.* **2005**, *103*, 2061.
[16] W. J. Hehre, R. F. Stewart, J. A. Pople, *J. Chem. Phys.* **1969**, *51*, 2657.
[17] W. J. Hehre, R. Ditchfield, R. F. Stewart, J. A. Pople, *J. Chem. Phys.* **1970**, *52*, 2769.
[18] J. Fernández Rico, R. López, A. Aguado, I. Ema, G. Ramírez, *Int. J. Quantum Chem.* **2001**, *81*, 148.
[19] M. Caffarel, *J. Chem. Phys.* **2019**, *151*, 064101.
[20] M. Nightingale, C. Umrigar, *Quantum Monte Carlo Methods in Physics and Chemistry*, Kluwer Academic Publishers, Boston, MA **1999**.
[21] S. Zhang, H. Krakauer, *Phys. Rev. Lett.* **2003**, *90*, 136401.
[22] P. M. Zimmerman, J. Toulouse, Z. Zhang, C. B. Musgrave, C. J. Umrigar, *J. Chem. Phys.* **2009**, *131*, 124103.
[23] B. M. Austin, D. Y. Zubarev, W. A. Lester, *Chem. Rev.* **2012**, *112*, 263.
[24] W. Koch, M. C. Holthausen, *A Chemist's Guide to Density Functional Theory*, 2nd ed., Wiley - VCH, Weinheim, New York **2001**.
[25] M. A. Watson, N. C. Handy, A. J. Cohen, *J. Chem. Phys.* **2003**, *119*, 6475.
[26] A. Förster, M. Franchini, E. van Lenthe, L. Visscher, *J. Chem. Theory Comput.* **2020**, *16*, 875.
[27] A. Förster, L. Visscher, *J. Comput. Chem.* **2020**, *41*, 1660.
[28] A. Förster, L. Visscher, *J. Chem. Theory Comput.* **2020**, *16*, 7381.
[29] A. Förster, L. Visscher, *J. Chem. Theory Comput.* **2021**, *17*, 5080.
[30] A. A. Holmes, N. M. Tubman, C. J. Umrigar, *J. Chem. Theory Comput.* **2016**, *12*, 3674.
[31] P. M. Zimmerman, *J. Phys. Chem. A* **2017**, *121*, 4712.
[32] P. M. Zimmerman, *J. Chem. Phys.* **2017**, *146*, 104102.
[33] P. M. Zimmerman, *J. Chem. Phys.* **2017**, *146*, 224104.
[34] B. O. Roos, *Int. J. Quantum Chem.* **1980**, *18*, 175.
[35] B. O. Roos, P. R. Taylor, P. E. Sigbahn, *Chem. Phys.* **1980**, *48*, 157.
[36] P. M. Zimmerman, A. E. Rask, *J. Chem. Phys.* **2019**, *150*, 244117.
[37] D.-K. Dang, P. M. Zimmerman, *J. Chem. Phys.* **2021**, *154*, 014105.
[38] J. F. Stanton, R. J. Bartlett, *J. Chem. Phys.* **1993**, *98*, 7029.
[39] J. Friedrich, M. Hanrath, M. Dolg, *J. Chem. Phys.* **2007**, *126*, 154110.
[40] F. A. Evangelista, E. Prochnow, J. Gauss, H. F. Schaefer, *J. Chem. Phys.* **2010**, *132*, 074107.
[41] F. A. Evangelista, J. Gauss, *J. Chem. Phys.* **2011**, *134*, 114102.
[42] SlaterGPU. https://github.com/ZimmermanGroup/SlaterGPU.
[43] B. I. Dunlap, J. W. D. Connolly, J. R. Sabin, *J. Chem. Phys.* **1979**, *71*, 3396.
[44] H.-J. Werner, F. R. Manby, P. J. Knowles, *J. Chem. Phys.* **2003**, *118*, 8149.
[45] R. A. Distasio Jr., R. P. Steele, Y. M. Rhee, Y. Shao, M. Head-Gordon, *J. Comput. Chem.* **2007**, *28*, 839.
[46] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, USA **2007**.
[47] P. M. Boerrigter, G. Te Velde, J. E. Baerends, *Int. J. Quantum Chem.* **1988**, *33*, 87.
[48] A. D. Becke, *J. Chem. Phys.* **1988**, *88*, 2547.
[49] C. W. Murray, N. C. Handy, G. J. Laming, *Mol. Phys.* **1993**, *78*, 997.
[50] O. Treutler, R. Ahlrichs, *J. Chem. Phys.* **1995**, *102*, 346.
[51] M. E. Mura, P. J. Knowles, *J. Chem. Phys.* **1996**, *104*, 9848.
[52] V. Lebedev, *USSR Comput. Math. Math. Phys.* **1976**, *16*, 10.
[53] *Implementations of the gamma functions.* http://www.netlib.org/cephes/.
[54] I. S. Ufimtsev, T. J. Martinez, *J. Chem. Theory Comput.* **2009**, *5*, 1004.
[55] N. Luehr, I. S. Ufimtsev, T. J. Martvnez, *J. Chem. Theory Comput.* **2011**, *7*, 949.
[56] P. Pokhilko, E. Epifanovsky, A. I. Krylov, *J. Chem. Theory Comput.* **2018**, *14*, 4088.
[57] S. Seritan, C. Bannwarth, B. S. Fales, E. G. Hohenstein, S. I. L. Kokkila-Schumacher, N. Luehr, J. W. Snyder, C. Song, A. V. Titov, I. S. Ufimtsev, T. J. Martínez, *J. Chem. Phys.* **2020**, *152*, 224110.
[58] E. Van Lenthe, E. J. Baerends, *J. Comput. Chem.* **2003**, *24*, 1142.
[59] J. E. T. Smith, B. Mussard, A. A. Holmes, S. Sharma, *J. Chem. Theory Comput.* **2017**, *13*, 5468.
[60] A. D. Chien, A. A. Holmes, M. Otten, C. J. Umrigar, S. Sharma, P. M. Zimmerman, *J. Phys. Chem. A* **2018**, *122*, 2714.
[61] J. Li, M. Otten, A. A. Holmes, S. Sharma, C. J. Umrigar, *J. Chem. Phys.* **2018**, *149*, 214110.
[62] K. R. Brorsen, *J. Chem. Theory Comput.* **2020**, *16*, 2379.
[63] Q. Sun, *J. Comput. Chem.* **2015**, *36*, 1664.
[64] P. M. Gill, B. G. Johnson, J. A. Pople, *Chem. Phys. Lett.* **1993**, *209*, 506.
[65] M. Khalilov, A. Timoveev, *J. Phys.: Conf. Ser.* **2021**, *1740*, 012056.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: D.-K. Dang, L. W. Wilson, P. M. Zimmerman, *J. Comput. Chem.* **2022**, *43*(25), 1680. https://doi.org/10.1002/jcc.26968