

Parallelisation in the CCSD(T) method

Zijun Zhao^{a,b}

^aCavendish Laboratory, Department of Physics, J J Thomson Avenue, Cambridge. CB3 0HE

^bYusuf Hamied Department of Chemistry, Lensfield Road, Cambridge. CB2 1EW

Abstract

In this work, a modern Fortran implementation of efficient and parallel molecular electronic structure calculations up to the completely renormalised CCSD(T) (CR-CCSD(T)) theory level is reported. The underlying theories and the strategies for efficient code implementation and parallelisation are discussed. Full binding curves of several challenging test cases including water and the fluorine, nitrogen and carbon dimers are obtained at up to CR-CCSD(T) level.

Contents

1	Introduction	1
1.1	Hartree-Fock theory	1
1.2	Møller-Plesset perturbation theory	2
1.3	Coupled-cluster theory	2
1.3.1	Coupled-cluster singles and doubles (CCSD)	2
1.3.2	Coupled-cluster singles and doubles with perturbative triples (CCSD(T))	3
2	Code implementation	5
2.1	Compilation and usage	5
2.2	Capabilities	5
2.2.1	Correct implementation of spin-free CCSD	5
2.3	Code layout	5
2.4	Tensor operations	6
2.4.1	Existing libraries	6
2.4.2	General matrix multiply	6
2.4.3	Tensor benchmarks and heuristics	6
2.5	Parallelisation strategies	7
2.5.1	Shared memory parallelism	7
2.5.2	Distributed memory parallelism	7
2.5.3	GPU acceleration	8
2.6	Computational complexity	9
3	Results	9
3.1	Breaking single bonds	9
3.1.1	Water	9
3.1.2	Fluorine	10
3.2	Breaking multiple bonds	10
3.2.1	Nitrogen	10
3.2.2	Carbon dimer	10
3.3	Parallel performance	10
4	Future work	11
5	Conclusions	13
Appendix A	The HF-SCF algorithm	13

1. Introduction

Modern molecular electronic structure theory mainly concerns itself with low-scaling, generalisable and systematically improvable methods of electron correlation. There are, broadly speaking, two branches of the field: wavefunction methods and density functional theory

(DFT). The former, which is the focus of this work, generally have high-order polynomial scaling, with Hartree-Fock (HF) starting at $\mathcal{O}(N^4)$, and post-HF theories starting at $\mathcal{O}(N^5)$ for Møller-Plesset second-order perturbation theory (MP2), up to full configuration interaction (FCI), which scales like $\mathcal{O}(N!)$. Though expensive, they are all systematically improvable. The latter has lower-order polynomial scaling, with the generalised gradient approximation (GGA) functionals having $\mathcal{O}(N^3)$ scaling, and the meta-GGA and hybrid functionals scaling as $\mathcal{O}(N^4)$. However, DFT does not provide a clear path to systematically improving the quality of calculations, although there exists the so-called Jacob's ladder that provides a heuristic for pseudo-systematic improvability [1].

Among the wavefunction-based theories, HF provides a single-determinant, uncorrelated wavefunction that forms the basis of all other post-HF theories. The three main approaches to wavefunction-based electron correlation are many-body perturbation theories (MBPT), coupled-cluster (CC) theories and configuration interaction (CI) theories.

1.1. Hartree-Fock theory

The Hartree-Fock theory is the cornerstone of all wavefunction-based electron correlation schemes. It starts with a semi-classically motivated one-electron effective Hamiltonian:

$$\hat{h}^{\text{HF}}(\mathbf{r}_i) = -\frac{1}{2}\nabla_i^2 - \sum_{\mu} \frac{1}{r_{\mu i}} + v^{\text{HF}}(\mathbf{r}_i) \quad (1)$$

where v^{HF} is the mean-field potential of all other electrons, and the μ indices denote nuclei. This effective

Hamiltonian, also known as the *Fock operator*, $\hat{f}(\mathbf{r}_i)$, can be seen to arise from the variational minimisation of a specific N -electron trial wavefunction $|\Psi_0\rangle$:

$$E_{\text{HF}} = \min_{\psi_i} \langle \Psi_0 | \hat{\mathbf{H}} | \Psi_0 \rangle \Leftarrow \hat{f} | \phi_p \rangle = \epsilon_p | \phi_p \rangle \quad (2)$$

where ϕ_i are the variationally optimised molecular orbitals, and the trial wavefunction Ψ_0 is an antisymmetrised product of N one-electron wavefunctions ψ_i , also known as a *Slater determinant*:

$$\Psi_0 = \hat{\mathcal{A}} \prod_i^N \psi_i, \quad \hat{\mathcal{A}} \equiv \sum_u^{N!} \sigma_u \hat{P}_u \quad (3)$$

where $\hat{\mathcal{A}}$ is the antisymmetriser, σ_u is the parity of the permutation brought about by \hat{P}_u .

In a finite basis set of atomic orbital (AO) functions χ_μ , ψ_i can be expanded as follows:

$$\psi_i = \sum_\mu c_{i\mu} \chi_\mu \quad (4)$$

The Roothaan Hall equation [2, 3] enables practical computation of the Hartree Fock equation within a finite basis set: where \mathbf{S} is the overlap matrix between AO basis functions, the equation is given by:

$$\mathbf{FC} = \mathbf{SC}\epsilon \quad (5)$$

This equation is then solved iteratively, as in [Algorithm 1](#), to yield the HF eigenvalues and eigenvectors, for use in post-HF methods.

1.2. Møller-Plesset perturbation theory

Møller-Plesset perturbation theory (MPPT) [4] is an instance of MBPT. It scales as $\mathcal{O}(N^5)$, and is one of the cheapest wavefunction-based electron correlation methods available. It is size-extensive (total correlation energy scales linearly with the size of the system), but only order-by-order [5, p. 724] (the energy correction formula for each order of perturbation, as a whole, is size-extensive, but individual terms that contribute may not be). The Møller-Plesset partitioning of the electronic Hamiltonian into a diagonal, zero-order one-electron operator and a perturbation operator is given by [6, p. 80]:

$$\hat{\mathbf{H}} = \underbrace{\sum_i^n [\hat{h}(\mathbf{r}_i) + v^{\text{HF}}(\mathbf{r}_i)]}_{\hat{\mathbf{H}}^{(0)}} + \underbrace{\sum_{i < j} \frac{1}{r_{ij}} - \sum_i^n v^{\text{HF}}(\mathbf{r}_i)}_{\hat{\mathbf{H}}^{(1)}} \quad (6)$$

As the zero-order Hamiltonian is the Fock operator, the zero-order, *i.e.* reference, wavefunctions are the Hartree-Fock wavefunctions and excitations thereof; and the zero-order energies are the sums of HF orbital energies in the reference functions. The substitution of these into the standard Rayleigh-Schrödinger perturbation equations [6, Ch. 2.2] gives:

$$E^{(2)} = \frac{1}{4} \sum_{ijab} \frac{(\langle ij | ab \rangle - \langle ij | ba \rangle)^2}{D_{ij}^{ab}} \quad (7)$$

where $D_{ij\dots}^{ab\dots} = \epsilon_i - \epsilon_a + \epsilon_j - \epsilon_b + \dots$ is known as the standard MBPT denominator, and $\langle ij | ab \rangle = \iint \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) r_{12}^{-1} \phi_a(\mathbf{r}_1) \phi_b(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2$ is the two-electron integral in the physicists' notation.

1.3. Coupled-cluster theory

Coupled-cluster theory is intimately related to MPPT, albeit starting at a higher computational complexity of $\mathcal{O}(N^6)$ for CCSD. It provides better systematic convergence towards the Full configuration interaction (FCI) limit with increasing truncation level. This is accomplished by the compact form of the *exponential ansatz*:

$$\begin{aligned} |\Psi_{\text{CC}}\rangle &= \exp(\hat{T}) |\Psi_0\rangle, \\ \hat{T} &= \sum_\mu^\kappa \hat{T}_\mu, \quad \kappa \text{ being the truncation level}, \\ \hat{T}_\mu &= \sum_{\substack{\text{perm.} \\ \text{unique}}} t_i^{a\dots} \hat{\tau}_{i\dots}^a, \quad \hat{\tau}_{i\dots}^a = \hat{a}_a^\dagger \hat{a}_i \dots, \end{aligned} \quad (8)$$

where the \hat{T} 's are *excitation operators*, the t 's are *cluster amplitudes* and $\hat{\tau}$'s are *cluster operators*.

The derivation of standard, unlinked coupled cluster equations usually starts with the projection of Schrödinger equation by the Møller-Plesset zero-order wavefunctions:

$$\hat{\mathbf{H}} |\Psi_{\text{CC}}^{\text{full}}\rangle = E_{\text{exact}} |\Psi_{\text{CC}}^{\text{full}}\rangle \quad (9a)$$

\Downarrow (projection by $\langle \Psi^{(0)} |$)

$$\langle \Psi_0 | \hat{\mathbf{H}} \exp(\hat{T}) | \Psi_0 \rangle = E_{\text{CC}} \quad (9b)$$

$$\langle \Psi_{i\dots}^a | \hat{\mathbf{H}} \exp(\hat{T}) | \Psi_0 \rangle = E_{\text{CC}} \langle \Psi_{i\dots}^a | \exp(\hat{T}) | \Psi_0 \rangle \quad (9c)$$

where [Equation \(9b\)](#) is known as the *energy equation*, and [Equation \(9c\)](#) are the *amplitude equations*, which are non-linear and require an iterative solution scheme.

1.3.1. Coupled-cluster singles and doubles (CCSD)

When setting the truncation level κ to 2, we obtain the CCSD equations. Proceeding from [Equations \(9b\)](#)

and (9c), one can readily derive the ‘master equations’, involving the excitation operators \hat{T}_i ’s. However in this form, these equations are not amenable to code implementation, due to the lack of explicitly constructable arrays and clearly vectorisable operations. For example, the CCSD ‘master equations’ are given below:

$$E_{CC} = \langle \Psi_0 | \hat{\mathbf{H}}(1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2) | \Psi_0 \rangle \quad (10a)$$

$$Ecct_i^a = \langle \Psi_i^a | \hat{\mathbf{H}}(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_1\hat{T}_2 + \frac{1}{6}\hat{T}_1^3) | \Psi_0 \rangle \quad (10b)$$

$$\begin{aligned} E(t_{ij}^{ab} + t_i^a t_j^b - t_i^b t_j^a) &= \langle \Psi_{ij}^{ab} | \hat{\mathbf{H}}(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \frac{1}{6}\hat{T}_1^3 \\ &+ \frac{1}{24}\hat{T}_1^4 + \hat{T}_2 + \frac{1}{2}\hat{T}_2^2 + \hat{T}_2\hat{T}_1 + \frac{1}{2}\hat{T}_1^2\hat{T}_2) | \Psi_0 \rangle \end{aligned} \quad (10c)$$

There are three, fundamentally equivalent, general strategies to generate so-called factorised, efficient and computer-friendly equations from the master equations:

1. **Configuration-space derivation** is the most ‘accessible’ way to proceed, as it mostly relies on the manipulation of first-quantisation expectation values based on the Slater-Condon rules [7, p. 70]. However, considerable effort is needed in this error-prone process to eliminate unknowns [6, Ch. 9.3.1], which becomes unmanageable quickly, even at the CCSDT level.
2. **Algebraic derivation** relies on the tedious and repeated applications of the generalised Wick’s theorem [8] in second quantisation. This is equivalent to the previous derivation, but the second-quantisation formulation makes it more amenable to string-based computer manipulation [9].
3. **Diagrammatic derivation** is an advanced technique based on the representation of the algebraic derivation with Hugenholtz, or spin-orbital, diagrams. In particular it greatly simplifies the contractions between operators, making errors easier to spot. Furthermore, the number of diagrams scales more favorably than the number of terms in configuration-space or algebraic derivations, due to the linked-diagram theorem [6, Ch. 5.8].

This work has implemented the spin-orbital, or unrestricted, formulation by [10], and the spatial, or more specifically spin-free formulation by [11]. The former is derived directly from the master equations, and the latter is based on a different diagrammatic method using Goldstone, or orbital, diagrams. See [12] for an overview of spin adaptation in coupled-cluster methods. An unpublished orthogonal spin-adaptation of [10] is given in the codebase of [13]¹. It is derived by *spin-integration*

or *spin-summation* techniques, essentially eliminating terms where the spin degrees of freedom would vanish under integration. It is noted in [12] as ‘mathematically satisfying’ but not as efficient.

Starting from RHF references, the two approaches give identical answers, but the latter will be approximately $2^6 = 64$ times faster, as all index ranges are halved in spatial CCSD.

These two references represent some of the most used formulations of CCSD due to their concise, factorised nature, where a small number of intermediates are required, and where all operations fall into a few classes of tensor contractions.

1.3.2. Coupled-cluster singles and doubles with perturbative triples (CCSD(T))

CCSD(T), commonly regarded as the ‘gold standard’ of quantum chemistry, offers energies within chemical accuracy (± 1.6 mHa) for electronic states dominated by a single reference (*e.g.* most closed-shell molecules near equilibrium bond lengths). It is non-iterative and scales as $\mathcal{O}(N^7)$. First proposed by [14], and elaborated in [15], the derivation of the CCSD(T) energy correction is not strictly a unique result from MBPT [16]. As noted in [14, 15], a number of alternative perturbative schemes exist, generally all correct to fourth-order in the MPPT expansion, and only differ in the selective inclusion of the fifth-order terms.

The derivations of the non-iterative perturbative triples methods all start from a formal MBPT (or more specifically, coupled-cluster perturbation theory (CCPT) [5, Ch. 14.3]) analysis of energy expressions in orders of perturbation, and including the lowest order terms that include the effects of triples excitations. These should also be directly available from a CCSD calculation. The same analysis can show that the CCSD energy expression is correct to third order in CCPT, hence any correction would need to come from fourth order onwards. For example, CCSD[T] [18] includes the only such fourth-order term, which is the projection of doubles excitations onto *connected triples* [5, eq. 14.6.49]:

$$E_T^{(4)} = \langle \bar{t}_2^{(1)} | [\hat{\Phi}, \hat{T}_3^{(2)}] | \Psi_0 \rangle \quad (11)$$

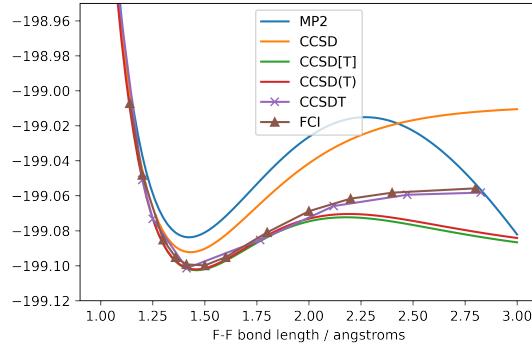
The CCSD(T) scheme further includes two more terms to the fifth order in CCPT that correspond to the projection of singles and doubles onto connected triples [5, eq. 14.6.50] similar to Equation (11).

The CCSD(T) (and CCSD[T]) methods only provide sub-milihartree energies consistently near equilibrium geometries, and in systems free from orbital

¹Accessible at https://github.com/psi4/psi4numpy/blob/master/Coupled-Cluster/RHF/helper_ccenergy.py.

(quasi)degeneracies. Much of chemistry, however, studies the energetics of bond-breaking, or that of exotic molecules (for example, the carbon and chromium dimers) where orbital degeneracies are commonplace. In these cases, CCSD(T)/[T] produces completely unphysical potential energy surfaces (PES) littered with spurious energy barriers, due to divergences in the MBPT denominators when the involved orbitals are (quasi)degenerate, also known as the *intruder state problem* [5, Ch. 14.5].

One such example is shown in [Figure 1](#), for the fluorine dimer, a challenging test case for many quantum chemical methods, despite only having a single chemical bond. Its difficulty is mainly due to the large number of low-lying degenerate orbitals, chemically interpreted as lone pairs.



[Figure 1](#): The energy of the fluorine molecule in the cc-pVDZ basis set as it is being pulled apart. The CCSD(T) and CCSD[T] methods, while performing on par with the full CCSDT method (data from [19] and the MRCC package [20, 21]), which is itself almost equivalent to FCI (data from [22]), near the equilibrium, break down completely in regions with significant multi-reference character. The CCSD method, while free of this defect, has qualitatively wrong energies at all geometries.

Traditionally in these cases, multi-reference methods are used. But a consensus method that unifies the treatment of static and dynamic correlation whilst providing a low scaling has yet to be found. Many theories do exist, such as CASPT2 [23], in which a CASSCF calculation that recovers static correlation is followed by a perturbative treatment to recover dynamic correlation.

A persistent trend, spearheaded by the Piecuch group, has been to use the single reference formalism as much as possible to recover the effects of static correlation. The theory, known as the method of moments of coupled-cluster (MMCC) equations [24], enables the formulation of the renormalised and completely renormalised CCSD(T) and CCSD[T] methods (R-CCSD(T)/[T] and CR-CCSD(T)/[T] respectively)

[25, 11], where an additional, *a posteriori* correction to the (T) or [T] energy corrections is introduced in the form of a denominator. This is done by arguments from the ‘Fundamental Theorem of the Formalism of β -nested Equations’ [25], which gives the exact energy difference between CC and FCI energies:

$$\delta = E_{\text{FCI}} - E_{\text{CC}}^{(m_A)} \\ = \sum_{n=m_A+1}^N \sum_{j=m_A+1}^n \frac{\langle \Psi^{\text{FCI}} | Q_n C_{n-j}(m_A) M_j(m_A) | \Psi_0 \rangle}{\langle \Psi^{\text{FCI}} | e^{T_{m_A}} | \Psi_0 \rangle} \quad (12)$$

where m_A is the cluster truncation, Q_n is the projector onto the n -tuples manifold, $M_j(m_A) = (H_N e^{T_A})_{C,j}$ where j means the j -particle component, and the C subscript restricts the expression to its connected terms, and $C_i(m_A) = (e^{T_A})_i$.

The use of an approximation to Ψ^{FCI} will still result in an improvement in the correlation energy, as $0 \leq \tilde{\delta} \leq \delta$, with $\tilde{\delta}$ being the inexact correction. Further, the first sum in [Equation \(12\)](#) can be truncated to m_B , resulting in the MMCC(m_A, m_B) formalism. For example, working in MMCC(2,3), using $\Psi^{\text{CCSD}[T]} \approx \Psi^{\text{FCI}}$, and approximating $M_3(2)$, the R-CCSD[T] energy is given by:

$$\Delta E^{\text{R-CCSD}[T]} = \frac{\langle \Psi^{\text{CCSD}[T]} | Q_3(V_N T_2)_C | \Psi_0 \rangle}{\langle \Psi^{\text{CCSD}[T]} | e^{T_1 + T_2} | \Psi_0 \rangle} \quad (13)$$

where V_N is the normal-ordered electron-repulsion operator. In all, the numerator is the same quantity as in [Equation \(11\)](#).

These denominators ($D^{[T]}$ and $D^{(T)}$) reflect the extent of the multi-reference character of the system, approaching 1 from above in the completely single-reference case. These are reported when available, together with the so-called \mathcal{T}_1 diagnostic [26] that serves a similar purpose:

$$\mathcal{T}_1 = \frac{\| \mathbf{t}_1 \|_F}{\sqrt{N}} \quad (14)$$

where \mathbf{t}_1 is the array of $\hat{\mathbf{T}}_1$ amplitudes, F indicates the Frobenius norm, and N is the total number of electrons.

More recently, this approach has been taken to its logical conclusion with the CR-CC(m_A, m_B) methods [28], where m_A is the truncation being improved upon, and m_B is the highest truncation being perturbatively treated. These approaches use a new, biorthogonal form of the MMCC equations with the ‘fundamental theorem’ mentioned above. CR-CC(2, 3) was shown to be rigorously size-extensive, unlike the MMCC(2,3)-based CR-CCSD(T) and related methods, and provide energies

competitive with full triples in the bond-breaking region, far exceeding the capabilities of standard CCSD(T), and those of its predecessors from the same authors.

2. Code implementation

2.1. Compilation and usage

The main executable, `els.x`, can be compiled with CMake by means of a shell script, `make.sh`, which has a debug/production toggle. The only external dependency, in addition to a working Fortran compiler, is a multi-threaded implementation of BLAS/LAPACK [43], such as OpenBLAS [44] or Intel MKL, and the Python-based pre-processor Fypp [45]. A Python utility script, `utils/psi4_integrals_nosym.py`, is provided to generate the geometry file and one- and two-electron atomic orbital (AO) integrals needed from the open-source Psi4 package [46] in Python, and a further wrapper script `utils/els_wrapper.py` is provided to automate calculations, along binding curves, for example. All important functionalities of the program, such as levels of theory, and various thresholds, can be controlled at run-time via a Fortran namelist file `els.in`, whose format is shown in Listing 1. This input file should be stored together with the integral files, and the program will read these files in when invoked in the same directory.

Listing 1: An example input file to `els.x`.

```
&elsinput
calc_type="CRCCSD(T)_spatial",
scf_e_tol=1e-6,
scf_d_tol=1e-7,
scf_diis_n_errmat=6,
ccsd_e_tol=1e-6,
ccsd_t_tol=1e-7,
ccsd_diis_n_errmat=8,
scf_maxiter = 50,
ccsd_maxiter = 50,
write_fcidump = .false.,
scf_read_guess = .false.,
scf_write_guess = .true.
/
```

2.2. Capabilities

This program can perform restricted Hartree Fock (RHF) and post-HF calculations starting from RHF references. RHF itself is accelerated by the DIIS method [47], and can read in initial guesses from previous geometries. MP2 calculations are coded with the standard, optimised series of $\mathcal{O}(N^5)$ loops instead of the naive $\mathcal{O}(N^8)$ loop [48, p. 131]. CCSD can be performed in either the spin-orbital formalism of [10], or

the spin-free formalism of [11], both accelerated by DIIS [49]. CCSD(T) can also be performed in the spin-orbital [17] or spin-free [11] forms. A family of triples correction methods from [11] are also implemented, including CCSD[T], renormalised CCSD[T] and CCSD(T), and completely renormalised CCSD[T] and CCSD(T), the latter four methods are currently exclusively available in GAMESS [50]. All of the code is accelerated by OpenMP [51], and the spin-orbital CCSD(T) code can be separately accelerated by OpenACC [52] on GPUs.

A tutorial-style website² was launched during the preparation of this report. Though in early stages, it eventually aims to illustrate many concepts in quantum chemistry, modern Fortran and scientific computing contained in this report.

2.2.1. Correct implementation of spin-free CCSD

Two typos in the spin-free CCSD paper [11] were painstakingly identified thanks to the rapid prototyping tools provided by Psi4NumPy [13] and the `numpy.einsum` function. These typos have been confirmed by the lead author and may lead to an erratum (private communications with Prof. Piotr Piecuch). The typos appear in Table 1 of [11]:

$$I_j^{ri} = (1 - \delta_j^i) f_j^i + 2v_{je}^{im} t_m^e - v_{ej}^{im} t_m^e + v_{ef}^{mi} (2t_{mj}^{ef} - t_{mj}^{fe}) + v_{ef}^{mi} t_{mj}^{ef} - v_{ef}^{im} t_{mj}^{ef} \quad (15a)$$

$$I_{bj}^{ia} = v_{bj}^{ia} + (v_{be}^{im} - \frac{1}{2}v_{eb}^{im}) t_{mj}^{ea} + v_{be}^{im} t_{mj}^{ea} - \frac{1}{2}v_{mb}^{ie} t_{jm}^{ae} - \frac{1}{2}v_{be}^{im} c_{mj}^{ae} + v_{be}^{ia} t_j^e - v_{bj}^{im} t_m^a \quad (15b)$$

2.3. Code layout

The source code can be found in the `src/` directory, and is laid out as follows

```
src/
  ├── ccisd.f90 ..... Restricted and unrestricted CCSD and CCSD(T)
  ├── const.F90 ..... Numeric types and constants
  ├── error_handling.f90 ..... Exception handler functions
  ├── geometry.f90 ..... Geometry read-in and calculations
  ├── hf.f90 ..... Hartree-Fock
  ├── integrals.f90 ..... Integral read-in and utilities
  ├── linalg.f90 ..... BLAS wrappers and utilities
  ├── main.F90 ..... Main program interface
  ├── mp2.f90 ..... Restricted MP2
  └── system.f90 ..... Input read-in and system derived type
```

²<https://brianz98.github.io/Modern-Electronic-Structure-with-Modern-Fortran/src/index.html>.

2.4. Tensor operations

The coupled-cluster equations in both [10] and [11] are reduced to a series of tensor contractions, and as such a general strategy of performing them needs to be devised.

A general tensor contraction can be represented as $A_i^a = B_{ei}^{mn} C_{mn}^{ea}$, or more simply as `eimn, mnea -> ia`, which we categorise as an ‘order-(P, Q)’ general tensor contraction, where P is the number of uncontracted, or physical, indices, and Q is the number of contracted, ‘dummy’, or virtual, indices, here $P = 2$ and $Q = 3$. The computational complexity is $\mathcal{O}(N^{P+Q})$.

2.4.1. Existing libraries

Libraries that perform optimised tensor contractions are becoming increasingly common due to their applications in machine learning, quantum computing, and quantum chemistry. For example, the now-famous `numpy.einsum` [53], or `opt_einsum` [54] libraries in Python, the `xtensor` [55], `taco` [56], `ambit` [57] libraries in C++, and the `ttb` [58] (Tensor Toolbox for Modern Fortran) package for Fortran.

However, in the interest of understanding the underlying algorithms of these libraries, we decided to hand-code all of the tensor contractions, developing a heuristic of best practice along the way.

We found that the general strategy employed in the abovementioned packages include a ‘path-finding’ algorithm to find the optimal ‘contraction path’ (see the `opt_einsum` documentation [54] for a good overview). This generally involve reshaping arrays that align the indices in a way that can be worked on by the all-important BLAS routines `?gemm`, which are detailed in the following section.

2.4.2. General matrix multiply

`?gemm` stands for ‘general matrix multiply’, with the `?` standing for one of `s`, `d`, `c` or `z` for single/double precision real/complex type.

The `?gemm` routines are powerful. Their superior performance derives from cache optimisations and their divide-and-conquer algorithm. They are also versatile, in that they allow generalised matrix multiplications for arrays of arbitrary dimensionalities, realised by the Fortran argument specifications of `dimension(1d?,*)` (`1d?` being the leading `dim`ension of one of the matrices) for the matrices. The use of the assumed-size arguments enables arrays of arbitrary dimensions to be passed in. As long as the contracted dimensions are aligned in memory layout, the multiplication will be performed correctly.

To exemplify, a contraction $C_{ij}^{ab} = A_{ij}^{mn} B_{mn}^{ab}$, with `dim_x` denoting the dimension along the `x` index, and the arrays laid out in default Fortran style (*i.e.*, `A(dim_i,dim_j,dim_m,dim_n)`), can be performed by the call in Listing 2.

Listing 2: An example `?gemm` call.

```
call ?gemm('N', 'N', &
           dim_i*dim_j, dim_a*dim_b, dim_m*dim_n, &
           1.0_p, &
           A, dim_i*dim_j, &
           B, dim_m*dim_n, &
           0.0_p, &
           C, dim_i*dim_j)
```

2.4.3. Tensor benchmarks and heuristics

Fortran excels at array-based numerical computations as there exist a number of intrinsic array manipulation functions and idioms. For example, the `reshape()` intrinsic performs a general reshape of high-dimensional arrays; the `sum()` performs reductions along selected axes or the entire array; the arithmetic operators are all overloaded with element-wise array operations.

These tools, combined with direct `do` loops, which can be multi-threaded with ease with OpenMP (see Section 2.5), provide an alternative to `?gemm` for all tensor contractions, especially when memory is limited, as `?gemm` can require multiple temporary arrays to hold reshaped arrays. For example, a contraction $I_a^i = v_{mi}^{ea} t_m^e$ can be performed as in Listing 3.

Listing 3: An example of an order-(2 2) tensor contraction.

```
do i = 1, nocc
  do a = 1, nvirt
    I_vo(a,i) = sum(v_oovv(:,i,:,a)*t1)
  end do
end do
```

To investigate the performance and scaling properties of these different strategies, a benchmark study was undertaken, as shown in Figure 2, where six most common classes of tensor contractions in CCSD, detailed in Table 1, is investigated³. The scaling curves offer a few clear lessons:

- The Fortran-specific OpenMP `workshare` construct does not work, at least with GNU compilers,

³After the conclusion of this scaling study it was realised that the ‘tensor dot’ operation can be performed with `?gemm` as well, with the correct leading dimension filled in as ‘1’, and this leads to faster contractions. Therefore this does not affect the conclusions here that `?gemm` should be used whenever possible.

- The Fortran intrinsic element-wise operations are not competitive with explicit loops, when both are in OpenMP parallel regions,
- `dgemm` is always superior to hand-coded loops when no reshapes are required,
- Reshapes are never the rate-limiting steps, and can be accelerated by OpenMP if desired, therefore `dgemm` is always the right choice.

Table 1: Types of tensor contractions in Figure 2

	Path	Order
Matmul	<code>im,ma->ia</code>	(2, 1)
Tensor dot	<code>em,miea->ia</code>	(2, 2)
Tensor contraction 1	<code>eimn,mnea->ia</code>	(2, 3)
Tensor contraction 2	<code>ijae,eb->ijab</code>	(4, 1)
Tensor contraction 3	<code>imab,jm->ijab</code>	(4, 1)
Tensor contraction 4	<code>mbie,jmae->bjia</code>	(4, 2)

2.5. Parallelisation strategies

The code detailed in this report is parallelised on CPU by OpenMP and threaded BLAS/LAPACK libraries, and on GPU for the spin-orbital CCSD(T) code by OpenACC. A review of MPI-based parallelism as used in quantum chemistry packages is also given.

2.5.1. Shared memory parallelism

OpenMP [51] offers directive-based, lightweight and shared memory parallelism based on the concept of multi-threading. It is easy to build on top of existing serial code. For this reason, OpenMP was chosen to be the base layer of parallelism in the program.

The most time-consuming part of the HF-SCF procedure is building the Fock matrix, given by

$$F_{\mu\nu}^{(i)} = H_{\mu\nu}^{\text{core}} + \sum_{\sigma\tau} D_{\sigma\tau}^{(i-1)} [2(\mu\nu|\sigma\tau) - (\mu\sigma|\nu\tau)] \quad (16)$$

This can be readily translated into Fortran do loops as shown in Listing 4.

Listing 4: The Fock-build code can be easily accelerated by OpenMP.

```
!$omp parallel do default(None) &
!$omp schedule(static, 10) collapse(2) &
!$omp shared(sys, density, fock, eri)
do j = 1, sys%nbasis
  do i = 1, sys%nbasis
    fock(i,j) = int_store%core_hamil(i,j)
    do l = 1, sys%nbasis
      do k = 1, sys%nbasis
        fock(i,j) = fock(i,j) + density(k,l) &
                     * (2*eri(i,j,k,l) - eri(i,k,j,l))
      end do
    end do
  end do
end do
!$omp end do
```

The OpenMP directives are completely self-contained and do not affect the serial code, and can be enabled at compile-time. Direct do-loop OpenMP acceleration is used when the arrays are not directly laid out in memory that are amenable to `?gemm` operations. In this case, the `eri` array is actually a lookup function for the electron-repulsion integrals stored in a 1-D array due to the high degree of permutational symmetry present in the tensor.

If the arrays are normal Fortran arrays, we can directly use a threaded BLAS library. OpenBLAS and Intel MKL are two options that have been extensively tested. In the case of OpenBLAS, an explicitly OpenMP-threaded version of it can be built by invoking `make USE_OPENMP=1`; and in the case of Intel MKL, a thorough ‘link-line advisor’ is provided on the Intel website⁴.

2.5.2. Distributed memory parallelism

In contrast to OpenMP, which is a shared memory model, message-passing based models assume distributed memory, where processes have isolated memory spaces, and information can only be exchanged by message passing. Popular implementations of message passing interface (MPI) include OpenMPI [59] and MPICH [60]. Mixtures of both paradigms exist, known as the *partitioned global address space* (PGAS) paradigm, including Coarray Fortran [61] and Global Arrays [62], and these are in general more flexible than the standard MPI in scientific computing.

As the code implemented in this work consists overwhelmingly of tensor contractions, and as discussed in Section 2.4.2, a BLAS/LAPACK implementation is able to carry out the most computationally expensive parts of

⁴See <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl-link-line-advisor.html>.

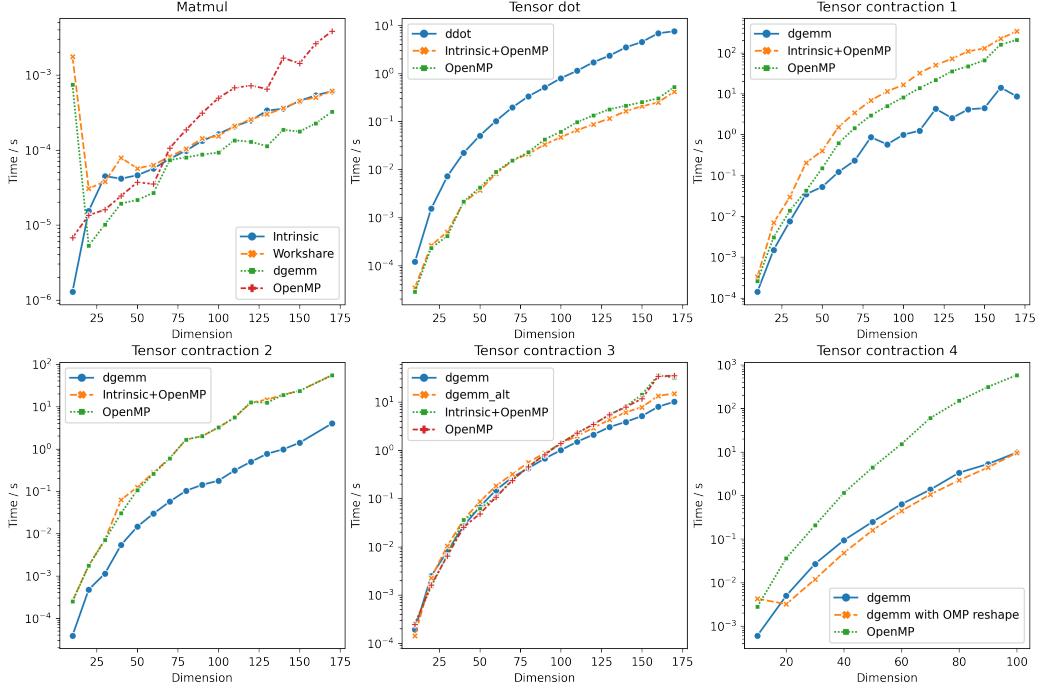


Figure 2: Timing comparisons of some common types of tensor contractions in CCSD(T). ‘intrinsic’ uses the single-threaded Fortran ‘matmul’ function, or uses the `sum` function to contract dummy indices in OMP loops; ‘workshare’ wraps the `parallel workshare` directive around the `matmul`; ‘dgemm’ calls a threaded `dgemm` routine; ‘OpenMP’ is direct `do` loops parallelised by OpenMP; ‘dgemm_alt’ uses an alternative `reshape` sequence; and ‘OMP reshape’ uses OpenMP to reshape the 4-dimensional arrays.

the code, a natural way to parallelise this portion of the code would be to use a MPI-parallelised implementation of BLAS/LAPACK, instead of a threaded one.

ScaLAPACK [63] does exactly this, with the MPI communications handled by a higher level library called BLACS (basic linear algebra communication subprograms). ScaLAPACK distributes arrays in a *two-dimensional block-cyclic* fashion (search keyword within [63]), which achieves optimal load balancing and the use of BLAS routines on the local array on each processor.

Most of the difficulty in implementing this however comes from the frequent need to reshape tensors, which in general requires the entire tensor to be available, resulting either in temporary copies of the entire tensor on all processors, against the spirit of MPI programming; or in complicated communication patterns that are hard to write and maintain.

Higher-level libraries that handle these difficulties do exist, typically using the PGAS paradigm: one that is used in the popular quantum chemistry packages NWChem [64] is Global Arrays (GA) [62], which abstracts away the low-level MPI communications and data distribution. Another library, used by the quan-

tum chemistry package GAMESS [50], is an in-house library known as the Distributed Data Interface (DDI) [65, 66], which uses ‘data servers’ that reside on each node for inter-node communication of distributed data, and shared-memory parallelism intra-node, together with user-specifiable replicated data for faster access of small tensors. However, even with the help of these libraries, the decisions of data locality and layout, and which parallelisation layer to use, are entirely non-trivial: see the detailed parallelisation schemes for a resolution of identity (RI) CCSD(T) code in [67].

2.5.3. GPU acceleration

Quantum chemistry has been a target of GPU acceleration due to the heavy use of tensor operations: GPU accelerated quantum chemistry packages include BrianQC [68], TeraChem [69, 70, 71], and NWChem [72, 73]. These typically incorporate bespoke kernels that require extensive tuning of parameters like block structure. However, there has been a proliferation of high-level libraries or directive-based paradigms that eschew the need for hand-coding device kernels. Well-known libraries include the NVIDIA math libraries like cuBLAS and cuTENSOR, both having APIs with C++

and Fortran; and even higher level libraries in interpreted languages, like CuPy [74] and TensorFlow [75] in Python. Directive-based GPU programming models include OpenACC [52], NVIDIA HPC SDK’s ‘standard parallelism’ support for NVIDIA GPU offloading via the Fortran do concurrent directives [76], and, recently, OpenMP (known as ‘target offloading’).

Of these strategies, OpenACC offers the best balance between the ease of implementation and flexibility, and an initial implementation is therefore pursued for the most computationally expensive part of the code, the spin-orbital CCSD(T) module.

The CPU-based CCSD(T) algorithm is a ‘batched’ algorithm as described in [77] (see Listing 5). This algorithm reduces the memory cost of storing a six-dimensional array and still takes advantage of vectorised instructions and cache efficiency of operating on contiguous arrays. To adapt this to GPU, which is inherently massively parallel, we can dispense with the temporary array and let each loop iteration be independent. Listing 6 shows the remarkable simplicity OpenACC offers for GPU programming, with fine grained control of block geometries.

Listing 5: The batched CCSD(T) algorithm.

```
e_T = 0.0
do i = 1, nocc
  do j = 1, nocc
    do k = 1, nocc
      t3(:,:,:,:) = 0.0
      do a = 1, nvirt
        do b = 1, nvirt
          do c = 1, nvirt
            t3(a,b,c) = comp_t3(i,j,k,a,b,c)
          end do
        end do
      end do
      e_T = e_T + t3_contribution(t3(:,:,:,:))
    end do
  end do
end do
```

Listing 6: The OpenACC CCSD(T) algorithm.

```
e_T = 0.0
!$acc data copyin(...)
!$acc parallel loop gang collapse(3) private(t3)
  reduction(+:e_T)
do i = 1, nocc
  do j = 1, nocc
    do k = 1, nocc
      !$acc loop worker collapse(3)
      do a = 1, nvirt
        do b = 1, nvirt
          do c = 1, nvirt
            t3 = comp_t3(i,j,k,a,b,c)
            e_T = e_T + t3_contribution(t3)
          end do
        end do
      end do
    end do
  end do
end do
!$acc end data
```

2.6. Computational complexity

A formal analysis of computational complexity of the code, as implemented, is presented in Table 2, where n_o is the number of occupied spatial orbitals, and n_v is the number of virtual spatial orbitals, and n_t is the total number of spatial orbitals. It is assumed that $n_v > n_o$. The memory requirements can in general be traded with increased computational cost. For example, in the spatial CCSD code, quantities of the type $(2\langle ij|ab\rangle - \langle ji|ab\rangle)$ appear frequently, we could choose to store all ‘antisymmetrised’ integrals of different types. As there are many permutations of orbital types, storing all of them would be prohibitive, hence they were only generated when needed.

3. Results

3.1. Breaking single bonds

3.1.1. Water

The water molecule is one of the most used model systems for quantum chemistry due to its small size, lack of orbital degeneracies, and the central role it plays in life. However, the C_{2v} symmetric stretch involving simultaneous breaking of two bonds still proves challenging for most standard theories. We calculated the C_{2v} binding curve of water/cc-pVDZ at a bond angle of 104.45° with our code, GAMESS, and Psi4, with results presented in Figure 3.

We can see that our code produces virtually the same results as those from GAMESS and Psi4, showing the correctness of our implementation. Identical results

Table 2: Formal computational complexity analysis of different modules in the code. RDS: rate determining step; mem.: memory

Subroutine	RDS	Computational complexity	Largest mem.	Memory complexity
do_rhf	Fock build	$\mathcal{O}(n_t^4)$	ERI storage	$\mathcal{O}(n_t^4)$
do_mp2_*	AO to MO transformation	$\mathcal{O}(n_t^5)$	Partially transformed temporary arrays	$\mathcal{O}(n_t^4)$
do_ccsd_*	t_2 update	$\mathcal{O}(n_o^2 n_v^4)$	ERI slices	$\mathcal{O}(n_v^4)$
do_ccsd_t_*	Connected t_3 amplitudes	$\mathcal{O}(n_o^3 n_v^4)$	ERI slices	$\mathcal{O}(n_o n_v^3)$

from Psi4, which uses different algorithms, show that this is not due to algorithmic quirks. Care was taken to track the RHF states in all calculations, by restarting from orbitals from the previous geometry, to make sure we did not converge onto low-lying symmetry-broken states.

The divergences could be caused by the intruder state problem, which can sometimes be circumvented by inserting a level shift in the MPBT denominators [78], or other more sophisticated schemes like in [79]. However, the fact that MP2 diverges in the opposite direction as other methods is surprising, and subtracts from the intruder state hypothesis. In fact, our binding curves match closely with those in [5, Ch. 5], where the divergence in MP2 is attributed to a divergence in the perturbation series instead of a simple numerical problem.

3.1.2. Fluorine

Continuing on the discussion from Section 1.3.2 and Figure 1, R- and CR-CCSD(T)/[T] results are presented for F₂ in Figure 4, together with the \mathcal{T}_1 diagnostic, the $D(T)$ and $D[T]$ denominators, all of which reflect the extent of multi-reference character of the system. In this case, it appears that the lower-level R-CCSD[T] is the closest to the FCI, but at the equilibrium region, CR-CCSD(T) has the least deviation from FCI. Moreover, the inset shows that the renormalised methods in fact reach noticeable maxima in the bond-breaking region, which are almost imperceptible in the completely-renormalised methods.

3.2. Breaking multiple bonds

3.2.1. Nitrogen

Moving from singly-bonded molecules, the triply-bonded nitrogen is another archetypal example of strong correlation. Due to extensive s-p mixing in N₂, the three highest occupied and three lowest unoccupied molecular orbitals (MOs) are very close in energy, and all contribute significantly to the ground-state wavefunction. As such, it is expected that at least a CCSDTQ56 calculation is required to produce good agreement with FCI, and as such, even the corrected CCSD(T) methods are

expected to fail, and indeed we see in Figure 5 that even CR-CCSD(T) has a dramatic, unphysical hump.

Stretched nitrogen is also known to be hard for iterative methods to converge, and in fact both Psi4 and GAMESS failed to converge N₂ at bond lengths longer than 2.3 angstroms for CCSD. Another difficulty is with RHF locating multiple solutions due to poor initial guesses, see Figure 6 (the ‘wrong’ solutions generally break spatial symmetry [80], and have seen extensive use in *non-orthogonal configuration interaction* (NOCI) [81]). Our code is able to remedy this by using the AO basis Fock matrix from a previous geometry as the initial guess, instead of using the bare Hamiltonian, \mathbf{H}^{core} . More sophisticated methods used in other packages include the superposition of atomic densities (SAD) [82] or potentials (SAP) [83].

3.2.2. Carbon dimer

The carbon dimer, and the nature of its bonding, has been a source of fascination for quantum chemists [84, 85, 86, 87]. A second-row homonuclear molecule, like N₂ and F₂, the formal bond order of C₂ is still under debate; and unlike N₂, C₂ is strongly multi-reference even at the equilibrium bond length (see the diagnostics in Figure 7). Interestingly, the \mathcal{T}_1 diagnostic shows an unusual shoulder feature in the intermediate region. This region has been identified by [88] to contain an avoided crossing, *i.e.*, strong quasidegeneracies, which may explain the observed local maximum. Figure 7 also shows density matrix renormalisation group (DMRG) results from [88]. DMRG is an approximate FCI solver whose results can be systematically extrapolated to FCI results, which is done in this reference. We can observe that none of the CC methods can even qualitatively describe the intermediate, strongly quasidegenerate region, and attain maximum errors of exceeding 50 mE_h.

3.3. Parallel performance

We finally examine the parallel speedup of the code. As shown in Figure 8, we observe that, although present, the parallel speedup is significantly sublinear, which does not bode well for massive parallelisation. The

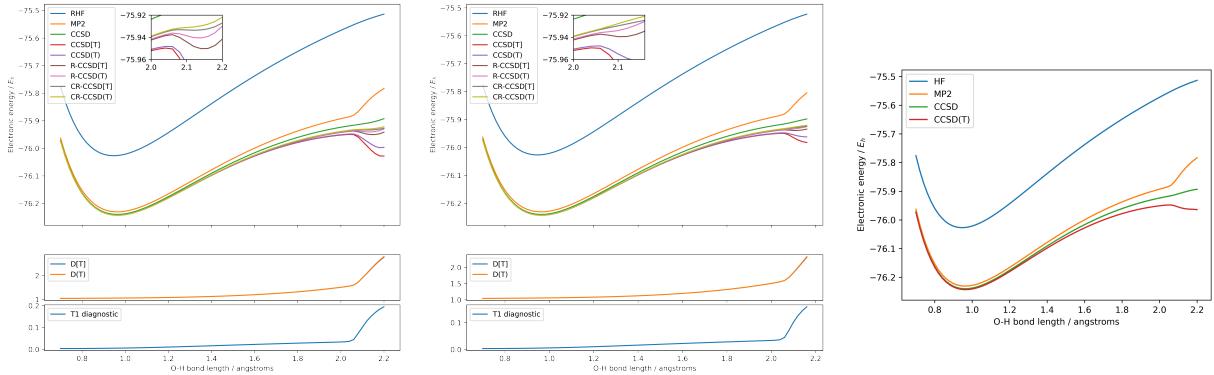


Figure 3: The first and second figures show results from our code and that of GAMESS, respectively, whereas the third comes from Psi4. In this and following figures, the denominators $D[T]$ $D(T)$ are both plotted but are too close to discern.

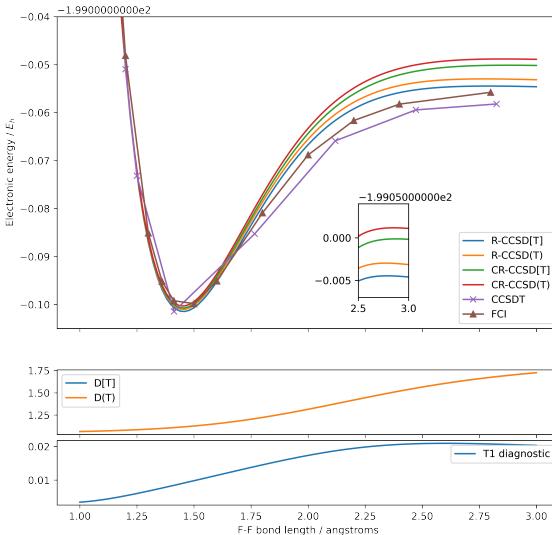


Figure 4: The performance of the renormalised and completely-renormalised methods on $F_2/\text{cc-pVDZ}$ are in stark contrast with those in Figure 1. This is intuitively understood from the $D(T)$ and $D[T]$ denominators becoming larger in the bond-breaking region, attenuating the unphysical humps present in standard CCSD(T) and [T] binding curves.

cause of this is most evident from the CCSD curve, which appears to obey Amdahl’s law [89], suggesting that the unparallelised or inefficiently parallelised portion of the code is already throttling the performance at a low thread count. The rate-limiting portion could be the repeated serial allocation and deallocation of temporary arrays, but more detailed profiling is needed to ascertain this. For a moderately large calculation (all-electron $H_2O/\text{cc-pVTZ}$ with $r_{O-H} = 2.0$ angstroms, 58 basis functions, running in C_1 symmetry), our code compares favourably with the state-of-the-art Psi4 package in tim-

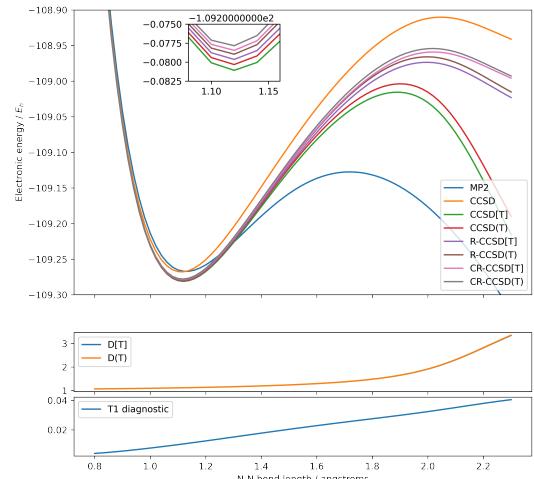


Figure 5: The binding curve of $N_2/\text{cc-pVDZ}$ at different levels of theory, showing the dramatic failings of low-truncation, single-reference methods. The energy denominators surge to almost 3 at the tail-end, signifying a very strong multi-reference character.

ings (Figure 9).

The GPU acceleration, although only implemented in one module, brings promising speedups, especially for medium-to-large systems, here a $75\times$ speedup is obtained compared to a single-threaded execution on the same H_2O system (see Figure 10).

4. Future work

This report contains many of the main components of a modern quantum chemistry package, but some in-

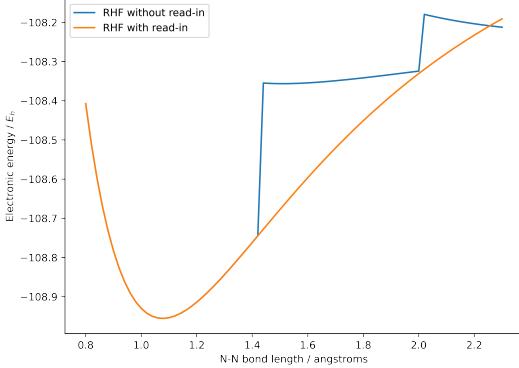


Figure 6: The failure of using core Hamiltonian guesses is laid bare in $\text{N}_2/\text{cc-pVDZ}$, where higher- (and lower-)energy HF solutions are located for stretched geometry, regardless of number of DIIS error matrices.

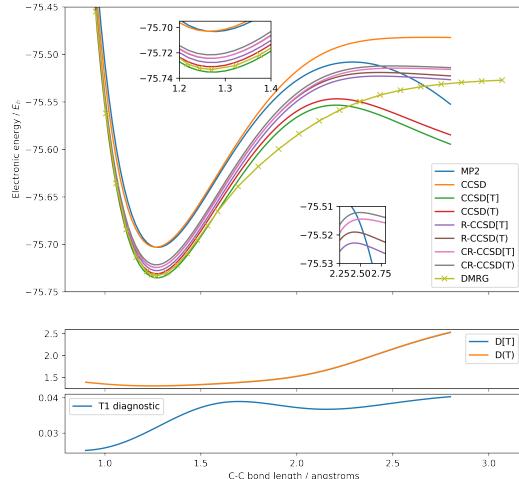


Figure 7: The binding curve of $\text{C}_2/\text{cc-pVDZ}$ at different levels of theory. The bottom inset accentuates that the renormalised methods are not completely satisfactory.

ingredients are missing and merit future work in understanding their workings. These include, but are not limited to, unrestricted Hartree-Fock (UHF) and support for open-shell calculations in higher levels of theory; an integral interface, for example with libcint [90] to generate of one- and two-electron integrals; investigating code-generation frameworks (like that of taco [56]) for the tedious and error-prone process of tensor contraction loops; interfacing with tensor libraries mentioned in this

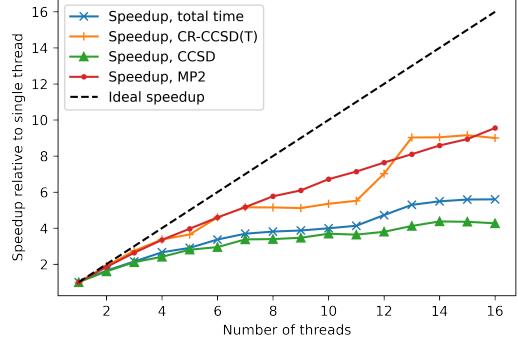


Figure 8: Parallel speedup of different modules of the code, there is a significant deviation from strong scaling.

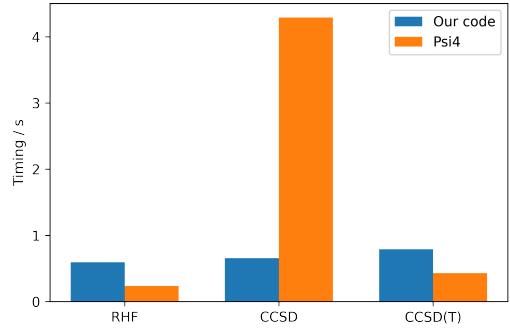


Figure 9: The wall-clock timing comparison between our code and Psi4, both run with 8 hardware threads.

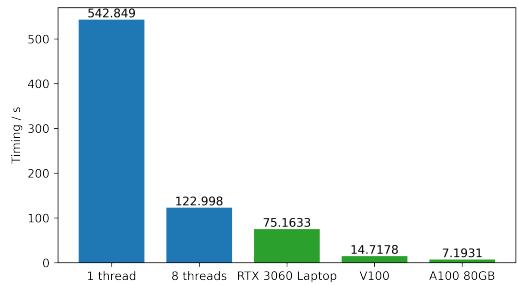


Figure 10: Timings for the spin-orbital CCSD(T). An AMD Ryzen 7 5800H was used for the CPU execution.

report; using CUDA libraries, especially cuTENSOR, for GPU acceleration; exploiting MPI for truly massive parallelism; and exploiting point-group symmetry, see libdpd (**d**irect **p**roduct **d**ecomposition) in Psi4 [46]; and the use of out-of-core algorithms/disk scratch files for

larger calculations.

5. Conclusions

In this report, a review of basic modern electronic structure theory was given. The details of a program implementing some of these theories were given, including a survey of best practices of tensor contraction and parallelisation. Finally, binding curves of some archetypal systems were presented, showing first of all the correctness and efficiency of our code, and also demonstrating the failings of the standard theories.

Acknowledgements

I would like to thank Maria-Andreea Filip and Nicholas Lee for their helpful code review, discussions, and help in reviewing this report, the rest of the Thom group for their unwavering support, and my supervisor, Dr Alex Thom, for his kind words in times of stress and helpful discussions. I would also like to thank Prof. T. Daniel Crawford's group for their quantum chemistry programming tutorial [91], without which this project would have seemed much more daunting. Finally I would like to graciously acknowledge Prof. David Wales for access to his GPU computing resources.

Appendix A. The HF-SCF algorithm

```

 $\mathbf{F}^{(0)} \leftarrow F_0;$ 
 $\mathbf{D}^{(0)} \leftarrow 0;$ 
 $E^{(0)} \leftarrow 0;$ 
 $i \leftarrow 1;$ 
while True do
     $\mathbf{F}'^{(i)} \leftarrow ort(\mathbf{F}^{(i)});$ 
     $\mathbf{C}'^{(i)}, \epsilon^{(i)} \leftarrow eig(\mathbf{F}'^{(i)});$ 
     $\mathbf{C}^{(i)} \leftarrow ort^{-1}(\mathbf{C}'^{(i)});$ 
     $\mathbf{D}^{(i)} \leftarrow (\mathbf{C}_{nocc}^{(i)})^T \mathbf{C}_{nocc}^{(i)};$ 
     $E^{(i)} \leftarrow \sum_{\mu\nu} D_{\mu\nu}^{(i)} (H_{\mu\nu}^{\text{core}} + F_{\mu\nu}^{(i)});$ 
    if  $abs(E^{(i)} - E^{(i-1)}) < e\_tol$  and  $norm(\mathbf{D}^{(i)} - \mathbf{D}^{(i-1)}) < d\_tol$  then
        | break;
    end
     $F^{(i+1)} \leftarrow H_{\mu\nu}^{\text{core}} + \sum_{\sigma\tau} D_{\sigma\tau}^{(i)} [2(\mu\nu|\sigma\tau) - (\mu\sigma|\nu\tau)];$ 
     $i \leftarrow i + 1;$ 
end

```

Algorithm 1: The HF-SCF algorithm. *ort* is an orthogonalisation procedure [7, p. 142], and *eig* is a (hermitian) diagonalisation procedure.

References

- [1] J. P. Perdew, K. Schmidt, **Jacob's ladder of density functional approximations for the exchange-correlation energy**, AIP Conference Proceedings 577 (1) (2001) 1–20, publisher: American Institute of Physics. doi:[10.1063/1.1390175](https://doi.org/10.1063/1.1390175). URL <http://aip.scitation.org/doi/10.1063/1.1390175>
- [2] C. C. J. Roothaan, **New Developments in Molecular Orbital Theory**, Reviews of Modern Physics 23 (2) (1951) 69–89, publisher: American Physical Society. doi:[10.1103/RevModPhys.23.69](https://doi.org/10.1103/RevModPhys.23.69). URL <https://link.aps.org/doi/10.1103/RevModPhys.23.69>
- [3] G. G. Hall, J. E. Lennard-Jones, **The molecular orbital theory of chemical valency VIII. A method of calculating ionization potentials**, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 205 (1083) (1951) 541–552, publisher: Royal Society. doi:[10.1098/rspa.1951.0048](https://doi.org/10.1098/rspa.1951.0048). URL <https://royalsocietypublishing.org/doi/10.1098/rspa.1951.0048>
- [4] C. Møller, M. S. Plesset, **Note on an Approximation Treatment for Many-Electron Systems**, Physical Review 46 (7) (1934) 618–622, publisher: American Physical Society. doi:[10.1103/PhysRev.46.618](https://doi.org/10.1103/PhysRev.46.618). URL <https://link.aps.org/doi/10.1103/PhysRev.46.618>
- [5] T. Helgaker, P. Jorgensen, J. Olsen, **Molecular electronic-structure theory**, John Wiley & Sons, 2014.
- [6] I. Shavitt, R. J. Bartlett, **Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory**, Cambridge Molecular Science, Cambridge University Press, Cambridge, 2009. doi:[10.1017/CBO9780511596834](https://doi.org/10.1017/CBO9780511596834). URL <https://www.cambridge.org/core/books/many-body-methods-in-chemistry-and-physics/D12027E4DAF75CE8214671D842C6B80C>
- [7] A. Szabo, N. S. Ostlund, **Modern quantum chemistry: introduction to advanced electronic structure theory**, Courier Corporation, 2012.
- [8] G. C. Wick, **The Evaluation of the Collision Matrix**, Physical Review 80 (2) (1950) 268–272, publisher: American Physical Society. doi:[10.1103/PhysRev.80.268](https://doi.org/10.1103/PhysRev.80.268). URL <https://link.aps.org/doi/10.1103/PhysRev.80.268>
- [9] F. A. Evangelista, **Wicked**. URL <https://github.com/fevangelista/wicked>
- [10] J. F. Stanton, J. Gauss, J. D. Watts, R. J. Bartlett, **A direct product decomposition approach for symmetry exploitation in many-body methods. I. Energy calculations**, The Journal of Chemical Physics 94 (6) (1991) 4334–4345, publisher: American Institute of Physics. doi:[10.1063/1.460620](https://doi.org/10.1063/1.460620). URL <http://aip.scitation.org/doi/abs/10.1063/1.460620>
- [11] P. Piecuch, S. A. Kucharski, K. Kowalski, M. Musiał, **Efficient computer implementation of the renormalized coupled-cluster methods: The R-CCSD[T], R-CCSD(T), CR-CCSD[T], and CR-CCSD(T) approaches**, Computer Physics Communications 149 (2) (2002) 71–96. doi:[10.1016/S0010-4655\(02\)00598-2](https://doi.org/10.1016/S0010-4655(02)00598-2). URL <https://www.sciencedirect.com/science/article/pii/S0010465502005982>
- [12] D. A. Matthews, J. Gauss, J. F. Stanton, **Revisitation of Nonorthogonal Spin Adaptation in Coupled Cluster Theory**, Journal of Chemical Theory and Computation 9 (6) (2013)

- 2567–2572, publisher: American Chemical Society. doi:10.1021/ct301024v.
 URL <https://doi.org/10.1021/ct301024v>
- [13] 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, C. D. Sherrill, **Psi4NumPy: An Interactive Quantum Chemistry Programming Environment for Reference Implementations and Rapid Development**, Journal of Chemical Theory and Computation 14 (7) (2018) 3504–3511, publisher: American Chemical Society. doi:10.1021/acs.jctc.8b00286.
 URL <https://doi.org/10.1021/acs.jctc.8b00286>
- [14] K. Raghavachari, G. W. Trucks, J. A. Pople, M. Head-Gordon, **A fifth-order perturbation comparison of electron correlation theories**, Chemical Physics Letters 157 (6) (1989) 479–483. doi:10.1016/S0009-2614(89)87395-6.
 URL <https://www.sciencedirect.com/science/article/pii/S0009261489873956>
- [15] K. Raghavachari, J. A. Pople, E. S. Replogle, M. Head-Gordon, **Fifth order Moeller-Plesset perturbation theory: comparison of existing correlation methods and implementation of new methods correct to fifth order**, The Journal of Physical Chemistry 94 (14) (1990) 5579–5586. doi:10.1021/j100377a033.
 URL <https://pubs.acs.org/doi/10.1021/j100377a033>
- [16] J. F. Stanton, **Why CCSD(T) works: a different perspective**, Chemical Physics Letters 281 (1) (1997) 130–134. doi:10.1016/S0009-2614(97)01144-5.
 URL <https://www.sciencedirect.com/science/article/pii/S0009261497011445>
- [17] J. D. Watts, J. Gauss, R. J. Bartlett, **Open-shell analytical energy gradients for triple excitation many-body, coupled-cluster methods: MBPT(4), CCSD+T(CCSD), CCSD(T), and QCISD(T)**, Chemical Physics Letters 200 (1) (1992) 1–7. doi:10.1016/0009-2614(92)87036-0.
 URL <https://www.sciencedirect.com/science/article/pii/0009261492870360>
- [18] M. Urban, J. Noga, S. J. Cole, R. J. Bartlett, **Towards a full CCSDT model for electron correlation**, The Journal of Chemical Physics 83 (8) (1985) 4041–4046, publisher: American Institute of Physics. doi:10.1063/1.449067.
 URL <https://aip.scitation.org/doi/abs/10.1063/1.449067>
- [19] K. Kowalski, P. Piecuch, **A comparison of the renormalized and active-space coupled-cluster methods: Potential energy curves of BH and F2**, Chemical Physics Letters 344 (1) (2001) 165–175. doi:10.1016/S0009-2614(01)00730-8.
 URL <https://www.sciencedirect.com/science/article/pii/S0009261401007308>
- [20] M. Kállay, P. R. Nagy, D. Mester, Z. Rolik, G. Samu, J. Csontos, J. Csóka, P. B. Szabó, L. Gyevi-Nagy, B. Hégely, I. Ladjánszki, L. Szegedy, B. Ladóczki, K. Petrov, M. Farkas, P. D. Mezei, d. Ganycz, **The MRCC program system: Accurate quantum chemistry from water to proteins**, The Journal of Chemical Physics 152 (7) (2020) 074107, publisher: American Institute of Physics. doi:10.1063/1.5142048.
 URL <https://aip.scitation.org/doi/full/10.1063/1.5142048>
- [21] M. Kállay, P. R. Surján, **Higher excitations in coupled-cluster theory**, The Journal of Chemical Physics 115 (7) (2001) 2945–2954, publisher: American Institute of Physics. doi:10.1063/1.1383290.
- URL <https://aip.scitation.org/doi/abs/10.1063/1.1383290>
- [22] L. Bytautas, T. Nagata, M. S. Gordon, K. Ruedenberg, **Accurate ab initio potential energy curve of F2. I. Nonrelativistic full valence configuration interaction energies using the correlation energy extrapolation by intrinsic scaling method**, The Journal of Chemical Physics 127 (16) (2007) 164317, publisher: American Institute of Physics. doi:10.1063/1.2800017.
 URL <https://aip.scitation.org/doi/full/10.1063/1.2800017>
- [23] K. Andersson, P. A. Malmqvist, B. O. Roos, A. J. Sadlej, K. Wolinski, **Second-order perturbation theory with a CASSCF reference function**, The Journal of Physical Chemistry 94 (14) (1990) 5483–5488. doi:10.1021/j100377a012.
 URL <https://pubs.acs.org/doi/10.1021/j100377a012>
- [24] K. Jankowski, J. Paldus, P. Piecuch, **Method of moments approach and coupled cluster theory**, Theoretica chimica acta 80 (4) (1991) 223–243. doi:10.1007/BF01117411.
 URL <https://doi.org/10.1007/BF01117411>
- [25] K. Kowalski, P. Piecuch, **The method of moments of coupled-cluster equations and the renormalized CCSD[T], CCSD(T), CCSD(TQ), and CCSDT(Q) approaches**, The Journal of Chemical Physics 113 (1) (2000) 18–35, publisher: American Institute of Physics. doi:10.1063/1.481769.
 URL <https://aip.scitation.org/doi/10.1063/1.481769>
- [26] T. J. Lee, G. E. Scuseria, **Achieving Chemical Accuracy with Coupled-Cluster Theory**, in: S. R. Langhoff (Ed.), **Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy**, Understanding Chemical Reactivity, Springer Netherlands, Dordrecht, 1995, pp. 47–108. doi:10.1007/978-94-011-0193-6_2.
 URL https://doi.org/10.1007/978-94-011-0193-6_2
- [27] J. Řezáč, L. Šimová, P. Hobza, **CCSD[T] Describes Noncovalent Interactions Better than the CCSD(T), CCSD(TQ), and CCSDT Methods**, Journal of Chemical Theory and Computation 9 (1) (2013) 364–369, publisher: American Chemical Society. doi:10.1021/ct3008777.
 URL <https://doi.org/10.1021/ct3008777>
- [28] P. Piecuch, M. Włoch, **Renormalized coupled-cluster methods exploiting left eigenstates of the similarity-transformed Hamiltonian**, The Journal of Chemical Physics 123 (22) (2005) 224105, publisher: American Institute of Physics. doi:10.1063/1.2137318.
 URL <https://aip.scitation.org/doi/10.1063/1.2137318>
- [29] R. J. Bartlett, J. Noga, **The expectation value coupled-cluster method and analytical energy derivatives**, Chemical Physics Letters 150 (1) (1988) 29–36. doi:10.1016/0009-2614(88)80392-0.
 URL <https://www.sciencedirect.com/science/article/pii/000926148803920>
- [30] W. Kutzelnigg, **Quantum chemistry in Fock space. I. The universal wave and energy operators**, The Journal of Chemical Physics 77 (6) (1982) 3081–3097, publisher: American Institute of Physics. doi:10.1063/1.444231.
 URL <http://aip.scitation.org/doi/10.1063/1.444231>
- [31] P.-D. Fan, P. Piecuch, **The Usefulness of Exponential Wave Function Expansions Employing One- and Two-Body Cluster Operators in Electronic Structure Theory: The Extended and Generalized Coupled-Cluster Methods**, in: J. R. Sabin, E. Brändas (Eds.), **Advances in Quantum Chemistry**, Vol. 51, Academic Press, 2006, pp. 1–57.

- [doi:10.1016/S0065-3276\(06\)51001-9](https://doi.org/10.1016/S0065-3276(06)51001-9).
 URL <https://www.sciencedirect.com/science/article/pii/S0065327606510019>
- [32] P. G. Szalay, M. Nooijen, R. J. Bartlett, Alternative ansätze in single reference coupled-cluster theory. III. A critical analysis of different methods, *The Journal of Chemical Physics* 103 (1) (1995) 281–298, publisher: American Institute of Physics. [doi:10.1063/1.469641](https://doi.org/10.1063/1.469641).
 URL <http://aip.scitation.org/doi/10.1063/1.469641>
- [33] F. A. Evangelista, Alternative single-reference coupled cluster approaches for multireference problems: The simpler, the better, *The Journal of Chemical Physics* 134 (22) (2011) 224102, publisher: American Institute of Physics. [doi:10.1063/1.3598471](https://doi.org/10.1063/1.3598471).
 URL <http://aip.scitation.org/doi/full/10.1063/1.3598471>
- [34] M. Schütz, H.-J. Werner, Low-order scaling local electron correlation methods. IV. Linear scaling local coupled-cluster (LCCSD), *The Journal of Chemical Physics* 114 (2) (2001) 661–681, publisher: American Institute of Physics. [doi:10.1063/1.1330207](https://doi.org/10.1063/1.1330207).
 URL <http://aip.scitation.org/doi/10.1063/1.1330207>
- [35] Z. Rolik, L. Szegedy, I. Ladjánszki, B. Ladóczki, M. Kállay, An efficient linear-scaling CCSD(T) method based on local natural orbitals, *The Journal of Chemical Physics* 139 (9) (2013) 094105, publisher: American Institute of Physics. [doi:10.1063/1.4819401](https://doi.org/10.1063/1.4819401).
 URL <https://aip.scitation.org/doi/full/10.1063/1.4819401>
- [36] J. F. Stanton, R. J. Bartlett, The equation of motion coupled-cluster method. A systematic biorthogonal approach to molecular excitation energies, transition probabilities, and excited state properties, *The Journal of Chemical Physics* 98 (9) (1993) 7029–7039, publisher: American Institute of Physics. [doi:10.1063/1.464746](https://doi.org/10.1063/1.464746).
 URL <http://aip.scitation.org/doi/abs/10.1063/1.464746>
- [37] P. Piecuch, K. Kowalski, I. S. O. Pimienta, M. J. McGuire, Recent advances in electronic structure theory: Method of moments of coupled-cluster equations and renormalized coupled-cluster approaches, *International Reviews in Physical Chemistry* 21 (4) (2002) 527–655, publisher: Taylor & Francis eprint: <https://doi.org/10.1080/0144235021000053811>. [doi:10.1080/0144235021000053811](https://doi.org/10.1080/0144235021000053811).
 URL <https://doi.org/10.1080/0144235021000053811>
- [38] B. Jeziorski, H. J. Monkhorst, Coupled-cluster method for multideterminantal reference states, *Physical Review A* 24 (4) (1981) 1668–1681, publisher: American Physical Society. [doi:10.1103/PhysRevA.24.1668](https://doi.org/10.1103/PhysRevA.24.1668).
 URL <https://link.aps.org/doi/10.1103/PhysRevA.24.1668>
- [39] I. Hubač, P. Neogrády, Size-consistent Brillouin-Wigner perturbation theory with an exponentially parametrized wave function: Brillouin-Wigner coupled-cluster theory, *Physical Review A* 50 (6) (1994) 4558–4564, publisher: American Physical Society. [doi:10.1103/PhysRevA.50.4558](https://doi.org/10.1103/PhysRevA.50.4558).
 URL <https://link.aps.org/doi/10.1103/PhysRevA.50.4558>
- [40] U. Sinha Mahapatra, B. Datta, D. Mukherjee, Molecular Applications of a Size-Consistent State-Specific Multireference Perturbation Theory with Relaxed Model-Space Coefficients, *The Journal of Physical Chemistry A* 103 (12) (1999) 1822–1830, publisher: American Chemical Society. [doi:10.1021/jp9832995](https://doi.org/10.1021/jp9832995).
 URL <https://doi.org/10.1021/jp9832995>
- [41] M. E. Harding, T. Metzroth, J. Gauss, A. A. Auer, Parallel Calculation of CCSD and CCSD(T) Analytic First and Second Derivatives, *Journal of Chemical Theory and Computation* 4 (1) (2008) 64–74, publisher: American Chemical Society. [doi:10.1021/ct700152c](https://doi.org/10.1021/ct700152c).
 URL <https://doi.org/10.1021/ct700152c>
- [42] J. F. Stanton, Many-body methods for excited state potential energy surfaces. I. General theory of energy gradients for the equation-of-motion coupled-cluster method, *The Journal of Chemical Physics* 99 (11) (1993) 8840–8847, publisher: American Institute of Physics. [doi:10.1063/1.465552](https://doi.org/10.1063/1.465552).
 URL <https://aip.scitation.org/doi/abs/10.1063/1.465552>
- [43] C. L. Lawson, R. J. Hanson, D. R. Kincaid, F. T. Krogh, *Basic Linear Algebra Subprograms for Fortran Usage*, *ACM Transactions on Mathematical Software* 5 (3) (1979) 308–323. [doi:10.1145/355841.355847](https://doi.org/10.1145/355841.355847).
 URL <https://doi.org/10.1145/355841.355847>
- [44] Z. Xianyi, W. Qian, Z. Yunquan, Model-driven Level 3 BLAS Performance Optimization on Loongson 3A Processor, in: 2012 IEEE 18th International Conference on Parallel and Distributed Systems, 2012, pp. 684–691, ISSN: 1521-9097. [doi:10.1109/ICPADS.2012.97](https://doi.org/10.1109/ICPADS.2012.97).
- [45] B. Aradi, *Fypp - Python powered Fortran metaprogramming* (Apr. 2021).
 URL <https://github.com/aradi/fypp>
- [46] D. G. A. Smith, L. A. Burns, A. C. Simmonett, R. M. Parrish, M. C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. Di Remigio, A. Alenaizan, A. M. James, S. Lehtola, J. P. Misiewicz, M. Scheurer, R. A. Shaw, J. B. Schriber, Y. Xie, Z. L. Glick, D. A. Sirianni, J. S. O'Brien, J. M. Waldrop, A. Kumar, E. G. Hohenstein, B. Pritchard, B. R. Brooks, H. F. Schaefer, A. Y. Sokolov, K. Patkowski, A. E. DePrince, U. Bozkaya, R. A. King, F. A. Evangelista, J. M. Turney, T. D. Crawford, C. D. Sherrill, *PSI4 1.4: Open-source software for high-throughput quantum chemistry*, *The Journal of Chemical Physics* 152 (18) (2020) 184108, publisher: American Institute of Physics. [doi:10.1063/5.0006002](https://doi.org/10.1063/5.0006002).
 URL <https://aip.scitation.org/doi/10.1063/5.0006002>
- [47] P. Pulay, Convergence acceleration of iterative sequences. the case of scf iteration, *Chemical Physics Letters* 73 (2) (1980) 393–398. [doi:10.1016/0009-2614\(80\)80396-4](https://doi.org/10.1016/0009-2614(80)80396-4).
 URL <https://www.sciencedirect.com/science/article/pii/0009261480803964>
- [48] F. Jensen, *Introduction to computational chemistry*, John Wiley & Sons, 2017.
- [49] G. E. Scuseria, T. J. Lee, H. F. Schaefer, Accelerating the convergence of the coupled-cluster approach: The use of the DIIS method, *Chemical Physics Letters* 130 (3) (1986) 236–239. [doi:10.1016/0009-2614\(86\)80461-4](https://doi.org/10.1016/0009-2614(86)80461-4).
 URL <https://www.sciencedirect.com/science/article/pii/0009261486804614>
- [50] G. M. J. Barca, C. Bertoni, L. Carrington, D. Datta, N. De Silva, J. E. Deustua, D. G. Fedorov, J. R. Gour, A. O. Gunina, E. Guidez, T. Harville, S. Irle, J. Ivanic, K. Kowalski, S. S. Leang, H. Li, W. Li, J. J. Lutz, I. Magoulas, J. Mato, V. Mironov, H. Nakata, B. Q. Pham, P. Piecuch, D. Poole, S. R. Pruitt, A. P. Rendell, L. B. Roskop, K. Ruedenberg, T. Sattasathuchana, M. W. Schmidt, J. Shen, L. Slipchenko, M. Sosonkina, V. Sundriyal, A. Tiwari, J. L. Galvez Vallejo, B. Westheimer, M. Wloch, P. Xu, F. Zahariev, M. S. Gordon, Recent developments in the general atomic and molecular electronic structure

- system, *The Journal of Chemical Physics* 152 (15) (2020) 154102. [doi:10.1063/5.0005188](https://doi.org/10.1063/5.0005188).
URL <http://aip.scitation.org/doi/10.1063/5.0005188>
- [51] M. Klemm, B. R. de Supinski, OpenMP Application Programming Interface Specification Version 5.0, 2018.
- [52] S. Chandrasekaran, G. Juckeland, *OpenACC for Programmers: Concepts and Strategies*, 1st Edition, Addison-Wesley Professional, 2017.
- [53] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, **Array programming with NumPy**, *Nature* 585 (7825) (2020) 357–362. [doi:10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
URL <https://www.nature.com/articles/s41586-020-2649-2>
- [54] D. G. a. Smith, J. Gray, **opt_einsum** - a python package for optimizing contraction order for einsum-like expressions, *Journal of Open Source Software* 3 (26) (2018) 753. [doi:10.21105/joss.00753](https://doi.org/10.21105/joss.00753).
URL <https://doi.org/10.21105/joss.00753>
- [55] J. Mabille, S. Corlay, W. Vollprecht, **xtensor**: Multi-dimensional arrays with broadcasting and lazy computing.
URL <https://github.com/xtensor-stack/xtensor>
- [56] R. Senanayake, C. Hong, Z. Wang, A. Wilson, S. Chou, S. Kamil, S. Amarasinghe, F. Kjolstad, **A sparse iteration space transformation framework for sparse tensor algebra**, *Proceedings of the ACM on Programming Languages* 4 (OOPSLA) (2020) 158:1–158:30. [doi:10.1145/3428226](https://doi.org/10.1145/3428226).
URL <https://doi.org/10.1145/3428226>
- [57] J. Turney, **ambit**.
URL <https://github.com/jturney/ambit>
- [58] A. Dutzler, **Tensor toolbox for modern fortran - high-level tensor manipulation in fortran**. [doi:10.5281/zenodo.4077378](https://doi.org/10.5281/zenodo.4077378).
URL <https://doi.org/10.5281/zenodo.4077378>
- [59] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, T. S. Woodall, Open MPI: Goals, concept, and design of a next generation MPI implementation, in: *Proceedings, 11th European PVM/MPI Users' Group Meeting*, Budapest, Hungary, 2004, pp. 97–104.
- [60] W. Gropp, **MPICH2: A New Start for MPI Implementations**, in: *Proceedings of the 9th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Springer-Verlag, Berlin, Heidelberg, 2002, p. 7.
- [61] M. Curcic, D. Rouson, *Modern Fortran: Building efficient parallel applications*, Manning Publications Company, 2020.
- [62] J. Nieplocha, R. J. Harrison, R. J. Littlefield, **Global arrays: A nonuniform memory access programming model for high-performance computers**, *The Journal of Supercomputing* 10 (2) (1996) 169–189. [doi:10.1007/BF00130708](https://doi.org/10.1007/BF00130708).
URL <https://doi.org/10.1007/BF00130708>
- [63] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. C. Whaley, *ScalAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.
- [64] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, W. A. de Jong, **NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations**, *Computer Physics Communications* 181 (9) (2010) 1477–1489. [doi:10.1016/j.cpc.2010.04.018](https://doi.org/10.1016/j.cpc.2010.04.018).
URL <https://www.sciencedirect.com/science/article/pii/S0010465510001438>
- [65] G. D. Fletcher, M. W. Schmidt, B. M. Bode, M. S. Gordon, **The Distributed Data Interface in GAMESS**, *Computer Physics Communications* 128 (1) (2000) 190–200. [doi:10.1016/S0010-4655\(00\)00073-4](https://doi.org/10.1016/S0010-4655(00)00073-4).
URL <https://www.sciencedirect.com/science/article/pii/S0010465500000734>
- [66] R. M. Olson, M. W. Schmidt, M. S. Gordon, A. P. Rendell, **Enabling the Efficient Use of SMP Clusters: The GAMESS/DDI Model**, in: *Proceedings of the 2003 ACM/IEEE conference on Supercomputing, SC '03*, Association for Computing Machinery, New York, NY, USA, 2003, p. 41. [doi:10.1145/1048935.1050191](https://doi.org/10.1145/1048935.1050191).
URL <https://doi.org/10.1145/1048935.1050191>
- [67] D. Datta, M. S. Gordon, **A Massively Parallel Implementation of the CCSD(T) Method Using the Resolution-of-the-Identity Approximation and a Hybrid Distributed/Shared Memory Parallelization Model**, *Journal of Chemical Theory and Computation* 17 (8) (2021) 4799–4822, publisher: American Chemical Society. [doi:10.1021/acs.jctc.1c00389](https://doi.org/10.1021/acs.jctc.1c00389).
URL <https://doi.org/10.1021/acs.jctc.1c00389>
- [68] G. J. Tornai, I. Ladjánszki, d. Rák, G. Kis, G. Cserey, **Calculation of Quantum Chemical Two-Electron Integrals by Applying Compiler Technology on GPU**, *Journal of Chemical Theory and Computation* 15 (10) (2019) 5319–5331, publisher: American Chemical Society. [doi:10.1021/acs.jctc.9b00560](https://doi.org/10.1021/acs.jctc.9b00560).
URL <https://doi.org/10.1021/acs.jctc.9b00560>
- [69] I. S. Ufimtsev, T. J. Martinez, **Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-Electron Integral Evaluation**, *Journal of Chemical Theory and Computation* 4 (2) (2008) 222–231, publisher: American Chemical Society. [doi:10.1021/ct700268q](https://doi.org/10.1021/ct700268q).
URL <https://doi.org/10.1021/ct700268q>
- [70] I. S. Ufimtsev, T. J. Martinez, **Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field Implementation**, *Journal of Chemical Theory and Computation* 5 (4) (2009) 1004–1015, publisher: American Chemical Society. [doi:10.1021/ct800526s](https://doi.org/10.1021/ct800526s).
URL <https://doi.org/10.1021/ct800526s>
- [71] I. S. Ufimtsev, T. J. Martinez, **Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics**, *Journal of Chemical Theory and Computation* 5 (10) (2009) 2619–2628, publisher: American Chemical Society. [doi:10.1021/ct9003004](https://doi.org/10.1021/ct9003004).
URL <https://doi.org/10.1021/ct9003004>
- [72] W. Ma, S. Krishnamoorthy, O. Villa, K. Kowalski, **GPU-Based Implementations of the Noniterative Regularized-CCSD(T) Corrections: Applications to Strongly Correlated Systems**, *Journal of Chemical Theory and Computation* 7 (5) (2011) 1316–1327, publisher: American Chemical Society. [doi:10.1021/ct1007247](https://doi.org/10.1021/ct1007247).
URL <https://doi.org/10.1021/ct1007247>
- [73] K. Bhaskaran-Nair, W. Ma, S. Krishnamoorthy, O. Villa, H. J. J. van Dam, E. Aprà, K. Kowalski, **Noniterative Multireference Coupled Cluster Methods on Heterogeneous CPU-GPU Systems**, *Journal of Chemical Theory and Computation* 9 (4) (2013) 1949–1957, publisher: American Chemical Society. [doi:10.1021/ct301130u](https://doi.org/10.1021/ct301130u).
URL <https://doi.org/10.1021/ct301130u>
- [74] R. Okuta, Y. Unno, D. Nishino, S. Hido, C. Loomis, **Cupy: A numpy-compatible library for nvidia gpu calculations**, in:

- Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS), 2017.
- URL http://learningsys.org/nips17/assets/papers/paper_16.pdf
- [75] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, software available from tensorflow.org (2015).
- URL <https://www.tensorflow.org/>
- [76] G. Ozen, G. Lopez, Accelerating Fortran DO CONCURRENT with GPUs and the NVIDIA HPC SDK.
- URL <https://developer.nvidia.com/blog/accelerating-fortran-do-concurrent-with-gpus-and-the-nvidia-hpc-sdk/>
- [77] A. P. Rendell, T. J. Lee, A. Komornicki, A parallel vectorized implementation of triple excitations in CCSD(T): application to the binding energies of the AlH₃, AlHF₂, AlH₂F, and AlF₃ dimers, *Chemical Physics Letters* 178 (5) (1991) 462–470. doi:[10.1016/0009-2614\(91\)87003-T](https://doi.org/10.1016/0009-2614(91)87003-T).
- URL <https://www.sciencedirect.com/science/article/pii/000926149187003T>
- [78] D. Stück, M. Head-Gordon, Regularized orbital-optimized second-order perturbation theory, *The Journal of Chemical Physics* 139 (24) (2013) 244109, publisher: American Institute of Physics. doi:[10.1063/1.4851816](https://doi.org/10.1063/1.4851816).
- URL <http://aip.scitation.org/doi/10.1063/1.4851816>
- [79] C. Li, F. A. Evangelista, Multireference Theories of Electron Correlation Based on the Driven Similarity Renormalization Group, *Annual Review of Physical Chemistry* 70 (1) (2019) 245–273, eprint: <https://doi.org/10.1146/annurev-physchem-042018-052416>. doi:[10.1146/annurev-physchem-042018-052416](https://doi.org/10.1146/annurev-physchem-042018-052416).
- URL <https://doi.org/10.1146/annurev-physchem-042018-052416>
- [80] A. J. W. Thom, M. Head-Gordon, Locating Multiple Self-Consistent Field Solutions: An Approach Inspired by Metadynamics, *Physical Review Letters* 101 (19) (2008) 193001, publisher: American Physical Society. doi:[10.1103/PhysRevLett.101.193001](https://doi.org/10.1103/PhysRevLett.101.193001).
- URL <https://link.aps.org/doi/10.1103/PhysRevLett.101.193001>
- [81] H. G. A. Burton, A. J. W. Thom, General Approach for Multireference Ground and Excited States Using Nonorthogonal Configuration Interaction, *Journal of Chemical Theory and Computation* 15 (9) (2019) 4851–4861, publisher: American Chemical Society. doi:[10.1021/acs.jctc.9b00441](https://doi.org/10.1021/acs.jctc.9b00441).
- URL <https://doi.org/10.1021/acs.jctc.9b00441>
- [82] J. H. Van Lenthe, R. Zwaans, H. J. J. Van Dam, M. F. Guest, Starting SCF calculations by superposition of atomic densities, *Journal of Computational Chemistry* 27 (8) (2006) 926–932, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.20393>. doi:[10.1002/jcc.20393](https://doi.org/10.1002/jcc.20393).
- URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20393>
- [83] S. Lehtola, Assessment of Initial Guesses for Self-Consistent Field Calculations. Superposition of Atomic Potentials: Simple yet Efficient, *Journal of Chemical Theory and Computation* 15 (3) (2019) 1593–1604, publisher: American Chemical Society. doi:[10.1021/acs.jctc.8b01089](https://doi.org/10.1021/acs.jctc.8b01089).
- URL <https://doi.org/10.1021/acs.jctc.8b01089>
- [84] G. H. Booth, D. Cleland, A. J. W. Thom, A. Alavi, Breaking the carbon dimer: The challenges of multiple bond dissociation with full configuration interaction quantum Monte Carlo methods, *The Journal of Chemical Physics* 135 (8) (2011) 084104, publisher: American Institute of Physics. doi:[10.1063/1.3624383](https://doi.org/10.1063/1.3624383).
- URL <http://aip.scitation.org/doi/full/10.1063/1.3624383>
- [85] C. Genovese, S. Sorella, The nature of the chemical bond in the dicarbon molecule, *The Journal of Chemical Physics* 153 (16) (2020) 164301, publisher: American Institute of Physics. doi:[10.1063/5.0023067](https://doi.org/10.1063/5.0023067).
- URL <https://aip-scitation-org.ezp.lib.cam.ac.uk/doi/full/10.1063/5.0023067>
- [86] M. Douay, R. Nietmann, P. F. Bernath, The discovery of two new infrared electronic transitions of C₂: B1Δg-A1Γu and the nivid_BΔg_AΔE_B, *Journal of Molecular Spectroscopy* 131 (2) (1988) 261–271. doi:[10.1016/0022-2852\(88\)90237-8](https://doi.org/10.1016/0022-2852(88)90237-8).
- URL <https://www.sciencedirect.com/science/article/pii/0022285288902378>
- [87] L. T. Xu, T. H. Dunning, Insights into the Perplexing Nature of the Bonding in C₂ from Generalized Valence Bond Calculations, *Journal of Chemical Theory and Computation* 10 (1) (2014) 195–201, publisher: American Chemical Society. doi:[10.1021/ct400867h](https://doi.org/10.1021/ct400867h).
- URL <https://doi.org/10.1021/ct400867h>
- [88] S. Wouters, W. Poelmans, P. W. Ayers, D. Van Neck, CheMPS2: A free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry, *Computer Physics Communications* 185 (6) (2014) 1501–1514. doi:[10.1016/j.cpc.2014.01.019](https://doi.org/10.1016/j.cpc.2014.01.019).
- URL <https://www.sciencedirect.com/science/article/pii/S0010465514000496>
- [89] D. P. Rodgers, Improvements in multiprocessor system design, *ACM SIGARCH Computer Architecture News* 13 (3) (1985) 225–231. doi:[10.1145/327070.327215](https://doi.org/10.1145/327070.327215).
- URL <https://doi.org/10.1145/327070.327215>
- [90] Q. Sun, Libcint: An efficient general integral library for gaussian basis functions, *Journal of Computational Chemistry* 36 (2015) 1664–1671. doi:[10.1002/jcc.23981](https://doi.org/10.1002/jcc.23981).
- URL <http://dx.doi.org/10.1002/jcc.23981>
- [91] K. Pearce, A. James, T. D. Crawford, D. Menéndez Crespo, B. G. Peyton, C++ Programming Tutorial in Chemistry.
- URL <https://github.com/CrawfordGroup/ProgrammingProjects>