Title

TeraChem: Accelerating electronic structure and ab initio molecular dynamics with graphical processing units

Permalink

https://escholarship.org/uc/item/07g93240

Journal

The Journal of Chemical Physics, 152(22)

ISSN

0021-9606

Authors

Seritan, Stefan
Bannwarth, Christoph
Fales, B Scott
et al.

Publication Date

2020-06-14

DOI

10.1063/5.0007615

Peer reviewed

TeraChem: Accelerating electronic structure and *ab initio* molecular dynamics with graphical processing units

Cite as: J. Chem. Phys. **152**, 224110 (2020); doi: 10.1063/5.0007615
Submitted: 18 March 2020 • Accepted: 19 May 2020 •
Published Online: 12 June 2020

Stefan Seritan,[1,2] Christoph Bannwarth,[1,2] B. Scott Fales,[1,2] Edward G. Hohenstein,[1,2] Sara I. L. Kokkila-Schumacher,[3] Nathan Luehr,[4] James W. Snyder, Jr.,[5] Chenchen Song,[6,7] Alexey V. Titov,[8] Ivan S. Ufimtsev,[9] and Todd J. Martínez[1,2,a)]

AFFILIATIONS

[1] Department of Chemistry and The PULSE Institute, Stanford University, Stanford, California 94305, USA
[2] SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, USA
[3] IBM, Thomas J. Watson Research Center, Yorktown Heights, New York 10598, USA
[4] NVIDIA, Santa Clara, California 95051, USA
[5] Adobe, San Jose, California 95110, USA
[6] Department of Physics, University of California Berkeley, Berkeley, California 94720, USA
[7] Molecular Foundry, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA
[8] Intel Corporation, Santa Clara, California 95054, USA
[9] Department of Structural Biology, Stanford University School of Medicine, Stanford, California 94305, USA

Note: This article is part of the JCP Special Topic on Electronic Structure Software.
[a)] **Author to whom correspondence should be addressed:** toddjmartinez@gmail.com

ABSTRACT

Developed over the past decade, TeraChem is an electronic structure and *ab initio* molecular dynamics software package designed from the ground up to leverage graphics processing units (GPUs) to perform large-scale ground and excited state quantum chemistry calculations in the gas and the condensed phase. TeraChem's speed stems from the reformulation of conventional electronic structure theories in terms of a set of individually optimized high-performance electronic structure operations (e.g., Coulomb and exchange matrix builds, one- and two-particle density matrix builds) and rank-reduction techniques (e.g., tensor hypercontraction). Recent efforts have encapsulated these core operations and provided language-agnostic interfaces. This greatly increases the accessibility and flexibility of TeraChem as a platform to develop new electronic structure methods on GPUs and provides clear optimization targets for emerging parallel computing architectures.

I. INTRODUCTION

Over the years, a wide variety of different electronic structure packages have become available, each with their own specific implementations and target applications. Practicing good design principles, such as code encapsulation and well-defined interfaces, enables quick prototyping for new methods and allows developers to focus on algorithmic advancements without disrupting downstream workflows. These principles remain important in today's scientific computing environment, where high-level languages such as Python are used as "glue" to enable interoperability between modules of one package or between several electronic structure codes. For example, Psi4[1] provides C++ and Python interfaces to their libraries and driver-level code and has even recently provided a set of reference electronic structure implementations leveraging a combination of Psi4 and standard Python libraries such as NumPy.[2] Meanwhile, lower-level offerings, such as LibInt,[3] LibXC,[4] and XCFun,[5] provide standard interfaces to Gaussian integrals and exchange–correlation (XC) functionals for density functional theory (DFT) and have seen some adoption throughout the community. In addition to the benefits already mentioned, well-chosen and physically motivated interfaces can greatly influence algorithmic development for new methods.

Encapsulation and clear interfaces are also critical when considering the use of hardware accelerators such as graphics processing units (GPUs). Originally designed for rendering pipelines, the GPU architecture is well-suited to take advantage of single-instruction multiple-data (SIMD) parallelism. Typical design principles for achieving high throughput on GPUs involve pipelining expensive central processing unit (CPU)–GPU memory transfers and minimizing idle threads in warps (groups of threads that execute instructions simultaneously), although recent advances in GPU technology have decreased the performance penalty for warp divergence and provided mechanisms to increase CPU–GPU memory bandwidth. GPUs typically have stronger single precision performance, particularly consumer-grade (e.g., NVIDIA GeForce GTX) cards that do not have the double precision hardware support available as in scientific-grade (e.g., NVIDIA Tesla) cards. It is hard to redesign general electronic structure algorithms for many methods with these constraints in mind, so a better approach is to build several highly tuned operations that can be used in many methods throughout the codebase.

The TeraChem software package was developed in 2008, following the release of NVIDIA's Compute Unified Device Architecture[6] (CUDA) toolchain for general-purpose GPU computing in 2007. The initial goal was to use GPUs to accelerate the evaluation of the electron repulsion integrals (ERIs),[7–10] which form the computational bottleneck in many electronic structure theories and have been the target of many efforts in the community.[11–19] This was most efficiently implemented in terms of Coulomb and exchange matrix builds, where contractions of the density matrix with the ERIs were computed directly. As the codebase grew, the most expedient route for implementing new methods was to reuse these encapsulated GPU-accelerated operations. The development of TeraChem's configuration-interaction (CI) library[20] directly followed this pattern, where one- and two-particle density matrix construction and other rate-limiting matrix–vector operations were accelerated on GPUs and then used as algorithmic building blocks for novel method development. This approach is successful as the energy of any non-DFT Hamiltonian system can be written as a summation of one- and two-electron operators contracted with the corresponding density matrices; hence, one should focus on obtaining density matrices and their operator expressions within the appropriate basis in a computationally feasible manner. For self-consistent field (SCF) methods, density matrices are straightforward to obtain through the diagonalization of the Fock matrix, but the operator expressions (i.e., contraction against the ERIs) are challenging. In CI approaches, getting the density matrices (or underlying wavefunctions) is already difficult, and therefore, significant effort has been expended to efficiently solve for the wavefunction and density matrices.

Together, these advances provide TeraChem with electronic structure methods that use GPUs effectively and allow for efficient on-the-fly energy and gradient evaluations, which form the basis of many quantum chemistry workflows. TeraChem has been used to perform *ab initio* molecular dynamics (AIMD) and optimizations on entire proteins[21,22] and to provide real-time interactive AIMD[23] on desktop workstations. The *ab initio* nanoreactor framework[24,25] combines accelerated AIMD with geometry and minimum energy path optimizations to perform automated reaction network discovery. An interface to the FMS90 package facilitates *ab initio* multiple spawning[26] (AIMS) simulations, which have been used to study photochemical processes in the gas phase and protein environments.[27–33] Nonadiabatic dynamics for multichromophoric light-harvesting assemblies are enabled through the *ab initio* exciton model.[34–36] A new socket-based interface for TeraChem based on Google's Protocol Buffers[37] has been developed to facilitate GPU acceleration of these higher-level quantum chemistry workflows and has now been tested with a large variety of applications.

Here, we present the current modular structure of the TeraChem software package, as shown in Fig. 1. First, we describe interfaces and key GPU implementation details of the five main libraries: IntBox, which provides direct atomic orbital (AO) integral evaluation routines; SQMBox, effectively a semiempirical Mulliken-approximated equivalent for IntBox; THCBox, which uses tensor hypercontraction (THC) to leverage rank-sparsity in the ERIs for alternate contraction schemes; CIBox, which provides efficient routines for determining CI wavefunctions; and CCBox, which provides coupled-cluster methods. Throughout the discussion of the various libraries, we also highlight the rise of several encapsulated GPU-accelerated operations that have become core algorithmic building blocks for most electronic structure methods in TeraChem. Next, we provide a short summary of our new TeraChem Protocol Buffer (TCPB) server interface. Finally, we conclude with a discussion of rapid electronic structure method development in the context of the aforementioned core electronic structure operations; in particular, we emphasize how encapsulation and well-defined interfaces to these operations provide a path forward for leveraging emerging parallel architectures without the need to redesign each electronic structure method implementation.

II. INTBOX

IntBox is a library for direct one- and two-electron integral evaluation for contracted atom-centered Gaussian basis sets, currently applicable for *spd* angular momenta. The IntBox interface, shown schematically in Fig. 2, is exposed as pure C for portability and simplicity. Users provide a Gaussian-type orbital (GTO) basis set, nuclear charge and position, and a generalized density matrix as inputs. IntBox then provides the following quantities, described in greater detail below, as outputs: one-electron (i.e., overlap, kinetic energy, and nuclear attraction) integrals, two-electron quantities such as Coulomb and exchange matrices, effective core potentials, spin–orbit couplings, and all corresponding gradient contributions. Overlap and kinetic energy integrals are performed solely on the CPU, while the remaining quantities are accelerated through the use of GPUs.

Each contracted GTO can be expressed as a linear combination of atom-centered primitive Gaussians,

$$\phi_\mu(\mathbf{r}) = \sum_k c_{\mu k}\chi_k(\mathbf{r}), \tag{1}$$

where μ, ν, λ, and σ index contracted atomic orbitals (AOs). Most electronic structure methods rely on several one-electron integrals,

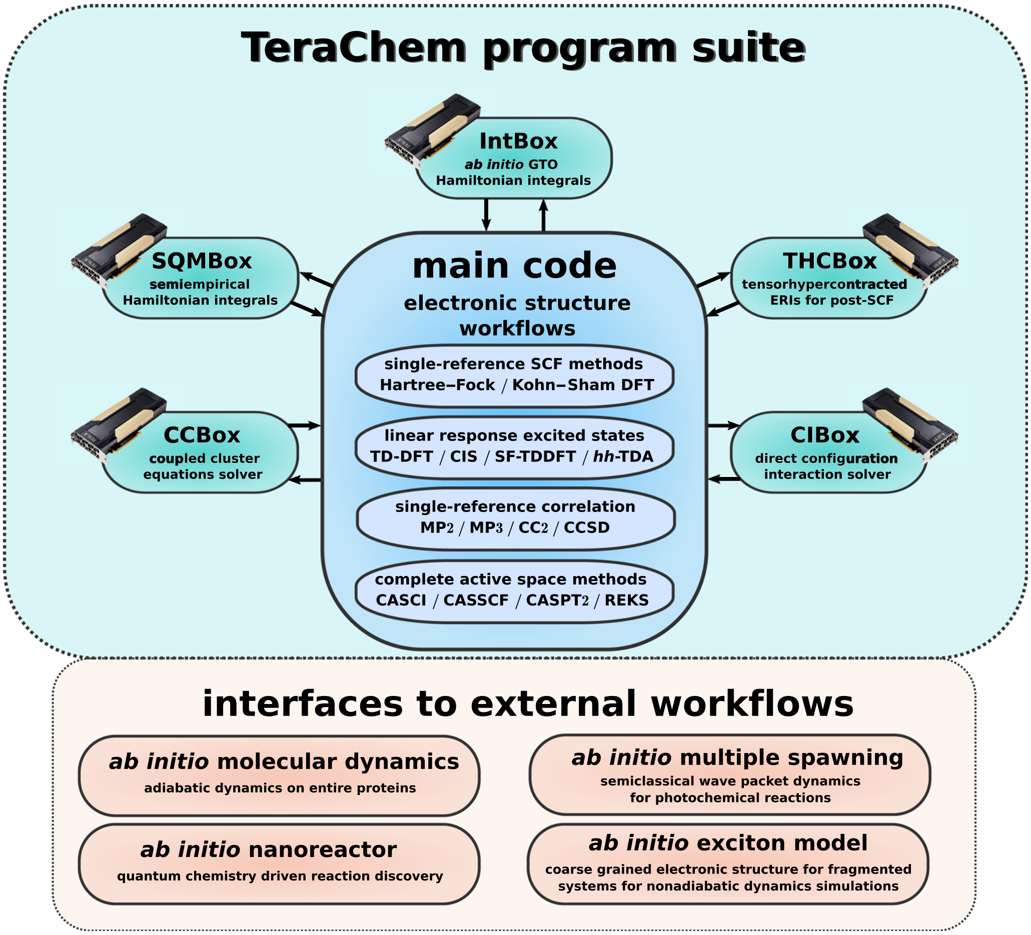

TeraChem program suite
IntBox
ab initio GTO Hamiltonian integrals
SQMBox
semiempirical Hamiltonian integrals
THCBox
tensorhypercontracted ERIs for post-SCF
main code
electronic structure workflows
single-reference SCF methods
Hartree–Fock / Kohn–Sham DFT
CCBox
coupled cluster equations solver
linear response excited states
TD-DFT / CIS / SF-TDDFT / hh-TDA
CIBox
direct configuration interaction solver
single-reference correlation
MP2 / MP3 / CC2 / CCSD
complete active space methods
CASCI / CASSCF / CASPT2 / REKS
interfaces to external workflows
ab initio molecular dynamics
adiabatic dynamics on entire proteins
ab initio multiple spawning
semiclassical wave packet dynamics for photochemical reactions
ab initio nanoreactor
quantum chemistry driven reaction discovery
ab initio exciton model
coarse grained electronic structure for fragmented systems for nonadiabatic dynamics simulations

FIG. 1. Schematic setup of the modular structure within the TeraChem program suite. The separate "boxes" represent individually optimized libraries to compute the essential quantities that are used in the electronic structure routines inside the main code. Furthermore, TeraChem provides potential energy surfaces (i.e., energies and gradients) for external higher-level workflows through the new server interface.

namely, the overlap, kinetic energy, and nuclear-attraction integrals, given as

$$S_{\mu\nu} = (\phi_\mu|\phi_\nu), \tag{2}$$

$$T_{\mu\nu} = \left(\phi_\mu\middle|-\frac{\nabla^2}{2}\middle|\phi_\nu\right), \tag{3}$$

$$V_{\mu\nu} = \left(\phi_\mu\middle|-\sum_A \frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|}\middle|\phi_\nu\right), \tag{4}$$

where A indexes atoms, while $\mathbf{r}$ and $\mathbf{R}$ are electronic and nuclear coordinates, respectively. The two-electron ERIs for the contracted GTOs are given as

$$(\mu_i\nu_j|\lambda_k\sigma_l) = c_{\mu i}c_{\nu j}c_{\lambda k}c_{\sigma l}(\chi_i\chi_j|\chi_k\chi_l), \tag{5}$$

$$(\mu\nu|\lambda\sigma) = \sum_{ijkl}(\mu_i\nu_j|\lambda_k\sigma_l), \tag{6}$$

where μ_i denotes the ith primitive corresponding to the μth contracted basis function, and the primitive integrals $(\chi_i\chi_j|\chi_k\chi_l)$ in Eq. (5) can be computed analytically.[38,39] The ERIs rarely appear in isolation; rather, many terms include the ERIs contracted with an arbitrary density matrix $D_{\lambda\sigma}$, as in the Coulomb and exchange matrices,

$$J_{\mu\nu} = \sum_{\lambda\sigma}(\mu\nu|\lambda\sigma)D_{\lambda\sigma}, \tag{7}$$

$$K_{\mu\nu} = \sum_{\lambda\sigma}(\mu\lambda|\nu\sigma)D_{\lambda\sigma}. \tag{8}$$

Since ERI evaluation is often the computational bottleneck in electronic structure codes, significant effort has been expended to use GPUs to accelerate these **J** (i.e., Coulomb) and **K** (i.e., exchange) builds from Eqs. (7) and (8) in IntBox.[8–10] The two-electron primitive integrals are presorted into batches on the CPU by angular momentum and density-weighted Schwarz bound,[40]

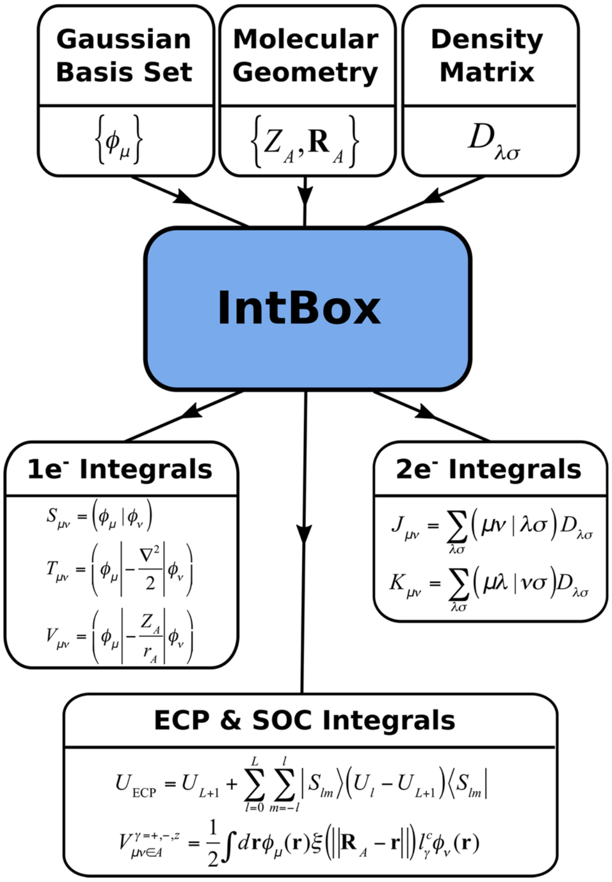

Gaussian Basis Set
$\{\phi_\mu\}$
Molecular Geometry
$\{Z_A, \mathbf{R}_A\}$
Density Matrix
$D_{\lambda\sigma}$
IntBox
1e- Integrals
$S_{\mu\nu} = (\phi_\mu | \phi_\nu)$
$T_{\mu\nu} = \left(\phi_\mu \middle| -\frac{\nabla^2}{2} \middle| \phi_\nu\right)$
$V_{\mu\nu} = \left(\phi_\mu \middle| -\frac{Z_A}{r_A} \middle| \phi_\nu\right)$
2e- Integrals
$J_{\mu\nu} = \sum_{\lambda\sigma} (\mu\nu | \lambda\sigma) D_{\lambda\sigma}$
$K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda | \nu\sigma) D_{\lambda\sigma}$
ECP & SOC Integrals
$U_{\text{ECP}} = U_{L+1} + \sum_{l=0}^{L} \sum_{m=-l}^{l} |S_{lm}\rangle (U_l - U_{L+1}) \langle S_{lm}|$
$V_{\mu\nu\in A}^{\gamma=+,-,z} = \frac{1}{2} \int d\mathbf{r} \phi_\mu(\mathbf{r}) \xi(\|\mathbf{R}_A - \mathbf{r}\|) l_\gamma^c \phi_\nu(\mathbf{r})$

FIG. 2. Schematic of the IntBox interface, which takes geometry, basis set, and density information as inputs and produces various one- and two-electron integral contractions. Gradient contributions for each set of integrals are also available but are not shown for clarity.

$$|(\mu_i\nu_j|\lambda_k\sigma_l)D_{\lambda\sigma}| \le \sqrt{(\mu_i\nu_j|\mu_i\nu_j)(\lambda_k\sigma_l|\lambda_k\sigma_l)}|D_{\lambda\sigma}|. \tag{9}$$

Each GPU kernel then uses the density-weighted Schwarz bound as a screening criterion, which is extremely effective when working in the AO basis where spatial sparsity can be exploited. Additionally, the presorting results in coalesced memory access and prevents thread divergence, which are all important factors in gaining high GPU performance. The early version of TeraChem screening for exchange builds neglected any integrals where the density-weighted Schwarz bound dropped below the integral threshold multiplied with a "guard" parameter.[9] In later work, this was replaced by monitoring the product of the Schwarz bound with the largest element of the corresponding density matrix block,

$$|(\mu_i\lambda_k|\nu_j\sigma_l)D_{\lambda\sigma}| \le \sqrt{(\mu_i\lambda_k|\mu_i\lambda_k)(\nu_j\sigma_l|\nu_j\sigma_l)}|D|_\infty. \tag{10}$$

Although less aggressive than the original "guard" screening procedure, this strategy still results in good scaling for **K** builds, as shown in Fig. 3. Although these operations formally scale as $O(N_{\text{AO}}^4)$, the observed scaling is often quadratic due to the exploitation of the aforementioned sparsity. The **J** builds have a smaller prefactor compared to **K** builds, but the **K** builds scale better because these interactions fall off more steeply than the Coulomb interaction.

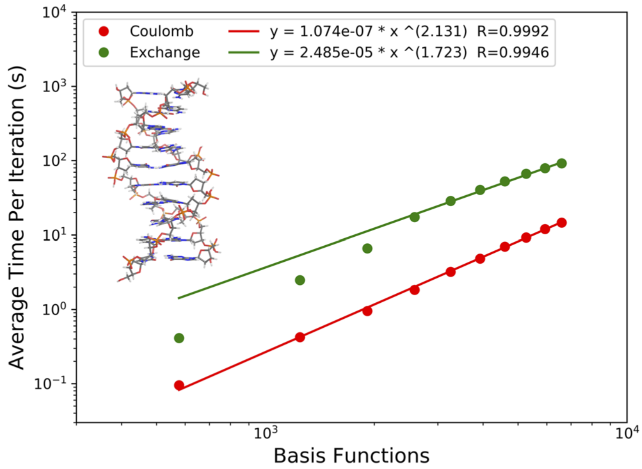


FIG. 3. Quadratic scaling behavior of Coulomb and exchange matrix builds applied to an isolated duplex of DNA (PDB ID: 1ZF7) at the RHF/6-31g* level of theory. Different system sizes were generated by sequentially adding base pair units to the DNA duplex. Conventional Fock builds were done in full double precision with all integral thresholds set to 10^{-11}. Calculations were run using a Tesla V100 GPU and a single core of a 3.4 GHz Intel Xeon E5-2643v4 CPU.

IntBox takes a tiered mixed precision strategy, where the largest integrals are computed in double precision, other significant integrals are computed in single precision, and the remainder are neglected. In the future, this strategy could also potentially be extended to half precision to take advantage of new hardware accelerators designed primarily for machine learning, such as tensor cores in GPUs. For iterative procedures such as the self-consistent field (SCF) procedure, dynamic precision schemes take this one step further by gradually tightening the single/double precision threshold over the course of the SCF.[41] Mixed and dynamic precision schemes work particularly well when combined with incremental Fock matrix builds,[42] as integral screening becomes more effective as many elements of the difference density approach zero. Through the use of automated code generation techniques, IntBox performance can be regarded as near-optimal for all available angular momenta (i.e., up to d functions)[43] and effective core potentials[44,45] on a variety of different NVIDIA GPU architectures.

The basic quantities computed by IntBox are straightforwardly used to construct the core Hamiltonian and Fock matrices in self-consistent field (SCF) procedures,

$$H_{\mu\nu}^{\text{core}} = T_{\mu\nu} + V_{\mu\nu}, \tag{11}$$

$$F_{\mu\nu}^{(P_{\lambda\sigma})} = H_{\mu\nu}^{\text{core}} + 2J_{\mu\nu}^{(P_{\lambda\sigma})} - K_{\mu\nu}^{(P_{\lambda\sigma})}, \tag{12}$$

where the superscripts in Eq. (12) indicate which density matrix is contracted against the ERIs; in the case of Hartree–Fock (HF) and Kohn–Sham (KS) density functional theory (DFT), the ground state density $P_{\lambda\sigma}$ is used. However, one can also leverage these quantities in a variety of other contexts; for example, the molecular orbital

(MO) basis ERIs are often needed for post-SCF methods and can be calculated by using **J** builds to half-transform the ERIs for each pair of MOs. Consider the following two term groupings for the AO to MO transformation:

$$(pq|rs) = \sum_{\mu} C_{\mu p}\left(\sum_{\nu} C_{\nu q}\left(\sum_{\lambda} C_{\lambda r}\left(\sum_{\sigma} C_{\sigma s}(\mu\nu|\lambda\sigma)\right)\right)\right), \tag{13}$$

$$(pq|rs) = \sum_{\mu\nu\lambda\sigma} C_{\mu p} C_{\nu q}(\mu\nu|\lambda\sigma)\underbrace{C_{\lambda r} C_{\sigma s}}_{P^{(rs)}_{\lambda\sigma}} = \sum_{\mu\nu} C_{\mu p} C_{\nu q} J_{\mu\nu}^{\left(P^{(rs)}_{\lambda\sigma}\right)}, \tag{14}$$

where p, q, r, and s index molecular orbitals and $C_{\mu p}$ are the molecular orbital coefficients. The traditional algorithm in Eq. (13) performs four sequential quarter-transformations and scales as $O(N_{\mathrm{AO}}^4 N_{\mathrm{MO}})$; meanwhile, Eq. (14) groups the first half-transformation into a Coulomb matrix and formally scales as $O(N_{\mathrm{AO}}^4 N_{\mathrm{MO}}^2)$ since one **J** build is required for each MO pair. However, the effective quadratic scaling of the **J** builds makes the algorithm in Eq. (14) favorable in practice. This is especially true when considering that this transformation is generally used for transforming small subsets of the MO space, such as for complete active space (CAS) methods. Since active space sizes are typically much smaller than the full orbital space, the scaling behavior of the **J** build dominates and the entire integral transformation step is observed to have subquadratic scaling with respect to system size.[46]

For methods such as Møller–Plesset perturbation theory or coupled-cluster singles and doubles (CCSD) that require the full MO space, factorization of the ERIs into products of third-order tensors—as in density fitting[40,47–50] (DF) or Cholesky decomposition[51,52] (CD) approximations—may be preferable,

$$(\mu\nu|\lambda\sigma) \approx \sum_{A} L^{A}_{\mu\nu} L^{A}_{\lambda\sigma}. \tag{15}$$

Here, the **L** tensors represent an incomplete CD of the ERIs. Since we have already shown that multiple **J** builds can be used to access the full ERI tensor, it is clearly possible to perform the CD of the ERIs using only **J** and **K** builds. The question that remains is whether or not such a **J**/**K**-based algorithm would be efficient enough to be useful. In Algorithm 1, we present our purely **J**/**K**-based approach to the CD of the ERIs. In order to define the pivots, we must have access to the diagonal elements of the ERI tensor [i.e., $(\mu\nu|\mu\nu)$-type integrals]. These can be accessed by either N_{AO}^2 **J** builds or N_{AO} **K** builds; a **K** build using a density matrix with a single entry of unity on the diagonal will result in a **K** matrix with N_{AO} of the diagonal ERIs on its diagonal. While there is some unnecessary overhead in this approach (i.e., the full K matrix is constructed when only the diagonal is needed), we find that it takes a small fraction of the total time required to perform the CD. With the diagonal elements in hand, we can begin the formation of the **L** tensors. Here, **J** builds are used to obtain the slices of the ERI tensor corresponding to the largest errors on the diagonal. The rest of the algorithm proceeds as usual. We use the resulting CD of the ERIs in the context of CCSD computations in our new CCBox module; we will discuss the performance of the **J**/**K**-based CD algorithm later in that context.

ALGORITHM 1. Cholesky decomposition using Coulomb and exchange matrices.

```
Initialize: $\mathbf{E} = 0$, $\mathbf{P} = 0$, $\mathbf{L} = 0$
for $m$ Basis functions do
    Set: $P_{mm} = 1$
    Form: $K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda|\nu\sigma) P_{\lambda\sigma}$
    for $n$ Basis functions do
        Set: $E_{mn,mn} = K_{nn}$
    end for
    Set: $P_{mm} = 0$
end for
Initialize: $N_{\mathrm{CD}} = 0$
while max($\mathbf{E}$) < $thresh$ do
    Set: $r,s$ = index of max($\mathbf{E}$)
    Set: $P_{rs} = 1$
    Form: $J_{\mu\nu} = \sum_{\lambda\sigma} (\mu\nu|\lambda\sigma) P_{\lambda\sigma}$
    for $m$ Basis functions do
        for $n$ Basis functions do
            Set: $L^{N_{\mathrm{CD}}}_{mn} = J_{mn}$
            for $A < N_{\mathrm{CD}}$ do
                Set: $L^{N_{\mathrm{CD}}}_{mn} = L^{N_{\mathrm{CD}}}_{mn} - L^{A}_{mn} L^{A}_{rs}$
            end for
        end for
    end for
    Set: $\alpha = \sqrt{L^{N_{\mathrm{CD}}}_{rs}}$
    for $m$ Basis functions do
        for $n$ Basis functions do
            if$m, n = r, s$ then
                Set: $L^{N_{\mathrm{CD}}}_{mn} = \alpha$
            else
                Set: $L^{N_{\mathrm{CD}}}_{mn} = L^{N_{\mathrm{CD}}}_{mn} / \alpha$
            end if
            Update: $E_{mn,mn} = E_{mn,mn} - L^{N_{\mathrm{CD}}}_{mn} L^{N_{\mathrm{CD}}}_{mn}$
        end for
    end for
    Set: $P_{rs} = 0$
    Set: $N_{\mathrm{CD}} = N_{\mathrm{CD}} + 1$
end while
```

Linear response excited state methods such as configuration-interaction singles (CIS) and Tamm–Dancoff approximation time-dependent density functional theory (TDA-TDDFT) can be formulated in terms of Fock-like matrix builds with a nonsymmetric transition density matrix,

$$T_{\lambda\sigma} = \sum_{ia} X_{ia} C_{\lambda i} C_{\sigma a}, \tag{16}$$

where i, j, k, and l index the occupied orbitals, a, b, c, and d index the virtual orbitals, and X_{ia} are the transition amplitudes given in the CIS/TDA-TDDFT response equation $\mathbf{AX} = \mathbf{X}\boldsymbol{\omega}$.[53,54] As another illustrative example, we consider the use of **J** and **K** matrix builds in the AO-direct formalism of the coupled-perturbed state-averaged complete active space SCF (CP-SA-CASSCF) equations. In our implementation, the coupled-perturbed multiconfiguration self-consistent field equations are solved iteratively, and Fock-like matrix builds with the following generalized density matrices are sufficient to recover the response due to the orbital-CI block of the Hessian matrix:

$$\tilde{\boldsymbol{D}} = \begin{bmatrix} 0 & \tilde{\kappa}^{\Theta}_{\text{clsd,act}} & \tilde{\kappa}^{\Theta}_{\text{clsd,virt}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \bar{\boldsymbol{D}} = \begin{bmatrix} 0 & 0 & 0 \\ -\tilde{\kappa}^{\Theta}_{\text{act,clsd}} & 0 & \tilde{\kappa}^{\Theta}_{\text{act,virt}} \\ 0 & 0 & 0 \end{bmatrix}, \tag{17}$$

where $\tilde{\kappa}^{\Theta}$ are estimates of the orbital Lagrangian multipliers.[55] These examples suggest that one can view **J** and **K** builds as general ERI contraction engines, where methods are reformulated in terms of Coulomb-like and exchange-like terms and the necessary entries are packed into density matrices for the GPU-accelerated contraction against the ERIs. Therefore, we consider the GPU-accelerated **J** and **K** matrix builds provided by IntBox as the first of several core building blocks for the electronic structure.

III. SQMBOX

The emergence of the electronic structure code TeraChem in 2008 is directly tied to the development of the GPU-accelerated IntBox library mentioned above. With this tool in hand, the entire electronic structure development in TeraChem has been pursued with the maxim of leveraging the basic quantities provided by IntBox. Semiempirical methods, in particular, semiempirical density functional tight-binding methods,[56,57] can be interpreted as a minimal basis set Hartree–Fock/Kohn–Sham density functional theory type treatment with a modified Hamiltonian. In most semiempirical methods, that boils down to applying the Mulliken approximation[58] to the one-electron and possibly two-electron integrals and subsequent semiempirical approximation of the remaining one- and two-center terms,

$$\langle \mu | \hat{O} | \nu \rangle = O_{\mu\nu} \approx \frac{1}{2} S_{\mu\nu} (O_{\mu\mu} + O_{\nu\nu}) \approx \frac{1}{2} S_{\mu\nu} (\tilde{O}_{\mu} + \tilde{O}_{\nu}), \tag{18}$$

$$\begin{aligned} (\mu\nu|\lambda\sigma) &\approx \frac{1}{4} S_{\mu\nu} S_{\lambda\sigma} [(\mu\mu|\lambda\lambda) + (\mu\mu|\sigma\sigma) + (\nu\nu|\lambda\lambda) + (\nu\nu|\sigma\sigma)] \\ &\approx \frac{1}{4} S_{\mu\nu} S_{\lambda\sigma} (\gamma_{\mu\lambda} + \gamma_{\mu\sigma} + \gamma_{\nu\lambda} + \gamma_{\nu\sigma}). \end{aligned} \tag{19}$$

The recently proposed GFN- and GFN2-xTB methods[59,60] furthermore share with *ab initio* methods the use of a nonorthogonal (though atomwise orthogonal) GTO basis set, thus requiring a setup that is formally completely equivalent to typical *ab initio* electronic structure workflows and only differs in terms of the explicit Hamiltonian matrix elements. We have recently added a semiempirical integral library called SQMBox to TeraChem. This library provides semiempirical equivalents to the basic quantities provided by the IntBox library, i.e., quantities that have been approximated as shown in Eqs. (18) and (19) (see Fig. 4). SQMBox evaluates the Mulliken-approximated integrals from the provided overlap integrals; therefore, the library is agnostic to the underlying basis function type and can, in principle, be combined with other AO types

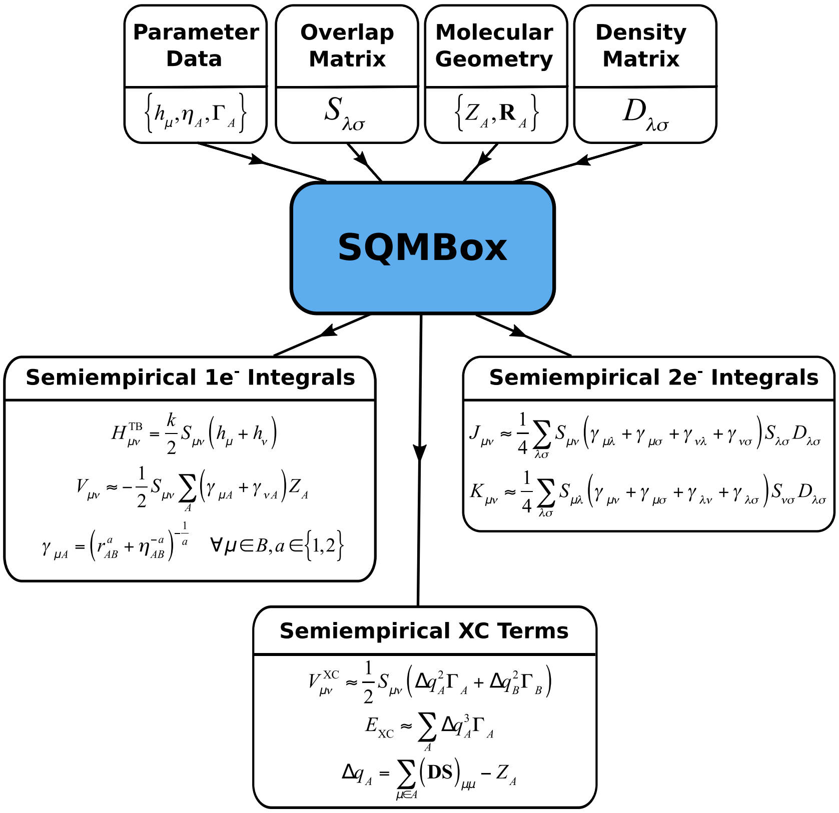

Parameter Data
$\{h_\mu, \eta_A, \Gamma_A\}$
Overlap Matrix
$S_{\lambda\sigma}$
Molecular Geometry
$\{Z_A, \mathbf{R}_A\}$
Density Matrix
$D_{\lambda\sigma}$
SQMBox
Semiempirical 1e⁻ Integrals
$H^{\text{TB}}_{\mu\nu} = \frac{k}{2} S_{\mu\nu} (h_\mu + h_\nu)$
$V_{\mu\nu} \approx -\frac{1}{2} S_{\mu\nu} \sum_A (\gamma_{\mu A} + \gamma_{\nu A}) Z_A$
$\gamma_{\mu A} = (r^a_{AB} + \eta^{-a}_{AB})^{-\frac{1}{a}} \quad \forall \mu \in B, a \in \{1,2\}$
Semiempirical 2e⁻ Integrals
$J_{\mu\nu} \approx \frac{1}{4} \sum_{\lambda\sigma} S_{\mu\nu} (\gamma_{\mu\lambda} + \gamma_{\mu\sigma} + \gamma_{\nu\lambda} + \gamma_{\nu\sigma}) S_{\lambda\sigma} D_{\lambda\sigma}$
$K_{\mu\nu} \approx \frac{1}{4} \sum_{\lambda\sigma} S_{\mu\lambda} (\gamma_{\mu\nu} + \gamma_{\mu\sigma} + \gamma_{\lambda\nu} + \gamma_{\lambda\sigma}) S_{\nu\sigma} D_{\lambda\sigma}$
Semiempirical XC Terms
$V^{\text{XC}}_{\mu\nu} \approx \frac{1}{2} S_{\mu\nu} (\Delta q^2_A \Gamma_A + \Delta q^2_B \Gamma_B)$
$E_{\text{XC}} \approx \sum_A \Delta q^3_A \Gamma_A$
$\Delta q_A = \sum_{\mu \in A} (\mathbf{DS})_{\mu\mu} - Z_A$

FIG. 4. Schematic of the SQMBox interface, which takes geometry and density information like the IntBox library and additionally Hamiltonian parameters and the overlap matrix. Gradient contributions are also available but are not shown for clarity. The third-order tight-binding terms are also computed by SQMBox, which are handled like density functional-type exchange correlation (XC) terms.

(e.g., Slater orbitals). Hamiltonians for the GFN-xTB and GFN2-xTB methods are currently available in SQMBox. Additionally, the library has full support for the Mulliken-approximated Fock exchange, which is absent in the original tight-binding methods due to the significant increase in the computational cost with conventional formulations.[56] However, the computational cost for the Mulliken-approximated exchange becomes negligible with our GPU implementation.[61]

The power of our modular electronic structure environment could be demonstrated by easily making novel method combinations available. We successfully combined the GFN-xTB Hamiltonians with the spin-restricted ensemble-referenced Kohn–Sham (REKS) density functional theory method.[62] Due to the entire electronic structure framework being built around the basic quantities from IntBox[63] and the one-to-one mapping of *ab initio* (i.e., IntBox) and semiempirical (i.e., SQMBox) Hamiltonian terms, the novel state-averaged (SA) state-interaction (SI) REKS-xTB method combination is available with the same full functionality as the *ab initio* implementation. Hence, the fairly cumbersome rederivation of gradients and nonadiabatic coupling elements[64] for a separate semiempirical workflow becomes unnecessary. This directly enabled us to conduct an initial benchmarking study, which clearly indicated that exchange is crucial to make xTB Hamiltonians applicable for photochemical problems. Reparameterizing an exchange-including xTB Hamiltonian for this purpose and combining SQMBox with other electronic structure workflows in TeraChem are ongoing developments.

IV. THCBOX

The THCBox library uses tensor hypercontraction (THC) to provide an alternate formalism for contracting density matrices against the ERIs by leveraging rank-reduction techniques rather than the element sparsity used by IntBox. Tensor hypercontraction factorizes the ERIs as

$$(\mu\nu|\lambda\sigma) = \sum_{PQ} X_\mu^P X_\nu^P Z^{PQ} X_\lambda^Q X_\sigma^Q, \tag{20}$$

where P, Q, R, and S are THC auxiliary function indices; in practice, the auxiliary functions are chosen as gridpoints in real space. THC has successfully reduced the formal scaling of many correlation methods, such as second-order Møller–Plesset perturbation theory (MP2),[65–67] scaled opposite spin MP2 (SOS-MP2),[68–70] second-order complete active space perturbation theory (CASPT2),[71] and coupled-cluster theories.[72–75] Therefore, the development of THCBox is aimed at providing these low-scaling THC-based correlation energy and gradient methods through a set of well-defined APIs.

The design of THCBox is divided into three layers: the bottom layer that constructs the THC tensors, the middle layer that provides tensor operations, and the upper layer that provides the library APIs for correlation energies and gradients, as shown in Fig. 5. THC-MP2 and THC-SOS-MP2 require the number of occupied/virtual orbitals and the molecular orbital energies and coefficients from a preceding SCF calculation as inputs and return the corresponding correlation energy and nuclear gradients. The THC-CASPT2 method takes the molecular energies and coefficients from a preceding complete active space self-consistent field (CASSCF) calculation, as well as the occupied-active and active-virtual off-diagonal blocks of the Fock matrix, the CASSCF configuration-interaction amplitudes, and a level-shifting value; at this time, only the correlation energy is implemented for the CASPT2 method. Both the bottom layer (i.e., THC construction) and the middle layer (i.e., tensor operations) are accelerated on the GPUs and described below.

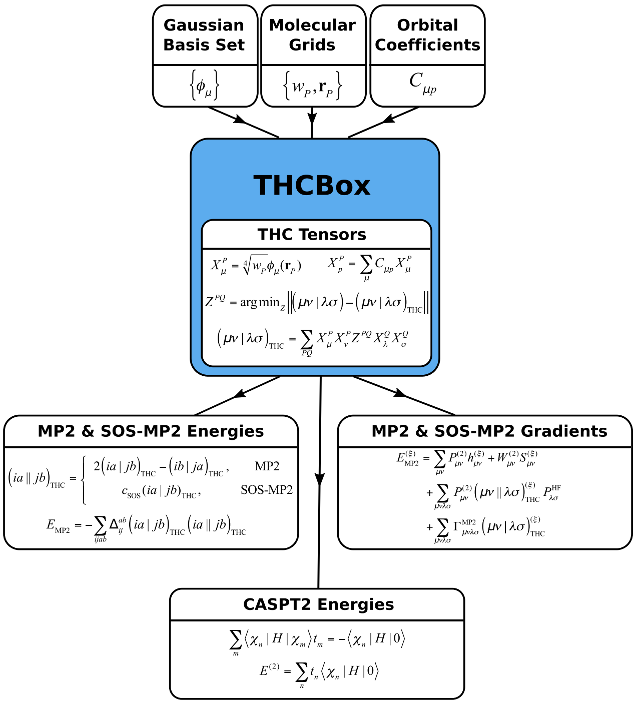

Gaussian Basis Set
$\{\phi_\mu\}$
Molecular Grids
$\{w_P, \mathbf{r}_P\}$
Orbital Coefficients
$C_{\mu p}$
THCBox
THC Tensors
$X_\mu^P = \sqrt[4]{w_P}\phi_\mu(\mathbf{r}_P)$
$X_p^P = \sum_\mu C_{\mu p} X_\mu^P$
$Z^{PQ} = \arg\min_Z \|(\mu\nu|\lambda\sigma) - (\mu\nu|\lambda\sigma)_{\mathrm{THC}}\|$
$(\mu\nu|\lambda\sigma)_{\mathrm{THC}} = \sum_{PQ} X_\mu^P X_\nu^P Z^{PQ} X_\lambda^Q X_\sigma^Q$
MP2 & SOS-MP2 Energies
$(ia\|jb)_{\mathrm{THC}} = \begin{cases} 2(ia|jb)_{\mathrm{THC}} - (ib|ja)_{\mathrm{THC}}, & \text{MP2} \\ c_{\mathrm{SOS}}(ia|jb)_{\mathrm{THC}}, & \text{SOS-MP2} \end{cases}$
$E_{\mathrm{MP2}} = -\sum_{ijab} \Delta_{ij}^{ab} (ia|jb)_{\mathrm{THC}} (ia\|jb)_{\mathrm{THC}}$
MP2 & SOS-MP2 Gradients
$E_{\mathrm{MP2}}^{(\xi)} = \sum_{\mu\nu} P_{\mu\nu}^{(2)} h_{\mu\nu}^{(\xi)} + W_{\mu\nu}^{(2)} S_{\mu\nu}^{(\xi)}$
$+ \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}^{(2)} (\mu\nu\|\lambda\sigma)_{\mathrm{THC}}^{(\xi)} P_{\lambda\sigma}^{\mathrm{HF}}$
$+ \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma}^{\mathrm{MP2}} (\mu\nu|\lambda\sigma)_{\mathrm{THC}}^{(\xi)}$
CASPT2 Energies
$\sum_m \langle \chi_n | H | \chi_m \rangle t_m = -\langle \chi_n | H | 0 \rangle$
$E^{(2)} = \sum_n t_n \langle \chi_n | H | 0 \rangle$

FIG. 5. Schematic of the THCBox interface, which takes basis set, molecular grid, and orbital coefficient information as inputs, generates the various THC tensors, and produces MP2/SOS-MP2/CASPT2 energies and MP2/SOS-MP2 gradients as outputs.

A. THC Tensor Construction Layer

The THC tensor construction layer is responsible for forming the X and Z tensors. The X tensor is computed as density of atomic orbitals on gridpoints,

$$X_\mu^P = \sqrt[4]{\omega_P}\phi_\mu(\mathbf{r}_P). \tag{21}$$

Due to the locality of atomic orbitals, the X tensor is inherently sparse. If sparsity is enabled, then the X tensor will be stored as a sparse matrix, where we keep track of the list of atomic orbitals that contribute to a certain group of points; otherwise, the X tensor is stored as one matrix with dimension N_{AO} by N_{Pt}.

Different variants of THC mainly differ in the definition of the Z tensor. In Least-Squares-AO-THC (LS-AO-THC),[68] the Z tensor is stored as one matrix with dimension N_{Pt} by N_{Pt} and is defined using a least-squares fitting procedure that minimizes the difference between the approximated integrals and targeting integrals,

$$Z = \underset{\tilde{Z}}{\arg\min} \left\| (\mu\nu|\lambda\sigma) - X_\mu^P X_\nu^P \tilde{Z}^{PQ} X_\lambda^Q X_\sigma^Q \right\|_2. \tag{22}$$

By inserting the density-fitting approximation, the analytical solution to the least-squares problem takes the following form:

$$Z^{PQ} = \sum_{RABS} \left[S^{-1}\right]^{PR} \cdot P^{RA} (A|B)^{-1} P^{BS} \cdot \left[S^{-1}\right]^{SQ}, \tag{23}$$

where

$$S^{PR} = \sum_{\mu\nu} X_\mu^P X_\nu^P \cdot X_\mu^R X_\nu^R, \tag{24}$$

$$P^{RA} = \sum_{\mu\nu} X_\mu^R X_\nu^R \cdot (\mu\nu|A). \tag{25}$$

The LS-Dual-Grid THC[70] method is a variation of LS-AO-THC where the three-electron integrals in Eq. (25) are evaluated numerically using a second set of dense grids indexed with $\tilde{P}$ as

$$(\mu\nu|A) \approx \sum_{\tilde{P}} \omega_{\tilde{P}} \phi_\mu(\mathbf{r}_{\tilde{P}}) \phi_\nu(\mathbf{r}_{\tilde{P}}) \int \frac{\psi_A(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_{\tilde{P}}|} d\mathbf{r}. \tag{26}$$

Analytical gradients are available for both LS-AO-THC and LS-Dual-Grid THC.[70]

The efficient construction of the THC tensors relies on exploiting both spatial sparsity and GPU acceleration. This includes using the Schwarz bound to screen AO pairs, as done in IntBox. In addition, $\phi_\mu(\mathbf{r}_P)\phi_\nu(\mathbf{r}_P)$ quickly vanishes as the gridpoint P moves away from the center of the primitive pair. In order to exploit this sparsity on the GPUs, we first divide the physical space into cubic boxes (each side equal to 5 a.u. in our implementation) and then populate primitive pairs and grid points into the boxes according to their center positions. For the purpose of coalesced memory access as well as avoiding thread divergence, the data are rearranged such that the grid and orbital information of the same box is stored together. The scaling of computing the X tensor is thus $O(N_{AO})$. For computing the Z tensor, by properly exploiting spatial sparsity, the contraction $\sum_{\mu\nu} X_\mu^R X_\nu^R \cdot (\mu\nu|A)$ that appears in Eq. (25) scales only as $O(N_{AO}^2)$, while Eq. (24) scales as $O(N_{AO})$. Therefore, the scaling of computing the Z tensor is $O(N_{AO}^3)$, which is dominated by the linear algebra (i.e., matrix multiplication and matrix inversion) in Eq. (23). We also provide Local-LS-AO-THC that overcomes the bottleneck of cubic scaling matrix inversion by using a localized least-squares procedure, a natural extension of the aforementioned gridpoint sparsity screening approach; however, this is currently only implemented for SOS-MP2 energies.[69]

B. Tensor Operation Layer

The middle layer of THCBox provides efficient implementations of tensor operations that involve the THC tensors; for all correlation methods except SOS-MP2, this layer is the most computationally expensive component in THCBox. These tensor operations are abstracted from the contraction patterns that appear repetitively in correlation energy and gradient evaluations and thus are unique to THCBox. One such example is the following contraction:

$$E = \sum_{\mu\nu\lambda\sigma,\mu'\nu'\lambda'\sigma'} (\mu\nu|\lambda\sigma)_{THC} \cdot T^{(1)}_{\mu\mu'} T^{(2)}_{\nu\nu'} T^{(3)}_{\lambda\lambda'} T^{(4)}_{\sigma\sigma'} \cdot \left(\mu'\nu'|\lambda'\sigma'\right)_{THC}, \tag{27}$$

which is required in computing intermediates for both MP2 and CASPT2 energies. A detailed list of tensor operations that have been implemented can be found in our previous work.[71]

Our current strategy for achieving high performance is through a combination of using the sparsity of the X tensor, as well as taking advantage of the mature BLAS library by flattening high order tensors to matrices or vectors. Currently, all BLAS level-1 and level-2 calls are performed on the CPU using MKL, while level-3 operations (i.e., matrix multiplication and diagonalization) are accelerated on the GPU using the cuBLAS or MAGMA libraries. In addition, we also implemented specialized functions for the following two operations:

$$Y_\mu^R = \sum_P T^{RP} X_\mu^P, \tag{28}$$

$$Y_\nu^P = \sum_\mu X_\mu^P T_{\mu\nu}, \tag{29}$$

which take advantage of the inherent sparsity structure of the X tensor. There are two drawbacks in the current implementation. The first is that some operations cannot be efficiently evaluated using the existing BLAS library. One such example is the hypercontraction (summing over a repeated index with more than two occurrences),

$$A_{\mu\nu}^P = \sum_Q Z^{PQ} X_\mu^Q X_\nu^Q, \tag{30}$$

which is one of the most expensive steps in evaluating MP2 and CASPT2 exchange-like energies. Second, the memory transfers between CPU and GPUs are too frequent in our current implementation to account for situations when the tensors cannot all fit into GPU memory, which often happens for large molecules. As the tensor operation layer contains the steepest scaling steps in the calculations, further optimizations of the layer that can overcome these two drawbacks would lead directly to better performance of the correlation methods.

V. CIBOX

The CIBox library provides access to GPU-accelerated software for solving configuration-interaction (CI) related problems. In addition to a direct determinantal CI program[20] with enhanced diagonalization performance,[76] tools are provided which enable spin purification methods[77] that improve eigenvalue problem subspace stability. Direct CI avoids the construction and diagonalization of the electronic Hamiltonian matrix, instead using an iterative procedure to determine a few of the low-lying eigenvalues (and their eigenvectors).[78] The computational bottleneck in direct CI is formation of $\boldsymbol{\sigma}$, the matrix–vector product of the Hamiltonian matrix and a trial vector,

$$\sigma = \mathbf{Hc}. \tag{31}$$

Wavefunction information is often presented in terms of one-electron properties, requiring evaluation of the one-particle density matrix (OPDM). Analytical energy gradients (using a Lagrange based formulation) rely on the efficient formation of the generalized OPDMs and two-particle density matrices (TPDMs),

$$\gamma^{AB}_{pq} = \sum_{IJ} c^A_I c^B_J \langle \Phi_I | \hat{E}_{pq} | \Phi_J \rangle, \tag{32}$$

$$\Gamma^{AB}_{pqrs} = \frac{1}{2} \sum_{IJ} c^A_I c^B_J \langle \Phi_I | \hat{E}_{pq} \hat{E}_{rs} - \delta_{qr} \hat{E}_{ps} | \Phi_J \rangle. \tag{33}$$

Here, I and J index configurations, **c** is the CI vector, $\hat{E}_{pq}$ is an excitation operator from orbital p to orbital q, and superscripts A and B denote that two separate CI vectors can be used in the generalized builds. The schematic in Fig. 6 shows that CIBox takes one- and two-electron integrals in the MO basis, generated by IntBox using the **J** matrix algorithm described in Eq. (14), and configuration space parameters (i.e., number of α/β electrons and active orbitals) as inputs. In return, CIBox provides energies, CI vectors, and string occupancy patterns as solutions to the CI eigenvalue problem for a given spin multiplicity, as well as allowing access to the underlying **σ** and generalized OPDM and TPDM builds from Eqs. (31)–(33), respectively.

As with IntBox, access to the underlying GPU-accelerated quantities enables the rapid development of new electronic structure methods. When developing the rank-reduced full CI (RR-FCI) method,[79] we initially constructed a pilot program using **σ** vectors from direct CI. This allowed us to focus our efforts on rapidly implementing the framework with knowledge that the **σ** vectors were correct. Once our RR-FCI program was completed, we were then able to derive and implement factorized **σ** vectors, knowing that the framework was reliable. This separation of concerns permits both efficient and robust prototyping. A second example is implementation of the time-dependent CASCI method in TeraChem.[80] In this case, the propagation of the electronic wavefunction corresponds to **σ** vector formation provided by CIBox for trial vectors in real and imaginary space. As a final example, we consider the aforementioned response of the orbital-CI Hessian matrix in the AO-direct CP-SA-CASSCF method; after building the Lagrangian multiplier density matrices from Eq. (17), an effective Hamiltonian can be built as

$$\tilde{H}_{IJ} = \sum_{tu} 2\gamma^{IJ}_{tu} \left(F^{(\tilde{D})}_{tu} + F^{(\bar{D})}_{tu} \right) + \frac{1}{2} \sum_{tuvw} 4\Gamma^{IJ}_{tuvw} \sum_{p} \bar{D}_{tp} (pu|vw), \tag{34}$$

where t, u, v, and w index active orbitals, γ^{IJ}_{tu} and Γ^{IJ}_{tuvw} use the generalized OPDM and TPDM build from CIBox, respectively, and the two-electron integrals come from IntBox following the workflow of Eq. (14). Finally, a **σ**-build using this effective Hamiltonian can then be constructed as part of the full Hessian matrix-vector product needed for the coupled-perturbed multiconfiguration self-consistent field equations.[55]

The **σ** and generalized OPDM/TPDM builds have proven to be crucial quantities for CI-based correlated electronic structure methods and therefore enjoy the same status as the **J** and **K** builds from IntBox as key building blocks for the electronic structure. In Fig. 7, we show CASCI benchmarks on myoglobin using a hybrid quantum mechanical/molecular mechanics (QM/MM) scheme with a (16, 16) active space (i.e., 16 electrons in 16 orbitals). We calculate three states (the ground state and two excited states) as a representative calculation for the Q_x and Q_y bands of the iron–porphyrin

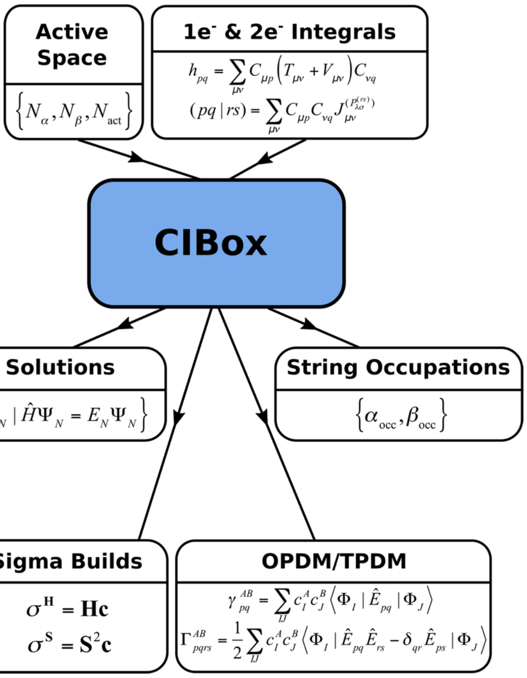

Active Space
$\{N_\alpha, N_\beta, N_{\text{act}}\}$
1e⁻ & 2e⁻ Integrals
$h_{pq} = \sum_{\mu\nu} C_{\mu p} (T_{\mu\nu} + V_{\mu\nu}) C_{\nu q}$
$(pq|rs) = \sum_{\mu\nu} C_{\mu p} C_{\nu q} J^{(P^{(rs)}_{\lambda\sigma})}_{\mu\nu}$
CIBox
CI Solutions
$\{\Psi_N, E_N \mid \hat{H}\Psi_N = E_N \Psi_N\}$
String Occupations
$\{\alpha_{\text{occ}}, \beta_{\text{occ}}\}$
Sigma Builds
$\sigma^{\mathbf{H}} = \mathbf{Hc}$
$\sigma^{\mathbf{S}} = \mathbf{S}^2\mathbf{c}$
OPDM/TPDM
$\gamma^{AB}_{pq} = \sum_{IJ} c^A_I c^B_J \langle \Phi_I | \hat{E}_{pq} | \Phi_J \rangle$
$\Gamma^{AB}_{pqrs} = \frac{1}{2} \sum_{IJ} c^A_I c^B_J \langle \Phi_I | \hat{E}_{pq} \hat{E}_{rs} - \delta_{qr} \hat{E}_{ps} | \Phi_J \rangle$

FIG. 6. Schematic of the CIBox interface, which takes active space information and one- and two-electron MO integrals as inputs, produces wavefunctions, energies, and occupation patterns, and provides sigma builds and one-, and two-particle density matrix builds.

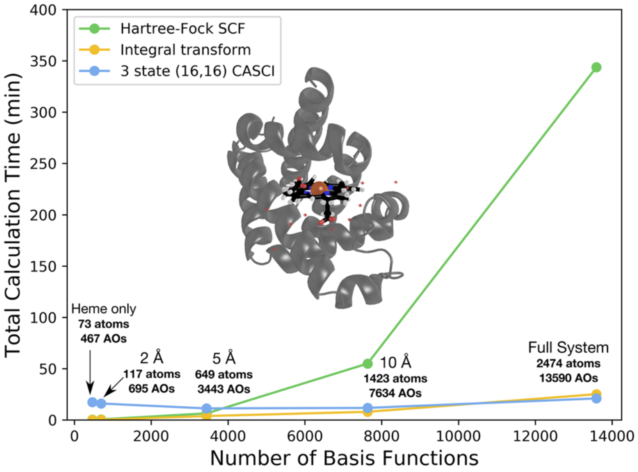


FIG. 7. Timings for key components in a hybrid QM/MM calculation of myoglobin (PDB ID: 3RGK) at the HF-CASCI/6-31G level of theory with a (16e, 16o) active space for the lowest three singlet states. Conventional Fock builds were done in full double precision with all integral thresholds set to 10^{-11}. Different QM regions were selected based on a distance criterion from the heme cofactor. Calculations were run on a Tesla V100 GPU using a single core of a 3.4 GHz Intel Xeon E5-2643v4 CPU.

complex and compare the **J**/**K** enabled SCF, the two-electron integral transform from Eq. (14), and CASCI utilizing **σ**, OPDM, and TPDM builds. Unsurprisingly, the cost of the CASCI procedure remains roughly constant with respect to the QM size and dominates for small regions. The two-electron integral transformation code becomes more expensive at large (i.e., over 1000 atoms and 5000 basis functions) QM regions, but both the transform and CASCI are insignificant when compared with the cost of the Hartree–Fock procedure for the reference orbitals. We additionally show that encapsulating these building blocks also provides a mechanism to make improvements to the underlying code without alteration of downstream software. This ability was leveraged recently when extending sigma, OPDM, and TPDM formation to multiple GPUs[81] and again when developing mixed precision algorithms for sigma vector formation.[82]

VI. CCBOX

The CCBox module is one of the newest features in TeraChem. This library performs GPU-accelerated CCSD[83–85] computations (as well as any method that can be written as a subset of the CCSD diagrams). This library is initialized with the definition of a single reference determinant (usually from an RHF computation); practically speaking, this definition comes from the molecular orbital coefficients and the assignment of those orbitals to the frozen core, occupied, and virtual orbital subspaces. The CCBox library also requires a CD of the ERIs to be provided. To construct the Fock operator, CCBox calls the **J**/**K** builds provided by IntBox. The CCSD program uses the spin-adapted formulation of Koch and co-workers for closed-shell singlet wavefunctions.[86] The GPU algorithm follows along similar lines to that of DePrince and co-workers.[87–89] We attempt to minimize data transfer to and from the GPU by performing array permutations in place of the GPU using custom CUDA kernels. All tensor contractions are performed as matrix multiplications using the cuBLAS library. Using 8 Tesla V100 GPUs, CCBox is able to perform CCSD computations with roughly 1300 basis functions and 100 atoms in less than one day on a single node. Additional performance timings and detailed comparisons to CPU-based CCSD implementations are available in an upcoming paper.[90] In Fig. 8, we show the timings of one CCSD iteration and our aforementioned **J**/**K**-based CD algorithm in a TZVP basis set[91] for a series of water clusters containing up to 30 molecules (930 basis functions). For the largest system, each CCSD iteration takes approximately 22 min and the CD takes 14.5 min. From the perspective of CCBox, the **J**/**K**-based CD algorithm is efficient enough to be useful, since usual CCSD computations take between 15 and 20 iterations to converge. In the context of planned extensions to equation-of-motion CCSD (EOM-CCSD), the time spent performing the CD will become an even smaller fraction of the total computation time.

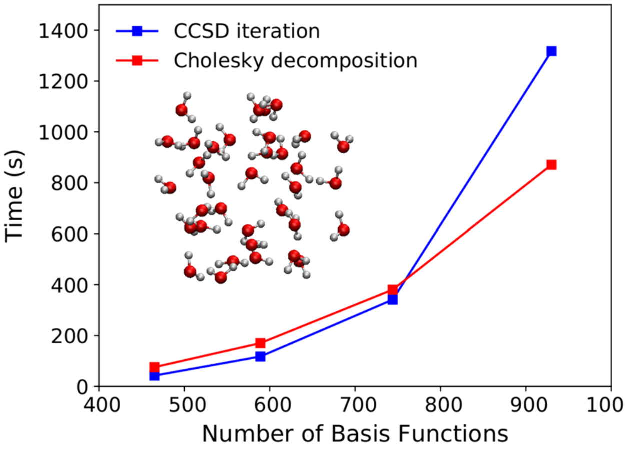
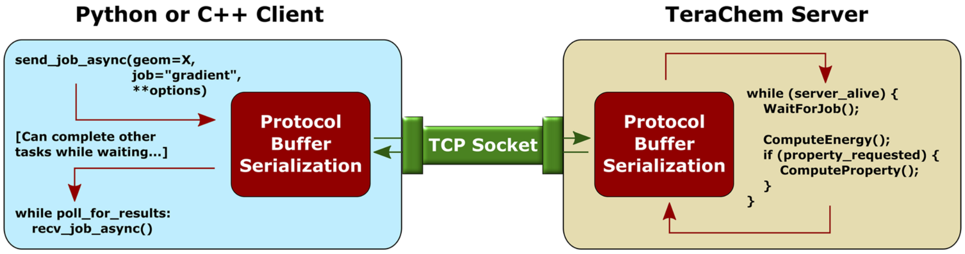


FIG. 8. Timings of one CCSD/TZVP iteration (with the frozen core approximation) compared to the timings of the Cholesky decomposition of the ERIs for a series of water clusters containing 15, 19, 24, and 30 molecules. The Cholesky decomposition was truncated when the error on the diagonal fell below 10^{-4} E_h. Timings were recorded with eight NVIDIA Tesla V100 GPUs and eight CPU threads running on a 20-core Intel Xeon E5-2698 CPU clocked at 2.2 GHz.

VII. TERACHEM PROTOCOL BUFFER (TCPB) SERVER

Dense GPU nodes have seen an upswing in popularity among high performance computing (HPC) clusters recently. For example, Oak Ridge's new Summit supercomputer features nodes with two 22-core IBM POWER9^{TM} CPUs and 6 Tesla V100 GPUs.[92] However, given the competitive nature of applying for time on these few select HPC resources, cloud providers such as Amazon Web Services[93] (AWS), Google Cloud[94] (GC), and NVIDIA GPU Cloud[95] (NGC) can be an attractive source of GPU computing resources. Although several file-based and Message Passing Interface (MPI) interfaces exist for TeraChem, they rely on tightly coupled computing resources (i.e., a shared file system or a specific network topology) and are not well-suited for modern distributed heterogeneous compute systems and cloud computing.

We have developed a socket-based interface based on Google's Protocol Buffers[37] for extensible, language-agnostic serialization, as presented in Fig. 9. Since there is no hard-coded serialization with protocol buffers, it is easy to handle general electronic structure calculations and simple to add additional capabilities to the interface without breaking backwards compatibility. The serialized protocol buffer is then sent in a standard type-length-value (TLV) TCP packet to a Python or C++ client. There are four types of protocol buffer messages: Status, Mol, JobInput, and JobOutput. Status messages include fields to provide information about the server availability, job status (e.g., accepted, rejected, in progress, or completed), and basic job information (e.g., job id and working directory). Mol messages include all the information that is normally stored in XYZ files, such as atom types, molecular coordinates, units, charge, and spin multiplicity. The JobInput message includes fields for specifying ground state methods, calculation runtype, and flags to compute optional properties. All additional input options, including those for post-SCF methods, are passed through as key-value pairs. In our experience, keeping the JobInput message "thin" is advantageous as it enables reuse of existing parsing logic, encouraging consistent job specifications between the standard input files and the TCPB server. For post-SCF methods, CIS/TDDFT and a variety of CAS methods (e.g., FOMO-CASCI, CASSCF, and CISNO-CASCI) are available through the TCPB server mode. Finally, the JobOutput message has fields for all requestable properties, which include



```
send_job_async(geom=X,
               job="gradient",
               **options)

[Can complete other
tasks while waiting...]

while poll_for_results:
  recv_job_async()
```

```
while (server_alive) {
  WaitForJob();

  ComputeEnergy();
  if (property_requested) {
    ComputeProperty();
  }
}
```

FIG. 9. Schematic overview and pseudocode for the TeraChem Protocol Buffer server and a Python client. Job input, output, and status are serialized by protocol buffers, and the type-length-value packet convention is used for communication over a TCP socket.

the following at the time of writing: energies, gradients, nonadiabatic coupling, CI vector overlap, atomic charge and spin, dipoles, MO energies/occupations, bond order matrices, relaxed and unrelaxed dipoles and transition dipoles for CIS methods, and transition dipoles for CAS methods. Additionally, paths to the output file and working directory are also included in the JobOuput message. Access to the working directory has proven useful for debugging, providing verbose error messages (as the last 10 lines of the output file), and access to binary file dumps, which is most commonly used to seed orbitals for subsequent calculations. Further information on the protocol buffers and communication strategy used by the TCPB interface is provided in the supplementary material.

The TeraChem Protocol Buffer (TCPB) server mode provides greater flexibility for loosely coupled computing resources and a clean interface for electronic structure calculations as a remote service. As we have shown in our recent work, this set of interfaces is more commensurate with the cloud computing environment.[96] This interface is in line with other efforts within the community to increase the interoperability of electronic structure codes, such as the calculators in the Atomic Simulation Environment (ASE),[97] the Molecular Sciences Software Institute's (MolSSI) QCEngine project,[98] or the MolSSI Driver Interface (MDI).[99] The TCPB server interface has been utilized for several quantum chemistry workloads, including *ab initio* molecular dynamics (AIMD), geometry minimization, and minimum energy path optimizations for automated reaction discovery,[24,25] and nonadiabatic dynamics with fragment-based models such as the *ab initio* exciton model.[34–36] Further development and inclusion of TeraChem in other downstream applications, such as real-time interactive AIMD,[23] AIMS,[26] and conical intersection optimization,[100] is currently ongoing.

VIII. CONCLUSIONS

In the 1980s, numerical linear algebra was revolutionized by recasting algorithms in terms of matrices and vectors to take advantage of the highly optimized BLAS and LAPACK libraries. In a similar vein, we consider the identification, encapsulation, and subsequent reuse of several core operations as an important step in the development of electronic structure software. In addition to providing code modularity and encapsulation, the existence of the right set of highly optimized routines influences the development of new algorithms. We believe that density matrices and efficient evaluation of their operator representations serve as a physically motivated encapsulation scheme, which is particularly powerful when combined with hardware accelerators such as GPUs, where tensor-like operations can be computed efficiently.

From the past decade of development in the TeraChem software package, two main classes of operations have arisen that change the way we look at electronic structure software development today. The first class of operations are for the evaluation of Hamiltonian matrix elements: IntBox provides **J** and **K** builds for evaluating contractions over the ERI tensor, THCBox uses rank-reduction techniques to offer alternate contraction schemes with reduced scaling, and SQMBox gives the Mulliken-approximated equivalents to IntBox. The second class of operations construct the wavefunction: the density matrix in SCF methods or transition density matrices for the excited state methods in TeraChem, the OPDM and TPDM from CIBox for configuration-interaction methods, and the coupled-cluster amplitudes in CCBox. In the case of CIBox, the $\boldsymbol{\sigma}$-build also provides an efficient way to contract the Hamiltonian and wavefunction. Together, we showed how reusing these electronic structure building blocks (i.e., **J** and **K** builds from IntBox and OPDM/TPDM/$\boldsymbol{\sigma}$ builds from CIBox) can inform new algorithmic breakthroughs, as exemplified by the efficient implementation of the AO-direct SA-CP-CASSCF equations and the novel combination of the GFN-xTB semiempirical methods with REKS. Our recent work in implementing implicit solvent models has highlighted two new one-electron operations: (i) the solvent cavity surface–solute density interaction and (ii) the solvent contribution to the Fock matrix.[101] These quantities are also useful for constructing electrostatic potentials (ESPs), restricted ESP charge fitting, and embedding schemes, so further development to encapsulate and test the broader applicability of these operations is ongoing.[102]

Another exciting avenue of current research is the development of these core electronic structure operations on emerging parallel architectures. During the discussion of mixed precision schemes in IntBox, it was already mentioned that mixed precision could be extended to make use of hardware support for half precision, which is available in the tensor cores of newer GPUs and is intended for machine learning. The latest generation of consumer-grade NVIDIA GPUs, the GeForce RTX line, also features RT cores designed for real-time ray tracing, which have not yet been used for the electronic structure. Tensor processing units (TPUs) may also provide similar computational motifs to GPUs and may provide alternate strategies for higher-order tensor operations. The success of IntBox and CIBox in accelerating SCF and CI-based methods also provides specific targets for exploring what gains can be made from using application-specific integrated circuits (ASICs) for the electronic structure. Field-programmable gate arrays (FPGAs) could serve as a stepping-stone toward electronic structure ASICs. While some work on using FPGAs in quantum chemistry has already been done,[103–105] creating an FPGA equivalent of IntBox or CIBox would have immediate performance benefits. The clear separation and encapsulation of these operations enable developers to rapidly prototype new methods while ensuring their implementations will efficiently leverage the next generation of hardware accelerators.

SUPPLEMENTARY MATERIAL

See supplementary material for a detailed description of the Protocol Buffers interface with usage example and ".proto" definition file.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through the Advanced Computing (SciDAC) program. S.S. acknowledges support from an NSF Graduate Research Fellowship. C.B. thanks the German National Academy of Sciences Leopoldina for support through the Leopoldina Fellowship Program (Project No. LPDS 2018-09).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, "Psi4: An open-source *ab initio* electronic structure program," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **2**, 556 (2012).

[2] D. G. A. Smith, L. A. Burns, D. A. Sirianni, D. R. Nascimento, A. Kumar, A. M. James, J. B. Schriber, T. Zhang, B. Zhang, A. S. Abbott, E. J. Berquist, M. H. Lechner, L. A. Cunha, A. G. Heide, J. M. Waldrop, T. Y. Takeshita, A. Alenaizan, D. Neuhauser, R. A. King, A. C. Simmonett, J. M. Turney, H. F. Schaefer, F. A. Evangelista, A. E. Deprince, T. D. Crawford, K. Patkowski, and C. D. Sherrill, "Psi4NumPy: An interactive quantum chemistry programming environment for reference implementations and rapid development," J. Chem. Theory Comput. **14**, 3504 (2018).

[3] E. F. Valeev LibInt, "A library for the evaluation of molecular integrals of many-body operators over Gaussian functions," https://github.com/evaleev/libint; accessed 31 January 2020.

[4] M. A. L. Marques, M. J. T. Oliveira, and T. Burnus, "Libxc: A library of exchange and correlation functionals for density functional theory," Comput. Phys. Commun. **183**, 2272 (2012).

[5] U. Ekström, L. Visscher, R. Bast, A. J. Thorvaldsen, and K. Ruud, "Arbitrary-order density functional response theory from automatic differentiation," J. Chem. Theory Comput. **6**, 1971 (2010).

[6] CUDA C Programming Guide, https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html; accessed 24 August 2018.

[7] I. S. Ufimtsev and T. J. Martínez, "Graphical processing units for quantum chemistry," Comput. Sci. Eng. **10**, 26 (2008).

[8] I. S. Ufimtsev and T. J. Martínez, "Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation," J. Chem. Theory Comput. **4**, 222 (2008).

[9] I. S. Ufimtsev and T. J. Martinez, "Quantum chemistry on graphical processing units. 2. Direct self-consistent-field implementation," J. Chem. Theory Comput. **5**, 1004 (2009).

[10] I. S. Ufimtsev and T. J. Martinez, "Quantum chemistry on graphical processing units. 3. Analytical energy gradients, geometry optimization, and first principles molecular dynamics," J. Chem. Theory Comput. **5**, 2619 (2009).

[11] K. Yasuda, "Two-electron integral evaluation on the graphics processor unit," J. Comput. Chem. **29**, 334 (2008).

[12] K. Yasuda, "Accelerating density functional calculations with graphics processing unit," J. Chem. Theory Comput. **4**, 1230 (2008).

[13] A. Asadchev, V. Allada, J. Felder, B. M. Bode, M. S. Gordon, and T. L. Windus, "Uncontracted rys quadrature implementation of up to G functions on graphical processing units," J. Chem. Theory Comput. **6**, 696 (2010).

[14] K. A. Wilkinson, P. Sherwood, M. F. Guest, and K. J. Naidoo, "Acceleration of the GAMESS-UK electronic structure package on graphical processing units," J. Comput. Chem. **32**, 2313 (2011).

[15] X. Wu, A. Koslowski, and W. Thiel, "Semiempirical quantum chemical calculations accelerated on a hybrid multicore CPU-GPU computing platform," J. Chem. Theory Comput. **8**, 2272 (2012).

[16] Y. Miao and K. M. Merz, "Acceleration of electron repulsion integral evaluation on graphics processing units via use of recurrence relations," J. Chem. Theory Comput. **9**, 965 (2013).

[17] Y. Miao and K. M. Merz, "Acceleration of high angular momentum electron repulsion integrals and integral derivatives on graphics processing units," J. Chem. Theory Comput. **11**, 1449 (2015).

[18] J. Kussmann and C. Ochsenfeld, "Hybrid CPU/GPU integral engine for strong-scaling *ab initio* methods," J. Chem. Theory Comput. **13**, 3153 (2017).

[19] J. Kalinowski, F. Wennmohs, and F. Neese, "Arbitrary angular momentum electron repulsion integrals with graphical processing units: Application to the resolution of the identity Hartree-Fock method," J. Chem. Theory Comput. **13**, 3160 (2017).

[20] B. S. Fales and B. G. Levine, "Nanoscale multireference quantum chemistry: Full configuration interaction on graphical processing units," J. Chem. Theory Comput. **11**, 4708 (2015).

[21] I. S. Ufimtsev, N. Luehr, and T. J. Martinez, "Charge transfer and polarization in solvated proteins from *ab initio* molecular dynamics," J. Phys. Chem. Lett. **2**, 1789 (2011).

[22] H. J. Kulik, N. Luehr, I. S. Ufimtsev, and T. J. Martinez, "*Ab initio* quantum chemistry for protein structures," J. Phys. Chem. B **116**, 12501 (2012).

[23] N. Luehr, A. G. B. Jin, and T. J. Martínez, "*Ab initio* interactive molecular dynamics on graphical processing units (GPUs)," J. Chem. Theory Comput. **11**, 4536 (2015).

[24] L.-P. Wang, A. Titov, R. McGibbon, F. Liu, V. S. Pande, and T. J. Martínez, "Discovering chemistry with an *ab initio* nanoreactor," Nat. Chem. **6**, 1044 (2014).

[25] L.-P. Wang, R. T. McGibbon, V. S. Pande, and T. J. Martinez, "Automated discovery and refinement of reactive molecular dynamics pathways," J. Chem. Theory Comput. **12**, 638 (2016).

[26] B. F. E. Curchod and T. J. Martínez, "*Ab initio* nonadiabatic quantum molecular dynamics," Chem. Rev. **118**, 3305 (2018).

[27] J. W. Snyder, B. F. E. Curchod, and T. J. Martínez, "GPU-accelerated state-averaged complete active space self-consistent field interfaced with *ab initio* multiple spawning unravels the photodynamics of provitamin D3," J. Phys. Chem. Lett. **7**, 2444 (2016).

[28] B. F. E. Curchod, A. Sisto, and T. J. Martínez, "*Ab initio* multiple spawning photochemical dynamics of DMABN using GPUs," J. Phys. Chem. A **121**, 265 (2017).

[29] D. Hollas, L. Šištík, E. G. Hohenstein, T. J. Martínez, and P. Slavíček, "Nonadiabatic *ab initio* molecular dynamics with the floating occupation molecular orbital-complete active space configuration interaction method," J. Chem. Theory Comput. **14**, 339 (2018).

[30] T. J. A. Wolf, D. M. Sanchez, J. Yang, R. M. Parrish, J. P. F. Nunes, M. Centurion, R. Coffee, J. P. Cryan, M. Gühr, K. Hegazy, A. Kirrander, R. K. Li, J. Ruddock, X. Shen, T. Vecchione, S. P. Weathersby, P. M. Weber, K. Wilkin, H. Yong, Q. Zheng, X. J. Wang, M. P. Minitti, and T. J. Martínez, "The photochemical ring-opening of 1,3-cyclohexadiene imaged by ultrafast electron diffraction," Nat. Chem. **11**, 504 (2019).

[31] K. J. Wilkin, R. M. Parrish, J. Yang, T. J. A. Wolf, J. P. F. Nunes, M. Guehr, R. Li, X. Shen, Q. Zheng, X. Wang, T. J. Martinez, and M. Centurion, "Diffractive imaging of dissociation and ground-state dynamics in a complex molecule," Phys. Rev. A **100**, 023402 (2019).

[32] R. Liang, F. Liu, and T. J. Martínez, "Nonadiabatic photodynamics of retinal protonated schiff base in channelrhodopsin 2," J. Phys. Chem. Lett. **10**, 2862 (2019).

[33] J. K. Yu, R. Liang, F. Liu, and T. J. Martínez, "First-principles characterization of the elusive i fluorescent state and the structural evolution of retinal protonated schiff base in bacteriorhodopsin," J. Am. Chem. Soc. **141**, 18193 (2019).

[34] A. Sisto, D. R. Glowacki, and T. J. Martinez, "*Ab initio* nonadiabatic dynamics of multichromophore complexes: A scalable graphical-processing-unit-accelerated exciton framework," Acc. Chem. Res. **47**, 2857 (2014).

[35] A. Sisto, C. Stross, M. W. Van Der Kamp, M. O'Connor, S. McIntosh-Smith, G. T. Johnson, E. G. Hohenstein, F. R. Manby, D. R. Glowacki, and T. J. Martinez, "Atomistic non-adiabatic dynamics of the LH2 complex with a GPU-accelerated: *Ab initio* exciton model," Phys. Chem. Chem. Phys. **19**, 14924 (2017).

[36] X. Li, R. M. Parrish, F. Liu, S. I. L. Kokkila Schumacher, and T. J. Martínez, "An *ab initio* exciton model including charge-transfer excited states," J. Chem. Theory Comput. **13**, 3493 (2017).

[37] Google Protocol Buffers, https://developers.google.com/protocol-buffers; accessed 3 September 2018.

[38] S. F. Boys, "Electronic wavefunctions I. A general method of calculation for the stationary states of any molecular system," Proc. R. Soc. London **200**, 542 (1950).

[39] P. M. W. Gill, "Molecular integrals over Gaussian basis functions," Adv. Quantum Chem. **25**, 141 (1994).

[40] J. L. Whitten, "Coulombic potential energy integrals and approximations," J. Chem. Phys. **58**, 4496 (1973).

[41] N. Luehr, I. S. Ufimtsev, and T. J. Martínez, "Dynamic precision for electron repulsion integral evaluation on graphical processing units (GPUs)," J. Chem. Theory Comput. **7**, 949 (2011).

[42] M. Häser and R. Ahlrichs, "Improvements on the direct SCF method," J. Comput. Chem. **10**, 104 (1989).

[43] A. V. Titov, I. S. Ufimtsev, N. Luehr, and T. J. Martinez, "Generating efficient quantum chemistry codes for novel architectures," J. Chem. Theory Comput. **9**, 213 (2013).

[44] C. Song, L.-P. Wang, and T. J. Martínez, "Automated code engine for graphical processing units: Application to the effective core potential integrals and gradients," J. Chem. Theory Comput. **12**, 92 (2016).

[45] C. Song, L.-P. Wang, T. Sachse, J. Preiß, M. Presselt, and T. J. Martínez, "Efficient implementation of effective core potential integrals and gradients on graphical processing units," J. Chem. Phys. **143**, 014114 (2015).

[46] E. G. Hohenstein, N. Luehr, I. S. Ufimtsev, and T. J. Martínez, "An atomic orbital-based formulation of the complete active space self-consistent field method on graphical processing units," J. Chem. Phys. **142**, 224103 (2015).

[47] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, "On the applicability of LCAO-Xα methods to molecules containing transition metal atoms: The nickel atom and nickel hydride," Int. J. Quantum Chem. **12**, 81 (1977).

[48] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, "On some approximations in applications of Xα theory," J. Chem. Phys. **71**, 3396 (1979).

[49] 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 (1993).

[50] O. Vahtras, J. Almlöf, and M. W. Feyereisen, "Integral approximations for LCAO-SCF calculations," Chem. Phys. Lett. **213**, 514 (1993).

[51] N. H. F. Beebe and J. Linderberg, "Simplifications in the generation and transformation of two-electron integrals in molecular calculations," Int. J. Quantum Chem. **12**, 683 (1977).

[52] I. Røeggen and E. Wisløff-Nilssen, "On the Beebe-Linderberg two-electron integral approximation," Chem. Phys. Lett. **132**, 154 (1986).

[53] C. M. Isborn, N. Luehr, I. S. Ufimtsev, and T. J. Martínez, "Excited-state electronic structure with configuration interaction singles and Tamm–Dancoff time-dependent density functional theory on graphical processing units," J. Chem. Theory Comput. **7**, 1814 (2011).

[54] J. B. Foresman, M. Head-Gordon, J. A. Pople, and M. J. Frisch, "Toward a systematic molecular orbital theory for excited states," J. Phys. Chem. **96**, 135 (1992).

[55] J. W. Snyder, B. S. Fales, E. G. Hohenstein, B. G. Levine, and T. J. Martínez, "A direct-compatible formulation of the coupled perturbed complete active space self-consistent field equations on graphical processing units," J. Chem. Phys. **146**, 174113 (2017).

[56] M. Elstner, D. Porezag, G. Jungnickel, J. Elsner, M. Haugk, T. Frauenheim, S. Suhai, and G. Seifert, "Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties," Phys. Rev. B **58**, 7260 (1998).

[57] M. Gaus, A. Goez, and M. Elstner, "Parametrization and benchmark of DFTB3 for organic molecules," J. Chem. Theory Comput. **9**, 338 (2013).

[58] J. M. André and G. Leroy, "On the calculation of polycenter integrals," Bull. Soc. Chim. Belg. **78**, 421 (1969).

[59] S. Grimme, C. Bannwarth, and P. Shushkov, "A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z = 1-86)," J. Chem. Theory Comput. **13**, 1989 (2017).

[60] C. Bannwarth, S. Ehlert, and S. Grimme, "GFN2-xTB: An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions," J. Chem. Theory Comput. **15**, 1652 (2019).

[61] C. Bannwarth and T. J. Martínez, "Novel method combinations enabled by interfacing a semiempirical integral library to a modular *ab initio* electronic structure framework" (unpublished) (2020).

[62] M. Filatov, "Spin-restricted ensemble-referenced Kohn–Sham method: Basic principles and application to strongly correlated ground and excited states of molecules," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **5**, 146 (2015).

[63] F. Liu, M. Filatov, and T. J. Martínez, "Analytical derivatives of the individual state energies in ensemble density functional theory method: II. Implementation on graphical processing units (GPUs)," chemRxiv:7985657.v1 (2019).

[64] M. Filatov, F. Liu, and T. J. Martínez, "Analytical derivatives of the individual state energies in ensemble density functional theory method. I. General formalism," J. Chem. Phys. **147**, 034113 (2017).

[65] E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, "Tensor hypercontraction density fitting. I. Quartic scaling second- and third-order Møller-Plesset perturbation theory," J. Chem. Phys. **137**, 044103 (2012).

[66] S. I. L. Kokkila Schumacher, E. G. Hohenstein, R. M. Parrish, L.-P. Wang, and T. J. Martínez, "Tensor hypercontraction second-order Møller–Plesset perturbation theory: Grid optimization and reaction energies," J. Chem. Theory Comput. **11**, 3042 (2015).

[67] R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, "Tensor hypercontraction. II. Least-squares renormalization," J. Chem. Phys. **137**, 224106 (2012).

[68] C. Song and T. J. Martínez, "Atomic orbital-based SOS-MP2 with tensor hypercontraction. I. GPU-based tensor construction and exploiting sparsity," J. Chem. Phys. **144**, 174111 (2016).

[69] C. Song and T. J. Martínez, "Atomic orbital-based SOS-MP2 with tensor hypercontraction. II. Local tensor hypercontraction," J. Chem. Phys. **146**, 034104 (2017).

[70] C. Song and T. J. Martínez, "Analytical gradients for tensor hyper-contracted MP2 and SOS-MP2 on graphical processing units," J. Chem. Phys. **147**, 161723 (2017).

[71] C. Song and T. J. Martínez, "Reduced scaling CASPT2 using supporting subspaces and tensor hyper-contraction," J. Chem. Phys. **149**, 044108 (2018).

[72] E. G. Hohenstein, S. I. L. Kokkila, R. M. Parrish, and T. J. Martínez, "Quartic scaling second-order approximate coupled cluster singles and doubles via tensor hypercontraction: THC-CC2," J. Chem. Phys. **138**, 124111 (2013).

[73] E. G. Hohenstein, S. I. L. Kokkila, R. M. Parrish, and T. J. Martínez, "Tensor hypercontraction equation-of-motion second-order approximate coupled cluster: Electronic excitation energies in O(N4) time," J. Phys. Chem. B **117**, 12972 (2013).

[74] R. M. Parrish, C. D. Sherrill, E. G. Hohenstein, S. I. L. Kokkila, and T. J. Martínez, "Communication: Acceleration of coupled cluster singles and doubles via orbital-weighted least-squares tensor hypercontraction," J. Chem. Phys. **140**, 181102 (2014).

[75] E. G. Hohenstein, R. M. Parrish, C. D. Sherrill, and T. J. Martínez, "Communication: Tensor hypercontraction. III. Least-squares tensor hypercontraction for the determination of correlated wavefunctions," J. Chem. Phys. **137**, 221101 (2012).

[76] R. M. Parrish, E. G. Hohenstein, and T. J. Martínez, ""Balancing" the block Davidson–Liu algorithm," J. Chem. Theory Comput. **12**, 3003 (2016).

[77] B. S. Fales, E. G. Hohenstein, and B. G. Levine, "Robust and efficient spin purification for determinantal configuration interaction," J. Chem. Theory Comput. **13**, 4162 (2017).

[78] B. Roos, "A new method for large-scale CI calculations," Chem. Phys. Lett. **15**, 153 (1972).

[79] B. S. Fales, S. Seritan, N. F. Settje, B. G. Levine, H. Koch, and T. J. Martínez, "Large scale electron correlation calculations: Rank-reduced full configuration interaction," J. Chem. Theory Comput. **14**, 4139–4150 (2018).

[80] W.-T. Peng, B. S. Fales, and B. G. Levine, "Simulating electron dynamics of complex molecules with time-dependent complete active space configuration interaction," J. Chem. Theory Comput. **14**, 4129 (2018).

[81] B. S. Fales and T. J. Martínez, "Efficient treatment of large active spaces through multi-GPU parallel implementation of direct configuration interaction," J. Chem. Theory Comput. **16**, 1586 (2020).

[82] B. S. Fales, R. M. Parrish, and T. J. Martínez, "Single and mixed precision direct configuration interaction" (unpublished) (2020).

[83] R. J. Bartlett and G. D. Purvis, "Many-body perturbation-theory, coupled-pair many-electron theory, and importance of quadruple excitations for correlation problem," Int. J. Quantum Chem. **14**, 561 (1978).

[84] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, "Electron correlation theories and their application to study of simple reaction potential surfaces," Int. J. Quantum Chem. **14**, 545 (1978).

[85] G. D. Purvis and R. J. Bartlett, "A full coupled-cluster singles and doubles model: The inclusion of disconnected triples," J. Chem. Phys. **76**, 1910 (1982).

[86] 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 (1996).

[87] A. E. DePrince and J. R. Hammond, "Coupled cluster theory on graphics processing units I. The coupled cluster doubles method," J. Chem. Theory Comput. **7**, 1287 (2011).

[88] A. E. DePrince, J. R. Hammond, and C. D. Sherrill, "Iterative coupled-cluster methods on graphics processing units," in *Electronic Structure Calculations on Graphics Processing Units*, edited by R. C. Walkerand and A. W. Goetz (Wiley, West Sussex, UK, 2016), p. 279.

[89] A. E. DePrince, M. R. Kennedy, B. G. Sumpter, and C. D. Sherrill, "Density-fitted singles and doubles coupled cluster on graphics processing units," Mol. Phys. **112**, 844 (2014).

[90] B. S. Fales, E. Curtis, K. G. Johnson, D. Lahana, S. Seritan, Y. Wang, H. Weir, T. J. Martínez, and E. G. Hohenstein, "Efficiency of coupled-cluster singles and doubles on modern stream processing architectures," J. Chem. Theory Comput. (submitted) (2020).

[91] A. Schäfer, C. Huber, and R. Ahlrichs, "Fully optimized contracted Gaussian basis sets of triple zeta valence quality for atoms Li to Kr," J. Chem. Phys. **100**, 5829 (1994).

[92] ORNL Launches Summit Supercomputer, https://www.ornl.gov/news/ornl-launches-summit-supercomputer; accessed 31 January 2020.

[93] Amazon EC2, https://aws.amazon.com/ec2/; accessed 3 September 2018.

[94] Google Cloud Compute Products, https://cloud.google.com/products/compute/; accessed 3 September 2018.

[95] NVIDIA GPU Cloud, https://www.nvidia.com/en-us/gpu-cloud/; accessed 3 September 2018.

[96] S. Seritan, K. Thompson, and T. J. Martínez, "TeraChem Cloud: A high-performance computing service for scalable distributed GPU-accelerated electronic structure calculations," J. Chem. Inf. Model. **60**, 2126 (2020).

[97] A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment: A Python library for working with atoms," J. Condens.: Matter Phys. **29**, 273002 (2017).

[98] The MolSSI Quantum Chemistry Archive, https://qcarchive.molssi.org/; accessed 31 January 2020.

[99] MolSSI Driver Interface Library, https://molssi.github.io/MDI_Library/html/index.html; accessed 18 February 2020.

[100] B. G. Levine, J. D. Coe, and T. J. Martínez, "Optimizing conical intersections without derivative coupling vectors: Application to multistate multireference second-order perturbation theory (MS-CASPT2)," J. Phys. Chem. B **112**, 405 (2008).

[101] F. Liu, N. Luehr, H. J. Kulik, and T. J. Martínez, "Quantum chemistry for solvated molecules on graphical processing units using polarizable continuum models," J. Chem. Theory Comput. **11**, 3131 (2015).

[102] F. Liu, D. M. Sanchez, H. J. Kulik, and T. J. Martínez, "Exploiting graphical processing units to enable quantum chemistry calculation of large solvated molecules with conductor-like polarizable continuum models," Int. J. Quantum Chem. **119**, e25760 (2019).

[103] A. Gothandaraman, G. D. Peterson, G. L. Warren, R. J. Hinde, and R. J. Harrison, "FPGA acceleration of a quantum Monte Carlo application," Parallel Comput. **34**, 278 (2008).

[104] M. Wielgosz, G. Mazur, M. Makowski, E. Jamro, P. Russek, and K. Wiatr, "Analysis of the basic implementation aspects of hardware-accelerated density functional theory calculations," Comput. Inform. **29**, 989 (2010).

[105] D. Yang, G. D. Peterson, and H. Li, "Compressed sensing and Cholesky decomposition on FPGAs and GPUs," Parallel Comput. **38**, 421 (2012).